Harmonious Balanced Partitioning of a Network of Agents

Pulkit Agarwal
Indian Institute of Technology Bombay
Mumbai, India
200050113@iitb.ac.in

Harshvardhan Agarwal
Indian Institute of Technology Bombay
Mumbai, India
200050050@iitb.ac.in

Vaibhav Raj
Indian Institute of Technology Bombay
Mumbai, India
200050148@iitb.ac.in

Swaprava Nath
Indian Institute of Technology Bombay
Mumbai, India
swaprava@iitb.ac.in

ABSTRACT

We consider the problem of balanced partitioning, i.e., dividing n agents into k groups of almost equal size ($\lfloor n/k \rfloor$ or $\lceil n/k \rceil$), where the agents form a friendship network, ensuring various fairness and efficiency criteria. The utility of an agent is the count of its friends in the same group as itself. When partitions into two groups are considered, we show that approximate envy-freeness related to the maximum degree of the graph can be obtained via a linear-time algorithm for arbitrary graphs. We also show that envy-freeness and core properties can be extended along with Pareto optimality in arbitrary graphs for such partitions. We then concentrate on the case of grid graphs having nodes on the 2D integer lattice, and demonstrate the impossibility of perfect envy-freeness. However, weaker guarantees like envy-freeness up to two friends are achievable for balanced k-partitions in a computationally efficient manner. We show that certain such balanced partitions belong to an exact and an approximate core when considering balanced 2-partitions.

KEYWORDS

Graphs; Balanced partition; Envy-freeness; Pareto optimality; Core.

ACM Reference Format:
Pulkit Agarwal, Harshvardhan Agarwal, Vaibhav Raj, and Swaprava Nath. 2025. Harmonious Balanced Partitioning of a Network of Agents. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Consider a school hosting a large sports event where multiple teams, formed from its students, compete for an overall prize. In such competitions, players are typically divided into (almost) equally-sized teams at random. However, a crucial aspect of team sports is the need for coordination among teammates, which adds an extra layer of complexity to the task.

Our study models situations like the one described above. Suppose we have symmetric friendship relations among n agents, represented by a network. The goal is to divide these n agents into k groups of nearly equal size, with an agent's utility defined as the number of friends they have within their own group. We call such a division a balanced k-partition (BP-k). In the example above, the higher utility of an agent in a partition is informally equivalent to the agent being able to coordinate better with their teammates.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

This problem of partitioning agents in an undirected friendship network into groups of almost equal size was introduced by Li et al. [18]. They posed it as an extension of the stable roommates problem [15], which requires an even number of agents and limits group size to exactly *two* agents. This problem typically considers preference profiles for each agent over all others. However, under our formulation, we focus only on binary preferences (friend or not), which are useful in practice where it may not be possible to obtain a full preference profile. Numerous works have also studied the three-dimensional generalization [3, 9, 14, 19, 20], where groups of *three* are formed.

Envy-freeness (EF) and efficiency in balanced k-partitions of graphs are natural considerations in this context. We begin by analyzing general graphs, establishing EF guarantees based on the maximum vertex degree. Our investigation reveals that even the seemingly simpler case of 2-partitions presents substantial technical challenges. However, graph bisections are crucial in various real-world applications, such as team partitions in sports, biological partitioning, and educational group formation, highlighting their practical importance. While Li et al. [18] conducted an in-depth study on trees, these structures are less representative of typical friendship networks. Consequently, we shift our focus to grid graphs, which offer more analytical tractability and stronger guarantees. Their unique combinatorial properties require the development of techniques distinct from those used by Li et al. [18]. They are also highly relevant in practical applications. For instance, Feldmann [11] emphasizes their importance in optimizing data distribution for parallel computing. Additionally, they serve as useful models for cities, where households (represented as agents) may prefer to be grouped with their neighbors for local activities such as cleanliness drives or community events.

1.1 Our Contributions

Our work builds on Li et al. [18] but introduces several key differences. We emphasize the role of graph structures, refine the concept of envy to make it more robust, and focus on balanced partitions meeting specific fairness criteria. Additionally, we examine Pareto optimality alongside envy-freeness and core, both individually and in combination.

Table 1: Summary of the results. The notation BP-k denotes a *balanced k-partition*, EF-r implies that no agent envies another by more than r, and GrG denotes a *grid graph*. For detailed definitions of the properties, please see Section 2.

	Results	**Guarantee**
General Graphs	$\forall G = (V, E)$, $\exists$ a EF-($\max\{\Delta(G) - 2, 2\}$) BP-2 (Theorem 1)	$O(\lvert V\rvert + \lvert E\rvert)$ algorithm (Algorithm 1)
	$\forall G = (V, E)$ and $\forall\ r \geqslant 0, \alpha \geqslant 1, \beta \geqslant 0$, 1. $\exists$ EF-r BP-2 $\implies$ $\exists$ (EF-r + PO) BP-2 2. $\exists$ (α, β)-core BP-2 $\implies$ $\exists$ ((α, β)-core + PO) BP-2 3. $\exists$ ((α, β)-core + EF-r) BP-2 $\implies$ $\exists$ ((α, β)-core + EF-r + PO) BP-2 (Theorems 2 and 3)	Existence
Grid Graphs	$\forall G = (V, E) \in$ GrG, $\exists$ a EF-2 BP-k, $\forall k \geqslant 2$ (Theorem 4) $\forall G = (V, E) \in$ GrG, $\exists$ a EF-2 BP-2 in $(1, 1)$-core (Theorem 5)	$O(\lvert V\rvert \log(\lvert V\rvert))$ algorithm (Algorithm 2)
	$\forall G = (V, E) \in$ GrG, $\exists$ a BP-2 in $(1, 0)$-core (Theorem 6)	Existence (Algorithm 3)

For any graph $G = (V, E)$ with $|V| = n$, we present a linear-time algorithm that constructs a BP-2 satisfying EF-($\max\{\Delta(G) - 2, 2\}$) for any arbitrary graph, where $\Delta(G)$ is the maximum degree of any vertex in G, and EF is defined in Definition 4. This result: (1) partially addresses the open question in [18] by proving the existence of a EF-2 BP-2 for graphs with $\Delta(G) \leqslant 4$, (2) improves the best-known result regarding the existence of a EF-($O(\sqrt{n})$) BP-2 [18] for graphs where $\Delta(G) \ll \sqrt{n}$, and (3) establishes a link between biconnected components and graph partitioning, offering a potential direction for future research on this problem. We also show that every graph admits a Pareto optimal (PO) BP-2 that can be constructed without compromising its fairness (EF) and stability (core) criteria.

For the special class of grid graphs, we show that a EF-2 BP-k can always be found via an efficient algorithm. For $k = 2$, the same algorithm returns a BP-2 that satisfies the additional guarantee of being in the $(1, 1)$-core (as defined in Definition 5). Furthermore, we prove that for grid graphs, a BP-2 in $(1, 0)$-core always exists. Table 1 highlights our major contributions. Due to space limitations, the detailed proofs of some results are available only in the full version of the paper [1].

1.2 Related Works

In the literature, most works have focused on finding minimum-cut balanced and unbalanced partitions [2, 12, 16, 22]. Li et al. [18] used min-cut partitions for a number of their results on the existence of balanced partitions in the core. Recent studies on balanced 2-partitioning of graphs (or *graph bisections*) have also looked at computational complexity and approximations of partitions in which every node has at least H neighbors in its own group [5, 6, 21]. If H is not constant, and is instead set to half of each node's degree, this aligns with Nash stability. Unlike these works, we develop an efficient algorithm to approximate the $H = 1$ case, which is used to prove that a EF-2 BP-2 exists for all graphs with $\Delta(G) \leqslant 4$.

