Envy-free Mechanisms with Minimum Number of Cuts

Reza Alijani[‡], **Majid Farhadi**[*], **Mohammad Ghodsi**[*†], **Masoud Seddighin**[*], and **Ahmad S. Tajik**[§]
Sharif University of Technology[*], Duke University[‡], University of Michigan – Ann Arbor[§]
Institute for Research in Fundamental Sciences (IPM) – School of Computer Science [†]
alijani@cs.duke.edu, {m_farhadi, mseddighin}@ce.sharif.edu, ghodsi@sharif.edu, tajik@umich.edu

Abstract

We study the problem of fair division of a heterogeneous resource among strategic players. Given a divisible heterogeneous cake, we wish to divide the cake among n players in a way that meets the following criteria: ($\mathbb{I}$) every player (weakly) prefers his allocated cake to any other player's share (such notion is known as envy-freeness), ($\mathbb{II}$) the mechanism is strategy-proof (truthful), and ($\mathbb{III}$) the number of cuts made on the cake is minimal. We provide methods, namely *expansion process* and *expansion process with unlocking*, for dividing the cake under different assumptions on the valuation functions of the players.

1 Introduction

The problem of dividing a cake among a set of individuals has been widely studied in the past 60 years. The subject was first defined by Steinhaus (1948). The description of the problem is as follows: given a heterogeneous cake and a set of players, with potentially different tendencies to different parts of the cake, how to cut the cake and distribute it among the players in a fair manner?

Several notions are defined for measuring the fairness of an allocation (see (Procaccia 2014) for details). One of the most important notions is *envy-freeness*. An allocation of the cake is envy-free if each player (weakly) prefers its allocated share to any other player's share.

Envy-free resource allocation has been vastly studied in the literature. For two players, the famous method of *cut and choose* guarantees envy-freeness of the allocation. For three players, Selfridge and Conway designed a protocol for finding an envy-free division of the cake. In their method, a player may receive more than one piece (see (Procaccia 2013) for details). Brams and Taylor generalized this method to an arbitrary number of players (1995). However, their method doesn't guarantee any upper bound on the number of cuts. Recently, in (2016), Aziz and Mackenzie suggested a bounded envy-free protocol for any number of players.

In some settings, the number of cuts is also important. In several papers (e.g. (Stromquist 1980), (Barbanel and Brams 2004), (Stromquist 2007), (Bei et al. 2012)) the cake cutting with minimum number of cuts has been studied. Each cut might have an additional cost. As an example, suppose the cake models a processing time that must be fairly allocated among a set of tasks. Every task-switch imposes an overhead; minimizing total amount of overhead would be equivalent to minimizing the number of cuts on the cake. In addition, players may not have any value for very small pieces made by a large number of cuts. In (Caragiannis, Lai, and Procaccia 2011), this issue was illustrated by the advertisement example: think of the cake as time and consider the allocation of advertising time. In such a setting, a large number of cuts can yield so small periods of time that are not useful for advertising. In an allocation with small number of cuts this problem is unlikely.

Stromquist, in (1980), proved the existence of an envy-free division of the cake among n players with $n-1$ cuts which is the minimum number of cuts required to divide a cake among n players. However, the proof is not constructive and does not yield a polynomial time algorithm. In (2007), he showed that no finite protocol can find an envy-free allocation with minimum number of cuts for $n \geq 3$. (Deng, Qi, and Saberi 2012) proved that the problem of finding an envy-free allocation of the cake, with a minimum number of cuts is PPAD-Complete. They also proposed an FPTAS for the case of three players.

In a number of the recent papers (e.g. (Caragiannis, Lai, and Procaccia 2011), (Brams et al. 2012), (Bei et al. 2012), (Maya and Nisan 2012), (Chen et al. 2013), (Aziz and Ye 2014)) some restricted classes of valuation functions have been studied. Piecewise constant and piecewise uniform valuation functions are two important special classes of valuation functions which are very important in practice. One of the important properties of these valuation functions is that they can be described concisely. (Kurokawa, Lai, and Procaccia 2013) proved that finding an envy-free allocation (in Robertson-Webb model) when the valuation functions are piecewise-uniform is as hard as solving the problem without any restriction on the valuation functions.

Recently, some studies considered the problem from a game theoretic viewpoint. Many cake cutting algorithms are not truthful. For example, even the *cut and choose* method which is relatively simple does not guarantee truthfulness. In (Brânzei et al. 2016), the strategic outcome of the cake cutting algorithms has been studied. They proved the existence of an approximate subgame perfect Nash equilibrium for a class of protocols. Another line of research which is more related to our work, attempts to find truthful mechanisms. Similar to fairness, there are different notions for the concept of truthfulness. In (Brams et al. 2006), a weak notion of truthfulness is defined: players don't risk telling a lie, if there exists a scenario (for other players valuations) in which lying results in a lower payoff. As an example, they showed that *cut and choose* protocol is weakly truthful. Maya and Nisan (2012) designed truthful and Pareto-efficient mechanisms to divide the cake between two players where each player is interested in a subset of the cake, uniformly.

In (2013), Chen *et al.* considered a strong notion of truthfulness (denoted by strategy-proofness), in which the players' dominant strategies are to reveal their true valuations over the cake. They presented a strategy-proof mechanism for the case when the valuation functions are piecewise uniform. They also designed a randomized algorithm that is envy-free and truthful in expectation, for piecewise linear valuation functions. However, their method for dividing the cake uses $\Omega(n^2m)$ cuts, where m is the number of pieces in each valuation function. Aziz and Ye (2014) considered the problem when valuation functions are piecewise constant/uniform. Based on parametric network flows, they designed an envy-free algorithm that is *group strategy-proof*[1] for piecewise uniform valuations. It is notable that their method becomes equivalent to mechanism 1 from (Chen et al. 2013), in the case of piecewise uniform valuations.

