Noname manuscript No.
(will be inserted by the editor)

Expand the Shares Together: Envy-free Mechanisms with a Small Number of Cuts ⋆

Masoud Seddighin · Majid Farhadi · Mohammad Ghodsi · Reza Alijani · Ahmad S. Tajik

Received: date / Accepted: date

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 to meet these conditions: ($\mathbb{I}$) every player (weakly) prefers his allocated cake to any other player's share (such notion is known as envy-freeness), ($\mathbb{II}$) the allocation is dominant strategy-proof (truthful) ($\mathbb{III}$) the number of cuts made on the cake is small. We provide methods for dividing the cake under different assumptions on the valuation functions of the players.

First, we suppose that the valuation function of every player is a single interval with a special property, namely *ordering property*. For this case, we propose a process called *expansion process* and show that it results in an envy-free and truthful allocation that cuts the cake into exactly n pieces.

Next, we remove the ordering restriction and show that for this case, an extended form of the expansion process, called *expansion process with unlocking* yields an envy-free allocation of the cake with at most $2(n-1)$ cuts. Furthermore, we show that in the *expansion process with unlocking*, the players may misrepresent their valuations to earn more share. In addition, we use a more complex form of the *expansion process with unlocking* to obtain

⋆ A preliminary version of this paper is appeared in AAAI 2017 [1].

Masoud Seddighin
Sarif University of Technology, E-mail: mseddighin@ce.sharif.edu

Majid Farhadi
Sharif University of Technology, E-mail: majid.farhadi94@gmail.com

Mohammad Ghodsi
Sharif University of Technology,
IPM - Institute for Research in Fundamental Sciences, E-mail: ghodsi@sharif.edu

Reza Alijani
Duke university, E-mail: rezaalijani71@gmail.com

Ahmad S. Tajik
University of Michigan - Ann Arbor, E-mail: ahmadshahab.t@gmail.com

an envy-free and truthful allocation that cuts the cake in at most $2(n-1)$ locations.

We also, evaluate our expansion method in practice. In the empirical results, we compare the number of cuts made by our method to the number of cuts in the optimal solution $(n-1)$. The experiments reveal that the number of cuts made by the expansion and unlocking process for envy-free division of the cake is very close to the optimal solution.

Finally, we study piecewise constant and piecewise uniform valuation functions with m pieces and present the conditions, under which a generalized form of expansion process can allocate the cake via $O(nm)$ cuts.

Keywords cake cutting, envy-free, mechanism design, approximation, fairness

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 [17]. 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 [16]). One of the most important notions is *envy-freeness*. An allocation of the cake is envy-free, if each player (weakly) prefers his allocated share to any other players' 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 [15] for details). Brams and Taylor generalized this method to an arbitrary number of players [9]. However, their method doesn't guarantee any upper bound on the number of cuts. Recently, Aziz and Mackenzie in [2], suggested a bounded envy-free protocol for any number of agents.

In some settings, the number of cuts is important. In several papers (e.g. [18], [4], [19], [5]) the cake cutting with minimum number of cuts has been studied. In some cases, each cut might have an additional cost. As an example, suppose that the cake models a processing time that must be fairly allocated among a set of tasks. Every task-switch imposes an overhead; minimizing the 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 [10], 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 issue is unlikely.

Stromquist, in [18], 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. He also proved in [19] that no finite protocol can find an envy-free allocation with the minimum number of cuts for $n \geq 3$. Deng, Qi, and Saberi in [12] showed 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 recent papers (e.g. [10, 7, 5, 14, 11, 3]) some restricted classes of valuation functions have been studied. Piecewise constant and piecewise uniform valuation functions are two 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 in [13] 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.

The classic cake cutting algorithms assume that the agents are not strategic and honestly report their valuations. However, this is not the case in many real life situations. Recently, some studies considered the problem from a game theoretic perspective and attempt to find truthful mechanisms for dividing the cake. Similar to fairness, there are different notions for the concept of truthfulness. Brams et. al. in [8], observed a weak notion of truthfulness: 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 [14] 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 [11], 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 approach for dividing the cake uses $\Omega(n^2 m)$ cuts, where m is the number of pieces in each valuation function. Aziz and Ye [3] 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 [11], in the case of piecewise uniform valuations.

[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

1.1 Our work

We investigate the problem of finding envy-free and truthful mechanisms with a small number of cuts. By truthful, we mean that the players must not gain from misreporting their valuation, regardless of the action of other players. By a small number of cuts, we mean that the number of cuts does not exceed $O(nm)$, where m is the complexity (i.e., number of steps) of each players' (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 methods 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 maintains 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 1).

Next, we combine the expansion process with another process called *unlocking*. Based on the expansion process with unlocking, we propose EFSC, which is an envy-free allocation that cuts the cake into at most $2n-1$ pieces in polynomial time (Theorem 2). To the best of our knowledge, no previous work tries to approximate the number of cuts in a fair solution. Furthermore, using a more complex form of the expansion process with unlocking, we propose EFGISM, which is a recursive, polynomial time, truthful, and envy-free algorithm that cuts the cake into at most $2n-1$ pieces (Theorem 3).

We evaluate the expansion with unlocking method in practice. We compare the number of cuts made by our method to the number of cuts in the optimal solution $(n-1)$. Interestingly, the number of cuts returned by our algorithm is very close to $n-1$, which shows that the algorithm is almost optimal (see Section 8 for more details).

Finally, we consider the case where the valuation functions are piecewise constant with m pieces. When the number of players is constant, we provide a $\mathsf{poly}(m)$ 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 $\mathsf{poly}(m,n)$ time, envy-free algorithm that cuts the cake in $O(nm)$ locations.

2 Model Description and Preliminaries

Throughout the 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 valuations functions 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 $S_\nu = \{\mathcal{I}_{\nu 1}, \mathcal{I}_{\nu 2}, \ldots, \mathcal{I}_{\nu k}\}$ 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 function 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 $D = \{P_1, P_2, \ldots, P_n\}$ of pieces, with each piece $P_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,|P_i|}\}$ 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 $D = \{P_1, P_2, \ldots, P_n\}$ 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., $\mathcal{T} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$. 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 exists an interval I in X such that $x \in I$. For example, in Fig 1, the set T is solid. For a solid set T, 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 Section 8, we demonstrate that slightly modified versions of our algorithms can handle the situations where this assumption does not hold.

We end this section with a simple observation.

Observation 21 *If* $\frac{a+b}{c+d} = \frac{e}{f}$ *holds for positive real numbers* a, b, c, d, e *and* f*, we have:*

$$\frac{a}{c} \leq \frac{e}{f} \iff \frac{b}{d} \geq \frac{e}{f}$$

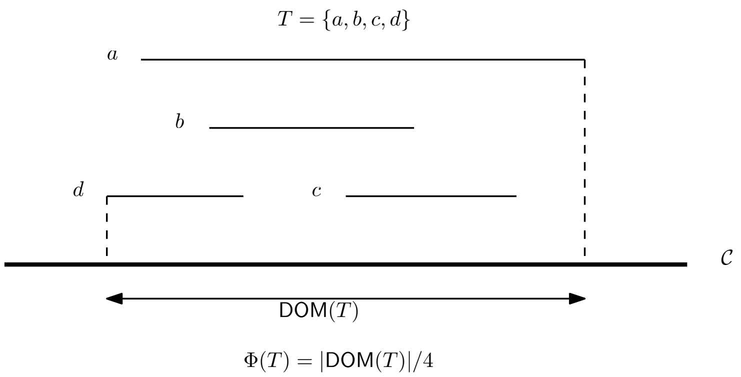


Fig. 1: Domain and density

According to Observation 21, If T is not solid, then there exists a subset $T' \subset T$ which is solid and $\Phi(T') \leq \Phi(T)$. We use Observation 21 in the proof of Lemmas 8 and 9.

