Consensus Halving for Sets of Items

Paul W. Goldberg[1], Alexandros Hollender[1], Ayumi Igarashi[2], Pasin Manurangsi[3], and Warut Suksompong[4](✉)

[1] University of Oxford, Oxford, UK
{paul.goldberg,alexandros.hollender}@cs.ox.ac.uk
[2] National Institute of Informatics, Tokyo, Japan
ayumi_igarashi@nii.ac.jp
[3] Google Research, Mountain View, USA
pasin@google.com
[4] National University of Singapore, Singapore, Singapore
warut@comp.nus.edu.sg

Abstract. Consensus halving refers to the problem of dividing a resource into two parts so that every agent values both parts equally. Prior work has shown that when the resource is represented by an interval, a consensus halving with at most n cuts always exists, but is hard to compute even for agents with simple valuation functions. In this paper, we study consensus halving in a natural setting where the resource consists of a set of items without a linear ordering. When agents have additive utilities, we present a polynomial-time algorithm that computes a consensus halving with at most n cuts, and show that n cuts are almost surely necessary when the agents' utilities are drawn from probabilistic distributions. On the other hand, we show that for a simple class of monotonic utilities, the problem already becomes PPAD-hard. Furthermore, we compare and contrast consensus halving with the more general problem of consensus k-splitting, where we wish to divide the resource into k parts in possibly unequal ratios, and provide some consequences of our results on the problem of computing small agreeable sets.

Keywords: Consensus halving · PPAD-hardness · Resource allocation

1 Introduction

Given a set of resources, how can we divide it between two families in such a way that every member of both families believes that the two resulting parts have the same value? This is an important problem in resource allocation and has been addressed several times under different names [1,15,20], with *consensus halving* being the name by which it is best known today [26].

In prior studies of consensus halving, the resource is represented by an interval, and the goal is to find an equal division into two parts that makes a small number of cuts in the interval.[1] Using the Borsuk-Ulam theorem from topology,

[1] Simmons and Su [26] assume that the resource is a two- or three-dimensional object but only consider cuts by parallel planes; their model is therefore equivalent to that of a one-dimensional object.

X. Chen et al. (Eds.): WINE 2020, LNCS 12495, pp. 384–397, 2020.
https://doi.org/10.1007/978-3-030-64946-3_27

Simmons and Su [26] established that for any continuous preferences of the n agents involved, there is always a consensus halving that uses no more than n cuts—this also matches the smallest number of cuts in the worst case. In addition, the same authors developed an algorithm that computes an ε-approximate solution for any given $\varepsilon > 0$, meaning that the values of the two parts differ by at most ε for every agent. Although the algorithm is more efficient than a brute-force approach, its running time is exponential in the parameters of the problem. This is in fact not a coincidence: Filos-Ratsikas and Goldberg [9] recently showed that ε-approximate consensus halving is PPA-complete, implying that the problem is unlikely to admit a polynomial-time algorithm. Filos-Ratsikas et al. [11] strengthened this result by proving that the problem remains hard even when the agents have simple valuations over the interval. In particular, the PPA-completeness result holds for agents with "two-block uniform" valuations, i.e., valuation functions that are piecewise uniform over the interval and assign non-zero value to at most two separate pieces.

While these hardness results stand in contrast to the positive existence result, they rely crucially on the resource being in the form of an interval. Most practical division problems do not fall under this assumption, including when we divide assets such as houses, cars, stocks, business ownership, or facility usage. When each item is homogeneous, a consensus halving can be easily obtained by splitting every item in half. However, since splitting individual assets typically involves an overhead, for example in managing a joint business or sharing the use of a house, we want to achieve a consensus halving while splitting only a small number of assets. Fortunately, a consensus halving that splits at most n items is guaranteed to exist regardless of the number of items—this can be seen by arranging the items on a line in arbitrary order and applying the aforementioned existence theorem of Simmons and Su [26]. The bound n is also tight: if each agent only values a single item and the n valued items are distinct, all of them clearly need to be split. Nevertheless, given that the items do not inherently lie on a line, the hardness results from previous work do not carry over. Could it be that computing a consensus halving efficiently is possible when the resource consists of a set of items?

1.1 Overview of Results

We assume throughout the paper that the resource is composed of m items. Each item is homogeneous, so the utility of an agent for a (possibly fractional) set of items depends only on the fractions of the m items in that set. For this overview we focus on the more interesting case where $n \le m$, but all of our results can be extended to arbitrary n and m.

We begin in Sect. 2 by considering agents with *additive* utilities, i.e., the utility of each agent is additive across items and linear in the fraction of each item. Under this assumption, we present a polynomial-time algorithm that computes a consensus halving with at most n cuts by finding a vertex of the polytope defined by the relevant constraints. This positive result stands in stark contrast with the PPA-hardness when the items lie on a line, which we obtain by discretizing an analogous hardness result of Filos-Ratsikas et al. [11]. We then show that improving the number of cuts beyond n is difficult: even computing a consensus halving that uses at most $n-1$ cuts more than the minimum possible for a given instance is NP-hard. Nevertheless, we establish that instances admitting a solution with fewer than n cuts are rare. In particular, if the agents' utilities for items are drawn independently from non-atomic distributions, it is almost surely the case that every consensus halving requires no fewer than n cuts.

