CSC304: Algorithmic Game Theory and Mechanism Design Fall 2016

Allan Borodin (instructor)
Tyrone Strangway and Young Wu (TAs)

December 5, 2016

Lecture 23

- Announcements
  - I am sorry for the confusuion due to Markus being unavailable for a short period of time. Hopefully everyone was able to submit the assignment by the extended due date.
  - I note again that students are responsible for material discussed in the lectures and tutorials whether or not that material is available in the lecture slides or text. In particular, students are responsible for this last week of material. There is no tutorial this coming Friday. I will use this Wednesday partially as a tutorial.
  - The makeup Wednesday class will take place in SS 1087.
- Todays agenda
  - Our last topic for mechanism design without money (and for the course) is fair division, chapter 11 in KP text. In particular, we are considering *cake cutting*, the cutting of a divisible object.
  - You should read up to section 11.1.1; I would skip section 11.1.1 and then read section 11.2 for enjoyment, some historical perspective and some alternative fairness conditions.

Cake cutting and fair division

Consider the problem of cutting (i.e. partitioning) a cake into some number n disjoint "pieces", say $A_1, \ldots, A_n$. We want to do this in a "fair way". By definition, we view our cake as being completely *divisible*. For our more mathematical/computational purpose, we can think of the cake as the unit interval $[0, 1]$

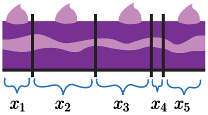

FIGURE 11.2. This figure shows a possible way to cut a cake into 5 pieces. The i^{th} piece is $B_i = [\sum_{k=1}^{i-1} x_k, \sum_{k=1}^{i} x_k)$. If the i^{th} piece goes to player j, then his value for this piece is $\mu_j(B_i)$.

Figure: Cutting a "cake" into 5 contiguous pieces

So what is a "piece" and what is "fair"? Maybe I only like certain parts of the cake while others like other parts? A "piece" is a finite collection of disjoint intervals. Pieces need not be contiguous.

Cake cutting definitions

We will assume that each agent (also called players) i has a valuation function $v_i : [0,1] \rightarrow \mathbb{R}^{\geq 0}$ that can be thought of as a cummulative distribution function. Namely,

1. For all $X \subseteq [0,1]$, $v_i(X) \geq 0$
2. For all $X, Y \subseteq [0,1] : X \cap Y = \varnothing$, $v_i(X \cup Y) = v_i(X) + v_i(Y)$
3. For all $x \in [0,1]$, $v_i(x,x) = 0$
4. $v_i([0,1]) = 1$

There are two natural definitions of "fair division":

- A division $A_1, \ldots, A_n$ is a *proportional division* if $v_i(A_i) \geq 1/n$ for all i. **Note:** This is called "fair" in the KP text but I prefer to use "fair" as a more intuitive concept.
- A division $A_1, \ldots, A_n$ is an *envy-free division* if $v_i(A_i) \geq v_i(A_j)$ for all i, j. iThat is, i does not envy j's allocation.

Proportional fairness vs envy-free fairness

The main concern is fairness. It turns out that envy-freeness is a more desireable property.

We need only consider $n \geq 2$ agents since $n = 1$ is trivial.

Every envy-free division is a proportional division

Proof: For any i, there must be some S_j such that $v_i(S_j) \geq 1/n$. Then since the division is envy-free, $v_i(A_i) \geq v_i(A_j) \geq 1/n$.

However, for $n \geq 3$, the converse is not necessarily true. Consider the following valuation profile:

$v_1([0, 1/3]) = 1/3, v_1([1/3, 2/3]) = 2/3, v_1([2/3, 1]) = 0$; v_2, v_3 have uniform valuations.

The division $A_1 = [0, 1/3), A_2 = [1/3, 2/3), A_3 = [2/3, 1]$ is proprtional since everyone is receiving a $1/3$ valuation but agent 1 envies the allocation to agent 2.

What other properties do we want in a fair cake cutting algorithm?

The following properties are all desireable but maybe not all are achieveable.

- The allocated pieces are contiguous.
- The protocol is truthful.
- The protocol is (approximately) socially optimal.
- The "complexity" of the cake-cutting algorithm is (approximately) optimal amongst envy-free or proportional divisions.

We need to define what measures of complexity we might want to consider. But before we do so, let us consider a natural algorithm for $n = 2$ agents. We will then see that the situation for $n \geq 3$ agent becomes more complicated.

The "Cut and Choose" algorithm for $n = 2$ agents

The following is a protocol (i.e. algorithm) you may have used and can be found in the Old Testement (see chapter notes):

The cut and choose algorithm

1. Agent 1 cuts the "cake" $[0, 1]$ into two equal parts according to his valuation; that is, $v_1(A_1) = v_1(A_2) = 1/2$.
2. Agent 2 chooses between A_1 and A_2.