A different dimension along which existing literature has varied is finding partitions with bounded group sizes. Levinger et al. [17] demonstrated the NP-completeness of the utility maximization problem in this setup, proposed a poly-time approximation algorithm for the task, and further explored stability through the notion of the core. Boehmer and Elkind [8] formulated the group-partitioning task from the viewpoint of diversity preferences, where agents belong to exactly one of two groups and each agent has a weak preference order on the composition of their group. Our work differs from these tasks both due to the complexity of forming a fixed-size coalition, and the presence of a direct binary preference order over all the participants. An orthogonal study was also done by Bilò et al. [7], that considered the notion of *swap stability* in fixed-size group partitions, defined as a partition in which no agent can improve its utility by swapping places with another agent without decreasing the utility of the other agent. Even though the underlying model is the same, our fairness notions differ significantly, with the notion of envy-freeness being stronger than swap stability as the latter does not allow swaps (or transfers) in which the utility of an agent increases at all.

One of the most overlooked aspects in balanced partitioning is that of Pareto optimality. Aziz et al. [4] were the first to introduce Pareto optimality in the context of coalition formation games, presenting various results on its computational intractability across different game classes, while Li et al. [18] only demonstrated that some of their BP-k algorithms fail to achieve PO. To the best of our knowledge, this is the first work to investigate the existence of PO bisections alongside other fairness guarantees.

2 PRELIMINARIES

Define $[\ell] = \{1, 2, \ldots, \ell\}$. Consider a group of agents (denoted by set V with $|V| = n$) connected via an undirected graph $G = (V, E)$, where each edge $(i, j) \in E$ represents that i and j are neighbors in G. The degree of node i in G is denoted by $d_G(i)$, and $\Delta(G) = \max_{i \in V} d_G(i)$ denotes the maximum degree of the graph G. The set of neighbors of agent i in G is denoted by $N_G(i)$. The subgraph of G induced by a set of vertices $S \subseteq V$ is denoted by $G[S]$.

Definition 1 (k-partition). *A k-partition of a graph $G = (V, E)$ is a set of k mutually exclusive and exhaustive subsets of V. Formally, it is the collection $Z = \{Z_1, Z_2, \ldots, Z_k\}$ s.t.*

$$Z_\ell \subseteq V, \forall \ell \in [k], Z_{\ell_1} \cap Z_{\ell_2} = \emptyset, \forall \ell_1 \neq \ell_2, \text{ and } \bigcup_{\ell \in [k]} Z_\ell = V$$

Definition 2 (Balanced k-partition). *A* balanced k-partition $Z = \{Z_1, Z_2, \ldots, Z_k\}$ *of a graph* $G = (V, E)$ *is a k-partition satisfying* $|Z_\ell| \in \{\lfloor n/k \rfloor, \lceil n/k \rceil\}, \forall \ell \in [k]$.

We will use the shorthand BP-k to mention a balanced k-partition. The set of neighbors of agent i in a k-partition Z is given by $N_G^Z(i) = \{j \in N_G(i) : i, j \in Z_\ell,\ \text{for some } \ell \in [k]\}$. The *utility* of agent i in a k-partition Z is defined as $u_i^G(Z) = |N_G^Z(i)|$, i.e., the number of neighbors of i in G that belong to the same subset as i in the partition Z. Wherever it is clear from context, we will drop G from every notation that uses it. The subset in which agent i belongs in a balanced k-partition Z is denoted by $Z^{-1}(i)$, i.e., $Z^{-1}(i) = \ell$, s.t. $i \in Z_\ell$.

Define a *swapping partition* of a given k-partition Z between two agents i and j as the new k-partition Z' such that

$$Z'_{Z^{-1}(i)} = (Z_{Z^{-1}(i)} \setminus \{i\}) \cup \{j\}$$
$$Z'_{Z^{-1}(j)} = (Z_{Z^{-1}(j)} \setminus \{j\}) \cup \{i\}$$
$$Z'_{Z^{-1}(p)} = Z_{Z^{-1}(p)}\ ,\ \forall p \in V \setminus \{i, j\}.$$

In other words, the swapping partition Z' swaps the subsets in which agents i and j belonged in Z. We will denote the swapping partition Z' with the notation $\mathbf{swap}(Z, i, j)$. It is easy to see that, $\mathbf{swap}(Z, i, j)$ is non-trivial only when $Z^{-1}(i) \neq Z^{-1}(j)$, i.e., i and j are in different subsets of Z.

Define a *transfer partition* of a given k-partition Z for agent i to another subset $Z_\ell, \ell \neq Z^{-1}(i), |Z_\ell| < |Z_{Z^{-1}(i)}|$ as the new k-partition Z' such that

$$Z'_{Z^{-1}(i)} = Z_{Z^{-1}(i)} \setminus \{i\}$$
$$Z'_\ell = Z_\ell \cup \{i\}$$
$$Z'_p = Z_p\ ,\ \forall p \neq Z^{-1}(i), \ell.$$

In simpler terms, the transfer moves an agent i from its original larger subset to a smaller subset. We will denote such a transferred partition Z' with the notation $\mathbf{tran}(Z, i, \ell)$.

We start off with an observation, used in defining envy.

Observation 1. *Given a pair of agents (i, j) and a* BP-k Z, $\mathbf{swap}(Z, i, j)$ *is also a* BP-k. *Also, if* $|Z_{Z^{-1}(j)}| < |Z_{Z^{-1}(i)}|$, *then* $\mathbf{tran}(Z, i, Z^{-1}(j))$ *is also a* BP-k, *and*

$$u_i\left(\mathbf{tran}(Z, i, Z^{-1}(j))\right) \geqslant u_i\left(\mathbf{swap}(Z, i, j)\right)$$

Proof. Note that both **swap** and **tran** are k-partitions. Here, **swap** is a BP-k as the sizes of all subsets in the new partition remain the same. And **tran** is a BP-k as we transfer node i from a larger subset of size $\lceil n/k \rceil$ to a smaller subset of size $\lfloor n/k \rfloor$, which simply exchanges the sizes of these subsets. Finally, taking $Z^{-1}(j) = \ell$,

$$u_i\left(\mathbf{tran}(Z, i, \ell)\right) = \begin{cases} u_i\left(\mathbf{swap}(Z, i, j)\right) + 1 & ;\ (i, j) \in E \\ u_i\left(\mathbf{swap}(Z, i, j)\right) & ;\ (i, j) \notin E \end{cases}$$

since j is in i's subset after **tran**, but not after **swap**. □

We can now define envy and envy-freeness as follows.

Definition 3 (Envy). *In a* BP-k X, *an agent i has an envy $r \geqslant 0$ towards another agent j if*

1. *case* $|X_{\ell_j}| < |X_{\ell_i}| : u_i(\mathbf{tran}(X, i, \ell_j)) - u_i(X) = r$.
2. *case* $|X_{\ell_j}| \geqslant |X_{\ell_i}| : u_i(\mathbf{swap}(X, i, j)) - u_i(X) = r$.

where $\ell_i = X^{-1}(i)$ *and* $\ell_j = X^{-1}(j)$.

The definition says that if agent i thinks that a swap or transfer with another agent outside its own subset in a partition can increase its utility, then it is envious of that agent. Here, if **tran** is possible, then it is preferred over **swap**, as that yields a weakly better utility for the agent (Observation 1).

