Fairer than Fair: Sharp Bounds for Connected Super-Proportional Cake Cutting

ZSUZSANNA JANKÓ, Corvinus University of Budapest, and ELTE Centre for Economic and Regional Studies, Hungary
ATTILA JOÓ, University of Hamburg, Germany and Technion Institute of Technology, Israel
EREL SEGAL-HALEVI*, Department of Computer Science, Ariel University, Israel
SHEUNG MAN YUEN, National University of Singapore, Singapore

We investigate the problem of fairly dividing a divisible heterogeneous resource, also known as a cake, among a set of n agents who may have different entitlements. We characterize the existence of a connected super-proportional (also called strongly-proportional) allocation—one in which every agent receives a contiguous piece worth strictly more than their proportional share. The characterization is supplemented with an algorithm that determines its existence using $O(n \cdot 2^n)$ queries. We devise a simpler characterization for agents with strictly positive valuations and with equal entitlements, and present an algorithm to determine the existence of such an allocation using $O(n^2)$ queries. We provide matching lower bounds in the number of queries for both algorithms. When a connected super-proportional allocation exists, we show that it can also be computed using a similar number of queries. We also consider the problem of deciding the existence of a connected allocation of a cake in which each agent receives a piece worth a small fixed value more than their proportional share, and the problem of deciding the existence of a connected super-proportional allocation of a pie (a 1-dimensional circular cake).

JAIR Associate Editor: Umberto Grandi

JAIR Reference Format:
Zsuzsanna Jankó, Attila Joó, Erel Segal-Halevi, and Sheung Man Yuen. 2026. Fairer than Fair: Sharp Bounds for Connected Super-Proportional Cake Cutting. *Journal of Artificial Intelligence Research* 85, Article 39 (April 2026), 31 pages. DOI: 10.1613/jair.1.19057

1 Introduction

Consider a group of siblings who inherited a land estate and would like to divide it fairly among themselves. The simplest procedure for attaining a fair division is to sell the land and divide the proceeds equally; this procedure guarantees each sibling a proportional share of the total land value.

But in some cases, it is possible to give each sibling a much better deal. As an example, suppose that the land estate contains one part that is fertile and arable, and one part that is barren but has potential for coal mining. This land is to be divided between two siblings, one of whom is a farmer and the other a coal factory owner. If we give the former piece of land to the farmer and the latter piece of land to the coal factory owner, both siblings will feel that they receive more than half of the total land value. Our main question of interest is: when is such a superior allocation possible?

*Corresponding Author.

Authors' Contact Information: Zsuzsanna Jankó, ORCID: 0000-0002-6149-4257, zsuzsanna.janko@uni-corvinus.hu, Corvinus University of Budapest, and ELTE Centre for Economic and Regional Studies, Budapest, Hungary; Attila Joó, ORCID: 0000-0002-4121-2830, balamud21@gmail.com, University of Hamburg, Hamburg, Germany and Technion Institute of Technology, Haifa, Israel; Erel Segal-Halevi, ORCID: 0000-0002-7497-5834, erelsgl@gmail.com, Department of Computer Science, Ariel University, Ariel, Israel; Sheung Man Yuen, ORCID: 0000-0002-8621-5402, ysm@u.nus.edu, National University of Singapore, Singapore.

DOI: 10.1613/jair.1.19057

We study this question in the framework of *cake-cutting*. In this setting, there is a divisible resource called a *cake*, which can be cut into arbitrarily small pieces without losing its value. The cake is represented simply by an interval which can model a one-dimensional object, such as time. There are n agents, each of whom has a personal measure of value over the cake. The goal is to partition the cake into n pieces and allocate one piece per agent such that the agents feel that they receive a “fair share” according to some fairness notion.

A common fairness criterion—nowadays called *proportionality*—requires that each agent i receives a piece of cake that is worth, according to i's valuation, at least $1/n$ of the total cake value. In his seminal paper, Steinhaus (1948) described an algorithm, developed by his students Banach and Knaster, that finds a proportional allocation; moreover, this allocation is *connected*—each agent receives a single contiguous part of the cake. This algorithm is now called the *last diminisher* algorithm.[1]

But the guarantee of proportionality allows for the possibility that each agent receives a piece worth *exactly* $1/n$; when this is the case, there is little advantage in using a cake-cutting procedure over selling the land and giving $1/n$ to each partner. A stronger criterion, called *super-proportionality* (Lindner and Rothe, 2024) or *strong-proportionality* (Barbanel, 1996a, 2005) , requires that each agent i receives a piece of cake worth *strictly more* than $1/n$ of the total cake value from i's perspective. This raises the question of when such a super-proportional allocation exists.

Obviously, a super-proportional allocation does not exist when all the agents' valuations are identical, since if any agent receives more than $1/n$ of the cake, then some other agent must receive less than $1/n$ of the cake. Interestingly, in all other cases, a super-proportional allocation exists. Even when *two* agents have non-identical valuations, there exists an allocation in which *all* n agents receive more than $1/n$ of the total cake value from their perspectives (Dubins and Spanier, 1961; Rebman, 1979). Woodall (1986) presented an algorithm for finding such a super-proportional allocation, given a “witness set” — a subset of the cake to which two agents assign different values. Barbanel (1996a) generalized this algorithm to agents with unequal entitlements, and Jankó and Joó (2022) presented a simple algorithm for this generalized problem and extended it to infinitely many agents (these algorithms do not deal with the question of how to obtain such a witness set).

The problem with all these algorithms is that, in contrast to the last diminisher algorithm for proportional cake-cutting, they do not guarantee a *connected* allocation. Connectivity is an important practical consideration when allocating cakes; for example, if the cake is the availability of a meeting room by time and needs to be allocated to different teams throughout the day, then a two-hour slot is easier for a team to utilize than six disjoint twenty-minute slots. Indeed, connectivity is the most commonly studied constraint in cake-cutting literature (Suksompong, 2021; Stromquist, 1980; Su, 1999; Stromquist, 2008; Goldberg et al., 2020; Elkind et al., 2022; Cechlárová et al., 2013). Ignoring this constraint may present each agent instead with a “countable union of crumbs” (Stromquist, 1980). Thus, our main questions of interest are:

> *What are the necessary and sufficient conditions for the existence of a connected super-proportional cake allocation? What are the query complexities to determine these conditions?*

This paper answers this question by presenting characterizations and algorithms for deciding whether a connected super-proportional allocation exists, and computing one such allocation if it exists. Another advantage of our algorithms over existing ones is that they do not require a witness set; they require only the cake and the valuations of the agents.[2] We analyze the number of queries required by each of our algorithms, and prove that this number is asymptotically optimal.

[1] In the Last Diminisher algorithm, an agent who assigns a value of $1/n$ to the smallest left part of the cake gets that piece, after which the remaining agents continue the process with the remaining cake until every agent receives a piece.

[2] We are grateful to an anonymous JAIR referee for noting this advantage.

1.1 Our Results

The cake to be allocated, modeled by a unit interval $[0, 1]$, is to be divided among n agents who may have different entitlements for the cake, with the entitlements summing to 1. Each agent receives an interval of the cake that is disjoint from the other agents' intervals. Each agent has a valuation function on the intervals of the cake that is non-negative, finitely additive, and continuous with respect to length. In this regard, the value of a single point is zero to every agent, and we can assume without loss of generality that agents receive *closed* intervals of the cake, and that any two agents' pieces can possibly intersect at the endpoints of their respective intervals.[3] In order to access agents' valuations in the algorithms, we allow algorithms to make EVAL and (right-)MARK[4] queries of each agent as in the standard Robertson-Webb model (Robertson and Webb, 1998). More details of our model are provided in Section 2.

In Section 3, we consider *hungry* agents—those who have positive valuations for any part of the cake with positive length. For agents with equal entitlements, we show that a connected super-proportional allocation exists if and only if there are two agents with different r-marks for some $r \in \{1/n, 2/n, \ldots, (n-1)/n\}$, where an r-mark is a point that divides the cake into two such that the left part of the cake is worth r to that agent. This implies that the existence of such an allocation can be decided using $n(n-1)$ queries. The proof of sufficiency is constructive, so a connected super-proportional allocation can be computed using $O(n^2)$ queries if it exists. We also prove that any algorithm that decides whether a connected super-proportional allocation exists must make at least $n(n-1)$ queries, giving an exact worst-case bound.

For hungry agents with possibly unequal entitlements, we show a lower bound of $n \cdot 2^{n-1}$ on the number of queries to decide whether a connected super-proportional allocation exists. The lower bound holds for every vector of entitlements that are *generic* — no two subsets of agents have the same total entitlement (Definition 1).

This lower bound exactly matches the query bound of an algorithm we present later on in Section 4.

In Section 4, we consider agents who are not necessarily hungry. The characterization from Section 3 for hungry agents with equal entitlements does not work for non-hungry agents, which motivates us to find another characterization by considering permutations of agents. We show that a connected super-proportional allocation exists if and only if there exists a permutation of agents such that when the agents go in the order as prescribed by the permutation and make their rightmost marks worth their entitlements to each of them one after another, the mark made by the last agent does not reach the end of the cake. This result holds regardless of the agents' entitlements. While an algorithm to determine this condition requires $n \cdot n!$ queries, we show that this number can be reduced (by a multiplicative factor of $2^{\omega(n)}$) to $n \cdot 2^{n-1}$ via dynamic programming, and prove an asymptotically matching lower bound of $(n-1) \cdot (2^{n-2} - n + 3)$ even for agents with equal entitlements. Therefore, for agents who are not necessarily hungry, we also obtain a tight bound of $\Theta(n \cdot 2^n)$, whether the entitlements are equal or not. We also show that, when a connected super-proportional allocation exists, it can be computed using $O(n \cdot 2^n)$ queries. Table 1 summarizes our results from Sections 3 and 4.

In Section 5, we consider a stronger fairness notion where each agent i needs to receive a connected piece of cake that is worth more than some given z_i, for some $z_i > w_i$, where w_i is agent i's entitlement. We show that the number of queries needed to decide whether such an allocation exists is in $O(n \cdot 2^n)$. This expression is approximately tight even for hungry agents with equal entitlements, whenever z_i is in the range $(1/n, 2/n)$ for all $i \in N$. However, when $z_i > 1/2$ for all $i \in N$, the decision problem becomes solvable in $O(n^2)$ queries.

In Section 6, we consider a connected super-proportional allocation of a *pie* instead of a cake, and show that no finite algorithm can decide the existence of such an allocation even for hungry agents with equal entitlements, demonstrating the intractability of the problem in this new setting.

[3]This is often assumed in cake-cutting literature; see for instance Procaccia (2016).

[4]We choose *right*-mark instead of the usual *left*-mark for convenience. Our algorithms still work if only left-mark queries are available (together with eval). See Appendix A for a more detailed explanation.

Table 1. Number of queries required to decide the existence of a connected super-proportional cake-allocation for n agents. (*) Lower bound is proved only for generic entitlements; upper bound holds for any entitlements.

entitlements	hungry agents	general agents
equal	$LB = UB = n(n-1) \in \Theta(n^2)$ (Theorems 3.6 and 3.3)	$LB = (n-1) \cdot (2^{n-2} - n + 3)$, $UB = n \cdot 2^{n-1}$; $\Theta(2^n n)$ (Theorems 4.6 and 4.5)
unequal (*)	$LB = UB = n \cdot 2^{n-1} \in \Theta(2^n n)$ (Theorems 3.8 and 4.5)	$LB = UB = n \cdot 2^{n-1} \in \Theta(2^n n)$ (Theorems 3.8 and 4.5)

1.2 Further Related Work

Lindner and Rothe (2024) present an exposition of the main concepts in fair cake-cutting, including super-proportionality. They claim that every proportional cake-cutting algorithm can be modified to yield a super-proportional allocation if the agents' valuations are different, but the valuations have to differ in a way that is specific to the algorithm (for example, the Even-Paz algorithm can yield a super-proportional allocation for hungry agents if in the first round, the $n/2$-th and $n/2+1$-th half-marks are different. These algorithm-specific conditions are sufficient but not necessary for the existence of super-proportional allocations.

The basic fairness notion of *proportionality* is well-studied in cake-cutting literature. It is known that a connected proportional allocation always exists for agents with equal entitlements and such an allocation can be computed using $\Theta(n \log n)$ queries (Steinhaus, 1948; Even and Paz, 1984; Woeginger and Sgall, 2007). Cseh and Fleiner (2020) presented an algorithm that finds a possibly non-connected proportional allocation for agents with general entitlements—in particular, their algorithm uses a finite but *unbounded* number of queries when agents have irrational entitlements. In contrast, we show that a connected *super-proportional* allocation may not exist, but can be computed (if it exists) using $\Theta(n \cdot 2^n)$ queries, even for irrational entitlements. A number of works studied the number of *cuts* required for a proportional allocation, rather than the number of queries (Segal-Halevi, 2019; Crew et al., 2020).

Another common fairness notion is *envy-freeness (EF)*. Analogously to proportionality, it has stronger variants called *strong-EF*, *super-EF* and *hyper-EF*; we discuss them briefly in the Conclusion section (Section 7).

2 Preliminaries

Let the cake be denoted by $C = [0, 1]$. The cake is to be allocated to a set of agents denoted by $[n] := \{1, \ldots, n\}$. A *piece of cake* is a finite union of closed intervals of the cake. An *allocation* of C is a partition of C into n pairwise-disjoint[5] pieces of cake $(X_1, \ldots, X_n)$ such that $C = X_1 \sqcup \cdots \sqcup X_n$; X_i is the piece allocated to agent i. An allocation is *connected* if X_i is a single interval for each $i \in [n]$.

The preference of each agent i is represented by a *valuation function* V_i such that $V_i(X)$ is the value of the piece $X \subseteq C$ to agent i. Each valuation function V_i is defined on the algebra over C generated by all intervals of C, and is non-negative (that is, $V_i(X) \geq 0$ for all $X \subseteq C$ in the algebra), finitely additive (that is, $V_i(X \cup Y) = V_i(X) + V_i(Y)$ for all disjoint $X, Y \subseteq C$ in the algebra), and normalized to one (that is, $V_i(C) = 1$). We assume that $F_i(x) := V_i([0, x])$ is a continuous function of x, and hence $V_i(\{x\}) = 0$ for all $x \in C$. Therefore, F_i is a non-decreasing function on C with $F_i(0) = 0$, $F_i(1) = 1$, and $V_i([x, y]) = F_i(y) - F_i(x)$. An agent i is *hungry* if $V_i(X) > 0$ for all intervals $X \subseteq C$ with positive length; this is equivalent to the condition that F_i is strictly increasing.

[5]As mentioned in Section 1.1, two pieces of cake are also considered disjoint if their intersection is a subset of the endpoints of their respective intervals.

Each agent i has an *entitlement* $w_i > 0$ of the cake such that $\sum_{i \in [n]} w_i = 1$. Let $\mathbf{w}$ denote $(w_1, \dots, w_n)$. We say that agents have *equal entitlements* if $w_i = 1/n$ for all $i \in [n]$. For each subset $N \subseteq [n]$ of agents, define $w_N = \sum_{i \in N} w_i$. Note that $w_\emptyset = 0$ and $w_{[n]} = 1$.

