

---

# On Hierarchies of Fairness Notions in Cake Cutting: From Proportionality to Super Envy-Freeness

---

**Arnav Mehra**  
Capital One  
mehraarn000@gmail.com

**Alexandros Psomas**  
Purdue University  
apsomas@purdue.edu

## Abstract

We consider the classic cake-cutting problem of producing fair allocations for  $n$  agents, in the Robertson–Webb query model. In this model, it is known that: (i) proportional allocations can be computed using  $O(n \log n)$  queries, and this is optimal for deterministic protocols; (ii) envy-free allocations (a subset of proportional allocations) can be computed using  $O\left(n^{n^{n^n}}\right)$  queries, and the best known lower bound is  $\Omega(n^2)$ ; (iii) perfect allocations (a subset of envy-free allocations) cannot be computed using a bounded (in  $n$ ) number of queries.

In this work, we introduce two hierarchies of new fairness notions: Harmonically Coalition-Resistant (HCR) and Linearly Coalition-Resistant (LCR). An allocation is HCR- $k$  if the allocation is complete and, for any subset of agents  $S$  of size at most  $k$ , every agent  $i \in S$  believes the value of all pieces allocated to agents in  $S$  to be at least  $\frac{1}{n-|S|+1}$ , making the union of all pieces allocated to agents not in  $S$  at most  $\frac{n-|S|}{n-|S|+1}$ ; for LCR- $k$  allocations, these bounds become  $\frac{|S|}{n}$  and  $\frac{n-|S|}{n}$ , respectively. Intuitively, these notions of fairness ask that, for every agent  $i$ , the collective value (from the perspective of agent  $i$ ) that a group of agents receives is limited. If the group includes  $i$ , its value is lower-bounded, and if the group excludes  $i$ , it is upper-bounded, thus providing the agent some protection against the formation of coalitions.

Our hierarchies bridge the gap between proportionality, envy-freeness, and super envy-freeness. HCR- $k$  and LCR- $k$  coincide with proportionality for  $k = 1$ . For all  $k \leq n$ , HCR- $k$  allocations are a superset of envy-free allocations (i.e., easier to find). On the other hand, for  $k \in [2, \lceil n/2 \rceil - 1]$ , LCR- $k$  allocations are incomparable to envy-free allocations. For  $k \geq \lceil n/2 \rceil$ , LCR- $k$  allocations are a subset of envy-free allocations (i.e., harder to find), while LCR- $n$  coincides with super envy-freeness: the value of each agent for their piece is at least  $1/n$ , and their value for the piece allocated to any other agent is at most  $1/n$ .

We prove that HCR- $n$  allocations can be computed using  $O(n^4)$  queries in the Robertson–Webb model. On the flip side, finding HCR-2 (and therefore all HCR- $k$  for  $k \geq 2$ ) allocations requires  $\Omega(n^2)$  queries, while LCR-2 (and therefore all LCR- $k$  for  $k \geq 2$ ) allocations cannot be computed using a bounded (in  $n$ ) number of queries. Our results reveal that envy-free allocations occupy a curious middle ground, between a computationally impossible notion of fairness, LCR- $\lceil n/2 \rceil$ , and a computationally “easy” notion, HCR- $n$ .

## 1 Introduction

We consider the classic problem of cake cutting, proposed by Hugo Steinhaus [Ste48], while in hiding during World War II [Wik25]. In this problem, there is a heterogeneous resource, the “cake,”

typically represented by the interval  $[0, 1]$ . There is a set  $N$  of  $n$  agents with different valuations functions  $V_1, \dots, V_n$  over subsets of the cake; these valuations functions are induced by probability measures over  $[0, 1]$  (and hence, for example, the value of every agent for the entire cake is equal to 1). The goal in this problem is to allocate to each agent a piece of the cake — a finite union of disjoint intervals — in a fair manner, under various notions of fairness. This simple model has served as a cornerstone for fair division, shaping many of the field’s foundational questions.

A major focus has been the complexity of computing fair allocations. The complexity of discrete cake-cutting protocols is measured by the number of queries they require in the query model suggested by Robertson and Webb [RW98] and later formalized (and named) by Woeginger and Sgall [WS07]. For example, it is well understood that *proportional allocations*, those where every agent  $i \in N$  values their piece at least  $1/n$ , can be computed using  $O(n \log n)$  queries [EP84]; and, this is tight: every deterministic protocol requires  $\Omega(n \log n)$  queries [EP11]. Another major notion of fairness is *equitability*, which requires that every agent has the same value for the piece allocated to them, i.e., for all agents  $i, j \in N$ ,  $V_i(A_i) = V_j(A_j)$  (where  $A_\ell$  is the piece allocated to agent  $\ell$ ). Procaccia and Wang [PW17] proved that equitable allocations cannot be computed in a bounded (in  $n$ ) number of queries. This result, in turn, rules out bounded protocols for another major notion of fairness, *perfection*: an allocation  $A$  is perfect if for all agents  $i, j \in N$ ,  $V_i(A_j) = 1/n$ .<sup>1</sup>

Arguably the most important notion of fairness — the holy grail of fair division — is *envy-freeness*; an allocation  $A$  is envy-free if every agent prefers their piece to the piece allocated to any other agent, i.e., for all agents  $i, j \in N$ ,  $V_i(A_i) \geq V_i(A_j)$ . It is easy to see that every perfect allocation is envy-free, and every envy-free and complete ( $\cup_i A_i = [0, 1]$ ) allocation is proportional. Unlike proportionality, equitability, and perfection, the complexity of computing (complete) envy-free allocations has been rather elusive. The existence of *bounded* protocols remained open for two decades until Aziz and

Mackenzie [AM16a] presented a protocol that requires at most  $O(n^{n^{n^{n^n}}})$  queries. The best known lower bound is  $\Omega(n^2)$  [Pro09], leaving an astronomical gap in our understanding.

In this paper, we zoom in on the landscape of fairness between proportionality and perfection, focusing on the query complexity of different notions in this less understood intermediate space.

## 1.1 Our contributions

We introduce two new hierarchies of fairness: Harmonically Coalition-Resistant (HCR) and Linearly Coalition-Resistant (LCR). An allocation  $A$  satisfies HCR- $k$  if, for every subset of agents  $S$  such that  $|S| \leq k$ , and for every agent  $i \in S$ , the value of  $i$  for the union of pieces allocated to agents not in  $S$  is at most  $\frac{n-|S|}{n-|S|+1}$ , i.e.,  $\sum_{j \notin S} V_i(A_j) = V_i(A_{N \setminus S}) \leq \frac{n-|S|}{n-|S|+1}$ . Equivalently,  $V_i(A_S) \geq \frac{1}{n-|S|+1}$ . For LCR- $k$ , the upper bound is  $\frac{n-|S|}{n}$ , and the lower bound is  $\frac{|S|}{n}$ . The existence of HCR- $k$  allocations is implied by the existence of envy-free allocations. Let  $A$  be an envy-free allocation; for all agents  $i, j \in N$ ,  $V_i(A_j) \leq V_i(A_i)$ . Adding up for all  $j \notin S$  gives  $\sum_{j \notin S} V_i(A_j) \leq (n - |S|)V_i(A_i) \leq (n - |S|)(1 - \sum_{j \in S} V_i(A_j))$ ; re-arranging gives the desired inequality. The existence of LCR- $k$  allocations is similarly implied by the existence of perfect allocations.

Our new fairness notions are interesting precisely because they occupy a natural but largely unexplored middle ground between proportionality and envy-freeness. HCR- $k$  allocations limit how much value groups of agents collectively derive from other agents’ shares; thus, they may be particularly suited for contexts where collective perceptions of fairness are important. LCR- $k$  allocations impose stronger conditions, which makes them appealing when perfection is desired (where the computational cost is prohibitive/unbounded). By providing a spectrum of fairness guarantees, these notions offer a more nuanced toolkit, enabling practitioners to better navigate the trade-off between computational complexity and fairness requirements.

In Appendix A we start by showing how our new notions relate to the existing notions of fairness; we focus on complete allocations (as, otherwise, envy-freeness is the easiest notion, since it is satisfied by an empty allocation). We show that the entire HCR- $k$  hierarchy lies between proportionality and envy-freeness, with HCR-1 being proportionality, but HCR- $n$  being a strict superset of envy-freeness. On the other hand, the entire LCR- $k$  hierarchy lies between proportionality and perfection, with LCR-1 also being proportionality, while LCR- $n$  is equivalent to *super envy-freeness*. An allocation

<sup>1</sup>Which, perhaps surprisingly, always exist [Lia40, Alo87]!

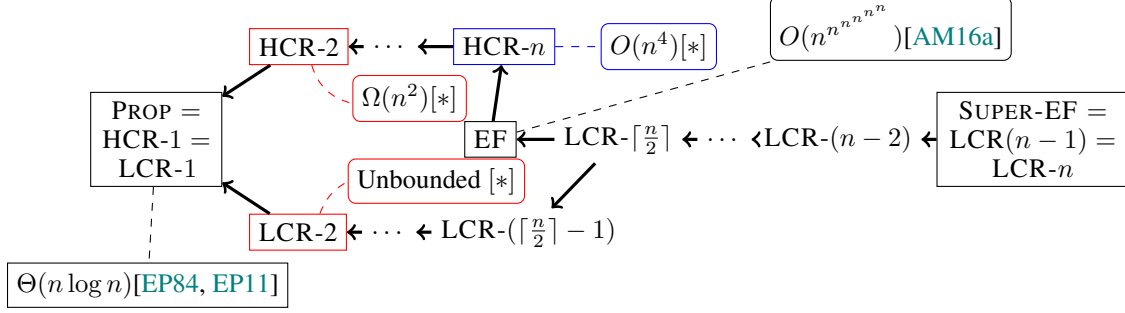


Figure 1: Relation between our new notions of fairness, Proportionality (PROP), Envy-Freeness (EF), and Super Envy-Freeness (SUPER-EF). A solid arrow from  $Y$  to  $X$  represents that notion  $Y$  implies notion  $X$ , i.e.  $X \leftarrow Y$  means  $X \supseteq Y$ .  $X \leftarrow Y$  implies that finding an allocation with property  $Y$  is a harder task. Our results, highlighted in colored rectangles, are marked with  $[*]$ .

$A$  is super envy-free if, for all agents  $i, j \in N$ ,  $V_i(A_i) \geq 1/n \geq V_i(A_j)$ . Regarding LCR- $k$  and envy-freeness, for  $2 \leq k \leq \lceil \frac{n}{2} \rceil - 1$ , LCR- $k$  is incomparable to envy-freeness; for  $k \geq \lceil \frac{n}{2} \rceil$ , LCR- $k$  implies envy-freeness. See Figure 1.

