A near equitable 2-person cake cutting algorithm

Katarína Cechlárová and Eva Pillárová

Faculty of Science, P. J. Šafárik University, Košice

Motivation

- 'fair' division of a certain resource (cake), between n people
- cake represented by interval $\langle 0, 1\rangle$
- players: different opinions about the values of different parts of the cake
- these valuations are private information of players
- various notions of fairness
- constructive approach is sought

Valuation of player i for interval $I = \langle p, q \rangle$ is denoted by $U_i(I) = U_i(p, q)$

Assumptions: the valuation is

(i) nonnegative, i.e. $U_i(I) \geq 0$ for each interval $I \subseteq \langle 0, 1 \rangle$,

(ii) additive, i.e. $U_i(I \cup J) = U_i(I) + U_i(J)$ for any two disjoint intervals I, J,

(iii) divisible, i.e. for each $I = \langle p, q \rangle \subseteq \langle 0, 1 \rangle$ and each $\lambda \in \langle 0, 1 \rangle$ there exists $r \in \langle p, q \rangle$ such that $U_i(p, r) = \lambda\, U_i(p, q)$, and

(iv) normalized, i.e. $U_i(0, 1) = 1$.

Valuation can be represented by a nonnegative integrable utility function $u_i : \langle 0, 1 \rangle \to \mathbb{R}$ such that $\int_0^1 u_i(x)\mathrm{d}x = 1$.

$$U_i(p, q) = \int_p^q u_i(x)\mathrm{d}x \text{ for } p \leq q$$

Notions of fairness

Definition

A simple cake division is a pair $d_c = (d, \varphi)$, where d is an $(n-1)$ tuple $(x_1, x_2 ..., x_{n-1})$ of cutpoints, $0 < x_1 < x_2 < ... < x_{n-1} < 1$, and $\varphi : N \to N$ is a permutation of N.

Definition

Let $d_c = (d, \varphi)$ be a simple cake division for the set of players N.
Then d_c is called

a) simple fair, if $U_{\varphi(j)}(x_{j-1}, x_j) \geq 1/n$ for each $j \in N$

b) exact, if $U_{\varphi(j)}(x_{k-1}, x_k) = 1/n$ for each $j, k \in N$

c) envy-free, if $U_{\varphi(j)}(x_{j-1}, x_j) \geq U_{\varphi(j)}(x_{k-1}, x_k)$ for each $j, k \in N$

d) equitable, if $U_{\varphi(j)}(x_{j-1}, x_j) = U_{\varphi(k)}(x_{k-1}, x_k)$ for each $j, k \in N$

Relations between notions of fairness

- each envy-free cake division is simple fair
- for $n = 2$, each simple fair cake division is envy-free, but this is not necessarily true for more than two players
- for $n > 2$ not every simple fair division is equitable (example: famous divide-and-choose procedure)
- if in an equitable division the common value of the pieces is at least $1/n$, fairness is ensured
- a fair equitable division may be neither envy-free nor exact
- for $n = 2$, equitability with values 1/2 is equivalent to exactness.

Brief history of cake cutting

- Talmud, the bankruptcy problem: one of the oldest recorded fair-division problems (Auman, Maschler 1985)
- the famous algorithm 'I cut, you choose' goes back the Hebrew Bible (Brams, Taylor 1999)
- a rigorous mathematical theory of fair division started (Steinhaus, Banach and Knaster 1948)
- existence results:
  - simple fair (Steinhaus 1948)
  - envy-free (Stronquist 1980)
- moving-knife algorithms (Gardner 1978)

Recommended reading: J. Robertson, W. Webb, *Cake Cutting Algorithms*, A.K. Peters, 1998.

Simple equitable cake divisions for two players

$e \in (0,1)$ is an equitable point if

$$U_1(0,e) = U_2(e,1) \text{ or, equivalently, } U_1(e,1) = U_2(0,e)$$

Set of all equitable points: $\mathcal{E}$

Theorem

For two players, the set $\mathcal{E}$ is always nonempty and connected. Moreover, $U_1(0,e)$ and $U_2(e,1)$ are constant on $\mathcal{E}$ and either $U_1(0,e) = 1/2$ or exactly one of the two simple divisions generated by e is simple fair for each $e \in \mathcal{E}$.

A finite algorithm uses a finite number of requests

- 'Cut the given cake piece into two pieces whose values are in the given ratio!' (cutting query)
- 'What is your value of the given cake piece?' (evaluation query)

No need to know the complete value functions of players.

Moving-knife algorithms are not finite.