Definition 1. We say that agents have *generic entitlements* if $w_N \neq w_{N'}$ for all distinct $N, N' \subseteq [n]$.

A *(cake-cutting) instance* consists of the set of agents, their valuation functions $(V_i)_{i \in [n]}$, and their entitlements $\mathbf{w}$.

Given an instance, an allocation $(X_1, \dots, X_n)$ is *proportional* (resp. *super-proportional*) if $V_i(X_i) \geq w_i$ (resp. $V_i(X_i) > w_i$) for all $i \in [n]$. For agents with equal entitlements, a proportional (resp. super-proportional) allocation requires every agent to receive a piece of cake with value at least (resp. greater than) $1/n$.

Algorithms can make eval and mark queries of each agent in the Robertson-Webb model. More specifically, for each agent $i \in [n]$, value $r \in [0, 1]$, and points $x, y \in C$ with $x \leq y$:

- $\textsc{Eval}_i(x, y)$ returns $V_i([x, y])$;
- $\textsc{Mark}_i(x, r)$ returns the *rightmost* (largest) point $z \in C$ such that $V_i([x, z]) = r$ if $V_i([x, 1]) \geq r$; [6] Otherwise ($V_i([x, 1]) < r$), $\textsc{Mark}_i(x, r)$ returns ∞.

For $i \in [n]$ and $r \in [0, 1]$, a point $x \in C$ is an *r-mark* of agent i if $V_i([0, x]) = r$. While the point returned by $\textsc{Mark}_i(0, r)$ is an r-mark of agent i, the converse is not true since $\textsc{Mark}_i(0, r)$ only returns the *rightmost* r-mark of agent i. However, when agent i is *hungry*, then the r-mark is unique, and the two notions coincide. Let $\mathcal{T}$ denote the set $\{1/n, 2/n, \dots, (n-1)/n\}$—we shall consider r-marks for $r \in \mathcal{T}$ in Section 3.1.

3 Hungry Agents

We begin with the simpler case where all agents are hungry. We first state a result which finds a connected super-proportional allocation of a cake for hungry agents using a small number of queries when given a connected *proportional* allocation in which one agent has a super-proportional piece. The proof proceeds by slightly moving the boundary between two adjacent agents' pieces such that an agent j who received exactly w_j eventually gets a slightly larger piece.

Lemma 3.1. *Let an instance with n hungry agents be given. Suppose that we are given a connected proportional allocation $(X_1, \dots, X_n)$ such that $V_i(X_i) > w_i$ for some $i \in [n]$. Then, there exists a connected super-proportional allocation, and such an allocation can be computed using $O(n)$ queries.*

Proof. First, we find the values of $V_j(X_j)$ for all $j \in [n]$. If $V_j(X_j) > w_j$ for all $j \in [n]$, then we are done. Otherwise, there exist two distinct agents $i, j \in [n]$ with neighboring pieces such that $V_i(X_i) > w_i$ and $V_j(X_j) = w_j$. By slightly moving the boundary between X_i and X_j, we can get a new allocation in which each of agents i and j receives a piece worth more than w_i and w_j respectively. To formally describe the process of moving the boundary, we consider two complementary cases.

Case 1: X_i is to the left of X_j. Denote $X_i = [z_1, z_2]$ and $X_j = [z_2, z_3]$. Let $y = \textsc{Mark}_i(z_1, w_i)$; note that $y \in (z_1, z_2)$ since $V_i(X_i) > w_i$. Let y^* be the midpoint of y and z_2. Adjust the two agents' pieces such that agent i now receives $[z_1, y^*]$ and agent j now receives $[y^*, z_3]$; see Figure 1 for an illustration.

Since $[z_1, y^*] \supsetneq [z_1, y]$ and the latter is worth w_i to hungry agent i, the new piece, $[z_1, y^*]$, is worth more than w_i to agent i. Likewise, since $[y^*, z_3] \supsetneq [z_2, z_3]$ and the latter is worth w_j to hungry agent j, the new piece, $[y^*, z_3]$, is worth more than w_j to agent j.

Case 2: X_i is to the right of X_j. Denote $X_j = [z_1, z_2]$ and $X_i = [z_2, z_3]$. Let $y = \textsc{Mark}_i(z_2, V_i(X_i) - w_i)$; note that $y \in (z_2, z_3)$ since $V_i(X_i) > w_i$. Let y^* be the midpoint of z_2 and y. Adjust the two agents' pieces such that agent j now receives $[z_1, y^*]$ and agent i now receives $[y^*, z_3]$.

[6] If $V_i([x, 1]) \geq r$, then a point z satisfying $V_i([x, z]) = r$ exists, and the rightmost such point also exists, due to the continuity of the valuations.

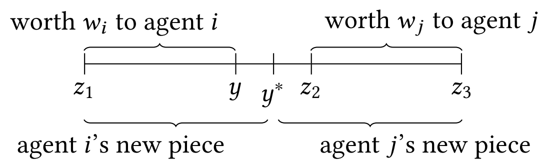

Fig. 1. Agent i's and j's new pieces in the proof of Lemma 3.1.

Since $[z_1, y^*] \supsetneq [z_1, z_2]$ and the latter is worth w_j to hungry agent j, the new piece, $[z_1, y^*]$, is worth more than w_j to agent j. Likewise, since $[y^*, z_3] \supsetneq [y, z_3]$ and the latter is worth w_i to hungry agent i (due to additivity, we have $V_i([y, z_3]) = V_i([z_2, z_3]) - V_i([z_2, y]) = w_i$), the new piece, $[y^*, z_3]$, is worth more than w_i to agent i.

In both Case 1 and Case 2, only agent i's and j's pieces change; all of the other agents' pieces do not change. All in all, one additional agent j receives more than w_j of the cake. Proceeding this way at most $n-1$ times yields a connected super-proportional allocation.

Finding the values of all $V_j(X_j)$ at the beginning requires n queries, while the adjustment of the boundaries between two agents' pieces requires a constant number of queries, so the total number of queries is in $O(n)$. □

We present the results separately for agents with equal entitlements and agents with possibly unequal entitlements. For n hungry agents with equal entitlements, we state in Section 3.1 a simple necessary and sufficient condition for the existence of a connected super-proportional allocation. We provide an asymptotically tight bound of $\Theta(n^2)$ for the number of queries needed by an algorithm to determine the existence of such an allocation, as well as to compute one such allocation if it exists. For agents with possibly unequal entitlements, we show in Section 3.2 that a lower bound number of queries needed to decide the existence of a connected super-proportional allocation is in $\Omega(n \cdot 2^n)$.

3.1 Equal Entitlements

Recall that $\mathcal{T} = \{1/n, 2/n, \ldots, (n-1)/n\}$. Our condition uses a particular set of r-marks: those with $r \in \mathcal{T}$.

Theorem 3.2. *Let an instance with n hungry agents with equal entitlements be given. Then, a connected super-proportional allocation exists if and only if there exist two distinct agents $i, j \in [n]$ and $r \in \mathcal{T}$ such that the r-mark of agent i is different from the r-mark of agent j.*

Proof. Since the agents are hungry, there is exactly one r-mark of agent i for each $r \in [0,1]$ and $i \in [n]$.

($\Rightarrow$) We prove the contraposition. Suppose that for each $r \in \mathcal{T}$, every agent has the same r-mark. Every agent also has the same 0-mark (at point 0) and the same 1-mark (at point 1). For each $t \in \{0, \ldots, n\}$, denote the common t/n-mark by z_t.

Consider now any connected allocation, which is represented by $n-1$ cuts on the cake. For each $t \in [n-1]$, denote the t-th cut from the left by x_t; also denote $x_0 = 0$ and $x_n = 1$. Each agent receives a piece $[x_{t-1}, x_t]$ for some $t \in [n]$, and every such piece is allocated to some agent.

Since $x_0 = z_0$ and $x_n = z_n$, there must be some $t \in [n]$ for which $x_{t-1} \geq z_{t-1}$ and $x_t \leq z_t$. This means that the piece $[x_{t-1}, x_t]$ is contained in the interval $[z_{t-1}, z_t]$. Let i denote the agent who receives the piece $[x_{t-1}, x_t]$. Then, agent i's value for her piece is

$$V_i([x_{t-1}, x_t]) \leq V_i([z_{t-1}, z_t]) = V_i([0, z_t]) - V_i([0, z_{t-1}]) = t/n - (t-1)/n = 1/n,$$

so the allocation is not super-proportional. This holds for any connected allocation; therefore, no connected super-proportional allocation exists.

($\Leftarrow$) Suppose that there exist two distinct agents $i, j \in [n]$ and $r \in \mathcal{T}$ such that the r-mark of agent i is different from the r-mark of agent j. We shall construct a connected super-proportional allocation by first constructing a connected *proportional* allocation such that at least one agent receives a piece with value more than $1/n$, then use Lemma 3.1 to construct a super-proportional one.

Let $t \in [n-1]$ be the integer such that $r = t/n$. Let i_L be an agent with the leftmost (smallest) r-mark among all the agents, and i_R be an agent with the rightmost (largest) r-mark among all the agents (if there are multiple agents with the same leftmost or rightmost r-mark, we can choose an agent arbitrarily in each case). Denote the leftmost r-mark by z_L and the rightmost r-mark by z_R. Note that $z_L < z_R$, since there are agents with different r-marks.

Since there are n agents, there are n r-marks (possibly some of them are equal) in the interval $[z_L, z_R]$. Let $x \in [z_L, z_R]$ be the t-th r-mark from the left. Then, there exists a partition of the agents into two subsets N_1 and N_2 such that

- $|N_1| = t$, and the r-mark of all agents in N_1 is at most x, and
- $|N_2| = n - t$, and the r-mark of all agents in N_2 is at least x.

Every agent in N_1 values $[0, x]$ at least r, and every agent in N_2 values $[x, 1]$ at least $1 - r$; see Figure 2 for an illustration.

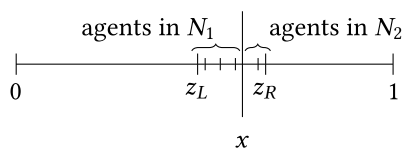

Fig. 2. The r-marks of all the agents in the proof of Theorem 3.2. The point x is at one of the r-marks and divides agents into N_1 and N_2.

Next, we consider any connected proportional cake-cutting algorithm as a black box (for example, last diminisher). We apply the algorithm on $[0, x]$ and N_1 such that every agent in N_1 receives a connected piece with value at least $1/t$ of her value of $[0, x]$, and apply the algorithm on $[x, 1]$ and N_2 such that every agent in N_2 receives a connected piece with value at least $1/(n-t)$ of her value of $[x, 1]$. We show that this allocation (of $C = [0, 1]$) is proportional. For an agent in N_1, since she values $[0, x]$ at least $r = t/n$, the piece she receives has value at least $(1/t)r = 1/n$. Likewise, for an agent in N_2, since she values $[x, 1]$ at least $1 - r = (n-t)/n$, the piece she receives has value at least $(1/(n-t))(1-r) = 1/n$.

Now, we show that agent i_L or i_R (or both) receives a piece with value strictly more than $1/n$. If $x = z_R$, then we claim that agent i_L receives such a piece. Since the r-mark of agent i_L is at $z_L < x$, we have $i_L \in N_1$. Since agent i_L is hungry, the piece $[0, x]$ is worth more than r to her, and so the piece she receives has value more than $(1/t)r = 1/n$. Otherwise, $x < z_R$, and a similar argument shows that agent i_R receives such a piece.

Having established a connected proportional allocation in which at least one agent receives more than $1/n$, we apply Lemma 3.1 to obtain a connected *super-proportional* allocation. □

It is interesting to compare the condition in Theorem 3.2 with the one for non-connected allocations. In both cases, a disagreement between *two* agents is sufficient for allocating *all* n agents more than their fair share. However, in the non-connected case, the disagreement can be in an r-mark for any $r \in (0, 1)$ (see the discussion in Section 1), whereas in the connected case, the disagreement should be in an r-mark for some $r \in \mathcal{T}$; the r-marks for other values of r are completely irrelevant.

It is clear from Theorem 3.2 that we can decide whether a connected super-proportional allocation exists for hungry agents with equal entitlements by checking the t/n-marks of all of the n agents for all $t \in [n-1]$. This is described in Algorithm 1. The number of queries used in the algorithm is at most $n(n-1)$.

Algorithm 1 Determining the existence of a connected super-proportional allocation for n hungry agents with equal entitlements.

```
for t = 1, ..., n − 1 do
    z_t ← Mark_1(0, t/n)                                  ▷ agent 1's t/n-mark
    for i = 2, ..., n do
        if Mark_i(0, t/n) ≠ z_t then return true
    end for
end for
return false
```

Theorem 3.3. *Algorithm 1 decides whether a connected super-proportional allocation exists for n hungry agents with equal entitlements using at most $n(n-1)$ queries.*

Next, we show a tight lower bound for the number of queries required to decide the existence of such an allocation for hungry agents. The idea behind the proof is that we must check the t/n-marks of all the agents and all $t \in [n-1]$; otherwise, we can craft two instances—one with the t/n-marks coinciding, and the other with some t/n-marks not coinciding—that are consistent with the information obtained by the algorithm and yet give opposite results.

For the following couple of proofs, $\mathbf{y}$ and $\mathbf{a}$ are row vectors, $\mathbf{x}$ and $\mathbf{b}$ are column vectors, A is a matrix. We recall a basic fact from linear algebra.

We say that a solvable linear equation system $A\mathbf{x} = \mathbf{b}$ *determines the expression* $\mathbf{a} \cdot \mathbf{x}$ if there exists a $\beta \in \mathbb{R}$ such that for every vector $\mathbf{x}$ with $A\mathbf{x} = \mathbf{b}$, we have $\mathbf{a} \cdot \mathbf{x} = \beta$. This means that if $\mathbf{x}$ is a solution to the equation-system $A\mathbf{x} = \mathbf{b}$, then the value of $\mathbf{a} \cdot \mathbf{x}$ is uniquely determined.

Lemma 3.4. *The solvable linear equation system $A\mathbf{x} = \mathbf{b}$ determines the expression $\mathbf{a} \cdot \mathbf{x}$ if and only if $\mathbf{a}$ is a linear combination of the rows of A.*

Furthermore, if $\mathbf{a} \cdot \mathbf{x}$ is not determined by $Ax = b$, then for every $\beta \in \mathbb{R}$, extending $A\mathbf{x} = \mathbf{b}$ with the new equation $\mathbf{a} \cdot \mathbf{x} = \beta$ results in a solvable system (that is, every real value is a possible value of $\mathbf{a} \cdot \mathbf{x}$).

