Hardness Results for Consensus-Halving

Aris Filos-Ratsikas[1], Søren Kristoffer Stiil Frederiksen[2], Paul W. Goldberg[1] and Jie Zhang[1]

[1] Department of Computer Science, University of Oxford, UK.
{aris.filos-ratsikas, paul.goldberg, jie.zhang}@cs.ox.ac.uk
[2] Department of Computer Science, Aarhus University, Denmark.
sorenf@gmail.com

Abstract. We study the *consensus-halving* problem of dividing an object into two portions, such that each of n agents has equal valuation for the two portions. The ϵ-approximate consensus-halving problem allows each agent to have an ϵ discrepancy on the values of the portions. We prove that computing ϵ-approximate consensus-halving solution using n cuts is in PPA, and is PPAD-hard, where ϵ is some positive constant; the problem remains PPAD-hard when we allow a constant number of additional cuts. It is NP-hard to decide whether a solution with $n-1$ cuts exists for the problem. As a corollary of our results, we obtain that the approximate computational version of the Continuous Necklace Splitting Problem [2,18] is PPAD-hard when the number of portions t is two.

1 Introduction

Suppose that two families want to split a piece of land into two regions such that every member of each family believes the land is equally divided, or suppose that a conference organizer wants to assign the conference presentations to the morning and the afternoon sessions, so that every participant thinks that the two sessions are equally interesting. Is it possible to achieve these objectives? If yes, how can it be done and how efficiently? What if we aim for "almost equal" instead of "equal"?

These real-life problems can be modeled as the *Consensus-Halving problem* [25]. More formally, there are n agents and an object to be divided; each agent may have a different opinion as to which part of the object is more valuable. The problem is to divide the object into two portions such that each of the n agents believes the two portions have equal value, according to her personal opinion. The division may need to cut the object into some pieces and then assign them to the two portions.

The Consensus-Halving problem is a typical fair division problem that studies how to divide a set of resources between a set of agents who have valuations on the resources, such that some fairness properties are fulfilled. The resources can be of many different kinds such as, commodities, heritage, commercial land, spectrum bandwidth, etc. The fair division literature, which dates back to the late 1940s [28], has studied a plethora of such problems, with *chore-division* [21,13], *rent-partitioning* [15,7,30] and the perhaps the most well-known one, *cake-cutting* [8,23] being notable examples.

The Consensus-Halving problem was studied by Simmons and Su [25], who proved that a solution with n cuts always exists and constructed a protocol that finds an approximate solution, which allows for a small discrepancy on the values of the two portions. Their proofs are based on one of the most applied theorems in topology, the *Borsuk-Ulam Theorem* [6] and its combinatorial analogue, known as *Tucker's Lemma* [31].

Although the authors in [25] establish that a solution is always possible and also provide a protocol for finding one such solution, they do not answer the question of "efficiency", i.e. how fast can a protocol find an (approximate) solution. The running time of their protocol is worst-case exponential-time,[3] but that does not preclude the possibility of worst-case polynomial-time algorithms for the problem. In this paper, we study the *computational complexity* of finding a solution to the Consensus-Halving problem.

[3] The protocol runs in time polynomial in the number of vertices in the triangulation of a geometrical object, which, to achieve a small discrepancy, has to be exponentially large; see more details in Section 2.

1.1 Our results

We are interested in the computational complexity of computing an ϵ-approximate solution to the Consensus-Halving problem, and the complexity of deciding whether given an input instance, $n-1$ cuts are sufficient to achieve an ϵ-approximate solution to Consensus-Halving. We establish three main results:

1. We prove that the problem of finding an ϵ-approximate solution to the Consensus-Halving problem for n agents using n cuts is in the computational class PPA; we obtain the result via a reduction to the computational version of Tucker's Lemma [20,1].

2. We prove that the problem of finding an ϵ-approximate solution to the Consensus-Halving problem for n agents using n cuts is PPAD-hard. Moreover, the problem remains PPAD-hard even if we allow a constant number of additional cuts. The result is established via a reduction from the approximate Generalized Circuit problem [9,11,24]. Our results imply that finding an ϵ-approximate solution to the Continuous Necklace Splitting Problem [18,2] is PPAD-hard when $t = 2$.

3. We prove that it is NP-hard to decide whether or not an ϵ-approximate solution to the Consensus-Halving problem for n agents using $n-1$ cuts exists. We establish the result via a reduction from 3SAT.

1.2 Related work

The study of fair division dates back to the pioneering work of Steinhaus [28]; since then, fair division problems have been in the center of interest in several scientific fields such as mathematics, economics or even political sciences. The two very popular books of Brams and Taylor [8], and Robertson and Webb [23] are excellent introductions to the field and contain very useful references.

The most popular fair division problem, which can also be seen as a generalization of several other settings, is the *cake cutting problem*, where there is a heterogeneous resource to be divided, a "cake", and a set of agents who have a valuation measure on every possible piece of cake. The goal is to divide the cake into pieces, and assign each piece to an agent, while at the same time satisfying some desirable property such as *envy-freeness*, i.e. a guarantee that no agent would rather have some other agent's piece or *proportionality*, i.e. a guarantee that each agent receives at least a $1/n$-portion of the cake, according to her valuation.

For envy-free cake cutting, the existence of a solution with $n-1$ cuts when there are n agents was first proved by Stromquist [29] and then later on by Simmons [26] (also see [30]); similarly to the Consensus-Halving problem, those proofs rely on topological lemmas. On the complexity and algorithmic aspect, Deng et al. [12] proved that finding and envy-free cake cutting solution with $n-1$ cuts is PPAD-complete, and propose an FPTAS to find an approximate envy-free solution with three agents and two cuts, when the agents have monotone utility functions.[4]

The most relevant work to ours is by Simmons and Su [25]. They prove the existence of an exact solution for Consensus-Halving by using the Borsuk-Ulam Theorem, and present a protocol to find an approximate solution, based on Tucker's Lemma. They also show that there exist instances of the problem where n cuts are in fact essential for a solution to be possible. The existence of solutions to the problem was already known since [14,3,4] but the algorithm in [25] is constructive, in the sense that it actually finds such a solution and furthermore, it does not require the valuations of the players to be additively separable over subintervals, like some of the previous papers do. Actually, for the case of valuations which are probability measures, the existence of a solution with n cuts was known since as early as the 1940s [18] and can also be obtained as an application of the Hobby-Rice Theorem [17] (also see [2]). In fact, [18] proves the existence of a solution for a more general problem, in which the interval is divided in t different portions that all agents value equally; the corresponding problem is often referred to as *Continuous Necklace Splitting* [2]. For arbitrary measures

[4] Crucially, their PPAD-hardness result requires that the agents' valuations are sufficiently general, i.e. they are not necessarily additively separable over subintervals.

(or even set functions that are not countably additive), the version of the problem was named *Consensus-1/k-division* in [25]. The computational complexity aspects of Consensus-Halving were not considered in any of the aforementioned papers.

In this paper, we prove that Consensus-Halving is associated with the computational classes PPA (Polynomial Parity Arguments) and PPAD (Polynomial Parity Arguments on Directed graphs) [20]. PPAD is the class of all total search problems reducible to the End-Of-Line problem (for details, see Section 2.3). In the End-Of-Line problem, there is an exponential size graph consisting of directed paths and cycles; the input of the problem includes a polynomial function that given any vertex in the graph can compute the incoming edge or the outgoing edge; then given any starting vertex in a directed path, the problem requires to output another degree-one vertex. The class PPA is defined similarly in terms of an undirected graph and the corresponding problem is called Leaf. PPAD is a subclass of PPA, and both are subclasses of TFNP, the class of total search problems for which a solution is verifiable in polynomial time.

The class PPAD was introduced by Papadimitriou [20] as a candidate of capturing the complexity of computing a Nash equilibrium in games. Indeed, Chen et al. [9] and Daskalakis et al. [11] proved that computing a Nash equilibrium is PPAD-complete. Their proofs go via the *Generalized Circuit* problem, which is also what we use for our main hardness result. Generalized circuits differ from usual circuits in the sense that they can contain cycles, which basically induces a continuous function on the gates, and the solution is guaranteed by Brouwer's fixed point theorem. The ϵ-approximate Generalized Circuit problem is implicitly proven to be PPAD-complete for exponentially small ϵ in [11] and explicitly for polynomial small ϵ [9]. The same problem was also used in [10] to prove that finding an approximate market equilibrium for the Arrow-Debreu market with linear and non-monotone utilities is PPAD-complete and in [19] to prove that finding an approximate solution of the Competitive Equilibrium with Equal Incomes (CEEI) for indivisible items is PPAD-complete. More recently, Rubinstein [24] showed that computing an ϵ-approximate solution for the Generalized Circuit problem is PPAD-complete for a constant ϵ, which implies that finding an ϵ-approximate Nash equilibrium is PPAD-complete for constant ϵ, in the context of graphical games. This improvement should also lead to stronger hardness results in [10] and [19], as well as other problems that rely on the Generalized Circuit problem.

Interestingly, the classes PPAD and PPA capture the complexity of many "search problem versions" of problems of a topological nature. For example, computational analogues of Sperner's Lemma [27] and Brouwer's and Kakutani's fixed point theorems [5] are all known to be in PPAD [20]. Interestingly, Aisenberg et al. [1] recently proved that the search problems associated with the Borsuk-Ulam Theorem and Tucker's Lemma are PPA-complete; the later result will be useful in our reductions.

Finally, it is worth noting that while our positive results ("in PPA") hold for quite general functions, the negative results ("PPAD-hardness") hold even for well-behaved functions like probability measures. In particular, our PPAD-hardness result implies that Continuous Necklace Splitting[5] [18,2] is PPAD-hard when $t = 2$.

1.3 Organization

In Section 2, we introduce necessary notation, computational problems, and complexity classes that we will study. In Section 3, we prove that the search version of ϵ-approximate Consensus-Halving problem for n agents using n cuts is in PPA. In Section 4, we prove that the problem is PPAD-hard when we use n cuts and remains PPAD-hard even if a constant number of additional cuts is allowed. In Section 5, we prove that it is NP-hard to decide whether or not an solution to the ϵ-approximate consensus-halving problem with $n - 1$ cuts exists.