Impossibility results: no finite algorithm can produce

- exact division for two players (Robertson and Webb 1997)
- envy-free division for three players where everybody gets a single piece (Stromquist 2008)

ε-equitability

Definition

Let $d_c = (d, \varphi)$ be a simple cake division and $\varepsilon > 0$ a real number. Then d_c is called ε-equitable if

$$|U_{\varphi(j)}(x_{j-1}, x_j) - U_{\varphi(k)}(x_{k-1}, x_k)| \leq \varepsilon \text{ for each } j, k \in N.$$

ε-exact algorithms

Robertson, Webb 1997

Player 1 cuts the cake into pieces smaller than $\varepsilon/2$

Then player 2 can reduce the pieces to be smaller than $\varepsilon/2$

A near exact division produced by a suitable assignment of the obtained pieces to players.

Disadvantage: many small pieces, scattered over the whole cake.

Simmons, Su 2003: Consensus halving

Theorem (Borsuk-Ulam)

For any continuous function $f : S^n \to \mathbb{R}^n$ there exist antipodal points $x, -x \in S^n$ such that $f(x) = f(-x)$.

Advantage: n cut points for n players

Algorithm `bisect`

begin a_1 :=$\mathtt{half}_1(0,1)$; b_1 :=$\mathtt{half}_2(0,1)$;
  if $a_1 = b_1$ **then** cut the cake in a_1 and halt;
  if $a_1 > b_1$ **then** rename the players;
  p_2^1 :=$\mathtt{half}_1(0,1)$; $q_2^1 := 1$; $p_2^2 := 0$; q_2^2 :=$\mathtt{half}_2(0,1)$; $j := 1$;
  repeat $j := j+1$; $a_j := \mathtt{half}_1(p_j^1, q_j^1)$; $b_j := \mathtt{half}_2(p_j^2, q_j^2)$;
    if $a_j \neq b_j$ **then**
    begin
      if $a_j < b_j$ **then** (comment: successful iteration)
      begin $\mathtt{right}(p_j^1, q_j^1, a_j, p_{j+1}^1, q_{j+1}^1)$; $\mathtt{left}(p_j^2, q_j^2, b_j, p_{j+1}^2, q_{j+1}^2)$ **end**;
      if $a_j > b_j$ **then** (comment: unsuccessful iteration)
      begin $\mathtt{left}(p_j^1, q_j^1, a_j, p_{j+1}^1, q_{j+1}^1)$; $\mathtt{right}(p_j^2, q_j^2, b_j, p_{j+1}^2, q_{j+1}^2)$ **end**;
    end
  until $a_j = b_j$ or $1/2^{j-1} < \varepsilon$;
  if $a_j \neq b_j$ **then** cut the cake in $c := \left(max\left\{p_j^1, p_j^2\right\} + min\left\{q_j^1, q_j^2\right\}\right)/2$
      else cut the cake in a_j;
end

Example 1

$$u_1(x) = \begin{cases} 2 & \text{if } x \in \langle 0, 1/4\rangle \\ 0 & \text{if } x \in (1/4, 3/4\rangle \\ 2 & \text{if } x \in (3/4, 1\rangle \end{cases} \qquad u_2(x) = \begin{cases} 3/2 & \text{if } x \in \langle 0, 1/3\rangle \\ 0 & \text{if } x \in (1/3, 2/3\rangle \\ 3/2 & \text{if } x \in (2/3, 1\rangle \end{cases}$$

Let $a_1 = 1/4 \qquad b_1 = 1/3$

$\mathcal{E} = \langle 1/3, 2/3\rangle$ with the common fair values equal to 1/2.

$$\lim_{j\to\infty} a_j = 3/4 \notin \mathcal{E} \qquad \lim_{j\to\infty} b_j = 1/3 \in \mathcal{E}$$

Example 2

$$u_1(x) = \begin{cases} 2 & \text{if } x \in \langle 0, 1/4) \\ 3 & \text{if } x \in \langle 1/4, 1/3) \\ 0 & \text{if } x \in \langle 1/3, 2/3) \\ 3/4 & \text{if } x \in \langle 2/3, 1\rangle \end{cases} \quad u_2(x) = \begin{cases} 1/2 & \text{if } x \in \langle 0, 1/2) \\ 0 & \text{if } x \in \langle 1/2, 3/4) \\ 2 & \text{if } x \in \langle 3/4, 7/8) \\ 4 & \text{if } x \in \langle 7/8, 1\rangle \end{cases}$$

$\mathcal{E} = \langle 1/2, 2/3\rangle$ with the common fair values equal to 3/4.

