How to Cut Cake

An Overview of Fair Online Resource Allocation Problems

Arkaprava Choudhury

University of Toronto

arka@cs.toronto.edu

UofT TSS

May 3, 2023

Presentation Overview

Breaking it down

Fair Online Resource Allocation Problems

Components

1. "Fair"
2. "Online"
3. "Resource allocation"

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*,

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*, and you have *n people* to allocate the resource to.

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*, and you have *n people* to allocate the resource to. You wish to maximize a particular objective through this allocation.

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*, and you have *n people* to allocate the resource to. You wish to maximize a particular objective through this allocation.

Definitions

1. Divisibility whether the resource can be divided, and, if applicable, the finest refinement possible

Allocation problems

Problem statement

You have some quantity, say m *units*, of some *resource*, and you have n *people* to allocate the resource to. You wish to maximize a particular objective through this allocation.

Definitions

1. Divisibility whether the resource can be divided, and, if applicable, the finest refinement possible
2. Homogeneity whether all parts of the resource are worth the same to each person

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*, and you have *n people* to allocate the resource to. You wish to maximize a particular objective through this allocation.

Definitions

1. Divisibility whether the resource can be divided, and, if applicable, the finest refinement possible
2. Homogeneity whether all parts of the resource are worth the same to each person
3. Allocation is a (perhaps partial) *partitioning* of the available resource amongst (a subset of) the *population*

Allocation problems

Problem statement

You have some quantity, say *m units*, of some *resource*, and you have *n people* to allocate the resource to. You wish to maximize a particular objective through this allocation.

Definitions

1. Divisibility whether the resource can be divided, and, if applicable, the finest refinement possible
2. Homogeneity whether all parts of the resource are worth the same to each person
3. Allocation is a (perhaps partial) *partitioning* of the available resource amongst (a subset of) the *population*
4. Objective we shall take this to be the net worth of the allocation, subject to *fairness*

Online algorithms

Intuitive idea

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case).

Online algorithms

Intuitive idea

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case). After each input agent is presented, the algorithm makes a decision (*irrevocably*, in this case).

(Minimal) online cake-cutting

Defining the problem

Informally...

Congratulations! Today is your birthday so you take a cake into the office to share with your colleagues at tea time. However, as some people have to leave early, you cannot wait for everyone to arrive before you start sharing (allocate) the cake. How do you proceed fairly?

— Toby Walsh (Online Cake Cutting, 2011)

(Minimal) online cake-cutting

Defining the problem

Informally...

Congratulations! Today is your birthday so you take a cake into the office to share with your colleagues at tea time. However, as some people have to leave early, you cannot wait for everyone to arrive before you start sharing (allocate) the cake. How do you proceed fairly?
— Toby Walsh (Online Cake Cutting, 2011)

Simplification

cake $\to I = [0, 1]$;

Allocation

Cutting

If S is a finite set of closed intervals, then:

1. S is a cutting;
2. $\forall [a, b] \in S$ and $c \in (a, b)$, the set $S \cup \{[a, c], [b, c]\} \setminus [a, b]$ is a cutting.

Allocation

Cutting

If S is a finite set of closed intervals, then:

1. S is a cutting;
2. $\forall [a,b] \in S$ and $c \in (a,b)$, the set $S \cup \{[a,c],[b,c]\} \backslash [a,b]$ is a cutting.

Allocation

An allocation of the cake $I = [0,1]$ among the set of agents $[n]$ is a partition of some cutting of $\{I\}$ into n subsets, $A_1, \ldots, A_n$.

Allocation

Cutting

If S is a finite set of closed intervals, then:

1. S is a cutting;
2. $\forall [a,b] \in S$ and $c \in (a,b)$, the set $S \cup \{[a,c],[b,c]\} \setminus [a,b]$ is a cutting.

Allocation

An allocation of the cake $I = [0,1]$ among the set of agents $[n]$ is a partition of some cutting of $\{I\}$ into n subsets, $A_1, \ldots, A_n$.

Simple allocation

An allocation using only n disjoint intervals.