We proceed to study the complexity of computing our new notions in the Robertson-Webb model. In Section 3 we prove our main upper bound, Theorem 1: there exists a protocol that computes complete, HCR- $n$  allocations, using at most  $O(n^4)$  queries in the Robertson-Webb model. The first key observation is that, a complete, proportional and “near-perfect” allocation  $A = (A_1, \dots, A_n)$  satisfies all the constraints required by HCR- $n$ ; concretely, completeness, along with  $V_i(A_i) \geq \frac{1}{n}$  and  $V_i(A_j) \geq \frac{1}{2n}$  for all  $j$ , implies HCR- $n$ . Therefore, it is natural to start with an  $\epsilon$ -perfect allocation — which can be computed using only  $O(n^3/\epsilon)$  queries [BM15] — and try to adjust it to satisfy the additional proportionality property. An obstacle in this approach is that, for any  $\epsilon > 0$ , it might be the case that the  $\epsilon$ -perfect allocation is such that  $V_i(A_j) > \frac{1}{n}$ , for all  $j \neq i$ , making it impossible to satisfy proportionality for  $i$ , unless we trim other agents’ pieces. This obstacle can be bypassed with the following trick: introduce  $\ell$  phantom agents, with arbitrary valuation functions, and find an  $\epsilon$ -perfect partition for the instance with  $n + \ell$  agents. Then, the “real” agents get almost equal allocations, no one gets a piece whose value is larger than  $\frac{1}{n}$ , and the allocations of the phantom agents can be combined into a new piece, the residue, to be allocated among the real agents, using, e.g., a clever recursion. However, intuitively, such a step forces us to find an allocation of the residue that is weighted proportional, where agents have different weights (or, equivalently, find an allocation such that agents’ utilities meet specific utility lower bounds). Unfortunately, this leads us to a dead-end: there is no bounded protocol for finding a proportional division with unequal shares [CF20]. The final piece of this puzzle is that pieces returned by the  $\epsilon$ -perfect call do not need to be matched to specific “real” agents, but should be available for anyone to claim.

In more detail, our algorithm, Algorithm 1, starts by finding an  $\epsilon$ -perfect allocation  $B$  for  $4n/3$  agents, where the  $n/3$  extra agents, agents  $n+1, \dots, 4n/3$ , have arbitrary valuation functions. Let  $B_1, \dots, B_n$  be the first  $n$  pieces of this allocation; pieces  $B_{n+1}, \dots, B_{4n/3}$  are combined into the residue  $R$ . Importantly, “real” agents are *not* allocated any of the  $B_j$  pieces (yet). Algorithm 1 then proceeds à la Dubins-Spanier [DS61] (or Last Diminisher [Ste48]). Let  $S$  be the subset of the  $B_j$  pieces still available. The algorithm asks every agent  $i$  to make a mark on  $R$ , such that their value to the left of the mark is equal to  $1/n - \max_{B \in S} V_i(B)$ . The agent  $i^*$  with the left-most mark (breaking ties arbitrarily) gets the two corresponding pieces, one from cutting the residue and their favorite piece from  $S$ , and exits the process. The process is repeated until a single agent  $j$  is left, who is allocated the entire remaining residue, and the unique piece left in  $S$ . We prove that Algorithm 1 computes a HCR- $n$  allocation using  $O(n^4)$  queries. Importantly, our query complexity bound relies on the fact that the algorithm of Brânzei and Miltersen [BM15] for finding  $\epsilon$ -perfect allocation produces at most  $O(n^2)$  intervals.

In Appendix B we prove our main lower bounds. First, we prove that computing a HCR-2 allocation requires  $\Omega(n^2)$  queries (Theorem 6), while computing a LCR-2 allocation is impossible using a bounded protocol (Theorem 7). To prove our first lower bound, Theorem 6, we first formalize the

amount of knowledge an algorithm can have about an agent  $i$  after  $t$  queries. We use the notion of *active intervals*. Intuitively, if an interval  $I$  is active for agent  $i$  at step  $t$ , denoted by  $I \in \Pi_i^t$ , then  $V_i(I)$  is known to the algorithm, but for all  $I' \subsetneq I$ , an adversary can pick  $V_i(I')$  to be anything from 0 to  $V_i(I)$ . That is, all these choices for  $V_i(I')$  are consistent with the query responses up until time  $t$ . This definition was also at the core of the  $\Omega(n^2)$  lower bound for envy-freeness, by Procaccia [Pro09]. Here, we argue that, if the adversary responds to queries as if the valuation functions are uniform for the entirety of an algorithm’s execution, the allocation  $A = (A_1, \dots, A_n)$  that the algorithm outputs must be such that  $A_i \in \Pi_i^T$  and  $V_i(A_i) = \frac{1}{n}$ , for all  $i \in N$ , where  $T$  is the number of queries the algorithm made before terminating. For LCR-2 to be satisfied,  $V_i(A_i) = \frac{1}{n}$  implies a condition on  $V_i(A_j)$ , that similarly needs to be “checked” (i.e.,  $I \in \Pi_i^T$ , for some  $I \subseteq A_j$ ) for all pairs of agents  $i, j \in N$ ; the  $\Omega(n^2)$  lower bound follows. Our second lower bound, Theorem 7, reduces the problem of finding an LCR-2 allocation to a known problem with unbounded query complexity: the problem of finding an exact division. In the exact division problem, we are given target values  $w_1, \dots, w_z$ , and are asked to split the cake into  $z$  pieces, so that every agent  $i$  has value  $w_\ell$  for piece  $\ell$ , for  $\ell = 1, \dots, z$ . Exact division is known to be impossible to solve with a bounded protocol, even for the case of two valuation functions and  $z = 2$  pieces with equal weights [RW98]. Theorem 7 immediately implies that the problem of finding a super-envy free allocation has unbounded query complexity in the Robertson-Webb model, which, to the best of our knowledge, was an open problem.

Finally, in Section 4, in light of the strong lower bounds for LCR-2, we consider approximations. The minimally weaker notion that we could hope to achieve is the following: (i) if  $|S| = 1$ , then  $V_i(A_S) \leq \frac{n-1}{n}$ , for all  $i \in S$  (i.e., the allocation is proportional), (2) if  $|S| = 2$ , then  $V_i(A_S) \leq \frac{n-2}{n}(1 + \delta)$ , for all  $i \in S$ . We show that this notion is possible to achieve with a bounded protocol. More generally, we define  $\delta$ -LCR- $k$  to be the set of complete and proportional allocations, such that if  $k \geq |S| \geq 2$ , then  $V_i(A_S) \leq \frac{n-|S|}{n}(1 + \delta)$ ; we prove that the query complexity of computing a  $\delta$ -LCR- $n$  allocation is  $O\left(\frac{n^6}{\epsilon} \frac{\ln(n/\epsilon)}{\ln(n)}\right)$ . To prove this result, we give an algorithm for finding complete, proportional, and  $\epsilon$ -perfect allocations using  $O\left(\frac{n^5}{\epsilon} \frac{\ln(1/\epsilon)}{\ln(n)}\right)$  queries, which might be of independent interest. The high-level blueprint of this algorithm, Algorithm 2, is similar to Algorithm 1: in phase one split the cake into approximately equal pieces and residue, and in phase two run a “cut-and-match Last Diminisher.” Algorithm 2 requires much stricter conditions from the pieces and residue at the end of phase one. As opposed to Algorithm 1, phase one in Algorithm 2 is implemented via a recursion: at each iteration the residue is split into an  $(n + 1)$ -piece  $\epsilon'$ -perfect allocation. Out of these  $n + 1$  pieces, one of them serves as the residue in the next iteration. The first  $n$  pieces are carefully matched to previously computed pieces. This process is done  $d$  times in total, where  $d$  and  $\epsilon'$  need to be carefully chosen so that the value of every agent for every “combined” piece (across iterations) is within the required bounds.

Our results provide significant insight into the spectrum of fairness notions between proportionality and super envy-freeness; see Figure 1. We observe that the two ways of strengthening proportionality, HCR-2 and LCR-2, lead to strikingly different lower bounds on the corresponding query complexity. Envy-freeness occupies a curious middle ground. The minimally weaker fairness notion, HCR- $n$ , can be solved in  $O(n^4)$  queries, while the minimally stronger fairness notion, LCR- $\lceil \frac{n}{2} \rceil$ , has unbounded query complexity. A concrete take-home message of our work regarding the query complexity of finding envy-free allocations is that, if a super polynomial lower bound exists, then there must exist a subproblem strictly harder than finding HCR- $n$  allocations which requires a super polynomial number of queries to solve.

## 1.2 Related work

As already discussed, proportional allocations can be computed using  $O(n \log n)$  queries [EP84], and this is tight for deterministic protocols [EP11]. Randomized algorithms can bypass this lower bound [EP06]. Surprisingly, finding a (weighted) proportional allocation when agents have unequal shares requires an unbounded number of queries [CF20].

Regarding envy-freeness, the cut-and-choose method gives an envy-free allocation for  $n = 2$  agents, and the Selfridge–Conway procedure gives an envy-free allocation for  $n = 3$  agents. The problem of finding an envy-free allocation for four agents, using a bounded number of queries, was resolved by [AM16b], and later improved by [ACF<sup>+</sup>18]. For  $n$  agents, the best known protocol requires

$O(n^{n^{n^{n^n}}})$  queries, while the currently best known lower bound is  $\Omega(n^2)$  [Pro09]. Equitable allocations, and therefore perfect allocations, cannot be found in a bounded number of queries [PW17].

Approximate solutions are typically easier to find. Brânzei and Miltersen [BM15] prove that finding an  $\epsilon$ -perfect allocation requires at most  $O(n^3/\epsilon)$  queries, and Brânzei and Nisan [BN22] prove that finding an  $\epsilon$ -envy free and connected allocation requires at most  $O(n/\epsilon)$  queries (and at least  $\Omega(\log(1/\epsilon))$  queries). Cechlárová and Pillárová [CP12] prove that finding an  $\epsilon$ -equitable and proportional allocation requires at most  $O(n(\log n + \log(1/\epsilon)))$  queries.

Segal-Halevi and Suksompong [SHS20, SHS23] and Segal-Halevi and Nitzan [SHN19] study cake cutting among groups. Specifically, Segal-Halevi and Suksompong [SHS20, SHS23] study contiguous and envy-free cake cutting among groups (where agents within a group also get a contiguous piece). Segal-Halevi and Nitzan [SHN19] study a problem where a cake must be divided and allocated to (pre-determined) groups of agents, and study the existence and query complexity of various fairness notions in this model. For example, in their unanimous FS allocations the  $n$  agents are placed in  $k$  groups,  $K$ , of size  $n/k$ , and for any group  $S \in K$  and any agent  $i \in S$  it must hold that  $V_i(A_S) \geq \frac{k}{n}$ . This condition is quite similar to LCR- $k$ ; however, it is considerably less restrictive, as it places no constraints on subsets of size less than  $k$ , or different groupings of the agents (LCR- $k$  considers *all* possible groupings of size at most  $k$ ). Segal-Halevi and Nitzan [SHN19] prove that unanimous-FS allocations require infinite queries; yet little is known about how close one can get with a bounded number of queries. Our algorithms provide allocations as close to LCR- $k$  as possible, with a bounded number of queries.