Definition 4 (EF-r). *For* $r \geqslant 0$, *a* BP-k X *is called* envy-free up to r (EF-r)*, if for every pair of agents* $i, j \in V$, *i envies j by at most r.*

Li et al. [18] only considered swaps as part of their definition of envy, but because of Observation 1, Definition 4 provides stronger assurances against any form of envy. This also ensures that all our results still hold in purview of the definition used by Li et al. [18]. This adjustment was primarily motivated by the fact that when an agent envies their neighbor, it is because they see their neighbor's group as better and would prefer to join it if possible. The utility gained from being with the neighbor is merely a bonus, as it still resolves their initial envy. We also make another useful observation about envy below.

Observation 2. *If agent i envies agent j by $r > 0$ in* BP-k X, *then* $X^{-1}(j) \neq X^{-1}(i)$ *and i has at least r neighbors in* $X_{X^{-1}(j)}$.

Proof. Since agent i has an envy $r > 0$ towards agent j in BP-k X, utility of agent i increases from its current utility $u_i(X)$ by r. This increase can either occur by a **tran** or a **swap** operation performed by i from $X_{X^{-1}(i)}$ to $X_{X^{-1}(j)}$ (and these two subsets must be different). In both cases, it is necessary that there exist at least r neighbors of i in $X_{X^{-1}(j)}$. □

Definition 5 (Core). *For* $\alpha \geqslant 1, \beta \geqslant 0$, *a* BP-$k$ X *is said to be in* (α, β)-core *if there does not exist another* BP-k X' *and an index* $\ell \in [k]$ *s.t.* $\forall i \in X'_\ell\ ,\ u_i(X') > \alpha \cdot u_i(X) + \beta$.

This definition only requires the utility of agents in X'_ℓ to increase. If, for some BP-k X, such a coalition X'_ℓ exists, we call X'_ℓ an (α, β)-*blocking coalition* of BP-k X. Wherever clear from context, we simply call X'_ℓ a blocking coalition.

A BP-k is said to be in the *core* for graph G if it satisfies $(1, 0)$-core property as defined above. For $k = 2$, the existence of core means that there is no other BP-2 $(S, V \setminus S)$, such that the utility of all nodes in S strictly increases in this new partition. Also note that $(1, 0)$-core is stronger than (α, β)-core for any other choice of $\alpha \geqslant 1, \beta \geqslant 0$.

Definition 6 (Pareto Optimal). *A* BP-k X *is called* Pareto optimal (PO)*, if there does not exist another* BP-k X' *such that for all nodes* $i \in V$, *we have* $u_i(X') \geqslant u_i(X)$, *and the strict inequality holds for at least one node* $j \in V$.

If X is not Pareto optimal, then the BP-k X' is called a *Pareto improvement* to partition X.

2.1 Graphs of interest

In this section, we provide a brief review of certain types of graphs and their properties that will be used later in the paper. We use the standard graph-theoretic terminology. Let $\overline{G}$ be the complement of graph G. We denote by K_n, C_n, and P_n, the complete undirected graph on n vertices, a simple cycle with n vertices, and a path graph with n vertices respectively. Also, we define the *comb graph*[1] on $2n$ vertices as $P_n \odot K_1$, i.e., the graph constructed by connecting n vertices in a path, each of which is connected to a pendant edge. This graph consists of $2n-1$ edges, and is denoted by $\mathbf{comb}_n$.

Biconnected graphs. A *biconnected graph* $G = (V, E)$ with $|V| > 1$ is a connected graph for which the subgraph $G[V \setminus \{v\}]$ remains connected for all $v \in V$.

Definition 7 (st-Numbering). *An st-numbering of $G = (V, E)$ with $|V| = n > 1$ and two vertices $s, t \in V$ is a mapping from V to $N = [n]$, such that the source s is labeled 1, the sink t is labeled n, and every vertex $i = 2, \dots, n-1$ is adjacent both to some vertex $h < i$ and to some vertex $j > i$.*

It is well known that for any biconnected graph with any arbitrarily chosen source s and sink t, an st-numbering exists and can be found in linear time [10]. A connected graph that is not a biconnected graph will have *cut vertices*, i.e. a vertex whose removal causes the graph to become disconnected. A *biconnected component* is a maximal biconnected subgraph in a graph.

Definition 8 (Block-Cut Tree). *Any connected graph decomposes into a tree with biconnected components (or blocks) and cut vertices as vertices, called the* block-cut tree *of the graph, where the blocks are attached to shared cut vertices.*

Hopcroft and Tarjan [13] gave a linear-time algorithm to find the block-cut tree for any connected graph.

Definition 9 (Leafy-Cut Vertex). *A* leafy-cut vertex *in a rooted block-cut tree is a cut vertex that does not have any other cut vertex as its descendant.*

A leafy-cut vertex z has a non-zero number of descendants which can be partitioned into sets of vertices $b_1, \dots, b_k$ ($k \geqslant 1$) such that $\forall i \in [k]$, $|b_i| > 0$ and $G[b_i \cup \{z\}]$ is a biconnected graph. We refer to each b_i as a *biconnected block.*

Grid graphs. A grid graph is a collection of nodes on the integer 2D coordinate lattice with possible edges between them only if the nodes are adjacently placed on the lattice.

Definition 10 (Grid Graph). *$G = (V, E)$ is a grid graph if we can create mappings $X : V \to \mathbb{Z}$ and $Y : V \to \mathbb{Z}$ such that for each pair of nodes $i, j \in V$, where $(X(i), Y(i)) \neq (X(j), Y(j))$,*

$$[(i, j) \in E] \implies [|X(i) - X(j)| + |Y(i) - Y(j)| = 1],$$

where X and Y are the mappings to the x and y-coordinates of the nodes on the lattice.

We will denote the set of grid graphs with GrG. We are now ready to present the main results.

3 MAIN RESULTS: GENERAL GRAPHS

We first look at envy-freeness and Pareto optimality properties for arbitrary graphs. For both these properties, we will need a lemma that formally shows the necessary and sufficient conditions for the existence of a EF-r BP-2 [2].

[1] The name of this graph comes from the fact that it can be drawn like a comb.
[2] From now on, we will omit G from our notations when it is clear from the context.

Lemma 1. *A* BP-2 X *in* $G = (V, E)$ *is* EF-r

(1) *if*

$$|N^X(i)| \geqslant \frac{d(i) - r}{2}, \ \forall i \in V, \ \textit{and,} \quad (1)$$

(2) *only if*

$$|N^X(i)| \geqslant \frac{d(i) - r - 1}{2}, \ \forall i \in V. \quad (2)$$

We state the conditions 1 and 2 separately since both of them will be used for proving the upcoming results.

Proof. (Condition 1: sufficiency): Let X be a BP-2. Suppose X is not EF-r. Hence, $\exists i, j \in V$ such that i has an envy $> r$ towards j. Case 1: $|X_{X^{-1}(i)}| > |X_{X^{-1}(j)}|$: here the violation of EF-r implies $u_i(\mathbf{tran}(X, i, X^{-1}(j))) - u_i(X) > r$. This implies, $(d(i) - |N^X(i)|) - |N^X(i)| > r$. This violates the condition in Equation (1). Case 2: $|X_{X^{-1}(i)}| \leqslant |X_{X^{-1}(j)}|$: here the violation of EF-r implies $u_i(\mathbf{swap}(X, i, j)) - u_i(X) > r$. Here, two subcases can occur, (a) i and j are neighbors in G: then $(d(i) - |N^X(i)| - 1) - |N^X(i)| > r$ (the additional -1 appears since j also changes its subset in a **swap**), and (b) i and j are not neighbors in G: then $(d(i) - |N^X(i)|) - |N^X(i)| > r$. Both these subcases violate the condition in Equation (1).