Agent preferences

Valuation

For each $j \in [n]$, define the valuation of agent j denoted by $v_j : 2^I \to \mathbb{R}_{\geq 0}$ given by $v_j(J) = \int_J f_j$ where f_j is a piecewise continuous value density function, such that, for all $j \in [n]$:

Agent preferences

Valuation

For each $j \in [n]$, define the valuation of agent j denoted by $v_j : 2^I \to \mathbb{R}_{\geq 0}$ given by $v_j(J) = \int_J f_j$ where f_j is a piecewise continuous value density function, such that, for all $j \in [n]$:

- normalized: $v_j(I) = 1$
- additive: for any two closed disjoint sub-intervals X, Y, $v_j(X \sqcup Y) = v_j(X) + v_j(Y)$

Agent preferences

Valuation

For each $j \in [n]$, define the valuation of agent j denoted by $v_j : 2^I \to \mathbb{R}_{\geq 0}$ given by $v_j(J) = \int_J f_j$ where f_j is a piecewise continuous value density function, such that, for all $j \in [n]$:

- normalized: $v_j(I) = 1$
- additive: for any two closed disjoint sub-intervals X, Y, $v_j(X \sqcup Y) = v_j(X) + v_j(Y)$

Set valuation

For a finite set of intervals S, we define, for all $j \in [n]$, $v_j(S) = \sum_{[a,b] \in S} v_j([a,b])$; in particular, v_j is additive.

Agent preferences

Space of possible valuations

Let $\tilde{C}^0$ denote the space of piecewise continuous functions on I.
Let $\mathscr{V}$ denote the set of possible valuations $v = (v_1, \ldots, v_n)$ where $v_j = \int f_j$, for $f_j \in \tilde{C}^0$.

Agent preferences

Space of possible valuations

Let $\tilde{C}^0$ denote the space of piecewise continuous functions on I.
Let $\mathscr{V}$ denote the set of possible valuations $v = (v_1, \ldots, v_n)$ where $v_j = \int f_j$, for $f_j \in \tilde{C}^0$.

Algorithm output

Given some algorithm $\textsc{Alg}$ for this problem and valuations $v_1, \ldots, v_n$, denote the resulting allocation by $A = (A_1, \ldots, A_n) = \textsc{Alg}(v_1, \ldots, v_n)$.

Input model

Complexity

It may take infinite precision to specify a valuation function without discretisation or other approximations. We will instead consider the query complexity with the following model.

Input model

Complexity

It may take infinite precision to specify a valuation function without discretisation or other approximations. We will instead consider the query complexity with the following model.

Robertson-Webb

Two oracles for each $j \in [n]$ as follows

1. $Eval_j(x, y)$ returns the value $v_j([x, y])$, where $[x, y] \subseteq I$;
2. $Cut_j(x, \alpha)$ returns the value $y \in [x, 1]$ such that $v_j([x, y]) = \alpha$.

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. (Strong) proportionality:
   "Each agent *feels* they got a fair share of the cake"

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. (Strong) proportionality:
   "Each agent *feels* they got a fair share of the cake"

$$\forall j \in [n], v_i(A_i) \geq 1/n$$

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy:
   "No agent is envious of some other agent's share"

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy:
   "No agent is envious of some other agent's share"

$$\forall i, j \in [n], v_i(A_i) \geq v_i(A_j)$$

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy
3. Equitability:
   "All agents are equally content with their share"

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy
3. Equitability:
   "All agents are equally content with their share"

$$\forall i, j \in [n], v_i(A_i) = v_j(A_j)$$

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy
3. Equitability
4. Truthfulness:
   "No agent can profit by falsifying their preferences"

Some classic requisites

Given an algorithm f for the cake-cutting problem, we define the following qualities.

1. Proportionality
2. No envy
3. Equitability
4. Truthfulness:
   "No agent can profit by falsifying their preferences"

$$\forall i \in [n], \forall w_i \in \mathscr{V}_i,$$
$$v_i(f(v_1, \ldots, v_i, \ldots, v_n)) \geq v_i(f(v_1, \ldots, w_i, w_n))$$

Online fairness criteria

Lemma

No envy implies proportionality.

Online fairness criteria

Lemma

No envy implies proportionality.

Lemma

No online cake cutting algorithm is proportional, envy-free, or equitable

Proof.

Suppose some agent i leaves before agent n arrives. A_i is then independent of v_n. If $v_n(A_i) = 1$, agent n will not value any allocation outside A_i. So, not proportional. Since no envy implies proportionality, not envy-free either.