Berliant et al. [BTD92], and later Husseinov [Hus11], study a notion of group fairness, *group envy-freeness*, where, similar to our work, the subsets are not predefined. However, their definition only considers groups of equal size: they call an allocation  $A = (A_1, \dots, A_n)$  group envy-free, if for every pair of groups of agents  $C_1$  and  $C_2$ , with  $|C_1| = |C_2|$ , there is no partition  $\{B_i\}_{i \in C_1}$  of  $\cup_{j \in C_2} A_j$ , such that, for all  $i \in C_1$ ,  $i$  prefers  $B_i$  to  $A_i$ , with strict preference for at least one  $i \in C_1$ . This notion is stronger (“harder to achieve”) than perfection; the focus of the aforementioned works is existence of such allocations, and compatibility with economic efficiency.

Further afield, numerous papers study fair allocation of indivisible items among groups of agents [CFSV19, FT22, CLS25, MS22, STZ23, KSV20, AR21].

## 2 Preliminaries

We have an infinitely divisible resource, the “cake,” denoted by the interval  $[0, 1]$ . A piece of cake refers to a finite set of disjoint intervals of  $[0, 1]$ . Our goal is to allocate the cake among a set  $N$  of  $n$  agents. An allocation  $A = (A_1, \dots, A_n)$  of the cake to the agents consists of  $n$  pieces, where  $A_i$  is the piece allocated to agent  $i$ , such that  $A_i \cap A_j = \emptyset$  for all pairs of agents  $i, j$ . An allocation  $A$  is *complete* if  $\cup_{i=1}^n A_i = [0, 1]$ . For an allocation  $A$  and a subset of agents  $S \subseteq N$ , we will use notation  $A_S = \cup_{i \in S} A_i$  to denote the union of all pieces allocated to the agents in  $S$ . We also use notation  $\bar{S} = N \setminus S$  for the complement of a set  $S$ .

Every agent  $i \in N$  has a valuation function  $V_i$  that assigns a non-negative value to any subinterval of  $[0, 1]$ . It is convenient to think of these values as being induced by a density function  $v_i$ . That is, for an interval  $I = [a, b]$ ,  $V_i(I) = \int_{x=a}^b v_i(x) dx$ . Valuations are (i) normalized:  $V_i([0, 1]) = \int_{x=0}^1 v_i(x) dx = 1$ , (ii) additive: for a set of disjoint intervals  $I_1, \dots, I_m$ ,  $V_i(\cup_{j=1}^m I_j) = \sum_{j=1}^m V_i(I_j)$ , (iii) non-atomic:  $\forall x, y \in [0, 1], \lambda \in [0, 1], \exists z \in [x, y]$  where  $V_i([x, z]) = \lambda \cdot V_i([x, y])$ .

### 2.1 Robertson-Webb Model

We study the complexity of cake-cutting algorithms in the model suggested by Robertson and Webb [RW98] and later formalized (and named) by Woeginger and Sgall [WS07]. This model allows for two types of queries:

1.  $\text{EVAL}_i(I)$ : Given an agent  $i$  and an interval  $I \subseteq [0, 1]$ , this query returns  $V_i(I)$ .
2.  $\text{CUT}_i(x, v)$ : Given an agent  $i$ , a point  $x \in [0, 1]$ , and a value  $v \in [0, 1]$ , this query returns the smallest point  $x' \in [x, 1]$  such that  $V_i([x, x']) = v$ .

To the best of our knowledge, every discrete cake-cutting algorithm can be (and has been) analyzed in this model. As a simple example, the cut-and-choose algorithm can be implemented by two queries as follows. First, a  $\text{CUT}_1(0, 1/2)$  will return the point  $y$  such that  $V_1([0, y]) = V_1([y, 1]) = 1/2$ . The algorithm can determine whether  $[0, y]$  or  $[y, 1]$  should be allocated to agent 1, by making a  $\text{EVAL}_2([0, y])$  and checking whether the value returned is at most  $1/2$ .

## 2.2 Fairness notions

Our goal is to produce allocations that are fair. An allocation  $A$  is **proportional** if  $V_i(A_i) \geq 1/n$  for all  $i \in N$ . An allocation  $A$  is **envy-free** if for all  $i, j \in N$ ,  $V_i(A_i) \geq V_i(A_j)$ . An allocation  $A$  is  **$\epsilon$ -perfect** if for all  $i, j \in N$ ,  $1/n + \epsilon \geq V_i(A_j) \geq 1/n - \epsilon$ ; an allocation  $A$  is **perfect** if it is 0-perfect. Finally, an allocation,  $A$ , is **super envy-free** if every agent values their own piece at least  $1/n$ , and values any other agent's piece at most  $1/n$ , i.e.,  $V_i(A_i) \geq \frac{1}{n} \geq V_i(A_j)$ , for all  $i, j \in N$ .

We write PROP, EF, SUPER-EF, and PERF for the set of all complete proportional, envy-free, super envy-free, and perfect allocations, respectively. For complete allocation, every perfect allocation is super envy-free, every super envy-free allocation is envy-free, and every envy-free allocation is proportional, and there exist perfect allocations that are not super envy-free, super envy-free allocations that are not envy-free, as well as envy-free allocations that are not proportional; that is,  $\text{PROP} \supsetneq \text{EF} \supsetneq \text{SUPER-EF} \supsetneq \text{PERF}$ .

In this paper, we define the following new notions of fairness.

**Definition 1** (Harmonically Coalition-Resistant- $k$  (HCR- $k$ )). Let  $k$  be an integer, such that  $n \geq k \geq 1$ . An allocation  $A$  is Harmonically Coalition-Resistant- $k$  (HCR- $k$ ) if it is complete and, for every non-empty subset of agents  $S \subseteq N$  such that  $|S| \leq k$ , and every  $i \in S$ ,  $V_i(A_{\bar{S}}) = \sum_{j \notin S} V_i(A_j) \leq \frac{n-|S|}{n-|S|+1}$ .

It is easy to see that the definition of Harmonically Coalition-Resistant-1 coincides with the definition of proportionality. Slightly overloading notation, we write HCR- $k$  for the set of all complete Harmonically Coalition-Resistant- $k$  allocations.

**Definition 2** (Linearly Coalition-Resistant- $k$  (LCR- $k$ )). Let  $k$  be an integer, such that  $n \geq k \geq 1$ . An allocation  $A$  is Linearly Coalition-Resistant- $k$  (LCR- $k$ ) if it is complete, for every non-empty subset of agents  $S \subseteq N$  such that  $|S| \leq k$ , and every  $i \in S$ ,  $V_i(A_{\bar{S}}) = \sum_{j \notin S} V_i(A_j) \leq \frac{n-|S|}{n}$ .

Slightly overloading notation, again, we write LCR- $k$  for the set of all complete Linearly Coalition-Resistant- $k$  allocations. Notice that every Linearly Coalition-Resistant- $k$  allocation is Harmonically Coalition-Resistant- $k$ , since  $\frac{n-|S|}{n} \leq \frac{n-|S|}{n-|S|+1}$  for all  $|S| \geq 1$ .

## 3 An Algorithm for HCR- $n$

In this section, we prove our main upper bound: HCR- $n$  allocations can be computed using  $O(n^4)$  queries in the Robertson-Webb model.

**Theorem 1.** Algorithm 1 computes a HCR- $n$  allocation using  $O(n^4)$  CUT and EVAL queries.

*Proof.* We first prove that every complete and proportional allocation  $A$  such that  $V_i(A_j) \geq \frac{1}{2n}$ , for all agents  $i, j$ , satisfies HCR- $n$ . Towards this, consider such an allocation  $A$ , and an arbitrary subset of agents  $S \subseteq N$  and  $i \in S$ .  $V_i(A_{\bar{S}}) = 1 - V_i(A_S) \leq 1 - \left(\frac{1}{n} + \frac{|S|-1}{2n}\right) = \frac{2n-|S|-1}{2n} \leq \frac{n-|S|}{n-|S|+1}$ , as long as  $|S| \leq n-1$ . For  $|S| = n$ ,  $V_i(A_{\bar{S}}) = 0 \leq \frac{n-|S|}{n-|S|+1}$ . Therefore,  $A \in \text{HCR-}n$ .

It remains to prove that Algorithm 1 finds an allocation with these properties (completeness, proportionality, and  $V_i(A_j) \geq \frac{1}{2n}$ ) using  $O(n^4)$  CUT and EVAL queries.

Algorithm 1 first computes an  $\frac{1}{4n}$ -perfect allocation  $B$  for  $4n/3$  agents, where the  $n/3$  extra agents, agents  $n+1, \dots, 4n/3$ , have arbitrary valuation functions. Let  $B_1, \dots, B_n$  be the first  $n$  pieces of this allocations; pieces  $B_{n+1}, \dots, B_{4n/3}$  are combined into the residue  $R$ , i.e.,  $R = \cup_{j=n+1}^{4n/3} B_j$ . Importantly, “real” agents are *not* allocated any of the  $B_j$  pieces (yet).