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 (i.e., valuation intervals). We use $exp(T)$ to refer to the expansion process on 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 Intervals. We expand the intervals in S concurrently, all from the endpoint. The expansion is performed in a way that maintains two invariants:($\mathbb{I}$) The expansion has the same speed for all the intervals so as the lengths of the intervals remain 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. 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 $I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$, 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.

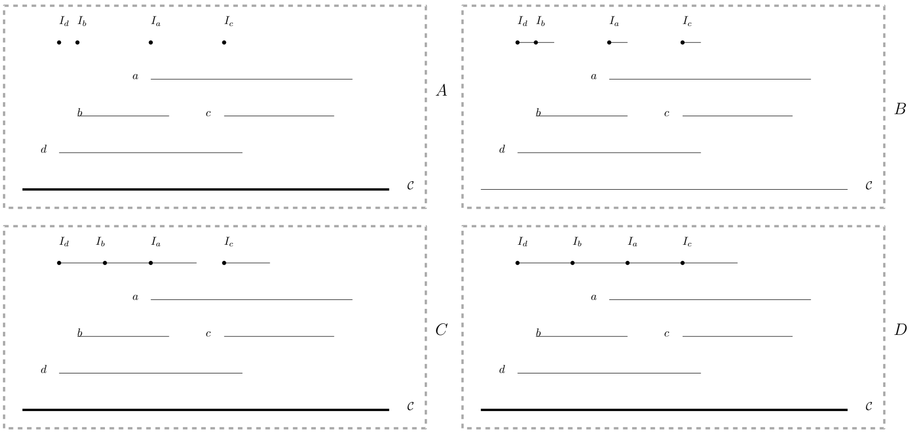


Fig. 2: An example of *expansion process*: (A): the intervals are single points, (B): I_d starts pushing I_b, (C): I_b starts pushing I_a, and (D): I_b becomes locked and the expansion process terminates

The expansion ends when an interval becomes locked. The termination condition ensures that the second invariant is always preserved. In Figure 2, you can see a detailed example of the expansion process.

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 $|I_i| = \Phi(T)$.

Observation 31 *During the expansion process, every interval I_i is either being pushed by another interval, or its starting point is still on α_i.*

3.1 Executing Expansion Process in Polytime

Despite the fact that we described the expansion process continuously, it can be efficiently implemented via sweeping the events. As said, the expansion process starts with allocating zero-length interval $[a_i = \alpha_i, b_i = \alpha_i]$ to p_i. We expand the shares until one of these events occur:

1. A chain becomes locked.
2. Two consecutive chains $\mathscr{C}_i$ and $\mathscr{C}_{i+1}$ get merged, i.e., the end-point of the last share interval in $\mathscr{C}_i$ gets stuck in starting point of the first share interval of $\mathscr{C}_{i+1}$.

If the first event occurs, we terminate the process. For the second event, we merge the chains $\mathscr{C}_i$ and $\mathscr{C}_{i+1}$ and continue to expand the shares. To expand the shares after each merging event, we should know the maximum length L that we can expand the shares, before the next event occures. Let $\mathscr{C}_1, \ldots, \mathscr{C}_m$ be the maximal chains in the current state. For each share interval I_i, let c_i be its zero-based index in the corresponding maximal chain, i.e., number of intervals that are pushing I_i. Simultaneous expansion of all the intervals by length l simply shifts each interval I_i by $c_i \times l$, if no chain gets locked nor two maximal chains merge. We can find maximum l by considering all possible events.

$$L = \min(\min_{i \leq n}\{\frac{\beta_i - b_i}{c_i + 1}\}, \min_{j<m}\{\frac{\text{gap}(\mathscr{C}_j, \mathscr{C}_{j+1})}{|\mathscr{C}_j|}\}) \tag{1}$$

where $\text{gap}(\mathscr{C}_j, \mathscr{C}_{j+1})$ denotes the size of currently unallocated piece of the cake between chains $\mathscr{C}_j$ and $\mathscr{C}_{j+1}$ and $|\mathscr{C}_j|$ denotes the number of intervals in $\mathscr{C}_j$. After expansion of the shares by size L, the process either terminates by a locked chain, or a pair of chains get merged. Considering the fact that the merged chains will remain together during the expansion process, number of merge events is bounded by $m - 1 < n$. Thus, the expansion process can be simulated in $\mathsf{poly}(n)$.

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:

$$\forall i, j \qquad \alpha_i \leq \alpha_j \iff \beta_i \leq \beta_j. \tag{2}$$

In other words, no interval is a sub-interval of another (unless they start or end in the same place). For example, when all the valuation intervals have the same length, Equation (2) is held.

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 of 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 $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ be a locked chain with maximal size in S. Since $\mathscr{C}$ is maximal, no interval is pushing I_{σ_1}. By Observation 31, 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 $\mathsf{DOM}(\mathscr{T}) = [\alpha_{\sigma_1}, \beta_{\sigma_k}]$.

Proof. By the structure of expansion procedure, we know $a_{\sigma_1} \leq a_{\sigma_2} \leq \ldots \leq a_{\sigma_k}$. Furthermore, by Equation (2) we have $b_{\sigma_1} \leq b_{\sigma_2} \leq \ldots \leq b_{\sigma_k}$. By the fact

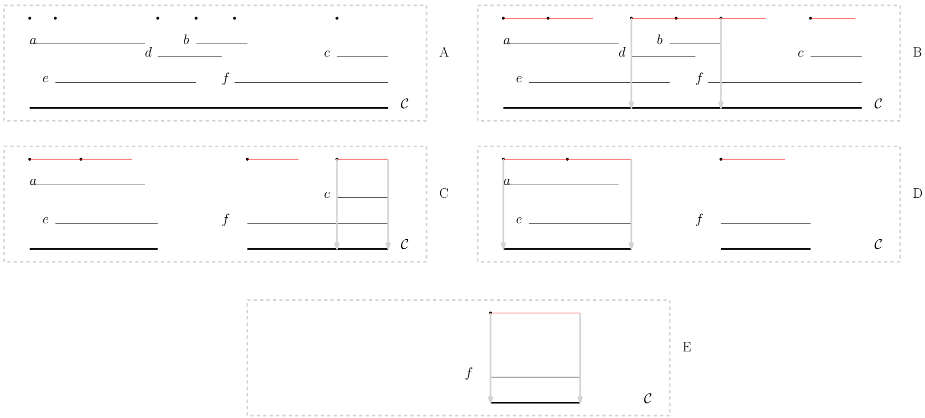


Fig. 3: An example of expansion and removing process: (A): Expansion starts with single point intervals (B): I_b is locked and the maximal chain I_d, I_b gets removed (C): Expansion continues until I_c gets locked and removed (D): I_a, I_e gets locked and removed (E): I_f is locked and removed. End of process.

that I_{σ_k} is locked and regarding the definition of chain, we have $a_{\sigma_1} = \alpha_{\sigma_1}$ and $b_{\sigma_k} = \beta_{\sigma_k}$. Therefore,

$$\mathsf{DOM}(\mathscr{C}) = [\min_{1 \leq j \leq k} a_{\sigma_j}, \max_{1 \leq l \leq k} b_{\sigma_l}] = [\alpha_{\sigma_1}, \beta_{\sigma_k}].$$

□

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

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

Proof. By the second invariant of the expansion process, we know that I_{σ_i} is entirely within $\mathcal{I}_{\sigma_i}$. Other intervals in $\mathscr{C}$ have the same length as I_{σ_i} and hence, their value for player p_{σ_i} can not be more than I_{σ_i}.