1.1 Our work

We investigate the problem of finding envy-free and truthful mechanisms with a small number of cuts. By small, we mean that the number of cuts does not exceed $O(nm)$, where m is the number of steps of each player's (piecewise constant) valuation function. To the best of our knowledge, this is the first study that aims to approximate the number of cuts.

The basis of our method is a simple and elegant process called *expansion process*. After describing the process, we start with the case, where each player's valuation function is piecewise constant with only one step and preserves a specific property that we name *ordering property*. For this case, we propose EFISM which is a polynomial time, strategy-proof and envy-free allocation with $n-1$ cuts (Theorem 2).

Next, we remove the ordering assumption and show that a generalized form of the *expansion process* can find an envy-free allocation that cuts the cake into at most $2n-1$ pieces in polynomial time (Theorem 3). Furthermore, using a more complex form of this process, we propose EFGISM, which is a polynomial time algorithm that is truthful, envy-free and cuts the cake into at most $2n-1$ pieces (Theorem 4).

In addition, we consider the case where the valuation functions are piecewise constant with m pieces. When the number of players is constant, we provide a $\mathrm{poly}(m)$ time algorithm for envy-free division of the cake with $n-1$ cuts. Finally, we consider the case that the players possess a particular property, namely *intersection property* and show that under this assumption, a modification of the *expansion process* yields a $\mathrm{poly}(m, n)$ time, envy-free algorithm that cuts the cake in $O(nm)$ locations.

[1]Group strategy-proof means no group of players can misreport their valuations, such that in the resulting allocation all of them earn more payoff

2 Model Description and Preliminaries

In this paper, we use the term interval for two purposes: valuation functions and the shares allocated to the players. For brevity, denote the former type of intervals by $\mathcal{I}$ and the latter by I. Also, we Suppose that for every valuation interval $\mathcal{I}_i, \mathcal{I}_i = [\alpha_i, \beta_i]$ and for every share interval $I_i, I_i = [a_i, b_i]$.

Given a set $\mathcal{N}$ of n players and a cake $\mathcal{C}$. We represent the cake by the interval $[0, 1]$. For every player $p_i \in \mathcal{N}$, a valuation function $\nu_i : [0, 1] \to \mathbb{R}$ is given.

For each $p_i \in \mathcal{N}$ and interval $I = [a, b]$, we define $V_i(I)$ as $\int_a^b \nu_i(x)dx$. Our assumption is that the values of the players are normalized, such that $V_i(\mathcal{C}) = 1$, for each player p_i. A *piece* of the cake, is a set of mutually disjoint sub-intervals of $[0, 1]$. For a piece P, we define $V_i(P)$ as $\sum_{I \in P} V_i(I)$.

A valuation function ν is *piecewise constant*, if there exists a set $S_\nu = \{\mathcal{I}_{\nu 1}, \mathcal{I}_{\nu 2}, \dots, \mathcal{I}_{\nu k}\}$ of mutually disjoint intervals, such that for any two points x, x' in $\mathcal{I}_{\nu i}$, $\nu(x) = \nu(x') = r_i$ and for any point x that does not belong to any interval in S_v, $\nu(x) = 0$. To put it simply, a piecewise constant valuation consists of a finite number of intervals, such that all the points in the same interval have the same value, and for the points that do not belong to any interval, the valuation is 0. We say ν has k steps, if $|S_\nu| = k$.

A *division* of the cake among a set $\mathcal{N}$ of n players is a set $D = \{P_1, P_2, \dots, P_n\}$ of pieces, with each piece $P_i = \{I_{i,1}, I_{i,2}, \dots, I_{i,|P_i|}\}$ being a set of intervals with the following two properties: ($\mathbb{I}$) every pair of intervals are mutually disjoint and ($\mathbb{II}$) no piece of the cake is left behind: $\bigcup_{i,j} I_{i,j} = \mathcal{C}$.

The number of cuts in division D is $(\sum_i |P_i|) - 1$. A division $D = \{P_1, P_2, \dots, P_n\}$ is *envy-free*, if for every player p_i and piece $P_j \in D$ the inequality $V_i(P_i) \geq V_i(P_j)$ holds.

The majority of this paper is focused on the case, where each valuation function is a single interval. For this case, we suppose that for every player $p_i \in \mathcal{N}$, $S_{v_i} = \{\mathcal{I}_i\}$, where $\mathcal{I}_i = [\alpha_i, \beta_i]$. Furthermore, denote by $\mathcal{T}$ the set of valuation intervals, i.e., $\mathcal{T} = \{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_n\}$. In this setting, the envy-free notion for a division D can be interpreted as follows: for each player p_i and $k \neq i$ we have

$$\sum_j |I_{i,j} \cap \mathcal{I}_i| \geq \sum_j |I_{k,j} \cap \mathcal{I}_i|.$$

For a set of intervals X, we define $\mathsf{DOM}(X)$ as the minimal interval that includes all members of X as sub-intervals; e.g., in the case that each valuation function is a single interval, for a set $T \subseteq \mathcal{T}$ we have:

$$\mathsf{DOM}(T) = [\min_{\mathcal{I}_j \in T} \alpha_j, \max_{\mathcal{I}_i \in T} \beta_i].$$