---

**ALGORITHM 1:** An algorithm for finding HCR- $n$  allocations
 

---

```

1 Let  $V_{n+1}^+, \dots, V_{4n/3}^+$  be  $n/3$  arbitrary valuation functions.
2  $B \leftarrow \frac{1}{4n}$ -PERFECT( $V_1, \dots, V_n, V_{n+1}^+, \dots, V_{4n/3}^+$ ).  $\triangleright$  Find an  $\epsilon$ -PERFECT allocation with the
   extra agents
3  $R \leftarrow \bigcup_{j=n+1}^{4n/3} B_j$ .  $\triangleright$  The pieces of the additional agents becomes the residue
4  $M \leftarrow \{1, \dots, n\}$ .  $\triangleright$  Initialize the set of active agents.
5 for  $X \in \{B_1, \dots, B_{n-1}\}$  do
6   Ask every agent  $i \in M$  to make a mark on  $R$  such that the piece to the left of the mark has
   value  $1/n - V_i(X)$ .
7   Let  $i^* \in M$  be the agent with the left-most mark on  $R$  (breaking ties arbitrarily).
8   Let  $R_{i^*}$  be the part of  $R$  to the left of  $i^*$ 's mark.
9    $A_{i^*} \leftarrow R_{i^*} \cup X$ .  $\triangleright$  Allocate to agent  $i^*$ 
10   $R \leftarrow R \setminus R_{i^*}$ .  $\triangleright$  Update the residue
11   $M \leftarrow M \setminus \{i^*\}$ .  $\triangleright$  Remove  $i^*$ 's piece from  $S$  and  $i^*$  from  $M$ 
12 end
13 for the remaining agent  $i \in M$ ,  $A_i \leftarrow B_n \cup R$ .
14 return  $A$ 

```

---

Algorithm 1 then proceeds à la Dubins-Spanier [DS61]/Last Diminisher [Ste48]. Let  $S$  be the subset of the  $B_j$  pieces still available. The algorithm asks every agent  $i$  to make a mark on  $R$ , such that their value to the left of the mark is equal to  $1/n - \max_{B \in S} V_i(B)$ . The agent  $i^*$  with the left-most mark (breaking ties arbitrarily) gets the two corresponding pieces (one from cutting the residue and their favorite piece from  $S$ ), and exits the process; we denote their final allocation by  $A_{i^*} = A'_{i^*} \cup R_{i^*}$ , where  $A'_{i^*}$  is their favorite piece in  $S$ , and (slightly overloading notation)  $R_{i^*}$  is the part of the residue allocated to them. The process is repeated until a single agent  $j$  is left, who is allocated the entire remaining residue, and the unique piece left in  $S$ .

**Query complexity.** Using the algorithm of Brânzei and Miltersen [BM15] for finding  $\epsilon$ -perfect allocation, computing  $(B, R)$  costs  $O((4n/3)^3/(1/4n)) = O(n^4)$  queries. This protocol produces at most  $O(n^2)$  intervals, and therefore, at most  $O(n^3)$  EVAL queries are necessary for every agent to compute the value of every interval. When the value of each interval is known, any query across any set of these intervals, which we refer to as a *super-query*, will require at most 2 queries (see proof below). Applying this, the “Last Diminisher” steps (the while loop in Algorithm 1) will cost another  $O(n^2)$  queries. Marking a piece of value  $\frac{1}{n} - V_i(S_j)$  on  $R$  for a single agent requires 1 super-query. So,  $O(n^2)$  super-queries, or  $O(n^2)$  actual queries are required for this step. Therefore, the overall complexity is  $O(n^4)$ .

**Claim 1.** Let  $\mathcal{I}$  be a collection of  $z$  noncontiguous intervals. If  $V_i(I_j)$  is known for all  $i \in N, I_j \in \mathcal{I}$ , then any super-query over any subset of intervals of  $\mathcal{I}$  will require at most 2 “real” queries.

*Proof.* Consider the super-query  $\text{EVAL}_i([x_1, x_2])$  where  $x_1 \in I$  and  $x_2 \in I'$  for some intervals  $I, I' \in \mathcal{I}$ . If  $I = I'$ ,  $x_1$  and  $x_2$  belong to the same interval, meaning only 1 actual query is required. If  $I \neq I'$ , then  $\text{EVAL}_i([x_1, x_2]) = \text{EVAL}_i([x_1, I_{\text{right}}]) + \text{EVAL}_i([I'_{\text{left}}, x_2]) + \sum_{I \in \mathcal{I}: I \text{ “between” } I, I'} V_i(I)$ . Now consider the super-query  $\text{CUT}_i(v_1, x_1)$  where  $x_1 \in I$  for some interval  $I \in \mathcal{I}$ . This requires at most 2 queries: First, perform the true query  $\text{EVAL}_i([x_1, I_{\text{right}}])$ . Then, using this evaluation along with those of the intervals to the right of  $I$  to determine the interval  $I'$  the mark  $x_2$  will be placed and the value  $v_2 = v_1 - V_i(x_1, I'_{\text{left}})$  that should be to the left of  $x_2$  in  $I'$ . Finally, only one actual query is required to determine the cut  $x_2$ ,  $x_2 = \text{CUT}_i(v_2, I'_{\text{left}})$ . Thus, given each agent’s evaluation of each noncontiguous interval, any super-query requires at most 2 actual queries.  $\square$

**Correctness of Algorithm 1.** All allocations are complete, by construction. First, we prove that each allocation is proportional, i.e.,  $V_i(A_i) \geq V_i(A'_i) \geq 1/n$ , which simply means that at each iteration, all remaining agents in  $M$  will be able to mark out a piece of value  $\frac{1}{n}$ . Consider some iteration  $t$ . Assuming that previous iterations were completed successfully, each agent in  $M$  did not have the left-most mark in  $R$ ,  $V_i(A'_j) \leq \frac{1}{n}$  for all  $i \in M, j \in N \setminus M$ . Let  $B_t$  be the piece arbitrarily chosen to

create an allocation in iteration  $t$ .  $V_i(R) + V_i(B_t) = 1 - \sum_{j \in N \setminus M} V_i(A'_j) - \sum_{B_j \in S \setminus B_t} V_i(B_j) \geq 1 - \frac{t-1}{n} - \frac{n-t}{n} = \frac{1}{n}$ . Thus, in this iteration, all agents will be able to place a mark on the residue. Starting with the trivial case of  $t = 1$  and inductively applying this logic shows that enough of  $R$  will remain for each agent remaining in  $M$  to place such a mark.

Regarding the second condition,  $V_i(A_j) \geq \frac{1}{2n}$  for all agents  $i, j$ , we have the following. By the fact that  $B$  is  $\frac{1}{4n}$ -perfect for the instance with  $4n/3$  agents:  $V_i(B_\ell) \geq \frac{1}{4n/3} - \frac{1}{4n} = \frac{1}{2n}$ . Observing that  $V_i(A_j) = V_i(A'_j) + V_i(R_j) \geq V_i(A'_j)$ , and  $A'_j$  is one the  $B_\ell$  pieces, completes the proof.  $\square$

#### 4 Algorithms for Relaxations of LCR- $n$

In this section, we study approximations of LCR- $k$ , aiming to bypass the strong lower bounds of Theorem 7. Since LCR-2 is impossible, but LCR-1 (aka, proportionality) is easy to achieve, the minimally weaker notion that we could hope to achieve is the following: (i) if  $|S| = 1$ , then  $V_i(A_S) \leq \frac{n-1}{n}$ , for all  $i \in S$  (i.e., the allocation is proportional), (ii) if  $|S| = 2$ , then  $V_i(A_S) \leq \frac{n-2}{n}(1 + \delta)$ , for all  $i \in S$ . More generally, we have

**Definition 3** ( $\delta$ -LCR- $k$ ). Let  $k$  be an integer, such that  $n \geq k \geq 1$ . An allocation  $A$  is  $\delta$ -Linearly Coalition-Resistant- $k$  (LCR- $k$ ) if it is proportional, and, for every non-empty subset of agents  $S \subseteq N$  such that  $2 \leq |S| \leq k$ , and every  $i \in S$ ,  $V_i(A_S) = \sum_{j \notin S} V_i(A_j) \leq \frac{n-|S|}{n}(1 + \delta)$ .

We prove that  $\delta$ -LCR- $n$  allocations can be computed efficiently, by expanding on our approach for HCR- $n$ , i.e., Algorithm 1. First, it is easy to see that a complete,  $\epsilon$ -perfect, and proportional allocation  $A$  is  $(\epsilon n)$ -LCR- $n$ : for all  $S \subseteq N$  such that  $2 \leq |S|$ , and  $i \in S$ ,  $V_i(A_S) \leq (n - |S|) \left( \frac{1}{n} + \epsilon \right) = \frac{n-|S|}{n}(1 + \epsilon n)$ . We give an algorithm, Algorithm 2, for finding a complete,  $\epsilon$ -perfect, and exactly proportional allocation that requires  $O\left(\frac{n^5}{\epsilon} \frac{\ln(1/\epsilon)}{\ln(n)}\right)$  queries, which might be of independent interest. To the best of our knowledge, the closest results in the literature are: (i) an algorithm that finds proportional and  $\epsilon$ -equitable allocations using  $O(n(\log n + \log(1/\epsilon)))$  queries, by Cechlárová and Pillárová [CP12], and (ii) the  $O(n^3/\epsilon)$  algorithm of Brânzei and Miltersen [BM15] for finding  $\epsilon$ -perfect allocations. Algorithm 2 immediately implies that the query complexity of finding a  $\delta$ -LCR- $n$  allocation is  $O\left(\frac{n^6}{\delta} \frac{\ln(1/\delta)}{\ln(n)}\right)$ .

**Theorem 2.** For all  $\epsilon > 0$ , there exists  $d \in \Theta\left(\frac{\ln(1/\epsilon)}{\ln(n)}\right)$  and  $\epsilon' \in \Theta\left(\frac{\epsilon}{n^2}\right)$ , such that Algorithm 2, with parameters  $d$  and  $\epsilon'$ , computes a complete,  $\epsilon$ -perfect, and proportional allocation using  $O\left(\frac{n^5}{\epsilon} \frac{\ln(1/\epsilon)}{\ln(n)}\right)$  CUT and EVAL queries.

Algorithm 2 differs from Algorithm 1 in its computation of  $S$  and the residue  $R$  prior to the final “Last Diminisher” phase (i.e. line 5 in Algorithm 1, line 11 in Algorithm 2). In Algorithm 1, the goal was for each piece of  $S$  to have value at least  $\frac{1}{2n}$ , but at most  $\frac{1}{n}$  for any agent. In Algorithm 2, the goal is similar, but quite harder. Each piece of  $S$  must have value at least  $\frac{1}{n} - \tilde{\epsilon}$  but at most  $\frac{1}{n}$  for any agent, for  $\tilde{\epsilon} = \frac{\epsilon}{n}$ ; as we show, this will suffice for  $\epsilon$ -perfection.

To achieve this Algorithm 2 computes  $S$  and  $R$  recursively. Starting with the entire cake, it computes an  $(n + 1)$ -piece  $\epsilon'$ -perfect allocation, designating a single piece of the allocation as  $R$  and the remaining  $n$  pieces as  $S$ . Then it repeats this process, treating the residue  $R$  itself as an entire cake, designating a piece of the new allocation as the new residue, and combining each of the remaining  $n$  pieces with a distinct piece in  $S$ . This process is done  $d$  times in total, with the goal that the expected value of each piece of  $S$  (for any agent) is spaced enough between  $\frac{1}{n} - \tilde{\epsilon}$  and  $\frac{1}{n}$ . This allows us to choose a non-zero  $\epsilon'$  that keeps the potential values of each piece in  $S$  for any agent within a desired range. In short, if the number of subdivisions  $d$  and the margin of error for near-perfect divisions  $\epsilon'$  are chosen correctly, then  $\frac{1}{n} - \tilde{\epsilon} \leq V_i(B_j) \leq \frac{1}{n}$  for all  $i, j \in N$ .