What properties are satisfied by cut and choose?

The "Cut and Choose" algorithm for $n = 2$ agents

The following is a protocol (i.e. algorithm) you may have used and can be found in the Old Testement (see chapter notes):

The cut and choose algorithm

1. Agent 1 cuts the "cake" $[0, 1]$ into two equal parts according to his valuation; that is, $v_1(A_1) = v_1(A_2) = 1/2$.
2. Agent 2 chooses between A_1 and A_2.

What properties are satisfied by cut and choose?

- Envy-free?

The "Cut and Choose" algorithm for $n = 2$ agents

The following is a protocol (i.e. algorithm) you may have used and can be found in the Old Testement (see chapter notes):

The cut and choose algorithm

1. Agent 1 cuts the "cake" $[0,1]$ into two equal parts according to his valuation; that is, $v_1(A_1) = v_1(A_2) = 1/2$.
2. Agent 2 chooses between A_1 and A_2.

What properties are satisfied by cut and choose?

- Envy-free? Yes. Agent 2 clearly gets the best of A_1 and A_2 and is hence envy-free; agent 1 chose the partition so as to have equal value so he is also envy-free. Division must then be proportional.
- Contiguous? Yes agent 1, can choose to "query" which location x satisfies $v_1[0,x) = v_1(x,1]$
- Complexity? One cut which is clearly optimal. What else might we measure?
- Truthful? To be more precise "ex-post IC"?
- Socially optimal?

Complexity meausres

Lets look a little closer at the complexity of cut and choose. As we said, the number of cuts was optimal. More generally, for any number n of agents, the minimum possible number of cuts is $n-1$ and this is achieveable iff the division is contiguous.

In the cut and choose protocol, besides the number of cuts, there were two other queries that can be considered as possible meausres of complexity. Agent 1 asked where to cut so as to obtain a piece of value $1/2$ and agent 2 asked for his value of a piece. (Agent 2 only had to ask about one piece as that determined the value of both A_1 and A_2.).

Here is the Robertson and Web [1998] complexity model:

1. Agent 1 used a *demand query*; namely given some value v and some current piece X (i.e. X is an interval in $[0,1]$), the agent i asked where to cut the piece X so as to obtain a piece Y (i.e. Y is a subinterval of X) such that $v_i(Y) = v$.
2. Agent 2 used a *value query*; namely, given a piece X, the agent asked for his value on this piece.

The moving knife proptocol: A proportional division for any number of agents

The moving knife protocol (due to Dubins and Spanier [1961]) is the following conceptually simple algorithm:

Moving knife protocol

Initialize: Let N be the set of n agents; $X := [0, 1]$;
  start the knife at the leftmost location 0.
While $|N| > 1$
  Move the knife to the right until some agent $i \in N$ yells "STOP" having observed value $v_i(Y) = 1/|N|$ for the piece Y to the left of the knife
  Cut the "cake" and give agent i piece Y; $N := N \setminus \{i\}$; $X := X \setminus Y$
End While The one remaining player gets the remaining piece.

Equivalently, the KP text presents the algorithm recursively.

Properties of the moving knife protocol

The following properties are easy to verify

- The allocation is a contiguous division using the minimum $n - 1$ cuts.
- The division is fair. Why?

Properties of the moving knife protocol

The following properties are easy to verify

- The allocation is a contiguous division using the minimum $n-1$ cuts.
- The division is fair. Why? Note that when the first cut is made, any one of the remaining $n-1$ agents (say agent j) has value for what remains $v_j(X \setminus Y) \geq 1 - \frac{1}{n} = \frac{n-1}{n}$ which has to be shared but now shared with only $n-1$ players. Hence (by induction) every agent gets a share with value at least $\frac{1}{n}$.
- More precisely, the first $n-1$ players to yell STOP get exactly a value of $\frac{1}{n}$ and the last player obtains a value at least $\frac{1}{n}$.
- The division is *not* necessarily envy-free for $n > 2$ agents as shown in figure 11.3 of the KP text (see next slide). What agent is guaranteed to not be envious?

The negative aspects of the moving knife protocol

As we just stated, the division may not be envy-free.

The Cake:			
value to player I	$\frac{1}{3}$	0	$\frac{2}{3}$
value to player II	0	$\frac{1}{2}$	$\frac{1}{2}$
value to player III	0	0	1

Figure: an envious division

In addition to not being envy-free, the moving knife requires an active referee who is slowly moving the knife. In other words this is not a discrete algorithm with a finite number of queries of the two types we have described. Each agent would need to be continuously asking for the value of the piece to the left of the knife. Evan and Paz [1984] adapted the moving knife so that it is a discrete algorithm in the Robertson and Web model using $O(n \log n)$ queries.