Furthermore, we define the density of X, denoted by $\Phi(X)$ as: $\lambda(X)/|X|$ where $\lambda(X)$ is the total length of $\mathsf{DOM}(X)$ that is covered by at least one interval in X. We call a set X of intervals *solid*, if for every point $x \in \mathsf{DOM}(X)$, there

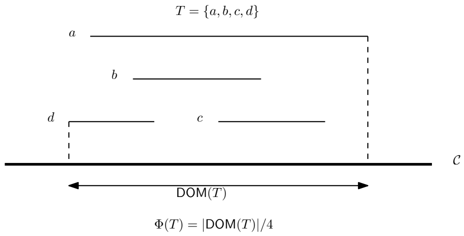

Figure 1: Domain and density

exists an interval I in X such that $x \in I$. For example, in Fig 1, the set T is solid. When T is solid, we have:

$$\lambda(T) = |\mathsf{DOM}(T)| = \max_{\mathcal{I}_i \in T} \beta_i - \min_{\mathcal{I}_j \in T} \alpha_j$$

Our assumption is that every piece of the cake is valuable for at least one player. In the Appendix[2], we show that slightly modified versions of our algorithms can handle the cases where this assumption does not hold.

3 The Expansion process

The main tool in our method for dividing the cake is a procedure called *expansion process*. The expansion process expands some associated intervals to the players, inside their desired area. We use $exp(T)$ to refer to the expansion process on the set T of valuation intervals. We initiate the expansion process for T by associating a zero length interval I_i at the beginning of its corresponding $\mathcal{I}_i \in T$, i.e., $I_i = [a_i = \alpha_i, b_i = \alpha_i]$. Denote by S, the set of these Intevals. We expand the intervals in S concurrently, all from the endpoint. The expansion is performed in a way that preserves two invariants:($\mathbb{I}$) The expansion has the same speed for all the intervals so as the lengths of the intervals remains the same and ($\mathbb{II}$) I_i always remains within $\mathcal{I}_i$.

During the expansion, the endpoint of an interval I_i may collide with the starting point of another interval I_j. In this case, I_i pushes the starting point of I_j forward during the expansion. The push continues to the end of the process. If I_i pushes I_j, we say I_i is *stuck* in I_j. Note that by the way we initiate the process, the intervals remain sorted according to their corresponding α_i's. Also in the special case of equal α_i for two players, the one with smaller β_i comes first.

Definition 1. *During the expansion, an interval I_i becomes locked, if the endpoint of I_i reaches β_i.*

Definition 2. *A* chain *is a sequence of intervals $I_{\sigma_1}, I_{\sigma_2}, \dots, I_{\sigma_k}$, with the property that for $1 \leq i < k$, I_{σ_i} is stuck in $I_{\sigma_{i+1}}$. A chain is locked, if I_{σ_k} is locked.*

The size of a chain is the number of intervals in that chain. By definition, a single interval is a chain of size 1.

The expansion ends when an interval becomes locked. The termination condition ensures that the second invariant is always preserved. In the Appendix , you can see a detailed example of the expansion process.

[2]The long version with appendix is available at www.cs.duke.edu/~alijani/EMNC-AAAI2017.pdf

Definition 3. *The expansion process for T is perfect, if the associated intervals cover the entire $\mathsf{DOM}(T)$. If the process terminates due to a locked interval before entirely covering $\mathsf{DOM}(T)$, the process is imperfect.*

Note that if an expansion process on T ends perfectly, then for every associated interval I_i, we have $|I_i| = \Phi(T)$.

Despite the fact that we described the expansion process continuously, it can be efficiently implemented via swiping of the events (see the Appendix for more details).

Observation 1. *During the expansion process, every interval I_i is either being pushed by another interval, or its starting point is still on α_i.*

4 EFISM: Special Interval Scheduling

In this section, we assume that the valuation function of each player is a single interval. In addition, we suppose that the intervals have the following property:

$$\forall i, j \qquad \alpha_i \leq \alpha_j \iff \beta_i \leq \beta_j \tag{1}$$

In other words, no interval is a sub-interval of another (unless they start or end in the same place). For this case, we present a polynomial time, envy-free, and truthful allocation with $n-1$ cuts. We name this algorithm as EFISM.

The idea in EFISM is repeatedly expanding the intervals and removing the locked chains. Let $\mathcal{T}$ be the valuation intervals corresponding to the players in $\mathcal{N}$. We begin by calling $exp(\mathcal{T})$. As described in Section 3, the procedure terminates either perfectly or imperfectly. In the first case we are done. Otherwise, at least one chain is locked. Let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \dots, I_{\sigma_k}$ be a locked chain in S with maximal size. Since $\mathscr{C}$ is maximal, no interval gets stuck in I_{σ_1}. By Observation 1, a_{σ_1} is exactly on α_{σ_1}. Let $\mathscr{T}$ be the set of valuation intervals corresponding to the intervals in $\mathscr{C}$.

Lemma 1. $\mathsf{DOM}(\mathscr{T}) = [\alpha_{\sigma_1}, \beta_{\sigma_k}]$.

Now, we allocate each I_{σ_i} to p_{σ_i}. Lemma 2 states that such an allocation is envy-free for $p_{\sigma_1}, p_{\sigma_2}, \dots, p_{\sigma_k}$.