□

Next, we remove players $p_{\sigma_1}, p_{\sigma_2}, \ldots, p_{\sigma_k}$ 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 $\mathcal{C}_r$ and $\mathcal{C}_l$. Let $\mathcal{N}_l$ and $\mathcal{N}_r$ be the set of players with their share inside $\mathcal{C}_l$ and $\mathcal{C}_r$. 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 part of the cake along with its corresponding players and updating the valuations, we perform this expansion and removal

Algorithm 1 EFISM algorithm

```
function EFISM(C = [a, b], N, T)
    if C ≠ ∅ then
        exp(T)                                        ▷ Expansion process on T
        Let 𝒞 = I_σ1, I_σ2, ..., I_σk be a maximal locked chain
        for 1 ≤ i ≤ k do
            Allocate I_σi to p_σi
        C_l = [a, α_σ1]
        C_r = [β_σk, b]
        for every p_k ∈ N do
            if a_k < a_σ1 then
                β_k = min(β_k, α_σ1)
                Add p_k to N_l
                Add I_k to T_l
            else if b_k > b_σk then
                α_k = max(α_k, β_σk)
                Add p_k to N_r
                Add I_k to T_r
        EFISM(C_l, N_l, T_l)
        EFISM(C_r, N_r, T_r)
```

separately 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 pseudocode for EFISM. In addition, In Figure 3 you can find a detailed example of EFISM.

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

Proof. Envy-freeness: We prove by induction on the number of players, n. For $n = 1$, the claim holds trivially. For $n > 1$, consider the first expansion process. If the process was perfect, by Lemma 2, the allocation is envy-free. Otherwise, If the process was imperfect, let $\mathscr{C}$ be a maximal locked chain and let $\mathcal{N}_{\mathscr{C}}$ be the players corresponding to the intervals in $\mathscr{C}$. By Lemma 2, none of the players in $\mathcal{N}_{\mathscr{C}}$ envy each other. Furthermore, since the entire $\mathsf{DOM}(\mathscr{C})$ is allocated, the players in $\mathcal{N}_{\mathscr{C}}$ do not envy the players in $\mathcal{N}_l$ and $\mathcal{N}_r$. By induction hypothesis, the allocation of $\mathcal{C}_r$ and $\mathcal{C}_l$ is envy free for $\mathcal{N}_r$ and $\mathcal{N}_l$, respectively. Since $\mathcal{C}_l$ ($\mathcal{C}_r$) has no value for the players in $\mathcal{N}_r$ ($\mathcal{N}_l$), the players in $\mathcal{N}_l$ and $\mathcal{N}_r$ do not envy each other.

Finally, regarding the facts that the expansion process for $\mathcal{T}_l$ and $\mathcal{T}_r$ expands the associated intervals at least to the length of intervals in $\mathscr{C}$, the players in $\mathcal{N}_l \cup \mathcal{N}_r$ do not envy the players in $\mathcal{N}_{\mathscr{C}}$.

Number of cuts: During the algorithm, one piece of the cake is associated to each player, which means that the total number of cuts on the cake is $n-1$. Note that no piece of $\mathcal{C}$ is left behind.

Truthfulness: For an arbitrary player p_i whose true valuation interval is $\mathcal{I}_i$, we show his utility cannot be increased by lying. However, first of all, we must clearly determine what do we mean by "lying"?.

Note that the EFISM algorithm is proposed for the case that the valuation intervals satisfy the *ordering property*, i.e., satisfy Equation (2). Regarding this,

p_i can change his valuation in a way that this property no longer is satisfied. Here, we show that p_i can not earn more payoff even by misreporting his valuation in a way that the *ordering property* no longer satisfies. Note that this type of misreporting can harm other properties of the algorithm; For example, Figure 4 shows an example that removing the *ordering property* results in an allocation that is no longer envy-free.

Let L be the length of allocated share to p_i in case he tells the truth and let $\mathscr{C}$ be the locked chain that includes I_i. Considering the expansion process as a continuous process that the shares grow with the same speed, let t be the (relative) time, when $\mathscr{C}$ becomes locked. Furthermore, let $\mathscr{T}$ be the set of valuation intervals related to the share intervals in $\mathscr{C}$. According to the structure of the expansion process, $\Phi(\mathscr{T}) = L$, and the size of every share interval in $\mathscr{C}$ is exactly L.

Let α'_i, β'_i be the starting point and the ending points of the interval that p_i states. We want to show that no matter what the value of α'_i and β'_i are, there would be no scenario in which p_i achieves more than L from $\mathsf{DOM}(\mathscr{T})$.

First, note that for every subset $\mathscr{T}'$ of $\mathscr{T}$, $\Phi(\mathscr{T}') \geq L$, which means if p_i tells the truth, no interval in $\mathscr{T}$ becomes locked during the expansion, before time t. Thus, if p_i lies in a way that a locked chain $\mathscr{C}'$ forms before time t, I_i would be one of the intervals in $\mathscr{C}'$. In other words, p_i can not force other intervals in $\mathscr{T}$ to lock earlier, without participating in the locked chain. On the other hand, if p_i satisfies in any time earlier than t, his share would be less than L.

Hence, the share of all the players in $\mathscr{C}$ remains at least L, even if p_i lies. Moreover, $|\mathsf{DOM}(\mathscr{T})| = L \times |\mathscr{T}|$ which means that p_i can not gain an interval with size more than $L \times |\mathscr{T}| - L \times (|\mathscr{T}| - 1) = L$ from $\mathsf{DOM}(\mathscr{T})$, independent of what he submits as his valuation interval. Regarding the fact that if he reports truthfully $|I_i| = L$, he can not gain more payoff by lying.

□

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 4. Obviously, 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. To overcome these issues, in the next section we introduce a more general form of the expansion process

5 Expansion Process with Unlocking

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

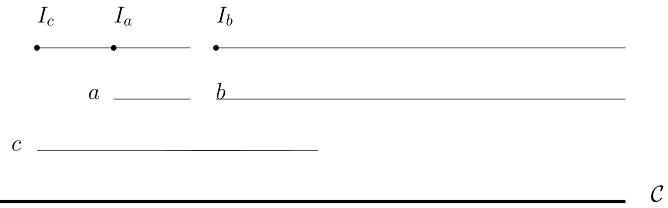


Fig. 4: EFISM for intervals *without* ordering property

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:

($\mathbb{I}$) : All the intervals of the permutation are members of the locked chain, i.e., $\forall_i, I_{\delta_i} \in \mathscr{C}$, and the last interval of the permutation is the locked interval, i.e., $\delta_r = \sigma_k$.

($\mathbb{II}$) : For every $j < r$, the share associated to player p_{δ_j} is totally within the valuation interval of player $p_{\delta_{j+1}}$ (with its endpoint strictly less than the endpoint of the valuation interval), i.e., $\forall_{1 \leq j \leq r-1}$, $a_{\delta_j} \geq \alpha_{\delta_{j+1}}$ and $b_{\delta_j} < \beta_{\delta_{j+1}}$.

($\mathbb{III}$) : The share associated to player p_{δ_r} is within the valuation interval of player p_{δ_1} (with its endpoint strictly less than the endpoint of the valuation interval), i.e., $\alpha_{\delta_1} \leq a_{\delta_r}$ and $\beta_{\delta_1} > b_{\delta_r}$.

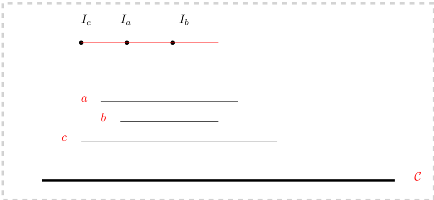
The intuition behind the definition of unlocking permutation is as follows: let $I_{\delta_1}, I_{\delta_2}, \ldots, I_{\delta_r}$ be a $\mathscr{C}$-unlocking permutation, where $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$. Then, we can change the order of the intervals in $\mathscr{C}$ 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 process and the expansion can be continued.

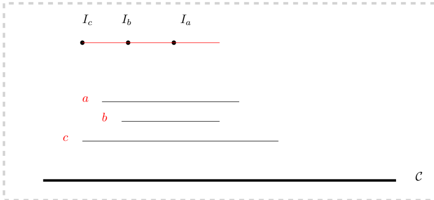
It is worthwhile to mention that there may be multiple locked intervals in a moment. In such case, we separately try to unlock each interval. For a set $\mathcal{T}$ of valuation intervals, we use U-$exp(\mathcal{T})$ to refer to the expansion process with unlocking for $\mathcal{T}$. See Figure 5 for an example of this process.

