Cake Cutting Algorithms for Piecewise Constant Valuations

Haris Aziz
NICTA and UNSW

Chun Ye
Columbia University

April 9, 2014
What is cake cutting?

- A metaphor for the allocation of a divisible one-dimensional heterogeneous good among multiple agents.
- **Goal:** finding an allocation that satisfies “fairness” and “efficiency” and also induces truthful reports.
- Application in scheduling of a valuable divisible resource such as server time.
- Originally studied by Polish mathematicians in the 1940s (Steinhaus, Knaster & Banach). Received a lot of attention recently from economists and computer scientists.
Interesting titles...

- A. D. Procaccia: **Cake cutting: not just child’s play.** CACM 2013
- O. Arzi, Y. Aumann, and Y. Dombb. **Throw one’s cake and eat it too.** SAGT 2011
- A. D. Procaccia **Thou shalt covet thy neighbor’s cake** IJCAI 2009
- J. Edmonds and K. Pruhs. **Cake cutting really is not a piece of cake.** SODA 2006
Table of Contents

Setting

Deterministic Algorithms

Impossibility results

Randomized Algorithms

Conclusions
A cake is represented by the interval $[0, 1]$. There is a set $N = \{1, \ldots, n\}$ of $n$ agents. Each agent has a piecewise continuous value density function $v_i : [0, 1] \rightarrow [0, \infty)$. For any piece of cake $X \subseteq [0, 1]$, define the valuation $V_i(X) = \int_X v_i(x) \, dx$.

Agent 1’s valuation: in orange. Agent 2’s valuation: horizontal lines. $V_1([0, 0.1]) = 0.1 \times 4$
A value density function \( v \) is piecewise constant if the cake can be partitioned into a finite number of intervals such that \( v \) is constant over each interval.

A piecewise uniform valuation function \( v \) is a special case of a piecewise constant valuation function where \( v \) either takes on the value \( k \) or 0 for some \( k > 0 \).

We consider a strategic version of the cake cutting problem where agents' valuation functions are private information. The algorithm receives as inputs each agent's reported valuation function.
Properties of allocations

An allocation is a partition of the cake into a set $A = \{X_1, \ldots, X_n, W\}$, where $X_i$ is the piece of cake allocated to agent $i$, and $W$ is a piece that is not allocated.

An allocation $X$ is:

- envy-free if $V_i(X_i) \geq V_i(X_j)$ for each pair of $i, j \in N$.\(^1\)
- proportional if $V_i(X_i) \geq \frac{1}{n} V_i([0, 1])$ for every $i \in N$.
- Pareto optimal if there does not exist $X'$ such that $V_i(X'_i) \geq V_i(X_i)$ for all $i \in N$ and $V_i(X'_i) > V_i(X_i)$ for some $j \in N$.
- non-wastefulness if every part of the cake that is desired by at least one agent is allocated to some agent who desires it.

\(^1\)Later on we will consider robust versions of envy-freeness and proportionality.
An example

$$X_1 = [0, 0.1] \cup [0.7, 1], \ X_2 = [0.3, 0.7]$$ is an envy-free, proportional, and Pareto optimal allocation.
Properties of cake cutting algorithms

- A (deterministic) cake cutting algorithm is a mapping from the set of valuation profiles to the set of allocations.
- An algorithm satisfies property $P$ (e.g., Pareto optimal) if it always returns an allocation that satisfies property $P$ (e.g., Pareto optimal).
- A cake cutting algorithm is strategyproof if it is a dominant strategy for every agent to report his valuation function truthfully for all valuation profiles.
### Research Question

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Robust Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Envy-free</td>
<td>SP in expectation</td>
</tr>
<tr>
<td></td>
<td>Proportional</td>
<td></td>
</tr>
</tbody>
</table>

When the agents have piecewise constant valuations,

1. for which (maximal) subsets of properties does there exist an algorithm that satisfies these properties?
2. which (minimal) subsets of properties are incompatible with each other?
Previous Work

   
   A polynomial-time, strategyproof, Pareto efficient, and envy-free, and proportional algorithm for uniform valuations.


