Contiguous Cake Cutting: Hardness Results and Approximation Algorithms

Paul W. Goldberg, Alexandros Hollender, Warut Suksompong
Department of Computer Science
University of Oxford

Abstract

We study the fair allocation of a cake, which serves as a metaphor for a divisible resource, under the requirement that each agent should receive a contiguous piece of the cake. While it is known that no finite envy-free algorithm exists in this setting, we exhibit efficient algorithms that produce allocations with low envy among the agents. We then establish NP-hardness results for various decision problems on the existence of envy-free allocations, such as when we fix the ordering of the agents or constrain the positions of certain cuts. In addition, we consider a discretized setting where indivisible items lie on a line and show a number of hardness results strengthening those from prior work.

1 Introduction

We consider the classical *cake cutting* problem, where we wish to divide a cake among a set of agents with different preferences over different parts of the cake. The cake serves as a metaphor for any divisible resource such as time or land, and our aim is to perform the division in a *fair* manner. This problem has a long and storied history that dates back over 70 years and has received attention from mathematicians, economists, and computer scientists alike (Brams and Taylor 1996; Robertson and Webb 1998; Procaccia 2016).

In order to reason about fairness, we need to specify when a division is considered to be fair. One of the most commonly used definitions is *envy-freeness*, which means that no agent envies another with respect to the division. In other words, among the pieces in the division, every agent receives their first choice. An early result by Dubins and Spanier (1961) shows that an envy-free allocation always exists for arbitrary valuations of the agents. However, as Stromquist (1980) noted, this result depends on a liberal definition of what constitutes a piece of cake, and an agent "who hopes only for a modest interval of cake may be presented instead with a countable union of crumbs."

In light of this concern, Stromquist (1980) strengthened the result of Dubins and Spanier by showing that it is possible to guarantee an envy-free allocation in which every agent receives a *contiguous* piece of the cake. Stromquist's result, together with its topological proof, is widely regarded as a cornerstone of the cake cutting literature. Nevertheless, since the result focuses only on the existence of a contiguous envy-free allocation, it leaves open the question of how to compute such an allocation. Almost 30 years later, Stromquist himself addressed this question and showed that under the Robertson-Webb model, where an algorithm is allowed to discover the agents' valuations through *cut* and *evaluate* queries, no finite algorithm can compute a contiguous envy-free allocation when there are at least three agents (Stromquist 2008).[1]

Although Stromquist's later result rules out the possibility of computing contiguous envy-free allocations in general, several important questions still remain. For instance, can we compute a contiguous allocation with low envy between the agents, and if so, how efficiently? How does the answer change if we know that the agents' valuations belong to a restricted class? What happens if we add extra requirements on the allocation, such as fixing a desired ordering of the agents or constraining the positions of certain cuts? The goal of this paper is to shed light on the complexity of contiguous cake cutting by addressing these questions.

1.1 Our Contributions

First, in Section 3 we present two algorithms that compute an allocation with low envy in polynomial time. As is standard in the cake-cutting literature, we represent the cake by the interval $[0, 1]$ and normalize the agents' valuations so that each agent has value 1 for the entire interval. Our first algorithm works for general valuations under the Robertson-Webb model and produces a contiguous allocation in which any agent has envy no more than $1/3$ towards any other agent. On the other hand, our second algorithm is specific to valuations where each agent only desires a single subinterval and has a uniform value over that interval—for such valuations, the algorithm produces a contiguous allocation with a lower envy of at most $1/4$.

Next, in Section 4, we consider variants of the cake-cutting problem where we impose constraints on the desired allocation. We show that for several natural variants, the de-

[1] For two agents, the well-known *cut-and-choose* protocol, which lets the first agent cut the cake into two equal pieces and lets the second agent choose the piece that she prefers, computes a contiguous envy-free allocation.

cision problem of whether there exists a contiguous envy-free allocation satisfying the corresponding constraints is NP-hard. In particular, this holds for the variants where (i) a certain agent must be allocated the leftmost piece; (ii) the ordering of the agents is fixed; and (iii) one of the cuts must fall at a given position. Fixing the ordering of the agents is relevant when there is a temporal ordering in which the agents must be served, e.g., due to notions of seniority or the ease of switching from one agent to another in the service. Likewise, fixing a cut point is applicable when we divide a parcel of land and there is a road crossing the parcel, so we cannot allocate a piece that lies on both sides of the road. Moreover, our construction serves as a general framework that can be used to obtain hardness results for other related variants.

Finally, in Section 5 we investigate a discrete analog of cake cutting, where there are indivisible items on a line and each agent is to be allocated a contiguous block of items. The discrete setting can be viewed as a type of restriction for the continuous setting, where cuts must be placed between discrete items. In addition to envy-freeness, we work with two other well-studied fairness notions: *proportionality* and *equitability*.[2] Using a single reduction, we show that deciding whether there exists a contiguous fair allocation is NP-hard for each of the three fairness notions as well as any combination of them; our result holds even when all agents have *binary* valuations[3] and moreover value the same number of items. This significantly strengthens a result of Bouveret et al. (2017), who established the hardness for proportionality and envy-freeness using additive but non-binary valuations. In addition, we prove that when the valuations are binary and every agent values a contiguous block of items, deciding whether a contiguous proportional allocation exists is also NP-hard.