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.

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

Definition 7 A permutation graph for a locked chain $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ 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}, edge $(v_{\sigma_i}, v_{\sigma_j})$ is in E_r, if $i < j$ and $\beta_{\sigma_i} > b_{\sigma_j}$.





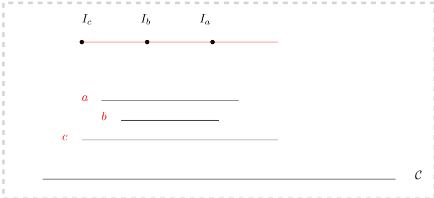


Fig. 5: An example of Expansion process with Unlocking

See Figure 6 for an example of permutation graph. A trivial necessary and sufficient condition for a chain $\mathscr{C}$ to be strongly locked is that $G_{\mathscr{C}}$ contains no cycle including v_{σ_k}. However, regarding the special structure of $G_{\mathscr{C}}$, we can define a more restricted necessary and sufficient condition for a strongly locked situation.

Definition 8 A directed circle 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, if and only if $G_{\mathscr{C}}$ admits no one-way cycle containing v_{σ_k}.*

Proof. To prove the non-trivial direction it suffices to show the existence of a one-way cycle for any non-strongly locked chain $\mathscr{C}$. Let $C = v_{\delta_1=\sigma_k}, v_{\delta_2}, \ldots, v_{\delta_r=\sigma_k}$ be the shortest cycle in $G_{\mathscr{C}}$ that includes v_{σ_k}. We claim that C contains exactly one edge from E_r (that is the edge $(v_{\delta_{r-1}}, v_{\sigma_k})$). As a contradiction, let $(v_{\delta_i}, v_{\delta_{i+1}})$ be the first edge in E_r that appears in C. By definition of E_r, we have $\delta_{i+1} > \delta_i$. Furthermore, by definition of E_l, we have:

$$\delta_1 > \delta_2 > \ldots > \delta_i < \delta_{i+1}.$$

Hence, $\delta_j > \delta_{i+1} > \delta_{j+1}$ for some $1 \leq j \leq i-1$. Note that since v_{δ_j} has a left edge to $v_{\delta_{j+1}}$, it has an edge to $v_{\delta_{i+1}}$ as well (see the structure of the expansion graph in Figure 6). Thus, $C' = v_{\sigma_k=\delta_1}, v_{\delta_2}, \ldots, v_{\delta_j}, v_{\delta i+1}, \ldots, v_{\delta_r=\sigma_k}$ is a shorter cycle that includes v_{σ_k}, which is a contradiction.

□

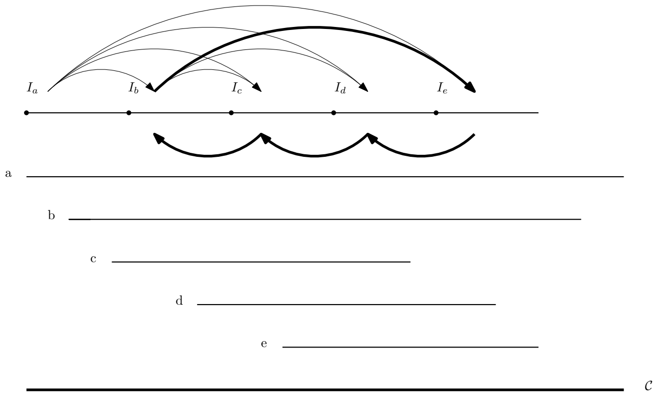


Fig. 6: An example of a permutation graph. Here the locked chain I_a, I_b, I_c, I_d, I_e can be unlocked by repermuting I_a, I_b, I_c, I_d, I_e to I_a, I_e, I_b, I_c, I_d

5.1 Expansion Process With Unlocking Can be Executed in Polytime

In this section, we extend the simulation described in 3.1 for a locked situation. Similarly, in each iteration, we expand all the shares by size L as determined in Equation (1). The difference is: when encountering a locked chain, regarding Lemma 3, we search for an unlocking one-way cycle (in polynomial time) and re-permute the shares through it. The process terminates when a chain becomes strongly locked.

As mentioned in Section 3.1, the number of expansions that join two maximal chains is bounded by n. Thus, it only remains to bound the number of alternating expansions and unlocking processes between maximal chain merging stages.

Consider an unlocking procedure, that unlocks a chain locked by a share I_{σ_i} that is positioned at c_{σ_i} (as defined in Section 3.1). We label this unlocking by the pair (σ_i, c_{σ_i}). The point is that no other unlocking in this maximal chain will be labeled by pair (σ_i, c_{σ_i}). We prove this by showing that after the unlocking process, I_{σ_i} goes to a new place and never returns back to c_{σ_i}. This is due to the fact that after the unlocking and its subsequent expansion, the endpoint of c_{σ_i} crosses β_{σ_i} and hence, I_{σ_i} can never return back to place c_{σ_i} by any unlocking re-permutation. Since no two unlocking procedures can have the same label, the total number of unlocking procedures is upper bounded by $O(n^2)$. Thus, one can simulate the Expansion process with Unlocking in polytime.

6 General Interval Scheduling

In this section, we assume that the valuation function of each player is an interval, without any restriction on the starting and ending points of the intervals. For this case, we suggest an envy-free and truthful allocation that

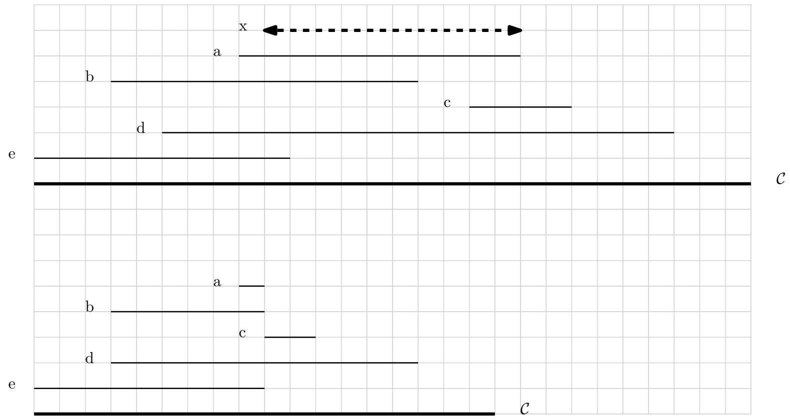


Fig. 7: The cake $\mathcal{C}$ and the intervals a, b, c, d and e before and after shrinking interval x.

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-$exp(.)$ on the remaining players' shares. This process allocates the entire cake or stops in a strongly locked situation. We prove some desirable properties for this situation and leverage these properties to allocate a piece of the cake to the players in the strongly locked chain. Next, we remove the satisfied players and shrink the allocated piece (as defined in Definition 9) and solve the problem recursively for the remaining players and the remaining part of the cake.

Definition 9 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 gluing 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 7).

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.

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

In this section, we propose EFSC, which is polynomial time envy-free algorithm with $2(n-1)$ cuts. Our algorithm is as follows: in the beginning, we run U-

$exp(\mathcal{T})$. The process either ends perfectly and the desired allocation is found, or a strongly locked chain appears. For the case that the process ends imperfectly, 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. Regarding the special structure of $G_{\mathscr{C}}$, Lemma 4 holds. This lemma is used to prove the envy-freeness of EFSC.

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

Proof. Let $P = v_{\sigma_k=\delta_1}, v_{\delta_2}, \ldots, v_{\delta_r=\sigma_\ell}$ be the path from v_{σ_k} to v_{σ_ℓ} using the edges in E_l. By definition of the left edge,

$$\sigma_k = \delta_1 > \delta_2 > \ldots > \delta_r = \sigma_\ell.$$

Note that if $\sigma_{\ell'} = \delta_j$ for some j, then we are done. Otherwise, since $\ell' > \ell$, $\delta_j < \sigma_{\ell'} < \delta_{j+1}$ for some $1 \leq j < l$. By the structure of the graph, v_{δ_j} has a left edge to $v_{\sigma_{\ell'}}$. Thus, $v_{\sigma_k=\delta_1}, v_{\delta_2}, \ldots, v_{\delta_j}, v_{\sigma_{\ell'}}$ is the desired path between v_{σ_k} and $v_{\sigma_{\ell'}}$.