*Proof of Theorem 2.* In Appendix C, we show that if the parameters  $\epsilon'$  and  $d$  are picked correctly, Algorithm 2 outputs an allocation where  $V_i(A_i) \geq \frac{1}{n}$ ,  $V_i(A_j) \geq \frac{1}{n} - \tilde{\epsilon}$ ,  $\forall i, j \in N$  for any  $\tilde{\epsilon} > 0$ . By picking  $\tilde{\epsilon} = \frac{\epsilon}{n}$ , our allocations are  $\epsilon$ -perfect (in addition to proportional and complete). To see this, notice that (i)  $V_i(A_j) \geq \frac{1}{n} - \tilde{\epsilon} = \frac{1}{n} - \frac{\epsilon}{n} \geq \frac{1}{n} - \epsilon$ , and (ii)  $V_i(A_j) \leq 1 - (n - 1) \left( \frac{1}{n} - \tilde{\epsilon} \right) =$

---

**ALGORITHM 2:** An algorithm for finding complete,  $\epsilon$ -perfect, and exactly proportional allocations
 

---

**Input:** Parameters  $\epsilon'$  and  $d$ .

```

1 Let  $V^+$  be an arbitrary valuation function.
2 Let  $B$  be a set of  $n$  empty pieces.  $R \leftarrow \{[0, 1]\}$   $\triangleright$  Initialize the residue as the entire cake
3 for  $t \in \{1, \dots, d\}$  do
4    $P \leftarrow \epsilon'$ -PERFECT( $V_1, \dots, V_n, V^+$ ) with respect to, and normalized on,  $R$ .
5    $R \leftarrow P_{n+1}$ 
6   for  $i \in [1, n]$  do
7      $B_i \leftarrow B_i \cup P_i$   $\triangleright$  Assign the other  $n$  pieces in  $B$  to a piece in  $S$ 
8   end
9 end
10  $M \leftarrow \{1, \dots, n\}$ 
11 for  $X \in \{B_1, \dots, B_{n-1}\}$  do
12   Ask every agent  $i \in M$  to make a mark on  $R$  such that the piece to the left of the mark has
     value  $1/n - V_i(X)$ .
13   Let  $i^* \in M$  be the agent with the left-most mark on  $R$  (breaking ties arbitrarily).
14   Let  $R_{i^*}$  be the part of  $R$  to the left of  $i^*$ 's mark.
15    $A_{i^*} \leftarrow R_{i^*} \cup X$ .  $\triangleright$  Allocate to agent  $i^*$ 
16    $R \leftarrow R \setminus R_{i^*}$ .  $\triangleright$  Update the residue
17    $M \leftarrow M \setminus \{i^*\}$ .  $\triangleright$  Remove  $i^*$ 's piece from  $S$  and  $i^*$  from  $M$ 
18 end
19 For the remaining agent  $i \in M$ ,  $A_i \leftarrow B_n \cup R$ .
20 return  $A$ 
  
```

---

$1 - (n-1) \left( \frac{1}{n} - \frac{\epsilon}{n} \right) = \frac{1}{n} + \epsilon - \frac{\epsilon}{n} \leq \frac{1}{n} + \epsilon$ . In the remainder of the proof, we prove that Algorithm 2 satisfies the desired property, and has query complexity  $O\left(\frac{n^4}{\epsilon} \cdot \frac{\ln(1/\tilde{\epsilon})}{\ln(n)}\right)$ .

**Query Complexity.** We claim that the algorithm requires at most  $O\left(\frac{n^4}{\epsilon} \cdot \frac{\ln(1/\tilde{\epsilon})}{\ln(n)}\right)$  to compute an allocation such that  $V_i(A_i) \geq \frac{1}{n}$ , and  $V_i(A_j) \geq \frac{1}{n} - \tilde{\epsilon}$ . The algorithm involves (i) computing  $d$   $(n+1)$ -piece  $\epsilon'$ -perfect allocations and (ii) performing a “cut-and-match Last Diminisher.” Brânzei and Miltersen give a protocol for finding  $\epsilon$ -perfect allocations that uses  $O\left(\frac{n^3}{\epsilon}\right)$  queries [BM15], and it is well-known that traditional “Last Diminisher” requires  $O(n^2)$  queries. However, it is premature to conclude that this would require  $O\left(\frac{n^3}{\epsilon'}\right) \cdot d + O(n^2) \in O\left(\frac{n^3}{\epsilon'} \cdot d\right)$  queries. Many of these “queries” are performed over a collection of noncontiguous intervals, rather than a contiguous cake, i.e., they are super-queries. To account for this, similar to our analysis of Algorithm 1, we maintain each agent’s valuation of each interval, so that each super-query requires at most 2 actual queries. This is shown as Claim 1 in Algorithm 1; an identical argument works here as well.

Next, we show that if we tracked each agent’s evaluation of each noncontiguous interval, then the algorithm would still require just  $O\left(\frac{n^3}{\epsilon'} \cdot d\right)$  (actual) queries. Prior to computing each  $\epsilon'$ -perfect allocation, we must normalize each agent’s valuation function over  $R$ . While we do not have access to valuation functions, this can be achieved by scaling the output of an EVAL super-query by  $\frac{1}{V_i(R)}$ , or scaling the input value of a CUT super-query by  $V_i(R)$ . Fortunately, since  $R$  is a set of intervals whose values are assumed to be known, this requires no additional queries. Thus, computing all  $d$   $\epsilon'$ -perfect allocations (with  $(n+1)$ -pieces each) requires  $O\left(\frac{n^3}{\epsilon'} \cdot d\right)$  (actual) queries. As for performing our “cut-and-match Last Diminisher”, at worst, an agent will perform EVAL super-queries on every piece in  $S$  and  $n$  CUT super-queries on  $R$ . Applying our prior result again, this translates to only  $O(n^2)$  actual queries in total (which is dominated by  $O\left(\frac{n^3}{\epsilon'} \cdot d\right)$ ).

Finally, we show that tracking each agent’s evaluation of each noncontiguous interval requires  $O(n^3 \cdot d)$  queries (which is also dominated by  $O\left(\frac{n^3}{\epsilon'} \cdot d\right)$ ). The  $\epsilon$ -perfect algorithm by Brânzei and

Miltersen [BM15] produces allocations with  $O(n^2)$  cuts (thus introducing  $O(n^2)$  noncontiguous intervals). To incorporate each new cut  $x$  on an existing interval  $I$  (thus creating intervals  $I_1^+ = [I_{\text{left}}, x]$  and  $I_2^+ = [x, I_{\text{right}}]$ ) for an agent  $i$ , we perform 1 query  $\text{EVAL}_i(I_1^+)$  to determine the values of the new intervals created ( $V_i(I_2^+)$  is deduced from  $V_i(I) - V_i(I_1^+)$ ). Thus, introducing a cut (and by extension a noncontiguous interval) only requires  $O(n)$  true queries to keep our noncontiguous interval evaluations up-to-date. To account for the scaling performed in iterations performed on the residue, no additional queries are necessary as the scaling factor can be computed using the value of the residue (which is comprised of intervals of known valuations). Since our  $d$   $\epsilon'$ -perfect allocations introduce  $O(n^2 \cdot d)$  cuts and our “cut-and-match Last Diminisher” introduces  $O(n)$  cuts (one for each of the first  $n - 1$  pieces allocated), tracking the desired information requires  $O(n^3 \cdot d) + O(n) \in O(n^3 \cdot d)$  queries in total.

Putting these pieces together, we can conclude that Algorithm 2 computes complete and proportional allocations, where  $V_i(A_j) \geq \frac{1}{n} - \tilde{\epsilon}$  for all  $i, j \in N$ , using  $O\left(\frac{n^3}{\epsilon'} \cdot d\right) = O\left(\frac{n^4}{\epsilon} \cdot \frac{\ln(1/\tilde{\epsilon})}{\ln(n)}\right)$  queries in total (and therefore a complete, proportional and  $\epsilon$ -perfect allocation using  $O\left(\frac{n^5}{\epsilon} \cdot \frac{\ln(1/\epsilon)}{\ln(n)}\right)$  queries).

**Correctness.** From our earlier discussion on query complexity and choice of parameters  $d$  and  $\epsilon'$ , we have that the first phase of the algorithm (lines 1 to 10) end with  $n$  pieces,  $B_1, \dots, B_n$ , such that  $\frac{1}{n} - \tilde{\epsilon} \leq V_i(B_j) \leq \frac{1}{n}$ , as well as a residue. Since the final allocation of every agent  $j$  is a superset of some piece  $B_\ell$ , we have that  $V_i(A_j) \geq V_i(B_\ell) \geq \frac{1}{n} - \tilde{\epsilon}$ . Following this stage of the algorithm, we can apply the same logic as done for Algorithm 1 to conclude that, for all  $i \in N$ ,  $V_i(A_i) \geq \frac{1}{n}$  after the “cut-and-match Last Diminisher” stage. Thus, by Theorem 2 we can conclude the final allocation  $A$  to be  $\delta$ -LCR- $n$ .  $\square$

## Acknowledgements

The authors would like to thank Yorgos Amanatidis, Yorgos Christodoulou, John Fearnley, and Vangelis Markakis for their input in the initial development of this project. The authors would also like to thank Ariel Procaccia for the valuable discussions and suggestions. Alexandros Psomas is supported in part by an NSF CAREER award CCF-2144208, and a research award from the Herbert Simon Family Foundation.

## References

- [ACF<sup>+</sup>18] Georgios Amanatidis, George Christodoulou, John Fearnley, Evangelos Markakis, Christos-Alexandros Psomas, and Eftychia Vakaliou. An improved envy-free cake cutting protocol for four agents. In *Algorithmic Game Theory: 11th International Symposium, SAGT 2018, Beijing, China, September 11-14, 2018, Proceedings 11*, pages 87–99. Springer, 2018.
- [Alo87] Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.
- [AM16a] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–427. IEEE, 2016.
- [AM16b] Haris Aziz and Simon Mackenzie. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 454–464, 2016.
- [AR21] Haris Aziz and Simon Rey. Almost group envy-free allocation of indivisible goods and chores. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 39–45, 2021.
- [BM15] Simina Brânzei and Peter Bro Miltersen. A dictatorship theorem for cake cutting. In *Twenty-fourth international joint conference on artificial intelligence*. Citeseer, 2015.
- [BN22] Simina Brânzei and Noam Nisan. The query complexity of cake cutting. *Advances in Neural Information Processing Systems*, 35:37905–37919, 2022.

