Hardness Results for Consensus-Halving

Aris Filos-Ratsikas[1]
École Polytechnique Fédérale de Lausanne, Switzerland
aris.filosratsikas@epfl.ch

Søren Kristoffer Stiil Frederiksen
Aarhus University, Denmark
sorensf@gmail.com

Paul W. Goldberg
University of Oxford, United Kingdom
paul.goldberg@cs.ox.ac.uk

Jie Zhang[2]
University of Southampton, United Kingdom
jie.zhang@soton.ac.uk

Abstract

The *Consensus-halving* problem is the problem of dividing an object into two portions, such that each of n agents has equal valuation for the two portions. We study the ϵ-approximate version, which allows each agent to have an ϵ discrepancy on the values of the portions. It was recently proven in [13] that the problem of computing an ϵ-approximate Consensus-halving solution (for n agents and n cuts) is PPA-complete when ϵ is *inverse-exponential*. In this paper, we prove that when ϵ is *constant*, the problem is PPAD-hard and the problem remains PPAD-hard when we allow a constant number of additional cuts. Additionally, we prove that deciding whether a solution with $n-1$ cuts exists for the problem is NP-hard.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases PPAD, PPA, consensus halving, generalized-circuit, reduction

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.24

Related Version A full version of the paper is available at `https://arxiv.org/abs/1609.05136`.

1 Introduction

Suppose that two families wish to split a piece of land into two regions such that every member of each family believes that 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* [27]. 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

[1] The author was supported by the ERC Advanced Grant 321171 (ALGAME).
[2] The author was supported by the ERC Advanced Grant 321171 (ALGAME).

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 24; pp. 24:1–24:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

into two portions such that each of the n agents believes that the two portions have equal value, according to her personal opinion. The division may need to cut the object into pieces and then label each piece appropriately to include it in one of the two portions.

The importance of the Consensus-halving problem - or to be precise, of its approximate version, where there is an associated precision parameter ϵ - other than the fact that it models real-life problems like the ones described above, lies in in the following fact: It is the first "natural" problem that is *complete* for the complexity class PPA, where "natural" here means that its does not contain a circuit explicitly in its definition; this was proven quite recently by Filos-Ratsikas and Goldberg [13]. PPA is a class of *total search problems* [19] defined in [22], and is a superclass of the class PPAD, which precisely captures the complexity of computing a Nash equilibrium [11, 9]. Therefore, generally speaking, a PPA-hardness result is stronger than a PPAD-hardness result.

Crucially however, the hardness result in [13] requires the precision parameter to be inverse-exponential in the number of agents and does not even provably preclude the possibility of efficient algorithms, if we allow larger discrepancies in the values for the two portions. In this paper, we prove that this is actually not possible[3], by showing that even when the allowed discrepancy is independent of the number of agents, the problem is PPAD-hard. Understanding the problem for increasing values of the discrepancy parameter is quite important in terms of capturing precisely its complexity and resembles closely the series of results establishing hardness of computing a mixed ϵ-Nash equilibrium, from ϵ being inverse-polynomial in [11, 9] to being constant in [26], as well as several other problems (see [26]). Additionally, one could imagine that solutions where constant discrepancies are acceptable are the ones arising in several real-life scenarios, such as splitting land.

1.1 Our results

We are interested in the computational complexity of computing an ϵ-approximate solution to the Consensus-halving problem where ϵ is a constant function of the number of agents, as well as the complexity of deciding whether given an input instance, $n-1$ cuts are sufficient to achieve an ϵ-approximate solution. We discuss our main results below.

- 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, 26].
- 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. Using the gadgetry already developed for the PPAD-hardness proof, we establish the result via a reduction from 3-SAT.
- 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 [22, 1].

We remark here that an earlier version of this paper actually predated [13], and some of the results in [13] are established by referencing the results in the present paper. Specifically:

[3] Under usual computational complexity assumptions, here that PPAD-hard problems do not admit polynomial-time algorithms.