Suppose allocation was equitable, so all agents receive some cake they value. Again, A_i is independent of v_n for the first leaving agent i. □

Online fairness criteria

Online proportionality

Weak proportionality

Each agent j is assigned at least r/k of the total value of the cake to their pieces where

1. r is the value of the remaining amount of unallocated cake when agent j arrives;
2. k is the number of agents yet to be allocated cake at this point.

Online fairness criteria

Online no envy

1. Weakly envy-free: agents do not value cake allocated to agents after their arrival more than their own;
2. Immediately envy-free: agents do not value cake allocated to any agent after their arrival and before their departure more than their own

Online fairness criteria

Online no envy

1. Weakly envy-free: agents do not value cake allocated to agents after their arrival more than their own;
2. Immediately envy-free: agents do not value cake allocated to any agent after their arrival and before their departure more than their own

Lemma

No envy implies weakly envy-free. Weakly envy-free implies immediately envy-free.

Online fairness criteria

Online equitability

First-come-first-serve

No agent's value of their assigned share can decrease if they arrive earlier in the input sequence and all other agents are left in the same relative positions; formally defined as *arrival monotone* or *meta-envy*).

Online fairness criteria

Online equitability

First-come-first-serve

No agent's value of their assigned share can decrease if they arrive earlier in the input sequence and all other agents are left in the same relative positions; formally defined as *arrival monotone* or *meta-envy*).

More generally, "no agent can profit by a change in their arrival order"

Online fairness criteria

Online equitability

First-come-first-serve

No agent's value of their assigned share can decrease if they arrive earlier in the input sequence and all other agents are left in the same relative positions; formally defined as *arrival monotone* or *meta-envy*).

More generally, "no agent can profit by a change in their arrival order"

$$\forall \{i, j\} \in \binom{[n]}{2},$$

$$v_i(f_i(v_1, \ldots, v_i, \ldots, v_j, \ldots, v_n)) \geq v_i(f_j(v_1, \ldots, v_j, \ldots, v_i, \ldots, v_n))$$

Lemma

Equitability implies arrival monotonicity.

Cut-and-choose algorithm

Each application shall [...] allow two mining operations. The Authority shall designate which part is to be reserved solely for the conduct of activities by the Authority.
— UN Convention on the Law of the Sea

Cut-and-choose algorithm

Algorithm 1 I cut but you choose

```
procedure CUT-AND-CHOOSE
    for j = 1 → n−1 rounds do
        The earliest arriving agent so far j cuts the remaining cake once, creating two disjoint intervals X, Y with X ⊔ Y = I_j.
        The second earliest arriving agent so far j+1 chooses whether to take X and leave, or give X to the cutting agent who leaves.
        I_{j+1} ← Y.
    end for
    The last remaining agent takes the leftover cake.
end procedure
```

Implementing cut-and-choose

Using Robertson-Webb oracles

Each iteration of the loop makes one oracle query to the *Cut* oracle, and another to the *Eval* oracle, so overall query complexity is $\Theta(n)$.

Fairness of cut-and-choose

Lemma

The online cut-and-choose procedure is weakly proportional and immediately envy free. However, it is not weakly envy free, equitable, truthful, or arrival monotonic.

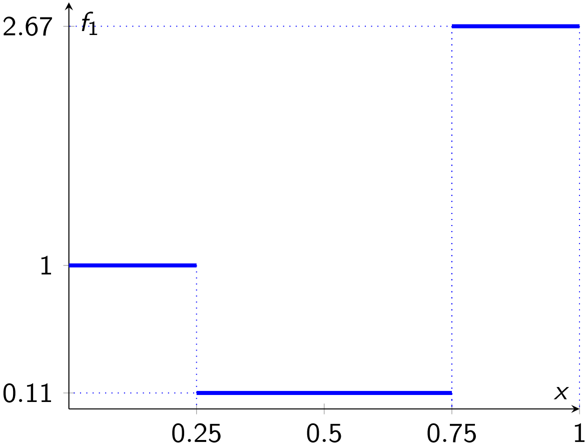
Proof.

Suppose agent i cuts a slice c_i. If allocated the slice, they would want $v_i(c_i) \geq r/k$. But, if not allocated this piece, they would want $v_i(c_i) \leq r/k$. Thus, the best option is to choose $v_i(c_i) = r/k$.

By generalization, this holds for all i, so this is weakly proportional. Also, trivially immediately envy-free.
Consider the following counter-example with 4 agents in the order $1 \to 2 \to 3 \to 4$ for the negative results. $\square$

Fairness of cut-and-choose (contd.)