- [BTD92] Marcus Berliant, William Thomson, and Karl Dunz. On the fair division of a heterogeneous commodity. *Journal of Mathematical Economics*, 21(3):201–216, 1992.
- [CF20] Ágnes Cseh and Tamás Fleiner. The complexity of cake cutting with unequal shares. *ACM Transactions on Algorithms (TALG)*, 16(3):1–21, 2020.
- [CFSV19] Vincent Conitzer, Rupert Freeman, Nisarg Shah, and Jennifer Wortman Vaughan. Group fairness for the allocation of indivisible goods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1853–1860, 2019.
- [CLS25] Ioannis Caragiannis, Kasper Green Larsen, and Sudarshan Shyam. A new lower bound for multi-color discrepancy with applications to fair division. *arXiv preprint arXiv:2502.10516*, 2025.
- [CP12] Katarína Cechlárová and Eva Pillárová. On the computability of equitable divisions. *Discrete Optimization*, 9(4):249–257, 2012.
- [DS61] Lester E Dubins and Edwin H Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1P1):1–17, 1961.
- [EP84] Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984.
- [EP06] Jeff Edmonds and Kirk Pruhs. Balanced allocations of cake. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06)*, pages 623–634, 2006.
- [EP11] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. *ACM Transactions on Algorithms (TALG)*, 7(4):1–12, 2011.
- [FT22] Uriel Feige and Yehonatan Tahan. On allocations that give intersecting groups their fair share. *arXiv preprint arXiv:2204.06820*, 2022.
- [Hus11] Farhad Husseinov. A theory of a heterogeneous divisible commodity exchange economy. *Journal of Mathematical Economics*, 47(1):54–59, 2011.
- [KSV20] Maria Kyropoulou, Warut Suksompong, and Alexandros A Voudouris. Almost envy-freeness in group resource allocation. *Theoretical Computer Science*, 841:110–123, 2020.
- [Lia40] AA Liapounoff. Sur les fonctions-vecteurs completement additives. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 4(6):465–478, 1940.
- [MS22] Pasin Manurangsi and Warut Suksompong. Almost envy-freeness for groups: Improved bounds via discrepancy theory. *Theoretical Computer Science*, 930:179–195, 2022.
- [Pro09] Ariel D Procaccia. Thou shalt covet thy neighbor’s cake. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.
- [PW17] Ariel D Procaccia and Junxing Wang. A lower bound for equitable cake cutting. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pages 479–495, 2017.
- [RW98] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can*. AK Peters/CRC Press, 1998.
- [SHN19] Erel Segal-Halevi and Shmuel Nitzan. Fair cake-cutting among families. *Social Choice and Welfare*, 53:709–740, 2019.
- [SHS20] Erel Segal-Halevi and Warut Suksompong. How to cut a cake fairly: A generalization to groups. *The American Mathematical Monthly*, 128(1):79–83, 2020.
- [SHS23] Erel Segal-Halevi and Warut Suksompong. Cutting a cake fairly for groups revisited. *The American Mathematical Monthly*, 130(3):203–213, 2023.
- [Ste48] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

- [STZ23] Jonathan Scarlett, Nicholas Teh, and Yair Zick. For one and all: Individual and group fairness in the allocation of indivisible goods. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 2466–2468, 2023.
- [Wik25] Wikipedia contributors. Hugo Steinhaus. [https://en.wikipedia.org/wiki/Hugo\\_Steinhaus](https://en.wikipedia.org/wiki/Hugo_Steinhaus), 2025. [Online; accessed 2-February-2025].
- [WS07] Gerhard J Woeginger and Jiří Sgall. On the complexity of cake cutting. *Discrete Optimization*, 4(2):213–220, 2007.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The paper and supplementary material contain the proofs of all theorems, lemmas, claims, etc stated in the abstract.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: See the introduction for a discussion of the scope of this paper. The model includes all assumptions needed for the results to hold. The results themselves (theorems, lemmas, etc) state any additional requirements.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The model is in Section 2. Proofs are included in the main body and supplementary material (technical appendix).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have checked the guidelines and we follow them.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The introduction discusses relevance to practice. The work is theoretical, so there are no societal impacts of the nature described in the guidelines.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper is theoretical; there are no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No LLMs were used.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## A Relations Between Fairness Notions

In this section, we show the relation between our new fairness notions, and existing fairness notions: proportionality, envy-freeness, and perfection.

### A.1 HCR-k Relations

**Theorem 3.**  $\text{PROP} = \text{HCR-1} \supsetneq \text{HCR-2} \supsetneq \dots \supsetneq \text{HCR-(}n-1\text{)} = \text{HCR-}n \supsetneq \text{EF}$ .

*Proof of Theorem 3.* The first equality,  $\text{PROP} = \text{HCR-1}$  is immediate from the definition of HCR-1: for  $|S| = 1$ ,  $1 - V_i(A_i) = V_i(A_{\bar{S}}) \leq \frac{n-1}{n} = 1 - \frac{1}{n}$ . It is equally easy to see that  $\text{HCR-}k \supsetneq \text{HCR-(}k+1\text{)}$ , for  $1 \leq k \leq n-2$ , since the conditions required to satisfy the latter are a subset of the conditions required to satisfy the former notion; Claim 2 gives an allocation  $A$  such that  $A \in \text{HCR-}k$ , but  $A \notin \text{HCR-(}k+1\text{)}$ , proving that  $\text{HCR-}k \supsetneq \text{HCR-(}k+1\text{)}$ .

**Claim 2.**  $\text{HCR-}k$  does not imply  $\text{HCR-(}k+1\text{)}$  for  $1 \leq k \leq n-1$ .

*Proof.* Let  $A \in \text{HCR-}k$  be an allocation such that, for all  $i \in N$ ,  $V_i(A_i) = \frac{1}{n-k+1}$ , and there exists an agent  $j_i \in N \setminus \{i\}$ , such that  $V_i(A_{j_i}) = 1 - \frac{1}{n-k+1}$  (all other  $V_i(A_w)$  are 0). It is straightforward to construct such an allocation.  $A \in \text{HCR-}k$ , since  $V_i(A_{\bar{S}}) \leq 1 - \frac{1}{n-k+1} \leq \frac{n-|S|}{n-|S|+1}$  for all  $S$  such that  $|S| \leq k$  and  $i \in S$ . However, by picking  $S$  such that  $i \in S$ ,  $j_i \notin S$ , and  $|S| = k+1$ , we have that  $V_i(A_{\bar{S}}) = 1 - \frac{1}{n-k+1} = \frac{n-k}{n-k+1} > \frac{n-|S|}{n-|S|+1}$ , i.e.,  $A \notin \text{HCR-(}k+1\text{)}$ .  $\square$

The last equality,  $\text{HCR-(}n-1\text{)} = \text{HCR-}n$ , again immediately follows from the definitions. Finally, Claim 3 proves that  $\text{HCR-}n \supsetneq \text{EF}$ .

**Claim 3.**  $\text{EF} \subsetneq \text{HCR-}n$ .

*Proof.* To see that  $\text{EF} \subseteq \text{HCR-}n$ , consider an allocation  $A \in \text{EF}$ ; we will show that  $A \in \text{HCR-}n$ . Let  $S \subseteq N$  be an arbitrary subset of agents of size  $|S| \leq n$ . By the definition of envy-freeness, we have that, for  $i, j \in N$ ,  $V_i(A_j) \leq V_i(A_i)$ . Consider  $i \in S$ ; adding up for all  $j \notin S$  we have that

$$\begin{aligned} V_i(A_{\bar{S}}) &\leq (n - |S|)V_i(A_i) \\ &\leq (n - |S|)V_i(A_S) && (V_i(A_i) \leq V_i(A_S)) \\ &= (n - |S|)(1 - V_i(A_{\bar{S}})). \end{aligned}$$

Re-arranging we get the desired inequality:  $V_i(A_{\bar{S}}) \leq \frac{n-|S|}{n-|S|+1}$ .

To see that  $\text{HCR-}n$  does not imply  $\text{EF}$ , consider an allocation  $A \in \text{HCR-}n$  such that, for all agents  $i \in N$   $V_i(A_i) = \frac{1}{3}$ ; furthermore, for every agent  $i \in N$ , there exist distinct agents  $j_i$  and  $\ell_i$  such that  $V_i(A_{j_i}) = \frac{1}{2}$  and  $V_i(A_{\ell_i}) = \frac{1}{6}$  (and  $V_i(A_w) = 0$  for all other agents  $w$ ). It is easy to see that such an allocation can be constructed.  $A \in \text{HCR-}n$ , since: (i) for all  $S \subseteq N$ ,  $|S| \leq n-2$ , and all  $i \in S$ ,  $V_i(A_{\bar{S}}) \leq 1 - V_i(A_i) = \frac{2}{3} \leq \frac{n-|S|}{n-|S|+1}$ , (ii) for all  $S \subseteq N$ ,  $|S| = n-1$ , and all  $i \in S$ ,  $V_i(A_{\bar{S}}) \leq 1 - (V_i(A_i) + \min\{V_i(A_{j_i}), V_i(A_{\ell_i})\}) \leq \frac{1}{2} \leq \frac{n-|S|}{n-|S|+1}$ , and (iii)  $V_i(A_{\bar{S}}) = 0$  for  $S = N$ . However,  $A \notin \text{EF}$ , since every agent  $i \in N$  envies agent  $j_i$ .  $\square$

This concludes the proof of Theorem 3.  $\square$

### A.2 LCR-k Relations

**Theorem 4.**  $\text{PROP} = \text{LCR-1} \supsetneq \text{LCR-2} \supsetneq \dots \supsetneq \text{LCR-(}n-1\text{)} = \text{LCR-}n = \text{SUPER-EF}$ .

*Proof of Theorem 4.* By the definition of  $\text{LCR-}k$ , we have that  $\text{PROP} = \text{LCR-1}$ ,  $\text{LCR-}k \supsetneq \text{LCR-(}k+1\text{)}$  for  $1 \leq k \leq n-1$ , and  $\text{LCR-(}n-1\text{)}$  is equivalent to  $\text{LCR-}n$ . The theorem follows from Claim 4 and Claim 5.