Lemma 2. *For every interval I_{σ_i} and I_{σ_j} in $\mathscr{C}$, we have $V_{\sigma_i}(I_{\sigma_i}) \geq V_{\sigma_i}(I_{\sigma_j})$.*

Next, we remove players $p_{\sigma_1}, p_{\sigma_2}, \dots, p_{\sigma_k}$ from $\mathcal{N}$. We also remove $\mathsf{DOM}(\mathscr{T})$ from $\mathcal{C}$. By removing $\mathsf{DOM}(\mathscr{T})$, the cake is divided into two sub-cakes: the piece to the right of $\mathsf{DOM}(\mathscr{T})$ and the piece to the left of $\mathsf{DOM}(\mathscr{T})$, respectively $\mathcal{C}_l$ and $\mathcal{C}_r$. Let $\mathcal{N}_l$ ($\mathcal{N}_r$) be the set of players with their share inside $\mathcal{C}_l$ ($\mathcal{C}_r$). Also, let $\mathcal{T}_l$ and $\mathcal{T}_r$ be the sets of valuation intervals corresponding to $\mathcal{N}_l$ and $\mathcal{N}_r$.

Now, we update the valuation functions of the players in $\mathcal{C}_l$ and $\mathcal{C}_r$. Specifically, for every player $p_i \in \mathcal{N}_l$ with $\beta_i > \alpha_{\sigma_1}$ we change the value of β_i to α_{σ_1}. Similarly, for every player $p_j \in \mathcal{N}_r$ with $\alpha_j < \beta_{\sigma_k}$ we change α_j to β_{σ_k}.

After removing the allocated piece along with its corresponding players and updating the valuations, we perform this expansion and removal independently for both $\mathcal{T}_l$ and $\mathcal{T}_r$. The process continues until all the players are removed. In Algorithm 1, you can find a psudocode for EFISM. In addition, you can find a detailed example in the Appendix.

Algorithm 1 EFISM algorithm

function EFISM($\mathcal{C}, \mathcal{N}, \mathcal{T}$)
    $\triangleright$ $\mathcal{C}$ corresponds to the interval $[a, b]$
  if $\mathcal{C} \neq \emptyset$ **then**
    $exp(\mathcal{T})$ $\triangleright$ Expansion process on T
    $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$: a maximal locked chain
    for $1 \leq i \leq k$ **do**
      Allocate I_{σ_i} to p_{σ_i}
    $\mathcal{C}_l = [a, \alpha_{\sigma_1}]$
    $\mathcal{C}_r = [\beta_{\sigma_k}, b]$
    for every $p_k \in \mathcal{N}$ **do**
      if $a_k < a_{\sigma_1}$ **then**
        $\beta_k = \min(\beta_k, \alpha_{\sigma_1})$
        Add $p_k, \mathcal{I}_k$ to $\mathcal{N}_l$, $\mathcal{T}_l$ respectively
      else if $b_k > b_{\sigma_k}$ **then**
        $\alpha_k = \max(\alpha_k, \beta_{\sigma_k})$
        Add $p_k, \mathcal{I}_k$ to $\mathcal{N}_r$, $\mathcal{T}_r$ respectively
    EFISM($\mathcal{C}_l, \mathcal{N}_l, \mathcal{T}_l$)
    EFISM($\mathcal{C}_r, \mathcal{N}_r, \mathcal{T}_r$)

Theorem 2. EFISM *is envy-free, truthful and cuts the cake in exactly* $n-1$ *locations.*

Remark that removing the ordering property described in the beginning of this section may result in an inappropriate allocation. For example, consider the input described in Figure 2. Clearly, running EFISM on this input does not yield an envy-free allocation; here p_c envies p_b. In addition, the allocation does not allocate the entire cake, because a piece between I_c and I_b is left over.

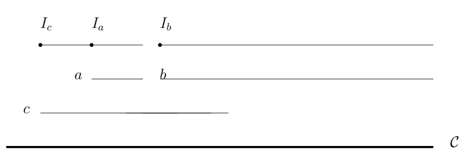

Figure 2: EFISM for intervals *without* ordering property

5 Expansion Process with Unlocking

In this section, we introduce a more general form of the expansion process. The idea is the fact that during the expansion process, there might be some cases that a locked chain can become unlocked by re-permuting some of its intervals.

Definition 4. *Let* $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ *be a maximal locked chain. A permutation* $I_{\delta_1}, I_{\delta_2}, \ldots, I_{\delta_r}$ *of the intervals in* $\mathscr{C}$ *is said to be* $\mathscr{C}$*-unlocking, if the following conditions are held:* $(\mathbb{I})$ $\forall_i, I_{\delta_i} \in \mathscr{C}$ *and* $\delta_r = \sigma_k$,$(\mathbb{II})$ *For all* $1 \leq j \leq r-1$, $a_{\delta_j} \geq \alpha_{\delta_{j+1}}$ *and* $b_{\delta_j} < \beta_{\delta_{j+1}}$,$(\mathbb{III})$ $\alpha_{\delta_1} \leq a_{\delta_r}$ *and* $\beta_{\delta_1} > b_{\delta_r}$.

The intuition behind the definition of unlocking permutation is as follows: let $I_{\delta_1}, I_{\delta_2}, \ldots, I_{\delta_r}$ be a $\mathscr{C}$-unlocking permutation, where $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$. Then, we can change the order of intervals in $\mathscr{C}$ by placing I_{δ_j} in the location of $I_{\delta_{j-1}}$ for $1 < j \leq r$ and placing I_{δ_1} in the location of I_{δ_r}. By the definition of unlocking permutation, after such operations $I_{\delta_r}(I_{\sigma_k})$ is no longer locked. Thus, I_{σ_k} is not a barrier for the expansion and the process can be continued.