1.2 Further Related Work

Since the seminal work of Stromquist (1980), a number of researchers have studied cake cutting in view of the contiguity condition. Su (1999) proved the existence of contiguous envy-free allocations using Sperner's lemma arguments. Deng, Qi, and Saberi (2012) showed that contiguous envy-free cake cutting is PPAD-complete; however, the result requires non-standard (e.g., non-additive) valuation functions. Aumann, Dombb, and Hassidim (2013) considered the problem of maximizing social welfare with contiguous pieces, while Bei et al. (2012) tackled the same problem with the added requirement of proportionality. Cechlárová and Pillárová (2012) and Cechlárová, Doboš, and Pillárová (2013) examined the existence and computation of contiguous equitable allocations—among other things, they showed that such an allocation is guaranteed to exist even if we fix the ordering of the agents. Aumann and Dombb (2015) analyzed the trade-off between fairness and social welfare in contiguous cake cutting.

The contiguity requirement has also been considered in the context of indivisible items. Marenco and Tetzlaff (2014) proved that if the items lie on a line and every item is positively valued by at most one agent, a contiguous envy-free allocation is guaranteed to exist. When each item can yield positive value to any number of agents, Barrera et al. (2015), Bilò et al. (2019), and Suksompong (2019) showed that various relaxations of envy-freeness can be fulfilled. In addition, contiguity has been studied in the more general model where the items lie on an arbitrary graph (Bouveret et al. 2017; Igarashi and Peters 2019; Bei et al. 2019).

Recently, Arunachaleswaran et al. (2019) developed an efficient algorithm that computes a contiguous cake division with multiplicatively bounded envy—in particular, each agent's envy is bounded by a multiplicative factor of 3. We remark that our approximation algorithms are incomparable to their result. On the one hand, their algorithm may return an allocation wherein an agent has value $1/4$ for her own piece and $3/4$ for another agent's piece—this corresponds to an additive envy of $1/2$. On the other hand, our algorithms may leave some agents empty-handed, leading to unbounded multiplicative envy. We also note that additive envy is the more commonly considered form of approximation, both for cake cutting (Deng, Qi, and Saberi 2012; Brânzei and Nisan 2017; 2019) and for indivisible items (Lipton et al. 2004; Caragiannis et al. 2016).

2 Preliminaries

For any positive integer n, let $[n] = \{1, 2, \ldots, n\}$. In our cake cutting setting, we consider the cake as the interval $[0, 1]$. There are n agents whose preferences over the cake are represented by valuation functions $v_1, \ldots, v_n$. Assume that these valuation functions are non-negative density functions over $[0, 1]$. We abuse notation and let $v_i(a, b) = v_i([a, b]) = \int_a^b v_i(x)dx$ for $0 \leq a \leq b \leq 1$. It follows that the valuations are non-negative, additive, and non-atomic (i.e., $v_i(a, a) = 0$). We assume further that the valuations are normalized so that $v_i(0, 1) = 1$ for every $i \in [n]$.

A *contiguous allocation* of the cake is a partition of $[0, 1]$ into n (possibly empty) intervals, along with an assignment of each interval to an agent, so that every agent gets exactly one interval. Note that this means that we cut the cake using $n - 1$ cuts. Formally, a contiguous allocation is represented by the cut positions $0 \leq x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq 1$ and a permutation $\pi : [n] \to [n]$ that assigns the intervals to the agents so that agent i receives the interval $[x_{\pi(i)-1}, x_{\pi(i)}]$, where we define $x_0 = 0$ and $x_n = 1$ for convenience.

We are interested in finding a contiguous allocation that is *envy-free*, i.e., no agent thinks that another agent gets a better interval. Formally, the contiguous allocation (x, π) is envy-free if for all $i, j \in [n]$, we have $v_i(x_{\pi(i)-1}, x_{\pi(i)}) \geq v_i(x_{j-1}, x_j)$. In some cases we will be interested in finding a contiguous allocation that is only *approximately* envy-free. For $\varepsilon \in [0, 1]$, the contiguous allocation (x, π) is *ε-envy-free* if for all $i, j \in [n]$, we have $v_i(x_{\pi(i)-1}, x_{\pi(i)}) \geq v_i(x_{j-1}, x_j) - \varepsilon$. In other words, any agent has envy that is at most a fraction ε of her value for the whole cake.

A typical way for an algorithm to access the valuation functions is through queries in the *Robertson-Webb model*: the algorithm can make *evaluate* queries—where it speci-

[2] See the definitions in Section 5.

[3] That is, the valuations are additive and each agent values each item either 0 or 1.

fies x, y and asks agent i to return the value $v_i(x, y)$—and *cut* queries—where it specifies x, α and asks agent i to return the leftmost point y such that $v_i(x, y) = \alpha$. A more restrictive class of valuations is that of *piecewise constant* valuations. A piecewise constant valuation function is defined by a piecewise constant density function on $[0, 1]$, i.e., a step function. This class of valuations can be explicitly represented as part of the input. A subclass of piecewise constant valuations is the class of *piecewise uniform* valuations, where the density function of agent i is either some fixed rational constant c_i or 0.

3 Approximation Algorithms

In this section, we present two algorithms for approximate envy-free cake cutting. Algorithm 1 works for arbitrary valuations and returns a $1/3$-envy-free allocation. On the other hand, Algorithm 2 can be used for piecewise uniform valuations with a single value-block and outputs a $1/4$-envy-free allocation. Note that such valuations are relevant, for example, when the agents are dividing machine processing time: each agent has a release date and a deadline for her job, so she would like to maximize the processing time she obtains after the release date and before the deadline.