□

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:

$$\forall \ell' \geq \ell \qquad \beta_{\sigma_{\ell'}} \leq b_{\sigma_k}. \tag{3}$$

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

$$\forall \ell' \geq \ell \qquad \alpha_{\sigma_{\ell'}} > a_{\sigma_{\ell-1}}. \tag{4}$$

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

Theorem 2 EFSC *is envy-free and cuts the cake in at most $2(n-1)$ locations.*

Proof. Envy-freeness: Let p_i and p_j be two arbitrary players. We want to show that p_i does not envy p_j. To show this, let r_i and r_j be the steps, that a piece of the cake is allocated p_i and p_j, respectively. There are three possibilities: $(\mathbb{I}) r_i < r_j$, $(\mathbb{II}) r_i = r_j$ and $(\mathbb{III}) r_i > r_j$. For $r_i < r_j$, regarding Inequality 4, $|I_j \cap \mathcal{I}_i|$ would be less that $|I_i|$, which means p_i does not envy p_j. If $r_i = r_j$, considering Lemma 2, p_i does not envy p_j. Finally, for the case $r_i > r_j$, by the fact that the total size of the share allocated to p_i is strictly larger than the one allocated to p_j, p_i does not envy p_j.

Number of cuts: We use induction to prove that the number cuts is at most $2(n-1)$. For $n = 1$, the expansion process allocates a single piece to the

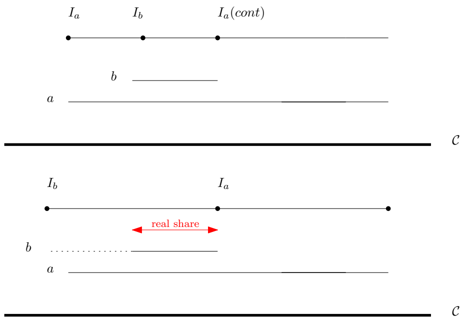


Fig. 8: Player b can increase his share by misreporting a_b

player that requires no cut. Now, suppose that the proposition is true for every $n' < n$. We prove it for n players. First, note that if the process U-$exp(.)$ ends successfully, the cake is divided via $n-1$ cuts. Thus, assume that the process is in a strongly locked situation. Let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \dots, I_{\sigma_k}$ be a strongly locked chain and let ℓ be the minimum index such that there is a path from v_{σ_k} to v_{σ_ℓ} in $G_{\mathscr{C}}$. In the algorithm we allocate $I_{\sigma_\ell}, I_{\sigma_{\ell+1}}, \dots, I_{\sigma_k}$ to players $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}$, respectively, and shrink the allocated part. Every player in $\{p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \dots, p_{\sigma_k}\}$ gets a single piece of the cake ($(k-\ell)$ cuts). Furthermore, by induction hypothesis, the remaining part of the cake requires at most $2(n-k+\ell-2)$ cuts. Considering the cuts in the beginning and the ending points of the shares related to $\mathscr{C}$, we have $2(n-k+\ell-2)+(k-\ell)+2 = 2(n-1)-k+\ell \leq 2(n-1)$ cuts.

□

6.2 EFGISM: truthful, envy-free allocation with $2(n-1)$ cuts

It is worth mentioning that EFSC is not truthful. Consider the example in Figure 8. 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-$exp(.)$ only for a special subset of players in every step. Lemma 5 constitutes the core of 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* DOM(T) *into at most $2|T|-1$ pieces and associate them to the intervals, such that:*(I) *total length of the pieces associated with any interval is exactly $\Phi(T)$.* (II) *the pieces allocated to any interval is totally within that interval.*

Proof. We use induction on $|T|$. For $|T| = 1$ the claim trivially holds: we can associate 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-$exp(T)$. If U-$exp(T)$ ends perfectly, then we are done. Otherwise, let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \dots, I_{\sigma_k}$

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} using the edges in E_l. In Lemma 6 we show ℓ is strictly greater than 1. This fact is later used to break the problem into two strictly smaller sub-problems and solve each sub-problem recursively.

Lemma 6 *Let $\mathscr{C} = I_{\sigma_1}, I_{\sigma_2}, \ldots, I_{\sigma_k}$ be a maximal strongly locked chain after running U-exp(T) and let ℓ be the minimum index, such that there is a directed path from v_{σ_k} to v_{σ_ℓ} in $G_{\mathscr{C}}$ using the edges in E_L. Then, we have $\ell > 1$.*

Proof. By contradiction, suppose $\ell = 1$. Regarding Equation (3), for all i we have $\beta_{\sigma_i} \leq b_{\sigma_k}$. Now, we show that for all i, $\alpha_{\sigma_i} \geq a_{\sigma_1}$. Let

$$x = \min_{1 \leq i \leq k} \alpha_{\sigma_i}$$

and

$$j = \arg \min_{1 \leq i \leq k} \alpha_{\sigma_i}.$$

In the beginning of the expansion process, the starting point of I_{σ_j} is on x. During the expansion, there may be situations that I_{σ_j} is replaced by another interval $I_{\sigma_{j'}}$, but by the definition of unlocking permutation, $\alpha_{\sigma_{j'}} = x$. Furthermore, the starting point of the rear interval in $\mathscr{C}$ always remains on x, because $\mathscr{C}$ is maximal and hence, no interval pushes the interval with the starting point on x. Therefore, in a strongly locked situation, $\alpha_{\sigma_1} = x$ and for everey $i > 1$, we have $\alpha_{\sigma_i} \geq x$. Thus, $\mathsf{DOM}(\mathcal{T}_{\mathscr{C}}) = [a_{\sigma_1}, b_{\sigma_k}]$, where $\mathcal{T}_{\mathscr{C}}$ is the valuation intervals corresponding to the share intervals in $\mathscr{C}$. Since, $[a_{\sigma_1}, b_{\sigma_k}]$ is covered by $\mathscr{C}$, $\Phi(\mathcal{T}_{\mathscr{C}}) = |I_{\sigma_1}| = |I_{\sigma_2}| = \cdots = |I_{\sigma_k}|$. On the other hand, since *U-exp*($\mathcal{T}$) was imperfect, $|I_{\sigma_k}| < \Phi(T)$, which contradicts the irreduciblinty of T. □

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

$$x = \beta_{\sigma_k} - (k - \ell + 1)\Phi(T). \tag{5}$$

In Lemma 7, we show that the location of x in $[0, 1]$ is some value between $a_{\sigma_{\ell-1}}$ and a_{σ_ℓ}. This fact, allows us to break $\mathsf{DOM}(T)$ into two pieces, both of which preserve the properties defined in Lemma 5.