Proof. If $\mathbf{a}$ is a linear combination of the rows of A, then there is a vector $\mathbf{y_a}$ with $\mathbf{y_a}A = \mathbf{a}$. But then for every $\mathbf{x}$ with $A\mathbf{x} = \mathbf{b}$ we have $\mathbf{a} \cdot \mathbf{x} = (\mathbf{y_a}A)\mathbf{x} = \mathbf{y_a}(A\mathbf{x}) = \mathbf{y_a} \cdot \mathbf{b}$, so the expression $\mathbf{a} \cdot \mathbf{x}$ is determined by $A\mathbf{x} = \mathbf{b}$ to $\beta = \mathbf{y_a} \cdot \mathbf{b}$.

Suppose that $\mathbf{a}$ is not a linear combination of the rows of A. For any $\beta \in \mathbb{R}$, define the linear system

$$\begin{bmatrix} A \\ \mathbf{a} \end{bmatrix} \cdot x = \begin{bmatrix} b \\ \beta \end{bmatrix} \tag{1}$$

It is sufficient to prove that the system (1) has a solution for every $\beta \in \mathbb{R}$. First, while the rows of A are linearly dependent, we remove rows of A (and the corresponding values of b); this yields a reduced system $A'x = b'$, with the same set of solutions ($Ax = b$ if and only if $A'x = b'$), such that the rows of A' are linearly independent. By the assumption on $\mathbf{a}$, the rows of the matrix $\begin{bmatrix} A' \\ \mathbf{a} \end{bmatrix}$ are linearly independent too. Hence, the system

$$\begin{bmatrix} A' \\ \mathbf{a} \end{bmatrix} \cdot x = \begin{bmatrix} b' \\ \beta \end{bmatrix} \tag{2}$$

has a solution x^* (it can be computed e.g. by Gaussian elimination). We can now bring back the removed rows from A, b, as they are linear combinations of the rows of A', b'; hence, x^* solves (1) too. □

Corollary 3.5. *A solvable linear equation system* $A\mathbf{x} = \mathbf{b}$ *cannot determine more than* $\mathrm{rank}(A)$ *linearly independent expressions.*

Theorem 3.6. *Any algorithm that decides whether a connected super-proportional allocation exists for n hungry agents with equal entitlements may require at least* $(n-1)$ *queries per agent; hence at least* $n(n-1)$ *queries overall.*

Proof. Let Alg be an algorithm that decides the existence of a connected super-proportional allocation for n hungry agents. We consider the case that all answers given to the algorithm are consistent with the instance where every agent's valuation is uniformly distributed over the cake. We claim that, in this case, Alg must ask each agent at least $n-1$ queries in order to answer correctly.

The proof is by contradiction. Suppose Alg asks some agent i fewer than $n-1$ queries. We prove that the answers are consistent both with a "yes" answer and with a "no" answer, and therefore the algorithm cannot know which answer is correct.

Each Mark_i or Eval_i query implies a constraint of the following form on agent i's valuation:

$$F_i(y) - F_i(x) = \beta$$

In case $x = 0$, the constraint becomes $F_i(y) = \beta$; in case $y = 1$, the constraint becomes $1 - F_i(x) = \beta$. If Alg asks i some $q < n-1$ queries, then there are q constraints. Every legal valuation function that satisfies all these q constraints is a possible valuation function for agent i. We call each value of $F_i(x)$ or $F_i(y)$ appearing in these constraints a *variable*; so we have q equations in at most $2q$ variables. We denote the set of these variables by Q. We denote the set of variables $F_i(t/n)$ for $t \in [n-1]$ by T.

By adding variables with 0 coefficients if needed, we may assume that all variables in T appear in the linear equation system, so we have a system of $q < n-1$ equations on the set $T \cup Q$ of at least $n-1$ variables. Let $A\mathbf{v} = \mathbf{b}$ be this linear equation system on $T \cup Q$. Note that this system is solvable: setting the value of every variable $F_i(x)$ to x provides a solution (corresponding to a uniform valuation). We denote this solution by v_0.

Given any solution v, we denote the value of v in the coordinate corresponding to variable $F_i(t/n)$ by $v[F_i(t/n)]$. In particular, for the solution corresponding to a uniform valuation we have $v_0[F_i(t/n)] = t/n$ for all $t \in [n-1]$.

The expressions $F_i(x)$ in T are linearly independent (they correspond to the standard basis vectors with respect to the variables in T). Since $q < n-1$, it follows from Corollary 3.5 that not all of them are determined by the system $A\mathbf{v} = b$. Let $F_i(t_0/n) \in T$ be a variable not determined by $A\mathbf{v} = b$. Then the system $A\mathbf{v} = b$ has a solution v_1 with $v_1[F_i(t_0/n)] \neq t_0/n$.

For every $\varepsilon \in \mathbb{R}$, denote $v_\varepsilon := (1-\varepsilon)v_0 + \varepsilon v_1$. Note that $A \cdot v_\varepsilon = (1-\varepsilon)A \cdot v_0 + \varepsilon \cdot A \cdot v_1 = (1-\varepsilon)b + \varepsilon b = b$, so v_ε is a solution to the system $Av = b$ too. If $\varepsilon > 0$, then $v_\varepsilon[F_i(t_0/n)] \neq t_0/n$. Furthermore, as $v_0[F_i(x)] < v_0[F_i(y)]$ for every variable $F_i(x), F_i(y) \in Q$ with $x < y$, for a sufficiently small ε it holds that $v_\varepsilon[F_i(x)] < v_\varepsilon[F_i(y)]$ for every variable $F_i(x), F_i(y) \in Q$ with $x < y$. Choose such a sufficiently small ε. Then there exists a valid valuation function F_i^* that takes the values prescribed by v_ε, i.e., $F_i^*(x) = v_\varepsilon[F_i(x)]$ whenever $F_i(x) \in Q$. Then $F_i^*(t_0/n) \neq t_0/n$. But $Av_\varepsilon = b$ ensures that all q queries are answered correctly according to the valuation F_i^*. Thus the answers of the queries are consistent with the scenario where $F_i(t_0/n) \neq t_0/n$ while all the agents other than i have uniform valuations, in which case the "yes" answer would be correct.

To sum: if Alg asks agent i fewer than $n-1$ queries, then i's answers are consistent with at least two different assignments to the variables $F_i(t/n)$:

(1) $F_i(t/n) = t/n$ for all $t \in [n-1]$;
(2) $F_i(t_0/n) \neq t_0/n$ for at least one $t_0 \in [n-1]$.

At the same time, the answers of the other agents are, by assumption, consistent with $F_j(t/n) = t/n$ for all $j \neq i$ and $t \in [n-1]$.

By Theorem 3.2, if agent i's true valuation satisfies assignment 1 (whereas $F_j(t/n) = t/n$ for all $j \neq i$ and $t \in [n-1]$), then no connected super-proportional allocation exists, so ALG must output "false". If agent i's true valuation satisfies assignment 2 (whereas $F_j(t/n) = t/n$ for all $j \neq i$ and $t \in [n-1]$), then a connected super-proportional allocation exists, so ALG must output "true". As ALG does not know which of these two options holds, it cannot know the correct answer. ◻

Theorems 3.3 and 3.6 show that the number of queries required to determine the existence of a connected super-proportional allocation for n hungry agents with equal entitlements is $n(n-1)$, which is in $\Theta(n^2)$. The same can be said for *computing* such an allocation—we can modify Algorithm 1 using the details in the proof of Theorem 3.2 to output a connected super-proportional allocation of the cake instead, if such an allocation exists. The number of queries required for computing the allocation (in addition to the queries for computing the t/n-marks) is in $O(n^2)$ too.

THEOREM 3.7. *The number of queries required to decide the existence of a connected super-proportional allocation for n hungry agents with equal entitlements, or to compute such an allocation if it exists, is in* $\Theta(n^2)$.

3.2 Possibly Unequal Entitlements

We now consider hungry agents who may not necessarily have equal entitlements. Since the entitlement of a subset of agents may not be a multiple of $1/n$, we cannot use the condition in Theorem 3.2 which uses r-marks for $r \in \mathcal{T}$. This requires us to devise a more general condition to determine the existence of a connected super-proportional allocation, which can be checked using $O(n \cdot 2^n)$ queries. Since the condition also works for non-hungry agents, we defer the discussion to Section 4.1 (see Theorems 4.4 and 4.5).

We now show an asymptotically-tight *lower bound* for the case when agents may have unequal entitlements. We show an even stronger result: for every vector of *generic entitlements* (Definition 1), the number of queries required to decide the existence of a connected super-proportional allocation is in $\Omega(n \cdot 2^n)$.

THEOREM 3.8. *Let* $\mathbf{w}$ *be any vector of generic entitlements. Then, any algorithm that decides whether a connected super-proportional allocation exists for n hungry agents with entitlements* $\mathbf{w}$ *may require at least* 2^{n-1} *queries per agent; hence at least* $n \cdot 2^{n-1}$ *queries overall.*

PROOF. Similarly to the proof of Theorem 3.6, let ALG be an algorithm that decides the existence of a connected super-proportional allocation for n hungry agents with entitlements $\mathbf{w}$. Suppose all answers given to the algorithm are consistent with each agent's valuation being uniformly distributed over the cake. These answers are consistent with a "no" answer. We prove below that, if ALG asks some agent i some $q < 2^{n-1}$ queries, the answers are also consistent with a "yes" answer, and therefore the algorithm cannot decide which answer is correct.

The answers of agent i yield q linear equations on variables $F_i(x)$ for some (at most $2q$ many) $x \in (0,1)$. As the entitlements are generic, we can order the 2^n different subsets of $[n]$ in strictly increasing order of their total entitlements, i.e., we label the nonempty proper subsets of $[n]$ by $N_1, \ldots, N_{2^n-2}$ such that $w_{N_1} < \cdots < w_{N_{2^n-2}}$. We define a set of $2^n - 2$ variables, which we denote by Y:

$$y_j := F_i(w_{N_j}) \qquad \forall j \in \{1, \ldots, 2^n - 2\}.$$

We call a set of assignments to these variables valid if the following inequalities are satisfied:

$$0 < y_1 < y_2 < \cdots < y_{2^n-3} < y_{2^n-2} < 1. \tag{3}$$

Note that $y_j := w_{N_j}$ for all j is a valid assignment (corresponding to agent i having a uniform valuation). Next, we define 2^{n-1} many expressions:

$$z_{N'} := F_i(w_{N' \cup \{i\}}) - F_i(w_{N'}) \qquad \forall N' \subseteq [n] \setminus \{i\}. \tag{4}$$

In other words, $z_{N'}$ represents the value for agent i of the piece between $w_{N'}$ and $w_{N' \cup \{i\}}$. Note that, as the entitlements are generic, the expressions $z_{N'}$ do not share variables with each other and therefore they are linearly independent. The rank of the matrix representing the equations is at most $q < 2^{n-1}$, so by Corollary 3.5, there is an expression z_{N^*} that is not determined by the system. As earlier, let v_0 be the solution where $v_0[F_i(w_{N'})] = w_{N'}$ for each $N' \subseteq [n]$. Let v_1 be another solution, where $v_1[F_i(w_{N^* \cup \{i\}})] - v_1[F_i(w_{N^*})] > w_i$ (which exists by Lemma 3.4). For every $\varepsilon \in \mathbb{R}$, denote $v_\varepsilon := (1-\varepsilon)v_0 + \varepsilon v_1$. Clearly, all the v_ε are solutions of the linear equation system. If ε is a small enough positive number, then $v_\varepsilon[y_j] < v_\varepsilon[y_{j+1}]$ for every $j < 2^n - 2$, whereas $v_\varepsilon[F_i(w_{N^* \cup \{i\}})] - v_\varepsilon[F_i(w_{N^*})] > w_i$. Choose such a sufficiently small ε. Then there exists a valid valuation function F_i^* that takes the values prescribed by v_ε, i.e., $F_i^*(x) = v_\varepsilon[F_i(x)]$ for each variable $F_i(x)$. The answers of the queries are consistent with the scenario where the valuation function of agent i is F_i^* while all the agents other than i have uniform valuations. We show that in this scenario the "yes" answer would be correct.

To do so, we construct a connected super-proportional allocation.

(1) The leftmost pieces are allocated to agents in N^* in an arbitrary order, where every agent $j \in N^*$ receives a piece of length w_j.
(2) Agent i receives the piece $[w_{N^*}, w_{N^*} + w_i]$.
(3) The remaining cake is allocated to the remaining agents such that every agent j receives a piece of length w_j.

Note that every agent $j \in N \setminus \{i\}$ receives a piece worth exactly w_j, since their valuation functions are uniform, whereas agent i receives a piece worth more than w_i because of $v_1[F_i(w_{N^* \cup \{i\}})] - v_1[F_i(w_{N^*})] > w_i$. By Lemma 3.1, a connected super-proportional allocation of the cake exists, and hence the answers are consistent with a "yes" answer. ◻

Using the results from Theorem 3.8 and from Theorem 4.5 later, we get a tight bound for hungry agents with generic entitlements.

Theorem 3.9. *The number of queries required to decide the existence of a connected super-proportional allocation for n hungry agents with generic entitlements, or to compute such an allocation if it exists, is in* $\Theta(n \cdot 2^n)$.

The lower bound in Theorem 3.8 is derived from the number of different values of w_{N_k}. In particular, a lower bound for number of queries is

$$\sum_{i=1}^{n} |\{w_N : N \neq \emptyset, N \subseteq [n], i \notin N\}|. \tag{5}$$

For *generic* entitlements, each term in the sum equals 2^{n-1}, so we get the lower bound of $n \cdot 2^{n-1}$ in Theorem 3.8. In contrast, for *equal* entitlements, each term in the sum equals $n-1$, so we get the lower bound of $n(n-1)$ in Theorem 3.6.

For entitlements that are neither generic nor equal, the resulting lower bound is between these two extremes. It is an interesting open question to find an algorithm with a query complexity matching the lower bound in (5) in these intermediate cases.

The main difficulty in extending our algorithm for equal entitlements (Algorithm 1) to unequal entitlements is due to the step in Theorem 3.2 where we used a black-box algorithm for *proportional* cake-cutting (such as last diminisher) to divide a part of the cake among the agents in N_1 and the other part among the agents in N_2. Such a black box algorithm does not exist for unequal entitlements, since a connected proportional allocation might not even exist for unequal entitlements in the first place.

Example 1 (A connected proportional allocation might not exist for agents with unequal entitlements)**.** Agent 1 assigns value 0.5 to each of $[0, 0.2]$ and $[0.8, 1]$, while agent 2 values $[0.4, 0.6]$ only, and the entitlements are 0.6 for agent 1 and 0.4 for agent 2. Any connected allocation that gives at least 0.6 to Agents 1 includes the middle interval, therefore Agent 2 receives 0. [7]

In Appendix A.3 we present an algorithm that decides whether there exists a connected proportional allocation for agents with unequal entitlements.

4 General Agents

We now consider the general case where agents need not be hungry. To motivate the need for more complex techniques in this section, we first explain the difficulties in extending the hungry-agents characterization to general agents.