Algorithm 1 $1/3$-Envy-Free Algorithm for Arbitrary Valuations

```
1:  procedure ApproximateEFArbitrary
2:    ℓ ← 0, N ← [n]
3:    for i ∈ N do
4:      M_i ← ∅
5:    while some agent in N values [ℓ, 1] at least 1/3 do
6:      for i ∈ N do
7:        if v_i(ℓ, 1) ≥ 1/3 then
8:          r_i ← leftmost point such that v_i(ℓ, r_i) = 1/3
9:        else
10:         r_i ← 1
11:     j ← arg min_{i∈N} r_i, r ← min_{i∈N} r_i
12:     M_j ← [ℓ, r]
13:     ℓ ← r, N ← N\{j}
14:   if N ≠ ∅ then
15:     j ← arbitrary agent in N
16:     M_j ← [ℓ, 1]
17:   else
18:     j ← last agent removed from N
19:     M_j ← M_j ∪ [ℓ, 1]
20:   return (M_1, ..., M_n)
```

While Algorithm 1 can be implemented for general valuations under the Robertson-Webb model, it also allows a simple interpretation as a moving-knife algorithm. In this interpretation, the algorithm works by moving a knife over the cake from left to right. Whenever the current piece has value $1/3$ to at least one remaining agent, the piece is allocated to one such agent. If the knife reaches the right end of the cake, then the piece is allocated to an arbitrary remaining agent if there is at least one remaining agent, and to the agent who received the last piece otherwise.

Theorem 3.1. *For n agents with arbitrary valuations, Algorithm 1 returns a contiguous $1/3$-envy-free allocation and runs in time polynomial in n assuming that it makes queries in the Robertson-Webb model.*

Proof. Every agent receives a single interval from the algorithm; the only possible exception is agent j in line 19. However, since j is chosen as the last agent removed from N, the interval M_j allocated to j earlier is adjacent to $[\ell, 1]$, meaning that j also receives a single interval. Hence the allocation is contiguous. Moreover, the algorithm only needs to make queries in lines 5, 7 and 8, and the number of necessary queries is clearly polynomial in n. The remaining steps can be implemented in polynomial time.

We now prove that the envy of an agent i towards any other agent is at most $1/3$. If i is assigned a piece in the while loop (line 5), i receives value at least $1/3$. This means that i's value for any other agent's piece is at most $2/3$, so i's envy is no more than $1/3$. Alternatively, after the while loop, i still has not received a piece, meaning that $N \neq \emptyset$ in line 14. By our allocation procedure in the while loop, i values any piece assigned in the while loop at most $1/3$. Furthermore, when the algorithm enters line 14, i values the interval $[\ell, 1]$ less than $1/3$. Since $[\ell, 1]$ is assigned to an agent who did not receive an interval earlier, it follows that i does not envy any other agent more than $1/3$, as claimed. □

Note that if we are only interested in having an algorithm that makes a polynomial number of queries, Brânzei and Nisan (2017) showed that for any $\varepsilon > 0$, a contiguous ε-envy-free allocation can be found using $O(n/\varepsilon)$ queries, which is polynomial in n for constant ε. Their algorithm works by cutting the cake into pieces of size $1/\varepsilon$ and performing a brute-force search over the space of all contiguous allocations with respect to these cuts; this algorithm therefore has exponential computational complexity (even for constant ε). By contrast, in the absence of the contiguity constraint, Procaccia (2016) gave a simple polynomial-time algorithm that computes an ε-envy-free allocation for any constant ε. His algorithm also starts by cutting the cake into pieces of size $1/\varepsilon$ and then lets agents choose their favorite pieces in a round-robin manner; consequently, the resulting allocation can be highly non-contiguous.

While we do not know whether the bound $1/3$ in our approximation can be improved under the computational efficiency requirement,[4] we show next that if the agents have piecewise uniform valuations and each agent only values a single interval, the envy can be reduced to $1/4$. Alijani et al. (2017) showed that if the valuations are as described and moreover the n valued intervals satisfy an "ordering property", meaning that no interval is a strict subinterval of another interval, then a contiguous envy-free allocation can be computed efficiently. Nevertheless, the ordering property is a very strong assumption, and indeed reducing the envy to

[4] For the case $n = 3$, Deng, Qi, and Saberi (2012) gave a fully polynomial-time approximation scheme that computes a contiguous ε-envy-free allocation for any $\varepsilon > 0$.

Algorithm 2 $1/4$-Envy-Free Algorithm for Uniform Single-Interval Valuations

▷ R_i : the single interval valued by agent i
▷ $\text{mid}(i)$: the midpoint of R_i
▷ A_i : part of R_i that is unallocated at the start of agent i's turn
▷ an interval is *restrained* if it is adjacent to an interval that has already been allocated