[5] The discrete version of Necklace Splitting is known to be in PPA by [1], whereas whether it is complete for the class is unknown.

2 Models and Computational Problems

In this section we review some of the basic definitions and results in combinatorial topology and computational complexity that will be used in the paper.

2.1 The Borsulk-Ulam Theorem and Tucker's Lemma

Denote the n-dimensional Euclidean space by $\mathbb{R}^n$. The $(n+1)$-dimensional unit ball is $B^{n+1} = \{\mathbf{x} \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} |x_i|^2 \leq 1\}$. Its surface is the n-dimensional sphere $S^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} |x_i|^2 = 1\}$. The $(n+1)$-dimensional cross-polytope is $P^{n+1} = \{\mathbf{x} \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} |x_i| \leq 1\}$, i.e., unit ball in the l_1-norm. Denote the surface of P^{n+1} by $C^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} |x_i| = 1\}$. The Borsuk-Ulam theorem is the following:

Theorem 1 (Borsuk-Ulam theorem). *Any continuous function $f : C^n \mapsto \mathbb{R}^n$ must have a vertex $\mathbf{x} \in C^n$, such that $f(\mathbf{x}) = f(-\mathbf{x})$.*

Note that although the theorem is originally stated on domain S^n, it is also true when the domain of f is the cross-polytope C^n.

In general, a *subdivision* or *simplicization* is the process of partitioning a geometric object into small objects such that any two such small objects either share a common facet or do not intersect. In particular, a *triangulation* of a geometric object is a procedure that partitions the original n-dimensional object into small n-simplices $\sigma_1, \ldots, \sigma_m$ such that for all i and j, $\sigma_i \cap \sigma_j$ either contains a common facet of σ_i and σ_j or a lower dimension face, or is empty. A triangulation T of S^n is *antipodally symmetric* if given simplex $\sigma \in T \cap S^n$, then $-\sigma \in T \cap S^n$, where $-\sigma$ is the simplex obtained by negating each vertices of σ. The notion can also be defined on other centrally symmetric objects, e.g., on C^n. The *d-skeleton* of a triangulation T is the collection of simplices of T of dimension at most d, i.e. $\{\sigma \in T : \dim(\sigma) \leq d\}$.

Tucker's Lemma can be formulated in a number of different ways; one is the following.

Lemma 1 (Tucker's Lemma). *Let T be an antipodally symmetric triangulation of C^n whose vertices are assigned labels from $\{+1, -1, +2, -2, \ldots, +n, -n\}$. The labels of antipodal vertices sum to zero, i.e., the labelling function λ satisfies $\lambda(-\mathbf{x}) = \lambda(\mathbf{x})$ for any vertex $\mathbf{x} \in C^n$. Then there must exist a 1-simplex (complementary edge) that its two vertices have opposite labels.*

In particular, we consider the following triangulation:

Definition 1 (Triangulation T). *Given C^n, use a set of equidistant parallel hyperplanes*

$$\left\{\{\mathbf{x} \in \mathbb{R}^{n+1} \mid x_i = k_j \cdot \tau\}, k_j = 0, \pm 1, \pm 2, \ldots, \pm \frac{1}{\tau}, i = 1, \ldots, n+1\right\}$$

to cut through C^n, such that all vertices on C^n are represented as $\mathbf{x} = (k_1\tau, \ldots, k_{n+1}\tau)$, where $|k_1| + \cdots + |k_{n+1}| = 1/\tau$. The full-dimensional subsimplices on C^n are obtained as convex hulls of sets of $n+1$ vertices whose values k_j differ by 1 in two entries, and are the same in the other $n-1$ entries.

We argue that T is indeed a triangulation of C^n. This will be established by the following two statements:

1. all of the full-dimensional minimum polytopes on $T \cap C^n$ are n-dimensional subsimplices,
2. given any two full-dimensional subsimplices, they either share a common facet or a lower dimension face, or they do not intersect.

First, regarding (1), suppose there is a full-dimensional minimum polytope on $T \cap C^n$ which is not an n-dimensional simplex. Then there must exist two vertices of the polytope that are not adjacent. By adjusting the value of the pairwise coordinates step by step, we can identify a series of adjacent vertices that constitute a path connecting these two vertices of the polygon. That is to say, such a polygon is actually subdivided into smaller polygons which contradicts the minimality assumption.

Regarding (2), denote two different subsimplices on $T \cap C^n$ by $\Delta_1^n = Conv(\mathbf{x}_1, \ldots, \mathbf{x}_{n+1})$ and $\Delta_2^n = Conv(\mathbf{x}'_1, \ldots, \mathbf{x}'_{n+1})$. Then either $\mathbf{x}_i = \mathbf{x}'_i$ for exactly n vertices, or $\mathbf{x}_i = \mathbf{x}'_i$ for less than n vertices. In the former case, Δ_1^n and Δ_2^n share a common facet, and in the later case they share a lower dimension face or do not intersect at all.

Clearly, T is a antipodally symmetric triangulation. We define the search version of Tucker's lemma as follows:

(n, τ)-TUCKER

Input: The antipodal symmetric triangulation T of C^n with mesh size τ; for any vertex $\mathbf{x}$ on C^n, a boolean circuit $\lambda : V(C^n) \to \{+1, -1, \cdots, +n, -n\}$ such that $\lambda(-\mathbf{x}) + \lambda(\mathbf{x}) = 0$.

Output: Two adjacent vertices $\mathbf{z}$, $\mathbf{z}'$ on C^n, with $\lambda(\mathbf{z}) + \lambda(\mathbf{z}') = 0$.

2.2 The CHALVING Problem

Following the fair division literature, we represent the object O as a line segment $[0, 1]$. Each agent in the set of agents $N = \{1, \ldots, n\}$ has its own valuation over any subset of interval $[0, 1]$. These valuations are:

- *bounded*, i.e. there exists $M > 0$, such that for any subinterval $X \subseteq [0, 1]$, it holds that $-M \le u_i(X) \le M$.
- *non-atomic*, i.e. agents have no value on any single point on the interval.

In other words, we define the valuation functions to be general enough and in particular they need not even be additively separable over subsets or even positive. Proving containment results in a class for more general functions makes the result stronger. On the other hand, the more specific the valuation functions, the stronger the hardness results are. Our hardness results (PPAD-hardness and NP-hardness) hold even for well-behaved functions, such as probability measures, i.e. measures u_i that satisfy non-negativity and countable additivity as well as $u_i(O) = 1$ for all agents i.

A set of k cuts $\{t_1, \ldots, t_k\}$, where $0 \le t_1 \le \ldots \le t_k \le 1$, means that we cut along the points $t_1, \ldots, t_k$, such that the object is cut into $k + 1$ pieces $X_i = [t_{i-1}, t_i], i = 1, \ldots, k + 1$, where $t_0 = 0, t_{k+1} = 1$. In case some cuts happen to be on the same point, say $t_{j-1} = t_j$, then the corresponding subinterval X_j is a single point on which agents have no value. We will consider cuts on the same points to be the same cut, e.g. if there is only one such occurrence, we will consider the number of cuts to be $k - 1$.

The consensus-halving problem is to divide the object O into two portions O_+ and O_-, such that every agent derives equal valuation from the two portions, i.e., $u_i(O_+) = u_i(O_-), \forall i \in N$. The ϵ-approximate consensus-halving problem allows that each agent has a small discrepancy on the values of the two partitions, i.e., $|u_i(O_+) - u_i(O_-)| < \epsilon$. With the continuity of the valuation functions, Simons and Su [25] prove the following results.

Lemma 2 ([25]). *A solution to the consensus-halving problem for n players that uses n cuts always exists and there exist valuations for which it is necessary to use precisely n cuts. Moreover, there exists a protocol for finding a solution to the ϵ-approximate consensus-halving problem.*

Take any point $\mathbf{x} = (x_1, \ldots, x_{n+1})$ on the n-dimensional cross-polytope, i.e., $\sum_{i=1}^{n+1} |x_i| = 1$. If we consider $|x_i|$ as the length of the piece X_i, i.e., $|x_i| = |t_{i-1} - t_i|$, and assign the piece X_i to the positive portion O_+ (O_-) if $x_i > 0$ ($x_i < 0$), respectively, then any such a point $\mathbf{x}$ corresponds to a way of cutting the object using n cuts and combining the $n + 1$ pieces into two portions O_+ and O_-.

Simons and Su [25] prove Lemma 2 by employing the Borsuk-Ulam Theorem. Given any point $\mathbf{x}$ on the n-dimensional cross-polytope, one can define the function f according to the valuation functions of the n agents. That is, the i-th coordinate of the image of $f(\mathbf{x})$ is equal to the value of the i-th agent over the positive portion O_+. It is easy to see that the i-th coordinate of $f(-\mathbf{x})$ is equal to the value of i-th agent over the negative portion O_-. So either of the antipodal points satisfying $f(\mathbf{x}) = f(-\mathbf{x})$ corresponds to a solution to the consensus-halving problem. With some further construction steps and using any path-following algorithm in the proof of Tucker's Lemma, Simons and Su [25] present a protocol for locating a solution to the

ϵ-approximate consensus-halving problem (which actually works even if the continuity condition is relaxed).

We define the following search problem, called (n, ϵ)-CHalving.

(n, k, ϵ)-CHalving
Input: The value density functions $v_i : O \to R_+, i = 1, \cdots, n$, for n agents.
Output: A partition (O_+, O_-) with k cuts such that $\lvert u_i(O_+) - u_i(O_-)\rvert \leq \epsilon$ for all agents $i \in N$.

We will also consider the following decision problem, called $(n, n-1, \epsilon)$-CHalving. Note that for n agents and $n-1$ cuts, a solution to ϵ-approximate consensus-halving problem is not guaranteed to exist.

$(n, n-1, \epsilon)$-CHalving
Input: The value density functions $v_i : O \to R_+, i = 1, \cdots, n$, for n agents.
Output: Yes, if a partition (O_+, O_-) with $n-1$ cuts such that $\lvert u_i(O_+) - u_i(O_-)\rvert \leq \epsilon$ for all agents $i \in N$ exists, and No otherwise.