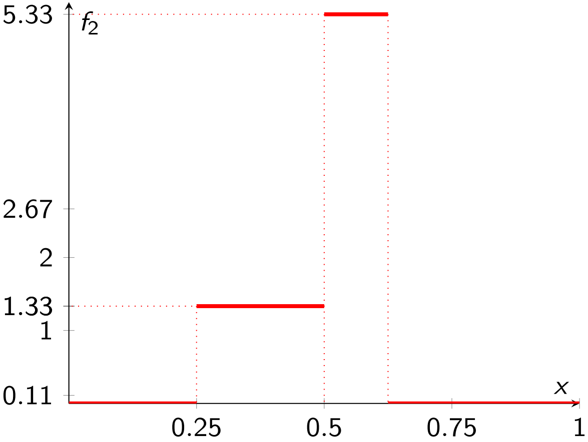
Proof.



□

Fairness of cut-and-choose (contd.)

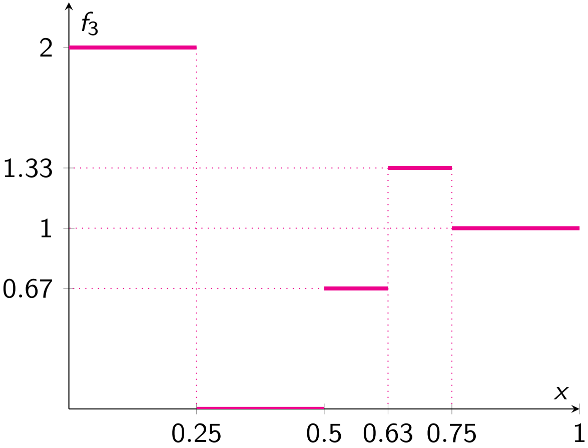
Proof.



□

Fairness of cut-and-choose (contd.)

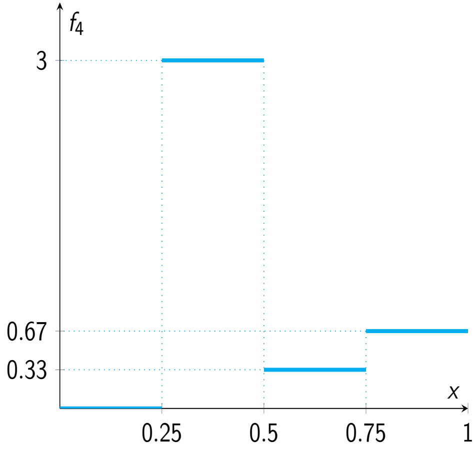
Proof.



□

Fairness of cut-and-choose (contd.)

Proof.



□

Limitations of cut-and-choose

In addition to the negative results mentioned above...

1. We must know n, i.e., the total number of agents partaking in the cake. If n is unknown to the algorithm, then the last agent will be forced into a deadlock.
2. Each agent may have to wait an unspecified amount of time until the next agent shows up, or until they can finally leave with a slice of cake.
3. In particular, since cut-and-choose is not arrival monotone, every agent will prefer to be the second agent instead of the first.

Discounted cut-and-choose

Additional assumptions

1. Each agent $j \in [n]$ has some discount factor δ_j.
2. We also have a maximum patience $T > 0$ which is common for all agents.
3. We have a clock available near the cake, so each agent can note their arrival time.
4. We have a counter near the cake, which each agent, upon their arrival, presses to increment by one.
5. The total number of agents is unknown to all agents (and to the algorithm), but the maximum possible number is n.

Discounted cut-and-choose

Additional assumptions

1. Each agent $j \in [n]$ has some discount factor δ_j.
2. We also have a maximum patience $T > 0$ which is common for all agents.
3. We have a clock available near the cake, so each agent can note their arrival time.
4. We have a counter near the cake, which each agent, upon their arrival, presses to increment by one.
5. The total number of agents is unknown to all agents (and to the algorithm), but the maximum possible number is n.

Arrival distribution

In particular, it is reasonable to assume that the arrival of agents is given by some truncated Poisson distribution with some parameter λ. Let λ be given.

Preliminary results

Lemma

For any interval of duration T, the probability of an agent arriving in this interval is $\lambda T + h(T)$, where $h(T) \to 0$ as $T \to 0$.

Let T be such that λT is sufficiently large.