1: **procedure** APPROXIMATEEFSINGLEINTERVAL
2: Order the agents $1, \ldots, n$ so that $|R_i| \leq |R_j|$ for all $i < j$
3: **for** $i = 1, \ldots, n$ **do**
4: **if** there exists a *restrained* interval $I \subseteq A_i$ with $v_i(I) = 1/4$ and $\text{mid}(i) \in I$ **then** ▷ Case 1
5: $M_i \leftarrow I$
6: **else if** there exists an interval $I \subseteq A_i$ with $v_i(I) = 1/4$ and $\text{mid}(i) \in I$ **then** ▷ Case 2
7: $S_i \leftarrow \{j > i \mid v_i(\min(\text{mid}(i), \text{mid}(j)), \max(\text{mid}(i), \text{mid}(j))) \leq 1/4\}$
8: $k \leftarrow \min S_i$
9: $M_i \leftarrow$ an interval $I \subseteq A_i$ with $v_i(I) = 1/4$, $\text{mid}(i) \in I$ and $\text{mid}(k) \in \partial I$ (i.e., an endpoint of I)
10: **else if** there exist $\ell < i$ with $\text{mid}(i) \in M_\ell$ and an interval $I \subseteq A_i$ adjacent to M_ℓ with $v_i(I) = 1/4$ **then** ▷ Case 3
11: $M_i \leftarrow I$
12: **else** ▷ Case 4
13: $M_i \leftarrow$ a largest restrained interval $I \subseteq A_i$ with $v_i(I) \leq 1/4$
14: **if** some two intervals M_q, M_r are adjacent (say, M_q is to the left of M_r) **then**
15: Extend M_q and all assigned intervals to its left as far as possible to the left.
16: Extend M_r and all assigned intervals to its right as far as possible to the right.
17: **else**
18: Extend assigned intervals arbitrarily to cover the remaining cake.
19: **return** $(M_1, \ldots, M_n)$

$1/4$ without this assumption already requires significant care in assigning the pieces.[5]

At a high level, Algorithm 2 first orders the agents from shortest to longest desired interval, breaking ties arbitrarily. For each agent in the ordering, if an interval of value $1/4$ containing the midpoint of her valued interval (perhaps at the edge of the former interval) has not been taken, the agent takes one such interval. Else, if an interval of value $1/4$ is available somewhere, the agent takes one such interval; here, if there are choices on both sides of the midpoint, the agent may need to be careful to pick the "correct" one. Otherwise, if no interval of value $1/4$ is available, the agent takes a largest available interval. At the end of this process, part of the cake may remain unallocated. If some pair of assigned intervals are adjacent, pick one such pair, and allocate the remaining cake by extending pieces away from the border between this pair. Else, extend the pieces arbitrarily to cover the remaining cake.

Theorem 3.2. *For n agents with piecewise uniform valuations such that each agent only values a single interval, Algorithm 2 returns a contiguous $1/4$-envy-free allocation and runs in time polynomial in n.*

Proof. One can check that Algorithm 2 assigns a single interval to every agent and can be implemented in polynomial time. It remains to show that the algorithm returns an allocation such that for any two agents i, j, agent i has envy at most $1/4$ towards agent j. For the purpose of this proof, when we refer to an interval M_i, we mean the interval before it is extended in the final phase of the algorithm (the *extension phase* starting at line 14). We denote by M_i^+ the corresponding extended interval that is returned by the algorithm. For any agent i and any interval I, the i-value of I is the value of I for agent i, i.e., $v_i(I)$.

When agent i's turn comes in the for-loop, it falls into exactly one of four possible cases: Case 1 (line 4), Case 2 (line 6), Case 3 (line 10) or Case 4 (line 12). Depending on which case applies, M_i is chosen accordingly. We say that the *single-direction extension* (SDE) property holds, if at least one agent does not fall into Case 2. It is easy to check that if the SDE property holds, then there are at least two allocated intervals M_q and M_r that are adjacent before the extension phase begins, and thus every interval M_i will be extended in a single direction.

It is clear that $v_i(M_j^+) \geq v_i(M_j)$ for all i, j. Furthermore, in all four cases it holds that agent i is allocated an interval of value at most $1/4$, i.e., $v_i(M_i) \leq 1/4$ for all i. Since $M_i \subseteq R_i$ and because of the way the agents are ordered, it follows that

$$v_i(M_j) \leq 1/4 \quad \text{for all } j \leq i \tag{1}$$

We now show that any agent i has envy at most $1/4$ at the end of the algorithm. Namely, we prove that for any agents i, j we have $v_i(M_j^+) \leq v_i(M_i^+) + 1/4$. We treat the four different cases that can occur during agent i's turn.

Cases 1 *and* 2. In both cases, M_i contains $\text{mid}(i)$ and has i-value $1/4$. This also holds for $M_i^+ \supseteq M_i$. Since the midpoint of R_i is contained in M_i^+, any other interval M_j^+ has i-value at most $1/2$. Thus, agent i has envy at most $1/4$.

[5] Alijani et al. (2017) also showed that for piecewise uniform valuations where each agent only values a single interval (without the ordering property assumption), one can efficiently compute an envy-free allocation with at most $2n - 1$ intervals in total.

Case 3. In this case, we again have $v_i(M_i) = 1/4$. However, this time we have $\text{mid}(i) \in M_\ell$, which implies that $v_i(M_j^+) \leq 1/2$ for all $j \neq \ell$. Thus, it remains to show that $v_i(M_\ell^+) \leq 1/2$. Since $\ell < i$, we have $v_i(M_\ell) \leq 1/4$. Thus, we need to show that the extension of M_ℓ to M_ℓ^+ increases the i-value by at most $1/4$. Since M_i was chosen to be adjacent to M_ℓ, it suffices to show that there is at most $1/4$ i-value available on the other side of M_ℓ.