Finally, we define the computational version of *Continuous Necklace Splitting* [18,2]; the Continuous Necklace Splitting Problem is similar to Consensus-Halving but instead of two portions that each agent values equally, the goal is to create t such portions.

(n, k, ϵ)-ConNeckSplit
Input: The value density functions $v_i : O \to R_+, i = 1, \cdots, n$, for n agents.
Output: A partition $(O_1, O_2, \ldots O_t)$ with $k \cdot (t-1)$ cuts such that $\lvert u_i(O_\ell) - u_i(O_j)\rvert \leq \epsilon$ for all ℓ and j and for all agents $i \in N$.

2.3 PPA, PPAD and Total Search Problems

Most of the problems that we will consider in this paper belong to the class of *total search problems*, i.e. search problems for which a solution is guaranteed to exist, regardless of the input. In particular, we will be interested in problems in the class TFNP, i.e. total search problems for which a solution is verifiable in polynomial time. Formally, a binary relation $P(x, y)$ is in the class TFNP if for every x, there exists a y of size bounded by a polynomial in $|x|$ such that $P(x, y)$ holds and $P(x, y)$ can be verified in polynomial time. The problem is given x, to find such a y in polynomial time.

An important subclass of TFNP is the class PPAD, defined by Papadimitriou in [20]. PPAD stands for "Polynomial Parity Argument on a Directed graph" and is defined formally in terms of the problem End-Of-Line:

End-of-Line
Input: Two boolean circuits S (for successor) and P (for predecessor) with n inputs and n outputs such that $P(0^n) = 0^n \neq S(0^n)$.
Output: A vertex x such that $P(S(x)) \neq x$ or $S(P(x)) \neq 0^n$.

A problem is *in* PPAD if it is polynomial-time reducible to End-Of-Line and it is PPAD-*complete* if End-Of-The-Line reduces to it in polynomial-time.

Given the definition above, we can observe that PPAD is defined in terms of an exponentially large digraph $G = (V, E)$ consisting of 2^n vertices with indegree and outdegree at most 1. An edge between vertices v_1 and v_2 is present in E if and only if $S(v_1) = v_2$ and $P(v_2) = v_1$. By construction, the point 0^n has indegree 0 and we are looking for a point with outdegree 0, which is guaranteed to exist. Note that the graph is given *implicitly* to the input, through a function that given any vertex v, returns its set of neighbours (predecessor and successor) in polynomial time.

PPAD is a subclass of the class PPA ("Polynomial Parity Argument") which is defined similarly, but in terms of an undirected graph in which every vertex has degree at most 2, and given a vertex of degree 1, we are asked to find another vertex of degree 1. The computational problem associated with the class is called Leaf, which is defined similarly:

> Leaf
>
> **Input:** A boolean circuit C with n inputs and at most $2n$ outputs, outputting the set $\mathcal{N}(y)$ of (at most two) neighbours of a vertex y, such that $|\mathcal{N}(0^n)| = 1$.
>
> **Output:** A vertex x such that $x \neq 0^n$ and $|\mathcal{N}(x)| = 1$.

A problem is the class PPA if it is polynomial-time reducible to Leaf.

2.4 Generalized Circuits

A *generalized circuit* $S = (V, \mathcal{T})$ consists of a set of nodes V and a set of gates $\mathcal{T}$ and let $N = |V|$ and $M = |\mathcal{T}|$. Every gate $T \in \mathcal{T}$ is a 5-tuple $T = (G, v_{in_1}, v_{in_2}, v_{out}, \alpha)$ where

- $G \in \{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\vee, G_\wedge, G_\neg\}$ is the type of the gate,
- $v_{in_1}, v_{in_2} \in V \cup \{nil\}$ are the first and second input nodes of the gate or *nil* if not applicable,
- $v_{out} \in V$ is the output node, and $\alpha \in [0,1] \cup \{nil\}$ is a parameter if applicable,
- for any two gates $T = (G, v_{in_1}, v_{in_2}, v_{out}, \alpha)$ and $T' = (G', v'_{in_1}, v'_{in_2}, v'_{out}, \alpha')$ in $\mathcal{T}$ where $T \neq T'$, they must satisfy $v_{out} \neq v'_{out}$.

Note that generalized circuits extend the standard boolean or arithmetic circuits in the sense that generalized circuits allow cycles in the directed graph defined by the nodes and gates. We define the search problem ϵ-Gcircuit [9,24]:

> ϵ-Gcircuit
>
> **Input:** A generalized circuit $S = (V, \mathcal{T})$.
>
> **Output:** A vector $\mathbf{x} \in [0,1]^N$ of values for the nodes, satisfying the conditions from Table 1.

Note that a solution to ϵ-Gcircuit always exists [9] and hence the problem is well-defined. Also, notice that for the logic gates $G_\vee, G_\wedge$ and $G_\neg$, if the input conditions are not fulfilled, the output is unconstrained, and for the multiplication gate, it holds that $\alpha \in (0,1]$. ϵ-Gcircuit was proven to be PPAD-complete implicitly or explicitly in [11,9] for inversely polynomial error ϵ and in [24] for constant ϵ. We state the latter theorem here as a lemma:

Lemma 3 ([24]). *There exists a constant $\epsilon > 0$ such that ϵ-Gcircuit is PPAD-complete.*

2.5 Roadmap

Here is the roadmap for the rest of the paper.

- In Section 3, we prove that (n, n, ϵ)-CHalving is contained in the class PPA. We obtain the result by a reduction to (n, τ)-Tucker, triangulated on the unit-sphere S^n. En route to the result, we prove that (n, τ)-Tucker on the unit-sphere is in PPA. In particular, we use a triangulation of the unit sphere that is *aligned with a flag of hemispheres* and explain how a constructive proof of Fan's combinatorial lemma proposed by Prescott and Su [22] can be converted into a reduction to Leaf. Then, we use the construction of Simmons and Su [25] to obtain a reduction from (n, n, ϵ)-CHalving to (n, τ)-Tucker.

Table 1. Gate constraint $T = (G, v_{in_1}, v_{in_2}, v_{out}, \alpha)$

Gate	Constraint
$(G_\zeta, nil, nil, v_{out}, \alpha)$	$\alpha - \epsilon \le \mathbf{x}[v_{out}] \le \alpha + \epsilon$
$(G_{\times\zeta}, v_{in_1}, nil, v_{out}, \alpha)$	$\alpha \cdot \mathbf{x}[v_{in_1}] - \epsilon \le \mathbf{x}[v_{out}] \le \alpha \cdot \mathbf{x}[v_{in_1}] + \epsilon$
$(G_=, v_{in_1}, nil, v_{out}, nil)$	$\mathbf{x}[v_{in_1}] - \epsilon \le \mathbf{x}[v_{out}] \le \mathbf{x}[v_{in_1}] + \epsilon$
$(G_+, v_{in_1}, v_{in_2}, v_{out}, nil)$	$\min(\mathbf{x}[v_{in_1}] + \mathbf{x}[v_{in_2}], 1) - \epsilon \le \mathbf{x}[v_{out}] \le \min(\mathbf{x}[v_{in_1}] + \mathbf{x}[v_{in_2}], 1) + \epsilon$
$(G_-, v_{in_1}, v_{in_2}, v_{out}, nil)$	$\max(\mathbf{x}[v_{in_1}] - \mathbf{x}[v_{in_2}], 0) - \epsilon \le \mathbf{x}[v_{out}] \le \max(\mathbf{x}[v_{in_1}] - \mathbf{x}[v_{in_2}], 0) + \epsilon$
$(G_<, v_{in_1}, v_{in_2}, v_{out}, nil)$	$\mathbf{x}[v_{out}] = \begin{cases} 1 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] < \mathbf{x}[v_{in_2}] - \epsilon; \\ 0 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] > \mathbf{x}[v_{in_2}] + \epsilon. \end{cases}$
$(G_\vee, v_{in_1}, v_{in_2}, v_{out}, nil)$	$\mathbf{x}[v_{out}] = \begin{cases} 1 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 1 \pm \epsilon \text{ or } \mathbf{x}[v_{in_2}] = 1 \pm \epsilon; \\ 0 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 0 \pm \epsilon \text{ and } \mathbf{x}[v_{in_2}] = 0 \pm \epsilon. \end{cases}$
$(G_\wedge, v_{in_1}, v_{in_2}, v_{out}, nil)$	$\mathbf{x}[v_{out}] = \begin{cases} 1 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 1 \pm \epsilon \text{ and } \mathbf{x}[v_{in_2}] = 1 \pm \epsilon; \\ 0 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 0 \pm \epsilon \text{ or } \mathbf{x}[v_{in_2}] = 0 \pm \epsilon. \end{cases}$
$(G_\neg, v_{in_1}, nil, v_{out}, nil)$	$\mathbf{x}[v_{out}] = \begin{cases} 1 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 0 \pm \epsilon; \\ 0 \pm \epsilon, & \text{if } \mathbf{x}[v_{in_1}] = 1 \pm \epsilon. \end{cases}$

- In Section 4, we prove that $(n, n+k, \epsilon)$-CHalving for constant k and constant ϵ is PPAD-hard. The result is established via a reduction from (ϵ')-Gcircuit, and the high level idea is the following. For each node of a (ϵ')-Gcircuit instance, we associate two agents in (n, ϵ)-CHalving and a designated interval from which the values of the node will be "read", given the position of a cut. For each gate of the circuit, we construct "value" gadgets on the consensus-halving instance, and associate a value gadget corresponding to a gate with the interval corresponding to the output node of the gate. We establish a construction that ensures that a Consensus-Halving partition of the interval corresponds to a valid assignment of values to the nodes, respecting the gate constraints. To extend the result to $n+k$ cuts, we create k copies of the original circuit and apply a similar construction. Finally, we explain how the reduction can be adapted to work for valuations which are probability measures and state the implication for the complexity of the Continuous Necklace Splitting problem when $t = 2$.

- Finally, in Section 5, we prove that $(n, n-1, \epsilon)$-CHalving is NP-hard. We establish the result via a reduction from 3SAT. Briefly, we will construct an instance of $(n, n-1, \epsilon)$-CHalving with $n = 2k+1$ agents (where k is the number of input variables of the 3SAT formula); the $2k$ agents will implicitly encode a generalized circuit, implementing the value gadgets corresponding to "AND" and "OR" gates of the boolean formula. The $(2k+1)$'th agent will have value in two distinct intervals, and will only be satisfied with a partition of $n-1$ cuts, if the 3SAT formula is satisfied and vice-versa.