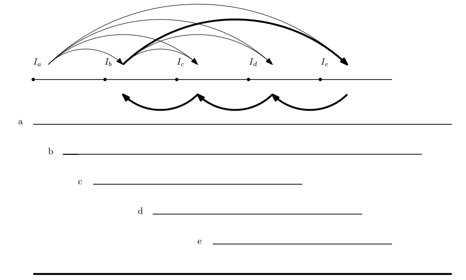

Figure 3: Example of a Permutation Graph. Here the locked chain I_a, I_b, I_c, I_d, I_e can be unlocked by permutation $\left(\begin{smallmatrix} I_a & I_b & I_c & I_d & I_e \\ I_a & I_e & I_b & I_c & I_d \end{smallmatrix}\right)$

Definition 5. *A locked chain* $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ *is strongly locked, if* $\mathscr{C}$ *admits no unlocking permutation that contains* I_{σ_k}.

Definition 6. *The expansion process with unlocking* U-$exp(.)$ *is strongly locked, if at least one of its chains is strongly locked.*

For a set T of valuation intervals, we use U-$exp(T)$ to refer to the expansion process with unlocking. The expansion process with unlocking is in fact, the same as expansion process with the exception that when the process is faced with a locked chain, it tries to unlock the chain by an unlocking permutation. If the chain becomes unlocked, the expansion goes on. The process runs until either the entire DOM(T) is allocated (perfect) or a strongly locked chain occurs (imperfect). In the Appendix you can find a detailed example.

It is worth mentioning that there may be multiple locked intervals in a moment. In such situations, we separately try to unlock each interval.

Definition 7. *A permutation graph for a locked chain* $\mathscr{C}$ *is a directed graph* $G_{\mathscr{C}}\langle V, E\rangle$. *For every interval* $I_{\sigma_i} \in \mathscr{C}$, *there is a vertex* v_{σ_i} *in* V. *The edges in* E *are in two types* E_l *and* E_r, *i.e.,* $E = E_l \cup E_r$. *The edges in* E_l *and* E_r *are determined as follows:* $(\mathbb{I})$ *For each* I_{σ_i} *and* I_{σ_j}, *the edge* $(v_{\sigma_i}, v_{\sigma_j})$ *is in* E_l, *if* $i > j$ *and* $\alpha_{\sigma_i} \leq a_{\sigma_j}$.$(\mathbb{II})$ *For each* I_{σ_i} *and* I_{σ_j}, *the edge* $(v_{\sigma_i}, v_{\sigma_j})$ *is in* E_r, *if* $i < j$ *and* $\beta_{\sigma_i} > b_{\sigma_j}$. *See Figure 3 for an example of permutation graph.*

A trivially necessary and sufficient condition for a chanin $\mathscr{C}$ to be strongly locked is that $G_{\mathscr{C}}$ contains no cycle containing v_{σ_k}. However, regarding the special structure of $G_{\mathscr{C}}$, we can define a stronger necessary and sufficient condition for a strongly locked situation.

Definition 8. *A directed cycle* C *in* $G_{\mathscr{C}}$ *is* one-way, *if it contains exactly one edge from* E_r.

Note that no cycle in $G_{\mathscr{C}}$ can contain only the edges from one of E_l or E_r. In Lemma 3, we use one-way cycles to give a necessary and sufficient condition for a chain to be strongly locked.

Lemma 3. *A chain* $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ *is strongly locked, iff* $G_{\mathscr{C}}$ *admits no one-way cycle that contains* v_{σ_k}.

6 EFGISM: General Interval Scheduling

In this section, we assume that the valuation function for each player is an interval, without any restriction on the starting and ending points of the intervals.

For this case, we propose an envy-free and truthful allocation that uses less than $2n$ cuts. Our algorithm for finding a proper allocation is based on the expansion process with unlocking. Generally speaking, we iteratively run $U\text{-}exp(.)$ process on the remaining players' shares. This process allocates the entire cake, or stops in an strongly locked situation. We prove some desirable properties for this situation and leverage those properties to allocate a piece of the cake to the players in the locked chain. Next, we remove the satisfied players and shrink the allocated piece (as defined in Definition 9) and solve the problem recursively for remaining players and the remaining part of the cake.

Definition 9 (shrink). *Let $\mathcal{C}$ be a cake and $I = [I_s, I_e]$ be an interval. By the term* shrinking *I, we mean removing I from $\mathcal{C}$ and glueing the pieces to the left and right of I together. More formally, every valuation interval $[\alpha_i, \beta_i]$ turns into $[f(\alpha_i), f(\beta_i)]$, where*

$$f(x) = \begin{cases} x & x < I_s \\ I_s & I_s \leq x \leq I_e \\ x - I_e + I_s & I_e < x \end{cases}$$

(see Figure 4). As a warm-up, we ignore the truthfulness property and show that the expansion process with unlocking yields an envy-free allocation with $2(n-1)$ cuts.