- While the authors of [13] provide a rather elaborate reduction to establish PPA-hardness of the problem, the inclusion in the class PPA is established with reference to the present paper. In turn, the inclusion result follows from a formalization of the ideas of the algorithms by [27] and [24] and Fan's version of Tucker's Lemma [12, 31].
- In [13], the authors obtain a *computational equivalence* between the *Necklace Splitting* problem [2] and ϵ-approximate Consensus-halving, for ϵ being at least inverse-polynomial. The inverse-polynomial dependence on ϵ implies that PPA-hardness of the former problem does not follow from their hardness result, but PPAD-hardness does follow from their reduction and our main result here.

Due to lack of space, some of the proofs and details are left for the the full version of the paper. Most emphasis is put on the main PPAD-hardness proof of Section 3, which is presented in sufficient detail, and the NP-hardness proof of Section 4. The exposition of the results in Section 5 is limited to higher-level intuition, with full proofs in the full version.

1.2 Related work

The Consensus-halving problem was explicitly formalized and studied firstly by Simmons and Su [27], 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]. The existence of solutions to the problem was already known since [16, 3, 4] but the algorithm in [27] 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 [20] and can also be obtained as an application of the Hobby-Rice Theorem [18] (also see [2]). Despite proposing an explicit protocol however, the authors in [27] do not answer the question of "efficiency", i.e. how fast can a protocol find an (approximate) solution and the running time of their protocol is worst-case exponential-time.[4]

To this end, Filos-Ratsikas and Goldberg [13] recently proved that the problem is PPA-complete, but as we explained in the introduction, the hardness holds only when the precision parameter is inversely exponential. Even more recently, the authors strengthened their result to PPA-completeness of the problem for inversely polynomial precision [14]. However, since our hardness result holds for constant precision, it is not subsumed by neither [13] or [14].

The computational classes PPA (Polynomial Parity Arguments) and PPAD (Polynomial Parity Arguments on Directed graphs) were introduced by Papadimitriou [22] in an attempt to capture the precise complexity of several interesting problems of a topological nature such as computational analogues of Sperner's Lemma [28] and Brouwer's and Kakutani's fixed point theorems [5], which are all known to be in PPAD [22]. Interestingly, Aisenberg et al. [1] recently proved that the search problems associated with the Borsuk-Ulam Theorem and Tucker's Lemma are PPA-complete; this is the starting point for the reduction in [13], but it will also be used for our "in-PPA" result, which complements the hardness result of [13].

Our PPAD-hardness reduction goes via the *Generalized Circuit* problem. Generalized circuits differ from usual circuits in the sense that they can contain cycles, a fact which basically induces a continuous function on the gates, and the solution is guaranteed to exist

[4] The protocol exhaustively iterates through all the vertices of triangulated geometric object, which, to achieve a small discrepancy, has to be exponentially large.

by Brouwer's fixed point theorem. The ϵ-approximate Generalized Circuit problem was implicitly proven to be PPAD-complete for exponentially small ϵ in [11] and explicitly for polynomial small ϵ [9], en route to proving that perhaps the most interesting problem in PPAD, that of computing a mixed-Nash equilibrium, is also complete for the class. The same problem was also used in [10] to prove that finding an approximate competitive equilibrium for the Arrow-Debreu market with linear and non-monotone utilities is PPAD-complete and in [21] to prove that finding an approximate solution of the Competitive Equilibrium with Equal Incomes (CEEI) for indivisible items is PPAD-complete. More recently, Rubinstein [26] 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; we reduce from that version of the problem. This improvement should also lead to stronger hardness results in [10] and [21], as well as other problems that rely on the Generalized Circuit problem.

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 fair division literature, which dates back to the late 1940s [29], has studied a plethora of such problems, with *chore-division* [23, 15], *rent-partitioning* [17, 7, 30] and the perhaps the most well-known one, *cake-cutting* [8, 25] being notable examples. Note that Consensus-halving is inherently different from cake-cutting, since the objective is that all participants are (approximately) equally satisfied with the solution, and they do not have "ownership" over the resulting pieces.