To this end, we prove that at the start of agent i's turn, M_ℓ cannot have at least $1/4$ of i-value available both on the left side and on the right side. Assume on the contrary that this is the case. Note, in particular, that M_ℓ is not restrained. Thus, M_ℓ was allocated in agent ℓ's turn by Case 2. We also know that $i \in S_\ell$, because $\text{mid}(i), \text{mid}(\ell) \in M_\ell$ and $v_\ell(M_\ell) = 1/4$. Now there are two cases:

- If $i = \min S_\ell$, then $\text{mid}(i) \in \partial M_\ell$. But in that case, at the start of agent i's turn, there exists a restrained interval $I \subseteq A_i$ with $v_i(I) = 1/4$ and $\text{mid}(i) \in I$. Thus, agent i would have been in Case 1 instead of 3.
- If $i > k = \min S_\ell$, then in agent k's turn, Case 1 will apply. Indeed, $\text{mid}(k) \in \partial M_\ell$ and thus there is at least $1/4$ of k-value available that contains $\text{mid}(k)$ (because there is enough space for $1/4$ of i-value and $i > k$). But if Case 1 applies, then M_k will be chosen to be adjacent to M_ℓ (since they both contain $\text{mid}(k)$), and M_ℓ will not have space available on both sides when agent i's turn comes.

Case 4. First, suppose that $v_i(M_i) < 1/4$. This means that M_i was a largest available interval in R_i. It follows that any agent $j > i$ can obtain an interval of i-value at most $v_i(M_i)$, since it is processed after i. For $j < i$, since agent i is in Case 4, the SDE property holds. Thus, M_j can be extended by at most $v_i(M_i)$, i.e., $v_i(M_j^+) \leq v_i(M_j) + v_i(M_i)$ for all j. With (1) it follows that the envy is at most $1/4$.

Now, consider the case where $v_i(M_i) = 1/4$. Any agent $j > i$ can obtain i-value $< 1/2$—otherwise, agent i would have fallen in Case 1 or 2. Consider any $j < i$:

- if $\text{mid}(i) \in M_j$, then both on the left and right side of M_j the space available has i-value $< 1/4$ (otherwise agent i would be in Case 1 or 3). Since the SDE property holds, it follows that $v_i(M_j^+) \leq v_i(M_j) + 1/4 \leq 1/2$ with (1).
- if $\text{mid}(i) \notin M_j$, then $v_i(M_j^+) < 1/2$. Otherwise, it means that M_j is extended in a single direction (SDE property) and takes over an interval of i-value at least $1/4$ that contains $\text{mid}(i)$. But then, agent i would be in Case 1 or 2.

This completes the proof. □

4 Hardness for Cake-Cutting Variants

In this section, we establish hardness results for a number of decision problems on the existence of contiguous envy-free allocations.

Theorem 4.1. *The following decision problems are NP-hard for contiguous cake cutting, even if we restrict the valuations to be piecewise uniform:*

- *Does there exist an envy-free allocation in which agent* 1 *obtains the left-most piece?*
- *Does there exist an envy-free allocation in which the pieces are allocated to the* n *agents in the order* $1, 2, \ldots, n$*?*
- *Does there exist an envy-free allocation such that there is a cut at position* x*, for* x *given in the input?*

These problems remain NP-hard if we replace envy-freeness by ε*-envy-freeness for any sufficiently small constant* ε*.*

This list is not exhaustive: additional results of the same flavor can be found in the full proof (Goldberg, Hollender, and Suksompong 2019).[6] The following proof sketch conveys the main ideas behind these results.

Proof Sketch. In order to prove that these decision problems are NP-hard, we reduce from 3-SAT. Namely, given a 3-SAT formula, we construct a cake-cutting instance such that the answer to the decision problem is "Yes" if and only if the 3-SAT formula is satisfiable. A bonus of our proof is that we construct a single cake-cutting instance that works for all of the decision problems mentioned in the Theorem statement and even a few more.

Let us give some insight into how this instance is constructed. Consider a 3-SAT formula $C_1 \vee \cdots \vee C_m$, where the C_i are clauses containing 3 literals using the variables $x_1, \ldots, x_n$ and their negations. The cake-cutting instance is constructed by putting together multiple small cake-cutting instances, so-called gadgets. For every clause C_i we introduce a Clause-Gadget with its three corresponding agents C_i^1, C_i^2 and C_i^3. The intuition here is that C_i^1 is associated to the first literal appearing in C_i, C_i^2 to the second one, and C_i^3 to the third one. For any Clause-Gadget agent A, we let $\ell(A)$ denote the associated literal. The valuations of these agents inside the gadget are as shown in Figure 1. We say that the gadget *operates correctly* if it contains exactly two cuts and the three resulting pieces go to the three agents C_i^1, C_i^2 and C_i^3. At this point we can already make a first key observation: if the gadget operates correctly, at least one of the three agents must be *sad*, i.e., obtain at most one out of its three blocks of value in this gadget.