3 (n, n, ϵ)-CHalving is in PPA

In this section, we prove that (n, n, ϵ)-CHalving is in PPA by a reduction to Leaf via (n, τ)-Tucker. So en route to our result, we prove that (n, τ)-Tucker is also in PPA. We are interested in triangulations that satisfy the following property.

Definition 2 (Aligned with hemispheres Triangulation). *[22] A* flag of hemispheres *in S^n is a sequence $H_0 \subset \cdots \subset H_n$ where each H_d is homeomorphic to a d-ball, and for $1 \le d \le n$, $\partial H_d = \partial(-H_d) = H_d \cap (-H_d) = H_{d-1} \cup (-H_{d-1}) \approxeq S^{d-1}$, $H_n \cap (-H_n) = S^n$, and $H_0, -H_0$ are antipodal points. A symmetric triangulation T of S^n is said to be* aligned with hemispheres *if we can find a flag of hemispheres such that H_d is contained in the d-skeleton of the triangulation.*

Note that the same definition can be adapted to our context by converting l_2-norm to l_1-norm. Our triangulation T defined in Definition 1 is aligned with hemispheres because the d-skeleton of the triangulation is cut by orthant hyperplanes and we can find a flag of hemispheres such that H_d is contained in it.

We now proceed to the first step of the proof, establishing the reduction from (n, τ)-Tucker to Leaf.

Lemma 4. (n,τ)-TUCKER *is in PPA.*

Proof. We reduce (n,τ)-TUCKER to the PPA problem LEAF, by using the proof by Prescott and Su [22]. In [22], the authors present a constructive proof of Fan's combinatorial lemma on labelings of triangulated spheres. Fan's lemma says that given a symmetric barycentric subdivision of the octahedral subdivision of the n-sphere S^n and a labelling from its vertices to $\{\pm 1,\dots,\pm m\}$, where $m \geq n+1$, such that labels at antipodal vertices sum to 0 and labels at adjacent vertices do not sum to 0, then there are an odd number of n-simplices whose labels are of the form

$$\{k_0, -k_1, k_2, \dots, (-1)^n k_n\}, \quad \text{where} \;\; 1 \leq k_0 < k_1 < \dots < k_n \leq m.$$

Their proof generalizes Fan's lemma in the sense that the result holds for a larger class of triangulations of S^n, that is, the lemma holds for any symmetric triangulation of S^n that is aligned with a flag of hemispheres. Their proof also enables us to start from any place on the sphere to construct a path whose endpoint is a top dimension simplex of the form $\{k_0, -k_1, k_2, \dots, (-1)^n k_n\}$. Fan's lemma is a generalization of Tucker's lemma in the sense that if less labels are allowed, i.e., $m = n$, then inevitably there will be adjacent vertices labeled by opposite colors. We sketch the proof of [22] and show how it can be converted into a reduction from (n,τ)-TUCKER to LEAF.

Given a triangulation aligned with hemispheres, we label the vertices of the triangulation as stated in Fan's lemma. The *carrier hemisphere* of a simplex σ in T is the minimal H_d or H_d that contains σ. A simplex is *alternating* if its vertex labels are distinct in magnitude and alternate in sign when arranged in monotone order of magnitude, i.e., the labels have the form

$$\{k_0, -k_1, k_2, \dots, (-1)^n k_n\} \;\; \text{or} \;\; \{-k_0, k_1, -k_2, \dots, (-1)^{n+1} k_n\}.$$

The *sign* of an alternating simplex is the sign of its smallest label. A simplex is *almost-alternating* if it is not alternating, but by deleting one of the vertices, the resulting simplex (a facet) is alternating. The sign of an almost-alternating simplex is defined to be the sign of any of its alternating facets. Thus any almost-alternating simplex must have exactly two facets that are alternating. Call an alternating or almost-alternating simplex *agreeable* if the sign of that simplex matches the sign of its carrier hemisphere.

Now we define a graph G. A simplex σ carried by H_d is a node of G if it is one of the following: (1) an agreeable alternating $(d-1)$-simplex; (2) an agreeable almost-alternating d-simplex; or (3) an alternating d-simplex. Two nodes σ and η are adjacent in G if all of the following hold: (a) one is the facet of the other, (b) $\sigma \cap \eta$ is alternating, and (c) the sign of the carrier hemisphere of $\sigma \cup \eta$ matches the sign of $\sigma \cap \eta$. Prescott and Su show that G is a graph in which every vertex has degree 1 or 2. Furthermore, a vertex has degree 1 if and only if its simplex is carried by $\pm H_d$ or is an n-dimensional alternating simplex.

To see how Fan's lemma implies Tucker's lemma and how Prescott and Su's proof can be converted to a reduction to LEAF, we now restrict our attention to the case when $m = n$. Since there are not enough labels for the existence of alternating n-simplices, there must exist some agreeable almost-alternating simplices with a complementary edge. We add those simplices as nodes to the graph G; it is easy to see that these nodes will be of degree 1. Since the path considered in [22] can start from any vertex, we can choose any vertex H_0 as the given degree 1 node in graph G. Following the path, the other degree 1 node in G corresponds to the almost-alternating simplex with a complementary edge. The graph G clearly only contains degree nodes of degree 1 or 2. □

We are now ready to prove our PPA containment result.

Theorem 2. (n,n,ϵ)-CHALVING *is in PPA.*

Proof. We reduce (n,n,ϵ)-CHALVING to (n,τ)-TUCKER. Given any instance of (n,n,ϵ)-CHALVING, we construct an instance of (nD,τ)-TUCKER based on the construction in [25]. We note that the coordinates of any vertex $\mathbf{x} \in C^n$ naturally correspond to a partition that uses n cuts on the $[0,1]$ interval.[6] This is because the coordinates of any vertex $\mathbf{x} \in C^n$ satisfy $\sum_{i=1}^{n+1} |x_i| = 1$, and a partition with n cuts on $[0,1]$ can

[6] The use of the $[0,1]$ interval is for convenience and without loss of generality; for any choice of the interval we could use a cross-polytope corresponding to a sphere of a different radius.

be interpreted as partitioning the interval into $n+1$ pieces such that the length of each piece is equal to $|x_i|, i = 1, \ldots, n+1$. Furthermore, if the sign of the i-th coordinate x_i is "+", piece $|x_i|$ is assigned to portion O_+; otherwise it is assigned to portion O_-.

We will use the triangulation T of C^n described in Definition 1 with $\tau = \epsilon/2M$, where M is the bound of the valuation function on any subinterval. It is easy to verify that for any two adjacent vertices $\mathbf{x}$ and $\mathbf{x}'$ (denote their associated portions by O_+, O_-, and O'_+,O'_-, respectively), it holds that

$$|v_i(O_+) - v_i(O'_+)| < \epsilon/2, \quad \text{for all } \ i \in [n].$$

We now label all the vertices on $T \cap C^n$. For any vertex $\mathbf{x} \in T \cap C^n$, denote

$$l = \arg\max_i \{|v_i(O_+) - v_i(O_-)|\},$$

and then label $\mathbf{x}$ by $+l$ if $v_l(O_+) - v_l(O_-) > 0$; label $\mathbf{x}$ by $-l$ if $v_l(O_+) - v_l(O_-) < 0$. Note that in case $v_l(O_+) - v_l(O_-) = 0$ then $\mathbf{x}$ corresponds to a solution of (n, n, ϵ)-CHalving. We claim that this labeling satisfies the boundary condition of Lemma 1. In summary, given an instance of (n, n, ϵ)-CHalving, we have constructed an instance of (n, τ)-Tucker.

Now, given a solution to (n, τ)-Tucker, i.e., two adjacent vertices $\mathbf{z}$ and $\mathbf{z}'$ that are assigned opposite labels, assume without loss of generality that $\mathbf{z}$ (with associated portions O_+ and O_-) is labelled by k and $\mathbf{z}'$ (with associated portions O'_+ and O'_-) is labelled by $-k$. According to the labelling, it holds that $v_k(O_+) - v_k(O_-) > 0$ and $v_k(O'_+) - v_k(O'_-) < 0$. Therefore, for all $i \in [n]$ it holds that

$$\begin{aligned}
|v_i(O_+) - v_i(O_-)| \le |v_k(O_+) - v_k(O_-)| &\le |(v_k(O_+) - v_k(O_-)) - (v_k(O'_+) - v_k(O'_-))| \\
&= |(v_k(O_+) - v_k(O'_+)) - (v_k(O_-) - v_k(O'_-))| \\
&\le |v_k(O_+) - v_k(O'_+)| + |v_k(O_-) - v_k(O'_-)| \\
&\le \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.
\end{aligned}$$

This means that the partition corresponding to the point $\mathbf{z}$ is a valid solution to (n, n, ϵ)-CHalving. □

4 $(n, n+k, \epsilon)$-CHalving is PPAD-hard

In this section, we will prove that finding an approximate partition for Consensus-Halving using n cuts is PPAD-hard, even if the allowed discrepancy between the two portions is a constant. In fact, we will extend our construction to show that the problem is still PPAD-hard, even if we are allowed to use $n+k$ cuts, for any constant k. First, we will prove that (n, n, ϵ)-CHalving is PPAD-hard; we will explain how to extend the result to $n+k$ cuts later on.

We describe the reduction from ϵ-Gcircuit that we will be using for the PPAD-hardness proof. Given an instance $S = (V, \mathcal{T})$ of ϵ-Gcircuit, we will construct an instance of (n, n, ϵ')-CHalving with $n = 2N$ agents, in which each node $v_i \in V$ of the circuit will correspond to two agents var_i and $copy_i$ and where ϵ' will be defined later. As a matter of convenience in the reduction, we will assume that for every gate $(G, v_{in_1}, v_{in_2}, v_{out}, \alpha)$ in $\mathcal{T}$, v_{in_1}, v_{in_2} and v_{out} are distinct. This does not affect the hardness of the problem as any ϵ-generalized circuit can be converted to this form by use of at most $2N$ additional equality-gates and nodes, and also since an $(\epsilon/2)$-approximate solution to the converted problem can clearly be converted to a solution in the original circuit.

