A Little Charity Guarantees Fair Connected Graph Partitioning*

Ioannis Caragiannis
Aarhus University
iannis@cs.au.dk

Evi Micha
University of Toronto
emicha@cs.toronto.edu

Nisarg Shah
University of Toronto
nisarg@cs.toronto.edu

Abstract

Motivated by fair division applications, we study a fair connected graph partitioning problem, in which an undirected graph with m nodes must be divided between n agents such that each agent receives a connected subgraph and the partition is *fair*. We study approximate versions of two fairness criteria: α-*proportionality* requires that each agent receives a subgraph with at least $1/\alpha \cdot m/n$ nodes, and α-*balancedness* requires that the ratio between the sizes of the largest and smallest subgraphs be at most α. Unfortunately, there exist simple examples in which no partition is reasonably proportional or balanced. To circumvent this, we introduce the idea of *charity*. We show that by "donating" just $n-1$ nodes, we can guarantee the existence of 2-proportional and almost 2-balanced partitions (and find them in polynomial time), and that this result is almost tight. More generally, we chart the tradeoff between the size of charity and the approximation of proportionality or balancedness we can guarantee.

1 Introduction

The problem of fair division concerns the allocation of a set of goods (or chores) fairly between a set of agents. Perhaps the most canonical model is cake-cutting, in which a heterogeneous divisible good, called cake, is divided between n agents. Under minimal assumptions, this model allows providing compelling fairness guarantees. For example, one can ensure proportionality [34], which demands that each agent's value for her allocation be at least $1/n$-th of her value for the entire cake, or the stronger notion of envy-freeness [22, 25], which demands that no agent strictly prefers another agent's allocation to her own.

However, many real-world applications pose additional constraints, which often make such strong fairness notions impossible to guarantee. A common constraint, which has received increasing attention recently, is *indivisibility*. Here, one assumes that the goods cannot be split, i.e., each good must be allocated entirely to a single agent. For example, when dividing an inheritance between heirs, goods such as a house or a piece of jewelry are indivisible. In this case, one can no longer guarantee proportionality or envy-freeness; think of allocating a single indivisible good between two agents. Nonetheless, "up to one good"-style relaxations can be guaranteed [8, 10, 18], which converge to providing exact proportionality or envy-freeness when each individual good is negligible compared to the set of all goods.

The situation becomes more dire when we impose another common constraint: *connectedness*. Bouveret et al. [6] introduced a model where the indivisible goods are nodes of a graph and the goal is to allocate to each agent a subset of goods that forms a connected subgraph. Examples of real-world applications where connectedness is desirable include allocation of offices to research groups in an academic building, land division [21], congressional redistricting[1], power grid islanding [33], and metadata partitioning in large-scale distributed storage systems [39].

*A preliminary version of this paper appeared in the proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI), 2022.

[1]This is the process of re-drawing electoral district boundaries. Formally, a graph of electoral precincts is divided into a fixed number of connected subgraphs (districts) with approximately equal populations [37].

While many of these applications have identical goods (meaning that all agents have the same value for each good), it is easy to see that even in this special case, no reasonable relaxation of proportionality or envy-freeness can be guaranteed, even if each individual good is negligible compared to the set of all goods. For example, consider $m \gg n$ identical goods connected via a star graph with a hub node connected to $m-1$ leaf nodes. Any way of partitioning the nodes into n connected bundles will produce a highly imbalanced partition in which one very large bundle has at least $m-n+1$ nodes while every other bundle has at most a single node.

This, in essence, is the fair graph partitioning problem that we study in this work. Formally, we are given an undirected graph $G = (V, E)$, where V is a set of m nodes and we want to partition it into $(V_1, \ldots, V_n)$ such that each V_i forms a connected subgraph. Borrowing from the fair division literature, we call this partition α-proportional if $\min_i \alpha \cdot |V_i| \geqslant m/n$, and α-balanced if $\max_i |V_i| \leqslant \alpha \cdot \min_i |V_i|$. It is easy to see that α-balancedness implies α-proportionality.[2] Balancedness and similar cardinality constraints have been investigated previously in various fair division contexts [1, 5, 27, 30]; in our case, note that 1-balancedness is equivalent to envy-freeness.

While the aforementioned star graph example rules out any reasonably fair partition, note that if we could keep just the hub node unallocated, we could partition the leaf nodes in a highly proportional and balanced manner. In the fair division literature, the idea of keeping a few goods unallocated, termed *charity*, has been used to achieve fairness guarantees that are even stronger than envy-freeness up to one good without the connectedness constraint [3, 9, 12, 13]. We borrow this idea and show that charity also helps improve fairness when connectedness is desired. In our context, the unallocated nodes can also be viewed as shared between agents; e.g., in land division, these can be public land accessible by all agents.

Formally, we seek a partition $(V_1, \ldots, V_n, R)$ of V, where the set of unallocated (or excluded) nodes R is small and each V_i is connected "via" R (i.e., there exists $R_i \subseteq R$ such that $V_i \cup R_i$ is connected). While α-balancedness definition remains unchanged, α-proportionality is now defined as $\alpha \cdot \min_i |V_i| \geqslant (m - |R|)/n$, so that α-balancedness still implies α-proportionality. Revisiting the star graph example, we can see that if we divide a star graph with a hub node connected to three leaf nodes between two agents, the best we can hope for with a single node exclusion is 2-balancedness and 1.5-proportionality. Generalizing this example, we later show (Theorem 1) that when dividing a graph between n agents, the best we can hope for with $n-1$ node exclusions is 2-balancedness and $(2-1/n)$-proportionality. This leads to our main research questions:

> *Is a 2-balanced or $(2-1/n)$-proportional partition of a graph between n agents guaranteed to exist with only $n-1$ node exclusions? If so, can we find such a partition in polynomial time? More generally, what is the tradeoff between the approximation of proportionality or balancedness we can achieve and the number of nodes we need to exclude?*

Since the number of nodes m can be much greater than n, following the fair division literature [13], we view excluding $O(n)$ nodes as "a little charity".

1.1 Our Results

We begin by the case where at most $n-1$ node exclusions are allowed. We prove a lower bound which shows that α-balancedness and α-proportionality cannot be guaranteed for any $\alpha < 2$ and $\alpha < 2 - 1/n$, respectively (Theorem 1).

Next, for $n \in \{2, 3\}$, we show that this bound is tight and such partitions can be found in polynomial time (Theorems 2 and 3). For higher values of n, we provide three efficient algorithms which obtain generally incomparable approximation guarantees: one ensures $(3 + O(n/m))$-balancedness and 3-proportionality, another ensures 4-balancedness and 2-proportionality, and the final one ensures $(2 + O(n^2/m))$-balancedness and $(2 - 1/n + O(n^2/m))$-proportionality. In particular, for fixed n, when $m \to \infty$, the final result matches the lower bound from Theorem 1. We conjecture that it should be possible to achieve 2-balancedness and $(2-1/n)$-proportionality for any n and m.

We also consider the tradeoff between the charity (number of node exclusions allowed) and approximations to balancedness or proportionality which can be guaranteed. While we provide almost tight bounds on this tradeoff when more than $n-1$ exclusions are allowed, we leave behind interesting open questions when fewer than $n-1$ exclusions are allowed. We also show hardness of checking the existence of balanced partitions with at most $n-1$ exclusions or

[2] Actually, it implies $(\alpha - (\alpha - 1)/n)$-proportionality.

approximately balanced partitions with no exclusions. In appendix, some miscellaneous extensions of our framework can be found.

1.2 Related Work

Our work is related to various models studied in mathematics, theoretical computer science, and multiagent systems.

In theoretical computer science, the problem of partitioning the nodes of a graph into connected subgraphs is well-studied. It is known that checking whether a partition into equal-sized connected subgraphs — hence, perfectly proportional and balanced — exists is NP-hard [24]; hence, this literature focuses on designing approximation algorithms for computing partitions that are close to optimal according to various criteria, such as maximizing the minimum size (related to proportionality) [11, 15] and minimizing the maximum size [14]. However, when even the optimal partitions are highly imbalanced, as in the star graph example from the introduction, such approximations are also unsatisfactory. The focus of our work is to provide worst-case bounds on balancedness and proportionality by allowing the exclusion of a few nodes (charity).

In mathematics, the related problem of partitioning the *edges* rather than nodes of a graph has received attention. For the special case of trees, this problem was introduced by Wu et al. [38], who proved the existence of 3-balanced and $(2-1/n)$-proportional edge partitions; note that this is without any edge exclusions. Later, Dye [23] improved the balancedness approximation to 2 for $n \in \{2, 3, 4\}$, Chu et al. [16] extended this result to all values of n, and Chu et al. [17] showed how to achieve this in linear time even when the edges are weighted. In Section 5, we make an connection between edge partitions of trees with no edge exclusions and node partitions of general graphs with at most $n-1$ node exclusions, allowing us to leverage the above results to obtain upper bounds for our problem.

Our primary motivation stems from the fair division literature in multiagent systems, where the goal is to partition the available goods between agents in a way that each agent receives a connected subset. While envy-freeness and proportionality can be achieved exactly when the goods are divisible, as in cake-cutting [35, 36], as illustrated in the introduction, not even a reasonable approximation of these guarantees can be provided when the goods are indivisible, modeled as nodes of a graph. Hence, this literature focuses on special families of graphs, such as path graphs, for which such guarantees can be provided [2, 4, 6], and on the computational complexity of the existence of fair connected allocations [20, 26, 29]. Our goal is to provide approximate fairness guarantees for *general graphs*, by using the idea of charity, which has been explored recently for fair division without the connectedness constraint [3, 9, 12, 13].

We remark that connected fair division has also been studied for *chores* rather than goods, with both divisible chores [19, 28] and indivisible ones [7].

2 Preliminaries

For $q \in \mathbb{N}$, define $[q] = \{1, \ldots, q\}$. Let $G = (V, E)$ be a graph with $|V| = m$. We denote with $G[X]$ the subgraph induced by $X \subseteq V$. We say that $(V_1, \ldots, V_n, R)$ is a *pseudo n-partition* of G if

1. $V = (\cup_{i \in [n]} V_i) \cup R$;
2. $V_i \cap V_j = \emptyset$ for distinct $i, j \in [n]$, and $V_i \cap R = \emptyset$ for all $i \in [n]$; and
3. $|R| \leqslant n - 1$.

When $|R| = 0$, we simply refer to it as an n-partition of G. A pseudo n-partition $(V_1, \ldots, V_n, R)$ is called *connected* if, for every $i \in [n]$, there exists $R_i \subseteq R$ such that the subgraph $G[V_i \cup R_i]$ is connected. Throughout the paper, we assume that G is connected and $m \geqslant n$, otherwise there may not exist any connected pseudo n-partition of G.

In our motivating fair division applications, the nodes of G are the goods, V_i is the set of goods allocated to agent i, and R is the set of goods left unallocated (charity). We are typically interested in the case where $n \ll m$, so a charity of $n-1$ out of m nodes is very little.

With such little charity, our goal is to find a connected pseudo n-partition $(V_1 \ldots, V_n, R)$ of G that is reasonably *fair*. We consider the following fairness desiderata.

Definition 1 (Balancedness)**.** For $\alpha \geqslant 1$, we say that a connected pseudo n-partition $(V_1, \ldots, V_n, R)$ is α-balanced if $\max_{i\in[n]} |V_i| \leqslant \alpha \cdot \min_{i\in[n]} |V_i|$. We refer to 1-balancedness simply as balancedness.

Definition 2 (Proportionality)**.** For $\alpha \geqslant 1$, we say that a connected pseudo n-partition $(V_1, \ldots, V_n, R)$ is α-proportional if $\alpha \cdot \min_{i\in[n]} |V_i| \geqslant (m - |R|)/n$. We refer to 1-proportionality simply as proportionality.

Note that if a connected pseudo n-partition $(V_1, \ldots, V_n, R)$ is α-balanced, then we have $m = |R| + \sum_{i\in[n]} |V_i| \leqslant |R| + |V_i| + (n-1) \cdot \alpha \cdot |V_i|$ for any $i \in [n]$, which, after some simplification, implies that the partition is also $(\alpha - (\alpha - 1)/n)$-proportional. In particular, 2-balancedness implies $(2 - 1/n)$-proportionality.

We remark that the most difficult case of our problem is when G is a tree. Trivially, any lower bounds derived for trees apply to the general case as well. But note that any upper bounds derived for trees can also be translated to the general case. This is because, given any algorithm for trees and an input graph G, we can apply the algorithm to any spanning tree of G (which can be computed efficiently). Any pseudo n-partition produced by the algorithm that is connected under the spanning tree must also be connected under G. Hence, throughout the paper, we assume G to be a tree without loss of generality.