For every variable x_j we introduce a Variable-Gadget with its two corresponding agents L_j and R_j. Apart from these two agents, some Clause-Gadget agents will also have a value-block inside this gadget. In more detail, all the Clause-Gadget agents that correspond to x_j or $\overline{x}_j$ will have a block of value inside the Variable-Gadget for x_j. Figure 2 shows how the value-blocks are arranged inside the gadget. We say that the gadget operates correctly if it contains exactly one cut and the two resulting pieces go to L_j and R_j. There is a second key observation to be made here. Assume

[6] However, if we fix all $n - 1$ cuts, the problem becomes solvable in polynomial time. Indeed, with all the cuts fixed, the resulting pieces are also all fixed. We can therefore construct a bipartite graph with the agents on one side and the pieces on the other side, where there is an edge between an agent and a piece exactly when receiving the piece would make the agent envy-free, and check the existence of a perfect matching in this graph.

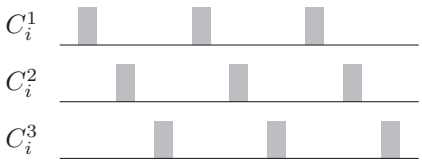

Figure 1: Clause-Gadget for clause C_i: the valuations of its three agents inside the gadget. Every block in this figure has value 0.24.

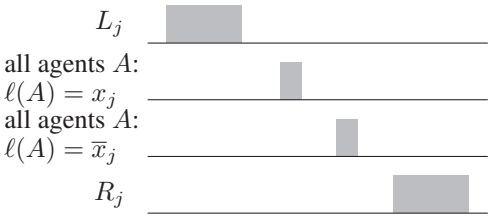

Figure 2: Variable-Gadget for variable x_j: the valuations of its two agents L_j and R_j, as well as the Clause-Gadget agents that have value in this gadget. The large blocks have value 1 each and the small blocks have value 0.28 each.

that all gadgets operate correctly. If some agent C_i^k with $\ell(C_i^k) = x_j$ (or $\overline{x}_j$) is sad, then the value-block of C_i^k in the Variable-Gadget for x_j has to contain a cut (otherwise C_i^k would be envious). Since the Variable-Gadget contains exactly one cut, it is impossible to have agents A and B with $\ell(A) = x_j$ and $\ell(B) = \overline{x}_j$ that are both sad.

The instance is constructed by positioning the gadgets one after the other on the cake. Starting from the left and moving to the right, we first put the Clause-Gadget for C_1, then C_2, and so on until C_m, and then the Variable-Gadget for x_1, then x_2, and so on until x_n. Between adjacent gadgets we introduce a small interval without any value-blocks. We say that an envy-free allocation is *nice* if all the gadgets operate correctly.

Let us now see how a nice envy-free allocation yields a satisfying assignment for the 3-SAT formula. For any agent C_i^k that is sad, we set the corresponding literal $\ell(C_i^k)$ to be true. This means that if $\ell(C_i^k) = x_j$, then we set x_j to be true, and if $\ell(C_i^k) = \overline{x}_j$, then we set x_j to be false. The first key observation above tells us that every Clause-Gadget has at least one sad agent. Thus, this assignment of the variables ensures that every clause is satisfied. However, we have to make sure that this assignment is consistent, i.e., we never set x_j to be both true and false. This consistency is enforced by the Variable-Gadget for x_j and the second key observation above.

Conversely, given a satisfying assignment for the 3-SAT formula, it is not too hard to construct a nice envy-free allocation. This proves NP-hardness for the decision problem "Does there exist a nice envy-free allocation?". In order to prove the result for the more natural decision problems stated in Theorem 4.1, the construction has to be extended with some additional work. □

5 Hardness for Indivisible Items

We now turn to a discrete analog of cake cutting, where we wish to allocate a set of indivisible items that lie on a line subject to the requirement that each agent must receive a contiguous block. As in cake cutting, we assume that the valuations of the agents over the items are additive, and that all items must be allocated. Besides envy-freeness, we consider the classical fairness notions of proportionality and equitability. An allocation is *proportional* if every agent receives value at least $1/n$ times her value for the whole set of items, and *equitable* if all agents receive the same value.

Unlike in cake cutting, for indivisible items there may be no allocation satisfying any of the three fairness properties, e.g., when two agents try to divide a single item. Bouveret et al. (2017) showed that deciding whether an envy-free allocation exists is NP-hard for additive valuations, and the same is true for proportionality; they did not consider equitability. In this section, we extend and strengthen their results in several ways. We consider *binary valuations*, which are additive valuations such that the value of each agent for each item is either 0 or 1. In other words, an agent either "wants" an item or not. Even though binary valuations are much more restrictive than additive valuations, as we will see, several problems still remain hard even for this smaller class.

First, we show that deciding whether a fair allocation exists is NP-hard for each of the three fairness notions mentioned. This hardness result holds for *any* non-empty combination of the three notions and even if all agents want the same number of items. Moreover, we present a reduction that establishes the hardness for all combinations in one fell swoop. We remark that the techniques of Bouveret et al. (2017) do not extend to the binary domain because each agent can have different values for different items in their construction. One may try to fix this by breaking items into smaller items to obtain a binary valuation, but each agent will require a different way of breaking items, and moreover there will be allocations in the new instance that cannot be mapped back to those in the original instance.

Theorem 5.1. *Let*

$$F = \{\textit{envy-freeness, proportionality, equitability}\},$$

and let $\emptyset \neq X \subseteq F$. Deciding whether an instance with indivisible items on a line admits a contiguous allocation satisfying all properties in X is NP-hard, even if all agents have binary valuations and value the same number of items.

Proof. We prove this result with a single reduction. Let I be an instance of 3-SAT with m clauses $C_1, \dots C_m$ using the variables $x_1, \dots, x_n$ and their negations. We create the following gadgets.