Recall that the condition we developed in Theorem 3.2 involves checking for the coincidence of r-marks of all the agents for $r \in \mathcal{T}$. However, there are some difficulties in generalizing the condition for non-hungry agents, even for equal entitlements. The proof of Theorem 3.2 relies crucially on the fact that, for each r, every agent has exactly one r-mark. This may not be true for non-hungry agents. For each agent $i \in [n]$, $F_i(x) = V_i([0, x])$ is a continuous function with domain $C = [0, 1]$ and range $[0, 1]$. For each $r \in [0, 1]$, the set of r-marks of agent i is $F_i^{-1}(\{r\})$. Since $\{r\}$ is a closed set and F_i is continuous, $F_i^{-1}(\{r\})$ is a non-empty closed set.

If agent i is not necessarily hungry, then the fact that F_i is non-decreasing implies the set of r-marks of agent i is thus a non-empty closed *interval* (though possibly the singleton set $[x, x] = \{x\}$), which we call the agent's *r-interval*.

Another difficulty is that there may be different instances with exactly the same t/n-intervals but which give different results regarding the existence of such an allocation. We show this via the following two examples.

Example 2. Consider a cake-cutting instance for $n = 3$ agents with equal entitlements where the cake is made up of 11 homogeneous regions. The following table shows the agents' valuations for each region.

Alice	9	0	0	0	9	0	0	0	0	0	9
Bob	1	4	4	3	1	5	1	1	2	4	1
Chana	1	8	2	2	1	1	1	2	4	4	1

All agents value[8] the entire cake at 27, so the t/n-marks are at values 9 and 18. Alice has t/n-intervals—the two intervals of zeros. Bob and Chana each has two t/n-intervals that are single points, denoted by vertical lines—note that both Bob and Chana are hungry. We show that no connected super-proportional allocation exists.

Suppose by way of contradiction that a connected super-proportional allocation exists. Alice must receive a piece with value larger than 9, so her piece must touch the middle 9 as well as either the left 9 or the right 9. In the former case, the cake remaining for Bob and Chana is at most:

Bob	1	5	1	1	2	4	1
Chana	1	1	1	2	4	4	1

In the latter case, the remaining cake is at most:

[7] Segal-Halevi (2019) generalizes this example to any number of agents, proving that least $2n - 2$ cuts might be necessary for a proportional allocation with different entitlements.

[8] The value of the cake should technically be normalized to 1, but this can be done by simply dividing every value by 27. We use integers here and in all subsequent examples for simplicity.

Bob	1	4	4	3	1
Chana	1	8	2	2	1

In both cases, no matter how the remaining cake is divided between Bob and Chana, at least one agent gets a piece of cake with value at most 9, so no connected super-proportional allocation exists.

Example 3. Consider the following instance modified from Example 2.

Alice	9	0	0	0	9	0	0	0	0	0	9
Bob	1	4	4	3	1	5	5	1	1	1	1
Chana	1	8	2	2	1	1	1	2	4	4	1

The t/n-intervals of the agents are identical to those in Example 2. However, a connected super-proportional allocation exists, as the following table shows:

Alice	9	0	0	0	9						
Bob						5	5				
Chana								2	4	4	1

Examples 2 and 3 show that the condition for determining the existence of a connected super-proportional allocation cannot be extended trivially from the result for hungry agents. Instead, let us discuss the extent to which the results from Section 3.1 can be extended. We start with a necessary condition regarding the r-intervals for $r \in \mathcal{T}$. This condition is similar to that in Theorem 3.2.

Proposition 4.1 (Necessary condition)**.** *Let an instance with n agents with equal entitlements be given. If there exists a connected super-proportional allocation, then there exist two distinct agents $i, j \in [n]$ and an integer $t \in [n-1]$ such that the (t/n)-interval of agent i is disjoint[9] from the (t/n)-intervals of agent j.*

Proof. Let a connected super-proportional allocation be given, and let $\sigma : [n] \to [n]$ be the permutation such that agent $\sigma(k)$ receives the k-th piece from the left. Suppose by way of contradiction that for each $t \in [n-1]$, the (t/n)-intervals of every pair of agents have non-empty intersection. We show by backward induction that for each $k \in [n]$, every agent in $\{\sigma(1), \ldots, \sigma(k)\}$ assigns a total value of at most k/n to the leftmost k pieces.

The base case of $k = n$ is clear—every agent assigns a value of at most $n/n = 1$ to the whole cake. Suppose that the statement is true for $k+1$ for some $k \in [n-1]$; we shall prove the statement for k. Since agent $\sigma(k+1)$ assigns a value of at most $(k+1)/n$ to the union of the leftmost $k+1$ pieces, the left endpoint of her piece must be strictly to the left of k/n-interval in order for her to receive a piece worth more than $1/n$. Now, consider agent $\sigma(i)$ for $i \in [k]$. Since agent $\sigma(i)$'s k/n-interval intersects with agent $\sigma(k+1)$'s k/n-interval, the remaining cake after removing $\sigma(k+1)$'s piece is worth at most k/n to agent $\sigma(i)$. This proves the inductive statement.

Now, the statement for $k = 1$ states that agent $\sigma(1)$ receives a piece worth at most $1/n$. This contradicts the assumption that the allocation is super-proportional. □

Next, we provide a sufficient condition for a connected super-proportional allocation using intervals of r-marks for $r \in \mathcal{T}$.

Proposition 4.2 (Sufficient condition)**.** *Let an instance with n agents with equal entitlements be given. Suppose there exists an integer $t \in [n-1]$ and a partition of the agents into N_1 and N_2 with $|N_1| = t$ and $|N_2| = n - t$, such that all (t/n)-intervals of agents in N_1 are contained in $[0, x)$ and all (t/n)-intervals of agents in N_2 are contained in $(x, 1]$ for some $x \in [0, 1]$. In other words, there exists a point x that separates the t/n-intervals of some t agents from the t/n-intervals of the remaining $n - t$ agents. Then a connected super-proportional allocation exists.*

[9]Unlike for pieces of cake where "disjoint" means *finite* intersection, we revert to the standard definition of "disjoint" to mean *empty* intersection for r-intervals.

This condition differs from the necessary condition of Proposition 4.1 in that it requires the r-intervals of *two complementary subsets* of agents, rather than just two agents, to be disjoint. A special case of this condition is that the r-intervals of *all* n agents are pairwise-disjoint (rather than just two agents).

Proof of Proposition 4.2. Let $r := t/n$. Note that each of the t agents in N_1 values the cake $[0, x]$ more than t/n, and each of the $n - t$ agents in N_2 values the cake $[x, 1]$ more than $(n - t)/n$. We apply any connected proportional cake-cutting algorithm on each of $[0, x]$ on N_1 and $[x, 1]$ on N_2 such that every agent receives a connected piece worth more than $1/n$. This gives a connected super-proportional allocation. □

Propositions 4.1 and 4.2 coincide for $n = 2$ agents, yielding the following result.

Corollary 4.3. *Let an instance with two agents with equal entitlements be given. Then, a connected super-proportional allocation exists if and only if the intervals of* $1/2$*-marks of the two agents are disjoint.*

The two conditions do not coincide for $n \geq 3$ agents, however. Moreover, the two above examples show that the conditions cannot be strengthened to coincide. Specifically:

- Example 2 satisfies the necessary condition in Proposition 4.1; moreover, the r-intervals of Bob and Chana are disjoint for *all* $r \in \mathcal{T}$; but there is no connected super-proportional allocation.
- Example 3 violates the sufficient condition in Proposition 4.2, as Alice's r-intervals contain the r-intervals of Bob and Chana for all $r \in \mathcal{T}$; but a connected super-proportional allocation exists.

Another possibility for strengthening the necessary condition is to require that the r-interval of one agent to be disjoint from every other agent's r-interval for some $r \in \mathcal{T}$. However, the following example shows that this is still not sufficient.

Example 4. Consider the following instance for $n = 3$ agents.

Alice	4	2	2	1	3
Bob	4	0	2	2	4
Chana	4	0	2	2	4

All agents value the entire cake at 12, so the 2/3-marks are at value 8, denoted by vertical lines. Alice's 2/3-mark is disjoint from Bob's and Chana's 2/3-marks. However, no connected super-proportional allocation exists. If Alice receives the leftmost piece or the rightmost piece, then the remaining cake is worth at most 8 to both Bob and Chana, and both of them cannot simultaneously get a piece worth more than 4 each since they have identical valuations. If Alice receives the middle piece instead, then Bob and Chana must receive the leftmost and the rightmost piece in some order. However, the leftmost piece must touch the third region, the rightmost piece must touch the fourth region, and Alice's piece is confined to the third and fourth regions which is only worth at most 3 to her.

Examples 2 to 4 show that the existence of a connected super-proportional allocation cannot be determined based on t/n-marks alone. This inspires us to find another condition that characterizes the existence of a connected super-proportional allocation.

In Section 4.1, we generalize the condition from Theorem 3.2 for n non-hungry agents, regardless of whether they have equal entitlements or not. We show that this condition can be checked by an algorithm using $O(n \cdot 2^n)$ queries. The result in Theorem 3.8 says that the lower bound number of queries needed for an algorithm to determine the existence of a connected super-proportional allocation for n hungry agents with generic entitlements is $\Omega(n \cdot 2^n)$—we show in Section 4.2 that this lower bound also applies to (not necessarily hungry) agents with *equal entitlements*.

4.1 Upper Bound

Our condition requires agents to mark pieces of cake one after another in a certain order. We now describe this operation more precisely. Let $\sigma : [n] \to [n]$ be a permutation of agents, and let $x \in C$ and $r_1, \dots, r_n \in [0, 1]$. The agents proceed in the order $\sigma(1), \dots, \sigma(n)$. Agent $\sigma(1)$ starts first and makes a mark at $x_1 = \textsc{Mark}_{\sigma(1)}(x, r_{\sigma(1)})$, the rightmost point such that $[x, x_1]$ is worth $r_{\sigma(1)}$ to her. Then, agent $\sigma(2)$ continues from x_1, and makes a mark at $x_2 = \textsc{Mark}_{\sigma(2)}(x_1, r_{\sigma(2)})$, the rightmost point such that $[x_1, x_2]$ is worth $r_{\sigma(2)}$ to her. Each agent $\sigma(i)$ repeats the same process of making a mark at $x_i = \textsc{Mark}_{\sigma(i)}(x_{i-1}, r_{\sigma(i)})$ such that $[x_{i-1}, x_i]$ is the largest possible piece worth $r_{\sigma(i)}$ to her. We shall overload the definition of Mark and define[10] $\textsc{Mark}_\sigma(x, \mathbf{r})$ as the point x_n resulting from this sequential marking process, where $\mathbf{r} = (r_1, \dots, r_n)$. If $[x_{i-1}, 1]$ is worth less than $r_{\sigma(i)}$ to agent $\sigma(i)$ at any point, then $\textsc{Mark}_\sigma(x, \mathbf{r})$ is defined as ∞. This operation is described in Algorithm 2.[11] Note that each $\textsc{Mark}_\sigma(x, \mathbf{r})$ operation requires at most n ($\textsc{Mark}_i$) queries.

Algorithm 2 Computing $\textsc{Mark}_\sigma(x, \mathbf{r})$ for n agents. Note the subscript σ is a permutation, not an agent index.

$x_0 \leftarrow x$
for $i = 1, \dots, n$ **do**
  $x_i \leftarrow \textsc{Mark}_{\sigma(i)}(x_{i-1}, r_{\sigma(i)})$
  if $x_i = \infty$ **then return** ∞
end for
return x_n

Our necessary and sufficient condition for n (possibly non-hungry) agents requires us to check whether the point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is less than 1 for some permutation σ. The point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is determined when agents go in the order as prescribed by σ and make their rightmost marks worth their entitlements to each of them one after another. The idea behind the proof is that starting from the agent who receives the rightmost piece in σ and going leftwards, each agent is able to move the boundaries of her piece such that she receives a small piece of cake with positive value ϵ from the right and gives away a small piece of cake with value $\epsilon/2$ to the agent on the left, thereby increasing the value of her piece by a positive value $\epsilon/2$.

Theorem 4.4. *Let an instance with n agents be given. Then, a connected super-proportional allocation exists if and only if there exists a permutation $\sigma : [n] \to [n]$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.*

Proof. ($\Rightarrow$) Suppose that a connected super-proportional allocation exists. Let $\sigma : [n] \to [n]$ be the permutation such that agent $\sigma(k)$ receives the k-th piece from the left in this allocation, and let $y_0, y_1, \dots, y_n$ be the points such that agent $\sigma(k)$ receives the piece $[y_{k-1}, y_k]$ with $y_0 = 0$ and $y_n = 1$. We shall show that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. Let $x_0, x_1, \dots, x_n$ be the points as described by Algorithm 2 for $\textsc{Mark}_\sigma(0, \mathbf{w})$. We shall show by induction that $x_k < y_k$ for all $k \in [n]$; then, $\textsc{Mark}_\sigma(0, \mathbf{w}) = x_n < y_n = 1$ gives the desired conclusion.

For the base case $k = 1$, we have $x_1 = \textsc{Mark}_{\sigma(1)}(0, w_{\sigma(1)})$, so x_1 is a point for which $V_{\sigma(1)}([0, x_1]) = w_{\sigma(1)}$. Since agent $\sigma(1)$ receives a piece $[y_0, y_1] = [0, y_1]$ worth more than $w_{\sigma(1)}$, we must have $x_1 < y_1$. For the inductive case, assume that $x_k < y_k$ for some $k \in [n-1]$, and consider $k+1$. We have $x_{k+1} = \textsc{Mark}_{\sigma(k+1)}(x_k, w_{\sigma(k+1)}) \le \textsc{Mark}_{\sigma(k+1)}(y_k, w_{\sigma(k+1)})$ since $x_k < y_k$. Since agent $\sigma(k+1)$ receives a piece $[y_k, y_{k+1}]$ worth more than $w_{\sigma(k+1)}$, we have $\textsc{Mark}_{\sigma(k+1)}(y_k, w_{\sigma(k+1)}) < y_{k+1}$. Therefore, the result $x_{k+1} < y_{k+1}$ holds, proving the induction statement.

[10] The subscript of Mark here is a permutation σ, not an agent number.

[11] Algorithm 2 can be described as a moving-knife procedure: a knife moves continuously over the cake, until agent $\sigma(1)$ values the piece to the left of the knife at $1/n$; then the knife stops and cuts the cake and the piece to its left is given to agent $\sigma(1)$. The process then repeats with agent $\sigma(2)$, etc. It is similar to the well-known moving-knife procedure of Dubins and Spanier (1961), except that there the order of agents is not fixed in advance: the first agent who values the piece to the left of the knife at $1/n$ stops the knife and gets the piece.

($\Leftarrow$) Suppose that there exists a permutation $\sigma : [n] \to [n]$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. Let $x_0, x_1, \dots, x_n$ be the points as described by Algorithm 2 for $\textsc{Mark}_\sigma(0, \mathbf{w})$. Since x_k, which is $\textsc{Mark}_{\sigma(k)}(x_{k-1}, w_{\sigma(k)})$, is the rightmost point z such that $[x_{k-1}, z]$ is worth $w_{\sigma(k)}$ to agent $\sigma(k)$, the piece $[x_{k-1}, y_k]$ is worth more than $w_{\sigma(k)}$ to agent $\sigma(k)$ whenever $y_k > x_k$.