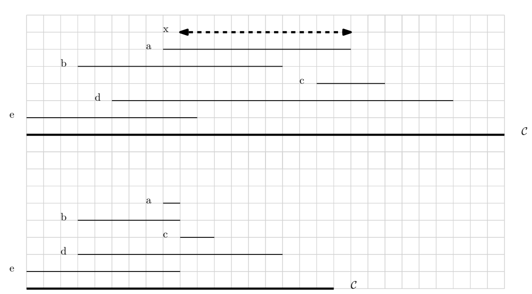

Figure 4: The cake $\mathcal{C}$ and the intervals a, b, c, d and e before and after shrinking interval x

6.1 Envy-free allocation with $2(n-1)$ cuts

For this case, our algorithm is as follows: In the beginning, we run $U\text{-}exp(\mathcal{T})$. The process either ends perfectly and the allocation is found, or a strongly locked chain appears. By the definition of strongly locked, we know that there exists a locked chain with no unlocking permutation. Let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ be a maximal strongly locked chain.

Now, consider $G_{\mathscr{C}}$. By Lemma 3, $G_{\mathscr{C}}$ contains no one-way cycle. Let ℓ be the minimum index, such that there is a directed path from v_{σ_k} to v_{σ_ℓ} using the edges in E_l.

Lemma 4. *There is a directed path from v_{σ_k} to every vertex $v_{\sigma_{\ell'}}$ with $\ell' > \ell$, using edges in E_l.*

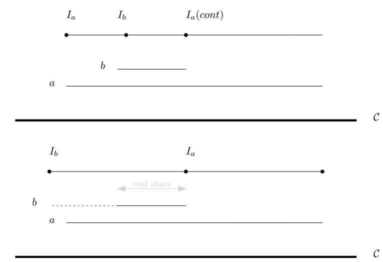

Figure 5: b can increase his share by misreporting a_b

Based on Lemma 4 and the fact that $G_{\mathscr{C}}$ contains no one-way cycle, there is no edge from $v_{\sigma_{\ell'}}$ to v_{σ_k} in E_r for any $\ell' \geq \ell$, which means:

$$\forall \ell' \geq \ell \qquad \beta_{\sigma_{\ell'}} \leq b_{\sigma_k} \tag{2}$$

On the other hand, there is no path from v_{σ_k} to $v_{\sigma_{\ell'}}$ for $\ell' < \ell$, that is:

$$\forall \ell' \geq \ell \qquad \alpha_{\sigma_{\ell'}} > a_{\sigma_{\ell-1}} \tag{3}$$

We now allocate every interval $I_{\sigma_{\ell'}}$ to $p_{\sigma_{\ell'}}$ for $\ell \leq \ell' \leq k$, remove $\{p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \ldots, p_{\sigma_k}\}$ from $\mathcal{N}$, and shrink the interval $[a_{\sigma_\ell}, b_{\sigma_k}]$. Next, we continue the expansion process with the remaining players and cake. The iteration between expansion process with unlocking and allocating the cake in the strongly locked situation goes on, until the entire cake is allocated.

Theorem 3. *The algorithm described above is envy-free, and cuts the cake in at most $2(n-1)$ locations.*

6.2 EFGISM Method

It is worth mentioning that the allocation described in section 6.1 is not truthful. Consider the example in Figure 5. It can be observed that player b can increase his share by misreporting α_b. In this section, we try to resolve this issue. Our strategy to deal with this difficulty is to run $U\text{-}exp(.)$ only for a special subset of players in every step. Lemma 5 plays the key role in our method.

Lemma 5. *Let T be a set of intervals, with the property that for every $T' \subset T$, $\Phi(T') > \Phi(T)$ (we call such set as irreducible). Then we can divide $\mathsf{DOM}(T)$ into at most $2|T|-1$ pieces and associate them to the intervals, such that:($\mathbb{I}$) the total length of the pieces associated with any interval is exactly $\Phi(T)$, ($\mathbb{II}$) the pieces allocated to any interval is totally within the interval.*

Proof. We use induction on $|T|$. For $|T| = 1$ the claim trivially holds: we can associate $\mathsf{DOM}(T)$ to the interval in T that needs no cut. Suppose that the proposition is true for $|T| < k$. We prove it for $|T| = k$. Consider $U\text{-}exp(T)$. If $U\text{-}exp(T)$ ends perfectly, then we are done. Otherwise, let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ be a maximal strongly locked chain after the process. Considering $G_{\mathscr{C}}$, let ℓ be the minimum index, such that there is a directed path from v_{σ_k} to v_{σ_ℓ}.

Lemma 6. $\ell > 1$.

By Lemma 4, we know that equations (2) and (3) are held for the chain $\mathscr{C}$. Now, let

$$x = \beta_{\sigma_k} - (k - \ell + 1) \cdot \Phi(T). \tag{4}$$

Lemma 7. $a_{\sigma_{\ell-1}} < x < a_{\sigma_\ell}$.

We show that the piece $[x, \beta_{\sigma_k}]$ can be allocated to players $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}$ using $2(k - \ell + 1) - 2$ cuts. For this, consider the valuation intervals $T' = \mathcal{I}'_{\sigma_\ell}, \mathcal{I}'_{\sigma_{\ell+1}}, \dots, \mathcal{I}'_{\sigma_k}$ such that:

$$\forall_{\ell \le i \le k} \qquad \mathcal{I}'_{\sigma_i} = (\max(x, \alpha_{\sigma_i}), \beta_{\sigma_i})$$

Note that $\mathsf{DOM}(T') = [x, \beta_{\sigma_k}]$ and hence,

$$\Phi(T') = \frac{\beta_{\sigma_k} - x}{k - \ell + 1} = \frac{b_{\sigma_k} - x}{k - \ell + 1} \tag{5}$$

Regarding Equation (4), $\Phi(T') = \Phi(T)$.