**Claim 4.**  $\text{LCR-}k$  does not imply  $\text{LCR-(}k+1\text{)}$ , for any  $1 \leq k < n-1$ .

*Proof.* Consider an allocation  $A \in \text{LCR-}k$  such that, for all  $i \in N$ : (i)  $V_i(A_i) = \frac{k}{n}$ , (ii)  $V_i(A_j) = 0$ , for all  $j \in Z_i$ , for some  $Z_i \subset N \setminus \{i\}$  such that  $|Z_i| = k$ , and (iii)  $V_i(A_j) = \left(1 + \frac{1}{n-k-1}\right) \cdot \frac{1}{n}$ , for all  $j \in N \setminus (Z_i \cup \{i\})$ .  $S_i = Z_i \cup \{i\}$ ;  $|S_i| = k + 1$ . Then,  $V_i(A_{S_i}) = V_i(A_i) = \frac{k}{n}$ , and therefore  $V_i(A_{\bar{S}_i}) = \frac{n-k}{n} < \frac{n-|S_i|}{n}$ , i.e.  $A \notin \text{LCR-}(k+1)$ .  $\square$

**Claim 5.**  $\text{LCR-}n = \text{SUPER-EF}$ .

*Proof.* Consider an allocation  $A \in \text{LCR-}n$ . For all  $S \subseteq N$  such that  $|S| = n - 1$ , and all  $i \in S$ ,  $V_i(A_S) \leq \frac{1}{n}$ ; however,  $\bar{S}$  has a single (arbitrary) agent. That is,  $V_i(A_{\bar{S}}) = V_i(A_j) \leq \frac{1}{n}$ , for all  $j \in N \setminus \{i\}$ . And, since  $\text{LCR-}n \subseteq \text{PROP}$ , we also have  $V_i(A_i) \geq \frac{1}{n}$ , therefore,  $A \in \text{SUPER-EF}$ .

Consider an allocation  $A \in \text{SUPER-EF}$ . We have that, for all  $i, j \in N$ ,  $j \neq i$ ,  $\frac{1}{n} \geq V_i(A_j)$ . Adding up this for all  $j \notin S$ , for some arbitrary set  $S \subseteq N$ ,  $|S| \leq n$ , such that  $i \in S$ , we have  $V_i(A_{\bar{S}}) \leq \frac{n-|S|}{n}$ , i.e.,  $A \in \text{LCR-}n$ .  $\square$

This concludes the proof of Theorem 4  $\square$

**Theorem 5.**  $\text{EF} \supsetneq \text{LCR-}\lceil \frac{n}{2} \rceil$ , but  $\text{EF} \not\supseteq \text{LCR-}(\lceil \frac{n}{2} \rceil - 1)$  and  $\text{EF} \not\supseteq \text{LCR-}2$ .

*Proof of Theorem 5.* Consider an allocation  $A \in \text{LCR-}\lceil \frac{n}{2} \rceil$ . Let  $i, j \in N$  be two arbitrary agents. Let  $S \subseteq N \setminus \{j\}$ , such that (i)  $i \in S$ , and (ii),  $|S| = \lceil \frac{n}{2} \rceil$ . We have that  $V_i(A_S) = 1 - V_i(A_{\bar{S}}) \geq 1 - \frac{n-|S|}{n} = \frac{1}{2} + \frac{n \bmod 2}{2n}$ . Let  $S'$  be such that  $\bar{S}' = (S \setminus \{i\}) \cup \{j\}$  (i.e.,  $S' = \{i\} \cup (N \setminus (\{j\} \cup S))$ ). We have that  $|S'| = n - \lceil \frac{n}{2} \rceil = \lfloor \frac{n}{2} \rfloor$ , therefore, since  $A \in \text{LCR-}\lceil \frac{n}{2} \rceil$ , and  $i \in S'$ ,  $V_i(A_{\bar{S}'}) \leq \frac{n-|S'|}{n} = \frac{1}{2} + \frac{n \bmod 2}{2n}$ . Therefore,  $V_i(A_S) \geq V_i(A_{\bar{S}'})$ . However,  $V_i(A_S) = V_i(A_i) + \sum_{z \in S \setminus \{i\}} V_i(A_z)$ , and  $V_i(A_{\bar{S}'}) = V_i(A_j) + \sum_{z \in S \setminus \{i\}} V_i(A_z)$ . Therefore, we have that  $V_i(A_i) \geq V_i(A_j)$ , i.e.,  $i$  does not envy  $j$ .

Next, notice that  $\text{LCR-}(\lceil \frac{n}{2} \rceil - 1)$  (and by extension,  $\text{LCR-}k$  for  $k < \lceil \frac{n}{2} \rceil - 1$ ) does not imply EF. Consider an allocation  $A \in \text{LCR-}(\lceil \frac{n}{2} \rceil - 1)$  such that, for all  $i \in N$ ,  $V_i(A_i) = (\lceil \frac{n}{2} \rceil - 1) \cdot \frac{1}{n}$ , and there exists an agent  $j_i$  such that  $V_i(A_{j_i}) = 1 - (\lceil \frac{n}{2} \rceil - 1) \cdot \frac{1}{n}$ , and  $V_i(A_z) = 0$ , for all agents  $z \neq i, j_i$ . Since  $V_i(A_i) < V_i(A_{j_i})$ , envy-freeness is violated.

Finally, EF does not imply LCR-2 (and by extension,  $\text{LCR-}k$  for  $k > 2$ ). Consider an allocation  $A \in \text{EF}$  such that, for all  $i \in N$  there exists an agent  $j_i$ , such that  $V_i(A_{j_i}) = 0$ , and  $V_i(A_z) = \frac{1}{n-1}$ , for all  $z \neq i, j_i$ . For  $S = \{i, j_i\}$  and  $n > 2$ ,  $V_i(A_{\bar{S}}) = \frac{n-2}{n-1} > \frac{n-|S|}{n} = \frac{2}{n}$ .  $\square$

## B Lower Bounds

In this section, we state our lower bounds for HCR-2 (Theorem 6) and LCR-2 (Theorem 7).

**Theorem 6.** Computing HCR-2 allocations requires  $\Omega(n^2)$  queries.

*Proof.* For this proof, we use many definitions and lemmas from the work of Procaccia [Pro09], who proves the  $\Omega(n^2)$  lower bound for finding envy-free allocations.

Consider an arbitrary algorithm. First, to analyze the information available to the algorithm at each step, we define, for every agent  $i \in N$  and every step  $t$ , a set of disjoint intervals  $\Pi_i^t$ , that is a partition of  $[0, 1]$ . We say that interval  $I \in \Pi_i^t$  is active with respect to agent  $i$  at step  $t$ .

$\Pi_i^t$ s are defined recursively.  $\Pi_i^0 = \{[0, 1]\}$ , since the only information available to the algorithm after 0 steps/queries is that  $V_i([0, 1]) = 1$ . Assuming that at step  $t$  we have  $\Pi_i^t$ , if at step  $t + 1$  the algorithm does not make a query for agent  $i$  (i.e., the query at step  $t + 1$  is  $\text{EVAL}_j$  or  $\text{CUT}_j$  for some  $j \neq i$ ), then  $\Pi_i^{t+1} = \Pi_i^t$ . Otherwise,  $\Pi_i^{t+1}$  gets updated accordingly. For example, if the query is  $\text{EVAL}_i(x_1, x_2)$ , where  $x_1 \in I_1$  and  $x_2 \in I_2$ , for some intervals  $I_1, I_2 \in \Pi_i^t$ , then, informally,

$$\Pi_i^{t+1} = (\Pi_i^t \setminus \{I_1, I_2\}) \cup \{[left(I_1), x_1], [x_1, right(I_1)], [left(I_2), x_2], [x_2, right(I_2)]\},$$

where for an interval  $I = [a, b]$ ,  $left(I) = a$  and  $right(I) = b$ . Intuitively, the algorithm at step  $t$  “knew”  $i$ ’s value for  $I_1$  and  $I_2$ , and after the  $\text{EVAL}_i(x_1, x_2)$ , it can infer (at most) the value of agent  $i$  for

four additional intervals:  $[left(I_1), x_1]$ ,  $[x_1, right(I_1)]$ ,  $[left(I_2), x_2]$ , and  $[x_2, right(I_2)]$  (noting, that the first two imply the value for  $I_1$  and the second two imply the value for  $I_2$ ). Procaccia [Pro09] proves two crucial lemmas:

**Lemma 1** ([Pro09]; Lemma 3.2). For all  $i \in N$  and stage  $t$ ,  $|\Pi_i^{t+1}| - |\Pi_i^t| \leq 2$ .

**Lemma 2** ([Pro09]; Lemma 3.3). For all  $i \in N$  and stage  $t$ ,  $\Pi_i^t$  has the following properties:

1. For every  $I \in \Pi_i^t$ ,  $V_i(I)$  is known to the algorithm at stage  $t$ .
2. For every  $I \in \Pi_i^t$ ,  $I' \subsetneq I$ , and  $0 \leq \lambda \leq 1$ , it might be the case (based on the information available to the algorithm at stage  $t$ ) that  $V_i(I') = \lambda V_i(I)$ .

Consider an adversary that, for every agent  $i \in N$ , responds as if the valuation of the agent was uniform over the interval  $[0, 1]$  (i.e., responds  $\text{EVAL}_i([x_1, x_2]) = x_2 - x_1$ , and  $\text{CUT}_i(x, v) = x + v$ ). Let  $T$  be the number of queries our algorithm asks before it terminates and outputs allocation  $A = (A_1, \dots, A_n) \in \text{HCR-2}$ .

First, we claim that for all  $i \in N$ , there exists an active interval  $I_i \subseteq A_i$ , such that (i)  $I_i \in \Pi_i^T$ , and (ii)  $V_i(I_i) \geq \frac{1}{n}$ . If this is not the case, one can define  $V_i$  (consistently with  $\Pi_i^T$ ) such that proportionality is violated. Concretely, (1) if  $I \notin \Pi_i^T$  for all  $I \subseteq A_i$  then we can concentrate the value of all  $I$  such that  $I \cap A_i \neq \emptyset$  to the sub-intervals outside of  $A_i$  (and therefore,  $V_i(A_i) = 0$ ), and (2) if  $V_i(I) < \frac{1}{n}$  for all  $I \subseteq A_i$ ,  $I \in \Pi_i^T$ , then we can pick  $V_i$  such that  $V_i(A_i) = \max_{I \in \Pi_i^T} V_i(I) < 1/n$ .

Since all queries until time  $T$  have been answered as if the valuations were uniform, it must be that  $1/n \leq V_i(I_i) = |I_i|$  for all  $i \in N$ . However, if for all  $i \in N$  we have: (i)  $I_i \subseteq A_i$ , (ii)  $|I_i| \geq 1/n$ , (iii)  $A_i \cap A_j = \emptyset$  for all  $j \neq i$ , and (iv)  $\bigcup_{i=1}^n A_i = [0, 1]$ , then it must be that  $|I_i| = |A_i| = 1/n$ , as well as  $V_i(A_i) = 1/n$ , for all  $i \in N$ . The HCR-2 property then implies that  $V_i(A_{N \setminus \{i,j\}}) = 1 - V_i(A_i) - V_i(A_j) \leq 1 - \frac{1}{n-1}$ , or  $V_i(A_j) \geq \frac{1}{n-1} - \frac{1}{n} = \frac{1}{n(n-1)}$  for all  $i, j \in N$ .