For ease of notation we consider a CHalving instance on the interval $[0, 6N]$. Let $d_i := 6(i-1)$; the two agents var_i and $copy_i$ that we construct for every node v_i have valuations

$$var_i = \begin{cases} border_i(t) + G^\tau(t), & \text{if } v_i \text{ is the output of } \tau \\ border_i(t), & \text{otherwise} \end{cases}$$

$$copy_i = \begin{cases} 4, & t \in [d_i+3, d_i+4] \cup [d_i+5, d_i+6] \\ 1, & t \in [d_i+1, d_i+2] \cup [d_i+4, d_i+5] \\ 0, & \text{otherwise} \end{cases}$$

$$\text{where} \quad border_i = \begin{cases} 4, & \text{if } t \in [d_i, d_i+1] \cup [d_i+2, d_i+3] \\ 0, & \text{otherwise} \end{cases}$$

Since each node is the output of at most one gate, var_i is well-defined. Note that apart from the valuation defined by the function G^τ, agents var_i and $copy_i$ only have valuations on the sub-interval $[d_i, d_{i+1}]$, i.e., the agents associated with node v_1 only have valuations on $[0, 6]$, the agents associated with v_2 only on have valuations on $[6, 12]$ and so on. Let $v_i^- := [d_i+1, d_i+2]$ and the right and left endpoints respectively be $v_{i,\ell}^-$ and $v_{i,r}^-$, (and analogously for $v_i^+ := [d_i+3, d_i+4]$, $v_{i,\ell}^+$ and $v_{i,r}^+$). Now, we are ready to define the functions G^τ according to Table 2. Notice that because of the assumption that the two input nodes and the output node are distinct, the graphs of the functions are as in Table 2. Figure 4 demonstrates an example of a Consensus-Halving instance corresponding to a small circuit.

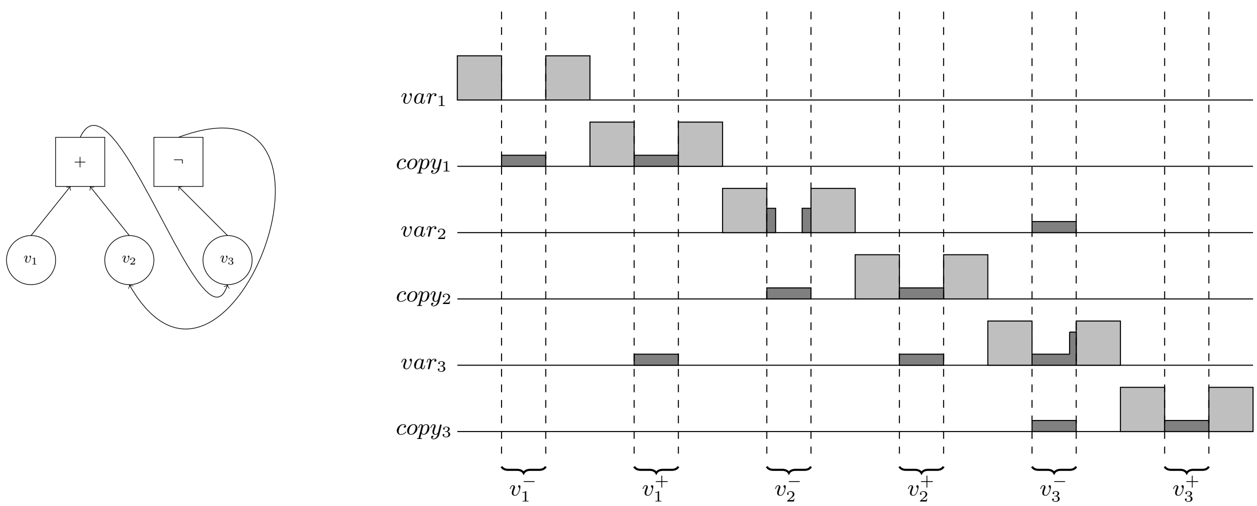


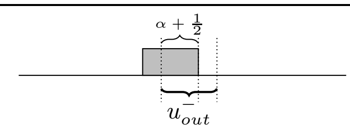
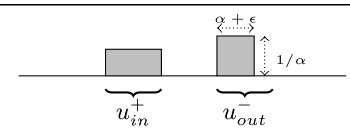
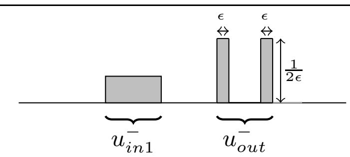
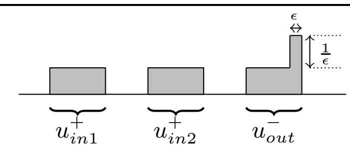
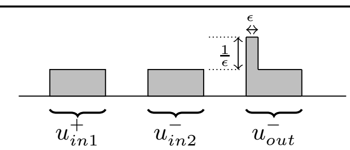
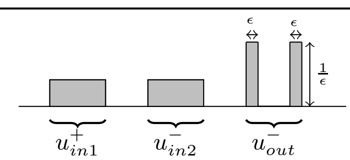
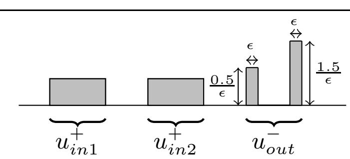
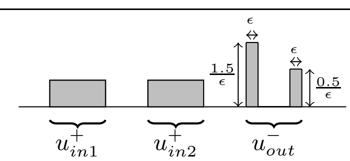
Fig. 1. An instance of ϵ-GCIRCUIT with the corresponding construction for (n, ϵ')-CHALVING.

Lemma 5. *Given the construction of a (n, n, ϵ')-CHALVING instance above, a partition with n cuts corresponds to a solution to ϵ-GCIRCUIT.*

Proof. First observe that since all of the agents var_i and $copy_i$ are constructed to have at least 3/4 of their valuation on $[d_i, d_i+3]$ and $[d_i+3, d_i+6]$ respectively, there must be at least one cut in each one of those intervals in any ϵ'-approximate solution to Consensus-Halving (with $\epsilon' < 1/4$) and therefore any ϵ'-approximate solution to Consensus-Halving with $2N$ cuts must have exactly one cut in each interval. Furthermore, since the constructed instance consists of $2N$ agents, by Lemma 2, such a partition with $2N$ cuts is guaranteed to exist.

Now consider such a solution $\mathcal{C}$ to (n, n, ϵ')-CHalving with $2N$ cuts. For each agent var_i (and associated gate G^τ, if any), since her valuation in v_i^- is at least the same as her valuation outside the interval $[d_i, d_i+3]$, the cut from $\mathcal{C}$ in $[d_i, d_i+3]$ must be in $[d_i+1-\epsilon', d_i+2+\epsilon']$, since $\mathcal{C}$ is a solution to (n, n, ϵ')-CHalving. We will assume without loss of generality that the leftmost piece of the partition $\mathcal{C}$ is in O_-. Notice then that for each node v_i, the piece on the left-hand side of the cut in v_i^- is always in O_- and the piece on the left-hand side of the cut in v_i^+ is always in O_+. Let the location of the cut be $d_i + 1 + t_i^-$ where $t_i^- \in [-\epsilon', 1+\epsilon']$.

Table 2. Agent preferences from gate $\tau = (G, v_{in_1}, v_{in_2}, v_{out}, \alpha)$. For the gate $G_{\times\zeta}$, the figure depicts the case when $\alpha + \epsilon < 1$.

	$G^\tau(t)$	Picture
G_ζ	$\begin{cases} 1 & \text{if } t \in [v^-_{out,\ell} + \alpha - \frac{1}{2}, v^-_{out,\ell} + \alpha + \frac{1}{2}] \\ 0 & \text{otherwise} \end{cases}$	
$G_{\times\zeta}$	$\begin{cases} 1 & \text{if } t \in v^+_{in} \\ 1/\alpha & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \min(\alpha+\epsilon, 1)] \\ 0 & \text{otherwise} \end{cases}$	
$G_\neg$	$\begin{cases} 1 & \text{if } t \in v^-_{in} \\ 1/2\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 1/2\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
G_+	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^+_{in_2} \\ 1 & \text{if } t \in [v^-_{out,\ell}, v^-_{out,r} - \epsilon] \\ 1/\epsilon + 1 & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
G_-	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^-_{in_2} \\ 1 & \text{if } t \in [v^-_{out,\ell} + \epsilon, v^-_{out,r}] \\ 1/\epsilon + 1 & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 0 & \text{otherwise} \end{cases}$	
$G_<$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^-_{in_2} \\ 1/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 1/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
$G_\vee$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^+_{in_2} \\ 0.5/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 1.5/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
$G_\wedge$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^+_{in_2} \\ 1.5/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 0.5/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	

Analogously, the same argument holds for agent $copy_i$ and the interval $[d_i + 3 - \epsilon', d_i + 4 + \epsilon']$, and define $t_i^+ \in [-\epsilon', 1 + \epsilon']$ similarly.

Now consider the agent $copy_i$ and the cut at location $d_i + 1 + t_i^-$. If $t_i^- \in [0, 1]$, then since agent $copy_i$ has valuation 1 on interval v_i^-, t_i^- of her valuation will be on a piece in O_- and $1 - t_i^-$ of her valuation will be on a piece in O_+. Then, since $\mathcal{C}$ is a solution to (n, n, ϵ')-CHalving, the cut in $d_i + 3 + t_i^+$ must be placed so that $|t_i^- - t_i^+| \leq \epsilon'/2$; similarly for the cases where $t_i^- \notin [0, 1]$. In other words, $copy_i$ ensures that the cut at $d_i + 1 + t_i^-$ is "copied" ϵ'-approximately.

We will interpret the solution $\mathcal{C}$ as a solution to ϵ-Gcircuit in the following way. For each node v_i and each associated cut at $d_i + 1 + t_i^-$ let

$$x_i := \begin{cases} 0 & , \quad t_i^- < 0 \\ t_i^- & , \quad t_i^- \in [0, 1] \\ 1 & , \quad t_i^- > 1 \end{cases} \tag{1}$$

and notice

$$|t_i^+ - x_i| \leq 2\epsilon' \quad , \quad |t_i^- - x_i| \leq 2\epsilon' \tag{2}$$

Now to complete the proof, we just need to argue that these variables satisfy the constraints of the gates of the circuit.