2 Preliminaries

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

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

For simplicity, the reader may assume that the valuations are represented as step functions (where agents have constant values over distinct intervals), although this is not necessary for the results to hold.[5]

A set of k cuts $\{t_1,\dots,t_k\}$, where $0 \le t_1 \le \ldots \le t_k \le 1$, means that we cut along the points $t_1,\dots,t_k$, such that the object is cut into $k+1$ pieces $X_i = [t_{i-1}, t_i]$ for $i = 1,\dots,k+1$, where $t_0 = 0$ and $t_{k+1} = 1$. A labelling of an interval X_i means that we assign a label $\ell \in \{+,-\}$ to X_i, which corresponds to including X_i in a set of intervals, either O_+ or O_-. 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., $u_i(O_+) =$

[5] The inclusion result actually holds for more general functions, while our hardness results (PPAD-hardness and NP-hardness) hold even for well-behaved functions, such as step functions. We note here that while an exact solution to Consensus-halving generally requires the valuations to be continuous, this is not necessary for the existence of an approximate solution; the algorithm of [27] can find such a solution assuming that valuations are bounded and non-atomic.

$u_i(O_-), \forall i \in N$. The ϵ-approximate Consensus-halving problem allows that each agent has a small discrepancy on the values of the two partitions, i.e., $|u_i(O_+) - u_i(O_-)| < \epsilon$. As discussed in the Introduction, such a solution always exists [27].

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

▶ **Problem 1.** (n, k, ϵ)-ConHalving.
Input: *The value density functions (valuation functions)* $v_i : O \to R_+, i = 1, \cdots, n$.
Output: *A partition* (O_+, O_-) *with* k *cuts such that* $|u_i(O_+) - u_i(O_-)| \leq \epsilon$.

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

▶ **Problem 2.** $(n, n-1, \epsilon)$-ConHalving.
Input: *The valensity functions* $v_i : O \to R_+, i = 1, \cdots, n$.
Output: Yes, *if a partition* (O_+, O_-) *with* $n-1$ *cuts such that* $|u_i(O_+) - u_i(O_-)| \leq \epsilon$ *for all agents* $i \in N$ *exists, and* No *otherwise.*

TFNP, PPA and PPAD: 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 [19].

An important subclass of TFNP is the class PPAD, defined by Papadimitriou in [22]. PPAD stands for "Polynomial Parity Argument on a Directed graph" and is defined formally in terms of the problem End-Of-Line [22]. The class 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 the successor $S(v_1)$ of node v_1 is v_2 and the predecessor $P(v_2)$ of node v_2 is 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 [22] and a problem is the class PPA if it is polynomial-time reducible to Leaf.

The formal definitions of End-Of-Line and Leaf are not required for the results presented in this version and therefore are left for the full version.

2.1 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 $v_{out} \neq v'_{out}$.

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

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

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, 26].

▶ **Problem 3.** 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.*

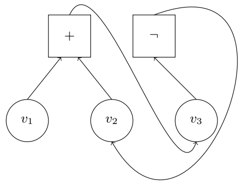
Note that a solution to ϵ-GCircuit always exists [9] and hence the problem is well-defined. Also, notice that for the logic gates $G_\vee, G_\wedge$ and $G_\neg$, if the input conditions are not fulfilled, the output is unconstrained, and for the multiplication gate, it holds that $\alpha \in (0, 1]$. ϵ-GCircuit was proven to be PPAD-complete implicitly or explicitly in [11, 9] for inversely polynomial error ϵ and in [26] for constant ϵ. We state the latter theorem here as a lemma:

▶ **Lemma 1** ([26]). *There exists a constant $\epsilon > 0$ such that ϵ-GCircuit is PPAD-complete.*

3 Consensus-Halving with $n + k$ cuts is is PPAD-hard

In this section, we will first 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. 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')-ConHalving 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 Consensus-halving instance on the interval $[0, 6N]$. Let $d_i := 6(i - 1)$.



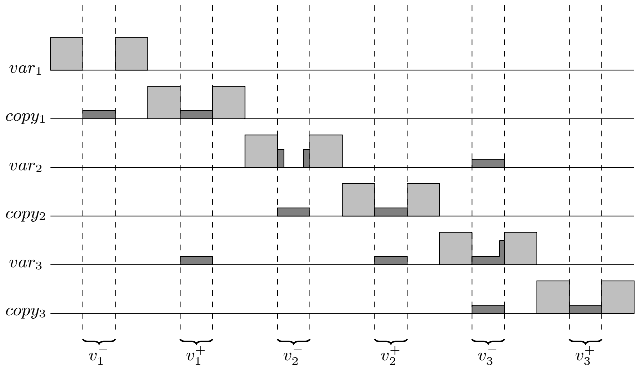


Figure 1 An instance of ϵ-GCIRCUITwith the corresponding construction for (n, ϵ')-CONHALVING.