Lemma 7 *Let $x = \beta_{\sigma_k} - (k - \ell + 1)\Phi(T)$, where ℓ is the minimum index, such that there is a directed path from v_{σ_k} to v_{σ_ℓ}. Then, we have $a_{\sigma_{\ell-1}} < x < a_{\sigma_\ell}$.*

Proof. Regarding Inequality (3) and the fact that $b_{\sigma_k} = \beta_{\sigma_k}$ (note that $\mathscr{C}$ is strongly locked) we have

$$\forall_{\ell \leq i \leq k} \qquad \beta_{\sigma_i} \leq \beta_{\sigma_k}.$$

Furthermore, by Inequality (4),

$$\forall_{\ell \leq i \leq k} \qquad \alpha_{\sigma_i} > a_{\sigma_{\ell-1}}.$$

Now, let $\mathcal{T}' = \{p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \ldots, p_{\sigma_k}\}$. We have:

$$\Phi(\mathcal{T}') = \frac{\mathsf{DOM}(\mathcal{T}')}{|\mathcal{T}'|} < \frac{\beta_{\sigma_k} - a_{\sigma_{\ell-1}}}{k - \ell + 1}.$$

On the other hand, regarding Equation (5),

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

Considering the fact that T has the minimum possible density, $x > a_{\sigma_{\ell-1}}$. Since U-$exp(T)$ was imperfect, we have: $|I_{\sigma_1}| = |I_{\sigma_2}| = \ldots = |I_{\sigma_k}| < \Phi(\mathcal{T})$. Hence,

$$\begin{aligned} x &= b_{\sigma_k} - (k - \ell + 1) \times \Phi(T) \\ &< b_{\sigma_k} - (k - \ell + 1) \times |I_{\sigma_k}| \\ &< b_{\sigma_{\ell-1}} = a_{\sigma_\ell}. \end{aligned}$$

Note that since the interval $I_{\sigma_{\ell-1}}$ is stuck in I_{σ_ℓ}, we have $b_{\sigma_{\ell-1}} = a_{\sigma_\ell}$. □

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

$$\Phi(T') = \frac{\beta_{\sigma_k} - x}{k - \ell + 1} = \frac{b_{\sigma_k} - x}{k - \ell + 1} \tag{6}$$

Regarding Inequality (5), $\Phi(T') = \Phi(T)$.

Lemma 8 *T' is irreducible, i.e., for all $T'' \subset T'$, we have $\Phi(T'') > \Phi(T')$.*

Proof. We say interval $\mathcal{I}'_{\sigma_i}$ is trimmed in T', if $\alpha_{\sigma_i} < x$. Note that for every interval $\mathcal{I}'_{\sigma_i}$ that is not trimmed, $\mathcal{I}'_{\sigma_i}$ is exactly $\mathcal{I}_{\sigma_i}$. Hence, if T'' contains no trimmed interval, regarding irreducibility of T, we have

$$\Phi(T'') > \Phi(T) = \Phi(T').$$

Thus, it only remains to prove the claim for the case that T'' contains a trimmed interval. By Observation 21, we can assume that $\mathsf{DOM}(T'')$ is solid. Let j be the maximum index, such that $\mathcal{I}'_{\sigma_j} \in T''$. Since I_{σ_j} is entirely within $\mathcal{I}_{\sigma_j}$, $\beta_{\sigma_j} \ge b_{\sigma_j}$. Since $b'_{\sigma_j} = b_{\sigma_j}$,

$$\mathsf{DOM}(T'') = [x, b_{\sigma_j}].$$

On the other hand, T'' contains at most $j - \ell + 1$ intervals. By Inequality (6),

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

and

$$\Phi(T'') \geq \frac{b_{\sigma_j} - x}{j - \ell + 1}.$$

Thus, we have

$$\Phi(T') = \frac{b_{\sigma_k} - x}{k - \ell + 1} = \frac{(b_{\sigma_j} - x) + (b_{\sigma_k} - b_{\sigma_j})}{(j - \ell + 1) + (k - j)} \tag{7}$$

Note that

$$\begin{aligned}\frac{(b_{\sigma_k} - b_{\sigma_j})}{(k - j)} &= |I_{\sigma_1}| = |I_{\sigma_2}| = \ldots = |I_{\sigma_k}| \\ &< \Phi(T) \\ &= \Phi(T'). \end{aligned} \tag{8}$$

Combining Inequalities (7) and (8) together with Observation 21 yields:

$$\frac{(b_{\sigma_j} - x)}{(j - \ell + 1)} > \Phi(T') \Rightarrow \Phi(T'') > \Phi(T').$$

□

Lemma 8 states that the set of intervals in T' admit the properties described in Lemma 5. Furthermore, regarding Lemma 6, T' is a strict subset of T. By induction hypothesis, we know that one can cut $\mathsf{DOM}(T')$ into at most $2(k - \ell + 1) - 2$ pieces and allocate them to players $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \ldots, p_{\sigma_k}$ such that both 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 $p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \ldots, p_{\sigma_k}$ 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 $\mathcal{N}_{T''} = \mathcal{N}_T \setminus \{p_{\sigma_\ell}, p_{\sigma_{\ell+1}}, \ldots, p_{\sigma_k}\}$ after shrinking $\mathsf{DOM}(T')$. Then, T'' is irreducible and $\Phi(T'') = \Phi(T')$.*

Proof. We have $\Phi(T) = \Phi(T')$, which means:

$$\frac{\mathsf{DOM}(T)}{|T|} = \frac{\mathsf{DOM}(T')}{|T'|}.$$

Thus,

$$\frac{\mathsf{DOM}(T) - \mathsf{DOM}(T')}{|\mathcal{T}| - |T'|} = \Phi(T) = \Phi(T'),$$

that is $\Phi(T'') = \Phi(T)$.

Call an interval $\mathcal{I}_i$ *shrinked*, if $\mathcal{I}_i \cup \mathsf{DOM}(T') \neq \emptyset$. Consider a set $\hat{T} \subset T''$ of intervals. If $\hat{T}$ contains no shrinked interval, then $\Phi(\hat{T})$ has the same value as before shrinking $\mathsf{DOM}(T')$, which by irreducibility, we have:

$$\Phi(\hat{T}) > \Phi(\mathcal{T}) = \Phi(T'').$$

On the other hand, assume that $\hat{T}$ contains at least one shrinked interval. Let $\hat{T}'$ be the set of intervals in $\hat{T} \cup T'$ before shrinking T'. Since $\hat{T}$ has a shrinked interval, $\hat{T}'$ is solid. By irreducibility, we know:

$$\Phi(\hat{T}') > \Phi(T) = \Phi(T') = \Phi(T'').$$

On the other hand, $|\mathsf{DOM}(\hat{T})| = |\mathsf{DOM}(\hat{T}')| - |\mathsf{DOM}(T')|$. Thus, we have:

$$\begin{aligned}\Phi(\hat{T}) &= \frac{|\mathsf{DOM}(\hat{T})|}{|\hat{T}|} \\ &= \frac{|\mathsf{DOM}(\hat{T}')| - |\mathsf{DOM}(T')|}{|\hat{T}'| - |T'|}\end{aligned}$$

Regarding Observation 21, we have $\Phi(\hat{T}) > \Phi(T) = \Phi(T'')$. □

According to Lemma 9, we can use induction hypothesis to show that T'' can be allocated to the players in $\mathcal{N}_{T''}$ via $2(\ell - 1) - 2$ cuts. 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 EFGISM as follows: among all subsets of $\mathcal{N}$, we find a subset whose corresponding intervals have 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 *Let N be a subset whose corresponding intervals have the minimum density and let T be the set share intervals which have the minimum density.Then, T can be found in polynomial time.*

Proof. There are n^2 different possible choices for the starting point and the ending points of $\mathsf{DOM}(T)$. By fixing these points, in order to minimize the density of T, we must add all the intervals that are within the selected domain to T. Thus, the set T can be found by minimizing over all possible choices of domain. □

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. Afterwards, we remove the players in N from $\mathcal{N}$ and shrink $\mathsf{DOM}(T)$ from $\mathcal{C}$. Next, we recursively allocate the remaining piece of the cake to remaining players using EFGISM. In Algorithm 2 you can find a pseudocode for EFGISM.