(Condition 2: necessity): Suppose X is a EF-r BP-2. This implies that $\forall i, j \in V$, i has an envy $\leqslant r$ towards j. Now consider the cases 1 and 2 above. Carrying out similar calculations we get that $(d(i) - |N^X(i)|) - |N^X(i)| \leqslant r$ for case 1 and the second subcase of case 2. But for the first subcase of case 2 (where $(i, j) \in E$), we get $(d(i) - |N^X(i)| - 1) - |N^X(i)| \leqslant r$. Hence the condition that satisfies *all* these conditions (and hence necessary) is given by Equation (2). □

3.1 Envy-Freeness

Theorem 1. *For every graph G, there exists a* BP-2 *that is* EF-$(\max\{\Delta(G) - 2, 2\})$*, which can be found in linear time.*

We first derive a corollary of Lemma 1 to get a sufficient condition for Theorem 1.

Lemma 2. *A* BP-2 X *of* $G = (V, E)$ *is* EF-$(\max\{\Delta - 2, 2\})$ *if for all nodes $i \in V$, at least one of the following two conditions holds: (1) $d(i) \leqslant 2$, or (2) $|N^X(i)| \geqslant 1$.*

Proof. For $d(i) \leqslant 2$, we know that $\frac{d(i)-2}{2} \leqslant 0 \leqslant |N^X(i)|$. Otherwise $|N^X(i)| \geqslant 1 \geqslant \frac{d(i)-(\Delta-2)}{2}$. Thus, $\forall i \in V, |N^X(i)| \geqslant \frac{d(i)-\max\{\Delta-2,2\}}{2}$. Following condition 1 of Lemma 1, we conclude that X is EF-$(\max\{\Delta - 2, 2\})$. □

By Lemma 2, to obtain a EF-$(\max\{\Delta - 2, 2\})$ BP-2, it suffices to ensure that every vertex with degree > 2 has at least 1 neighbor in the same group. For simplicity, we refer to the two subsets of the partition by the colors red and blue. We construct the BP-2 sequentially, first for biconnected graphs, then for connected graphs, and finally for arbitrary graphs.

Lemma 3. *For any biconnected graph $G = (V, E)$ and $r, b > 1$ with $r + b = |V|$, Algorithm 1 returns a 2-partition $X = (\mathcal{R}, \mathcal{B})$ with $|\mathcal{R}| = r, |\mathcal{B}| = b$ s.t. $|N^X(i)| \geqslant 1, \ \forall i \in V$.*

Algorithm 1 ConnPart: Block – Cut Partition

Input: Connected graph $G = (V, E)$, integers r, b
Require: $r, b > 1;\ r + b = |V|$
Output: $X = \{\mathcal{R}, \mathcal{B}\}$, $|\mathcal{R}| = r$, $|\mathcal{B}| = b$

1: **if** G is biconnected **then**
2: $\quad S \leftarrow$ **st-numbering**(G); $\mathcal{R} \leftarrow \{v \in V \mid S(v) \leqslant r\}$
3: $\quad$ **return** $X = \{\mathcal{R},\ V \setminus \mathcal{R}\}$
4: **end if**
5: $\mathcal{T} \leftarrow$ **BlockCutTree**(G) rooted at a cut vertex c
6: $\mathcal{R} \leftarrow \emptyset$; $V_{\text{rem}} \leftarrow V$
7: **StartLoop:**
8: $z \leftarrow$ Arbitrary leafy-cut vertex of $\mathcal{T}[V_{\text{rem}}]$
9: z has k biconnected blocks $b_1, b_2, \ldots, b_k$ as children
10: $C \leftarrow \bigcup_{i=1}^{k} b_i$ // b_i does not contain z, $\forall i \in [k]$
11: **if** $r > |C| + 1$ **then**
12: $\quad$ // Color all children of z red, and trim the graph
13: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup C$; $r \leftarrow r - |C|$; $V_{\text{rem}} \leftarrow V \setminus \mathcal{R}$
14: $\quad$ **goto** StartLoop
15: **else if** $r = |C| + 1$ **then**
16: $\quad$ // Color z and all its children red
17: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup C \cup \{z\}$
18: **else if** $r < |C| + 1$ **then**
19: $\quad G \leftarrow G[V_{\text{rem}}]$; $\mathcal{T} \leftarrow \mathcal{T}[V_{\text{rem}}]$
20: $\quad$ // If r is odd, we first color odd nodes to make r
21: $\quad$ // even. Pairs are then selected from the b_i's.
22: $\quad$ // Finally, an odd element of some b_i's may be
23: $\quad$ // chosen till the requirement of r nodes is met.
24: $\quad \mathcal{R} \leftarrow \mathcal{R} \cup$ SelectPairsFirst$(G, \mathcal{T}, z, \{b_i\}, r, b)$
25: **end if**
26: **return** $X = (\mathcal{R},\ V \setminus \mathcal{R})$

For connected graphs $G = (V, E)$, we use their rooted block-cut tree $\mathcal{T}$ to color the graph appropriately in Algorithm 1. We first color all vertices in V blue, and then change the color of $r > 1$ vertices to red to get a desired 2-partition X. Here, in each iteration (starting at Line 7 of Algorithm 1), we color all the biconnected children blocks of some leafy-cut vertex red (Line 13), till r nodes have been colored red. Lemma 4 deals with the case when the remaining nodes, which are to be colored red, fit within the subtree of a leafy-cut vertex z (Line 18). In this case, we call the SelectPairsFirst procedure (exact algorithm given in the full version of this paper [1]), which colors some of the nodes red in this subtree of z, while satisfying Equation (3) below for all nodes. The coloring is done by using an st-numbering of the biconnected blocks with the source as node z.

Lemma 4. *For a connected graph $G = (V, E)$ and its rooted block-cut tree $\mathcal{T}$, let z be a leafy-cut vertex in $\mathcal{T}$ with biconnected blocks $\{b_i\}_{i=1}^{k}$ as its children. Then for any $1 < r \leqslant \Sigma_{i=1}^{k}|b_i|,\ b > 1,\ r + b = |V|$, $\exists$ 2-partition $X = (\mathcal{R}, \mathcal{B})$ with $|\mathcal{R}| = r, |\mathcal{B}| = b$ such that $\forall\ i \in V$,*

$$\text{either } d(i) = 1, \text{ or } |N^X(i)| \geqslant 1 \tag{3}$$

Also, for any $j \in [k]$,

$$|b_j| = 1 \text{ and } b_j \subseteq \mathcal{B} \implies z \in \mathcal{B}$$

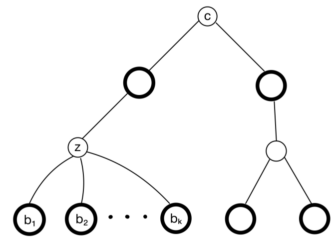

Figure 1: Block-cut tree $\mathcal{T}$ rooted at a cut vertex c, with some leafy-cut vertex z having children biconnected blocks $\{b_i\}_{i=1}^{k}$.