Next, in Sect. 3, we address the broader class of *monotonic* utilities, wherein an agent's utility for a set does not decrease when any fraction of an item is added to the set. For such utilities, we show that the problem of computing a consensus halving with at most n cuts becomes PPAD-hard, thereby providing strong evidence of its computational hardness.[2] Perhaps surprisingly, this hardness result holds even for the class of utility functions that we call "symmetric-threshold utilities", which are very close to being additive. Indeed, such utility functions are additive across items; for each item, having a sufficiently small fraction of the item is the same as not having the item at all, having a sufficiently large fraction of it is the same as having the whole item, and the utility increases linearly in between. On the other hand, we present a number of positive results for monotonic utilities when the number of agents is constant in the full version of our paper [13].

In Sect. 4, we provide some implications of our results on the "agreeable sets" problem studied by Manurangsi and Suksompong [18]. A set is said to be *agreeable* to an agent if the agent likes it at least as much as the complement set. Manurangsi and Suksompong proved that a set of size at most $\left\lfloor \frac{m+n}{2} \right\rfloor$ that is agreeable to all agents always exists, and this bound is tight. They then gave polynomial-time algorithms that compute an agreeable set matching the tight bound for two and three agents. We significantly generalize this result by exhibiting efficient algorithms for any number of agents with additive utilities, as well as any *constant* number of agents with monotonic utilities. In addition, we present a short alternative proof for the bound $\left\lfloor \frac{m+n}{2} \right\rfloor$ via consensus halving.

Finally, in Sect. 5, we study the more general problem of *consensus k-splitting* for agents with additive utilities. Our aim in this problem is to split the items into k parts so that all agents agree that the parts are split according to some given ratios $\alpha_1, \ldots, \alpha_k$; consensus halving corresponds to the special case where $k = 2$ and $\alpha_1 = \alpha_2 = 1/2$. Unlike for consensus halving, however, in consensus k-splitting we may want to cut the same item more than once when $k > 2$, so we cannot assume without loss of generality that the number of cuts is equal to the number of items cut. For any k and any ratios $\alpha_1, \ldots, \alpha_k$, we show that there exists an instance in which cutting $(k-1)n$ items is necessary. On the other hand, a generalization of our consensus halving algorithm from Sect. 2 computes a consensus k-splitting with at most $(k-1)n$ cuts in polynomial time, thereby implying that the bound $(k-1)n$ is tight for both benchmarks. We also illustrate

[2] We refer to [22, Chapter 20] for a discussion of the complexity class PPAD.

further differences between consensus k-splitting and consensus halving, both with respect to item ordering and from the probabilistic perspective.

1.2 Related Work

Consensus halving falls under the broad area of *fair division*, which studies how to allocate resources among interested agents in a fair manner [4,5,19]. Common fairness notions include *envy-freeness*—no agent envies another agent in view of the bundles they receive—and *equitability*—all agents have the same utility for their own bundle. The fair division literature typically assumes that each recipient of a bundle is either a single agent or a group of agents represented by a single preference. However, a number of recent papers have considered an extension of the traditional setting to groups, thereby allowing us to capture the differing preferences within the same group as in our introductory example with families [16,17,25]. Note that a consensus halving is envy-free for all members of the two groups; moreover, it is equitable provided that the utilities of the agents are additive and normalized so that every agent has the same value for the entire set of items.

A classical fair division algorithm that dates back over two decades is the *adjusted winner procedure*, which computes an envy-free and equitable division between two agents [4].[3] The procedure has been suggested for resolving divorce settlements and international border disputes, with one of its advantages being the fact that it always splits at most one item. Sandomirskiy and Segal-Halevi [24] investigated the problem of attaining fairness while minimizing the number of shared items, and gave algorithms and hardness results for several variants of the problem. Like in our work, both the adjusted winner procedure and the work of Sandomirskiy and Segal-Halevi [24] assume that items are homogeneous and, as in Sect. 2, that the agents' utilities are linear in the fraction of each item and additive across items. Moreover, both of them require the assumption that all items can be shared—if some items are indivisible, then an envy-free or equitable allocation cannot necessarily be obtained.[4]

Besides consensus halving, another problem that also involves dividing items into equal parts is *necklace splitting*, which can be seen as a discrete analog of consensus halving [1,12]. In a basic version of necklace splitting, there is a necklace with beads of n colors, with each color having an even number of beads. Our task is to split the necklace using at most n cuts and arrange the resulting pieces into two parts so that the beads of each color are evenly distributed between both parts. Observe that the difficulty of this problem lies in the spatial

[3] See http://www.nyu.edu/projects/adjustedwinner for a demonstration and implementation of the procedure.