Algorithm 2 EFGISM algorithm

function EFGISM($\mathcal{N}, \mathcal{T}, \mathcal{C}$)
  if $\mathcal{C} \neq \emptyset$ **then**
    $T = \arg\min_{T' \subseteq \mathcal{T}} \Phi(T')$
    $N =$ players with interval in T
    $Allocate(N, \mathsf{DOM}(T))$ ▷ By Lemma 5
    $Shrink(\mathcal{C}, \mathsf{DOM}(T))$ ▷ $\mathcal{T}$ is also updated
    EFGISM($\mathcal{N} \setminus N, \mathcal{T}, \mathcal{C}$)

Theorem 3 EFGISM *is envy-free, truthful, and uses at most* $2(n-1)$ *cuts.*

Proof. Envy-freeness and truthfulness: We credit the proof for envy-freeness and truthfulness to [11]. In [11], the following statement is proved (restated for the case of intervals):

> "Let $\mathcal{A}$ be the algorithm that in each step finds a set $\mathcal{T}$ of intervals with minimum $\Phi(\mathcal{T})$ and allocates it to the agents corresponding to the intervals in $\mathcal{T}$, such that every agent gets a share of size $\Phi(\mathcal{T})$ that is totally within his valuation interval; then $\mathcal{A}$ is envy-free and truthful."

Regarding the fact that our algorithm has the same structure as stated above, the algorithm is envy-free and truthful.

Number of cuts: We use induction to prove that the algorithm cuts the cake in at most $2(n-2)$ locations. For $n = 1$, the algorithm trivially allocates the entire cake to the player which needs no cut $(2 \times (1-1))$. Now, consider the first step of the algorithm, when a set $\mathcal{T}$ of intervals are selected. By Lemma 5, we cut $\mathcal{T}$ into at most $2(\mathcal{T}-1)$ points. Furthermore, we shrink $\mathsf{DOM}(\mathcal{T})$ and solve the problem recursively for the remaining part. By induction hypothesis, the recursive part needs at most $2(n-\mathcal{T}-1)$ cuts. In addition, two cuts are needed in the beginning and ending point of $\mathsf{DOM}(\mathcal{T})$. Thus, total number of cuts would be:

$$2(\mathcal{T}-1) + 2(n-\mathcal{T}-1) + 2 = 2(n-1).$$

□

7 Piecewise Constant Functions

In this section, we study a more general case of the problem 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, $|S_i| \leq m$. Here, we assume that for every p_i, $|S_i| = m$. This is without loss of generality, since one can break an interval into several sub-intervals. Thus, for every player p_i, we suppose $S_i = \{\mathcal{I}_{i,1}, \mathcal{I}_{i,2}, \ldots, \mathcal{I}_{i,m}\}$.

This section consists of two parts. In the first part, we show that for a constant number of players, one can find an 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 $n=3$ [12]. In Theorem 4, we show that for a constant number of players with piecewise constant valuation, the problem can be solved in time $\mathsf{poly}(m)$.

Theorem 4 *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 [18] we know that there exists an envy-free allocation with $n-1$ cuts. In such an allocation, there are $n-1$ cutting points. Let $0 \leq c_1 \leq c_2 \leq \cdots \leq c_{n-1} \leq 1$ be those cutting points. In addition, the valuation of each player can be specified by $2m$ constant points (2 constant points for each step) and m constant values which describe the density 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 uniform 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 i'th piece created by the cutting points ($[c_{i-1}, c_i]$) for each player p_j as a linear combination of the cutting points. However, in order to satisfy envy-freeness, we also need to know how the pieces will be allocated to the players. If we know all this information is given, 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 to the linear program is an envy-free allocation with $n-1$ cuts.

If we couldn'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 at least one of these linear programs finds a feasible solution. Hence, for constant n, by solving a polynomial number of different linear programs, we can find an envy-free allocation. □

In the second part, we exploit the 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 $p_k (k \neq i)$, $|R_{i,j,k}| \leq 1$. For this case, we prove Theorem 5.

Theorem 5 *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 $O(nm)$ places.*

Proof. Consider an instance of the problem with nm players where the valuation function of player $p_{i,j}$ is $\mathcal{I}_{i,j}$. We run 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 $2(nm-1)$ locations. Let $P_{i,j}$ be the set of intervals allocated to $p_{i,j}$ in EFGISM. We show that the allocation that allocates $P_i = \bigcup_{1 \le j \le m} P_{i,j}$ to player p_i is also envy-free.

To prove envy-freeness, we use a structural property of the expansion process: by the first invariant of the expansion process, the final solution allocates to every player $p_{i,j}$ a set of pieces that are totally within $\mathcal{I}_{i,j}$. In addition, note that for interval $\mathcal{I}_{i,j}$, $|R_{i,j,k}| \le 1$ holds for every player p_k. We have $V_i(P_i) = \sum_{1 \le j \le m} V_i(P_{i,j})$ and $V_i(P_k) = \sum_{1 \le j \le m} V_i(P_{k,j})$. Furthermore, by the 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 $V_{i,j}(P_{i,j}) \ge V_{i,j}(P_{k,l})$. Regarding the fact that $\mathcal{I}_{i,j} \cap \mathcal{I}_{k,l'} = \emptyset$ for all $l' \neq l$, we have $V_{i,j}(P_{i,j}) \ge \sum_l V_{i,j}(P_{k,l})$. Thus,

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

$$V_i(P_i) \ge \sum_j \sum_l V_{i,j}(P_{k,l}). \tag{9}$$

The right hand side of Equation (9) is at least $V_i(P_k)$. □

8 Expansion Process with Unlocking in Practice

In addition to obtaining worst-case guarantee on the number of cuts, we wish to evaluate the behavior of the expansion process with unlocking method in practice. Our experiments aim at illustrating the performance of the expansion and unlocking method from the aspect of the number of cuts. As we show in this section, the expansion and unlocking method achieves significant performance in practice.

In Sections 3.1, and 5.1 we briefly discussed the implementation details of the expansion and unlocking processes. We implemented both the processes and the codes are available in `ce.sharif.edu/~m_farhadi/source`.

As mentioned in Section 2, our basic assumption is that every piece of the cake is valuable to at least one player. In Lemma 11 we show that this assumption is w.l.o.g since every algorithm for the former can be extended to the case that the cake has zero-valued pieces, without any additional cuts. In fact, Lemma 11 demonstrates how to prevent additional cuts on the cake when parts of it have zero valuation to all the players.

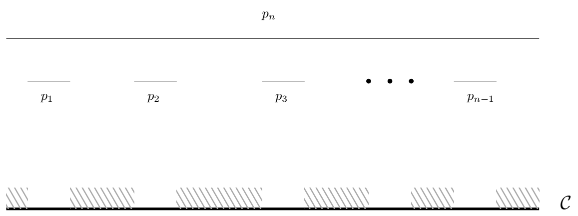


Fig. 9: The hatched pieces are allocated to p_n in EFGISM

Lemma 11 *The assumption that every piece of the cake is valuable to at least one player is w.l.o.g.*

Proof. Let $\{Z_1, \ldots, Z_q\}$ be the set of maximal intervals that are zero valued for all the players. Shrink every Z_i in $\mathcal{C}$ into a point z_i to achieve a new cake $\mathcal{C}_c$ in which our valuation assumption holds. Now, utilizing the algorithms proposed in this paper (or any other allocation algorithm), we divide $\mathcal{C}_c$ among the players. Next, for each Z_i, we allocate it to the player that received (left / right neighborhood) of z_i. Z_i will introduce a single cut (at its right / left endpoint) in $\mathcal{C}$ only if $\mathcal{C}_c$ contains a cut at z_i. Thus, the number of cuts is identical for $\mathcal{C}$ and $\mathcal{C}_c$. Finally, considering the fact that each Z_i has a zero valuation for all the players, one can easily verify that envy-freeness and truthfulness of the mechanism are preserved. □

8.1 Study Design

To test the performance of the expansion and unlocking procedure, we ran a set of experiments on the EFSC method. We generated cake cutting instances with between 2 and 500 players. For every $2 \leq i \leq 500$, we generated 4 tests containing i players. For every test, we draw the valuation interval of every player uniformly from $(0, 1)$. To generate a random interval, we separately sampled the end-points from $\mathcal{U}(0, 1)$. For every test, we calculated two objectives:

- C_{tot}: total number of cuts that are made on the cake by EFSC.
- C_{max}: the maximum number of pieces allocated to any player by EFSC.