Constant gate $\tau = (G_\zeta, nil, nil, v_{out}, \alpha)$: The valuation of agent var_{out} for the intervals $[d_i, d_i + 1 + \alpha]$ and $[d_i + 1 + \alpha, d_i + 3]$ is the same and since the height of the agent's value density function is at least 1 in $[d_i, d_i + 3]$,[7] it holds that $t_{out}^- \in [\alpha - \epsilon', \alpha + \epsilon']$. Then, by Equation 2, it holds that $x_{out} \in [\alpha - 3\epsilon', \alpha + 3\epsilon']$, so for $\epsilon' < \epsilon/3$ the gate constraint is satisfied.

Multiplication-by-scalar gate $\tau = (G_{\times\zeta}, v_{in}, nil, v_{out}, \alpha)$. Notice that for any given cut t_{in}^+ and $1 - \alpha \geq \epsilon$, it holds that $t_{out}^- \in [\alpha t_{in}^+ + \epsilon/2 - \epsilon', \alpha t_{in}^+ + \epsilon/2 + \epsilon']$ as the height of G^τ in v_{out}^- is at least 1. Similarly, for the case $1 - \alpha < \epsilon$ and any given cut t_{in}^+, it holds that $t_{out}^- \in [\alpha t_{in}^+ + (1-\alpha)/2 - \epsilon', \alpha t_{in}^+ + (1-\alpha)/2 + \epsilon']$ as the height of G^τ in v_{out}^- is at least 1. In particular, since $1 - \alpha < \epsilon$, it also holds that $t_{out}^- \in [\alpha t_{in}^+ + \epsilon/2 - \epsilon', \alpha t_{in}^+ + \epsilon/2 + \epsilon']$ for this case as well. Then, by Equation 2, it holds that $x_{out} \in [\alpha t_{in}^+ + \epsilon/2 - 3\epsilon', \alpha t_{in}^+ + \epsilon/2 + 3\epsilon']$ and since $\alpha \leq 1$ it also holds that $x_{out} \in [\alpha x_{in} + \epsilon/2 - 5\epsilon', \alpha x_{in} + \epsilon/2 + 5\epsilon']$, again by Equation 2. Then the gate constraint is satisfied whenever $\epsilon' < \epsilon/10$.

Addition gate $\tau = (G_+, v_{in_1}, v_{in_2}, v_{out}, nil)$. If for the cuts $t_{in_1}^+$ and $t_{in_2}^+$ it holds that $t_{in_1}^+ + t_{in_2}^+ < 1 - \epsilon + 4\epsilon'$ then $t_{out}^- \in [t_{in_1}^+ + t_{in_2}^+ - 5\epsilon', t_{in_1}^+ + t_{in_2}^+ + 5\epsilon']$ as the height of G^τ in v_{out}^- is at least 1. This then implies that $x_{out} \in [x_{in_1}^+ + x_{in_2}^+ - 11\epsilon', x_{in_1}^+ + x_{in_2}^+ + 11\epsilon']$, by Inequality 2. On the other hand, when $t_{in_1}^+ + t_{in_2}^+ \geq 1 - \epsilon + 4\epsilon'$, then by Definition 1, it holds that $x_{in_1} + x_{in_2} \in [1 - \epsilon, 1]$ and clearly $t_{out}^- \in [1 - \epsilon, 1 + \epsilon']$ which by Definition 1 implies that $x_{out} \in [1 - \epsilon, 1]$. The gate constraints are satisfied for $\epsilon' < \epsilon/11$ for each of the cases.

Subtraction gate $\tau = (G_-, v_{in_1}, v_{in_2}, v_{out}, nil)$. Analogously to the addition gate described above, when for the cuts $t_{in_1}^+$ and $t_{in_1}^+$ it holds that $t_{in_1}^+ - t_{in_2}^+ > \epsilon - 4\epsilon'$ then $t_{out}^- \in [t_{in_1}^+ - t_{in_2}^+ - 5\epsilon', t_{in_1}^+ - t_{in_2}^- + 5\epsilon']$ as the height of G^τ in v_{out}^- is at least 1. This implies that $x_{out} \in [x_{in_1}^+ - x_{in_2}^- - 11\epsilon', x_{in_1}^+ - x_{in_2}^- + 11\epsilon']$ by Inequality 2. On the other hand when $t_{in_1}^+ - t_{in_2}^- \leq \epsilon - \epsilon'$, which implies that $x_{in_1} + x_{in_2} \in [0, \epsilon]$ by Definition 1, it clearly holds that $t_{out}^- \in [-\epsilon', \epsilon]$ and hence by Definition 1, we have $x_{out} \in [0, \epsilon]$. The gate constraints are satisfied for $\epsilon' < \epsilon/11$ for each of the cases.

Less-than-equal gate $\tau = (G_<, v_{in_1}, v_{in_2}, v_{out}, nil)$. We will consider three cases, depending on the positions of the cuts $t_{in_1}^+$ and $t_{in_2}^-$. First, when $|t_{in_1}^+ - t_{in_2}^-| < \epsilon - 4\epsilon'$, by Inequality 2 it holds that $|x_{in_1} - x_{in_2}| < \epsilon$ and the output of the gate is unconstrained. When $t_{in_1}^+ - t_{in_2}^- \geq \epsilon - 4\epsilon'$ then by Inequality 2 it holds

[7] Notice that the constant gate is the only gate where $border_i$ and G^τ overlap.

that $x_{in_1} \geq x_{in_2} + \epsilon$. Additionally, since the height of G^τ in $[v^-_{out,r} - \epsilon, v^-_{out,r}]$ is at least 1, it holds that $t^-_{out} \in [1-\epsilon, 1+\epsilon']$, which by Definition 1 implies that $x^-_{out} \in [1-\epsilon, 1]$ and the gate constraint is satisfied. The argument for the case when $t^-_{in_2} > t^+_{in_1} - 2\epsilon'$ is completely symmetrical.

Logic OR gate $\tau = (G_\vee, v_{in_1}, v_{in_2}, v_{out}, nil)$. We will consider three cases depending on the position of the cuts $t^+_{in_1}$ and $t^+_{in_2}$. First, when $t^+_{in_1} + t^+_{in_2} < 0.4$ it holds that $t^-_{out} \in [-\epsilon', \epsilon]$ and hence by Definition 1, it holds that $x_{out} \in [0, \epsilon]$. Furthermore, by Inequality 2 it holds that $x_{in_1} + x_{in_2} < 0.4 + 4\epsilon'$ and for $\epsilon' < 1/40$, it also holds that $x_{in_1}, x_{in_2} < 0.5$ and the gate constraint is satisfied. Next, when $t^+_{in_1} + t^+_{in_2} \in [0.4, 0.8]$ then by Inequality 2, it holds that $x_{in_1}, x_{in_2} \in [0.4 - \epsilon', 0.8 + 4\epsilon']$ and in particular, when $\epsilon' < 1/40$ then it also holds that $x_{in_1} + x_{in_2} \in [0.3, 0.9]$ and the output of the gate in unconstrained. Finally when $t^+_{in_1} + t^+_{in_2} > 0.8$, it holds that $t^-_{out} \in [1-\epsilon, 1+\epsilon']$ and hence by Definition 1, we have that $x_{out} \in [1-\epsilon, 1]$. Furthermore, by Inequality 2 we have that $x_{in_1} + x_{in_2} > 0.8 + 4\epsilon'$ which is greater than 0.9 when $\epsilon' < 1/40$ which implies that at least one of the two inputs is greater than ϵ. In particular, the gate's output lies in $[1-\epsilon, \epsilon]$ when the inputs are smaller than ϵ or greater than $1-\epsilon$ and at least one of them is greater than $1-\epsilon$. This shows that the gate constraint is satisfied.

Logic AND gate $\tau = (G_\wedge, v_{in_1}, v_{in_2}, v_{out}, nil)$. Analogously to the Logical OR gate, we will consider three cases, depending on the position of the cuts $t^+_{in_1}$ and $t^+_{in_2}$. First, when $t^+_{in_1} + t^+_{in_2} < 1.4$, it holds that $t^-_{out} \in [-\epsilon', \epsilon]$ and by Inequality 1, we have that $x_{out} \in [0, \epsilon]$. On the other hand, by Inequality 2, we have that $x_{in_1} + x_{in_2} < 1.4 + 4\epsilon'$ which is at most 1.5 for $\epsilon' < 1/40$ and the gate constraint is satisfied for the same reason as in the third case of the Logic OR gate above (using the argument symmetrically, for values smaller than ϵ instead of at larger than $1-\epsilon$). Next, when $1.4 \leq t^+_{in_1} + t^+_{in_2} \leq 1.8$, by Inequality 2 and for $\epsilon' < 1/40$, it holds that $x_{in_1} + x_{in_2} \in [1.3, 1.9]$ and the output of the gate is unconstrained. Finally, when $t^+_{in_1} + t^+_{in_2} > 1.8$, by Inequality 2 and for $\epsilon' < 1/40$, it holds that $x_{in_1} + x_{in_2} > 1.7$. Furthermore, it holds that $t^-_{out} \in [1-\epsilon, 1+\epsilon']$ and hence by Definition 1, we have that $x_{out} \in [1-\epsilon, 1]$ and the gate constraint is satisfied.

Logic NOT gate $\tau = (G_\neg, v_{in}, v_{in_2}, v_{out}, nil)$. We will consider three cases, depending on the location of the cut t^-_{in}. First, if $t^-_{in} < 0.4$, it holds that $t^-_{out} \in [-\epsilon', \epsilon]$ and hence by Definition 1, we have that $x_{out} \in [0, \epsilon]$. Furthermore, by Inequality 2, it holds that $x_{in} < 0.4 + 4\epsilon'$ which is at most 0.5 when $\epsilon' < 1/40$ and the gate constraint is satisfied because for any value of the input smaller than ϵ, the output is in $[0, \epsilon]$. Next, when $t^-_{in} \in [0.4, 0.8]$ and for $\epsilon < 1/40$, by Inequality 2 it holds that $x_{in} \in [0.3, 0.7]$ and the output of the gate in unconstrained. Finally, when $t^-_{in} > 0.8$, it holds that $t^-_{out} \in [1-\epsilon, 1+\epsilon']$ and by Definition 1, we have that $x_{out} \in [1-\epsilon, 1]$. Furthermore, by Inequality 2 and for $\epsilon' < 1/40$, it holds that $x_{in} > 0.9$ and the gate constraint is satisfied for a reason analogous to the one described above.