We shall define the points $y_1, \dots, y_n \in C$ in the reverse order such that $y_k > x_k$ for all $k \in [n]$. Define $y_n = 1 > \textsc{Mark}_\sigma(0, \mathbf{w}) = x_n$. Next, for each $k \in [n-1]$, assume that y_{k+1} is defined such that $y_{k+1} > x_{k+1}$. Since $[x_k, y_{k+1}]$ is worth more than $w_{\sigma(k+1)}$ to agent $\sigma(k+1)$, it must be worth $w_{\sigma(k+1)} + \epsilon_{k+1}$ to agent $\sigma(k+1)$ for some $\epsilon_{k+1} > 0$. Define $y_k = \textsc{Mark}_{\sigma(k+1)}(x_k, \epsilon_{k+1}/2)$. Then, we have $y_k > x_k$. This completes the definition of $y_1, \dots, y_n$.

Let $y_0 = x_0 = 0$. We shall show that the allocation with the cut points at $y_0, \dots, y_n$ such that $[y_{k-1}, y_k]$ is allocated to agent $\sigma(k)$ for $k \in [n]$ is super-proportional. Agent $\sigma(1)$ receives $[y_0, y_1] = [x_0, y_1]$ which is worth more than $w_{\sigma(1)}$ to her. For $k \in \{2, \dots, n\}$, since $[x_{k-1}, y_k]$ is worth $w_{\sigma(k)} + \epsilon_k$ and $[x_{k-1}, y_{k-1}]$ is worth $\epsilon_k/2$ to agent $\sigma(k)$, the piece $[y_{k-1}, y_k]$ is worth $(w_{\sigma(k)} + \epsilon_k) - \epsilon_k/2 > w_{\sigma(k)}$ to agent $\sigma(k)$. This completes the proof. □

The condition in Theorem 4.4 reduces to the condition in Theorem 3.2 for hungry agents with equal entitlements, that is, when $\mathbf{w} = (1/n, \dots, 1/n)$. In particular, when every agent has the same r-mark for each $r \in \mathcal{T}$, then each of the n marks made in the $\textsc{Mark}_\sigma(0, \mathbf{w})$ operation coincides at some $x_i \in \mathcal{T} \cup \{1\}$ for every permutation, and so $\textsc{Mark}_\sigma(0, \mathbf{w}) = 1$ for all σ. This corresponds to the case where no connected super-proportional allocation exists.

The analysis in Theorem 4.4 relies crucially on the fact that the $\textsc{Mark}_i$ queries return the *rightmost* points. If the *leftmost* points are returned instead, then the condition does not work—this can be seen from Example 2 where Chana, Alice, and Bob could (left-)mark their respective $1/n$ piece of the cake one after another in this order and still have a positive-valued cake left, but no connected super-proportional allocation exists as we demonstrated in Example 2.

Alice		0 0 9		0 9
Bob			5 1 1 2	4 1
Chana	1 8			4 1

We can determine whether the condition in Theorem 4.4 holds by checking each permutation σ to see whether the point $\textsc{Mark}_\sigma(0, \mathbf{w})$ is less than 1 for some σ. Since there are $n!$ possible permutations of $[n]$ and each $\textsc{Mark}_\sigma$ operation requires at most n queries, the total number of queries required in the algorithm is at most $n \cdot n!$.

However, we can reduce the number of queries to $n \cdot 2^{n-1}$ by dynamic programming. Our approach is similar to the method used in Aumann et al. (2012)—in their work, they iteratively find a value w such that there exists a connected allocation where every agent receives *at least* w, while here we require every agent i to receive a connected piece with value *strictly more* than w_i.

We now describe our algorithm. For every subset $N \subseteq [n]$, our algorithm caches the *best* mark b_N obtained by the subset of agents. The best mark b_N is the leftmost point possible over all permutations of the agents in N when the agents go in the order as prescribed by the permutation and make their rightmost marks worth their entitlements to each of them one after another. The algorithm aims to compute this point for every N.

The best mark for the empty set of agents is initialized as $b_\varnothing = 0$. Thereafter, for every $k \in [n]$, we assume that the best mark for every subset of $k-1$ agents is calculated earlier and cached. We now need to find b_N for every subset $N \subseteq [n]$ with k agents. The last agent to make the best mark for N could be any of the agents $i \in N$. Therefore, for each $i \in N$, we retrieve the best mark for $N \setminus \{i\}$, which is $b_{N \setminus \{i\}}$ and has been cached earlier, and let agent i make the rightmost mark such that the cake starting from $b_{N \setminus \{i\}}$ is worth w_i to her. By iterating through all $i \in N$, we find the leftmost such point and cache this point as b_N. When $k = n$, we obtain $b_{[n]}$, which is the best $\textsc{Mark}_\sigma(0, \mathbf{w})$ over all permutations σ. Therefore, the algorithm returns "true" if $b_{[n]} < 1$, and "false" otherwise. This implementation reduces the number of queries by a multiplicative factor of $2^{\omega(n)}$.

This algorithm is described in Algorithm 3. The correctness of the algorithm relies on the statement in Theorem 4.4 and the fact that $b_{[n]}$ in the algorithm is less than 1 if and only if there exists a permutation $\sigma : [n] \to [n]$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.

Algorithm 3 Determining the existence of a connected super-proportional allocation for n agents with possibly different entitlements.

$b_\emptyset \leftarrow 0$
for $k = 1, \ldots, n$ **do**
  for each subset $N \subseteq [n]$ with $|N| = k$ **do**
    $b_N \leftarrow \infty$
    for each agent $i \in N$ **do**
      $y \leftarrow \textsc{Mark}_i(b_{N \setminus \{i\}}, w_i)$
      if $y < b_N$ **then** $b_N \leftarrow y$ ▷ this finds the "best" b_N
    end for
  end for
end for
if $b_{[n]} < 1$ **then return** true **else return** false

Theorem 4.5. *For n agents with any entitlement vector $\mathbf{w}$, Algorithm 3 decides whether a connected super-proportional allocation exists using at most $n \cdot 2^{n-1}$ queries.*

Proof. To show that Algorithm 3 is correct, it suffices to show that $b_{[n]}$ in the algorithm is less than 1 if and only if there exists a permutation $\sigma : [n] \to [n]$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$, by Theorem 4.4.

($\Rightarrow$) If $b_{[n]}$ in the algorithm is less than 1, then $b_{[n]}$ is contributed by some agent $i_n \in [n]$ making the rightmost w_{i_n}-mark after $b_{[n]\setminus\{i_n\}}$. Let $\sigma(n) = i_n$. We then consider the agent i_{n-1} contributing the rightmost $w_{i_{n-1}}$-mark for $b_{[n]\setminus\{i_n\}}$, and so on. Repeat the procedure $n-1$ times to obtain the identities of the agents $\sigma(n-1), \ldots, \sigma(1)$. Then, σ is the desired permutation.

($\Leftarrow$) Suppose there exists a permutation $\sigma : [n] \to [n]$ such that $\textsc{Mark}_\sigma(0, \mathbf{w}) < 1$. For each $k \in [n]$, let $N_k = \{\sigma(1), \ldots, \sigma(k)\}$, and let x_k^σ be the mark where agents $\sigma(1), \ldots, \sigma(k)$ make their rightmost mark worth their entitlements to each of them one after another in this order. We prove by induction on k that $b_{N_k} \leq x_k^\sigma$. The base case of $k = 1$ is clear, as the two quantities are equal. Assume that the inequality is true for $k \in [n-1]$; we shall prove the result for $k+1$. The point $b_{N_{k+1}}$ is the smallest point over all permutations where agents $\sigma(1), \ldots, \sigma(k+1)$ make their rightmost $1/n$-mark one after another in some order. In particular, x_{k+1}^σ is one of these points under consideration. Therefore, we must have $b_{N_{k+1}} \leq x_{k+1}^\sigma$, proving the induction statement. Then, we have $b_{[n]} = b_{N_n} \leq x_n^\sigma = \textsc{Mark}_\sigma(0, \mathbf{w}) < 1$.

Next, we show that the number of queries made by Algorithm 3 is at most $n \cdot 2^{n-1}$. Let $k \in [n]$ be given. There are $\binom{n}{k}$ subsets N with cardinality k, and for each N, each of the $|N| = k$ agents makes a mark query. This means that $k\binom{n}{k}$ queries are made. Hence, the total number of queries is $\sum_{k=1}^{n} k\binom{n}{k} = n \cdot 2^{n-1}$ by a combinatorial identity. □

4.2 Lower Bound

Theorem 3.8 provides a lower bound for hungry agents with unequal entitlements; we shall now prove a similar lower bound for general agents with equal entitlements.

At a high level, the technique used is similar to that in the proofs of Theorems 3.6 and 3.8: we use an adversarial argument where we construct an instance with agents having uniform valuations on the cake such that no super-proportional allocation exists, but tweak the valuations slightly depending on the queries made. However, the details from the proof of Theorem 3.6 cannot be used directly since the existence of a connected super-proportional allocation is not solely dependent on the r-marks for $r \in \mathcal{T}$ for non-hungry agents (see the discussion at the beginning of Section 4), and the details from the proof of Theorem 3.8 cannot be used directly since Theorem 3.8 requires the entitlements to be generic.

Instead, we construct the following instance with $n \geq 3$ agents. The cake is divided into $2n-1$ parts. The odd parts (that is, the 1st, 3rd, …, $(2n-1)$-th parts) are non-valuable to agents 1 to $n-1$, and worth $1/n$ each to agent n. The even parts (that is, the 2nd, 4th, …, $(2n-2)$-th parts) are valuable to agents 1 to $n-1$, and non-valuable to agent n. For $i \in [n-1]$, agent i's first $n-2$ valuable parts (that is, the 2nd, 4th, …, $(2n-4)$-th parts) are worth $a_i/(n-2)$ each to agent i for some carefully selected a_i, and the last valuable part (that is, the $(2n-2)$-th part) is worth $1-a_i$ to agent i. See Figure 3 for an illustration.

Agent 1	0	$a_1/(n-2)$		0	$a_1/(n-2)$	0	$1-a_1$	0
⋮	⋮	⋮	…	⋮	⋮	⋮	⋮	⋮
Agent $n-1$	0	$a_{n-1}/(n-2)$	(total: $n-2$	0	$a_{n-1}/(n-2)$	0	$1-a_{n-1}$	0
Agent n	$1/n$	0	identical copies)	$1/n$	0	$1/n$	0	$1/n$

Fig. 3. Construction of the cake used in the proof of Theorem 4.6.

Consider a connected super-proportional allocation with equal entitlements. Agent n's piece has to include pieces from at least two consecutive odd parts in order for her value to be greater than $1/n$. By a clever choice of a_i for $i \in [n-1]$, we force these two odd parts to be the *rightmost* odd parts. This leaves the remaining $2n-4$ parts for agents 1 to $n-1$. Removing all the non-valuable parts for these agents, the remaining valuable parts of the cake are worth a_i to agent $i \in [n-1]$. Divide all valuations and entitlements by a_i for each $i \in [n-1]$. Then, this is equivalent to a cake with value 1 to every agent such that each agent's entitlement is $w'_i = 1/na_i$. If we select the a_i's carefully such that $\sum_{i\in[n-1]} w'_i = 1$ and the entitlements w'_i's are *generic*, then we can invoke Theorem 3.8 to show that the lower bound number of queries is in $\Omega(n \cdot 2^n)$.

Theorem 4.6. *Any algorithm that decides whether a connected super-proportional allocation exists for n agents with equal entitlements requires $\Omega(n \cdot 2^n)$ queries.*

Proof. Let M be a sufficiently large constant (particularly, $M \geq 2^n n^2$), and for each $i \in [n-1]$, define $w'_i = \frac{M+2^{i-1}}{(n-1)M+2^{n-1}-1}$ and $a_i = \frac{1}{nw'_i}$. Note that $\sum_{i\in[n-1]} w'_i = 1$.

Consider a cake with $2n-1$ parts as illustrated in Figure 3. We first show that the valuations are valid. For each $i \in [n-1]$, w'_i is positive, which means that a_i and hence $a_i/(n-2)$ are positive. We have $1-a_i = \frac{M-2^{n-1}+2^{i-1}n+1}{n(M+2^{i-1})}$, which is positive since $M > 2^{n-1}$. Moreover, the sum of all valuations is $(n-2) \cdot a_i/(n-2) + (1-a_i) = 1$. So the valuations are valid.

We show that any algorithm that makes fewer than $(n-1) \cdot (2^{n-2} - n + 3)$ queries may not be able to decide whether a connected super-proportional allocation exists.

We assume that the answer to every query made by the algorithm is consistent with the instance where the valuation of each agent is uniformly distributed within each of their valuable parts, which are the even parts for agents $i \in [n-1]$ and the odd parts for agent n. We show that these answers are compatible both with a "no" outcome and with a "yes" outcome; therefore, the algorithm cannot decide the correct answer.

Proof that answers are compatible with "no". Consider the instance where the valuation of each agent is uniformly distributed within each of their valuable parts. We show that a connected super-proportional allocation cannot exist in this instance.

Suppose on the contrary that a connected super-proportional allocation exists. Recall that agents have equal entitlements, which means that every agent receives a piece worth more than $1/n$. Agent n's piece has to include pieces from at least two consecutive odd parts in order for her value to be greater than $1/n$, which means that agent n's piece has to contain at least one of the valuable parts of agent $i \in [n-1]$ completely.

We now show that for each $i \in [n-1]$, we have $a_i/(n-2) > 1 - a_i$. We have

$$\frac{a_i}{n-2} - (1 - a_i) = \frac{M - 2^{i-1}n^2 + n(2^{n-1} + 2^i - 1) + 1}{(n-2)n(M + 2^{i-1})} > 0,$$

where the inequality holds because $M > 2^{i-1}n^2$. This shows that $a_i/(n-2) > 1 - a_i$, which implies that each of the left valuable parts is worth more than the rightmost valuable part for agent $i \in [n-1]$.

For $i \in [n-1]$, since $a_i/(n-2) > 1 - a_i$, no matter which valuable part(s) of agent i is given to agent n, each of the remaining $\le n-2$ valuable parts is worth at most $a_i/(n-2)$ to agent i. Since the valuations are uniformly distributed within each valuable part, the total overlap between i's piece and the valuable parts should be larger than a fraction $(1/n) \div (a_i/(n-2)) = (n-2)w'_i$ of a valuable part.

Therefore, the total overlap between the pieces of agents $1, \ldots, n-1$ and the valuable parts is larger than

$$\sum_{i=1}^{n-1}(n-2)w'_i = (n-2)\sum_{i=1}^{n-1} w'_i = n-2$$

valuable parts. This is not possible, as there are only $n-2$ valuable parts. Hence, no connected super-proportional allocation exists.

Proof that answers are compatible with "yes". We show by construction that the information provided to the algorithm is also consistent with an instance with a connected super-proportional allocation.