To meet this condition, it must be that for every agent  $i$  and piece  $A_j$ , there exists an active interval  $I \in \Pi_i^T$  such that (i)  $I \subseteq A_j$ , and (ii)  $V_i(I) \neq 0$ . Similarly to our earlier argument, if this is not the case, one can define  $V_i$  (consistently with  $\Pi_i^T$ ) such that  $V_i(A_j) = 0$  ( $< \frac{1}{n(n-1)}$ ). Concretely, we can pick  $V_i$  such that  $V_i(I \cap A_j) = 0$  and  $V_i(I \setminus A_j) = V_i(I)$  for all  $I \in \Pi_i^T$  and  $I \cap A_j \neq \emptyset$ .

Since allocations are pair-wise disjoint, we must then have at least  $n$  active intervals per agent by time  $T$ , i.e.  $|\Pi_i^T| \geq n$  for every agent  $i$ . Since  $|\Pi_i^0| = 1$ , Lemma 1 implies that  $|\Pi_i^T| \leq 2T + 1$ , and therefore  $2T + 1 \geq n$ , i.e. the algorithm makes at least  $\frac{n-1}{2}$  queries to every agent  $i \in N$ . Overall, the algorithm makes at least  $n \cdot \frac{n-1}{2} \in \Omega(n^2)$  queries overall.  $\square$

**Theorem 7.** Computing LCR-2 allocations requires an infinite number of queries, for all  $n \geq 3$ .

*Proof.* We prove the statement for all odd  $n$ ,  $n \geq 3$ ; the proof can be easily adjusted to even  $n$ .

Consider an instance where  $\lfloor \frac{n}{2} \rfloor$  agents — agents 1 through  $\lfloor \frac{n}{2} \rfloor$  — have the same valuation function  $V_1$ ,  $\lfloor \frac{n}{2} \rfloor$  agents — agents  $\lfloor \frac{n}{2} \rfloor + 1$  through  $n-1$  — have the same valuation function  $V_2$ , and the last agent, agent  $n$ , has valuation function  $V_n(x) = \frac{V_1(x) + V_2(x)}{2}$  for all  $x \subseteq [0, 1]$ .

Let  $A$  be a LCR-2 allocation. For every agent  $i = 1, \dots, \lfloor \frac{n}{2} \rfloor$ , let  $j_i = i + \lfloor \frac{n}{2} \rfloor$ . Since agent  $i$  has valuation  $V_1$ , we have that  $V_1([0, 1] \setminus (A_i \cup A_{j_i})) \leq 1 - 2/n$ , or simply,  $V_1(A_i \cup A_{j_i}) \geq 2/n$ , from the definition of LCR-2. It also holds that  $V_2(A_i \cup A_{j_i}) \geq 2/n$ . We have  $\bigcup_{i=1}^{\lfloor \frac{n}{2} \rfloor} (A_i \cup A_{j_i}) = [0, 1] \setminus A_n$ , and  $V_1([0, 1]) = V_2([0, 1]) = 1$ , therefore,  $V_1(A_n) = 1 - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} V_1(A_i \cup A_{j_i}) \leq 1 - \lfloor \frac{n}{2} \rfloor \frac{2}{n} = 1 - \frac{n-1}{2} \frac{2}{n} = \frac{1}{n}$ . Similarly,  $V_2(A_n) \leq \frac{1}{n}$ . Since LCR-2 implies proportionality,  $V_n(A_n) = \frac{V_1(A_n) + V_2(A_n)}{2} \geq \frac{1}{n}$ , i.e.,  $V_1(A_n) + V_2(A_n) \geq \frac{2}{n}$ . Therefore, it must be that  $V_1(A_n) = V_2(A_n) = \frac{1}{n}$ .

Since  $V_1(A_i \cup A_{j_i}) \geq \frac{2}{n}$ , for all  $i = 1, \dots, \lfloor \frac{n}{2} \rfloor$ , and  $\lfloor \frac{n}{2} \rfloor \frac{2}{n} = \frac{n-1}{2} \frac{2}{n} = 1 - \frac{1}{n}$  (which is exactly the value of  $[0, 1] \setminus A_n$ ), it must be that  $V_1(A_i \cup A_{j_i}) = \frac{2}{n}$ . Similarly, it must be that  $V_2(A_i \cup A_{j_i}) = \frac{2}{n}$ .

Therefore, we overall have that, there are  $\lfloor \frac{n}{2} \rfloor + 1$  pieces,  $A_n$  and the  $(A_i \cup A_{j_i})$ s, such that,  $V_1$  and  $V_2$  have the same value of  $\frac{1}{n}$  for the first piece, and the same value of  $\frac{2}{n}$  for all remaining pieces.

Given  $n$  valuation functions  $U_1, \dots, U_n$  and weights  $w_1, \dots, w_k$ , asking for a partition of the cake into  $k$  pieces  $I_1, \dots, I_k$  such that  $U_i(I_j) = w_j$ , is known as the *exact division*, or consensus splitting, problem. This problem is known to be impossible to solve with a bounded protocol, even for the case of two valuation functions and  $k = 2$  pieces with equal weights [RW98]. Since finding a LCR-2 allocation would give a solution to the exact division problem for two valuation functions and  $\lfloor \frac{n}{2} \rfloor + 1$  pieces, we can conclude that there is no bounded protocol for finding a LCR-2 allocation.

Note that for even  $n$ , the proof is near-identical: the only difference is that the  $n$ -th agent is not necessary (i.e., half the agents have valuation  $V_1$  and the other half have valuation  $V_2$ ), and using the same arguments we would conclude that  $V_1(A_i \cup A_{j_i}) = V_2(A_i \cup A_{j_i}) = \frac{2}{n}$ , for all  $i = 1, \dots, n/2$ .  $\square$

## C Picking the parameters of Algorithm 2

Consider the partition,  $B$ , created in the  $t^{\text{th}}$  iteration of line 4. Due to the normalization of  $R$  at each iteration,  $B$  will have the following property:

$$\left( \frac{1}{n+1} - \epsilon' \right)^t \leq V_i(B_j) \leq \left( \frac{1}{n+1} + \epsilon' \right)^t, \forall i, j \in N$$

Let  $S'$  describe the state of  $S$  prior to the final “Last Diminisher” phase of the algorithm (i.e. line 11).  $S'_i$  will be comprised of 1 piece from each  $\epsilon'$ -perfect partition,  $B$ , created. As discussed, each piece of  $S'$  needs to have value at least  $\frac{1}{n} - \tilde{\epsilon}$  but at most  $\frac{1}{n}$  for any agent.

$$V_i(S'_j) \leq \sum_{t=1}^d \left( \frac{1}{n+1} + \epsilon' \right)^t \leq \frac{1}{n}$$

$$V_i(S'_j) \geq \sum_{t=1}^d \left( \frac{1}{n+1} - \epsilon' \right)^t \geq \frac{1}{n} - \tilde{\epsilon}$$

Notice that by polynomial expansion, we can see that:

$$\left( \frac{1}{n+1} + \epsilon' \right)^t - \left( \frac{1}{n+1} \right)^t \geq \left( \frac{1}{n+1} \right)^t - \left( \frac{1}{n+1} - \epsilon' \right)^t$$

Therefore, if we define  $d$  such that  $\sum_{t=1}^d \left( \frac{1}{n+1} \right)^t \geq \frac{1}{n} - \frac{\tilde{\epsilon}}{2}$  (the average of the required upper and lower bounds of  $V_i(S'_j)$ ), then for increasing  $\epsilon'$ , the computed upper bound of  $V_i(S'_j)$  will exceed  $\frac{1}{n}$  before the computed lower bound dips below  $\frac{1}{n} - \tilde{\epsilon}$ . With this in mind, we define  $d$  as described:

$$\sum_{t=1}^d \left( \frac{1}{n+1} \right)^t = \frac{1 - \left( \frac{1}{n+1} \right)^{d+1}}{1 - \frac{1}{n+1}} \geq \frac{1}{n} - \frac{\tilde{\epsilon}}{2}$$

$$d = \left\lceil \frac{\ln \left( \frac{\tilde{\epsilon}n}{2} \right)}{\ln \left( \frac{1}{n+1} \right)} \right\rceil = \left\lceil \frac{\ln(2) + \ln(1/\tilde{\epsilon}) + \ln(1/n)}{\ln(n+1)} \right\rceil \in \Theta \left( \frac{\ln(1/\tilde{\epsilon})}{\ln(n)} \right).$$

Now, we define  $\epsilon'$  such that  $V_i(S'_j) \leq \frac{1}{n}$ , noting that if the required upper bound holds, then so will the required lower bound. For simplicity, we first loosen the upper bound:

$$\begin{aligned}
V_i(S'_j) &\leq \sum_{t=1}^d \left( \frac{1}{n+1} + \epsilon' \right)^t \\
&= \left( \frac{1}{n+1} + \epsilon' \right) \cdot \frac{1 - \left( \frac{1}{n+1} + \epsilon' \right)^d}{1 - \left( \frac{1}{n+1} + \epsilon' \right)} \\
&\leq \left( \frac{1}{n+1} + \epsilon' \right) \cdot \frac{1 - \left( \frac{1}{n+1} \right)^d}{1 - \left( \frac{1}{n+1} + \epsilon' \right)} \\
&\leq \left( \frac{1}{n+1} + \epsilon' \right) \cdot \frac{1 - \frac{1}{n+1} \cdot \frac{\tilde{\epsilon}n}{2}}{1 - \left( \frac{1}{n+1} + \epsilon' \right)}.
\end{aligned}$$

Enforcing that this last expression is at most  $1/n$  we have  $\frac{\left( \frac{1}{n+1} + \epsilon' \right)}{1 - \left( \frac{1}{n+1} + \epsilon' \right)} \leq \frac{1}{n \left( 1 - \frac{1}{n+1} \cdot \frac{\tilde{\epsilon}n}{2} \right)}$ , or

$$\epsilon' \leq \frac{1}{n \left( 1 - \frac{1}{n+1} \cdot \frac{\tilde{\epsilon}n}{2} \right) + 1} - \frac{1}{n+1} = \frac{\tilde{\epsilon}n^2}{2(n+1)^2 \left( n + 1 - \frac{\tilde{\epsilon}n^2}{2(n+1)} \right)}.$$

Considering that  $\frac{\tilde{\epsilon}n^2}{2(n+1)} > 0$ , we can tighten this constraint by decreasing the upper bound as so:

$$\epsilon' \leq \frac{\tilde{\epsilon}n^2}{2(n+1)^3} \in \Theta \left( \frac{\tilde{\epsilon}}{n} \right).$$

Thus, the aforementioned conditions will hold for some  $d \in \Theta \left( \frac{\ln(1/\tilde{\epsilon})}{\ln(n)} \right)$  and  $\epsilon' \in \Theta \left( \frac{\tilde{\epsilon}}{n} \right)$ .