Second iteration successful: $A \geq 1/3$ and $B \leq 3/4$
All next iterations unsuccessful, since $a_j > 2/3$ and $b_j < 1/2$.
Both players: pieces of value 3/4.
$\lim_{j\to\infty} a_j = 2/3$ $\qquad$ $\lim_{j\to\infty} b_j = 1/2$

Second iteration unsuccessful:
$a_j \in (1/4, 1/3)$, $b_j \in (3/4, 7/8)$ for $j > 2$.
All following iterations sucessful,
$\lim_{j\to\infty} a_j = 1/3$ $\qquad$ $\lim_{j\to\infty} b_j = 3/4$

Properties of the algorithm

Lemma

In each iteration $j \geq 2$*:*

(a) $\langle p_j^i, q_j^i \rangle \subset \langle p_{j-1}^i, q_{j-1}^i \rangle$ *for both players* $i = 1, 2$

(b) $U_i(p_j^i, q_j^i) = 1/2^{j-1}$ *for both players* $i = 1, 2$

(c) $p_j^1 = a_\ell$ *and* $q_j^2 = b_\ell$ *where* ℓ*: the last successful iteration before* j

(d) $U_1(0, a_j) = U_2(b_j, 1)$

(e) $U_1(0, p_j^1) = U_2(q_j^2, 1)$*, hence* $U_1(0, A) = U_2(B, 1)$

(f) $\langle p_j^1, q_j^1 \rangle \cap \langle p_j^2, q_j^2 \rangle \neq \emptyset$

(g) $|U_1(0, c) - U_2(c, 1)| \leq 1/2^{j-1}$ *for each* $c \in \langle p_j^1, q_j^1 \rangle \cap \langle p_j^2, q_j^2 \rangle$

Proof of the Lemma

(a) Directly from the definition of algorithm `bisect`.

(b) Directly from the definition of algorithm `bisect`.

(c) Directly from the definition of algorithm `bisect`.

(d) Induction on j. Assume, that $U_1(0, a_k) = U_2(b_k, 1)$ for every $k < j$. Then
$U_1(0, a_j) = U_1(0, p_j^1) + U_1(p_j^1, a_j) = U_1(0, a_\ell) + U_1(p_j^1, q_j^1)/2 = U_2(b_\ell, 1) + U_2(p_j^2, q_j^2)/2 = U_2(b_j, q_j^2) + U_2(q_j^2, 1) = U_2(b_j, 1)$,
hence the desired equality follows.

(e) Follows from (c) and (d).

Proof of the Lemma – continued

(f) In the first iteration we have $\langle p_1^k, q_1^k \rangle = \langle 0, 1 \rangle$ for both k; in the second one $\langle p_2^1, q_2^1 \rangle = \langle a_1, 1 \rangle$ and $\langle p_2^2, q_2^2 \rangle = \langle 0, b_1 \rangle$ and since after eventuall renaming of player we have $a_j < b_j$, the claim holds for $j = 1, 2$. Now let us suppose that $\left\langle p_j^1, q_j^1 \right\rangle \cap \left\langle p_j^2, q_j^2 \right\rangle \neq \emptyset$ for some $j \geq 2$. Those two intervals intersect if and only if $p_j^1 \leq q_j^2$ and simultaneously $p_j^2 \leq q_j^1$. The algorithm proceeds to the next iteration only if $a_j \neq b_j$ and we distinguish now the successful and unsuccessful iteration.

Proof of the Lemma – continued

(i) If the iteration is successful, which happens if $a_j < b_j$, the new working intervals are $\left\langle a_j, q_j^1 \right\rangle$ and $\left\langle p_j^2, b_j \right\rangle$. Now the inequality $p_j^2 \leq q_j^1$ folows from the induction hypothesis and $a_j < b_j$ from the definition of the successful iteration.

(ii) In an unsuccessful iteration is the new working intervals are $\left\langle p_j^1, a_j \right\rangle$ and $\left\langle b_j, q_j^2 \right\rangle$. Again, the inequality $p_j^1 \leq q_j^2$ folows from the induction hypothesis and $b_j < a_j$ holds because the iteration was unsuccessful.

In both cases, the intersection of working intervals in nonempty also in the following iteration and by induction the claim is proved.