Let agent n receive the rightmost two consecutive odd parts, so that agent n receives more than $1/n$. This leaves the remaining $2n-4$ parts for agents 1 to $n-1$. Removing all the non-valuable parts for all agents $i \in [n-1]$, the remaining valuable parts of the cake are worth a_i to agent i. Divide all valuations and entitlements by a_i for each $i \in [n-1]$—note that this does not change the existence of a connected super-proportional allocation. Then, this is equivalent to a cake with value 1 to every agent in $[n-1]$ such that agent i's entitlement is $1/na_i = w'_i$. Note that $\sum_{i \in [n-1]} w'_i = 1$, so we have reduced the problem to finding a connected super-proportional allocation on a reduced instance with $n-1$ hungry agents such that agent $i \in [n-1]$ has an entitlement of w'_i.

We claim that the entitlements are generic. To see this, let $N, N' \subseteq [n-1]$ such that $\sum_{i \in N} w'_i = \sum_{i \in N'} w'_i$. Since the denominators of the w'_i's are equal to each other, we have $\sum_{i \in N}(M + 2^{i-1}) = \sum_{i \in N'}(M + 2^{i-1})$. Since M is larger than $\sum_{i \in [n-1]} 2^{i-1}$, we must have $|N| = |N'|$, which implies that $\sum_{i \in N} 2^{i-1} = \sum_{i \in N'} 2^{i-1}$. The only way this is possible is when $N = N'$, which proves that the entitlements are generic. In this modified instance, we have $n-1$ hungry agents with generic entitlements.

Recall that the algorithm makes fewer than $(n-1) \cdot (2^{n-2} - n + 3)$ queries. Therefore, by the pigeonhole principle, there exists some agent $i \in [n-1]$ who is asked some $q < 2^{n-2} - n + 3$ queries.

We now prove that there exists an instance consistent with the information provided by the queries that admits a connected super-proportional allocation. The proof is similar to the one of Theorem 3.8 except that, in addition to the q equations implied by the agents' answers to the queries, there are $n-3$ additional equations implied by the structure of the instance: for every $t \in [n-3]$, we have in the reduced instance $F_i(t/(n-2)) = t/(n-2)$. All in all, there are $q + (n-3) < 2^{n-2} = 2^{(n-1)-1}$ equations on values of F_i, and the number of agents in the reduced instance is $n - 1$. Hence, the arguments in the proof of Theorem 3.8 imply that, for some valuation of i consistent with the q answers given to the algorithm, a connected super-proportional allocation exists. □

The upper bound from Theorem 4.5 and the lower bound from Theorem 4.6 imply that the number of queries required to determine the existence of a connected super-proportional allocation is in $\Theta(n \cdot 2^n)$, even for agents with equal entitlements. The same tight bound also holds for *computing* such an allocation if it exists. The upper bound can be shown by modifying Algorithm 3 slightly by following the details in the second half of the proof of Theorem 4.4. The lower bound follows immediately from the lower bound on the decision problem, as if a faster algorithm existed, we could use it to decide existence.

Theorem 4.7. *For non-hungry agents, when the entitlements are either equal or generic, the number of queries required to decide the existence of a connected super-proportional allocation for n agents, or to compute such an allocation if it exists, is in $\Theta(n \cdot 2^n)$.*

5 Stronger than Super-proportional

We have so far only considered allocations which are *super-proportional*—agents receive pieces with value strictly more than their entitlements. Strong proportionality does not guarantee that agents receive pieces beyond just a small crumb more than their proportional piece. It would indeed be useful if we can guarantee that agents receive a fixed positive amount more than their entitlements. This motivates us to consider an even stronger fairness notion: given target values $z_i \geq w_i$ for all $i \in [n]$, can each agent i receive a piece with value more than z_i?

It is easy to adapt Algorithm 3 to this setting by replacing w_i in Step 6 of the algorithm with z_i—this gives an upper bound number of queries to determine the existence of such an allocation, or to compute such an allocation if it exists.

Theorem 5.1. *Let $\mathbf{w}$ be any vector of entitlements, and let $\mathbf{z}$ be any vector with $z_i \geq w_i$ for all $i \in [n]$. Then, there exists an algorithm that decides whether a connected allocation exists for n agents in which each agent i receives a piece with value more than z_i using at most $n \cdot 2^{n-1}$ queries.*

An interesting special case of Theorem 5.1 is deciding whether there exists an allocation with a high *egalitarian value*. The egalitarian value of an allocation is the smallest value of an agent's piece in an allocation. In particular, an allocation is proportional iff its egalitarian value is at least $1/n$, and super-proportional iff its egalitarian value is larger than $1/n$. We are interested in algorithms that find a connected allocation in which the egalitarian value is larger than some threshold, iff such an allocation exists.

Definition 2. Given a real number $g \in (0, 1)$, a *g-egalitarian allocation* is a cake allocation in which the value of every agent is larger than g.

Theorem 5.1 implies:

Corollary 5.2. *For any $n \geq 2$ and any $g \geq 1/n$, there is an algorithm that decides whether a connected g-egalitarian allocation exists using at most $n \cdot 2^{n-1}$ queries.*

We now show an exponential lower bound for uncountably many values of g. The proof is similar to that of Theorem 4.6. We will use a technical lemma on connected allocations of cakes with a special structure.

Lemma 5.3. *Consider a cake divided into two parts: left and right. Suppose each agent i assigns value u_i to the left part and $1 - u_i$ to the right part (the valuation is distributed uniformly within each part).*

If there exists a connected g-egalitarian allocation, then there exists a connected g-egalitarian allocation in which the agents are ordered from left to right by descending order of u_i, that is, if i is to the left of j then $u_i \geq u_j$.

Proof. Without loss of generality, we assume that the cake is $[0, 2]$, the left part is $[0, 1]$ and the right part is $[1, 2]$.

Let $(X_1, \ldots, X_n)$ be a connected g-egalitarian allocation, and suppose that there are two adjacent pieces X_i, X_j such that X_i is to the left of X_j but $u_i < u_j$. Denote $X_i = [x, y]$ and $X_j = [y, z]$. We construct an allocation X' which is the same as X except that $X'_j = [x, y']$ and $X'_i = [y', z]$ (that is, agents i and j swap their positions and move the boundary between them, but no other agents are affected). We show that $V_i(X'_i) \geq V_i(X_i)$ and $V_j(X'_j) \geq V_j(X_j)$, so X' is still a g-egalitarian allocation.

If $z \leq 1$ (so both pieces are contained in the left part) or $x \geq 1$ (so both pieces are contained in the right part), then we can simply take $y' = x + (z - y)$. Now each of i, j gets a piece at the same part and of the same length as in X, so their values remains the same.

Otherwise, $x < 1 < z$. As y' moves from left to right, the value of X'_j increases and the value of X'_i decreases. The condition $V_j(X'_j) \geq V_j(X_j)$ yields a lower bound on y', and the condition $V_i(X'_i) \geq V_j(X_j)$ yields an upper bound on y'. It is clear that each of these bounds can be satisfied on its own; we now show that both bounds can be satisfied simultaneously by showing that the lower bound is smaller than the upper bound. We consider two cases:

Case 1: $y \leq 1$. Then

$$\begin{aligned} V_i(X_i) &= (y - x)u_i \\ V_j(X_j) &= (1 - y)u_j + (z - 1)(1 - u_j) \\ &= (2 - y - z)u_j + (z - 1). \end{aligned}$$

If $V_j([x, 1]) \geq V_j(X_j)$, then we can take $y' \leq 1$. Then,

$$\begin{aligned} V_j(X'_j) &= (y' - x)u_j \\ V_i(X'_i) &= (2 - y' - z)u_i + (z - 1), \end{aligned}$$

so the lower bound required for j is

$$\begin{aligned} (y' - x)u_j &\geq (2 - y - z)u_j + (z - 1) \\ y' &\geq (2 + x - y - z) + (z - 1)/u_j \end{aligned}$$

and the upper bound required for i is

$$\begin{aligned} (2 - y' - z)u_i + (z - 1) &\geq (y - x)u_i \\ (2 + x - y - z) + (z - 1)/u_i &\geq y' \end{aligned}$$

These bounds are compatible as $u_i < u_j$ and $z > 1$.

Otherwise, we take $y' \geq 1$. Then,

$$\begin{aligned} V_j(X'_j) &= (1 - x)u_j + (y' - 1)(1 - u_j) \\ &= (2 - x - y')u_j + (y' - 1) \\ V_i(X'_i) &= (z - y')(1 - u_i) \end{aligned}$$

so the lower bound required for j is

$$\begin{aligned} (2 - x - y')u_j + (y' - 1) &\geq (2 - y - z)u_j + (z - 1) \\ y'(1 - u_j) &\geq z(1 - u_j) - (y - x)u_j \\ y' &\geq z - \frac{u_j}{1 - u_j}(y - x) \end{aligned}$$

and the upper bound required for i is

$$(z - y')(1 - u_i) \geq (y - x)u_i$$
$$z(1 - u_i) - (y - x)u_i \geq y'(1 - u_i)$$
$$z - (y - x)\frac{u_i}{1 - u_i} \geq y'$$

$u_i < u_j$ implies $\frac{u_i}{1-u_i} < \frac{u_j}{1-u_j}$; as $y > x$, the bounds are compatible.

Case 2: $y \geq 1$. This case is handled analogously to the previous case.

In both cases, it is possible to select y' such that the values of both i and j either remain the same as in X' or increase. Hence, the new allocation X' is g-egalitarian.

By swapping each adjacent pair in a similar way, after $O(n^2)$ swaps we end up with a connected g-egalitarian allocation where the agent order is as claimed in the lemma. □

Now we return to proving lower bound on the number of queries.

Theorem 5.4. *(a) For any $n \geq 3$ and any $g \in (\frac{1}{n}, \frac{1}{\lceil n/2 \rceil})$, every algorithm for deciding the existence of a connected g-egalitarian allocation requires at least $(n/2) \cdot 2^{n/2-1}$ queries.*

(b) If in addition $g \in (\frac{1}{n}, \frac{1}{n-1})$, then every algorithm for deciding the existence of a connected g-egalitarian allocation requires at least $(n-1) \cdot 2^{n-2}$ queries.

Both results hold even for hungry agents.

Proof. We prove part (a) first. Given $n \geq 3$ and $g \in (\frac{1}{n}, \frac{1}{\lceil n/2 \rceil})$, let $k := \lfloor n/2 \rfloor$.

We will compute some $n - k$ positive real numbers $w'_1, \ldots, w'_{n-k}$, and consider a cake with two parts—the *left* part and the *right* part. The left part is worth g/w'_i to each agent $i \in [n-k]$ and $1-kg$ to each agent $n-k+1, \ldots, n$. The right part is worth $1 - g/w'_i$ to each agent $i \in [n-k]$ and kg to each agent $n-k+1, \ldots, n$. The agents' valuations are uniform within each part. See Figure 4 for an illustration.

Agent 1	g/w'_1	$1 - g/w'_1$
⋮	⋮	⋮
Agent $n-k$	g/w'_{n-k}	$1 - g/w'_{n-k}$
Agent $n-k+1$	$1 - kg$	kg
⋮	⋮	⋮
Agent n	$1 - kg$	kg

Fig. 4. Construction of the cake used in the proof of Theorem 5.4.

Note that $kg < \dfrac{\lfloor n/2 \rfloor}{\lceil n/2 \rceil} \leq 1$, so the valuations of agents $n-k+1, \ldots, n$ are valid. We will choose the w'_i to satisfy the following constraints:

(1) $w'_i > g$ for all $i \in [n-k]$, so the valuations of these agents are valid too;

(2) $1 - kg < g/w'_i$ for all $i \in [n-k]$, so that by Lemma 5.3, if a connected g-egalitarian allocation exists, we can assume that all agents $1, \ldots, n-k$ receive their pieces to the left of all agents $n-k+1, \ldots, n$. We will prove below that we can even assume agents $1, \ldots, n-k$ receive their pieces entirely inside the left part.

Specifically, let $M \geq 2^n$ be a sufficiently large constant (to be specified shortly), and for each $i \in [n-k]$, define $w'_i = \frac{M+2^{i-1}}{(n-k)M+2^{n-k}-1}$. Note that $\sum_{i \in [n-k]} w'_i = \frac{(n-k)M+\sum_{i \in [n-k]} 2^{i-1}}{(n-k)M+2^{n-k}-1} = 1$.

Since $\lim_{M \to \infty} w'_i = \frac{1}{n-k}$, we can choose a value $M \geq 2^n$ such that, for all $i \in [n]$, w'_i is as close as we want to $\frac{1}{n-k}$. In particular:

- Since $g < \frac{1}{\lceil n/2 \rceil} = \frac{1}{n-k}$, for a sufficiently large M we have $w'_i > g$.
- Since $g > 1/n$, which implies $\frac{1}{1/g-k} > \frac{1}{n-k}$, for M sufficiently large we have $w'_i < \frac{1}{1/g-k}$, which implies $1 - kg < g/w'_i$.

We now show that any algorithm that makes fewer than $(n/2)(2^{n/2-1})$ queries may not be able to decide whether a connected super-proportional allocation exists. As in the proofs of Theorems 3.6 and 3.8, we assume that the answer to every query made by the algorithm is consistent with the instance where the valuation of each agent is uniformly distributed in each of the left and the right parts. We show that the answers are compatible both with a "no" answer and with a "yes" answer, and therefore the algorithm cannot decide which answer is correct.

Proof that answers are compatible with "no". Consider the instance where the valuation of each agent is uniformly distributed in each part. Note that all agents are hungry. We show that a connected g-egalitarian allocation cannot exist.

Suppose on the contrary that such an allocation exists. As $1 - kg < g/w'_i$ for all $i \in [n-k]$, by Lemma 5.3 we can assume that all k agents $n-k+1, \ldots, n$ get their pieces at the right. Moreover, since each of these agents needs a value larger than g, they must consume the entire right part. Therefore, all agents in $[n-k]$ must receive a piece contained in the left part.

But in order to get a piece with value larger than g, each agent $i \in [n-k]$ must receive a fraction larger than w'_i of the left part. Therefore, the $n-k$ agents in the left part receive a fraction strictly larger than $\sum_{i \in [n-k]} w'_i = 1$ of the left part in total. This is impossible, and hence, no such allocation exists.

Proof that answers are compatible with "yes". We shall now show by construction that the information provided to the algorithm is also consistent with an instance with a connected allocation in which each agent receives a piece with value more than g.

Let agents $n-k+1, \ldots, n$ share the right part of the cake, so that each of these agents receives a piece with value g. While the value of this piece is not more than g yet, we will fix this later. Divide all valuations of the left part of the cake by g/w'_i for each $i \in [n-k]$; now each such agent values the remaining cake at 1. For a g-egalitarian allocation, it is sufficient to find an allocation of the left part in which each agent $i \in [n-k]$ receives a piece with value more than w'_i. Note that $\sum_{i \in [n-1]} w'_i = 1$, so we have reduced the problem to finding a connected allocation on a modified instance with $n-k$ hungry agents such that agent $i \in [n-k]$ has an entitlement of w'_i.