Notation

Define $S_{i,j}$ for $i, j \in [n]$ as

$$S_{i,j} = \frac{(1-\delta_j)\delta_j^{n-1-i}}{2-\delta_j-\delta_j^{n-i}}$$

Modified algorithm

Outline

When each agent j arrives...

1. They note the number of agents already arrived and increment the counter, say it now reads k. First, suppose $k \neq n$.

Modified algorithm

Outline

When each agent j arrives...

1. They note the number of agents already arrived and increment the counter, say it now reads k. First, suppose $k \neq n$.
2. If there is an agent already waiting, after having cut C' and left C, agent j chooses the slice C' when $v_j(C') \geq \delta_j S_{k,j} v_j(C)$.
3. If there is no agent waiting, agent j cuts a slice C' from the remaining cake C s.t. $v_j(C') = S_{k,j} v_j(C)$. They then wait for time T.

Modified algorithm

Outline

When each agent j arrives...

1. They note the number of agents already arrived and increment the counter, say it now reads k. First, suppose $k \neq n$.
2. If there is an agent already waiting, after having cut C' and left C, agent j chooses the slice C' when $v_j(C') \geq \delta_j S_{k,j} v_j(C)$.
3. If there is no agent waiting, agent j cuts a slice C' from the remaining cake C s.t. $v_j(C') = S_{k,j} v_j(C)$. They then wait for time T.
4. If no agent arrives in this duration, agent j runs off with C'.
5. If another agent arrives, then the procedure restarts accordingly.

Modified algorithm

Outline

When each agent j arrives...

1. They note the number of agents already arrived and increment the counter, say it now reads k. First, suppose $k \neq n$.
2. If there is an agent already waiting, after having cut C' and left C, agent j chooses the slice C' when $v_j(C') \geq \delta_j S_{k,j} v_j(C)$.
3. If there is no agent waiting, agent j cuts a slice C' from the remaining cake C s.t. $v_j(C') = S_{k,j} v_j(C)$. They then wait for time T.
4. If no agent arrives in this duration, agent j runs off with C'.
5. If another agent arrives, then the procedure restarts accordingly.
6. If $k = n$ and no agent waiting, then run off with the cake!
7. If an agent is waiting, then agent j chooses a slice greedily.

Fairness and limitations

Theorem

The above procedure is immediately envy free, arrival monotone, and truthful. It is not weakly envy free or equitable.

Fairness and limitations

Theorem

The above procedure is immediately envy free, arrival monotone, and truthful. It is not weakly envy free or equitable.

Limitations

1. Nefarious agents may be encouraged to leave before time T has passed.
2. Nefarious agents may delay cutting the cake until very little of the duration T remains.

Online moving knife

1. several rounds of cutting ($n-1$ rounds for minimal cutting)
2. in each round, the algorithm moves a knife from the left to the right, and only stops when some agent declares it to stop
3. at that point, the algorithm cuts the cake and that agent leaves with their share of cake, i.e., the part to the left of the cut.

Offline Dubins-Spanier in Robertson-Webb

Algorithm 2 Offline Dubins-Spanier

procedure DUBINS-SPANIER
    $x_1 \leftarrow 0$; $I_1 \leftarrow [x_1, 1]$; $S_1 \leftarrow [n]$.
    for $j = 1 \rightarrow n-1$ rounds **do**
        $i^* \leftarrow \arg\min\{Cut_i(x_j, 1/n) : i \in S_j\}$
        $x_{j+1} \leftarrow Cut_{i^*}(x_j, 1/n)$
        Allocate $[x_j, x_{j+1}]$ to agent i^*.
        $I_{j+1} \leftarrow [x_{j+1}, 1]$; $S_{j+1} \leftarrow S_j \setminus \{i^*\}$.
    end for
    The last remaining agent takes the leftover cake.
end procedure

Online Dubins-Spanier

Briefly...

Online Dubins-Spanier

Briefly...

1. Given $k < n$, start a moving knife procedure with the first k agents.
2. At the end of the procedure, if the last agent is yet to come, then wait for the next agent and restart the procedure with k agents again.
3. If there are no more agents to come, restart with $k-1$ agents. Repeat until only one agent remains. Allocate the remainder of the cake to that agent.

Online Dubins-Spanier

Briefly...