Lemma 8. *For all $T'' \subset T'$, we have $\Phi(T'') > \Phi(T')$.*

Lemma 8 shows that the set of intervals in T' admit the properties described in Lemma 5. Furthermore, regarding Lemma 6, T' is a proper subset of T. By induction hypothesis, we know that we can cut $\mathsf{DOM}(T')$ into at most $2(k - \ell + 1) - 2$ pieces and assign them to players $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}$ such that both of the properties in Lemma 5 are satisfied. Denote by $\mathcal{N}_T$ the players with valuations in T. We shrink $\mathsf{DOM}(T')$ and remove the players $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}$ from $\mathcal{N}_T$. Lemma 9 assures that the conditions in Lemma 5 are held for the remaining cake and remaining players.

Lemma 9. *Let T'' be the intervals related to the players in $\mathcal{N}_{T''} = \mathcal{N}_T \setminus \{p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}\}$ after shrinking $\mathsf{DOM}(T')$. Then, T'' is irreducible with $\Phi(T'') = \Phi(T')$.*

According to Lemma 9, we can use induction hypothesis to show that the set T'' can be allocated to the players in $\mathcal{N}_{T''}$ with $2(\ell - 1) - 2$ cuts. The total number of cuts would be $2(\ell - 1) - 2 + 2(k - \ell + 1) - 2 = 2k - 4$ cuts plus two cuts on x and β_{σ_k} that results in $2k - 2$ cuts. ☐

Based on lemma 5, we introduce the algorithm EFGISM as follows: among all subsets of $\mathcal{N}$, we find a subset such that their corresponding intervals has the minimum density (and the set with minimum size, if there were multiple options). Let N be this subset and let T be the intervals corresponding to the players in N. In Lemma 10, we show that T (and consequently N) can be found in polynomial time.

Lemma 10. *T can be found in polynomial time.*

Since T has the minimum possible density, T is irreducible. Hence, we can allocate to every player in N, a piece from $\mathsf{DOM}(T)$ with the properties defined in Lemma 5. Next, we remove the players in N from $\mathcal{N}$ and shrink $\mathsf{DOM}(T)$ from $\mathcal{C}$. Now, we recursively assign the remaining piece of the cake to remaining players using EFGISM. In Algorithm 2 you can find a psudocode for EFGISM.

Algorithm 2 EFGISM algorithm

```
function EFGISM(𝒩, 𝒯, 𝒞)
    if 𝒞 ≠ ∅ then
        T = arg min_{T'⊆𝒯} Φ(T')
        N = players with interval in T
        Allocate(N, DOM(T))                 ▷ By Lemma 5
        Shrink(𝒞, DOM(T))                   ▷ 𝒯 is also updated
        EFGISM(𝒩 \ N, 𝒯, 𝒞)
```

Theorem 4. EFGISM *is envy-free and truthful and uses at most $2(n-1)$ cuts.*

We credit the proof for truthfulness of EFGISM to (Chen et al. 2013).

7 Piecewise Constant functions

In this section, we consider a more general case in which the valuation functions of the players are piecewise constant. Denote by m the maximum number of intervals that every valuation function can have, that is, for every player p_i, $|S_i| \le m$. Here, we assume that for every p_i, $|S_i| = m$. This is without loss of generality, since we can break an interval into several sub-intervals. Thus, for every player p_i, we suppose $S_i = \{\mathcal{I}_{i,1}, \mathcal{I}_{i,2}, \dots, \mathcal{I}_{i,m}\}$.

This section consists of two parts. In the first part, we show that for a constant number of players, one can find the envy-free allocation with $n-1$ cuts in time $\mathsf{poly}(m)$. Next, in the second part, we utilize the expansion process with unlocking to devise a $\mathsf{poly}(n, m)$ time, envy-free algorithm with $O(nm)$ cuts on the cake.

Recall that finding an envy-free allocation with $n-1$ cuts for n players is PPAD$-$complete even for the case of $n = 3$ (Deng, Qi, and Saberi 2012). In Theorem 5, we show that for a constant number of players with piecewise constant valuation, the problem can be solved in time $\mathsf{poly}(m)$.

Theorem 5. *An envy-free allocation with $n-1$ cuts can be found for a constant number of players whose valuation functions are piecewise constant with m steps in time* $\mathsf{poly}(m)$.

Proof. Firstly, note that from ((Stromquist 1980)) we know there exists an envy-free allocation with $n-1$ cuts. In such an allocation there are $n-1$ cutting points. Let $0 \le c_1 \le c_2 \le \dots \le c_{n-1} \le 1$ be those cutting points and $c_0 = 0$, $c_n = 1$ be the start and end of the cake. In addition, for each player, their valuation function can be described by $2m$ constant points (2 constant points for each step) and m constant values which are the density value of each step. Therefore, there are at most $2mn$ constant points on the cake in a way that each player likes the cake between two consecutive constant points uniformly. In other words, the density value of the cake between two consecutive constant points is a constant value, for each of the players.

Now, if we know the range of each cutting point (it can be between which of the two consecutive constant points) then we can write the value of the ith piece created by cutting points ($[c_{i-1}, c_i]$) for each player j as a linear function of the cutting points. However, in order to satisfy the envy-freeness we also need to know how the pieces will be allocated to the players. If we know all of these informations then we can formulate the problem as a linear program ($n(n-1)$ constraints for envy-freeness, $n-1$ constraints guarantees $0 \leq c_1 \leq c_2 \leq \cdots \leq c_{n-1} \leq 1$, and other constraints fix the range of the cutting points). Any feasible solution of the linear program is an envy-free allocation with $n-1$ cuts.