The two agents var_i and $copy_i$ that we construct for every node v_i have valuations

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

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

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

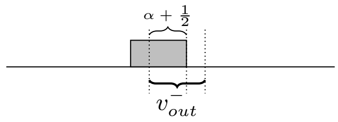
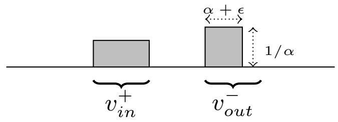
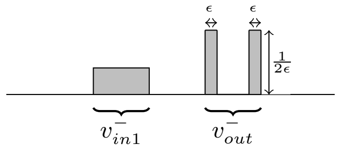
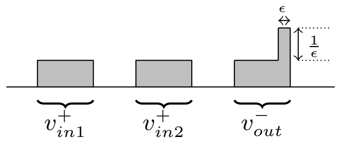
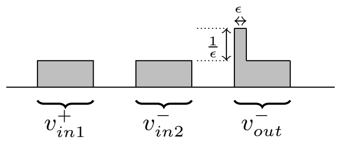
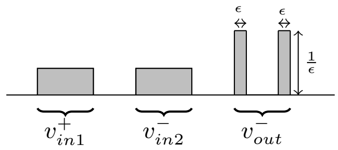
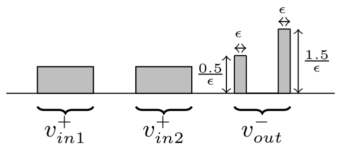
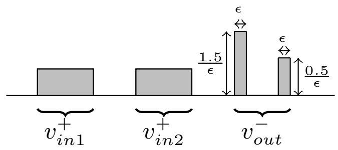
Since each node is the output of at most one gate, var_i is well-defined. Note that apart from the valuation defined by the function G^τ, agents var_i and $copy_i$ only have valuations on the sub-interval $[d_i, d_{i+1}]$, i.e., the agents associated with node v_1 only have valuations on $[0, 6]$, the 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 1 demonstrates an example of a Consensus-halving instance corresponding to a small circuit.

▶ **Lemma 2.** *Given the construction of a (n, n, ϵ')-CONHALVING instance above, for $\epsilon' < \min\{\epsilon/11, 1/40\}$, 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 [27], such a partition with $2N$ cuts is guaranteed to exist.

Now consider such a solution $\mathcal{C}$ to (n, n, ϵ')-CONHALVING 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')-CONHALVING. We will assume without loss of generality that the leftmost piece of the partition $\mathcal{C}$ is in O_-. Notice then

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_-	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^-_{in_2} \\ 1 & \text{if } t \in [v^-_{out,\ell} + \epsilon, v^-_{out,r}] \\ 1/\epsilon + 1 & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 0 & \text{otherwise} \end{cases}$	
$G_<$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^-_{in_2} \\ 1/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 1/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
$G_\vee$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^+_{in_2} \\ 0.5/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 1.5/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	
$G_\wedge$	$\begin{cases} 1 & \text{if } t \in v^+_{in_1} \cup v^+_{in_2} \\ 1.5/\epsilon & \text{if } t \in [v^-_{out,\ell}, v^-_{out,\ell} + \epsilon] \\ 0.5/\epsilon & \text{if } t \in [v^-_{out,r} - \epsilon, v^-_{out,r}] \\ 0 & \text{otherwise} \end{cases}$	

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 $t_i^- \in [-\epsilon', 1 + \epsilon']$. 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')-ConHalving, 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

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

To complete the proof, we just need to argue that these variables satisfy the constraints of the gates of the circuit. Due to lack of space, we will only argue the correctness of some of the gates here; the arguments for the remaining gates follow a similar spirit and are presented in detail in the full version.

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 $[d_i, d_i+3]$,[6] 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.

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 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

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

$t^-_{out} \in [-\epsilon', \epsilon]$ 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.

Given the discussion above, by setting $\epsilon' < \min\{\epsilon/11, 1/40\}$[7], the gate constraints are satisfied, and the vector (x_i) obtained from $\mathcal{C}$ is a solution to ϵ-GCIRCUIT. ◀

Now from Lemma 2, we obtain the following result.