Given the discussion above, by setting $\epsilon' < \min\{\epsilon/11, 1/40\}$[8], all of the gate constraints are satisfied, and the vector (x_i) obtained from $\mathcal{C}$ is a solution to ϵ-GCIRCUIT. □

Now from Lemma 5, we obtain the following result.

Theorem 3. *There exists a constant $\epsilon' > 0$ such that (n, n, ϵ')-CHALVING is PPAD-hard.*

Proof. Recall that in the proof of Lemma 5, ϵ' was constrained to be at most $\min\{1/40, \epsilon/11\}$ and in particular by Lemma 3, there exists a constant ϵ' that would make the reduction work. Recall however that we "expanded" the instance of (n, ϵ')-CHALVING from the interval $[a, b]$ to $[0, 6N]$ for convenience, which implies that after rescaling the instance to a constant interval $[a, b]$, the allowed error ϵ' goes down to $O(1/n)$. To get a constant error ϵ', we simply multiply all valuations by N. □

Theorem 3 implies that although a solution with n cuts is generally desirable, it might be hard to compute, even for a relatively simple class of valuations like those used in the reduction. In fact, we can extend our results to the more general case of finding a partition with $n + k$ cuts where k is a constant.

[8] We can in fact assume some $\epsilon \leq 11/40$, as the smaller the ϵ, the harder the problem is, since we are interested in establishing hardness for some constant ϵ.

Theorem 4. *Let k be any constant. Then there exists a constant ϵ' such that $(n, n+k, \epsilon')$-*CHALVING *is PPAD-hard.*

Proof. Let $S = (V, \mathcal{T})$ be an instance of ϵ-GCIRCUIT with N nodes, consisting of smaller identical sub-circuits $S_i = (V_i, \mathcal{T_i})$, for $i = 1, 2, \ldots, k+1$, with with $N/(k+1)$ nodes each such that for all $i, j \in [k+1]$ such that $i \neq j$, it holds that $V_i \cap V_j = \emptyset$. and $\mathcal{T_i} \cap \mathcal{T_j} = \emptyset$. In other words, the circuit S consists of $k+1$ copies of a smaller circuit S_i that do not have any common nodes or gates. Furthermore, for convenience, assume without loss of generality that for two nodes l and m such that $u_l \in V_i$ and $u_m \in V_j$, with $i < j$, it holds that $l < m$. In other words, the labeling of the nodes is such that nodes in circuits with smaller indices have smaller indices.

Let H be the instance of (n, n, ϵ')-CHALVING corresponding to the circuit S following the reduction described in the beginning of the section and recall that $n = 2N$ in the construction. Note that according to the convention adopted above for the labeling of the nodes, for $i < j$, the agents corresponding to V_i lie in the interval $[\ell_i, r_i]$, whereas the agents corresponding to V_j lie in the interval $[\ell_j, r_j]$ and $r_i \leq \ell_j$. In other words, agents corresponding to sub-circuits with smaller indices are placed before agents with higher indices, and there is no overlap between agents corresponding to different sub-circuits.

Now suppose that we have a solution to $(n, n+k, \epsilon')$-CHALVING. Since there is no overlap between valuations corresponding to different sub-circuits, an approximate solution with $n+k$ cuts for the instance H implies that there exists some interval $[\ell_i, r_i]$ corresponding to the set of nodes V_i of sub-circuit S_i, such that at least $n/(k+1)$ cuts lie in $[\ell_i, r_i]$, otherwise the total number of cuts on H would be at least $n+k+1$. Since there are exactly $n/(k+1)$ agents with valuations on $[\ell_i, r_i]$, this would imply an approximate solution for n' agents with n' cuts and the problem reduces to (n, n, ϵ')-CHALVING. □

From the construction of the instance of Consensus-Halving used in the reduction, it is clear that the valuation functions are measures that satisfy non-negativity and additivity. The results still hold even if the valuations are probability measures, i.e. for each i, $u_i(O) = 1$.

Corollary 1. *Let k be any constant. Then there exists a constant ϵ' such that $(n, n+k, \epsilon')$-*CHALVING *is PPAD-hard, even when the valuations u_i are probability measures.*

Proof. By the construction of the instance of Consensus-Halving used in the proof of Theorems 3 and 4, it is clear that the valuation functions are measures satisfying countable additivity (i.e. $u_i(I_1 \cup I_2) = v_i(I_1) + v_i(I_2)$, for two intervals I_1 and I_2), non-negativity (i.e. $v_i(I) \geq 0$ for any interval $I \subseteq O$) and null empty set ($u_i(\emptyset) = 0$). To make them into measures such that $u_i(O) = u_j(O)$ for all agents $i, j \in N$, we can increase the valuations of agents for the intervals $[d_i, d_i+1]$, $[d_i+2, d_i+3]$ (for agents var_i) or $[d_i+3, d_i+4]$, $[d_i+5, d_i+6]$ (for agents $copy_i$) accordingly, without affecting the position of any cuts or introducing any new cuts. Informally, we can alter the valuation of agents for their "large valuation blocks" of value 4 to make sure all agents value the whole object equally. To make the measure into a probability measure, we can rescale the interval O accordingly, to make sure $u_i(O) = 1$. □

Finally, since Consensus-Halving is a special case of Continuous Necklace Splitting, our results imply the PPAD-hardness of that problem, when $t = 2$. Note that while the definition of the problem in [2] requires continuous probability measures, the original definition of [18] does not impose a continuity requirement (see also [16], Corollary 2).

Corollary 2. *Let k be any constant. Then there exists a constant ϵ' such that $(n, n+k, \epsilon')$-*CONNECKSPLIT *is PPAD-hard, when $t = 2$.*

5 $(n, n-1, \epsilon)$-CHALVING is NP-hard

In the previous section, we proved that the problem of finding an approximate solution with n players and n cuts is PPAD-complete. Recall that for that case, Theorem 2 guarantees that a solution exists. For n players and $n-1$ cuts however, we don't have the same guarantee. In this section, we prove that deciding whether an approximate solution with n players and $n-1$ cuts exists is NP-hard.

Theorem 5. *Ther exists a constant $\epsilon' > 0$ such that $(n, n-1, \epsilon')$-CHALVING is NP-hard.*

Proof. We will first describe the construction that we will use in the reduction. For consistency with the previous section, we will denote the error of the Consensus-Halving problem by ϵ' and the error of the (implict) generalized circuits by functions of ϵ. Let $R_\epsilon(S)$ be the construction for the reduction of Section 4 that encodes an ϵ-generalized circuit S into an $(n, n-1, \epsilon')$-CHALVING halving instance when $\epsilon' < \epsilon/11$. We will reduce from 3-SAT, which is known to be NP-complete.
Let ϕ be any 3-SAT formula with m clauses, $k \leq 3m$ variables $x_1, \ldots, x_k$, and let $\epsilon > 0$ be given. For convenience of notation, let $\delta = \epsilon/11$. We will (implicitly) create a generalized circuit S with the following building blocks:

- k input nodes $x_1, \ldots, x_k$ corresponding to the variables $x_1, \ldots, x_k$.

- k sub-circuits $\text{Bool}(x_i)$ for $i = 1, 2, \ldots, k$ that input the real value $x_i \in [0, 1]$ and output a boolean value $x_i^{bool} \in [0, 4\delta] \cup [1 - 4\delta, 1]$ (see the lower stage of Figure 5). The allowed error for these circuits will be δ. The implementation of the circuit in terms of the gates of the generalized circuit can be seen in Algorithm 1. Note that the sub-circuit containing all the $\text{Bool}(x_i)$ sub-circuits has at most $4k$ nodes as each $\text{Bool}(x_i)$ sub-circuit could be implemented with one constant gate, one subtraction gate, one addition gate and one equality gate; the latter is to maintain the convention that all inputs to each gate are distinct.

- A sub-circuit $\Phi(x_1^{bool}, \ldots, x_k^{bool})$ that implements the formula ϕ, inputing the boolean variables x_i^{bool} and outputting a value x_{out} corresponding to the value of the assignment plus the error introduced by the approximate gates. The allowed error for this circuit will be 4δ. A pictorial representation of such a circuit can be seen in Figure 5; note that the circuits $\text{Bool}(x_i)$ are also shown in the picture. This circuit has at most $k + 3m$ nodes. First, there might be k possible negation gates to negate the input variables. Secondly, for each clause, in order to implement an OR gate of fan-in 3, we need 2 OR gates of fan-in 2, for a total of $2m$ gates for all clauses. Finally, in order to simulate the AND gate with fan-in m, we need m AND gates of fan-in 2. Overall, since $k \leq 3m$, we need at most $6m$ nodes to implement this sub-circuit, using elements of the generalized circuit.

- A sub-circuit $\text{Rebool}(x_1, \ldots, x_k, x_{out})$ that inputs the variables x_i, for $i = 1, 2, \ldots, k$ and the variable x_{out} and computes the function

  $$\min(x_{out}, \max(x_1, 1 - x_1), \ldots, \max(x_k, 1 - x_k)).$$

  The function can be computed using the gates of the generalized circuit as shown in Algorithm 2. Let x_{out}^{bool} be the output of that sub-circuit with allowed error 4δ. Note that this circuit has at most $16k$ nodes. Each min and max operation requires 8 nodes and we need to do $2k$ such computations overall; k for the k max operations and k to implement the min operation of fan-in k with min operations of fan-in 2. Again, since $k \leq 3m$, this sub-circuit requires at most $48m$ nodes in total.