This 2-partition can be found in $O(|V| + |E|)$ time using the SelectPairsFirst$\left(G, \mathcal{T}, z, \{b_i\}_{i=1}^{k}, r, b\right)$ *procedure.*

The following lemma shows why Algorithm 1 satisfies the condition of the lemma above.

Lemma 5. *For any connected graph $G = (V, E)$ and $r, b > 1$ with $r + b = |V|$, Algorithm 1 returns a 2-partition $X = (\mathcal{R}, \mathcal{B})$ s.t. $|\mathcal{R}| = r, |\mathcal{B}| = b$ and $\forall i \in V$, Equation (3) holds.*

Proof. If G is biconnected, then we are done by Lemma 3. Suppose G is not biconnected. Let $\mathcal{T}$ be the block-cut tree of G rooted at some cut vertex c. Suppose z is the chosen leafy-cut vertex on Line 8, with $k \geqslant 1$ biconnected blocks $\{b_i\}_{i=1}^{k}$ as children. An example of this setup is shown in Figure 1. Also let $C = \cup_{i=1}^{k} b_i$, and note that $|C| \geqslant 1$ (since every biconnected block has at least 1 vertex). Here, the graph being partitioned is $G[V_{\text{rem}}]$ (initially, $V_{\text{rem}} = V$). In each iteration (Line 7), there are 3 possible situations:

Case 1: $r \leqslant |C|$: We apply the SelectPairsFirst procedure (Line 24). By Lemma 4, every $i \in V_{\text{rem}}$ must have either $d_{G[V_{\text{rem}}]}(i) = 1$ or $|N^X(i)| \geqslant 1$. If Equation (3) is not satisfied by i in the final partition X of G, then the second condition cannot hold, giving $d_{G[V_{\text{rem}}]}(i) = 1$ (i.e. $|b_j| = 1$) and $d_G(i) > 1$. Thus, i is a cut vertex in G, and so it will have a pre-colored red child (from some previous iteration). If i is red, then this child is a same-colored neighbor of i. Else if i is blue, then Lemma 4 gives that its parent z must also be blue. Hence, we reach a contradiction in both situations, and every node $i \in V_{\text{rem}}$ satisfies Equation (3).

Case 2: $r = 1 + |C|$: z and all its children are colored red (Line 17). Note that $G[C \cup \{z\}]$ and $G[V_{\text{rem}} \setminus (C \cup \{z\})]$ are red and blue colored connected graphs respectively, each having size > 1. Thus, every node in V_{rem} has at least 1 neighbor in the same group, satisfying Equation (3).

Case 3: $r > 1 + |C|$: Color all nodes in C red (Line 13), and reduce r to $r' = r - |C|$. For any $j \in [k]$, there are 2 cases:

(1) $|b_j| > 1$, so that every node $i \in b_j$ has a same-colored red neighbor in b_j (since $G[b_j]$ is connected).
(2) $|b_j| = 1$, so that there is only one node $v \in b_j$. If $d_G(v) = 1$ (i.e. v is only joined to z in G), then v already satisfies Equation (3). Otherwise, if v is a cut vertex in G, then v (which is colored red) must have a pre-colored red child (from some previous iteration), giving $|N^X(v)| \geqslant 1$.

Thus, Equation (3) holds for all $i \in C$. The graph is now trimmed to $G' = G[V_{\text{rem}} \setminus C]$, which is still a connected graph. And the problem reduces to finding a 2-partition $X' = (\mathcal{R}', \mathcal{B}')$ in this connected graph G' with $|\mathcal{R}'| = r', |\mathcal{B}'| = b$ (where r', b are still > 1 and $r' < r$). We now repeat the same procedure for this subgraph. □

Lemmas 2 and 5 together give the desired BP-2 for connected graphs, as given in Lemma 6 below.

Lemma 6. *For any connected graph $G = (V, E)$, there exists a* BP-2 $X = (\mathcal{R}, \mathcal{B})$ *with* $|\mathcal{R}| \geq |\mathcal{B}|$ *saitsfying* EF-$(\max\{\Delta - 2, 2\})$.

Proof. By Lemmas 2 and 5, for any connected graph with $|V| \geqslant 4$, taking $|\mathcal{R}| = r = \lceil |V|/2 \rceil$ in Algorithm 1 returns the desired EF-$(\max\{\Delta - 2, 2\})$ BP-2. And, if $|V| \leqslant 3$, then Observation 2 ensures the existence of a EF-2 BP-2 (since any node has $\leqslant 2$ neighbors, so it cannot have envy > 2). □

Proof. (of Theorem 1) Divide an arbitrary graph G into its connected components $c_1, c_2, \ldots, c_m$. For $j \in [m]$, let $G_j = \bigcup_{i=1}^{j} c_i$, with $G_m = G$. We will use induction on $j \in [m]$ to get a EF-$(\max\{\Delta - 2, 2\})$ BP-2 for all G_j. By Lemma 6, G_1 has such a partition. For the inductive step, suppose the desired BP-2 X exists for G_{j-1} $(j > 1)$. If there are more blue nodes in X than red nodes, then color c_j as per Lemma 6, with at least as many red nodes as blue nodes. Else color c_j using the same algorithm and then reverse its colors so that it has more blue nodes than red nodes. This ensures that the absolute difference between the number of red and blue nodes in this new 2-partition of $G_j = G_{j-1} \cup c_j$ is at most 1, giving us a BP-2 of G_j. Also, since the partition in each component is EF-$(\max\{\Delta - 2, 2\})$ and there are no edges between components, this final BP-2 will also be EF-$(\max\{\Delta - 2, 2\})$, completing the induction and our proof for Theorem 1.

(Complexity): As per the above proof, to find the final BP-2, we have to run Algorithm 1 for each c_j, $j \in [m]$ once. Algorithm 1 requires finding the block-cut tree of a connected graph and the st-numbering within different biconnected components, both of which are linear-time algorithms. SelectPairsFirst procedure is also called only once (in Line 24 of Algorithm 1), and it is also linear time according to Lemma 4. This ensures that for each connected component of G, Algorithm 1 runs in linear time, making the overall algorithm finish in $O(|V| + |E|)$ time. □

3.2 Pareto Optimality

Theorem 2. *For every graph G with a* BP-2 X *that is* EF-r *for some $r \geqslant 0$, there exists a* BP-2 X' *that is* EF-r *and* PO.

Proof. Suppose some graph G has a BP-2 X that is EF-r. If X is PO, then we are done. Otherwise, we perform a sequence of Pareto improvements to X till we get a PO BP-2 X'. We claim that X' is EF-r. Suppose X' is not EF-r. Then by condition 1 of Lemma 1, $\exists i \in V$ such that $d(i) - r > 2|N^{X'}(i)|$. Since all variables in this inequality are natural numbers, this implies that $d(i) - r - 1 \geqslant 2|N^{X'}(i)|$ or $|N^{X'}(i)| \leqslant \frac{d(i)-r-1}{2}$. However, since X is EF-r, by condition 2 of Lemma 1, we have $|N^{X}(i)| \geqslant \frac{d(i)-r-1}{2}$. The partition X' is obtained by sequential Pareto improvements from X, and hence $|N^{X}(i)| \leqslant |N^{X'}(i)|$. Combining these three inequalities, we get

$$|N^{X}(i)| = \frac{d(i) - r - 1}{2} = |N^{X'}(i)|. \tag{4}$$

Suppose the BP-2's are given by $X = (\mathcal{R}, \mathcal{B})$ and $X' = (\mathcal{R}', \mathcal{B}')$, and WLOG $i \in \mathcal{R}, \mathcal{R}'$. From Equation (4), we have $(d(i) - |N^{X}(i)|) - |N^{X}(i)| = r + 1$, i.e., 'the number of blue ($\mathcal{B}$) neighbors of i' minus 'the number of red ($\mathcal{R}$) neighbors of i' is $r+1$. Consider the following two possible cases.