[4] This motivates relaxations such as *envy-freeness up to one item (EF1)* and *envy-freeness up to any item (EFX)*, which have been extensively studied in the last few years (e.g., [6,21]). However, as Sandomirskiy and Segal-Halevi [24] noted, when a divorcing couple decides how to split their children or two siblings try to divide three houses between them, it is unlikely that anyone will agree to a bundle that is envy-free up to one child or house.

ordering of the beads—the problem would be trivial if the beads were unordered items as in our setting. While consensus halving and necklace splitting have long been studied by mathematicians, they recently gained significant interest among computer scientists thanks in large part to new computational complexity results [9–11]. In particular, the PPA-completeness result of Filos-Ratsikas and Goldberg [9] for approximate consensus halving was the first such result for a problem that is "natural" in the sense that its description does not involve a polynomial-sized circuit.

2 Additive Utilities

We first formally define the problem of consensus halving for a set of items. There is a set $N = [n]$ of n agents and a set $M = [m]$ of m items, where $[r] := \{1, 2, \dots, r\}$ for any positive integer r. A *fractional set of items* contains a fraction $x_j \in [0, 1]$ of each item j. We will mostly be interested in fractional sets of items in which only a small number of items are fractional—that is, most items have $x_j = 0$ or 1. Agent i has a utility function u_i that describes her nonnegative utility for any fractional set of items; for an item $j \in M$, we sometimes write $u_i(j)$ to denote $u_i(\{j\})$. A *partition of M into fractional sets of items $M_1, \dots, M_k$* has the property that for every item $j \in M$, the fractions of item j in the k fractional sets sum up to exactly 1.

Definition 1. *A* consensus halving *is a partition of M into two fractional sets of items M_1 and M_2 such that $u_i(M_1) = u_i(M_2)$ for all $i \in N$. An item is said to be* cut *if there is a positive fraction of it in both parts of the partition.*

In this section, we assume that the agents' utility functions are *additive*. This means that for a set M' containing a fraction x_j of item j, the utility of agent i is given by $u_i(M') = \sum_{j \in M} x_j \cdot u_i(j)$. Observe that under additivity, M' forms one part of a consensus halving exactly when

$$\sum_{j \in M} x_j \cdot u_i(j) = \frac{1}{2} \sum_{j \in M} u_i(j) \qquad \forall i \in N. \tag{1}$$

As we mentioned in the introduction, a consensus halving with no more than n cuts is guaranteed to exist regardless of the number of items. Our first result shows that such a division can be found efficiently for additive utilities.

Theorem 1. *For n agents with additive utilities, there exists a polynomial-time algorithm that computes a consensus halving with at most $\min\{n, m\}$ cuts.*

Proof. If $n \geq m$, a partition that divides every item in half is clearly a consensus halving and makes $m = \min\{n, m\}$ cuts. We therefore assume from now on that $n \leq m$ and describe a polynomial-time algorithm that computes a consensus halving using no more than n cuts.

The main idea of our algorithm is to start with the trivial consensus halving where $x_1 = x_2 = \cdots = x_m = 1/2$, and then gradually reduce the number of cuts.

We stop when the process cannot be continued, at which point we show that the consensus halving must contain at most n cuts. Our algorithm is presented below.

1. Let $x_1 = x_2 = \cdots = x_m = 1/2$.
2. Let S denote the set of n equations $\sum_{j \in M} \left(y_j - \frac{1}{2}\right) \cdot u_i(j) = 0$ for $i \in N$, and let $T = \emptyset$.
3. While there exists a solution $(y_1, \ldots, y_m) \neq (x_1, \ldots, x_m)$ to $S \cup T$, do the following:
   (a) For every $j \in M$ such that $y_j \neq x_j$, compute

   $$\gamma_j := \begin{cases} \frac{1-x_j}{y_j-x_j} & \text{if } y_j > x_j; \\ \frac{x_j}{x_j-y_j} & \text{if } y_j < x_j. \end{cases}$$

   (b) Let $j^* = \operatorname{argmin}_{j \in M, y_j \neq x_j} \gamma_j$.
   (c) For every $j \in M$, let $s_j := (1 - \gamma_{j^*}) \cdot x_j + \gamma_{j^*} \cdot y_j$, and update the value of x_j to s_j.
   (d) Add the equation $y_{j^*} = x_{j^*}$ to T.
4. Output $(x_1, \ldots, x_m)$.

Finding a solution $(y_1, \ldots, y_m)$ to $S \cup T$ that is not equal to $(x_1, \ldots, x_m)$ or determining that such a solution does not exist (Step 3) can be done in polynomial time via Gaussian elimination.[5] Moreover, it is obvious that the other steps of the algorithm run in polynomial time.