The continuing story for cake cutting for the $n > 2$ agents

All of the following non-trivial cake cutting protocols result in non-contiguous but discrete computation divisions. (Note: for $n > 2$, it is not possible in genetal to have an envy-free continuous divisions.)

- For $n = 3$, Steinhaus [1943] gave a protocol that ylelds a proportional division using at most 3 cuts. This protocol is not envy-free.
- For $n = 3$, Selfridge and Conway [1960] gave a envy-free protocol with at most 5 cuts.
- For $n = 4$, Brams and Taylor [1995] gave an envy-free protocol with an unbounded number of cuts. That is, for any instance the algorithm uses some finite number of cuts and queries but that number depends on the instance; that is, for all c, there is an instance requiring at least c cuts.
- Using Spereer/s lemma (section 11.1.1 in KP), Su [1999] gave a non constructive proof that there exist envy-free divisions for all n.

The latest developments in this continuing story of cake cutting

- After 20 more year, Aziz ans MacKenzie [2016, spring] gave an envy-free protocol for $n = 4$ agents with a bounded number of cuts.
- In the latest development, the same Aziz and McKenzie [2016, fall] gave an envy-free protocol for all n using a bounded number of cuts and queries. However, so far the bound on the number of cuts is astronomically infeasible even for $n = 2$.

The latest developments in this continuing story of cake cutting

- After 20 more year, Aziz ans MacKenzie [2016, spring] gave an envy-free protocol for $n = 4$ agents with a bounded number of cuts.
- In the latest development, the same Aziz and McKenzie [2016, fall] gave an envy-free protocol for all n using a bounded number of cuts and queries. However, so far the bound on the number of cuts is astronomically infeasible even for $n = 2$.

  Namely, the current bound is :

$$n^{n^{n^{n^{n}}}}$$

.

Some impossibility results

- Edmonds and Pruhs [2006]: Any proportional cake cutting algorithm require $\Omega(n \log n)$ queries in the Robertson and Web modeli matching (asymptotically) the Evan and Paz discrete version of the moving knife.
- Stromquist [2008]: For $n \geq 3$, no envy free algorithm can produce a contiguous allocation.
- Proccacia [2009]: Any envy free algorithm requires $\Omega(n^2)$ queries in the Robertson and Web model.
- Caragiannis et al [2009]: The *price of proportionality* is $\Theta(\sqrt{n})$ and the price of envy-freeness is at least $\Omega(\sqrt{n})$ and at worst $O(n)$. These concepts are in analogy with the price of anarchy and measure how much fairness costs relative to an optimal allocation in terms of social welfare.

See Proccacia [2016] for a recent survey of such complexity results.

The Selfridge and Conway envy-free protocol for $n = 3$

For a sense of how involved cake cutting procedures can be (even when the number of cuts is small), here is the Selfridge and Conway protocol (using CMU lecture notes by Ariel Proccacia).

- Stage 0
  - 0.1 Agent 1 cuts the cake into three equal parts according to v_1
  - 0.2 Agent 2 "trims" his most valuable piece (according to v_2) of these three pieces so that the trimmed piece Z has the same value as the second most valuable piece M. Lets say that Y is what has been trimmed off and X is the what remains of the entire cake after Y is removed. Note: Z and M are each one of the three pieces in X.
- Stage 1: Dividing X
  - 1.1 Agent 3 chooses a piece of X
  - 1.2 If agent 3 chooses the trimmed piece Z then agent 2 chooses M. Otherwise, if agent 3 chooses L one of the other two pieces in X, then agent 2 chooses Z.
  - 1.3 Agent 1 chooses the remaining piece of X

The last part of the $n = 3$ protocol; dividing the trimmed off part Y

Lets say that agent $i \in \{2, 3\}$ chose Z and agent $j \in \{2, 3\}$ chose L

- Stage 2: Dividing the trimmed off piece Y
  - 2.1 Agent j divides Y into 3 equal pieces according to v_j.
  - 2.2 These three pieces of Y are allocated in the following order: Agent i chooses first, then agent 1, and then agent j.

It is easy to see that the protocol uses 5 cuts if $Y \neq \varnothing$ and 2 cuts if $Y = \varnothing$.

Proving envy-freeness is done by proving that each agent is envy free with respect to their piece in X and with respect to their piece in Y.

Completing the argument for envy-freeness

For the envy-freeness of the division of X, agent 3 chooses first so she is envy-free. Agent 2 gets one of his best two equal valued pieces and agent 1 gets a piece other than Z which has the same value to her.

For the envy-freeness of the division of Y, agent i chooses first so he is envy-free. Agent 1 chooses before agent j so she is not envious of him. Agent 1 is also not envious of agent i because i's share of both X and Y is at most $1/3$ with respect to v_1. Finally, agent j is not envious about his share of Y since he divided Y into three equally valued pieces.