Note that, despite the fact that the average number of pieces allocated to any player in EFGISM is less than $2n/n = 2$, there are situations in which a player may receive $\Omega(n)$ pieces. For example, consider the instance in Figure 9. Since in EFSC the share allocated to every player is totally within his valuation interval, the hatched parts of the cake will be allocated to p_n, which means his share includes n pieces of the cake.

We compare C_{tot} and C_{max} obtained from every test to the values in the optimal solution. We already know that in the optimal allocation, the number of cuts is $n - 1$ and every player receives a continuous piece of the cake. Therefore,

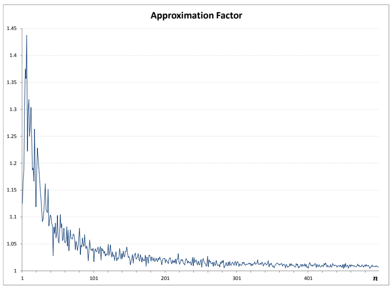


Fig. 10: approximation factor of EFSC

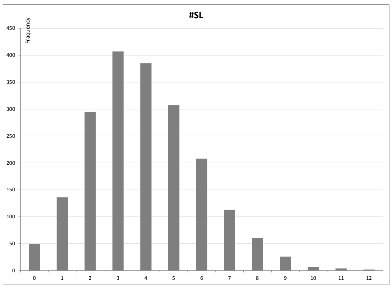


Fig. 11: $\#SL$ in different instances

we compare C_{tot} with $n-1$ and C_{max} with 1. We call the ratio $C_{tot}/(n-1)$ the *approximation factor* of the algorithm. Furthermore, we call C_{max} the *slabbing factor* of the algorithm.

8.2 Experimental Results

The results of the performed experiments are depicted in Figures 10, 11, and 12. Figure 10 shows the approximation factor of EFSC for different problem sizes. As you can see, the approximation factor rapidly approaches 1 by increasing n. The average approximation factor over all the experiments was 1.001.

Note that the number of cuts in EFSC is exactly equal to $n-1+\#SL$, where $\#SL$ is the number of times that the algorithm is encountered a strongly locked situation. In fact, for every strongly locked situation, we need one additional cut on the cake. In Figure 11 you can find the number of strongly locked chains occurred in the experiments. As you can see, none of the tests encountered more than 12 strongly locked situations. The average number of strongly locked situations over the performed experiments is 3.96. As an interesting open question, one can theoretically provide an upper-bound on the expected number of strongly locked situations.

It is worth to mention that in 0.024 of the tests (49 tests out of 2000 total tests), the returning allocations were optimal. However, the ordering property defined in Section 4 are not necessarily preserved in these instances.

As mentioned, in contrast to the number of cuts, our method provides no approximation better than $n-1$ on the number of pieces allocated to a single player. But the worst-case scenarios such as the one illustrated in Figure 9 are very unlikely to happen in practice. In our experiments, no player received more than 5 pieces in any test. Furthermore, in average, the maximum number of pieces allocated to the players in every test was 2.052. In Figure 12 you can find the value of average slabbing factor for different problem sizes. Interestingly, the average number of the slabbing factor is a decreasing function of the problem size.

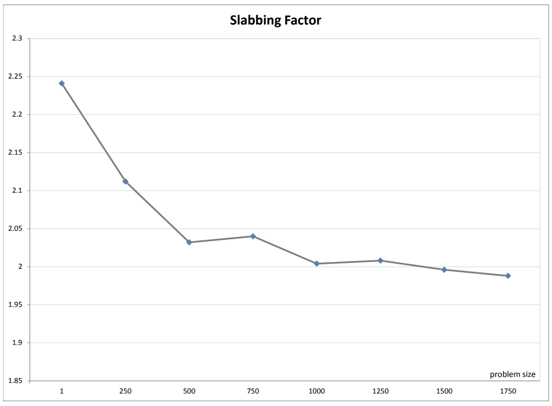


Fig. 12: Average slabbing factor for different problem sizes.

9 Discussion

In this paper, we introduced the expansion process for envy-free and truthful allocation of the cake with a small number of cuts. The process is designed for the case that the valuation of each player is a single interval. A future direction would be generalizing this process for piecewise-uniform and piecewise-constant valuations. We believe that a generalized form of EFGISM can handle the case that the valuation functions are piecewise-uniform with k steps, with $O(nk)$ cuts. To present such an algorithm, we only need to extend Theorem 5 for more generalized valuations.

In a very recent work, Bei et al. [6] proved that no deterministic truthful and envy-free mechanism exists with $n-1$ cuts. Thus, a gap between $n-1$ and $2n-1$ cut remains. We conjecture that the number of cuts made by the expansion process with unlocking is optimal for the case that the valuation functions are single intervals, i.e., no allocation can guarantee both envy-freeness and truthfulness with less than $2n-1$ cuts.

The experimental results in Section 8 demonstrate the high performance of the expansion and unlocking process in term of the number of cuts. An interesting direction would be supporting these results by providing theoretical proofs for the stochastic settings.

References

1. R. Alijani, M. Farhadi, M. Ghodsi, M. Seddighin, and A. S. Tajik, *Envy-free mechanisms with minimum number of cuts*, in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

2. H. Aziz and S. Mackenzie, *A discrete and bounded envy-free cake cutting protocol for any number of agents*, in Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on, IEEE, 2016, pp. 416–427.
3. H. Aziz and C. Ye, *Cake cutting algorithms for piecewise constant and piecewise uniform valuations*, in International Conference on Web and Internet Economics, Springer, 2014, pp. 1–14.
4. J. B. Barbanel and S. J. Brams, *Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond*, Mathematical Social Sciences, 48 (2004), pp. 251–269.
5. X. Bei, N. Chen, X. Hua, B. Tao, and E. Yang, *Optimal proportional cake cutting with connected pieces.*, in Twenty-sixth AAAI Conference on Artificial Intelligence, 2012.
6. X. Bei, N. Chen, G. Huzhang, B. Tao, and J. Wu, *Cake cutting: Envy and truth*.
7. S. J. Brams, M. Feldman, J. K. Lai, J. Morgenstern, and A. D. Procaccia, *On maxsum fair cake divisions.*, in Twenty-sixth AAAI Conference on Artificial Intelligence, 2012.
8. S. J. Brams, M. A. Jones, C. Klamler, et al., *Better ways to cut a cake*, Notices of the AMS, 53 (2006), pp. 1314–1321.
9. S. J. Brams and A. D. Taylor, *An envy-free cake division protocol*, American Mathematical Monthly, (1995), pp. 9–18.
10. I. Caragiannis, J. K. Lai, and A. D. Procaccia, *Towards more expressive cake cutting*, in IJCAI proceedings-international joint conference on artificial intelligence, vol. 22, 2011, p. 127.
11. Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia, *Truth, justice, and cake cutting*, Games and Economic Behavior, 77 (2013), pp. 284–297.
12. X. Deng, Q. Qi, and A. Saberi, *Algorithmic solutions for envy-free cake cutting*, Operations Research, 60 (2012), pp. 1461–1476.
13. D. Kurokawa, J. K. Lai, and A. D. Procaccia, *How to cut a cake before the party ends.*, in Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013.
14. A. Maya and N. Nisan, *Incentive compatible two player cake cutting*, in International Workshop on Internet and Network Economics, Springer, 2012, pp. 170–183.
15. A. D. Procaccia, *Cake cutting: not just child's play*, Communications of the ACM, 56 (2013), pp. 78–87.
16. A. D. Procaccia, *Cake cutting algorithms*, in Handbook of Computational Social Choice, chapter 13, Citeseer, 2015.
17. H. Steinhaus, *The problem of fair division*, Econometrica, 16 (1948).
18. W. Stromquist, *How to cut a cake fairly*, American Mathematical Monthly, (1980), pp. 640–644.
19. W. Stromquist, *Envy-free cake divisions cannot be found by finite protocols*, the electronic journal of combinatorics, 15 (2008), p. 11.