We next prove the correctness of our algorithm, starting with arguing that $(x_1, \ldots, x_m)$ forms a consensus halving. Since we start with a consensus halving $x_1 = \cdots = x_m = 1/2$, it suffices to show that each execution of the loop in Step 3 preserves the validity of the solution. Observe that, since both $(x_1, \ldots, x_m)$ and $(y_1, \ldots, y_m)$ are solutions to the Eqs. (1), their convex combination (in Step 3c) also satisfies the Eqs. (1). Furthermore, for each j such that $y_j \neq x_j$, the value γ_j is chosen so that if we replace γ_{j^*} by γ_j in the formula for s_j, we would have $s_j = 1$ for the case $y_j > x_j$, and $s_j = 0$ for the case $y_j < x_j$. Since $\gamma_{j^*} \leq \gamma_j$, we have that $s_j \in [0, 1]$ for all j such that $y_j \neq x_j$. In addition, the value of x_j does not change for j such that $y_j = x_j$. Thus, $(x_1, \ldots, x_m)$ remains a consensus halving throughout the algorithm.

Finally, we are left to show that at most n items are cut in the output $(x_1, \ldots, x_m)$. As noted above, our definition of γ_j ensures that $x_{j^*} \in \{0, 1\}$ after the execution of Step 3c. Furthermore, as the constraint $y_{j^*} = x_{j^*}$ is then immediately added to T, the value of x_{j^*} does not change for the rest of the algorithm. As a result, every item $j \in T$ is uncut. Thus, it suffices to show that $|T| \geq m - n$ at the end of the execution.

[5] Specifically, if the linear equations in $S \cup T$ lead to a unique solution $(x_1, \ldots, x_m)$, then Gaussian elimination immediately results in this solution. Otherwise, Gaussian elimination will yield a row echelon form; by setting one of the non-pivots y_j to be an arbitrary number not equal to x_j, we obtain a solution that is not equal to $(x_1, \ldots, x_m)$.

When the while loop in Step 3 terminates, $(x_1, \ldots, x_m)$ must be the unique solution to $S \cup T$. Recall that a system of linear equations with m variables can only have a unique solution when the number of constraints is at least m. This means that $|S \cup T| \geq m$ at the end of the algorithm. Since $|S| = n$, we must have $|T| \geq m - n$, as desired. □

Note that the above algorithm can be viewed as finding a vertex of the polytope defined by the constraints (1) and $0 \leq x_j \leq 1$ for all $j \in M$. In fact, it suffices to use a generic algorithm for this task; however, to the best of our knowledge, such algorithms often involve solving a linear program, whereas the algorithm presented above is conceptually simple and can be implemented directly. We also remark that our algorithm works even when some utilities $u_i(j)$ are negative, i.e., some of the items are goods while others are chores. Allocating a combination of goods and chores has received increasing attention in the fair division community [2,3].

As we discussed in the introduction, an important reason behind the positive result in Theorem 1 is the lack of linear order among the items. Indeed, as we show next, if the items lie on a line and we are only allowed to cut the line using n cuts, finding a consensus halving becomes computationally hard. This follows from discretizing the hardness result of Filos-Ratsikas et al. [11] and holds even if we allow the consensus halving to be approximate instead of exact. Formally, when the items lie on a line, we may place a number of cuts, with each cut lying either between two adjacent items or at some position within an item. All (fractional or whole) items between any two adjacent cuts must belong to the same fractional set of items in a partition, where the left and right ends of the line also serve as "cuts" in this requirement (see Fig. 1 for an example). We say that a partition into fractional sets of items (M_1, M_2) is an *ε-approximate consensus halving* if $|u_i(M_1) - u_i(M_2)| \leq \varepsilon \cdot u_i(M)$ for every agent i.

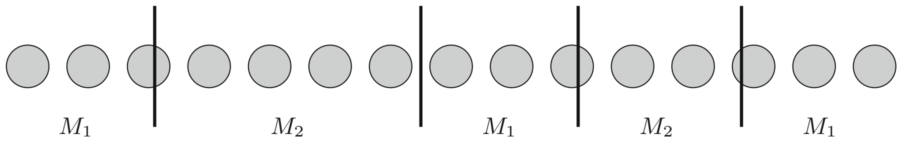


Fig. 1. Consensus halving for items on a line: in this example there are 15 items (represented by gray balls) that lie on a line and we have used 4 cuts to obtain a partition into fractional sets of items (M_1, M_2). The labels M_1 and M_2 indicate the set to which each segment belongs.

Theorem 2. *Suppose that the items lie on a line. There exists a polynomial p such that finding a $1/p(n)$-approximate consensus halving for n agents with at most n cuts on the line is PPA-hard, even if the valuations are binary and every agent values at most two contiguous blocks of items.*

The proof of Theorem 2, along with all other omitted proofs, can be found in the full version of our paper [13].