Following the notation introduced above, let $R_\delta(\text{Bool})$, $R_{4\delta}(\Phi)$ and $R_{4\delta}(\text{Rebool})$ denote the valuations of the agents in the instance of Consensus-Halving corresponding to those sub-circuits, according to the reduction described in Section 4. In other words, based on the circuit described above, we create an instance H of Consensus-Halving where we have:

- $2k$ agents (as each node corresponds to two agents, var_i and $copy_i$) that correspond to the input variables $x_1, \ldots, x_k$, who are not the output of any gate
- at most $2(4k + k + 3m + 16k)$ nodes corresponding to the internal nodes and the output node of the circuit.
- an additional agent with valuation

$$u_n = \begin{cases} 1, \text{if } t \in [b - 18m\epsilon' - 1, b - 18m\epsilon'] \\ 1, \text{if } t \in [b, b+1] \\ 0, \text{otherwise} \end{cases}$$

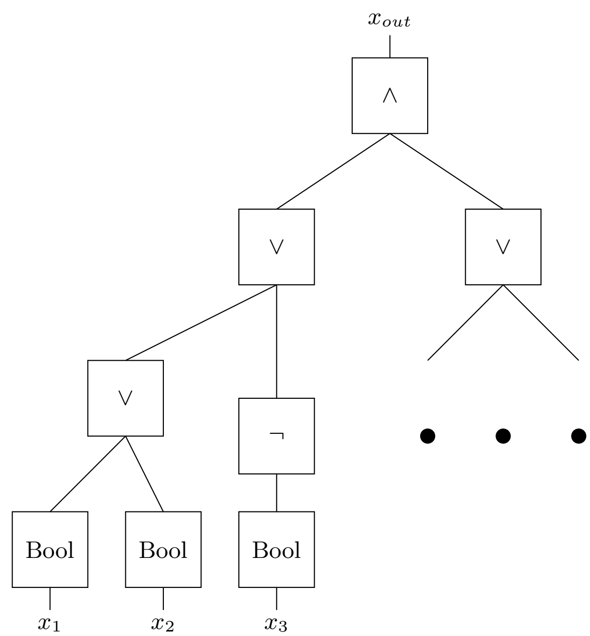


Fig. 2. A generalized circuit corresponding to a $3SAT$ formula ϕ, where the first clause is $(x_1 \vee x_2 \vee \overline{x_3})$. The nodes of the circuit between different layers are omitted.

where $[a, b]$ is the interval where the value of x_{out}^{bool} is "read" in the instance of Consensus-Halving, i.e. the interval where the cut $t_{out}^{bool}-$ will be placed in the Consensus-Halving solution.

Recall Definition 1 from Section 4 and note that as far as agent n is concerned, any cut $t_{out}^{bool}-$ such that $1 - 18m\epsilon \leq x_{out}^{bool} \leq 1$ is a Consensus-Halving solution.

We will now argue about the correctness of the reduction. Let n be the number of agents and notice that there are $n-1$ agents that correspond to the nodes of the circuit and a single agent constraining the value of x_{out}^{bool}. Notice that since the allowed error for the sub-circuit Rebool$(x_1, \ldots, x_k, x_{out})$ is 4δ, the total additive error of the agents of $R_{4\delta}$(Rebool) will be at most $4\delta \cdot 48m \leq 18m\epsilon'$.

First, assume that there exists a a solution to ϵ'-approximate Consensus-Halving with $n-1$ cuts. By the correctness of the construction of Section 4 and the fact that $\epsilon' < \epsilon/11 = \delta$, the solution encodes a valid assignment to the variables of the generalized circuit S. Due to the valuation of agent n, the output of C must satisfy

$$x_{out}^{bool} \geq 1 - 18m\epsilon' - \epsilon',$$

otherwise the corresponding cut $t_{out}^{bool}-$ could not be a part of a valid solution. Since the total additive error for the circuit Rebool$(x_1, \ldots, x_k, x_{out})$ is at most $18m\epsilon'$, if we choose $\epsilon' < 1/90m$, it holds that

$$x_{out}^{bool} \geq 4/5 - \epsilon' \text{ which implies that } x_{out} \geq 3/4,$$

by the function implemented by the circuit Rebool$(x_1, \ldots, x_k, x_{out})$. For the same reason, for each $i = 1, \ldots, k$ it holds that

$$x_i \in [0, 1/4] \cup [3/4, 1]$$

Algorithm 1 Computing $bool(x)$.

$a \leftarrow x - 1/4$
$bool \leftarrow a + a$

Algorithm 2 Computing $\min(x, y)$ and $\max(x, y)$.

$a \leftarrow x - y$; $b \leftarrow y - x$; $c \leftarrow a + b$
$d \leftarrow c/2$; $\ell \leftarrow (x/2) + (y/2)$
$min \leftarrow \ell - d$; $max \leftarrow \ell + d$

and hence the output of $\mathrm{Bool}(x_i)$ will lie in $[0, 4\delta] \cup [1 - 4\delta, 1]$, which means that the inputs $x_1^{bool}, \ldots, x_k^{bool}$ to the gates of the sub-circuit $\Phi(x_1^{bool}, \ldots, x_k^{bool})$ will be treated correctly as boolean values by the gates of the circuit (since the allowed error of the sub-circuit is 4δ). Since the circuit $\Phi(x_1^{bool}, \ldots, x_k^{bool})$ computes the boolean operations correctly and $x_{out} \geq 3/4$, the formula ϕ is satisfiable.
For the other direction, assume that ϕ is satisfiable and let $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_k)$ be a satisfying assignment. First we set the values of the variables $x_1, \ldots, x_k$ to 0 or 1 according to $\tilde{x}$ and then we propagate the values up the circuit S using the exact operation of the gates, which by our construction can be encoded to an instance of exact Consensus Halving for the $(n - 1)$ agents corresponding to the nodes of S, i.e. the first $n - 1$ will be exactly satisfied with the partition resulting from the encoded satisfying assignment. For the n-th agent, again, since the total additive error is bounded by $18m\epsilon'$, agent will be satisfied with the solution. □

Again, we remark here that by using similar arguments as in the proof of Corollary 1 we can prove that the result holds even when the valuations are probability measures.

6 Conclusion and Future Work

The PPAD-hardness of Consensus-Halving is is a strong indicator that the problem is computationally hard in the worst case. The main open problem raised here is to classify its complexity exactly: it is likely to be complete for either PPAD or PPA, but which one? PPA-completeness would be of particular interest, since to our knowledge, all problems that are known to be complete for PPA have a "PPA-like" circuit in their definitions. Our "in PPA" result follows the Simmons/Su approach of applying the Borsuk/Ulam principle to the space of possible solutions. An "in PPAD" result would need to identify structure that is present in instances derived from Consensus-Halving problems, which is not present in general Borsuk-Ulam problems. At present, Consensus-Halving joins a number of problems known to belong to PPA, whose precise complexity is not resolved (others include Smith's Theorem, see also Open Questions 3 and 4 of [1]).

Going beyond the Consensus-Halving problem in its original definition, it looks interesting to consider further the effects of allowing additional cuts, and the computation of exact or approximate solutions. Concretely, one might wonder how good an approximate Consensus-Halving solution can be computed in polynomial time, given, say, two cuts for each agent.

References

1. James Aisenberg, Maria Luisa Bonet, and Sam Buss. 2-D Tucker is PPA complete. *ECCC TR15*, 163, 2015.
2. Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987.
3. Noga Alon and Douglas B West. The Borsuk-Ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986.
4. Julius B Barbanel. Super envy-free cake division and independence of measures. *Journal of Mathematical Analysis and Applications*, 197(1):54–60, 1996.
5. Kim C Border. *Fixed point theorems with applications to economics and game theory.* Cambridge University Press, 1989.
6. Karol Borsuk. Drei sätze über die n-dimensionale euklidische sphäre. *Fundamenta Mathematicae*, 1(20):177–190, 1933.
7. Steven J Brams and D Marc Kilgour. Competitive fair division. *Journal of Political Economy*, 109(2):418–443, 2001.
8. Steven J Brams and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press, 1996.

9. Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *FOCS*, volume 6, page 47th, 2006.
10. Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 181–190. ACM, 2013.
11. Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.
12. Xiaotie Deng, Qi Qi, and Amin Saberi. Algorithmic solutions for envy-free cake cutting. *Operations Research*, 60(6):1461–1476, 2012.
13. Martin Gardner. *Aha! Aha! insight*, volume 1. Scientific American, 1978.
14. Charles H Goldberg and Douglas B West. Bisection of circle colorings. *SIAM Journal on Algebraic Discrete Methods*, 6(1):93–106, 1985.
15. Claus-Jochen Haake, Matthias G Raith, and Francis Edward Su. Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4):723–749, 2002.
16. Theodore P Hill. A proportionality principle for partitioning problems. *Proceedings of the American Mathematical Society*, 103(1):288–293, 1988.
17. Charles R Hobby and John R Rice. A moment problem in l 1 approximation. *Proceedings of the American Mathematical Society*, 16(4):665–670, 1965.
18. Jerzy Neyman. Un theoreme d'existence. *C. R. Acad. Sci. Paris Ser. A-B 222*, pages 843–845, 1946.
19. Abraham Othman, Christos Papadimitriou, and Aviad Rubinstein. The complexity of fairness through equilibrium. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 209–226. ACM, 2014.
20. Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.
21. Elisha Peterson and Francis Edward Su. Four-person envy-free chore division. *Mathematics Magazine*, 75(2):117–122, 2002.
22. Timothy Prescott and Francis Edward Su. A constructive proof of Ky Fan's generalization of Tucker's lemma. *Journal of Combinatorial Theory, Series A*, 111(2):257–265, 2005.
23. Jack Robertson and William Webb. Cake-cutting algorithms: Be fair if you can. 1998.
24. Aviad Rubinstein. Inapproximability of Nash equilibrium. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 409–418. ACM, 2015.
25. Forest W Simmons and Francis Edward Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Mathematical social sciences*, 45(1):15–25, 2003.
26. FW Simmons. Private communication to Michael Starbird. 1980.
27. Emanuel Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 6, pages 265–272. Springer, 1928.
28. Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1), 1948.
29. Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.
30. Francis Edward Su. Rental harmony: Sperner's lemma in fair division. *The American mathematical monthly*, 106(10):930–942, 1999.
31. Albert William Tucker. Some topological properties of disk and sphere... *Proc. First Canadian Math. Congress, Montreal*, pages 285–309, 1945.