We will often work with rooted trees. Given a tree $G = (V, E)$ and a node $v \in V$, let $T = (G, v)$ denote the tree G rooted at v. Given a node $u \in V$, let $ST(u, T)$, $c(u, T)$, and $p(u, T)$ denote the subtree, the set of children nodes, and the parent node of u, respectively ($p(v, T)$ is undefined); let $\mathrm{level}(u, T)$ denote the length of the (unique) path from u to the root v in T, with $\mathrm{level}(v, T) = 1$. Define the height of tree T as $\mathrm{height}(T) = \max_{u\in V} \mathrm{level}(u, T)$. We drop T from the notation when it is clear from the context.

3 A Lower Bound

We begin by showing that we cannot hope to provide any guarantee better than 2-balancedness or $(2-1/n)$-proportionality. This uses a generalization of the example used in the introduction to establish these lower bounds for $n = 2$. In later sections, we design algorithms that (almost) achieve these bounds.

Theorem 1. *There exists an instance in which no connected pseudo n-partition is α-balanced for any $\alpha < 2$ or α-proportional for any $\alpha < 2 - 1/n$.*

Proof. Let $\ell \geqslant n$ be an integer. Consider the graph $G = (V, E)$ that consists of $2n - 1$ paths of length ℓ each, denoted $P_1, \ldots, P_{2n-1}$, and $n - 1$ additional "hub" nodes, denoted $h_1, \ldots, h_{n-1}$. Hence, $|V| = \ell \cdot (2n - 1) + n - 1$. For $j \in [n - 2]$, h_j is connected to h_{j+1} as well as to one of the endpoints of paths P_{2j-1} and P_{2j}. Finally, h_{n-1} is connected to one of the endpoints of paths P_{2n-3}, P_{2n-2}, and P_{2n-1}.

First, we show that there is no connected pseudo n-partition $(V_1, ..., V_n, R)$ such that $|V_i| \geqslant \ell + 1$ for all $i \in [n]$. For the sake of contradiction, assume that such a partition exists. We show that each path intersects at most one of the parts. Indeed, if there exist $j \in [2n - 1]$ and distinct $i, i' \in [n]$ such that $P_j \cap V_i \neq \emptyset$ and $P_j \cap V_{i'} \neq \emptyset$, then the part that contains the node in $P_j \cap (V_i \cup V_{i'})$ farthest from the hub that P_j is attached to would have size at most $\ell - 1$, which is a contradiction. Since there are $2n - 1$ paths and each intersects at most one part, by the pigeonhole principle, there must exist $i^* \in [n]$ such that V_{i^*} intersects with at most one path P_{j^*}. Since $|V_{i^*}| \geqslant \ell + 1$, it must contain at least one hub node v. Since each hub node is attached to at least two paths, v must be attached to a path $P_{j'}$ different from P_{j^*}. Since $|R| \leqslant n - 1 < \ell = |P_{j'}|$, we have $P_{j'} \not\subseteq R$; hence, there must exist $i' \in [n] \setminus \{i\}$ such that $V_{i'} \cap P_{j'} \neq \emptyset$. However, since the hub node v that $P_{j'}$ is attached to is allocated to V_i, by the connectedness constraint we have $V_{i'} \subseteq P_{j'}$, implying $|V_{i'}| \leqslant \ell$, which is a contradiction.

We have established that in any connected pseudo n-partition, there exists $i \in [n]$ such that $|V_i| \leqslant \ell$. If it is α-proportional, then we need

$$\alpha \cdot \ell \geqslant \frac{m - |R|}{n} \geqslant \frac{(2n - 1) \cdot \ell}{n},$$

which implies $\alpha \geqslant 2 - 1/n$. Since α-balancedness implies $(\alpha - (\alpha - 1)/n)$-proportionality for any $\alpha \geqslant 1$, the impossibility of achieving α-proportionality for $\alpha < (2 - 1/n)$ implies the impossibility of getting α-balancedness for $\alpha < 2$.

□

Algorithm 1: 2-balancedness and 1.5-proportionality for $n = 2$

Input: Tree $G = (V, E)$ with $|V| = m$ nodes.
Output: A connected pseudo 2-partition.

1: $r \leftarrow$ arbitrary node in V
2: $T \leftarrow$ tree (G, r) rooted at r
3: Find a node u^* such that $|ST(u^*, T)| \geqslant \lceil m/3 \rceil > ST(v, T)$ for every child v of u^*
4: **if** $|ST(u^*, T)| = \lceil m/3 \rceil$ **then**
5: $\quad (V_1, V_2, R) \leftarrow (ST(u^*, T), V \setminus ST(u^*, T), \emptyset)$
6: **else**
7: $\quad R \leftarrow \{u^*\}, V_1 \leftarrow \emptyset$
8: $\quad$ **for** $v \in c(u^*, T)$ **do**
9: $\quad\quad V_1 \leftarrow V_1 \cup ST(v, T)$
10: $\quad\quad$ **if** $|V_1| \geqslant \lceil m/3 \rceil$ **then**
11: $\quad\quad\quad$ **break**
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad V_2 \leftarrow V \setminus (V_1 \cup \{u^*\})$
15: **end if**
16: **return** (V_1, V_2, R)

4 Optimal 2-Partitions and 3-Partitions

In this section, we show that the lower bound from Theorem 1 is tight when $n \in \{2, 3\}$. For these cases, we design efficient algorithms for finding connected pseudo n-partitions that are 2-balanced (and thus, $(2 - 1/n)$-proportional). The algorithm for $n = 2$, Algorithm 1, is of particular interest, as we will use it as a subroutine in the next section to derive bounds for higher values of n.

Algorithm 1 returns a connected 2-balanced pseudo 2-partition with $|R| \leqslant 1$ as follows. It roots the given tree arbitrarily, and then finds a node u^* at maximal depth whose subtree has at least $\lceil m/3 \rceil$ nodes. If the subtree has exactly $\lceil m/3 \rceil$ nodes, it assigns the subtree as one part and the rest of the tree as the other part (not excluding any node). Otherwise, it excludes u^*, and adds subtrees of its children iteratively to a part until the part has at least $\lceil m/3 \rceil$ nodes. The remaining nodes form the other part. A similar trick has been used previously in the literature; see, e.g., [32] and [31].

Theorem 2. *When $n = 2$, Algorithm 1 runs in polynomial time and returns a connected pseudo 2-partition that is 2-balanced and, hence, 1.5-proportional.*

Proof. We have already argued that Algorithm 1 can be implemented efficiently. It is also easy to check that it returns a connected pseudo 2-partition. Now, we show that it achieves 2-balancedness, which implies 1.5-proportionality, as argued in Section 2.

First, consider the case where $|ST(u^*, T)| = \lceil m/3 \rceil$. In this case, since $|R| = 0$, we need to show that $\min(|V_1|, |V_2|) \geqslant \lceil m/3 \rceil$. This is already satisfied for $V_1 = ST(u^*, T)$, and we have $|V_2| = m - \lceil m/3 \rceil \geqslant \lceil m/3 \rceil$.

Next, consider the case where $|ST(u^*)| > \lceil m/3 \rceil$. In this case, since $|R| = 1$, we need to show that $\min(|V_1|, |V_2|) \geqslant \lceil (m-1)/3 \rceil$. For V_1, this follows by its construction. Also, consider the last subtree $ST(v, T)$ added to V_1 in Line 9. Before adding this subtree, V_1 must have had at most $\lceil m/3 \rceil - 1$ nodes. Further, since u^* is a node of maximal height with $|ST(u^*, T)| \geqslant \lceil m/3 \rceil$, we must have $|ST(v, T)| \leqslant \lceil m/3 \rceil - 1$ for the child v of u^*. Hence, we have $|V_1| \leqslant 2(\lceil m/3 \rceil - 1)$, implying that $|V_2| \geqslant m - 1 - 2(\lceil m/3 \rceil - 1) \geqslant \lceil (m-1)/3 \rceil$. The theorem follows. $\square$

We make a note of the following fact established in the proof of Theorem 2, which we will use in the next section when using Algorithm 1 as a subroutine and deriving bounds for higher values of n.

Corollary 1. *Algorithm 1 returns a connected 2-partition (V_1, V_2, R) such that $\min(|V_1|, |V_2|) \geqslant \lceil (m - |R|)/3 \rceil$.*

Next, we establish a similar result for $n = 3$.

Theorem 3. *When $n = 3$, there exists a connected pseudo 3-partition that is 2-balanced and, thus, $5/3$-proportional, and it can be computed in polynomial time.*

Proof. For a tree $G = (V, E)$ and node $r \in V$, let G_r denote G rooted at r. For a set $S \subseteq V$, let $H(S, G_r)$ denote the union of subtrees rooted at nodes in S in G_r, so $|H(S, G_r)| = \sum_{u \in S} |ST(u, G_r)|$. Note that this can be computed efficiently. To make later analysis simpler, we will use the convention that $|H(\emptyset, G_r)| = \infty$. For a node u, a subset of its children $S \subseteq c(u, T)$ is called "good" under G_r if $|H(S, G_r)| \geqslant (m-2)/4$. For a root r, let $C_{u,r}$ denote the good subset of node u with the smallest $|H(C_u, G_r)|$, i.e.,

$$C_{u,r} = \arg\min_{S \subseteq c(u,G_r):|H(S,G_r)| \geqslant (m-2)/4} |H(S, G_r)|.$$

Note that if $|ST(u, G_r)| - 1 \geqslant (m-2)/4$, then $C_{u,r} \neq \emptyset$, otherwise $C_{u,r} = \emptyset$. Note that computing $C_{u,r}$ effectively requires solving the subset sum problem. Since this can be solved in pseudopolynomial time in general and the elements of the set in our case — the sizes of the subtrees rooted at the children of u — are upper bounded by m, this can be solved in polynomial time for our case.

Let $(u_1, r_1) \in \arg\min_{(u,r) \in V} |H(C_{u,r}, G_r)|$. That is, over all possible combinations of roots and nodes, we find that u_1 has a good subset with the smallest total subtree size with root r_1. First, we claim that $|H(C_{u_1,r_1}, G_{r_1})| < (m-2)/2$. Suppose for contradiction that $|H(C_{u_1,r_1}, G_{r_1})| \geqslant (m-2)/2$. If $H(C_{u_1,r_1}, G_{r_1})$ has at least two subtrees, then the smallest subtree, say rooted at node v, has size at most $|H(C_{u_1,r_1}, G_{r_1})|/2$. Hence, $|H(C_{u_1,r_1} \setminus \{v\}, G_{r_1})| \geqslant |H(C_{u_1,r_1}, G_{r_1})|/2 \geqslant (m-2)/4$. This violates the definition of C_{u_1,r_1} because $C_{u_1,r_1} \setminus \{v\}$ is a good subset of children of u_1 with a smaller total subtree size. On the other hand, if $H(C_{u_1,r_1}, G_{r_1})$ consists of only one subtree, say rooted at node v, then $|ST(v, G_{r_1})| - 1 = |H(C_{u_1,r_1}, G_{r_1})| - 1 \geqslant (m-2)/4$. Hence, $C_{v,r_1} \neq \emptyset$, so $|H(C_{v,r_1}, G_{r_1})| < |H(C_{u_1,r_1}, G_{r_1})|$, which contradicts the definition of u_1.

Set $V_1 = H(C_{u_1,r_1}, G_{r_1})$ and $R = \{u_1\}$. As $(m-2)/4 \leqslant |H(C_{u_1,r_1}, G_{r_1})| < (m-2)/2$, we can write that $|H(C_{u_1,r_1}, G_{r_1})| = (m-2)/4 + x$, where $0 \leqslant x < (m-2)/4$.

First, suppose $x < (3m+2)/20$. Then, $|H(C_{u_1,r_1}, G_{r_1})| < (2m-2)/5$. Let $T' = T \setminus (H(C_{u_1,r_1}, G_{r_1}) \cup \{u_1\})$. Then, we have that $|T'| \geqslant (3m+2)/5 - 1 = (3m-3)/5$. In this case, we run Algorithm 1 on T'; let (V_1', V_2', R') be its output. We set $V_2 = V_1'$ and $V_3 = V_2'$, and update $R \leftarrow R \cup R'$. Then, from Corollary 1, we know that $|V_i| \geqslant \lceil (m-1)/5 \rceil$ for $i \in \{2, 3\}$. As $|V_1| = |H(C_{u_1,r_1}, G_{r_1})| < (2m-2)/5$, we have that $|V_1| \leqslant \lfloor (2m-2)/5 \rfloor \leqslant 2 \cdot \lceil (m-1)/5 \rceil \leqslant 2 \cdot |V_i|$ for each $i \in \{2, 3\}$. Moreover, from Corollary 1, we have that for each $i \in \{2, 3\}$, $|V_i| \leqslant \frac{2}{3} \cdot (|T'| - |R'|) = \frac{2}{3} \cdot (m - |V_1| - 1 - |R'|)$. Thus, we need to show $2 \cdot |V_1| \geqslant \frac{2}{3} \cdot (m - |V_1| - 1 - |R'|)$, which simplifies to $4|V_1| \geqslant m - 1 - |R'|$. Because $|V_1| \geqslant (m-2)/4$, this holds.[3]

Next, we focus on the case where $x \geqslant (3m+2)/20$. We now change the root to u_1. Recall that $V_1 = H(C_{u_1,r_1}, G_{r_1})$. With the root being u_1, we have that for any $u \in V \setminus V_1$, $ST(v, G_{u_1}) \cap V_1 = \emptyset$. We now repeat our search for a good subset with the smallest total subtree size. Specifically, let

$$u_2 \in \arg\min_{u \in V \setminus V_1} |H(C_{u,u_1}, G_{u_1})|.$$

Note that $|H(C_{u_2,u_1}, G_{u_1})| \geqslant |H(C_{u_1,r_1}, G_{r_1})|$, as otherwise we would have found (u_2, u_1) combination when searching for (u_1, r_1).