Although Theorem 1 allows us to efficiently compute a consensus halving with no more than n cuts in any instance, for some instances there exists a solution using fewer cuts. An extreme example is when all agents have the same utility function, in which case a single cut already suffices. This raises the question of determining the least number of cuts required for a given instance. Unfortunately, when there is a single agent, deciding whether there is a consensus halving that leaves all items uncut is already equivalent to the well-known NP-hard problem PARTITION. For general n, even computing a division that uses at most $n-1$ cuts more than the optimal solution is still computationally hard, as the following theorem shows.

Theorem 3. *For n agents with additive utilities, it is NP-hard to compute a consensus halving that uses at most $n-1$ cuts more than the minimum number of cuts for the same instance.*

Theorem 3 implies that there is no hope of finding a consensus halving with the minimum number of cuts or even a non-trivial approximation thereof in polynomial time, provided that P $\neq$ NP. Nevertheless, we show that instances that admit a consensus halving with fewer than n cuts are rare: if the utilities are drawn independently at random from probability distributions, then it is almost surely the case that any consensus halving needs at least n cuts. We say that a distribution is *non-atomic* if it does not put positive probability on any single point.

Theorem 4. *Suppose that for each $i \in N$ and $j \in M$, the utility $u_i(j)$ is drawn independently from a non-atomic distribution $\mathcal{D}_{i,j}$. Then, with probability 1, every consensus halving uses at least $\min\{n, m\}$ cuts.*

As our final remark of this section, consider utility functions that are again additive across items, but for which the utility of each item scales *quadratically* as opposed to linearly in the fraction of the item. That is, for a set M' containing a fraction x_j of item j, the utility of agent i is given by $u_i(M') = \sum_{j \in M} x_j^2 \cdot u_i(j)$. Even though these utility functions appear different from the ones we have considered so far, it turns out that the set of consensus halvings remains exactly the same. Indeed, a partition (M_1, M_2) is a consensus halving under the quadratic functions if and only if

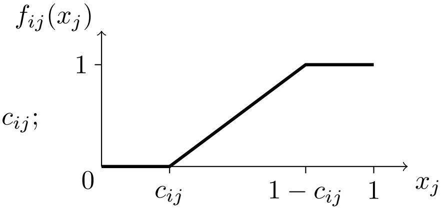
$$\sum_{j \in M} x_j^2 \cdot u_i(j) = \sum_{j \in M} (1 - x_j)^2 \cdot u_i(j) \qquad \forall i \in N.$$

Since $x_j^2 - (1 - x_j)^2 = x_j - (1 - x_j) = 2x_j - 1$, the above condition is equivalent to (1), so all of our results in this section apply to the quadratic functions as well.

3 Monotonic Utilities

Next, we turn our attention to utility functions that are no longer additive as in Sect. 2. We assume that the utilities are *monotonic*, meaning that the utility of an agent for a set of items cannot decrease upon adding any fraction of an item to the set. Our main result is that finding a consensus halving is computationally hard for such valuations; in fact, the hardness holds even when the utilities take on a specific structure that we call *symmetric-threshold*. Symmetric-threshold utilities are additive over items, and linear with symmetric thresholds within every item. Formally, the utility of agent i for a fractional set of items M' containing a fraction $x_j \in [0, 1]$ of each item j can be written as $u_i(M') = \sum_{j \in M} f_{ij}(x_j) \cdot u_i(j)$, where

$$f_{ij}(x_j) := \begin{cases} 0 & \text{if } x_j \le c_{ij}; \\ \frac{x_j - c_{ij}}{1 - 2c_{ij}} & \text{if } c_{ij} < x_j < 1 - c_{ij}; \\ 1 & \text{if } x_j \ge 1 - c_{ij}, \end{cases}$$



where $c_{ij} \in [0, 1/2)$ is the *threshold* or *cap* of agent i for item j. Intuitively, symmetric-threshold utilities model settings where having a small fraction of an item is the same as not having the item at all, while having a large fraction of the item is the same as having the whole item. The point where this threshold behavior occurs is controlled by the cap c_{ij}, which can be different for every pair $(i, j) \in N \times M$. It is easy to see that the resulting utility functions are indeed monotonic. Note that although general monotonic utility functions do not necessarily admit a concise representation (see the discussion preceding Theorem 7), symmetric-threshold utility functions can be described succinctly.

Even though symmetric-threshold utility functions are very close to being additive, we show that finding a consensus halving for such utilities is computationally hard. Recall that a partition (M_1, M_2) is an ε-approximate consensus halving if $|u_i(M_1) - u_i(M_2)| \le \varepsilon \cdot u_i(M)$ for every agent i.