Proof of the Lemma – continued

(g) Additivity of value functions and (b) imply that $0 \leq U_1(p_j^1, c) \leq 1/2^{j-1}$ and $0 \leq U_2(c, q_j^2) \leq 1/2^{j-1}$. Then, using (e)

$$\begin{aligned} & |U_1(0,c) - U_2(c,1)| \\ = \; & |U_1(0,p_j^1) + U_1(p_j^1,c) - (U_2(c,q_j^2) + U_2(q_j^2,1))| \\ = \; & |U_1(p_j^1,c) - U_2(c,q_j^2)| \leq 1/2^{j-1} \end{aligned}$$

and the claim is proved.

Correctness of the algorithm

Theorem

For $\varepsilon > 0$, algorithm `bisect` *outputs a fair ε-equitable division.*

Proof.
If the procedure ends because $a_j = b_j$, the obtained simple division is equitable. If not, claim (f) of Lemma 1 ensures that ε-equitability for the given $\varepsilon > 0$ is achieved. Fairness is ensured by choosing the appropriate players' order: $(1, 2)$ if the condition in line 3 of the algorithm was not fulfilled, otherwise $(2, 1)$.

Properties of the algorithm for positive utility functions

Theorem

If the utility functions of both players are everywhere positive then there is a unique equitable point e, the execution of the algorithm is unique and in each iteration $j \geq 2$ we have:

(a) $min\{a_j, b_j\} \leq e \leq max\{a_j, b_j\}$

(b) $e \in \left\langle p_j^1, q_j^1 \right\rangle \cap \left\langle p_j^2, q_j^2 \right\rangle$

(c) $|U_1(0,e) - U_1(0,a_j)| < 1/2^j$ *and* $|U_2(e,1) - U_2(b_j,1)| < 1/2^j$

(d) $\lim_{j\to\infty} a_j = \lim_{j\to\infty} b_j = e$

Proof of the Theorem

(a) We distinguish three cases:

- $a_j = e$. Then, using Lemma $1(d)$, we have $U_2(b_j, 1) = U_1(0, a_j) = U_1(0, e) = U_2(e, 1)$. As function $U_2(x, 1)$ is strictly decreasing, this implies $b_j = e$.
- $a_j < e$. Since $U_1(0, x)$ is strictly increasing and $U_2(x, 1)$ is strictly decreasing in x, relations $U_2(e, 1) = U_1(0, e) > U_1(0, a_j) = U_2(b_j, 1)$ imply $e < b_j$.
- Similarly, $a_j > e$ implies $e > b_j$.

Summarizing: there are exactly three possibilities: $a_j = e = b_j$ or $a_j < e < b_j$ or $b_j > e > a_j$ and the claim is proved.

Proof of the Theorem

(b) Induction on j. Suppose that $e \in \langle p_k^1, q_k^1 \rangle \cap \langle p_k^2, q_k^2 \rangle$ for each $k < j$. According to (a), there are three different cases.

- $a_{j-1} = e = b_{j-1}$. Algorithm **bisect** terminates and there is nothing to be proved for j.
- $a_{j-1} < e < b_{j-1}$. According to the definition of the algorithm, iteration $j-1$ is successful, $p_j^1 = a_{j-1} < e < q_{j-1}^1 = q_j^1$ and $p_j^2 = p_{j-1}^2 < e < b_{j-1} = q_j^2$, the claim is proved.
- $a_{j-1} > e > b_{j-1}$. In this case iteration $j-1$ is unsuccessful, $p_j^1 = p_{j-1}^1 < e < a_{j-1} = q_j^1$ and $p_j^2 = b_{j-1} < e < q_{j-1}^2 = q_j^2$, so the claim follows.

Proof of the Theorem

(c) From the definition of bisection points and Lemma 1(b) we have for each j:

$$U_1(p_j^1, a_j) = U_1(a_j, q_j^1) = 1/2^j \text{ and } U_2(p_j^2, b_j) = U_2(b_j, q_j^2) = 1/2^j.$$

We again distinguish three cases:

- $a_j = b_j = e$ then the assertion is trivial.
- $a_j > e > b_j$. Then, using (b), $|U_1(0,e) - U_1(0,a_j)| = U_1(e, a_j) < U_1(p_j^1, a_j) = 1/2^j$. Similarly, $|U_2(e,1) - U_2(b_j,1)| = U_2(b_j, e) < U_2(b_j, q_j^2) = 1/2^j$.
- Case $a_j < e < b_j$ is proved similarly.

(d) Follows from (c).

Open questions

- Algorithm for divisions in the ratio approximately $r : s$
- Determine the maximum number v such that there exists a simple equitable division assigning each player a piece with value at least v?