Let us update $R \leftarrow R \cup \{u_2\}$. We partition $H(C_{u_2,u_1}, G_{u_1})$ into two parts, S_1 and S_2, such that each subtree in $H(C_{u_2,u_1}, G_{u_1})$ is included entirely in one of the two parts and $||S_1| - |S_2||$ is minimized among all such partitions. Assume, without loss of generality, that $|S_1| \geqslant |S_2|$. Let $T' = T \setminus (H(C_{u_1,r_1}, G_{r_1}) \cup H(C_{u_2,u_1}, G_{u_1}) \cup \{u_1\} \cup \{u_2\})$.

We set $V_2 = S_1$ and $V_3 = S_2 \cup T'$. We show that (V_1, V_2, V_3, R) is a connected 2-balanced pseudo 3-partition. First, we claim that $|V_i| \geqslant (m-2)/8 + x/2$ for each $i \in \{2, 3\}$.

Suppose that $|V_2| = |S_1| < (m-2)/8 + x/2$. As $|S_1| \geqslant |S_2|$, we have that

$$|S_1| + |S_2| = |H(C_{u_2,u_1}, G_{u_1})| < (m-2)/4 + x,$$

[3] This holds if $|R'| = 1$. When running Algorithm 1, we can force it to exclude a node by excluding an arbitrary node from the bigger part, as long as the bigger part has at least two nodes, i.e., if the tree passed to Algorithm 1 has at least three nodes. We can force $|T'| \geqslant 3$ if $m \geqslant 6$. For $m \leqslant 5$, it can be checked by brute force that the theorem holds.

which contradicts $|H(C_{u_2,u_1}, G_{u_1})| \geqslant |H(C_{u_1,r_1}, G_{r_1})|$.

Now assume that $|V_3| < (m-2)/8 + x/2$. Then, we have that

$$\begin{aligned} |S_1| = |V_2| &= |T \setminus (V_1 \cup V_3 \cup \{u_1, u_2\})| \\ &\geqslant m - \left(\frac{m-2}{4} + x + \frac{m-2}{8} + \frac{x}{2} + 2\right) \\ &= \frac{5m+6}{8} - \frac{3x}{2} - 2. \end{aligned} \tag{1}$$

Notice that either $|S_1| < (m-2)/4$ or $|S_1| \geqslant (m-2)/4 + x$, as otherwise in the first step we would find S_1 instead of $H(C_{u_1,r_1}, G_{r_1})$. If $|S_1| < (m-2)/4$, then from Equation (1) we get $x > (m-2)/4$, which is impossible because we derived $x < (m-2)/4$ earlier. Thus, we have that $|S_1| \geqslant (m-2)/4 + x$. As each subtree in $H(C_{u_2,u_1}, G_{u_1})$ has size at most $(m-2)/4 - 1$ (otherwise, we would be able to find a better good subset when finding C'_{u_2}), this means that there must be at least two subtrees of $H(C_{u_2,u_1}, G_{u_1})$ in S_1.[4] We partition S_1 into two parts, denoted by A_1 and A_2, such that each subtree in S_1 is fully contained in one of the two parts, and among all such partitions, $||A_1| - |A_2||$ is minimized. Since S_1 has at least two subtrees both parts must be non-empty. Assume, without loss of generality, that $|A_1| \geqslant |A_2|$. As $|S_1| \geqslant (m-2)/4 + x$, we have that $|A_1| \geqslant (m-2)/8 + x/2$. Consider the following partition (S'_1, S'_2) of $H(C_{u_2,u_1}, G_{u_1})$: $S'_1 = A_1$ and $S'_2 = A_2 \cup S_2$. Since $|S'_1| = |A_1| \geqslant (m-2)/8 + x/2 > |S_2|$, where the last inequality holds because $|S_2| \leqslant |V_3|$ and we assumed $|V_3| < (m-2)/8 + x/2$ in this case. Further, since $|A_2| > 0$, we have $|S'_2| > |S_2|$. Hence, (S'_1, S'_2) is a different partition with a higher minimum size, implying that $||S'_1| - |S'_2|| < ||S_1| - |S_2||$, which is the desired contradiction.

From all the above we have that $|V_i| \geqslant (m-2)/8 + x/2$, and hence $2|V_i| \geqslant |V_1|$, for each $i \in \{2,3\}$. Now, we show that $|V_i| \leqslant 2((m-2)/8 + x/2)$ for each $i \in \{2,3\}$. Indeed, notice that

$$\begin{aligned} |V_2| &= |T \setminus (V_1 \cup V_3 \cup \{u_1, u_2\})| \\ &\leqslant m - \left(\frac{m-2}{4} + x + \frac{m-2}{8} + \frac{x}{2} + 2\right) \\ &= \frac{5m+6}{8} - \frac{3x}{2} - 2 \end{aligned}$$

Plugging in $x \geqslant (3m+2)/20$, we get $|V_2| \leqslant 2((m-2)/8 + x/2)$. Since this argument is symmetric, we also similarly have $|V_3| \leqslant 2((m-2)/8 + x/2)$. Thus, $2 \cdot |V_2| \geqslant |V_3|$ and $2 \cdot |V_3| \geqslant |V_2|$, while $|V_1| \geqslant |V_i|$ for $i \in \{2,3\}$ and the theorem follows. □

The tightness of the lower bound from Theorem 1 for $n \in \{2,3\}$ leads us to make the following conjecture:

Conjecture 1. *For any $n \geqslant 2$, every graph admits a connected pseudo n-partition that is 2-balanced (and hence, $(2 - 1/n)$-proportional), and it can be computed efficiently.*

In the next section, we present a series of results which almost resolve this conjecture.

5 Upper Bounds for Higher n

We present three key upper bounds that hold for all $n \geqslant 2$. The first is via a fairly straightforward algorithm that uses Algorithm 1 for $n = 2$ recursively to obtain $(3 + O(n/m))$-balancedness and 3-proportionality. The second algorithm uses the key idea from Algorithm 1 of finding a subtree of some desired size, and iteratively applies it to achieve 4-balancedness and 2-proportionality; while the balancedness approximation gets worse when $n \ll m$, the proportionality approximation improves and matches the lower bound of $2 - 1/n$ from Theorem 1 in the limit when $n \to \infty$. Finally, by making an interesting connection to the literature on *edge partitions* of a tree, we show that $(2 + O(n^2/m))$-balancedness and $(2 - 1/n + O(n^2/m))$-proportionality can be achieved, which matches the respective lower bounds from Theorem 1 for each n in the limit when $m \to \infty$.

[4] The only exception is when $H(C_{u_2,u_1}, G_{u_1})$ is composed of a single subtree of size exactly $(m-2)/4$ and $S_1 = H(C_{u_2,u_1}, G_{u_1})$. However, since $|H(C_{u_2,u_1}, G_{u_1})| \geqslant |H(C_{u_1,r_1}, G_{r_1})| = (m-2)/4 + x$, we have $x = 0$, so $|V_1| = |V_2| = (m-2)/4$. In this case, we have $|V_3| \geqslant m - |V_1| - |V_2| - 2 = m - (m-2)/2 - 2 = (m+2)/2 \geqslant (m-2)/8$, which is the desired contradiction.

Algorithm 2: $(3 + O(n/m))$-balancedness and 3-proportionality for $n \geqslant 2$

Input: Tree $G = (V, E)$ and integer $n \geqslant 2$.
Output: A connected pseudo n-partition.

1: $C^0 \leftarrow \{G\}$; $R^0 \leftarrow \emptyset$
2: **for** $i = 1$ to $n - 1$ **do**
3: $T^i \leftarrow$ largest tree in C^{i-1} (break ties arbitrarily)
4: $(V_1^i, V_2^i, \hat{R}^i) \leftarrow$ Call Algorithm 1 on T^i
5: $H_1^i \leftarrow T^i[V_1^i]$, $H_2^i \leftarrow T^i[V_2^i]$
6: **if** $\hat{R}^i \neq \emptyset$ **then**
7: Let $u^i \in \hat{R}^i$ {This is unique}
8: For $j \in \{1, 2\}$, if H_j^i has at least two neighbors of u^i, connect an arbitrarily chosen neighbor to every other neighbor {This ensures that H_j^i is now a tree}
9: **end if**
10: $C^i \leftarrow C^{i-1} \cup \{H_1^i, H_2^i\} \setminus T^i$
11: $R^i \leftarrow R^{i-1} \cup \hat{R}^i$
12: **end for**
13: **return** $(V_1, \ldots, V_n, R)$, where $V_1, \ldots, V_n$ are the sets of nodes of the trees in C^{n-1} and $R = R^{n-1}$.

Let us begin with our first result of this section. At a high level, Algorithm 2 works simply as follows: it starts with the entire input tree as a single part, and repeatedly divides the largest existing part into two using Algorithm 1 until there are n parts. One issue is that when Algorithm 1 excludes a node, the two parts it returns may become disconnected, preventing us from applying Algorithm 1 to them in future iterations; this is because Algorithm 1 assumes its input to be a tree. This is easily fixed by adding artificial edges between the neighbors of the excluded node to ensure that the parts returned by Algorithm 1 become trees. This is not a problem because if a part returned at the end of Algorithm 2 is connected due to the artificially added edges, it would also be connected via the excluded nodes.

Theorem 4. *When $n \geqslant 2$ and $m \geqslant n \cdot (n-1)$, Algorithm 2 runs in polynomial time and returns a connected pseudo n-partition that is $(3 + 6n/m)$-balanced and 3-proportional.*

Proof. First, we argue that the algorithm is valid. Because we call Algorithm 1 once in each iteration and Line 8 converts any parts it returns into trees, we can see that C^i contains $i+1$ trees and R^i contains at most i nodes for each $i \in \{0\} \cup [n-1]$. Since $m \geqslant n$, the largest tree T^i in the i-th iteration has at least two nodes, allowing us to call Algorithm 1. Finally, as argued above, the pseudo n-partition returned is connected since the use of any artficial edge added by Line 8 can be replaced by a path that uses the excluded nodes. We now prove the fairness guarantees.

Using induction on i, we show that $3 \cdot \min_{T \in C^i} |T| + 1 \geqslant \max_{T \in C^i} |T|$ and $\min_{T \in C^i} |T| \geqslant \frac{m - |R^i|}{3(i+1)}$ for $i \in \{0\} \cup [n-1]$. First, let us argue that this implies the desired approximations. The second part of the inductive claim, applied at $i = n-1$, already yields 3-proportionality. Let $T_{\max}$ and $T_{\min}$ be the largest and the smallest trees in C^{n-1}, respectively. Note that $|T_{\min}| \geqslant \frac{m-n+1}{3n} \geqslant \frac{m}{6n}$, where the first inequality follows from the proportionality guarantee and the second inequality follows from the fact that $m \geqslant 2n - 1$. Now, with the former part of the inductive claim applied at $i = n-1$, we have

$$|T_{\max}| \leqslant 3 \cdot |T_{\min}| + 1 \leqslant 3 \cdot |T_{\min}| + \frac{6n}{m} \cdot |T_{\min}|,$$

implying the desired approximation of balancedness.

It is easy to check that the induction claim holds in the base case of $i = 0$. Fix $i \in [n-1]$ and assume it holds for iterations $1, \ldots, i-1$. Consider iteration i.