1. $|\mathcal{B}| = \lfloor n/2 \rfloor < \lceil n/2 \rceil$: According to Definition 3, i can apply a **tran** operation from $\mathcal{R}$ to $\mathcal{B}$ and increase its utility by $r + 1$, which contradicts X being EF-r.
2. $|\mathcal{B}| = \lceil n/2 \rceil$: Consider an arbitrary node $j \in \mathcal{B}$ and a possible **swap** of i with j. If j is not a neighbor of i, then on swapping groups with j, the utility of i increases by $r + 1$, which contradicts X being EF-r. So every $j \in \mathcal{B}$ can only be a neighbor of i, or equivalently, $\mathcal{B} = N(i) \setminus N^{X}(i)$. But, we also have $|N^{X}(i)| = |N^{X'}(i)|$, which gives $\lceil n/2 \rceil = |B| = d(i) - |N^{X}(i)| = d(i) - |N^{X'}(i)| \leqslant |\mathcal{B}'|$, where the last inequality comes from the fact that the blue neighbors of i must be at most the whole of $\mathcal{B}'$. Since X' is a BP-2, $|\mathcal{B}'|$ is also bounded from above by $\lceil n/2 \rceil$, and hence $|\mathcal{B}'| = d(i) - |N^{X'}(i)| = \lceil n/2 \rceil \Rightarrow \mathcal{B}' = N(i) \setminus N^{X'}(i)$.
   Thus, we have $|\mathcal{R}'| = \lfloor n/2 \rfloor$, which means that only the **swap** operation is possible for i in X'. Pick an arbitrary $j' \in \mathcal{B}'$ for i to **swap** with. This **swap** leads to a change in utility of i from $|N^{X'}(i)|$ to $d(i) - |N^{X'}(i)| - 1$, which is $d(i) - 2|N^{X'}(i)| - 1 = r$. This contradicts the fact that i has an envy $> r$ with some $j' \in \mathcal{B}'$.

Since both conclusions above lead to a contradiction, our original assumption on X' not being EF-r must be false, and X' is the desired PO EF-r partition. □

Theorem 3. *Consider an arbitrary graph G that has a* BP-2 X *which belongs to the (α, β)-core for some $\alpha \geqslant 1, \beta \geqslant 0$.*

1. *Then there exists a* PO BP-2 X' *in the (α, β)-core.*
2. *If X is also* EF-r *for some $r \geqslant 0$, there exists a* BP-2 X' *in the (α, β)-core, which is both* EF-r *and* PO.

Proof. Part 1: Assume, for contradiction, that X is not PO. Let X' be PO partition obtained by sequentially improving X (as we did for Theorem 2). Suppose X' is not in (α, β)-core. Then there exists a blocking coalition S of size either $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$ such that for the BP-2 $Y = (S, V \setminus S)$, $u_i(Y) > \alpha \cdot u_i(X') + \beta$ for all $i \in S$. But, by construction, $u_i(X') \geqslant u_i(X)$ since X' was obtained by sequentially improving X. Hence, S is also a blocking coalition for BP-2 X, which contradicts the fact that X is in (α, β)-core.

Part 2: Now, suppose X is also EF-r. Construct X' in a similar way as part 1. Part 1 shows that X' will be in the (α, β)-core. By Theorem 2, X' will also be EF-r. □

Discussion: While we examine the interplay between different fairness notions in the theorems above, it is essential to recognize that each of these properties remains important in its own right. For instance, a PO BP-2 in the core is not guaranteed to hold the envy-freeness guarantees we determined in Theorem 1. To illustrate this, for any $m \geqslant 3$, consider constructing the graph G_m with $2(m + 1)$ vertices comprising of a disjoint K_m and K_{m+1}, and a vertex $v \notin K_m \cup K_{m+1}$ connected to exactly m vertices of K_{m+1}. Partition G_m into two groups, one consisting of K_m and v, and the other one with K_{m+1}. If $v' \in K_{m+1}$ is the vertex that v is not joined to, then v envies v' by $m = \Delta(G) - 1$, even though this BP-2 is PO and in the core.

4 MAIN RESULTS: GRID GRAPHS

In Section 3, we saw the fairness and efficiency guarantees for general graphs. These results have their limitations owing to the arbitrary structure of the graphs. We focus on grid graphs in this section to investigate whether a relatively simpler graph structure can yield stronger fairness and stability properties. In the following two subsections, we will consider envy-freeness and core properties, respectively, for GrGs.

4.1 Envy-Freeness

We start this section by showing the incompatibility of BP-k and EF-0 for all $k \geq 2$ even in GrGs.

Example 1. *For any $k \geq 2$, the graph $G_k = \mathbf{comb}_{k+1} \cup \overline{K_{k-2}}$ consisting of a comb graph with $2(k+1)$ vertices and $k-2$ isolated vertices, has no* EF-0 BP-k. *To see why, note that each subset in a* BP-k *X of G_k must have 3 nodes. Let S_1 be the set of nodes with degree 1. As $|S_1| = k+1$ and there are k groups, $\exists\, i, j \in S_1$ that are in the same group X_ℓ (say). But if X is* EF-0*, then every node in S_1 must have their only neighbor in the same subset. This would imply that $|X_\ell| \geq 4$ (i, j and their 1 neighbor each), which is a contradiction.* □

However, if the EF guarantee is relaxed to EF-2, we show that there is an efficient algorithm (Algorithm 2) to find a BP-k in GrGs, for any $k \geq 2$. The algorithm traverses the grid graph column-wise along the integer lattice $\mathbb{Z}^2$ from the left-top corner to the right-bottom corner, placing a contiguous sequence of traversed vertices in one subset of the partition.[3] The traversal is continued until there is no vertex left. This method first creates $n \pmod{k}$ such subsets of size $\lceil |V|/k \rceil$, and then remaining subsets of size $\lfloor |V|/k \rfloor$. An example of a BP-2 created by Algorithm 2 is shown in Figure 2.

Theorem 4. *For all $k \geq 2$ and $G = (V, E) \in$ GrG, Algorithm 2 returns a* EF-2 BP-k *X in $O(|V| \log(|V|))$ time.*

Proof. Assume, for contradiction, that BP-k X (returned by Algorithm 2) is not EF-2. Then $\exists i, j \in V$ such that i envies j by at least 3. By Observation 2, i has at least 3 neighbors in $X_{X^{-1}(j)}$. Define $N_{TL}(i)$ as the set of neighbors (can be empty) to the top and left of i in the integer lattice $\mathbb{Z}^2$. Formally, if $i = (x, y)$ on the lattice, then $N_{TL}(i) = \{(x, y+1), (x-1, y)\} \cap N(i)$. Similarly define $N_{BR}(i) = \{(x, y-1), (x+1, y)\} \cap N(i)$ as the neighbors of i that lie to the bottom and right of i. Note that $|N_{BR}(i)|, |N_{TL}(i)| \leq 2$. Since at least 3 neighbors of i are in the same subset, by the pigeonhole principle, $\exists\, u \in N_{TL}(i),\ v \in N_{BR}(i)$ such that both nodes u and v belong to the same subset $X_{X^{-1}(j)}$ of the partition returned by Algorithm 2. But then all nodes visited between u and v in Algorithm 2 should also be in $X_{X^{-1}(j)}$, which includes i. This leads to a contradiction.