- Clause-Gadget: For every clause C_i we introduce three agents: C_i^1, C_i^2, C_i^3. Each of these agents is associated with one of the three literals that appear in the clause C_i. We denote by $\ell(C_i^k)$ the literal associated with C_i^k. For every clause C_i we construct a Clause-Gadget. The gadget consists of four contiguous items that are all valued by all three agents C_i^1, C_i^2, C_i^3, and by no one else.

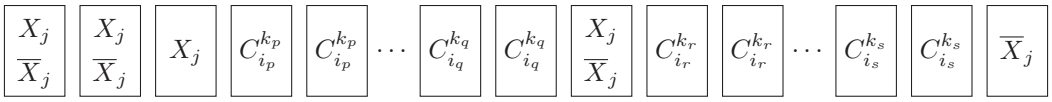

Figure 3: Variable-Gadget for variable x_j. Every item is represented by a rectangle containing the agent(s) who value it.

- Variable-Gadget: For every variable x_j we introduce two agents, X_j and $\overline{X}_j$, and construct a Variable-Gadget as follows (Figure 3). Starting from the left, create two items that are valued by both X_j and $\overline{X}_j$ (and no one else). Then, create one item that is valued only by X_j. Then, for every C_i^k such that $\ell(C_i^k) = x_j$, create two items that are valued only by C_i^k. Then, create an item that is valued by both X_j and $\overline{X}_j$. Then, for every C_i^k such that $\ell(C_i^k) = \overline{x}_j$, create two items that are valued only by C_i^k. Finally, create an item that is valued only by $\overline{X}_j$.

We combine these gadgets to create the instance R as follows. Starting from the left, construct the Clause-Gadget for each clause C_i. Then, construct the Variable-Gadget for each variable x_j. Thus, we obtain an instance with $3m+2n$ agents and $4m + (5n + 6m) = 5n + 10m$ items.

Claim. The following statements hold:

- Any contiguous allocation in R where every agent gets at least two items they value yields a satisfying assignment for I. This holds even if the allocation is *partial*, i.e., some items are not allocated.
- Any satisfying assignment for I yields a contiguous envy-free allocation in R where every agent gets exactly two items they value.

We leave the proof of the Claim to the full version of this paper (Goldberg, Hollender, and Suksompong 2019).

The final step of the proof is to introduce one last gadget. The Special-Gadget creates $3m+2n+7$ new agents; denote the set of these agents by N. The gadget consists of $2(3m + 2n)+14 = 6m+4n+14$ new items. These items are valued by all agents in N. For every $i \in [m]$ and $k \in [3]$, C_i^k values all new items except the rightmost six. For every $j \in [n]$, X_j and $\overline{X}_j$ value all new items except the rightmost four.

The Special-Gadget is added to the right end of R and yields the final instance R'. Note that in R' there are $6m + 4n + 7$ agents and every agent values exactly $6m + 4n + 14$ items. Now consider any contiguous allocation for R'.

- If the allocation is proportional, then every agent gets at least $\lceil (6m + 4n + 14)/(6m + 4n + 7) \rceil = 2$ items they value. It follows that the agents in N get all the new items, because $2|N| = 2(3m + 2n + 7) = 6m + 4n + 14$. This means that the other agents get at least two items they value in R. By the claim above, we obtain a satisfying assignment.
- If the allocation is equitable, then all agents get exactly s items they value, for some $s \geq 0$. The Special-Gadget contains an item (in fact, many) that is valued by all agents. Since this item will be allocated to someone, $s = 0$ is not possible. Also $s \geq 3$ is not possible, because the $3m + 2n + 7$ agents in N all like the exact same $2(3m + 2n + 7)$ items. Now, since all $6m + 4n + 7$ agents value the first $(6m+4n+14)-6 = 6m+4n+8$ items in the Special-Gadget, at least one of them will be allocated to two of those (by the pigeonhole principle). It follows that $s = 1$ is also impossible. Thus, only $s = 2$ remains, and we again obtain a satisfying assignment by the claim.

Since envy-freeness implies proportionality, it follows that any X-allocation for R' yields a satisfying assignment for the 3-SAT instance I, for any non-empty $X \subseteq \{\text{envy-free, proportional, equitable}\}$. On the other hand, any satisfying assignment for the 3-SAT instance yields an envy-free and equitable allocation for R', by assigning two contiguous Special-Gadget items to each agent in N and then using the claim. □

In the construction used for our proof of Theorem 5.1, each agent values at most four contiguous block of items. In light of this result, one may naturally wonder whether the hardness continues to hold if, for example, every agent values a single block of items. We show that this is the case for proportionality, provided that we drop the requirement that all agents value the same number of items. Note that if each agent values a contiguous block of items *and* all agents value the same number of items, deciding whether a proportional allocation exists can in fact be done in polynomial time; see the details in the full version of this paper (Goldberg, Hollender, and Suksompong 2019).

Theorem 5.2. *Deciding whether an instance with indivisible items on a line admits a contiguous proportional allocation is NP-hard, even if the valuations are binary and every agent values a contiguous block of items.*

The proof of this result, as well as that of the next result, uses a reduction from the NP-complete problem 3-PARTITION and can be found in the full version.