For the first part, recall that we call Algorithm 1 on the largest tree T^i in C^{i-1}. By Corollary 1, we have that $\min(|V_1^i|, |V_2^i|) \geqslant \lceil (|T^i| - 1)/3 \rceil$. Hence, $3 \cdot \min(|V_1^i|, |V_2^i|) + 1 \geqslant |T^i|$. By the first part of the induction hypothesis at iteration $i-1$, we have that $3 \cdot \min_{T \in C^{i-1}} |T| \geqslant \max_{T \in C^{i-1}} |T| = |T^i|$. For C^i, note that

$\min_{T \in C^i} |T| \geqslant \min(\min_{T \in C^{i-1}} |T|, |V_1^i|, |V_2^i|)$, whereas $\max_{T \in C^i} |T| \leqslant \max_{T \in C^{i-1}} |T| = |T^i|$. Hence, the first part of the induction hypothesis holds at iteration i.

For the second part, note that the second part of the induction hypothesis at iteration $i-1$ already implies

$$\min_{T \in C^{i-1}} |T| \geqslant \tfrac{m-|R^{i-1}|}{3i} \geqslant \tfrac{m-|R^i|}{3(i+1)}.$$

Hence, we only need to establish that $\min(|V_1^i|, |V_2^i|) \geqslant \frac{m-|R^i|}{3(i+1)}$ as well. By the pigeonhole principle, the largest tree T^i in C^{i-1} on which we call Algorithm 1 in the i-th iteration must have size at least $\frac{m-|R^{i-1}|}{i}$. Using Corollary 1, we have

$$\min(|V_1^i|, |V_2^i|) \geqslant \frac{|T^i| - |\hat{R}^i|}{3} \geqslant \frac{\frac{m-|R^{i-1}|}{i} - |\hat{R}^i|}{3}.$$

When $|\hat{R}^i| = 0$, it is easy to see that this is at least $\frac{m-|R^{i-1}|}{3i} \geqslant \frac{m-|R^i|}{3(i+1)}$. Hence, assume $|\hat{R}^i| = 1$. Then, we need

$$\frac{\frac{m-|R^{i-1}|}{i} - 1}{3} \geqslant \frac{m - |R^{i-1}| - 1}{3(i+1)}.$$

After some simplification, we see that this is equivalent to $m \geqslant i^2 + |R^{i-1}|$, which holds since $|R^{i-1}| \leqslant i \leqslant n-1$. □

Next, we show how to achieve 4-balancedness and 2-proportionality. We use the key idea from Algorithm 1 of finding a subtree of some desired size and iteratively apply it to separate out one part at a time from the tree. An interesting detail, and the driving force behind the balancedness guarantee, is that because we cannot exactly control the size of the parts being separated out, we keep adjusting the desired size of the next part based on the actual size of the previous part created. This ensures that when ℓ parts are created, their total size stays close to $\ell \cdot m/n$, leaving the size of the remaining tree close to $(n-\ell) \cdot m/n$. In particular, after $n-1$ parts are created, the remaining tree, much of which forms the last part, is not too large.

To make our analysis work, we need to ensure that $|R| = n-1$. Hence, we need $m \geqslant 2n-1$, so that even after removing $n-1$ nodes, we can always create an n-partition with non-empty parts. We remark that Line 6 can be implemented efficiently similarly to Line 3 of Algorithm 1.

When Line 20 of Algorithm 3 excludes node u_i in iteration i, T_i may become disconnected as the subtrees rooted at children of u_i become disconnected from each other and from the rest of the tree. This is fixed by adding artificial edges connecting every child of u_i that remains in T_i to the parent of u_i. As mentioned above, if a part is connected using these artificial edges, it is also connected using excluded nodes instead. If u_i is the root of the tree, we can imagine creating an artificial new root node, connecting it to all children of u_i, but not counting this artificial root node in future computations of subtree sizes. Note that, unlike in Algorithm 2, we do not just connect an arbitrary neighbor of u_i in T_i to its remaining neighbors because this can alter the rooted tree structure, which we use in this algorithm.

Theorem 5. *When $n \geqslant 2$ and $m \geqslant 2n-1$, Algorithm 3 runs in polynomial time and returns a connected pseudo n-partition that is 4-balanced and 2-proportional.*

Proof. As explained above, the addition of artificial edges in Line 19 ensure that the remaining graphs (T_i-s) are trees and the parts being created (V_i-s) are connected via the excluded nodes. Later, in Lemma 2, we will establish that for $i \in [n-1]$, $x_{i-1} \leqslant 1$ and $|T_i| \geqslant \lceil (n-i) \cdot s \rceil \geqslant \lceil s \rceil \geqslant \lceil s(1+x_{i-1})/2 \rceil$. Hence, the algorithm will be able to successfully find node u_i in every iteration i and proceed without any issues. Since at most a single node is added to R in each of $n-1$ iterations, we clearly have $|R| \leqslant n-1$. This establishes that the algorithm is valid (i.e., it produces a connected pseudo n-partition at the end. It is also easy to see that the algorithm runs in polynomial time). Hence, it remains to establish its balancedness and proportionality guarantees.

As $m \geqslant 2n-1$, we have that $s = \frac{m-(n-1)}{n} \geqslant 1$. Before proceeding further, we need the following observation.

Lemma 1. *For any $y \geqslant 0$, $s(1+y) \geqslant \lceil s(1+y)/2 \rceil$.*

Algorithm 3: 4-balancedness and 2-proportionality for $n \geqslant 2$

Input: Tree $G = (V, E)$ and integer $n \geqslant 2$.
Output: A connected pseudo n-partition.

1: $r \leftarrow$ arbitrary node in V
2: $T \leftarrow$ tree (G, r) rooted at r
3: $R \leftarrow \emptyset$;$\forall i \in [n], V_i \leftarrow \emptyset$
4: $s \leftarrow \frac{m-(n-1)}{n}$, $x_0 \leftarrow 0$, $T_1 \leftarrow T$
5: **for** $i = 1$ to $n-1$ **do**
6: Find a node u_i such that $|ST(u_i, T_i)| \geqslant \lceil s(1 + x_{i-1})/2 \rceil > |ST(v, T_i)|$ for all $v \in c(u_i, T_i)$
7: **if** $|ST(u_i, T_i)| = \lceil s(1 + x_{i-1})/2 \rceil$ **then**
8: $V_i \leftarrow ST(u_i, T_i)$
9: $T_{i+1} \leftarrow T_i \setminus ST(u_i, T_i)$
10: **else**
11: $R = R \cup \{u_i\}$
12: **for** $u' \in c(u_i, T_i)$ **do**
13: $V_i \leftarrow V_i \cup ST(u', T_i)$
14: **if** $|V_i| \geqslant \lceil s(1 + x_{i-1})/2 \rceil$ **then**
15: **break**
16: **end if**
17: **end for**
18: $T_i \leftarrow T_i \setminus V_i$
19: Connect each $v \in c(u_i, T_i)$ to $p(u_i, T_i)$
20: $T_{i+1} \leftarrow T_i \setminus \{u_i\}$
21: **end if**
22: $x_i \leftarrow 1 + x_{i-1} - |V_i|/s$
23: **end for**
24: $S \leftarrow$ set of $n - 1 - |R|$ arbitrary nodes from T_n
25: $V_n \leftarrow T_n \setminus S$, $R \leftarrow R \cup S$
26: **return** $(V_1, \ldots, V_n, R)$

Proof. As $s \geqslant 1$ and $y \geqslant 0$, we have $s(1 + y) \geqslant 1$. Now, if $s(1 + y) \geqslant 2$, then we have

$$\lceil s(1 + y)/2 \rceil \leqslant s(1 + y)/2 + 1 \leqslant s(1 + y).$$

Otherwise, we have $2 > s(1 + y) \geqslant 1$, so $s(1 + y) \geqslant 1 = \lceil s(1 + y)/2 \rceil$. □

Next, we prove the following lemma inductively, and establish several structural properties that hold during the execution of the algorithm.

Lemma 2. *For each $i \in \{0\} \cup [n-1]$, the following hold:*

- $0 \leqslant x_i \leqslant 1$,
- $\lceil s(1 + x_{i-1})/2 \rceil \leqslant |V_i| \leqslant s(1 + x_{i-1})$ *if* $i \geqslant 1$,
- $|\cup_{j \in [i]} V_j| = (i - x_i) \cdot s$, *and*
- $|T_{i+1}| \geqslant \lceil (n - i) \cdot s \rceil$.

Proof. We prove the lemma using induction on i. The base case of $i = 0$ trivially holds because $x_0 = 0$ and $T_1 = T$. Fix $i \geqslant 1$. Suppose the induction hypothesis holds for $0, 1, \ldots, i-1$.

Note that $|V_i| \geqslant \lceil s(1 + x_{i-1})/2 \rceil$ holds by construction (Lines 7 and 14). If the condition in Line 7 works, then we have $|V_i| = \lceil s(1 + x_{i-1})/2 \rceil \leqslant s(1 + x_{i-1})$ by Lemma 1. Otherwise, since we keep adding subtrees of size at most

$\lceil s(1+x_{i-1})/2\rceil - 1$ until $|V_i| \geqslant \lceil s(1+x_{i-1})/2\rceil$ (Line 14), we have $|V_i| \leqslant 2\cdot(\lceil s(1+x_{i-1})/2\rceil - 1) \leqslant s(1+x_{i-1})$. Hence, the second claim holds.

For the third claim, we use the fact that $|\cup_{j\in[i]} V_j| = |\cup_{j\in[i-1]} V_j| + |V_i| = (i-1-x_{i-1})\cdot s + |V_i|$. To establish that this is equal to $(i-x_i)\cdot s$, we need $|V_i| = (1+x_{i-1}-x_i)\cdot s$, which holds by the definition of x_i in Line 22.

For the fourth claim, since at most $n-1$ nodes are excluded at any point during the execution of the algorithm, we have

$$\begin{aligned}|T_{i+1}| &\geqslant m-(n-1)-|\cup_{j\in[i]} V_j| \\ &= n\cdot s-(i-x_i)\cdot s \geqslant (n-i)\cdot s.\end{aligned}$$

Since $|T_{i+1}|$ is an integer, we also have $|T_{i+1}| \geqslant \lceil (n-i)\cdot s\rceil$.

For the first claim, recall that $x_i = 1+x_{i-1}-|V_i|/s$. But $(1+x_{i-1})/2 \leqslant |V_i|/s \leqslant 1+x_{i-1}$ from the second claim. Hence, $0 \leqslant x_i \leqslant (1+x_{i-1})/2$. Using $x_{i-1} \leqslant 1$ from the induction hypothesis, we get $0 \leqslant x_i \leqslant 1$ as desired. □

Combining the first two claims from Lemma 2, we have that $\lceil s/2\rceil \leqslant |V_i| \leqslant 2s$ for $i \in [n-1]$. Let us now estimate $|V_n|$. From the third claim of Lemma 2 applied at $i = n-1$, we have

$$\begin{aligned}|T_n| &= |T \setminus (\cup_{i\in[n-1]} V_i \cup R)| \\ &= m-(n-1-x_{n-1})\cdot s-|R|\end{aligned}$$

Note that $V_n = T_n \setminus S$, where S is a set of $n-1-|R|$ arbitrary nodes from T_n. Hence,

$$|V_n| = m-(n-1-x_{n-1})\cdot s-(n-1) = (1+x_{n-1})\cdot s,$$

where the second transition follows since $m-(n-1) = n\cdot s$. Using $0 \leqslant x_{n-1} \leqslant 1$ from Lemma 2, we have $s \leqslant |V_n| \leqslant 2s$. Hence, in conclusion, we have $\lceil s/2\rceil \leqslant |V_i| \leqslant 2s$ for all $i \in [n]$, which clearly implies 4-balancedness. Since we force $|R| = n-1$, we have $s = (m-(n-1))/n = (m-|R|)/n$, so this also implies 2-proportionality. □

Next, we show that $(2+O(n^2/m))$-balancedness and $(2-1/n+O(n^2/m))$-proportionality can be obtained by making a connection to the literature on *edge partitions* of trees. We say that $(E_1,\ldots,E_n)$ is an *n-edge-partition* of a tree $G = (V,E)$ if $E_i \cap E_j = \emptyset$ for all distinct $i,j \in [n]$ and $\cup_{i\in[n]} E_i = E$. We say that it is *connected* if, for each $i \in [n]$, the subgraph formed by the edges in E_i is connected (hence, also a tree). For $\alpha \geqslant 1$, we say that it is α-balanced if $\max_{i\in[n]} |E_i| \leqslant \alpha\cdot\min_{i\in[n]} |E_i|$ and α-proportional if $\alpha\cdot\min_{i\in[n]} |E_i| \geqslant |E|/n$, where $|E_i|$ and $|E|$ refer to the number of *edges* in those sets. Observe that α-balancedness also implies $(\alpha-(\alpha-1)/n)$-proportionality in this context. In particular, 2-balancedness implies $(2-1/n)$-proportionality.