Theorem 5. *There exists a constant $\varepsilon > 0$ such that finding an ε-approximate consensus halving for n agents with monotonic utilities that uses at most n cuts is PPAD-hard, even if all agents have symmetric-threshold utilities.*

At a high level, we prove this result by reducing from a modified version of the *generalized circuit* problem. The generalized circuit problem is the main tool that has been used (implicitly or explicitly) to prove hardness of computing Nash equilibria in various settings [7,8,23]. A generalized circuit is a generalization of an arithmetic circuit, because it allows *cycles*, which means that instead of a simple computation, the circuit now represents a constraint satisfaction problem. The version of the problem we use is different from the standard one in two aspects. First, instead of the domain $[0, 1]$, we use $[-1, 1]$, which is more adapted to the consensus halving problem. Second, we will only allow the circuit to use three types of arithmetic gates. We show that these modifications do not change the complexity of the problem.

4 Connections to Agreeable Sets

We now present some implications of results from consensus halving on the setting of computing agreeable sets. Let us first formally define the agreeable set problem, introduced by Manurangsi and Suksompong [18].[6] As in consensus halving, there is a set N of n agents and a set M of m items. Agent i has a monotonic utility function u_i over *non-fractional* sets of items, where we assume the normalization $u_i(\emptyset) = 0$; this corresponds to a set function.

Definition 2. *A subset of items $M' \subseteq M$ is said to be* agreeable to agent *i if $u_i(M') \geq u_i(M \backslash M')$.*

As one of their main results, Manurangsi and Suksompong [18] showed that for any n and m, there exists a set of at most $\min\left\{\lfloor \frac{m+n}{2} \rfloor, m\right\}$ items that is agreeable to all agents, and this bound is tight. Their proof relies on a graph-theoretic statement often referred to as "Kneser's conjecture", which specifies the chromatic number for a particular class of graphs called Kneser graphs. Here we present a short alternative proof that works by arranging the items on a line in arbitrary order, applying consensus halving, and rounding the resulting fractional partition. As a bonus, our proof yields an agreeable set that is composed of at most $\lfloor n/2 \rfloor + 1$ blocks on the line.

Theorem 6 ([18])**.** *For n agents with monotonic utilities, there exists a subset $M' \subseteq M$ such that*

$$|M'| \leq \min\left\{\left\lfloor \frac{m+n}{2} \right\rfloor, m\right\}$$

and M' is agreeable to all agents.

Proof. Let $s = \left\lfloor \frac{m+n}{2} \right\rfloor$. If $s \geq m$, the entire set of items M has size $m = \min\{s, m\}$ and is agreeable to all agents due to monotonicity, so we may assume that $s \leq m$. Arrange the items on a line in arbitrary order, and extend the utility functions of the agents to fractional sets of items in a continuous and monotonic fashion.[7] Consider a consensus halving with respect to the extended utilities that uses at most n cuts on the line; some of the cuts may cut through items, whereas the remaining cuts are between adjacent items. Let $r \leq n$ be the number of items that are cut by at least one cut. Without loss of generality, assume that the first part M' contains no more full items than the second part M'', so M' contains at most $\left\lfloor \frac{m-r}{2} \right\rfloor$ full items. By moving all cut items from

[6] The notion of agreeability was introduced in an earlier conference version of the paper [27]. Gourvès [14] considered an extension of the problem that takes into account matroidal constraints.

[7] For example, one can use the *Lovász extension* or the *multilinear extension* (see the full version of our paper [13]). .

M'' to M' in their entirety, M' contains at most $\lfloor \frac{m-r}{2} \rfloor + r = \lfloor \frac{m+r}{2} \rfloor \le s$ items. Since we start with a consensus halving and only move fractional items from M'' to M', we have that M' is agreeable to all agents. Moreover, one can check that M' is composed of at most $\lceil \frac{n+1}{2} \rceil = \lfloor \frac{n}{2} \rfloor + 1$ blocks on the line.

In light of Theorem 6, an important question is how efficiently we can compute an agreeable set whose size matches the worst-case bound. Manurangsi and Suksompong [18] addressed this question by providing a polynomial-time algorithm for two agents with monotonic utilities and three agents with "responsive" utilities, a class that lies between additive and monotonic utilities. They left the complexity for higher numbers of agents as an open question, and conjectured that the problem is hard even when the number of agents is a larger constant. We show that this is in fact not the case: the problem can be solved efficiently for any number of agents with additive utilities, as well as for any *constant* number of agents with monotonic utilities. Note that since the input of the problem for monotonic utilities can involve an exponential number of values (even for constant n), and consequently may not admit a succinct representation, we assume a "utility oracle model" in which the algorithm is allowed to query the utility $u_i(M')$ for any $i \in N$ and $M' \subseteq M$.

Theorem 7. *There exists a polynomial-time algorithm that computes a set of at most* $\min\left\{\lfloor \frac{m+n}{2} \rfloor, m\right\}$ *items that is agreeable to all agents, for each of the following two cases:*

(i) All agents have additive utilities.
(ii) All agents have monotonic utilities and the number of agents is constant (assuming access to a utility oracle).

5 Consensus k-Splitting

In this section, we address two important generalizations of consensus halving, both of which were mentioned by Simmons and Su [26]. In *consensus splitting*, instead of dividing the items into two equal parts, we want to divide them into two parts so that all agents agree that the split satisfies some given ratio, say two-to-one. In *consensus* $1/k$*-division*, we want to divide the items into k parts that all agents agree are equal. We consider a problem that generalizes both of these problems at once.