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

Proof. Recall that in the proof of Lemma 2, ϵ' was constrained to be at most $\min\{1/40, \epsilon/11\}$ and in particular by Lemma 1, there exists a constant ϵ' that would make the reduction work. Recall however that we "expanded" the instance of (n, ϵ')-CONHALVING 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.

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

[7] 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.

Now suppose that we have a solution to $(n, n+k, \epsilon')$-ConHalving. 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')-ConHalving. ◀

4 Consensus-Halving with $n - 1$ cuts is NP-hard

We have proved that the problem of finding an approximate solution with n players and n cuts is PPAD-complete. For n players and $n-1$ cuts however, we no longer have a guarantee that a solution exists. We prove that deciding whether this is the case or not is NP-hard.

▶ **Theorem 5.** *There exists a constant $\epsilon' > 0$ such that $(n, n-1, \epsilon')$-ConHalving 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 3, that encodes an ϵ-generalized circuit S into an $(n, n-1, \epsilon')$-ConHalving 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 $\mathrm{Bool}(x_i)$ 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 2). 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 $\mathrm{Bool}(x_i)$ sub-circuits has at most $4k$ nodes as each $\mathrm{Bool}(x_i)$ 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 2; note that the circuits $\mathrm{Bool}(x_i)$ 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 $\mathrm{Rebool}(x_1, \ldots, x_k, x_{out})$ that inputs the variables x_i, for $i = 1, 2, \ldots, k$ and the variable x_{out} and computes the function

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

Algorithm 1 Computing $bool(x)$.

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

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

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

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 3. 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

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

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 3 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 $\text{Rebool}(x_1, \ldots, x_k, x_{out})$ is 4δ, the total additive error of the agents of $R_{4\delta}(\text{Rebool})$ 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 3 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 $\text{Rebool}(x_1, \ldots, x_k, x_{out})$ is at most $18m\epsilon'$, if we choose $\epsilon' < 1/90m$, it

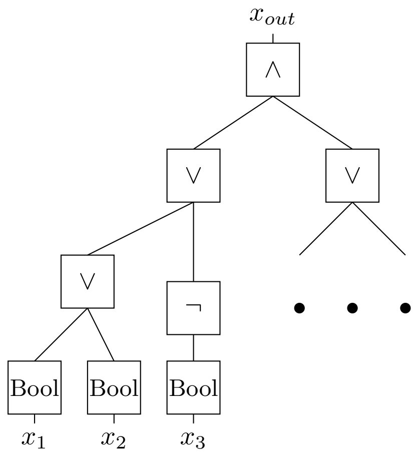


Figure 2 A generalized circuit corresponding to a 3-SAT formula ϕ, where the first clause is $(x_1 \vee x_2 \vee \overline{x_3})$. The nodes of the circuit between different layers are omitted. The layer at the output layer that "restores" the boolean values is also not shown, therefore x_{out} is the outcome of the emulated formula ϕ.

holds that $x_{out}^{bool} \geq 4/5 - \epsilon'$, which implies that $x_{out} \geq 3/4$, by the function implemented by the circuit $\mathrm{Rebool}(x_1, \ldots, x_k, x_{out})$. For the same reason, for each $i = 1, \ldots, k$ it holds that $x_i \in [0, 1/4] \cup [3/4, 1]$, 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 $18m\epsilon'$, the agent will be satisfied with the solution. ◀

5 Consensus-Halving with n cuts is in PPA

In this section, we prove that (n, n, ϵ)-ConHalving is in PPA. As we discussed in the introduction, this result of ours was referenced in [13] to complement the PPA-hardness reduction of the inverse-exponential precision version and obtain PPA-completeness.

▶ **Theorem 6.** *(n, n, ϵ)-ConHalving is in PPA.*

For establishing this result, we construct a reduction from (n, n, ϵ)-ConHalving to the PPA-complete problem Leaf which goes via (n, T)-Tucker the computational version of Tucker's Lemma.

More precisely, to prove that (n, n, ϵ)-ConHalving is in PPA, we follow the main idea of the algorithm provided in [27] for obtaining a Consensus-halving solution: the coordinates of any vertex $\mathbf{x}$ in the unit cross polytope C^n naturally correspond to a partition that uses n cuts on the $[0, 1]$ interval. 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 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 note that 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 a sphere of a different radius.