Note that edge partitions are similar to node partitions, except we seek to partition the edges without excluding any edges. For connected node partitions, we argued in Section 1, using the star graph as an example, that no reasonable approximation of balancedness or proportionality can be obtained without excluding any nodes. However, it turns out that there exist reasonably balanced and proportional edge partitions of a tree that do not require any edge exclusions.

Theorem 6 (16)**.** *For any $n \geqslant 2$, every tree admits a connected n-edge-partition that is 2-balanced and, hence, $(2-1/n)$-proportional, and such a partition can be computed in polynomial time.*

In the following lemma, we show that a connected n-edge-partition of a tree (with no edge exclusions) can be used to obtain a connected pseudo n-partition of the nodes (with at most $n-1$ node exclusions) while almost preserving the balancedness and proportionality guarantees.

Before we proceed further, recall that for node partitions, our assumption of the input graph being a tree was without loss of generality because a connected pseudo n-partition of a spanning tree of the graph is also a connected pseudo n-partition of the graph itself; both the graph and its spanning tree have the same set of nodes. This does not hold for edge partitions. In particular, an n-edge-partition of a spanning tree of a graph is not even an n-edge-partition of the graph, since the additional edges in the graph not included in the spanning tree also need to be partitioned. In that sense, we are using the aforementioned result on edge partitions for the *special case* of trees to derive a result on pseudo node partitions for general graphs.

Lemma 3. *For any $n \geqslant 2$, given a connected n-edge-partition $(E_1, \ldots, E_n)$ of a tree $G = (V, E)$, we can compute, in polynomial time, a connected pseudo n-partition $(V_1, \ldots, V_k, R)$ of V (i.e., with $|R| \leqslant n-1$) such that $|E_i|+1-|R| \leqslant |V_i| \leqslant |E_i| + 1$ for each $i \in [n]$.*

Proof. Given a subset of edges E', let $V(E')$ denote the set of nodes with at least one edge of E' incident on them. Construct a multigraph H with $E_1, \ldots, E_n$ as nodes. For all distinct $i, j \in [n]$ and $v \in V(E_i) \cap V(E_j)$, add an edge between the nodes representing E_i and E_j with label 'v'.

We argue that this multigraph is acyclic. For the sake of contradiction, suppose that it has a cycle $(E_{i_1}, E_{i_2}, \ldots, E_{i_\ell}, E_{i_1})$. For $r \in [\ell]$, let v_r be the label of the edge of the cycle connecting E_{i_r} to $E_{i_{r+1}}$ (or to E_{i_1} if $r = \ell$). For $r \in [\ell]$, since $V(E_r)$ includes both v_{r-1} (or v_ℓ if $r = 1$) and v_r, there is a path between these vertices in G. Combining these paths together, we obtain a cycle in G, which is impossible since G is a tree.

Hence, H is acyclic. Since it has n nodes, it must have at most $n-1$ edges. Let R be the set of nodes appearing as labels of edges in H. Then, $|R| \leqslant n-1$. Further, for each $i \in [n]$, set $V_i = V(E_i) \setminus R$. Recalling that $|V(E_i)| = |E_i|+1$ for a tree E_i, It is easy to see that $|E_i| + 1 - |R| \leqslant |V_i| \leqslant |E_i| + 1$. Finally, since $V(E_i)$ is connected, it follows that with $R_i = V(E_i) \cap R$, $V_i \cup R_i = V(E_i)$ is connected, implying that $(V_1, \ldots, V_n, R)$ is a connected pseudo n-partition of G, as desired. □

We now use Lemma 3 to translate the guarantee in Theorem 6 to our setting.

Theorem 7. *When $n \geqslant 2$ and $m \geqslant 4n^2$, every graph admits a connected pseudo n-partition of its nodes that is $(2 + 8n^2/m)$-balanced and $(2 - 1/n + 8n^2/m)$-proportional, and one such solution can be computed in polynomial time.*

Proof. As mentioned before, for our node partition problem, we can assume the input graph G to be a tree. Note that it has m nodes and $m-1$ edges. Consider the connected pseudo n-partition $(V_1, \ldots, V_n, R)$ of its nodes produced by applying Lemma 3 to the n-edge-partition $(E_1, \ldots, E_n)$ provided by Theorem 6. From Theorem 6, we have that $\min_{i \in [n]} |E_i| \geqslant \frac{m-1}{2n-1}$ (the proportionality guarantee) and $2 \min_{i \in [n]} |E_i| \geqslant \max_{i \in [n]} |E_i|$ (the balancedness guarantee).

Using Lemma 3 and the fact that $|R| \leqslant n-1$, we obtain that

$$\begin{aligned}\min_{i \in [n]} |V_i| &\geqslant \min_{i \in [n]} |E_i| + 1 - (n-1) \geqslant \frac{m-1}{2n-1} + 2 - n \\ &= \frac{m - (2n-3)(n-1)}{2n-1} \geqslant \frac{m - 2n^2}{2n-1}.\end{aligned}$$

Because $m \geqslant 4n^2$, we have that

$$\min_{i \in [n]} |V_i| \geqslant \frac{m}{2(2n-1)}. \tag{2}$$

It is also easy to check that when $m \geqslant 4n^2$, we have that

$$\left(1 + \frac{4n^2}{m}\right) \cdot \left(1 - \frac{2n^2}{m}\right) = 1 + \frac{2n^2}{m}\left(1 - \frac{4n^2}{m}\right) \geqslant 1.$$

Hence,

$$\min_{i \in [n]} |V_i| \geqslant \frac{m - 2n^2}{2n-1} = \frac{m \cdot (1 - \frac{2n^2}{m})}{n \cdot (2 - 1/n)} \geqslant \frac{m}{n \cdot (2 - 1/n) \cdot (1 + 4n^2/m)} \geqslant \frac{m}{n \cdot (2 - 1/n + 8n^2/m)},$$

which establishes the desired proportionality approximation. Now, for balancedness, we notice that

$$\begin{aligned}\max_{i \in [n]} |V_i| &\leqslant \max_{i \in [n]} |E_i| + 1 \leqslant 2 \min_{i \in [n]} |E_i| + 1 \\ &\leqslant 2\left(\min_{i \in [n]} |V_i| + n - 2\right) + 1\end{aligned}$$

$$
\begin{aligned}
&= 2 \min_{i \in [n]} |V_i| + (2n-3) \\
&\leqslant 2 \min_{i \in [n]} |V_i| + \frac{(2n-3) \cdot 2 \cdot (2n-1) \cdot \min_{i \in [n]} |V_i|}{m} \\
&\leqslant (2 + 8n^2/m) \cdot \min_{i \in [n]} |V_i|,
\end{aligned}
$$

where the first and the third transitions follow from Lemma 3, the second transition follows from the balancedness guarantee in Theorem 6, and the fifth transition follows from Equation (2). This completes the proof. □

For fixed n, in the limit when $m \to \infty$, Theorem 7 provides 2-balancedness and $(2 - 1/n)$-proportionality, matching the lower bound from Theorem 1 and settling Conjecture 1. However, when m is not too large, the guarantee provided by Theorem 4 or Theorem 5 can be better.

6 The Fairness-Charity Tradeoff

In this section, we consider the tradeoff between the limit on charity (the maximum number of nodes we are allowed to exclude) and the approximations to balancedness and proportionality we can guarantee. Given a graph $G = (V, E)$ and $d \in \{0\} \cup \mathbb{N}$, $(V_1, \ldots, V_n, R)$ is called a d-pseudo n-partition of G if it is a partition of V and $|R| \leqslant d$. As before, we say that it is connected if, for each $i \in [n]$, there exists $R_i \subseteq R$ such that $G[V_i \cup R_i]$ is a connected subgraph of G.

The next two results focus on $d > n - 1$ and provide an almost tight tradeoff. Let us introduce the lower bound first.

Theorem 8. *Fix any $m, n \geqslant 2$ and $c \geqslant 0$ such that $\ell = \frac{m-n+1}{2n-1} \in \mathbb{N}$ and $\ell > (c+1) \cdot (n-1)$. Then, there exists an instance with m nodes in which no connected d-pseudo n-partition is $(2 - c/\ell)$-balanced when $d < (c+1) \cdot (n-1)$, and no connected d-pseudo n-partition is α-balanced for any $\alpha < 2 - c/\ell$ when $d = (c+1) \cdot (n-1)$.*

Proof. Consider the instance in the proof of Theorem 1. It can be checked that the first part of the proof holds whenever $|R| < \ell$ (which holds because the theorem statement assumes that $\ell > (c+1) \cdot (n-1) \geqslant d \geqslant |R|$), and establishes that any d-pseudo n-partition $(V_1, \ldots, V_n, R)$ must admit $i \in [n]$ such that $|V_i| \leqslant \ell$.

If this partition is α-balanced, then $|V_j| \leqslant \alpha \cdot \ell$ for all $j \in [n] \setminus \{i\}$. Using the definition of ℓ, we have that

$$
\ell + (n-1) \cdot \alpha \cdot \ell + d \geqslant |V_i| + \sum_{j \in [n] \setminus \{i\}} |V_j| + |R| = m = (2n-1)\ell + (n-1),
$$

which implies

$$
\alpha \geqslant 2 - \frac{\frac{d}{n-1} - 1}{\ell}.
$$

Hence, when $d = (c+1) \cdot (n-1)$, we get $\alpha \geqslant 2 - c/\ell$, and when $d < (c+1) \cdot (n-1)$, we get $\alpha > 2 - c/\ell$, as needed. □

One implication of this lower bound is that if we hope to achieve α-balancedness for any *constant* $\alpha < 2$, then we must have $c = \Omega(\ell)$, i.e., $d = \Omega(m)$. Hence, a little charity ($o(m)$ exclusions) would not suffice for this purpose. This shows that 2 is the best constant approximation to balancedness we can hope for with just a little charity. Next, we provide an upper bound via a simple algorithm which starts with any α-balanced connected pseudo n-partition (i.e., with at most $n - 1$ exclusions) and repeatedly excludes a node from the largest part until either perfect balancedness is achieved or a total of d nodes are excluded.

Theorem 9. *Fix any $m, n \geqslant 2$, $c \geqslant 0$, $d = (c+1) \cdot (n-1)$, $\alpha \geqslant 1$, and $\hat{\ell} = \frac{m-n+1}{\alpha n - (\alpha - 1)}$. Given any graph of m nodes and any connected $(n-1)$-pseudo n-partition of it that is α-balanced, we can efficiently compute a d-pseudo n-partition that is $(\alpha - c/\hat{\ell})$-balanced.*

Proof. Let $(V_1, \ldots, V_n, R)$ be a connected pseudo n-partition with $R \leqslant n-1$ that is α-balanced. Then, we have $\max_{i\in[n]} |V_i| \leqslant \alpha \cdot \min_{i\in[n]} |V_i|$ and, since α-balancedness implies $(\alpha - (\alpha-1)/n)$-proportionality, we also have

$$\min_{i\in[n]} |V_i| \geqslant \frac{m - |R|}{\alpha n - (\alpha - 1)} \geqslant \frac{m - n + 1}{\alpha n - (\alpha - 1)} = \hat{\ell}. \tag{3}$$

Now, let us repeatedly exclude a node from the largest part until either 1-balancedness is achieved or a total of $d = (c+1)\cdot(n-1)$ nodes are excluded. Let $(\hat{V}_1, \ldots, \hat{V}_n, \hat{R})$ be the resulting d-pseudo n-partition. Trivially, note that it is still connected. If it is 1-balancedness, we are done. Otherwise, note that we must have excluded at least $d - |R| \geqslant c\cdot(n-1)$ additional nodes. Since we never touch the smallest part, the size of the largest part must reduces by at least 1 after every $n-1$ exclusions. Thus, at the end, we must have

$$\begin{aligned}
\max_{i\in[n]} |\hat{V}_i| &\leqslant \alpha \cdot \min_{i\in[n]} |\hat{V}_i| - c \\
&= \alpha \cdot \min_{i\in[n]} |V_i| - c \\
&\leqslant (\alpha - c/\hat{\ell}) \cdot \min_{i\in[n]} |V_i| \\
&= (\alpha - c/\hat{\ell}) \cdot \min_{i\in[n]} |\hat{V}_i|,
\end{aligned}$$

where the second inequality follows from Equation (3). This is the desired result. □

In Section 5, we established that α-balanced connected pseudo n-partitions exist for $\alpha \approx 2$ (in particular, with $\alpha \to 2$ when $m \to \infty$). Note that with $\alpha = 2$, the upper bound from Theorem 9 would precisely match the lower bound from Theorem 8. Thus, assuming that 2-balanced connected pseudo n-partitions exist, taking such a partition and repeatedly excluding a node from the largest part provides optimal balancedness for any $d > n-1$.

With $d < n-1$, the situation becomes more complex as it does not seem easy to start from a connected pseudo n-partition with (at most) $n-1$ exclusions and re-include some nodes while maintaining n connected parts. First, we show that decreasing the charity limit by just one increases the balancedness lower bound from 2 to 3.