   A polynomial-time, envy-free, and proportional algorithm for constant valuations.
Challenge of strategyproofness

Challenge of strategyproofness: even the cut and choose protocol is not strategyproof!

Assume that the first agent only desires a specific small piece of cake, whereas the second agent uniformly values the cake. The first agent can obtain its entire desired piece, instead of just half of it, by carving that piece out.
Three new algorithms

- CCEA (Controlled Cake Eating Algorithm)
- MEA (Market Equilibrium Algorithm)
- MCSD (Mixed Constrained Serial Dictatorship)
Table of Contents

Setting

Deterministic Algorithms

Impossibility results

Randomized Algorithms

Conclusions
CCEA (Controlled Cake Eating Algorithm)

- Partition cake into intervals according to the discontinuity points. E.g.,

```
0.1 0.3 0.5 1
```

The union of the discontinuity points are \{0.1, 0.3, 0.5\}. They form the intervals \([0, 0.1], [0.1, 0.3], [0.3, 0.5],\) and \([0.5, 1]\).

- Each agent ‘eats’ his most preferred interval at the same rate. Each agent gets as much fraction of an interval as he ‘eats’.

- Since agent may be indifferent between multiple intervals, we use parametric network flows to solve the problem. \(^2\)

\(^2\)This was first proposed by Katta and Sethuraman for the random assignment problem for an extension of the PS rule.
CCEA (Controlled Cake Eating Algorithm)

\[ I_1 = [0, 0.1], \quad I_2 = [0.1, 0.3], \quad I_3 = [0.3, 0.5], \quad I_4 = [0.5, 1]. \]

\[ X_1 = [0, 0.1] \cup [0.7, 1], \quad X_2 = [0.3, 0.7]. \]
CCEA (Controlled Cake Eating Algorithm)

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/Robust Proportional</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Envy-free Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Properties of CCEA
**MEA (Market Equilibrium Algorithm)**

- Write down a convex program to satisfy Pareto optimality and envy-freeness. This particular convex program was shown to be polynomial-time via a primal-dual algorithm (N. Devanur and C. H. Papadimitriou and A. Saberi and V. Vazirani. JACM 2008)
- Envy-freeness requires examining the dual program.
- $x_{ij}$: length of interval $I_j$ that is allocated to agent $i$. $\ell_j$: total length of interval $I_j$; $v_{ij}$ is $i$’s value for interval $j$; and $u_i$ is the utility that agent $i$ derives from his allocation.

$$\max \sum_{i=1}^{n} \log(u_i)$$

s.t. $u_i = \sum_{j=1}^{k} v_{ij} x_{ij} \quad \forall i = 1, \ldots, n$

$$\sum_{i=1}^{n} x_{ij} \leq \ell_j \quad \forall j = 1, \ldots, k$$

$$x_{ij} \geq 0 \quad \forall i, j.$$
MEA (Market Equilibrium Algorithm)

\[ X_1 = [0, 0.1] \cup [0.75, 0.1] \]
\[ X_2 = [0.3, 0.75] \]

(The allocation maximizes Nash welfare)
## Table: Properties of MEA

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/Robust Proportional</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Envy-free Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MEA**
### MEA and CCEA: piecewise uniform valuations

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/Robust Proportional</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Envy-free</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Proportional</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Properties of MEA and CCEA for piecewise uniform
CCEA and MEA are equivalent for uniform valuations!

**CCEA**
- Robust envy-free
- Robust Proportional
- Non-wasteful

**MEA**
- Envy-free
- Proportional
- Pareto optimal

**Mechanism 1**
(Chen et al., GEB 2013)
- Robust envy-free
- Robust Proportional
- Pareto optimal
  - Group strategyproof
Table of Contents

Setting

Deterministic Algorithms

Impossibility results

Randomized Algorithms

Conclusions
## Impossibilities I, II and III

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Robust Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Envy-free Proportional</td>
<td>SP in expectation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Robust Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Envy-free Proportional</td>
<td>SP in expectation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Robust Proportional</td>
<td>SP</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Envy-free Proportional</td>
<td>SP in expectation</td>
</tr>
</tbody>
</table>
Table of Contents

Setting

Deterministic Algorithms

Impossibility results

Randomized Algorithms

Conclusions
MCSD (Mixed Constrained Serial Dictatorship)

CRSD (Constrained Random Serial Dictatorship): a modification of random serial dictatorship where a serial ordering is drawn at random and each agent is allocated a piece of cake of length $1/n$ that maximizes his utility out of the remaining pieces with respect to the serial ordering.