Definition 3. *Let* $\alpha_1, \ldots, \alpha_k > 0$ *be real numbers such that* $\alpha_1 + \cdots + \alpha_k = 1$*. A* consensus k-splitting with ratios $\alpha_1, \ldots, \alpha_k$ *is a partition of* M *into* k *fractional sets of items* $M_1, \ldots, M_k$ *such that*

$$\frac{u_i(M_1)}{\alpha_1} = \frac{u_i(M_2)}{\alpha_2} = \cdots = \frac{u_i(M_k)}{\alpha_k} \qquad \forall i \in N.$$

When the ratios are clear from context, we will simply refer to such a partition as a consensus k-splitting.

As in Sect. 2, we will assume that the utility functions are additive, in which case our desired condition is equivalent to $u_i(M_\ell) = \alpha_\ell \cdot u_i(M)$ for all $i \in N$ and $\ell \in [k]$.

While there is no reason to cut an item more than once in consensus halving, one may sometimes wish to cut the same item multiple times in consensus k-splitting in order to split the item across three or more parts. Hence, even though the number of cuts made is always at least the number of items cut, the two quantities are not necessarily the same in consensus k-splitting. If there are n items and each agent only values a single distinct item, then it is clear that we already need to make $(k-1)n$ cuts for any ratios $\alpha_1, \dots, \alpha_k$, in particular $k-1$ cuts for each item. Nevertheless, it could still be that for some ratios, it is always possible to achieve a consensus k-splitting by cutting fewer than $(k-1)n$ items. We show that this is not the case: for any set of ratios, cutting $(k-1)n$ items is necessary in the worst case.

Theorem 8. *For any ratios $\alpha_1, \dots, \alpha_k > 0$, there exists an instance with additive utilities in which any consensus k-splitting with these ratios cuts at least $(k-1)n$ items.*

Next, we show that computing a consensus k-splitting with at most $(k-1)n$ cuts can be done efficiently using a generalization of our algorithm for consensus halving (Theorem 1). Note that such a splitting also cuts at most $(k-1)n$ items.

Theorem 9. *For n agents with additive utilities and ratios $\alpha_1, \dots, \alpha_k$, there is a polynomial-time algorithm that computes a consensus k-splitting with these ratios using at most $(k-1) \cdot \min\{n, m\}$ cuts.*

As in Theorem 1, our algorithm does not require the nonnegativity assumption on the utilities and therefore works for combinations of goods and chores.

When the items lie on a line, there is always a consensus halving that makes at most n cuts on the line and therefore cuts at most n items—this matches the upper bound on the number of items cut in the absence of a linear order. Theorem 9 shows that the bound n continues to hold for consensus splitting into two parts with any ratios. As we show next, however, this bound is no longer achievable for some ratios with ordered items, thereby demonstrating another difference that the lack of linear order makes.[8]

Theorem 10. *Let $n \geq 2$, $k = 2$ and $(\alpha_1, \alpha_2) = (\frac{1}{n}, \frac{n-1}{n})$. There exists an instance such that the n agents have additive utilities, the items lie on a line, and any consensus k-splitting with ratios α_1 and α_2 makes at least $2n-4$ cuts on the line.*

For consensus halving, Theorem 4 shows that in a random instance, any solution almost surely uses at least the worst-case number of cuts $\min\{n, m\}$. One might consequently expect that an analogous statement holds for consensus k-splitting, with $(k-1) \cdot \min\{n, m\}$ cuts almost always being required. However,

[8] See the definition of the consensus halving problem on a line before Theorem 2.

we show that this is not true: even in the simple case where $n = 1$ and the agent's utilities are drawn from the uniform distribution over $[0, 1]$, it is likely that we only need to make one cut (instead of $k - 1$) for large m.

Theorem 11. *Let $n = 1$, and suppose that the agent's utility for each item is drawn independently from the uniform distribution on $[0, 1]$. For any ratios $\alpha_1, \dots, \alpha_k > 0$, with probability approaching 1 as $m \to \infty$, there exists a consensus k-splitting with these ratios using at most one cut. Moreover, there is a polynomial-time algorithm that computes such a solution.*

6 Conclusion

In this paper, we studied a natural version of the consensus halving problem where, in contrast to prior work, the items do not have a linear structure. We showed that computing a consensus halving with at most n cuts in our version can be done in polynomial time for additive utilities, but already becomes PPAD-hard for a class of monotonic utilities that are very close to additive. We also demonstrated several extensions and connections to the problems of consensus k-splitting and agreeable sets.