We claim that the entitlements are generic. To see this, let $N, N' \subseteq [n-k]$ such that $\sum_{i \in N} w'_i = \sum_{i \in N'} w'_i$. Since the denominators of the w'_i's are equal to each other, we have $\sum_{i \in N}(M + 2^{i-1}) = \sum_{i \in N'}(M + 2^{i-1})$. Since M is larger than $\sum_{i \in [n-1]} 2^{i-1}$, we must have $|N| = |N'|$, which implies that $\sum_{i \in N} 2^{i-1} = \sum_{i \in N'} 2^{i-1}$. The only way this is possible is when $N = N'$, which proves that the entitlements are generic.

In this modified instance, we have $n-k$ hungry agents with generic entitlements. Recall that the algorithm makes fewer than $(n/2) \cdot (2^{n/2-1})$ queries, which is fewer than $(n-k) \cdot (2^{n-k-1})$ queries, so some agent $i \in [n-k]$ is asked some $q < 2^{n-k-1}$ queries.

By the construction in the proof of Theorem 3.8, there exists an instance consistent with the information provided by the queries that admits a connected super-proportional allocation for agents in $[n-k]$.

We now have a connected proportional allocation such that each agent in $[n-k]$ receives a piece with value more than g and each agent in $\{n-k+1,\dots,n\}$ receives a piece with value exactly g. As the agents are hungry, using a proof similar to that of Lemma 3.1, we can slightly move the boundaries of agents $n-k,\dots,n$'s pieces such that every agent receives a piece with value more than g. This concludes the proof of part (a).

For part (b), we apply exactly the same proof but with $k=1$. □

Remark 5.5. Aumann et al. (2012) prove a closely-related result,[12] but in a different computational model. They assume that the agents' valuations are piecewise-constant and given explicitly to the algorithm. In this model, they prove that it is NP-hard to approximate the optimal egalitarian value to a factor better than 2, when the number of agents is variable.[13] Specifically, they show a reduction from 3-dimensional matching (3DM) such that, if the answer to the 3DM instance is "yes", then some allocation gives each agent at least some value g; and if the answer is "no", then every allocation gives some agent at most $g/2$. In their reduction, $g \geq \frac{4}{n}$.

Bu and Song (2023) strengthened the above result by proving that the optimal egalitarian welfare is NP-hard to approximate to any constant factor, when the number of agents is constant. [14] Specifically, they show a reduction from 3-SAT such that, if the answer to the 3-SAT instance is "yes", then some allocation gives each agent at least $1/3$; and if the answer is "no", then every allocation gives some agent at most $1/(2r+1)$, where r is any positive integer.

In contrast to both these works, we focus on the Robertson-Webb query model, which is harder as it gives less information to the algorithm (only specific queries rather than the entire valuation). For this model we provide an unconditional exponential lower bound. Also, our result holds for a different range of egalitarian values: $g \in \left(\frac{1}{n}, \frac{1}{\lceil n/2 \rceil}\right)$; when n is even, our result holds for $g \in (\frac{1}{n}, \frac{2}{n})$.

Interestingly, when g is much larger, the problem is solvable in polynomially many queries!

Theorem 5.6. *For any $n \geq 3$ hungry agents and any $g > 1/2$, there exists an algorithm for deciding the existence of a connected g-egalitarian allocation using $O(n^2)$ queries.*

Proof. A g-egalitarian allocation requires giving each agent a value larger than g. As an intermediate step, we present Algorithm 4, that decides the existence of an allocation in which each agent receives value at least g.

To prove the correctness of Algorithm 4, we make an observation that holds for any $g \in (0,1)$. If there exists an allocation with egalitarian value at least g, in which agent i receives the leftmost piece, then the g-mark of i is at most as large as the $(1-g)$-marks of all other $n-1$ agents.

Now, when $g > 1/2$, if there exists an agent i whose g-mark is at most as large as the $(1-g)$-marks of all other $n-1$ agents, then there is *only one* such agent, as when $g > 1/2$, the $(1-g)$-mark of every agent is strictly smaller than the g-mark of that agent. Therefore, if an allocation with egalitarian value at least g exists, then agent i must receive the leftmost piece in that allocation. So we can assume that agent i receives the leftmost piece, and proceed recursively to check if the remaining cake can be divided among the remaining agents. If no such agent i exists, then the observation implies that an allocation with egalitarian value at least g does not exist.

Finally, to check whether there exists an allocation with egalitarian value larger than g, we check whether the allocation returned by Algorithm 4 has allocated the entire cake (i.e., the rightmost piece ends at 1). If it is the case, then an allocation with egalitarian value larger than g does not exist; otherwise, we can move all piece

[12]The result appears in the arXiv paper (Aumann et al., 2012), but not in the published paper (Aumann et al., 2013).
[13]For a constant number of agents, they provide a PTAS.
[14]For a constant number of agents, they provide a polynomial-time algorithm.

Algorithm 4 Determining the existence of a connected allocation for n hungry agents in which each agent gets value at least g, for some $g > 1/2$.

for $i = 1, \dots, n$ **do**
  $y_i \leftarrow \textsc{Mark}_i(0, g)$
  $z_i \leftarrow \textsc{Mark}_i(0, 1-g)$
end for
if $n = 1$ **then**
  Give $[0, y_i]$ to agent 1;
  return true
end if
if for some $i \in [n]$, $y_i \leq z_j$ for all $j \neq i$ **then**
  Give $[0, y_i]$ to agent i;
  Recursively apply the algorithm to the remaining cake $[y_i, 1]$ and the remaining agents $N \setminus \{i\}$.
else
  return false
end if

Table 2. Query complexity for deciding the existence of a connected allocation with a given bound on the egalitarian value.

Bound on egalitarian value	**Query complexity**	Source
At least $1/n$	$\Theta(n \log n)$	Even and Paz (1984)
More than $1/n$	$\Theta(n^2)$	Theorem 3.7
More than $g \in \left(\frac{1}{n}, \frac{1}{n-1}\right)$	$\Theta(n \cdot 2^n)$	Theorem 5.4(b)
More than $g \in \left[\frac{1}{n-1}, \frac{1}{\lceil n/2 \rceil}\right)$	$\Omega(n \cdot 2^{n/2}), O(n \cdot 2^n)$	Theorem 5.4(a)
More than $g \in \left[\frac{1}{\lceil n/2 \rceil}, \frac{1}{2}\right]$	$O(n \cdot 2^n)$	Corollary 5.2
More than $g \in \left(\frac{1}{2}, 1\right)$	$O(n^2)$	Theorem 5.6

boundaries slightly rightwards, as in Lemma 3.1 (using the assumption that all agents are hungry), to get an allocation with egalitarian value larger than g.

The algorithm makes $2n$ queries in each iteration, and runs for n iterations; therefore the total number of queries is in $O(n^2)$. □

Theorems 5.1 and 5.4 together give an asymptotically tight bound of $\Theta(n \cdot 2^n)$ on the number of queries, even for hungry agents with equal entitlements.

Table 2 summarizes our results on the query complexity of deciding the existence of (and computing) a connected allocation that guarantees each agent a piece with a certain value for n hungry agents.

For $g \in \left[\frac{1}{n-1}, \frac{1}{2}\right]$, the exact query complexity remains open.

6 Pies

We now consider a *pie*, where the resource is modeled by a circle instead of by an interval. We use $C = [0, 1)$ to represent the pie, but in contrast to the cake version, the endpoints 0 and 1 are "joined" together—they are considered the same point. Therefore, the piece $[0, a] \cup [b, 1)$ is also considered a connected piece for any $a, b \in C$ with $a \leq b$. Several papers have studied the special properties of pie-cutting (Stromquist, 2007; Thomson, 2007; Brams et al., 2008; Barbanel et al., 2009).

We show that the problem of deciding the existence of a connected super-proportional allocation of a pie is intractable.

Theorem 6.1. *No finite algorithm can decide the existence of a connected super-proportional allocation of a pie, even for hungry agents with equal entitlements.*

Proof. Suppose by way of contradiction that some finite algorithm decides the existence of a connected super-proportional allocation of a pie for n agents with equal entitlements. Assume that the information provided to the algorithm by eval and mark queries is consistent with that where every agent's valuation is uniformly distributed over the pie (in which case there is no connected super-proportional allocation of the pie), and so the algorithm should output "false". However, we shall now show that the information provided to the algorithm is also consistent with an instance with a connected super-proportional allocation. This means that the algorithm is not able to differentiate between the two, resulting in a contradiction.

Let P be the set of all points on the pie mentioned by the algorithm or by the queries—for example, if an $\textsc{Eval}_i(x, y)$ query is made by the algorithm, or if a $\textsc{Mark}_i(x, r)$ query is made by the algorithm and y is returned, then x and y are added to P. Since the algorithm is finite, P is finite. For $x \in C$, define $\bar{x} = \{x, x + 1/n, \ldots, x + (n-1)/n\} \subseteq C$ where all numbers in the set are modulo 1. (From now on, every point mentioned is modulo 1.) Since P is finite, there exists a point $x \in C$ such that $\bar{x} \cap P = \emptyset$. Fix x. Let $\epsilon \in (0, 1/n)$ be a number smaller than the distance between any element in $\bar{x}$ and any element in P.

Construct agent 1's valuation function as follows:

- $V_1([x - \epsilon, x]) = \epsilon/2$;
- $V_1([x, x + \epsilon]) = \epsilon + \epsilon/2$;
- $V_1([x + \epsilon, x - \epsilon]) = 1 - 2\epsilon$;
- The valuation is distributed uniformly within each of these three intervals.

Intuitively, the valuation is uniform except that a value of $\epsilon/2$ is moved from the interval $[x - \epsilon, x]$ to the interval $[x, x + \epsilon]$. Note that this construction is consistent with our assumptions, since the interval $[x - \epsilon, x + \epsilon]$ is entirely contained between two adjacent points of P, and the valuation between these two points is consistent with a uniform valuation over the entire pie.

Here is an example for $n = 4$. Suppose the algorithm asked about the points in $P = \{0, 0.1, \ldots, 0.9\}$. Let $x = 0.53$, so $\bar{x} = \{0.53, 0.78, 0.03, 0.28\}$. Let $\epsilon := 0.01$. Agent 1's valuation is given in the following table (the valuation is uniform within each interval):

Interval:	[0, .52]	[.52, .53]	[.53, .54]	[.54, 1]
Value:	.52	.005	.015	.46

All other $n - 1$ agents have valuation functions uniformly distributed over the pie. Note that all agents are hungry. By changing the axis to start at the point x, we see that the $1/n$-mark (starting at the point x) of agent 1 is at $x + 1/n - \epsilon/2$ while that of the other agents are at $x + 1/n$. Therefore, the $1/n$-mark of agent 1 is different from that of the other agents. By Theorem 3.2, there exists a connected super-proportional allocation starting from the point x. This means that the algorithm is not able to differentiate between the two instances. □

7 Conclusion

We have studied necessary and sufficient conditions for the existence of a connected super-proportional allocation on the interval cake (Theorems 3.2 and 4.4). We have shown that computing this condition requires $\Theta(n \cdot 2^n)$ queries even for agents with equal entitlements (Theorem 4.7) or hungry agents with generic entitlements (Theorem 3.9), and $\Theta(n^2)$ for hungry agents with equal entitlements (Theorem 3.7). The same bounds hold for the computation of such an allocation if it exists. We have also shown that for connected allocations where each agent receives some given value $g > 1/n$, the number of queries to decide the existence of such allocations is in $\Theta(n \cdot 2^n)$ (Theorems 5.1 and 5.4). Using binary search on g, one can find the optimal g for which a g-egalitarian allocation exists in time $\Theta(n \cdot 2^n \cdot d)$, where d is the required accuracy (the required number of decimal digits in the output). Finally, we have shown that no finite algorithm can decide the existence of a connected super-proportional allocation of a pie (Theorem 6.1).

A natural question that arose from our work is whether there is an algorithm that (asymptotically) attains the lower bound in (5) for hungry agents with entitlements that are neither generic nor equal.

Additionally, our work can be extended in several ways.

Beyond the unit interval. We can consider cakes with more complex topologies, such as graphical cakes (Bei and Suksompong, 2021), tangled cakes (Igarashi and Zwicker, 2023), and two-dimensional cakes (Segal-Halevi et al., 2017).

Chores and generalized super-proportionality. Chore-cutting is a variant of cake-cutting in which the agents consider the 'cake' undesirable, so they want as little cake as possible. A super-proportional chore allocation is an allocation in which $V_i(X_i) < \frac{1}{n}$ for all $i \in [n]$.

More generally, let $\mathbf{s} := (s_1, \ldots, s_n)$ be a vector in which each element is one of the comparison symbols $\{<, =, >\}$. Let us call an allocation **s**-*proportional* if $V_i(X_i) s_i w_i$ for all $i \in [n]$ (super-proportionality corresponds to the special case in which s_i is $' >'$ for all $i \in [n]$; super-proportionality for chores corresponds to the case that s_i is $' <'$ for all $i \in [n]$). Barbanel (2005, Theorem 5.54) proved that, if at least two agents have non-identical valuations, then an **s**-proportional allocation exists for any **s**. Under what conditions does a *connected* **s**-proportional allocation exist?

Envy-freeness. Proportionality and super-proportionality compare the value of each agent to a predefined threshold. There is another family of fairness notions, in which the value of each agent is compared to the values of every other agent. Formally, an allocation $(X_1, \ldots, X_n)$ is called

- *Envy-free* — if $V_i(X_i) \geq V_i(X_j)$ for all $i, j \in [n]$ (George and Marvin, 1958).
- *Strongly envy-free* — if $V_i(X_i) > V_i(X_j)$ for all $i, j \in [n]$ (Barbanel, 1996b).
- *Super envy-free* — if $V_i(X_i) > 1/n > V_i(X_j)$ for all $i, j \in [n]$ (Barbanel, 1996b). Note that super envy-freeness implies strong envy-freeness, which implies both envy-freeness and super-proportionality.

It is known that, in every cake-cutting instance, a connected envy-free allocation exists (Stromquist, 1980; Su, 1999). What conditions guarantee the existence of a connected super-proportional allocation that is also envy-free?

Barbanel (1996b, 2005) proved that a strongly-envy-free allocation exists if and only if the agents' valuations are pairwise-different (no two agents have the same valuations), and a super-envy-free allocation exists if and only if the agents' valuations are linearly independent. Webb (1999) and Chèze (2025) presented algorithms for finding a super envy-free allocation when it exists. What conditions guarantee the existence of a *connected* strongly / super envy-free allocation?