(Complexity): For the time complexity, note that Algorithm 2 sorts V in $O(|V| \log(|V|))$ time, and then linearly iterates through all the vertices in this sorted order, making the whole algorithm complete in $O(|V| \log(|V|))$ time. □

[3]Here traversal is done over all vertices that exist in the graph on the lattice irrespective of whether they are joined by an edge or not.

Algorithm 2 L2R: Left-to-Right

Input: $G = (V, E) \in$ GrG, partition size $k \geq 2$
Output: BP-k X of G
Suppose $|V| = n$ such that $n = q \cdot k + r,\ 0 \leq r < k$
// Arrange nodes from left-top to right-bottom
Sort nodes in V using $(x, -y)$ as key
// First r groups have $\lceil n/k \rceil$ nodes
// Next $(k - r)$ groups have $\lfloor n/k \rfloor$ nodes
for cnt = 1 to k **do**
  $X_{\text{cnt}} \leftarrow \emptyset$; num $\leftarrow 1$
  size $\leftarrow \lceil n/k \rceil$ **if** cnt $\leq r$ **else** $\lfloor n/k \rfloor$
  while num $\leq$ size **do**
    $v \leftarrow$ left uppermost node in V ; $V \leftarrow V \setminus \{v\}$
    $X_{\text{cnt}} \leftarrow X_{\text{cnt}} \cup \{v\}$; num $\leftarrow$ num + 1
  end while
end for

4.2 Core

Theorem 5. *For every $G = (V, E) \in$ GrG, there is a polynomially computable* EF-2 BP-2 *that lies in the $(1, 1)$-core.*

Consider the partition X returned by Algorithm 2 (L2R) on a GrG $G = (V, E)$ for $k = 2$. From Theorem 4, no node has more than 2 neighbors in the other subset in X, i.e., for all $i \in V, u_i(X) \geq d(i) - 2$. We classify the nodes into three categories, based on their neighbors in the other group.

Definition 11. *Let $X = L2R\,(G = (V, E), k = 2)$. Then we call a node $i \in V$, (1) an* internal *node if $u_i(X) = d(i)$, (2) an* edge *node if $u_i(X) = d(i) - 1$, and (3) a* corner *node if $u_i(X) = d(i) - 2$.*

For instance in Figure 2, D is an internal node, B is an edge node and L is a corner node.

Observation 3. *There can be at most 2 corner nodes. If both exist, there must be an edge between them, and they cannot share a common neighbor.*

Proof. Let $X = (\mathcal{R}, \mathcal{B})$. Let $c_R \in \mathcal{R}$ be the bottom-most node in the rightmost column of $\mathcal{R}$, and $c_B \in \mathcal{B}$ be the uppermost node in the leftmost column of $\mathcal{B}$. Then c_B will be visited immediately after c_R in the traversal done in Algorithm 2. For any $i \in \mathcal{R},\ i \neq c_R$, i will have at most 1 neighbor in $\mathcal{B}$ (a neighbor in $\mathcal{B}$ can only lie to its right). So the only possible corner node in $\mathcal{R}$ is c_R. Similarly, for any $j \in \mathcal{B},\ j \neq c_B$, j will have at most 1 neighbor in $\mathcal{R}$ (that can lie to its left), making c_B the only possible corner node in $\mathcal{B}$. Thus, there are at most 2 corner nodes. Finally, if both c_R and c_B are corner nodes, then they must be joined to each other to ensure that each of them has 2 neighbors in the other subset. And, since there can be no C_3 in a grid graph, the 2 adjacent corner nodes cannot share a common neighbor. □

We first give a lemma that will aid us in proving Theorem 5.

Lemma 7. *No internal or edge node can be a part of a $(1, 1)$-blocking coalition in* BP-2 *X.*

Proof. For a node i to be a part of a $(1, 1)$-blocking coalition S (Definition 5), we must have $u_i((S, V \setminus S)) > u_i(X) + 1$, which

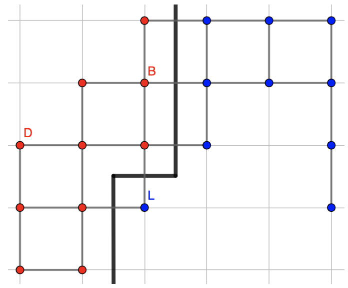

Figure 2: Partitioning done by Algorithm 2 (L2R) for $k = 2$.

gives $u_i(X) \leqslant d(i) - 2$ (since the utility of i is bounded above by the degree). This is only possible for corner nodes (Definition 11). □

Proof. (of Theorem 5) According to Theorem 4, BP-2 X returned by Algorithm 2 is already EF-2. For the sake of contradiction, suppose a $(1, 1)$-blocking coalition S exists in partition X. Using Lemma 7, we see that only the corner nodes can be a part of S, which gives $|S| \leqslant 2$ (using Observation 3). So any vertex in S can have at most 1 neighbor in S. Thus, for any $i \in S$, we have $u_i((S, V \setminus S)) \leqslant 1 \leqslant u_i(X) + 1$, which is a contradiction. This proves that such a blocking coalition S cannot exist, and so the BP-2 returned by Algorithm 2 is both EF-2 and in the $(1, 1)$-core. □

Discussion: For arbitrary graphs, Li et al. [18] showed that when $k \geqslant 3$, a BP-k in the $(1, 0)$-core may not exist, with the graph C_{k+1} as the desired counter-example. Since odd cycles can not exist in GrGs, the same example does not work when k is even. So for even k, we can instead consider P_{k+1} as an instance of a GrG that has no BP-k in $(1, 0)$-core (the same proof works). Thus, for $k \geqslant 3$, there may be no BP-k in $(1, 0)$-core even for GrGs. Li et al. [18] also raised an open problem about the existence of a BP-2 belonging to $(1, 0)$-core in arbitrary graphs. We show that for GrGs, we can always find such a partition.

Theorem 6. *For every $G \in$ GrG, Algorithm 3 returns a* BP-2 *that belongs to* $(1, 0)$*-core.*

Consider Algorithm 2 (L2R) for $k = 2$, which returns a BP-2 X. Suppose it is not in the core (otherwise we have the required partition). Then there exists a blocking coalition S in our partition X. Let $X' = (S, V \setminus S)$ be another BP-2. As per Definition 5, for a node i to be a part of S, we must have $u_i(X') > u_i(X)$. Thus, its utility in X' must strictly increase over that in X. We have the following lemma.

Lemma 8. *No internal node can lie in the blocking coalition S as defined above. Also, if an edge node $j \in S$, then all of its neighbors are also in S (i.e., $N(j) \subseteq S$).*

Proof. An internal node has all neighbors of the same color as itself, so its utility cannot increase any further. Thus, it cannot be part of S. Suppose an edge node $j \in S$. Since $u_j(X') \geqslant u_j(X) + 1 = d(j)$, and the utility of j is also bounded above by $d(j)$, we get equality here, i.e. $u_j(X') = d(j)$ (all neighbors of j are in S). □

Algorithm 3 CoreL2R: Core Left-to-Right

Input: $G = (V, E) \in$ GrG
Output: BP-2 X of G

```
X ← L2R(G, k = 2); // Algorithm 2
if noBlocking(X) then return X
else
    S = blockingSet(X) ; X′ = (S, V \ S)
    if noBlocking(X′) then return X′
    else
        S′ = blockingSet(X′); return (S′, V \ S′)
    end if
end if
```

Proof. (of Theorem 6) If X' is in core then we have the desired partition. Otherwise, there is some blocking coalition S' in X' as well. Consider any $i \in S'$. There are 2 possible scenarios:

Case 1: $i \in S' \cap S$: then $u_i((S', V \setminus S')) > u_i(X') > u_i(X) \implies u_i((S', V \setminus S')) \geqslant u_i(X) + 2$. This is only possible for corner nodes, as only they can have 2 neighbors from the other group in X. And, if a corner node i is in S', then all its neighbors must also be in S' to ensure the above condition (since $u_i(X) = d(i) - 2$).