While our PPAD-hardness result serves as strong evidence that consensus halving for a set of items is computationally hard for non-additive utilities, it remains open whether the result can be strengthened to PPA-completeness—indeed, the membership of the problem in PPA follows from a reduction to consensus halving on a line, as explained in the introduction. Obtaining a PPA-hardness result will most likely require new ideas and perhaps even new insights into PPA, since all existing PPA-hardness results for consensus halving heavily rely on the linear structure. Of course, it is also possible that the problem is in fact PPAD-complete. In addition to consensus halving, settling the computational complexity of the agreeable sets problem for a non-constant number of agents with monotonic utilities would also be of interest.

Acknowledgments. This work was partially supported by the European Research Council (ERC) under grant number 639945 (ACCORD), by an EPSRC doctoral studentship (Reference 1892947), and by JST, ACT-X.

References

1. Alon, N.: Splitting necklaces. Adv. Math. **63**(3), 247–253 (1987)
2. Aziz, H., Caragiannis, I., Igarashi, A., Walsh, T.: Fair allocation of indivisible goods and chores. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 53–59 (2019)
3. Bogomolnaia, A., Moulin, H., Sandomirskiy, F., Yanovskaya, E.: Competitive division of a mixed manna. Econometrica **85**(6), 1847–1871 (2017)
4. Brams, S.J., Taylor, A.D.: Fair Division: From Cake-Cutting to Dispute Resolution. Cambridge University Press, Cambridge (1996)

5. Brams, S.J., Taylor, A.D.: The Win-Win Solution: Guaranteeing Fair Shares to Everybody. W. W. Norton & Company, New York (1999)
6. Caragiannis, I., Kurokawa, D., Moulin, H., Procaccia, A.D., Shah, N., Wang, J.: The unreasonable fairness of maximum Nash welfare. ACM Trans. Econ. Comput. **7**(3), 12:1–12:2 (2019)
7. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. J. ACM **56**(3), 14:1–14:57 (2009)
8. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. SIAM J. Comput. **39**(1), 195–259 (2009)
9. Filos-Ratsikas, A., Goldberg, P.W.: Consensus halving is PPA-complete. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC), pp. 51–64 (2018)
10. Filos-Ratsikas, A., Goldberg, P.W.: The complexity of splitting necklaces and bisecting ham sandwiches. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC), pp. 638–649 (2019)
11. Filos-Ratsikas, A., Hollender, A., Sotiraki, K., Zampetakis, M.: Consensus-halving: does it ever get easier? In: Proceedings of the 21st ACM Conference on Economics and Computation (EC), pp. 381–399 (2020)
12. Goldberg, C.H., West, D.B.: Bisection of circle colorings. SIAM J. Algebraic Discrete Methods **6**(1), 93–106 (1985)
13. Goldberg, P.W., Hollender, A., Igarashi, A., Manurangsi, P., Suksompong, W.: Consensus halving for sets of items. CoRR abs/2007.06754 (2020)
14. Gourvès, L.: Agreeable sets with matroidal constraints. J. Comb. Optim. **37**(3), 866–888 (2018). https://doi.org/10.1007/s10878-018-0327-1
15. Hobby, C.R., Rice, J.R.: A moment problem in L1 approximation. Proc. Am. Math. Soc. **16**(4), 665–670 (1965)
16. Kyropoulou, M., Suksompong, W., Voudouris, A.A.: Almost envy-freeness in group resource allocation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), pp. 400–406 (2019)
17. Manurangsi, P., Suksompong, W.: Asymptotic existence of fair divisions for groups. Math. Soc. Sci. **89**, 100–108 (2017)
18. Manurangsi, P., Suksompong, W.: Computing a small agreeable set of indivisible items. Artif. Intell. **268**, 96–114 (2019)
19. Moulin, H.: Fair Division and Collective Welfare. MIT Press, Cambridge (2003)
20. Onicescu, O.: Un théorème d'existence. Annali di Matematica Pura ed Applicata **103**(1), 1–2 (1974). https://doi.org/10.1007/BF02414140
21. Plaut, B., Roughgarden, T.: Almost envy-freeness with general valuations. SIAM J. Discrete Math. **34**(2), 1039–1068 (2020)
22. Roughgarden, T.: Twenty Lectures on Algorithmic Game Theory. Cambridge University Press, Cambridge (2016)
23. Rubinstein, A.: Inapproximability of Nash equilibrium. SIAM J. Comput. **47**(3), 917–959 (2018)
24. Sandomirskiy, F., Segal-Halevi, E.: Fair division with minimal sharing. CoRR abs/1908.01669 (2019)
25. Segal-Halevi, E., Nitzan, S.: Fair cake-cutting among families. Soc. Choice Welf. **53**(4), 709–740 (2019). https://doi.org/10.1007/s00355-019-01210-9
26. Simmons, F.W., Su, F.E.: Consensus-halving via theorems of Borsuk-Ulam and Tucker. Math. Soc. Sci. **45**(1), 15–25 (2003)
27. Suksompong, W.: Assigning a small agreeable set of indivisible items to multiple players. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), pp. 489–495 (2016)