Theorem 10. *For any $n \geqslant 2$, $d < n-1$, and $\epsilon > 0$, there exists an instance in which no connected d-pseudo n-partition is α-balanced for any $\alpha < 3 - \epsilon$.*

Proof. Let $\ell \geqslant n-1$ be an integer such that $(n-2)/\ell \leqslant \epsilon$. Consider the graph $G = (V, E)$ that consists of $2n$ paths of length ℓ each, denoted by $P_1, \ldots, P_{2n}$, and $n-1$ hub nodes denoted by $h_1, \ldots, h_{n-1}$. Hence, $|V| = 2\ell n + n - 1$. For $j \in [n-2]$, h_j is connected to h_{j+1}; h_1 is connected to one of the endpoints of paths P_1, P_2 and P_3; for $j \in \{2, \ldots, n-2\}$, hub h_j is connected to one of the endpoints of the paths P_{2j} and P_{2j+1}; and hub h_{n-1} is connected to one of the endpoints of the paths P_{2n-2}, P_{2n-1} and P_{2n}. Assume for contradiction that $(V_1, \ldots, V_n, R)$ is a connected α-balanced d-pseudo n-partition with $\alpha < 3 - \epsilon$ and $d < n-1$.

First, we show that each V_i intersects with at most two paths. Assume for contradiction that for some $i^* \in [n]$, V_{i^*} intersects r paths, denoted $P_{j_1}, \ldots, P_{j_r}$, with $r \geqslant 3$.

Suppose no other part intersects with any of the paths $P_{j_1}, \ldots, P_{j_r}$. Then, we have $|V_{i^*}| \geqslant 3\ell - (n-2)$. Since $(V_1, ..., V_n, R)$ is α-balanced with $\alpha < 3 - \epsilon$, we need $|V_{i'}| > \ell$ for each $i' \in [n] \setminus \{i^*\}$, given that $(n-2)/\ell \leqslant \epsilon$. We show that for each $i' \in [n] \setminus \{i^*\}$, $V_{i'}$ intersects with at least two paths from $P_{j_1}, \ldots, P_{j_r}$. If this is false for some $V_{i'}$, then it must contain at least one hub h_j. Then, none of the paths attached to h_j can intersect with any part other than $V_{i'}$, or else this part would have size at most ℓ. Hence, each path attached to hub h_j must be contained in $V_{i'} \cup R$. Since no path can be fully contained in R (as $|R| < n-1 \leqslant \ell$), each path attached to h_j must intersect with $V_{i'}$. Since there are at least two paths connected to any hub, this contradicts $V_{i'}$ not intersecting with at least two paths. Thus, we have that V_{i^*} intersects with at least 3 paths, and every other $V_{i'}$ intersects with at least two paths. Further, no two parts can intersect the same path, otherwise one of them would have size at most ℓ. However, this requires at least $3 + 2\cdot(n-1) = 2n+1$ paths in total, but we only have $2n$ paths. This is a contradiction.

Now, suppose that there are q parts, with $q > 0$, denoted $V_{i_1}, \ldots, V_{i_q}$ that intersect with at least one of the paths $P_{j_1}, \ldots, P_{j_r}$. Since V_{i^*} intersects with all of them, by the connectedness requirement, it must be the case that any V_{i_p}

intersects with only one of these paths, and hence $|V_{i_p}| \leqslant \ell - 1$ for any $p \in [q]$. If $r \geqslant q+3$, then V_{i^*} intersects with at least three paths that no other part intersects, and thus $|V_{i^*}| \geqslant 3\ell - (n-2)$. But then, $(V_1, ..., V_n, R)$ is not α-balanced with $\alpha < 3 - \epsilon$. Thus, $r \leqslant q+2$. But, then at least $2n - (q+2)$ paths remain and $n - (q+1)$ parts can intersect with them, and as $2n - q - 2 > 2 \cdot (n - q - 1)$ when $q > 0$, from the pigeon-hole principle we have that there is part $V_{i''}$ such that $|V_{i''}| \geqslant 3\ell - (n-2)$. But as $|V_{i_p}| \leqslant \ell - 1$ for any $p \in [q]$, the partition is not α-balanced with $\alpha < 3 - \epsilon$.

Hence, we have established that each V_i intersects with at most two paths. Now, assume that there exists a part V_i that intersects with at most one path. Then, as each $V_{i'} \in [n] \setminus \{i\}$ intersects with at most two paths, this means that at most $2n - 1$ paths intersect with some part, and hence all the nodes of at least one path should be excluded which is impossible as $\ell \geqslant n - 1 > d$. Hence, we conclude that each V_i intersects with exactly two paths.

If h_1 is not excluded, then as no part can intersect with one or three paths, all the nodes of at least one path among P_1, P_2 and P_3 should be excluded which is impossible as $\ell \geqslant n - 1 > d$. Thus, h_1 is excluded. With the same arguments, we conclude that h_{n-1} should be excluded. Now, assume that some h_{j^*} with $j \in [n-2] \setminus 1$ is not excluded, but instead h_{j^*} belongs to some V_i. Then, V_i should intersects with both $P_{2(j-1)+2}$ and $P_{2(j-1)+3}$, and any part that intersects with $\cup_{j \in [j^*-1]} P_{2(j-1)+1} \cup P_{2(j-1)+1} \cup h_j$ cannot intersect with $\cup_{j \in \{j^*+1,..,n-1\}} P_{2(j-1)+1} \cup P_{2(j-1)+1} \cup h_j$, due to connectivity constraints. But now we see that in the former set there is only an odd number of the paths, i.e. P_1 to $P_{2(j^*-1)+1}$, and thus there is no way to assign them into parts such that each part intersects with exactly two of them. □

Next, we establish a different lower bound that is better when $d < n/3$.

Theorem 11. *For any $n \geqslant 2$ and $d < n$, there exists an instance in which no connected d-pseudo n-partition is α-balanced for any $\alpha < n/d$.*

Proof. Let $\ell \geqslant n - 1$. Let L be a large integer (to be decided later). Consider the graph $G = (V, E)$ that consists of $d+1$ hubs $h_1, \ldots, h_{d+1}$ connected in a path, with each hub connected to an endpoint of L disjoint paths of length ℓ each. Thus, in total, the graph has $m = (d+1) \cdot (L \cdot \ell + 1)$ nodes. Let $(V_1, \ldots, V_n, R)$ be any connected d-pseudo n-partition. Note that at least one of the hubs cannot be excluded. Suppose that $h_j \in V_i$ for some $j \in [d+1]$ and $i \in [n]$.

First, suppose that some other $V_{i'}$ intersects with at least one of the L paths connected to h_j. Then, due to the connectedness constraint, we have $|V_{i'}| \leqslant \ell$. However, by the pigeonhole principle, there must exist i'' such that $|V_{i''}| \geqslant (1/n) \cdot ((d+1) \cdot (L \cdot \ell + 1) - d)$. Hence,

$$\frac{|V_{i''}|}{|V_{i'}|} \geqslant \frac{(1/n) \cdot ((d+1) \cdot (L \cdot \ell + 1) - d)}{\ell},$$

and the right hand side grows arbitrarily as $L \to \infty$. Hence, for a sufficiently large L, this establishes a contradiction with α-balancedness for $\alpha < n/(d+1)$.

Next, suppose no other part intersects with any of the L paths connected to h_j. Then, we have $|V_i| \geqslant L \cdot \ell + 1 - d$. On the other hand, by the pigeonhole principle, there exists i' such that $|V_{i'}| \leqslant (1/n) \cdot (d+1) \cdot (L \cdot \ell + 1)$. Hence,

$$\frac{|V_i|}{|V_{i'}|} \geqslant \frac{L \cdot \ell + 1 - d}{(1/n) \cdot (d+1) \cdot (L \cdot \ell + 1)}.$$

As $L \to \infty$, this approaches $n/(d+1)$, which establishes a contradition to α-balancedness for $\alpha < n/(d+1)$. □

We believe that this bound is tight up to a constant factor; that is, it should be possible to achieve $O(n/d)$-balancedness with d exclusions for any $d < n$. In particular, with a single exclusion, we believe it should be possible to achieve $O(n)$-balancedness. Below, we prove a weaker result: $O(n)$-proportionality can be achieved with a single exclusion. Note that this implies that the smallest part has size $\Omega(m/n^2)$. Since the largest part can have size at most m, this also implies $O(n^2)$-balancedness.

Theorem 12. *For any $n \geqslant 2$, every graph admits a connected 1-pseudo n-partition that is $O(n)$-proportional, and it can be computed in polynomial time.*

Algorithm 4: $2(n-1)$-proportionality with $R \leqslant 1$

Input: Tree $G = (V, E)$ and integer $n \geqslant 2$.
Output: A connected pseudo n-partition with $R \leqslant 1$.

```
r ← arbitrary node in V
T_0 ← tree (G, r) rooted at r
s ← 2(n − 1)
i ← 0
while there exists u with m/(n · s) ⩽ |ST(u, T_i)| ⩽ m/n do
    i ← i + 1
    V_i ← ST(u, T_{i−1})
    T_i ← T_{i−1} \ V_i
end while
u* ← r
if i < n − 1 then
    while true do
        if there exists u ∈ c(u*, T_i): ST(u, T_i) ⩾ m/(n · s) then
            u* ← u
        else
            break
        end if
    end while
    R ← {u*}
    for j = i + 1 to n − 1 do
        for u ∈ c(u*, T_i) do
            V_j ← V_j ∪ ST(u, T_i)
            if |V_j| ⩾ m/(n · s) then
                T_i ← T_i \ V_j
                break
            end if
        end for
    end for
    V_n = T_i \ {u*}
else
    V_n ← T_i
end if
return (V_1, ..., V_n, R)
```

Proof. Consider Algorithm 4. First, we argue that the algorithm is valid. We see that if Lines 6 to 8 are repeated i times, then i parts are created with $m/(n \cdot s) \leqslant |V_j| \leqslant m/n$ for $j \in [i]$. If $i = n-1$, then the last part consists of the remaining tree. Clearly, these parts are connected since they are subtrees of the main tree and also each of them has size at least $m/(n \cdot s)$. If $i < n-1$, then $|T_i| = |T \setminus \cup_{j \in [i]} V_i| \geqslant 2 \cdot m/n$. Next, we find node u^* that is as close to the root as possible such that $|ST(u^*, T_i)| > m/n$, while $|ST(u, T_i)| < m/(n \cdot s)$ for any $u \in c(u^*, T_i)$. Such a node always exists. Indeed, as $|ST(r, T_i)| > m/n$, then if some subtree that is rooted to one of r's child, say u, has size at least $m/(n \cdot s)$, then it should has size more than m/n, since otherwise the iteration in Lines 6 to 8 would not have stopped. Then, the algorithm considers the subtrees that rooted to children of u and if some of them has size at least $m/(n \cdot s)$, by following the same arguments as above the algorithm checks the sizes of subtrees that rooted to children of this child of u and so on. It is obvious that this procedure should stop before we reach a leaf of the tree.

Then, we construct each V_j with $j \in [i+1, n-1]$ by adding subtrees rooted to nodes in $ST(u^*, T_i)$ until the size of V_j becomes at least equal to $m/(n \cdot s)$. As each such subtree has size at most $m/(n \cdot s) - 1$ and before the last subtree added to V_j, V_j's size was at most $m/(n \cdot s) - 1$, we get that $|V_j| \leqslant 2 \cdot (m/(n \cdot s) - 1)$. Hence

$$\begin{aligned} |\cup_{j \in [i, n-1]} V_j| \leqslant |\cup_{j \in [1, n-1]} V_j| &< (n-1) \cdot \frac{2 \cdot m}{n \cdot 2 \cdot (n-1)} \\ &= m/n. \end{aligned}$$

Thus, as $ST(u^*, T_i) > m/n$, we can construct up to $n-1$ parts by using subtrees that are rooted to u^*. Clearly all these parts are connected through node u^*. Moreover, since $ST(r, T_i) \geqslant 2 \cdot m/n$, and $|\cup_{j \in [i, n-1]} V_j| \leqslant m/n$, then at least m/n nodes left after constructing the first $n-1$ parts which can consist the last part. Again, the last part is connected using u^*.

Regarding proportionality, we see that any part has size at least $m/(n \cdot s)$ and the statement follows. □

7 Complexity

In this section, we contemplate the complexity of checking whether an (approximately) balanced connected pseudo partition exists. To that end, we present two hardness results. The first one considers exact balancedness when $n-1$ exclusions are allowed.