Given a sub-division of this sphere into small simplices (i.e. a *triangulation*) T of mesh size τ, we label each point of the triangulation by the label of the agent that is most dissatisfied by the corresponding set of cuts (and the sign indicates the direction of the discrepancy). This labelling satisfies the boundary conditions of Tucker's lemma and solutions to (n,T)-Tucker correspond to solutions of (n,n,ϵ)-ConHalving. In simple words, we show that the algorithm of [27] solves the computational version of Consensus-halving, using an algorithm for the computational version of Tucker as a subroutine.

The "in PPA" result is then established by the fact that Tucker is in PPA, i.e. it reduces to Leaf in polynomial time; this was already known from [22], where the problem is defined with respect to a subdivision of a hypercube. Technically, the algorithm of [27] that we use in our reduction requires the problem to be defined on the triangulation of a cross polytope, so one would have to prove that this version of the problem is in PPA as well. While this was already sketched in [22], we also prove it here via explaining how a constructive proof of Fan's combinatorial lemma [12] proposed by Prescott and Su [24] can be converted into a reduction to Leaf. The details along with all the necessary definitions are included in the full version.

6 Conclusion and Future Work

Our work takes an extra step in the direction of capturing the exact complexity of the Consensus-halving problem for all precision parameters. While, as we mentioned in the introduction, the techniques developed in [13] were successfully extended to obtain PPA-hardness of the problem for an inverse-polynomial precision parameter [14], it seems unlikely that they could be applicable when the precision is constant. In that sense, our main result is not implied by [13, 14], neither can it be subsumed by modifications to those reductions, even those involving highly non-trivial alterations. In other words, it seems that a PPA-completeness result for constant precision would require techniques fundamentally different from those used in [13, 14], and one can not even exclude the possibility of the problem being complete for PPAD instead.

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, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3):14, 2009.

10 Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. The complexity of non-monotone markets. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 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 Ky Fan. Simplicial maps from an orientable n-pseudomanifold into Sm with the octahedral triangulation. *Journal of Combinatorial Theory*, 2(4):588–602, 1967.

13 Aris Filos-Ratsikas and Paul W. Goldberg. Consensus Halving is PPA-Complete. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC)*, pages 51–64. ACM, 2018.

14 Aris Filos-Ratsikas and Paul W. Goldberg. The Complexity of Splitting Necklaces and Bisecting Ham Sandwiches. *arXiv preprint arXiv:1805.12559*, 2018.

15 Martin Gardner. *Aha! Aha! insight*, volume 1. Scientific American, 1978.

16 Charles H. Goldberg and Douglas B. West. Bisection of circle colorings. *SIAM Journal on Algebraic Discrete Methods*, 6(1):93–106, 1985.

17 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.

18 Charles R. Hobby and John R. Rice. A moment problem in L1 approximation. *Proceedings of the American Mathematical Society*, 16(4):665–670, 1965.

19 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

20 Jerzy Neyman. Un theoreme d'existence. *C. R. Acad. Sci. Paris Ser. A-B 222*, pages 843–845, 1946.

21 Abraham Othman, Christos H. Papadimitriou, and Aviad Rubinstein. The complexity of fairness through equilibrium. In *Proceedings of the 15th ACM conference on Economics and Computation (EC)*, pages 209–226. ACM, 2014.

22 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.

23 Elisha Peterson and Francis Edward Su. Four-person envy-free chore division. *Mathematics Magazine*, 75(2):117–122, 2002.

24 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.

25 Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can*. Natick: AK Peters, 1998.

26 Aviad Rubinstein. Inapproximability of Nash equilibrium. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing (STOC)*, pages 409–418. ACM, 2015.

27 Forest W. Simmons and Francis Edward Su. Consensus-halving via theorems of Borsuk-Ulam and Tucker. *Mathematical Social Sciences*, 45(1):15–25, 2003.

28 Emanuel Sperner. Neuer Beweis für die Invarianz der Dimensionszahl und des Gebietes. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 6 (1), pages 265–272. Springer, 1928.

29 Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1), 1948.

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.