If we can't find a feasible solution for one linear program then we need to check the next possibility of the range of the cutting points and allocation of the pieces. In the worst case, we need to check every possibility which means that we need to solve $\frac{n\times(2mn+n-1)!}{(2mn)!} = O(m^n)$ different linear programs. Finally, we know that such an allocation exists and one of the linear programs finds a feasible solution. Hence, for constant n, by solving polynomial number of different linear programs, we can find an envy-free allocation. $\square$

In the second part, we exploit expansion method with unlocking to find a proper allocation. Here, we assume that the valuation functions have a special property, namely, *intersection property*. Denote by $R_{i,j,k}$ the set of intervals in S_k that have a non-empty intersection with $\mathcal{I}_{i,j}$. We suppose that for every valuation interval $\mathcal{I}_{i,j}$ and every player $p_k (k \neq i)$, $|R_{i,j,k}| = 1$. For this case, we prove Theorem 6.

Theorem 6. *Let $\mathcal{N}$ be a set of players whose valuation functions are piecewise constant with m steps. Assuming that the* intersection property *holds, there exists a* $\mathsf{poly}(m,n)$ *time allocation algorithm that is envy-free and cuts the cake in $O(nm)$ locations.*

Proof. Consider an instance of the problem with nm players, where the valuation function of player $p_{i,j}$ is $\mathcal{I}_{i,j}$. Now, we execute EFGISM for this instance. By the properties of EFGISM, we know that the resulting allocation is envy-free and cuts the cake in at-most $2(nm-1)$ places. Let $P_{i,j}$ be the set of intervals allocated to $p_{i,j}$ in EFGISM. We show that the allocation that allocates $P_i = \bigcup_{1\leq j\leq m} P_{i,j}$ to player p_i is also envy-free.

To prove envy-freeness, we use an structural property of the expansion process: by the first invariant of the expansion process, the final allocation would allocate to every player $p_{i,j}$ a set of pieces that are totally within $\mathcal{I}_{i,j}$. In addition, note that for interval $\mathcal{I}_{i,j}$, $|R_{i,j,k}| = 1$ for every player p_k. We have $V_i(P_i) = \sum_{1\leq j\leq m} V_i(P_{i,j})$ and $V_i(P_k) = \sum_{1\leq j\leq m} V_i(P_{k,j})$. Furthermore, by intersection property, at most one valuation interval of p_k, say $\mathcal{I}_{k,l}$ has a non-empty intersection with $\mathcal{I}_{i,j}$. By the envy-freeness of EFGISM, we know that $p_{i,j}$ prefers his share to the share allocated to $p_{k,l}$, That is $V_{i,j}(P_{i,j}) \geq V_{i,j}(P_{k,l})$. Regarding the fact that $\mathcal{I}_{i,j} \cap \mathcal{I}_{k,l'} = \emptyset$ for all $l' \neq l$, we have $V_{i,j}(P_{i,j}) \geq \sum_l V_{i,j}(P_{k,l})$. Thus,

$$\sum_j V_{i,j}(P_{i,j}) \geq \sum_j \sum_l V_{i,j}(P_{k,l})$$

$$V_i(P_i) \geq \sum_j \sum_l V_{i,j}(P_{k,l}).$$

The right hand side of above equation is at least $V_i(P_k)$. $\square$

References

Aziz, H., and Mackenzie, S. 2016. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, 454–464. ACM.

Aziz, H., and Ye, C. 2014. Cake cutting algorithms for piecewise constant and piecewise uniform valuations. In *International Conference on Web and Internet Economics*, 1–14. Springer.

Barbanel, J. B., and Brams, S. J. 2004. Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond. *Mathematical Social Sciences* 48(3):251–269.

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

Brams, S. J., and Taylor, A. D. 1995. An envy-free cake division protocol. *American Mathematical Monthly* 9–18.

Brams, S. J.; Jones, M. A.; Klamler, C.; et al. 2006. Better ways to cut a cake. *Notices of the AMS* 53(11):1314–1321.

Brams, S. J.; Feldman, M.; Lai, J. K.; Morgenstern, J.; and Procaccia, A. D. 2012. On maxsum fair cake divisions. In *AAAI*.

Brânzei, S.; Caragiannis, I.; Kurokawa, D.; and Procaccia, A. D. 2016. An algorithmic framework for strategic fair division. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Caragiannis, I.; Lai, J. K.; and Procaccia, A. D. 2011. Towards more expressive cake cutting. In *IJCAI*.

Chen, Y.; Lai, J. K.; Parkes, D. C.; and Procaccia, A. D. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior* 77(1):284–297.

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

Kurokawa, D.; Lai, J. K.; and Procaccia, A. D. 2013. How to cut a cake before the party ends. In *AAAI*.

Maya, A., and Nisan, N. 2012. Incentive compatible two player cake cutting. In *International Workshop on Internet and Network Economics*, 170–183. Springer.

Procaccia, A. D. 2013. Cake cutting: not just child's play. *Communications of the ACM* 56(7):78–87.

Procaccia, A. D. 2014. Cake cutting algorithms.

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

Stromquist, W. 1980. How to cut a cake fairly. *American Mathematical Monthly* 640–644.

Stromquist, W. 2007. Envy-free cake divisions cannot be found by finite protocols. In *Fair Division*.