Theorem 13. *Checking whether a balanced connected pseudo n-partition (with at most $n-1$ exclusions) exists is NP-complete.*

Proof. We use a reduction from 3-Partition problem which is defined as following. We are given $3n$ integers a_1,...,a_{3n} and a value A such that $\frac{A}{4} < a_i < \frac{A}{2}$ for any $i \in [3n]$ and $\sum_{i \in [3n]} a_i = n \cdot A$. A 3-Partition problems admits a solution if the numbers can be partitioned into n triples such that each triple adds up to A. The problem is *strongly* NP-complete which means that it is NP-complete even when all a_i-s and A are polynomially bounded.

Given an instance I of 3-Partition problem, we construct a graph $G_I = (V_I, E_I)$ as following. Let $L = (n-3) \cdot A$. We create $(n-1) \cdot n$ paths, denoted by P_i for $i \in [(n-1) \cdot n]$ such that, for $i \in [3n]$, $|P_i| = a_i + L$, while all the remaining paths have size equal to $A + L$. Moreover, we add $n-1$ nodes denoted by $s_1, ..., s_{n-1}$ such that s_j is connected with s_{j+1} for each $j \in [n-2]$. Next, we connect each s_j to one of the endpoints of paths $P_{n(j-1)+\ell}$, for $\ell \in [n]$. We show that G_I admits a balanced pseudo n-partition with $|R| \leqslant n-1$ if an only if I admits a solution.

If I admits a solution, then we can construct a balanced pseudo n-partition by setting $R = \{s_1, ..., s_{n-1}\}$, assigning to the same part every two paths P_j and $P_{j'}$ if and only if a_j and $a_{j'}$ are assigned to the same triple in the solution of I, and assigning $n-4$ paths of size $A + L$ to each V_i.

Now, assume that $(V_1, ..., V_n, R)$ is a balanced pseudo m-partition with $|R| \leqslant n-1$. First, notice that

$$\begin{aligned} &\sum_{i \in [n(n-1)]} |P_i| \\ &= \sum_{i \in [3n]} |P_i| + \sum_{i=3n+1}^{[n(n-1)]} |P_i| \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in[3n]} (a_i + L) + \sum_{i=3n+1}^{[n(n-1)]} (A+L) \\
&= nA + 3nL + ((n-4)\cdot n)(A+L) \\
&= n(A + 3L + (n-4)(A+L)).
\end{aligned}
$$

Let $B = A + 3L + (n-4)(A+L) = (n^2 - 3n)\cdot A$. Notice that for each P_i it holds that $A/4 + L < P_i \leqslant A + L$, as for each a_i we have that $\frac{A}{4} < a_i < \frac{A}{2}$, and since

$$A/4 + L = A/4 + (n-3)\cdot A > B/n = (n-3)\cdot A$$

and

$$
\begin{aligned}
A + L = A + (n-3)\cdot A = (n-2)\cdot A &< B/(n-2) \\
&= n(n-3)\cdot A/(n-2),
\end{aligned}
$$

we conclude that $B/n < |P_i| < B/(n-2)$ for each $i \in [n\cdot(n-1)]$. Notice that $|V_I| = n\cdot B + (n-1)$. We show that $|V_i| = B$ for each $i \in [n]$. Indeed, if $|V_i| \geqslant B+1$ for each $i \in [n]$, this means that we need at least $n\cdot B + n$ nodes, but $|V_I| = n\cdot B + n - 1 < n\cdot B + n$ which is impossible. On the other hand, if $|V_i| \leqslant B-1$, then $|\cup_{i\in[n]} V_i| \leqslant n(B-1)$ and then it should hold that $|R| > n-1$ which is also impossible. Now, note that if there are two parts V_i and $V_{i'}$ and $v, u \in P_j$ such that $u \in V_i$ and $v \in V_{i'}$, then either $|V_i| < B/(n-2)$ or $|V_{i'}| < B/(n-2)$ which is impossible. Hence, the nodes of each path P_j are assigned either to one V_i or R. Next, we show that $|R \cap (\cup_{j\in[n-1]} s_j)| > 0$. Let $Q_j = \cup_{\ell\in[n]} P_{n(j-1)+\ell}$. Notice that when some s_j does not belong in R, there are two possibilities. Either there is $|V_i| < B/(n-2)$ as V_i consists of at most one path of Q_j, which is not possible, or all the nodes in $Q_j \cup s_j$ are assigned to some V_i and R. As $|Q_j \cup s_j| \geqslant B+2$ and $|V_i| = B$, it should hold that $|Q_j \cap R| \geqslant 2$. Hence, if $|R \cap (\cup_{j\in n-1} s_j)| = 0$, then we have that for any $j \in [n-1]$, $|Q_j \cap R| \geqslant 2$, which means that $|R| \geqslant 2(n-1)$, but $|R| \leqslant n-1$. Now, we show that each s_j belongs to R. Suppose for contradiction that there exists s_j that is not included in R. With the same reasoning as above, we have that $|Q_j \cap R| \geqslant 2$ which means that there exists $s_{j'}$ that does not belong in R. But as $|Q_{j'} \cap R| \geqslant 2$, then there are two other $s_{j''}$ and $s_{j'''}$ that do not belong in R, but as $|Q_{j''} \cap R| \geqslant 2$ and $|Q_{j'''} \cap R| \geqslant 2$ then there are four other s_j-s that do not belong in R and so on. By continuing with this reasoning, we conclude that $|R \cap (\cup_{j\in n-1} s_j)| = 0$, which we show above that it is impossible. Hence, $R = \cup_{j\in[n-1]} s_j$. Now, we show that a part V_i can have size B if at most $n-4$ paths of size $A+L$ are assigned to it. Assume for contradiction that V_i contains $(n-3)$ such paths. Then,

$$
\begin{aligned}
|V_i| = B &= A + 3L + (n-4)\cdot(A+L) \\
&= 2L + (n-3)\cdot(A+L).
\end{aligned}
$$

Now, notice V_i cannot contain one path P_i of size less than $A+L$, as $|P_i| < A/2 + L < 2L$, cannot contain two such paths as P_i and $P_{i'}$ as $P_i + P_{i'} > A/2 + 2L > 2L$, and obviously cannot contain more than two such paths. If V_i contains $(n-2)$ paths of size $A+L$, then

$$
\begin{aligned}
|V_i| = B &= A + 3L + (n-4)\cdot(A+L) \\
&= L + (n-2)\cdot(A+L) - A.
\end{aligned}
$$

and then V_i cannot contain any other path as $|P_i| > L$ for any $i \in [n\cdot(n-1)]$. Similarly, we see that V_i cannot contain more that $n-2$ paths of size $A+L$. As there are $n(n-4)$ such paths and each V_i should contain at least $n-4$ of them, we conclude that each V_i contains exactly $n-4$ paths of size $A+L$. Now, it is clear that each V_i has size $B = A + 3L + (n-4)\cdot(A+L)$, if I admits a solution. □

The second result considers exact as well as approximate balancedness when no exclusions are allowed.

Theorem 14. *For any $\alpha < 2$, checking whether an α-balanced connected n-partition (with no exclusions) exists is NP-complete.*

Proof. We use a polynomial-time reduction from 2P2N-3SAT problem, the variant of 3SAT, in which each variable appears twice as positive and twice as negative literal. Let ϕ be an instance of 2P2N-3SAT which consists of r boolean variables, denoted by $x_1, ..., x_r$, and a 3-CNF formula with t clauses.

Let T be an integer. Given ϕ, we construct an instance with $n = 4r$ agents and a graph G_ϕ with $m = 12rT$ nodes, which has the following properties:

- If ϕ is satisfiable, then G has a balanced connected n-partition.
- If ϕ is not satisfiable, then any connected n-partition is at least 2-balanced.

In our reduction, we define a *core graph*, whose nodes are either *light* or *heavy*. The actual instance is then obtained by attaching a *path* of $T - 1$ nodes to each light core node and $2T - 1$ nodes to each heavy core node. The core graph has a *variable cycle* of eight light nodes for each variable, one heavy node for each clause (called the *clause node*), and $2r - t$ heavy nodes, which we call the *garbage collectors*. The light nodes in the variable cycle corresponding to variable x_i are denoted by $c_{i,1}$, $y_{i,1}$, $d_{i,1}$, $\overline{y}_{i,1}$, $c_{i,2}$, $y_{i,2}$, $d_{i,2}$, and $\overline{y}_{i,2}$; the cycle has edges connecting consecutive nodes in this ordering as well as the edge $(\overline{y}_{i,2}, c_{i,1})$. We refer to the nodes $y_{i,1}$, $y_{i,2}$, $\overline{y}_{i,1}$, and $\overline{y}_{i,1}$ as *literal nodes*. For each literal x_i (similarly for $\overline{x}_i$), we distinguish between the first and the second clauses in which this literal appears. The literal node $y_{i,1}$ (respectively, $y_{i,2}$) is connected to the clause node corresponding to the first (respectively, second) clause in which literal x_i appears. Similarly, node $\overline{y}_{i,1}$ (respectively, $\overline{y}_{i,2}$) is connected to the node corresponding to the first (respectively, second) clause in which literal $\overline{x}_i$ appears. All the literal nodes are also connected to all the garbage collectors.

We will first show that in any better than 2-balanced connected n-partition, an agent who gets a light or a heavy node should get the whole path attached to this node as well. First assume that some agent is allocated a light node but not the whole path attached to it. Then, some other agent should get the remaining at most $T - 1$ nodes in the path (and no other node, due to connectivity). On the other hand, since the total number of nodes is $12rT$ and there are $4r$ agents, some agent gets at least $3T$ nodes. Hence, this allocation cannot be even 3-balanced.

So, any agent who is allocated a light node should get its whole path as well. Now assume that there is some agent who gets some heavy node but not the whole path attached to it. Partition all agents into groups as follows. Consider an agent a who is allocated a light or heavy node together with its whole path attached. Agent a may also include some heavy nodes but not the whole path attached to them. We group this agent together with the agents who use part of paths attached to heavy nodes used by agent a. The crucial observation is that in the full paths allocated to agent a, the number of nodes cannot be $4T$; this would imply that the allocation is not better than 2-balanced. So, agent a has at most $3T$ nodes in full paths, $k \geqslant 0$ additional heavy nodes with part of the paths attached to them, which are also allocated to (at least) k other agents. So, the total number of nodes allocated to these (at least) $k + 1$ agents is at most $(3 + 2k)T$, which gives an average of $\frac{3+2k}{k+1}T$ per agent. This holds for all groups of agents, including at least one group with $k > 0$ (by our assumption above). Hence, the average number of nodes per agent is strictly smaller than $3T$, contradicting the fact that all the $12rT$ nodes are allocated to the $4r$ agents.

We have completed the proof that any better than 2-balanced connected n-partition should allocate a core node together with its attached path to the same agent. Then, a better than 2-balanced connected n-partition should actually be a balanced connected n-partition, allocating either three light nodes or one light and one heavy node per agent. This gives $2r$ agents who are allocated one heavy and one light node and $2r$ agents who get three light nodes each. There is additional structure such an allocation should have. As the light nodes in different variable cycles are not adjacent and are connected only through heavy node, exactly two agents get three light connected nodes each from a variable cycle. Notice that these allocations should leave either the two positive or (exclusive) the two negative literal nodes in each variable cycle unallocated, so that there are $2r$ unallocated nodes in variable cycles that are literal nodes which can be matched with the $2r$ clause nodes and garbage collectors. Hence, a balanced n-partition simulates a boolean assignment to the variables (depending on whether the light nodes in a variable cycle that are not bundled together with two other light nodes are the positive or the negative literal nodes).

Now, notice that each clause node has an incident unallocated literal node if and only if there is a satisfying assignment for ϕ. If such an assignment exists, then t of the literal nodes can be bundled together with the clause nodes and the remaining unallocated $2r - t$ literal nodes will be bundled together with the garbage collectors. Otherwise, if a satisfying assignment for ϕ does not exist, some clause node has no adjacent light node unallocated and no balanced n-partition exists. □

8 Discussion

Our work leaves open a number of directions for the future. For example, does there always exist a 2-balanced and $(2 - 1/n)$-proportional connected pseudo n-partition with at most $n - 1$ node exclusions? While we chart out a tight fairness-charity tradeoff when more than $n - 1$ exclusions are allowed, what happens when fewer exclusions are allowed? In particular, does there always exist an $O(n)$-balanced connected pseudo n-partition with at most a single exclusion? Do restricted families of graphs (especially those with higher connectivity) admit better fairness guarantees?

It would also be interesting to consider natural extensions and modifications of our model. What if, instead of excluding nodes, we are allowed to assign a few nodes to multiple parts? What if we allow nodes to have weights, and redefine proportionality and balancedness in terms of the total node weights of the different parts? In the appendix, we provide some guarantees in both these cases when $n = 2$. More broadly, it would be exciting to investigate the effectiveness of charity in the general fair division framework, where agents can have *heterogeneous* valuations for the nodes.