Case 2: $i \in S' \cap T$: then $u_i((S', V \setminus S')) \geqslant u_i(X') + 1$, which means that i must have at least 1 neighbor $\in S$ so that its utility can increase from that in X'. Call any such neighbor j. As per Lemma 8, j cannot be an internal node. If it is an edge node, all its neighbors will also be in S (Lemma 8), which would imply that it cannot have a neighbor $i \in T$. So the only option left is j being a corner node, i.e. i's neighbors in S can only be corner nodes. Since a node cannot be a common neighbor of two corner nodes (Observation 3), i will have exactly one neighbor in S. Thus, to increase the utility of i in $(S', V \setminus S')$, all neighbors of i must also be in S'.

In both the cases above, we see that any $i \in S'$ must have all its neighbors also in S', implying that there is no edge between S' and $V \setminus S'$. Then the BP-2 $(S', V \setminus S')$ is in the core, as desired. □

5 CONCLUSIONS AND FUTURE WORK

We design efficient algorithms for approximately envy-free balanced partitions and establish existential results on stable and Pareto optimal partitions. Our findings raise several key questions:

- Can EF-2 BP-2 be shown to exist in all biconnected graphs and extended to connected graphs via block-cut trees?
- For arbitrary graphs, can consecutive $(1, 0)$-blocking coalitions (as in Algorithm 3) reach a BP-2 in $(1, 0)$-core within $O(\Delta(G))$ steps?
- Does every GrG admit a EF-1 BP-k? We shed more light on this in the full version [1].
- Can a PO BP-2 from Theorems 2 and 3 be found efficiently, and can this extend to BP-k for $(k \geqslant 3)$?

ACKNOWLEDGMENTS

The supports of a MATRICS grant (MTR/2021/000367) and a Core Research Grant (CRG/2023/001442) from SERB, Govt. of India, a TCS grant (MOU/CS/10001981-1/22-23), and an IIT Bombay grant (2022058) are gratefully acknowledged.

REFERENCES

[1] Pulkit Agarwal, Harshvardhan Agarwal, Vaibhav Raj, and Swaprava Nath. 2024. Harmonious Balanced Partitioning of a Network of Agents. (2024). https://www.cse.iitb.ac.in/~swaprava/papers/balanced.pdf

[2] Konstantin Andreev and Harald Räcke. 2004. Balanced Graph Partitioning. In *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures* (Barcelona, Spain) *(SPAA '04)*. Association for Computing Machinery, New York, NY, USA, 120–124. https://doi.org/10.1145/1007912.1007931

[3] Esther M. Arkin, Sang Won Bae, Alon Efrat, Kazuya Okamoto, Joseph S.B. Mitchell, and Valentin Polishchuk. 2009. Geometric stable roommates. *Inform. Process. Lett.* 109, 4 (2009), 219–224. https://doi.org/10.1016/j.ipl.2008.10.003

[4] Haris Aziz, Felix Brandt, and Paul Harrenstein. 2013. Pareto optimality in coalition formation. *Games and Economic Behavior* 82 (2013), 562–581. https://doi.org/10.1016/j.geb.2013.08.006

[5] Cristina Bazgan, Zsolt Tuza, and Daniel Vanderpooten. 2010. Satisfactory graph partition, variants, and generalizations. *European Journal of Operational Research* 206, 2 (2010), 271–280. https://doi.org/10.1016/j.ejor.2009.10.019

[6] Freya Behrens, Gabriel Arpino, Yaroslav Kivva, and Lenka Zdeborová. 2022. (Dis) assortative partitions on random regular graphs. *Journal of Physics A: Mathematical and Theoretical* 55, 39 (2022), 395004.

[7] Vittorio Bilò, Gianpiero Monaco, and Luca Moscardelli. 2022. Hedonic games with fixed-size coalitions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9287–9295.

[8] Niclas Boehmer and Edith Elkind. 2020. Stable Roommate Problem with Diversity Preferences. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 96–102. https://doi.org/10.24963/ijcai.2020/14 Main track.

[9] Ágnes Cseh, Michael McKay, and David Manlove. 2022. Envy-freeness in 3D Hedonic Games. *arXiv preprint arXiv:2209.07440* (2022). arXiv:2209.07440 [cs.GT]

[10] J. Ebert. 1983. st-Ordering the vertices of biconnected graphs. *Computing* 30 (1983), 19–33. https://api.semanticscholar.org/CorpusID:6570953

[11] Andreas Feldmann. 2012. *Balanced Partitions of Grids and Related Graphs.* Ph.D. Dissertation.

[12] Michael R. Garey and David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., USA.

[13] John Hopcroft and Robert Tarjan. 1973. Algorithm 447: Efficient Algorithms for Graph Manipulation. *Commun. ACM* 16, 6 (jun 1973), 372–378. https://doi.org/10.1145/362248.362272

[14] Chien-Chung Huang. 2007. Two's Company, Three's a Crowd: Stable Family and Threesome Roommates Problems. In *Algorithms – ESA 2007*, Lars Arge, Michael Hoffmann, and Emo Welzl (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 558–569.

[15] Robert W Irving. 1985. An efficient algorithm for the "stable roommates" problem. *Journal of Algorithms* 6, 4 (1985), 577–595. https://doi.org/10.1016/0196-6774(85)90033-1

[16] B. W. Kernighan and S. Lin. 1970. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal* 49, 2 (1970), 291–307. https://doi.org/10.1002/j.1538-7305.1970.tb01770.x

[17] Chaya Levinger, Amos Azaria, and Noam Hazon. 2023. Socially Aware Coalition Formation with Bounded Coalition Size. *arXiv preprint arXiv:2008.06179* (2023). arXiv:2306.01378 [cs.GT]

[18] Lily Li, Evi Micha, Aleksandar Nikolov, and Nisarg Shah. 2023. Partitioning Friends Fairly. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'23)*. AAAI Press, Article 644, 8 pages. https://doi.org/10.1609/aaai.v37i5.25713

[19] Michael McKay. 2022. *Algorithmic aspects of fixed-size coalition formation.* PhD Thesis. University of Glasgow.

[20] Michael McKay and David Manlove. 2021. The Three-Dimensional Stable Roommates Problem with Additively Separable Preferences. In *Algorithmic Game Theory*, Ioannis Caragiannis and Kristoffer Arnsfelt Hansen (Eds.). Springer International Publishing, Cham, 266–280.

[21] Dor Minzer, Ashwin Sah, and Mehtaab Sawhney. 2023. On Perfectly Friendly Bisections of Random Graphs. *arXiv preprint arXiv:2305.03543* (2023).

[22] Renata Sotirov. 2018. Graph bisection revisited. *Annals of Operations Research* 265 (2018), 143–154.