More generally, suppose there is an n-by-n matrix S, in which each entry $s_{i,j}$ is one of the comparison symbols $\{<, =, >\}$. Let us call an allocation *S-envy-free* if $V_i(X_j) s_{i,j} \frac{1}{n}$ for all $i, j \in [n]$ (super envy-freeness corresponds to the special case in which $s_{i,i} ='>'$ and $s_{i,j} ='<'$ for all $i, j \in N, j \neq i$). An allocation is S-envy-free if and only if there exists an n-by-n matrix Z with $\sum_{j=1}^{n} z_{i,j} = 0$ for all $i \in [n]$ and $z_{i,j} s_{i,j} 0$, such that $V_i(X_j) = \frac{1}{n} + z_{i,j}$ for all $i, j \in [n]$; the latter condition is called *hyper envy-freeness with respect to* Z (Chèze and Amodei, 2019); it can be extended to agents with different entitlements by replacing $\frac{1}{n}$ with the entitlement w_j. Chèze and Amodei (2019) proved conditions for the existence of hyper-envy-free allocations, and provided an efficient algorithm for deciding whether a given instance has a S-envy-free allocation. Can these results be extended to connected allocations?

Acknowledgments

A preliminary version of this work appeared in Proceedings of ECAI 2024 — the 27th European Conference on Artificial Intelligence (Jankó et al., 2024). The present version contains new results on stronger than super-proportional allocations (Section 5) and pies (Section 6), improved lower bounds in Theorems 3.6 and 3.8 (the new lower bounds exactly match the upper bounds), a discussion of why the characterization for hungry agents cannot be generalized to non-hungry agents (Section 4), as well as all proofs omitted from the conference version (Theorems 4.4 to 4.6).

This work was partially supported by the Hungarian Scientific Research Fund (OTKA grant K143858), and the Hungarian Academy of Sciences under its Momentum Programme, grant number LP2021-2, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 513023562, by NKFIH OTKA-129211, by the Israel Science Foundation grant 712/20 and 1092/24, and by the Singapore Ministry of Education under grant number MOE-T2EP20221-0001. We thank Tassle, Neal Young,[15] Paul Rubin[16] and Rob Pratt[17] for their helpful answers, Warut Suksompong for his feedback on an earlier draft, and participants of the COMSOC online seminar and the anonymous ECAI 2024 and JAIR reviewers for their very helpful comments.

We are particularly grateful to an anonymous ECAI 2024 reviewer for suggesting the solution to Case 2 in the proof of Lemma 3.1, and to Chaya Keller for her help with the proof of Lemma 3.4.

References

Yonatan Aumann, Yair Dombb, and Avinatan Hassidim. 2012. Computing socially-efficient cake divisions. *CoRR* abs/1205.3982 (2012).

Yonatan Aumann, Yair Dombb, and Avinatan Hassidim. 2013. Computing socially-efficient cake divisions. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 343–350.

Julius Barbanel. 1996a. Game-theoretic algorithms for fair and strongly fair cake division with entitlements. *Colloquium Mathematicae* 69, 1 (1996), 59–73. http://eudml.org/doc/210327

Julius B Barbanel. 1996b. Super envy-free cake division and independence of measures. *J. Math. Anal. Appl.* 197, 1 (1996), 54–60.

Julius B Barbanel. 2005. *The geometry of efficient fair division.* Cambridge University Press.

Julius B Barbanel, Steven J Brams, and Walter Stromquist. 2009. Cutting a pie is not a piece of cake. *The American Mathematical Monthly* 116, 6 (2009), 496–514.

Xiaohui Bei and Warut Suksompong. 2021. Dividing a graphical cake. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*. 5159–5166.

[15] https://cstheory.stackexchange.com/q/53901
[16] https://or.stackexchange.com/a/12949
[17] https://or.stackexchange.com/q/12951

Steven J Brams, Michael A Jones, and Christian Klamler. 2008. Proportional pie-cutting. *International Journal of Game Theory* 36 (2008), 353–367.

Xiaolin Bu and Jiaxin Song. 2023. Maximize egalitarian welfare for cake cutting. In *International Workshop on Frontiers in Algorithmics*. Springer, 263–280.

Katarína Cechlárová, Jozef Doboš, and Eva Pillárová. 2013. On the existence of equitable cake divisions. *Information Sciences* 228 (2013), 239–245.

Guillaume Chèze. 2025. Envy-free cake cutting: a polynomial number of queries with high probability: G. Chèze. *Social Choice and Welfare* (2025), 1–20.

Guillaume Chèze and Luca Amodei. 2019. How to cut a cake with a gram matrix. *Linear Algebra Appl.* 560 (2019), 114–132.

Logan Crew, Bhargav Narayanan, and Sophie Spirkl. 2020. Disproportionate division. *Bulletin of the London Mathematical Society* 52, 5 (2020), 885–890.

Ágnes Cseh and Tamás Fleiner. 2020. The complexity of cake cutting with unequal shares. *ACM Transactions on Algorithms (TALG)* 16, 3 (2020), 1–21.

Lester E Dubins and Edwin H Spanier. 1961. How to cut a cake fairly. *The American Mathematical Monthly* 68, 1P1 (1961), 1–17.

Edith Elkind, Erel Segal-Halevi, and Warut Suksompong. 2022. Mind the gap: Cake cutting with separation. *Artificial Intelligence* 313 (2022), 103783.

Shimon Even and Azaria Paz. 1984. A note on cake cutting. *Discrete Applied Mathematics* 7, 3 (1984), 285–296.

Gamow George and Stern Marvin. 1958. Puzzle-math.

Paul W Goldberg, Alexandros Hollender, and Warut Suksompong. 2020. Contiguous cake cutting: Hardness results and approximation algorithms. *Journal of Artificial Intelligence Research* 69 (2020), 109–141.

Ayumi Igarashi and William S Zwicker. 2023. Fair division of graphs and of tangled cakes. *Mathematical Programming* (2023), 1–45.

Zsuzsanna Jankó and Attila Joó. 2022. Cutting a cake for infinitely many guests. *The Electronic Journal of Combinatorics* 29, 1 (2022).

Zsuzsanna Jankó, Attila Joó, Erel Segal-Halevi, and Sheung Man Yuen. 2024. On Connected Strongly-Proportional Cake-Cutting. In *Proceedings of the 27th European Conference on Artificial Intelligence*. 3356–3363.

Claudia Lindner and Jörg Rothe. 2024. *Cake-Cutting: Fair Division of Divisible Goods*. Springer Nature Switzerland, Cham, 507–603. doi:10.1007/978-3-031-60099-9_8

Ariel D Procaccia. 2016. Cake cutting algorithms. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 13, 311–329.

Kenneth Rebman. 1979. How to get (at least) a fair share of the cake. *Mathematical Plums (Edited by R. Honsberger), The Mathematical Association of America* (1979), 22–37.

Jack Robertson and William Webb. 1998. *Cake-Cutting Algorithms: Be Fair if You Can*. Peters/CRC Press.

Erel Segal-Halevi. 2019. Cake-cutting with different entitlements: How many cuts are needed? *J. Math. Anal. Appl.* 480, 1 (2019), 123382.

Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. 2017. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics* 70 (2017), 1–28.

Hugo Steinhaus. 1948. The problem of fair division. *Econometrica* 16 (1948), 101–104.

Walter Stromquist. 1980. How to cut a cake fairly. *Amer. Math. Monthly* 87, 8 (1980), 640–644.

Walter Stromquist. 2007. A Pie That Can't Be Cut Fairly (revised for DSP). In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Walter Stromquist. 2008. Envy-free cake divisions cannot be found by finite protocols. *the electronic journal of combinatorics* 15, 1 (2008), R11.

Francis Edward Su. 1999. Rental harmony: Sperner's lemma in fair division. *Amer. Math. Monthly* 106, 10 (1999), 930–942.
Warut Suksompong. 2021. Constraints in fair division. *ACM SIGecom Exchanges* 19, 2 (2021), 46–61.
William Thomson. 2007. Children crying at birthday parties. Why? *Economic Theory* 31, 3 (2007), 501–521.
William A Webb. 1999. An algorithm for super envy-free cake division. *Journal of mathematical analysis and applications* 239, 1 (1999), 175–179.
Gerhard J Woeginger and Jiří Sgall. 2007. On the complexity of cake cutting. *Discrete Optimization* 4, 2 (2007), 213–220.
Douglas R Woodall. 1986. A note on the cake-division problem. *J. Comb. Theory, Ser. A* 42, 2 (1986), 300–301.

A Left-Marks and Right-Marks

Our results assume that algorithms have access to eval and right-mark queries in the Robertson-Webb model. We show that our results also hold if algorithms have access to eval and *left-mark* queries.

A.1 Definitions

For each agent $i \in [n]$, value $r \in [0,1]$, and point $x \in C$, define *left-mark* such that $\textit{Left-Mark}_i(x,r)$ is the *leftmost* (smallest) point $z \in C$ such that $V_i([x,z]) = r$ (such a point exists due to the continuity of the valuations); if $V_i([x,1]) < r$, then $\textsc{Left-Mark}_i(x,r)$ returns ∞. We define $\textit{Right-Mark}_i(x,r)$ to be the same as $\textsc{Mark}_i(x,r)$ in the main text, but use "right-mark" in this section to avoid confusion with left-mark. Note that for a hungry agent i, $\textsc{Left-Mark}_i(x,r) = \textsc{Right-Mark}_i(x,r)$ for all $x \in C, r \in [0,1]$.

Let $\mathcal{I}$ be an instance with n agents with valuation functions $(V_i)_{i\in[n]}$ and entitlements $\mathbf{w}$. A *mirrored instance* $\overleftarrow{\mathcal{I}}$ of $\mathcal{I}$ is the instance with n agents with valuation functions $(\overleftarrow{V}_i)_{i\in[n]}$ and entitlements $\mathbf{w}$ such that $\overleftarrow{V}_i([x,y]) = V_i([1-y, 1-x])$ for all $x, y \in C$ with $x \leq y$. In other words, the cake in $\overleftarrow{\mathcal{I}}$ is "mirrored" from the cake in $\mathcal{I}$. A class of instances C is *closed under mirror* if $\mathcal{I} \in C$ implies that $\overleftarrow{\mathcal{I}} \in C$.

Note that the classes of instances we consider in our results are closed under mirror.

A.2 Super-proportional allocations

We now show that if there is an algorithm, having access to right-mark queries, that can determine the existence of a connected super-proportional allocation in a class of instances closed under mirror, then there exists another algorithm, having access to left-mark queries, that can also do the same in the same class of instances. Furthermore, the number of queries made by the new algorithm is within a multiplicative factor of 2 from the number of queries made by the old algorithm. This shows that our results hold in whichever model of Robertson-Webb, and the use of right-mark is only for convenience.

Proposition A.1. *Let C be a class of instances closed under mirror. Suppose there exists an algorithm $\mathcal{A}$ such that for each instance in C, algorithm $\mathcal{A}$ can determine the existence of a connected super-proportional allocation in the instance using at most k eval and right-mark queries and using no left-mark queries. Then, there exists an algorithm $\mathcal{B}$ such that for each instance in C, algorithm $\mathcal{B}$ can determine the existence of a connected super-proportional allocation in the instance using at most $2k$ eval and left-mark queries and using no right-mark queries.*

Proof. Let $\mathcal{I}$ be an instance in C in which we wish to determine the existence of a connected super-proportional allocation using eval and left-mark queries and using no right-mark queries. By assumption, there exists an algorithm $\mathcal{A}$ that can determine the existence of a connected super-proportional allocation in $\overleftarrow{\mathcal{I}}$ using at most k eval and right-mark queries and using no left-mark queries.

Our algorithm $\mathcal{B}$ simulates algorithm $\mathcal{A}$ on $\mathcal{I}$ as follows:

- **Case 1: $\mathcal{A}$ makes an $\textsc{Eval}_i(x,y)$ query on $\overleftarrow{\mathcal{I}}$.**
  Then, algorithm $\mathcal{B}$ makes an $\textsc{Eval}_i(1-y, 1-x)$ query on $\mathcal{I}$.
- **Case 2: $\mathcal{A}$ makes a $\textsc{Right-Mark}_i(x,r)$ query on $\overleftarrow{\mathcal{I}}$.**
  Then, algorithm $\mathcal{B}$ makes a $\textsc{Left-Mark}_i(0, \textsc{Eval}_i(0, 1-x) - r)$ query on $\mathcal{I}$.

Let us verify that the two implementations are identical.

- Suppose $r = \textsc{Eval}_i(x,y)$ on $\overleftarrow{\mathcal{I}}$. Then, $\textsc{Eval}_i(1-y, 1-x)$ on $\mathcal{I}$ is equal to $V_i([1-y, 1-x])$, which is equal to $\overleftarrow{V}_i([x,y]) = r$.
- Suppose $y = \textsc{Right-Mark}_i(x,r)$ on $\overleftarrow{\mathcal{I}}$. Then, since the value of $\textsc{Eval}_i(0, 1-x)$ on $\mathcal{I}$ is equal to $V_i([0, 1-x])$ and $\textsc{Left-Mark}_i(0,k)$ on $\mathcal{I}$ is equal to $1-\textsc{Right-Mark}_i(0, 1-k)$ on $\overleftarrow{\mathcal{I}}$ for any $k \in [0,1]$, $\textsc{Left-Mark}_i(0, \textsc{Eval}_i(0, 1-x)-r)$ on $\mathcal{I}$ is equal to $1-\textsc{Right-Mark}_i(0, 1-(V_i([0,1-x])-r))$ on $\overleftarrow{\mathcal{I}}$. But $1-V_i([0,1-x]) = 1-\overleftarrow{V}_i([x,1]) = \overleftarrow{V}_i([0,x])$, which means that the result is equal to $1 - \textsc{Right-Mark}_i(0, \overleftarrow{V}_i([0,x]) + r)$. This is equal to $1 - \textsc{Right-Mark}_i(x,r) = 1-y$, which is the mirrored point of y.

This shows that if algorithm $\mathcal{A}$ determines the existence of a connected super-proportional allocation on $\overleftarrow{\mathcal{I}}$, then $\mathcal{B}$ determines the existence of a connected super-proportional allocation on $\mathcal{I}$. Note that algorithm $\mathcal{B}$ makes at most two times the number of queries that $\mathcal{A}$ makes. □

A.3 Proportional allocations

With unequal entitlements, even a connected *proportional* allocation may not exist. Algorithm 3 can be modified to determine the existence of a connected proportional allocation for agents with unequal entitlements (and to output one if it exists) by using *left-marks* instead of right-marks.

Algorithm 5 Determining the existence of a connected proportional allocation for n agents.

```
b_∅ ← 0
for k = 1, ..., n do
    for each subset N ⊆ [n] with |N| = k do
        b_N ← ∞
        for each agent i ∈ N do
            y ← LEFT-MARK_i(b_{N\{i}}, w_i)
            if y < b_N then b_N ← y                ▷ this finds the "best" b_N
        end for
    end for
end for
if b_[n] ≤ 1 then return true else return false
```

Proposition A.2. *Algorithm 5 decides whether a connected proportional allocation exists for n agents using at most* $n \cdot 2^{n-1}$ *queries.*

We remark that our algorithm is similar to that in Aumann et al. (2012, Theorem 4) where they find an approximate optimum egalitarian welfare on a piece of cake—here, we extend it to agents with possibly unequal entitlements.

Received 07 May 2025; accepted 06 February 2026.