Acknowledgements

Shah was partially supported by an NSERC Discovery Grant.

References

[1] X. Bei, X. Lu, P. Manurangsi, and W. Suksompong. The price of fairness for indivisible goods. *Theory of Computing Systems*, pages 1–25, 2021.

[2] Xiaohui Bei, Ayumi Igarashi, Xinhang Lu, and Warut Suksompong. The price of connectivity in fair division. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5151–5158, 2021.

[3] B. Berger, A. Cohen, M. Feldman, and A. Fiat. (almost full) efx exists for four agents (and beyond). arXiv:2102.10654, 2021.

[4] Vittorio Bilò, Ioannis Caragiannis, Michele Flammini, Ayumi Igarashi, Gianpiero Monaco, Dominik Peters, Cosimo Vinci, and William S Zwicker. Almost envy-free allocations with connected bundles. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS)*, number 14, page 21, 2019.

[5] A. Biswas and S. Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018.

[6] Sylvain Bouveret, Katarína Cechlárová, Edith Elkind, Ayumi Igarashi, and Dominik Peters. Fair division of a graph. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 135–141, 2017.

[7] Sylvain Bouveret, Katarína Cechlárová, and Julien Lesca. Chore division on a graph. *Autonomous Agents and Multi-Agent Systems*, 33(5):540–563, 2019.

[8] E. Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[9] I. Caragiannis, N. Gravin, and X. Huang. Envy-freeness up to any item with high nash welfare: The virtue of donating items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 527–545, 2019.

[10] I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[11] Frédéric Chataigner, Liliane RB Salgado, and Yoshiko Wakabayashi. Approximation and inapproximability results on balanced connected partitions of graphs. *Discrete Mathematics and Theoretical Computer Science*, 9 (1):177–192, 2007.

[12] B. R. Chaudhury, J. Garg, K. Mehlhorn, R. Mehta, and P. Misra. Improving efx guarantees through rainbow cycle number. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, 2021. To appear.

[13] B. R. Chaudhury, T. Kavitha, K. Mehlhorn, and A. Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM Journal on Computing*, 50(4):1336–1358, 2021.

[14] Guangting Chen, Yong Chen, Zhi-Zhong Chen, Guohui Lin, Tian Liu, and An Zhang. Approximation algorithms for the maximally balanced connected graph tripartition problem. *Journal of Combinatorial Optimization*, pages 1–21, 2020.

[15] Janka Chlebíková. Approximating the maximally balanced connected partition problem in graphs. *Information Processing Letters*, 60(5):225–230, 1996.

[16] An-Chiang Chu, Bang Ye Wu, Hung-Lung Wang, and Kun-Mao Chao. A tight bound on the min-ratio edge-partitioning problem of a tree. *Discrete Applied Mathematics*, 158(14):1471–1478, 2010.

[17] An-Chiang Chu, Bang Ye Wu, and Kun-Mao Chao. A linear-time algorithm for finding an edge-partition with max-min ratio at most two. *Discrete Applied Mathematics*, 161(7-8):932–943, 2013.

[18] V. Conitzer, R. Freeman, and N. Shah. Fair public decision making. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 629–646, 2017.

[19] Sina Dehghani, Alireza Farhadi, MohammadTaghi HajiAghayi, and Hadi Yami. Envy-free chore division for an arbitrary number of agents. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2564–2583, 2018.

[20] A. Deligkas, E. Eiben, R. Ganian, T. Hamm, and S. Ordyniak. The parameterized complexity of connected fair division. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 139 – 145., 2021.

[21] Raghuveer Devulapalli. *Geometric partitioning algorithms for fair division of geographic resources*. PhD thesis, University of Minnesota, 2014.

[22] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *American Mathematical Monthly*, 68(1):1–17, 1961.

[23] Shane Dye. A note on the minimum bounded edge-partition of a tree. *Discrete Applied Mathematics*, 157(13): 2958–2963, 2009.

[24] Martin E Dyer and Alan M Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 10(2):139–153, 1985.

[25] G. Gamow and M. Stern. *Puzzle-Math*. Viking, 1958.

[26] G. Greco and F. Scarcello. The complexity of computing maximin share allocations on graphs. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2006 – 2013, 2020.

[27] D. Halpern and N. Shah. Fair and efficient resource allocation with partial information. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 224–230, 2021.

[28] Sandy Heydrich and Rob van Stee. Dividing connected chores fairly. *Theoretical Computer Science*, 593:51–61, 2015.

[29] A. Igarashi and D. Peters. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 2045 – 2052, 2019.

[30] Alain Leroux and Justin Leroux. Fair division with no information. *Economic Theory*, 24(2):351–371, 2004.

[31] B. Li, L. Li, A. Sun, C. Wang, and Y. Wang. Approximate group fairness for clustering. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 6381–6391, 2021.

[32] E. Micha and N. Shah. Proportionally fair clustering revisited. In *Proceedings of the 47th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 85:1–85:16, 2020.

[33] Saleh Soltan, Mihalis Yannakakis, and Gil Zussman. Doubly balanced connected graph partitioning. *ACM Transactions on Algorithms (TALG)*, 16(2):1–24, 2020.

[34] H. Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[35] W. Stromquist. How to cut a cake fairly. *American Mathematical Monthly*, 87(8):640–644, 1980.

[36] F. E. Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.

[37] Justin C Williams Jr. Political redistricting: a review. *Papers in Regional Science*, 74(1):13–40, 1995.

[38] Bang Ye Wu, Hung-Lung Wang, Shih Ta Kuan, and Kun-Mao Chao. On the uniform edge-partition of a tree. *Discrete Applied Mathematics*, 155(10):1213–1223, 2007.

[39] Jan-Jan Wu, Pangfeng Liu, and Yi-Chien Chung. Metadata partitioning for large-scale distributed storage systems. In *Proceedings of the 3rd IEEE International Conference on Cloud Computing*, pages 212–219, 2010.

Appendix

A Node-Weighted Graphs

In this section, we focus on the case that it is given a vertex-weighted connected graph $G = (V, E, w)$ where $w : V \to \mathbb{R}_{\geqslant 0}$. For every $V' \subseteq V$, $w(V') = \sum_{v \in V'} w(v)$. With a slight abuse of notation, given a vertex-weighted connected graph $G = (V, E, w)$, we denote with $w(G)$ the total weight of the graph, i.e. let $w(G) = \sum_{v \in V} w(v)$. In this case, balancedness is defined using the weights of the different parts. However, if the graph consists of $m - 1$ nodes of negligible weight and one node that has a huge weight, then we can cannot guarantee any finite approximation of balancedness. Hence, here we introduce an approximation notion which is aligned with the the notion envy-freeness up to ℓ items in the classic fair division literature.

Definition 3. A connected pseudo n-partition $(V_1, ..., V_n, R)$ is called envy-free up to ℓ nodes (EFℓ) for $\ell \geqslant 0$, if for every $i, j \in [n]$, there exists $S \subseteq V_j$ with $|S| \leqslant \ell$ such that

$$w(V_i) \geqslant w(V_j) - w(S).$$

Next, we show that when $n = 2$ is cases that we cannot achieve 2-balancedness, envy-freeness up to one ℓ nodes is possible and vice versa.

Theorem 15. *If $n = 2$ then for any vertex-weighted connected graph $G = (V, E, w)$ there exists a connected pseudo 2-partition (V_1, V_2, R), with $R \leqslant 1$ which is either 2-balanced or EF1.*

Proof. Let G' be a spanning tree of G and let $T = (G', r)$ be a tree rooted to an arbitrary node r. Moreover, we denote with W the total weight of the graph, i.e. $W = w(G)$ Starting from the highest level (the leaves) of T, we find the first node, say u^*, such that $w(ST(u^*)) \geqslant W/3$. Notice that every subtree that is rooted in a child of u^* has weight less than $W/3$, as otherwise the procedure would stop to a child of u^*. We distinguish into two cases.

Case I: $w(ST(u^*)) - w(u^*) \geqslant W/3$. If $w(T \setminus ST(u^*)) \geqslant W/3$, then we set $V_1 = ST(u^*)$ and $V_2 = T \setminus ST(u^*)$. Then, since $|V_1| \leqslant W/3$ and $|V_2| \leqslant W/3$, the theorem follows. Now, we focus on the case that $w(T \setminus ST(u^*)) < W/3$. Let $T' = (G', u^*)$. In this case, notice that all the subtrees that rooted to a node that is located at the second level of T' has weight less than $W/3$. We partition all the subtrees in $Q = \{ST(u', T') : u' \in c(u^*, T')\}$ into two sets S_1 and S_2 such that $|w(S_1) - w(S_2)|$ is minimized[5]. Without loss of generality, assume that $w(S_1) \geqslant w(S_2)$. If $w(S_1) \leqslant 2 \cdot w(S_2)$, then we set $V_1 = S_1$, $V_2 = S_2$ and $R = \{u^*\}$, and (V_1, V_2, R) is clearly 2-balanced. Next, we assume that $w(S_1) > 2 \cdot w(S_2)$. If S_1 consists of at least two subtrees, we denote with S_{min} the subtree with the smallest size in S_1. Then, if we set $S_1' = S_1 \setminus S_{min}$ and $S_2' = S_2 \cup S_{min}$, we have that $w(S_1') \geqslant w(S_1)/2 > w(S_2)$, while $w(S_2') > w(S_2)$ which means that $|w(S_1') - w(S' - 2)| < |w(S_1') - w(S' - 2)|$ which is a contradiction. On the other hand, if S_1 consists of only one subtree rooted to a child of u^*, then we set $V_1 = S_1$ and $V_2 = T \setminus S_1 = S_2 \cup \{u^*\}$. Then, as $w(S_1) \geqslant w(S_2)$, we have that $w(V_1) \geqslant w(V_2) - w(u^*)$. Moreover as any subtree in Q has size at most $W/3$, we get that $w(S_1) \leqslant W/3$ and hence $w(V_1) \leqslant w(V_2)$. Thus, (V_1, V_2, R) is EF1.

Case II: $w(ST(u^*)) - w(u^*) < W/3$. If $w(ST(u^*)) \leqslant 2 \cdot W/3$, then we set $V_1 = ST(u^*)$ and $V_2 = T \setminus ST(u^*)$, and then (V_1, V_2, R) is clearly 2-balanced. Otherwise, we have that $w(u^*) > W/3$, while $w(T') - w(ST(u^*, T')) < W/3$. Let $\mathcal{S} = \{T' \setminus ST(u^*, T'), ST(u^*, T') \setminus \{u^*\}\}$. If we set

$$V_1 = \{u^*\} \cup \arg\min_{S \in \mathcal{S}} w(S)$$

and

$$V_2 = \arg\max_{S \in \mathcal{S}} w(S),$$

then since $w(V_2) \leqslant W/3$ and $w(V_2) \geqslant w(V_1) - w(u^*)$, we conclude that (V_1, V_2, R) is EF1. □

[5]This can be done in polynomial time using classic techniques from dynamic programming

B Node Sharing

In this section, we consider a modified model where instead of excluding nodes, we allow a few nodes to be assigned to multiple parts. More formally, we say that $(V_1, \ldots, V_n)$ is a *shared pseudo n-partition* of G if

1. $V = (\cup_{i \in [n]} V_i)$; and
2. $\cap_{i \in n} V_i \leqslant n - 1$.

A shared pseudo n-partition $(V_1, \ldots, V_n)$ is called *connected* if each $G[V_i]$ is connected. The definition of balancedness remains the same. Next, we show that a slightly modified version of Algorithm 1 returns a connected shared pseudo 2-partition that is 2-balanced, but with a larger guaranteed size for each part.

Theorem 16. *When $n = 2$, a modified version Algorithm 1 of returns a connected shared pseudo 2-partition (V_1, V_2) that is 2-balanced and $\min(|V_1|, |V_2|) \geqslant \lceil n/3 \rceil$.*

Proof. Consider Algorithm 1 but in Line 7 instead of assign u^* to R, u^* is assigned to V_1 and V_2 as well.

The first case where $|ST(u^*, T)| = \lceil m/3 \rceil$, the algorithm does the same operation and in a similar way as in the proof of Theorem 2, we get that $\min(|V_1|, |V_2|) \geqslant \lceil n/3 \rceil$.

Next, consider the case where $|ST(u^*)| > \lceil m/3 \rceil$. With similar arguments as in the proof of Theorem 2, we conclude that $\lceil m/3 \rceil \leqslant |V_1| \leqslant 2(\lceil m/3 \rceil - 1$. Thus, we have that $|V \setminus V_1| \geqslant m - 1 - 2(\lceil m/3 \rceil - 1) \geqslant \lceil (m-1)/3 \rceil$. But as $V_2 = V \setminus V_1 \cup \{u^*\}$, the theorem follows. $\square$