MCSD (Mixed Constrained Serial Dictatorship)

- We de-randomize CRSD by combining the allocation of all of its $n!$ possible realizations into a single allocation.
- However, the allocation is stated as outputting fractions of intervals.
- Another conversion step from the fractions of intervals into an actual allocation is required to ensure strategyproofness, and this is done via another randomization procedure.

SP not possible without final randomization; SP achieved with one random bit!
MCSD (Mixed Constrained Serial Dictatorship)

- Permutation 1, 2: agent 1 asks for $[0, 0.1] \cup [0.5, 0.9]$; agent 2 asks for $[0.3, 0.5] \cup [0.9, 1]$.

- Permutation 2, 1: agent 2 asks for $[0.3, 0.8]$; agent 1 asks for $[0, 0.1] \cup [0.8, 1]$.

- Permutation 1, 2: agent 1 gets for $\frac{[0,0.1]}{2} \cup \frac{[0.5,0.9]}{2}$; agent 2 gets $\frac{[0.3,0.5]}{2} \cup \frac{[0.9,1]}{2}$.

- Permutation 2, 1: agent 2 gets $\frac{[0.3,0.8]}{2}$; agent 1 gets $\frac{[0,0.1]}{2} \cup \frac{[0.8,1]}{2}$.

$X_1 = [0, 0.1] \cup [0.3, 0.4] \cup [0.5, 0.65] \cup [0.8, 0.9] \cup [0.9, 0.95]$  
$X_2 = [0.4, 0.5] \cup [0.65, 0.8] \cup [0.95, 1]$
Randomized algorithms

CRSD
- Strategyproof
- Robust Proportional in Expectation
- Unanimous

MCSD
- Strategyproof in Expectation
- Robust Proportional
- Unanimous
<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Fairness</th>
<th>Incentives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto optimal</td>
<td>Robust Envy-free/Envy-free</td>
<td>Group-SP</td>
</tr>
<tr>
<td>Non-wasteful</td>
<td>Robust Proportional</td>
<td>SP in expectation</td>
</tr>
<tr>
<td>Unanimous</td>
<td>Proportional</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- CCEA (Controlled Cake Eating Algorithm)
- MEA (Market Equilibrium Algorithm)
- MCSD (Mixed Constrained Serial Dictatorship)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Restriction</th>
<th>DET</th>
<th>R-EF</th>
<th>EF</th>
<th>R-PROP</th>
<th>PROP</th>
<th>GSP</th>
<th>W-GSP</th>
<th>SP</th>
<th>PO</th>
<th>NW</th>
<th>UNAN</th>
<th>POLYT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCEA</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>CCEA</td>
<td>pw uniform</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MEA</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MEA</td>
<td>pw uniform</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MCSD</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCSD</td>
<td>pw uniform</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCSD</td>
<td>2 agents</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Table: Properties satisfied by the cake cutting algorithms for pw (piecewise) constant valuations: DET (deterministic), R-EF (robust envy-freeness), EF (envy-freeness), R-PROP (robust proportionality), PROP (proportionality), GSP (group strategyproof), W-GSP (weak group strategyproof), SP (strategyproof), PO (Pareto optimal), NW (non-wasteful), UNAN (unanimity) and POLYT (polynomial-time).
Final Words

Open Problems

▶ Does there exist a deterministic strategyproof and proportional algorithm for piecewise constant valuations?

▶ Characterizations of the algorithms with respect to the properties they satisfy.

Nice Surveys


Final Words

Open Problems

- Does there exist a deterministic strategyproof and proportional algorithm for piecewise constant valuations?

- Characterizations of the algorithms with respect to the properties they satisfy.

Nice Surveys


THANK YOU!