Finally, we show that under the same conditions as Theorem 5.2, deciding whether there exists a proportional and equitable allocation, or an equitable allocation that gives the agents positive value, are both NP-hard. Since agents do not all value the same number of items (unlike in Theorem 5.1), we normalize the valuations so that if agent i values x_i items, she has value $1/x_i$ of each of them (so her total value is 1).

Theorem 5.3. *Deciding whether an instance with indivisible items on a line admits*

- *a contiguous allocation that is both proportional and equitable;*
- *a contiguous equitable allocation in which the agents receive positive value*

are both NP-hard, even if the valuations are binary and every agent values a contiguous block of items.

6 Conclusion

In this paper, we study the classical cake cutting problem with the contiguity constraint and establish several hardness results and approximation algorithms for this setting. It is worth noting that while our $1/3$-envy-free algorithm (Algorithm 1) is simple, lowering the envy to $1/4$ for the restricted class of uniform single-interval valuations (Algorithm 2) already requires significantly more work. Pushing the approximation factor down further even for this class or the class of piecewise uniform valuations while maintaining computational efficiency is therefore a challenging direction. Of course, it is possible that there are hardness results for sufficiently small constants—this is not implied by the work of Deng, Qi, and Saberi (2012), as their PPAD-completeness result relies on more complex valuation functions.

On the hardness front, we provide constructions that serve as frameworks for deriving NP-hardness results for both cake cutting and indivisible items. Nevertheless, our frameworks do not cover questions related to the utilities of the agents, for instance whether there exists a contiguous envy-free allocation of the cake in which the first agent receives at least a certain level of utility. Extending or modifying our constructions to deal with such questions is an interesting direction for future research.

Acknowledgments

This work was partially supported by the European Research Council (ERC) under grant number 639945 (ACCORD) and by an EPSRC doctoral studentship (Reference 1892947). We would like to thank the anonymous reviewers for their helpful comments.

References

Alijani, R.; Farhadi, M.; Ghodsi, M.; Seddighin, M.; and Tajik, A. S. 2017. Envy-free mechanisms with minimum number of cuts. In *AAAI*, 312–318.

Arunachaleswaran, E. R.; Barman, S.; Kumar, R.; and Rathi, N. 2019. Fair and efficient cake division with connected pieces. In *WINE*, Forthcoming.

Aumann, Y., and Dombb, Y. 2015. The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation* 3(4):23.

Aumann, Y.; Dombb, Y.; and Hassidim, A. 2013. Computing socially-efficient cake divisions. In *AAMAS*, 343–350.

Barrera, R.; Nyman, K.; Ruiz, A.; Su, F. E.; and Zhang, Y. 2015. Discrete envy-free division of necklaces and maps. *CoRR* abs/1510.02132.

Bei, X.; Chen, N.; Hua, X.; Tao, B.; and Yang, E. 2012. Optimal proportional cake cutting with connected pieces. In *AAAI*, 1263–1269.

Bei, X.; Igarashi, A.; Lu, X.; and Suksompong, W. 2019. Connected fair allocation of indivisible goods. *CoRR* abs/1908.05433.

Bilò, V.; Caragiannis, I.; Flammini, M.; Igarashi, A.; Monaco, G.; Peters, D.; Vinci, C.; and Zwicker, W. S. 2019. Almost envy-free allocations with connected bundles. In *ITCS*, 14:1–14:21.

Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; and Peters, D. 2017. Fair division of a graph. In *IJCAI*, 135–141.

Brams, S. J., and Taylor, A. D. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.

Brânzei, S., and Nisan, N. 2017. The query complexity of cake cutting. *CoRR* abs/1705.02946.

Brânzei, S., and Nisan, N. 2019. Communication complexity of cake cutting. In *EC*, 525.

Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A. D.; Shah, N.; and Wang, J. 2016. The unreasonable fairness of maximum Nash welfare. In *EC*, 305–322.

Cechlárová, K., and Pillárová, E. 2012. On the computability of equitable divisions. *Discrete Optimization* 9(4):249–257.

Cechlárová, K.; Doboš, J.; and Pillárová, E. 2013. On the existence of equitable cake divisions. *Information Sciences* 228:239–245.

Deng, X.; Qi, Q.; and Saberi, A. 2012. Algorithmic solutions for envy-free cake cutting. *Operations Research* 60(6):1461–1476.

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

Goldberg, P. W.; Hollender, A.; and Suksompong, W. 2019. Contiguous cake cutting: Hardness results and approximation algorithms. *CoRR* abs/1911.05416.

Igarashi, A., and Peters, D. 2019. Pareto-optimal allocation of indivisible goods with connectivity constraints. In *AAAI*, 2045–2052.

Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *EC*, 125–131.

Marenco, J., and Tetzlaff, T. 2014. Envy-free division of discrete cakes. *Discrete Applied Mathematics* 164:527–531.

Procaccia, A. D. 2016. Cake cutting algorithms. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 13, 311–329.

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

Stromquist, W. 1980. How to cut a cake fairly. *The American Mathematical Monthly* 87(8):640–644.

Stromquist, W. 2008. Envy-free cake divisions cannot be found by finite protocols. *The Electronic Journal of Combinatorics* 15:#R11.

Su, F. E. 1999. Rental harmony: Sperner's lemma in fair division. *The American Mathematical Monthly* 106(10):930–942.

Suksompong, W. 2019. Fairly allocating contiguous blocks of indivisible items. *Discrete Applied Mathematics* 260:227–236.