1. Given $k < n$, start a moving knife procedure with the first k agents.
2. At the end of the procedure, if the last agent is yet to come, then wait for the next agent and restart the procedure with k agents again.
3. If there are no more agents to come, restart with $k-1$ agents. Repeat until only one agent remains. Allocate the remainder of the cake to that agent.

Lemma

The online moving knife procedure is weakly proportional and immediately envy free. However, it is not (weakly) envy free or arrival monotonic.

Dubins-Spanier procedure

Generalized Dubins-Spanier Theorem.

Theorem

Consider a set S and n agents, and let $\mathbb{U}$ be a σ-algebra on S. Suppose each agent j has a countably-additive and nonatomic value measure $v_i : \mathbb{U} \to \mathbb{R}$. Let K be a k-partition of S. Then, the set of all $n \times k$ matrices $[M]_{ij}$ is a compact and convex set in the space of all real-valued $n \times k$ matrices.

Competitive analysis

Utilitarian measure

Consider the objective function given by

$$\mathrm{obj}(f) = \frac{1}{\sum_j v_j(A_j)}.$$

Competitive analysis

Utilitarian measure

Consider the objective function given by

$$\mathrm{obj}(f) = \frac{1}{\sum_j v_j(A_j)}.$$

Then, since $v_1(A_1) \geq \frac{1}{n}$, it follows that $\sum_j v_j(A_j) \geq \frac{1}{n}$.

Competitive analysis

Utilitarian measure

Consider the objective function given by

$$\mathrm{obj}(f) = \frac{1}{\sum_j v_j(A_j)}.$$

Then, since $v_1(A_1) \geq \frac{1}{n}$, it follows that $\sum_j v_j(A_j) \geq \frac{1}{n}$.
For any offline algorithm, the sum cannot exceed n, since $v_j(A_j) \leq 1$ for all $j \in [n]$.

Competitive analysis

Utilitarian measure

Consider the objective function given by

$$\mathrm{obj}(f) = \frac{1}{\sum_j v_j(A_j)}.$$

Then, since $v_1(A_1) \geq \frac{1}{n}$, it follows that $\sum_j v_j(A_j) \geq \frac{1}{n}$.
For any offline algorithm, the sum cannot exceed n, since $v_j(A_j) \leq 1$ for all $j \in [n]$.
The competitive ratio is then $O(n^2)$. We can construct examples to show tightness.

Truthfulness

Existing work

There exist deterministic non-minimal cutting algorithms which guarantee truthfulness. There also exist randomized **minimal** cutting algorithms guaranteeing truthfulness.

Truthfulness

Existing work

There exist deterministic non-minimal cutting algorithms which guarantee truthfulness. There also exist randomized **minimal** cutting algorithms guaranteeing truthfulness. There are also several negative results for deterministic truthful (online) cake-cutting.

Truthfulness

Existing work

There exist deterministic non-minimal cutting algorithms which guarantee truthfulness. There also exist randomized **minimal** cutting algorithms guaranteeing truthfulness. There are also several negative results for deterministic truthful (online) cake-cutting.

Open question

With what restrictions can we sacrifice randomness without losing minimalism?

Truthfulness

Existing work

There exist deterministic non-minimal cutting algorithms which guarantee truthfulness. There also exist randomized **minimal** cutting algorithms guaranteeing truthfulness. There are also several negative results for deterministic truthful (online) cake-cutting.

Open question

With what restrictions can we sacrifice randomness without losing minimalism?

Wikipedia page has a nice summary!

Collusion

Walsh and other papers also look at cases where agents can fix strategies amongst themselves, colluding to get more valuable pieces of cake. In particular, the online-cut-and-choose protocol is resistant to collusion, but the online Dubins-Spanier is not.

Other query models

Simultaneous encoding

All agents succinctly report their discretized value allocations on arrival. This helps with moving from query complexity to space complexity.

More variations of resource allocation

1. Multi-cake
2. Homogeneous goods
3. Indivisible goods
4. Combinatorial auctions

References

T. Walsh (2011)
Online Cake Cutting
Algorithmic Decision Theory. ADT 2011.

L. E. Dubins, E. H. Spanier (1961)
How to Cut a Cake Fairly
American Mathematical Monthly. Jan 1961.

K. Kubo, Y. Manabe (2016)
A Non-Blocking Online Cake-Cutting Protocol
Third International Conference on Mathematics and Computers in Sciences and in Industry

The End

Questions? Comments?