Sperner's Lemma, Its Applications, and the Complexity Class PPAD

Biaoshuai Tao

John Hopcroft Center for Computer Science

bstao@sjtu.edu.cn

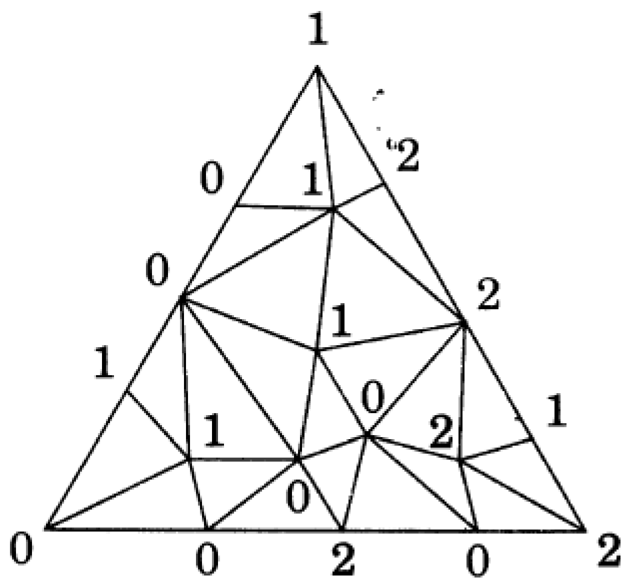
https://jhc.sjtu.edu.cn/~bstao/

Sperner’s Lemma

Simplicial Subdivision

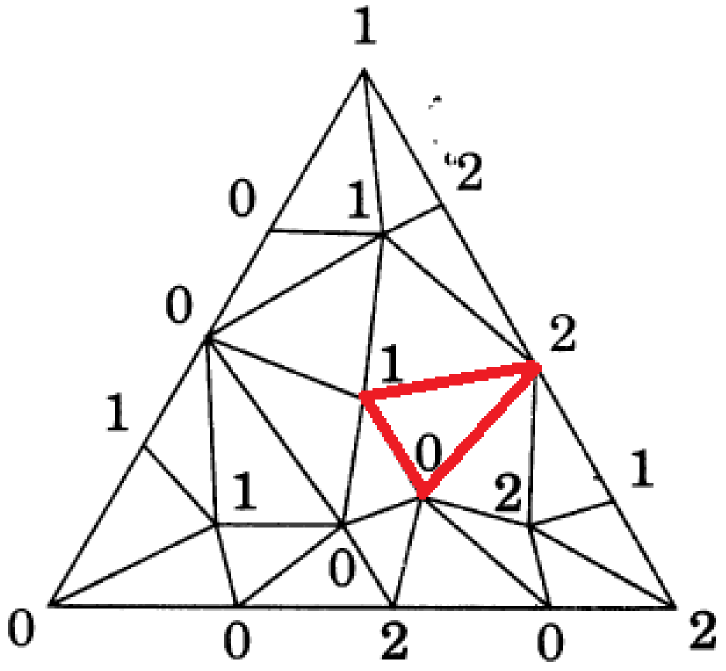
- A simplicial subdivision of a large triangle T is a partition of T into triangular cells such that every intersection of two cells is a common edge of corner.
- Nodes: corners of cells
- Proper coloring: assignments of colors from $\{0,1,2\}$ to the nodes, avoiding color i on the i-th edge of T for $i \in \{0,1,2\}$
  - In particular, the i-th corner of T must be colored i
- Fully-colored triangle: a cell having all three colors

  on its corners.



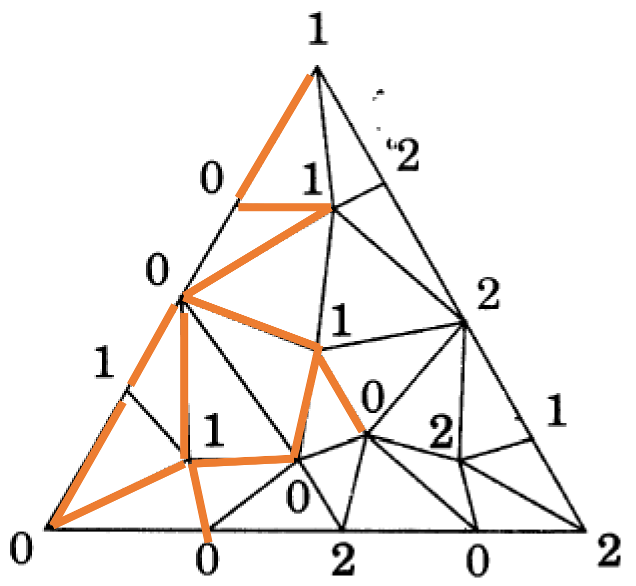
Sperner's Lemma

- **Theorem** [Sperner's Lemma (1928)]. Every properly colored simplicial subdivision has a fully-colored triangle.



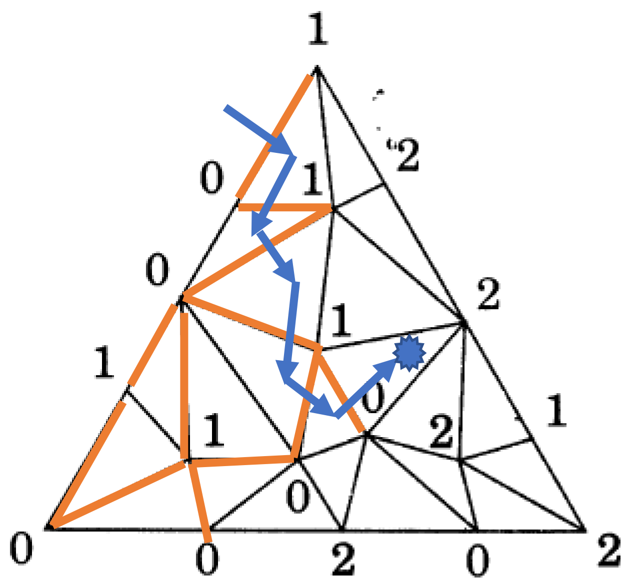
Sperner's Lemma (Intuitions)

- **Theorem** [Sperner's Lemma (1928)]. Every properly colored simplicial subdivision has a fully-colored triangle.
- Build a "door" at those 0-1 edges.



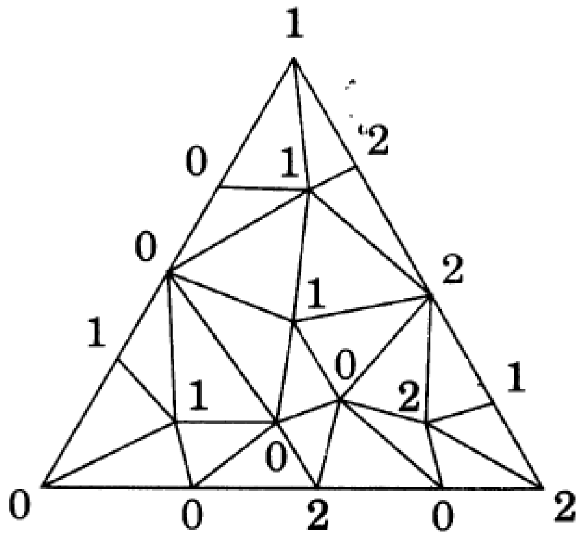
Sperner's Lemma (Intuitions)

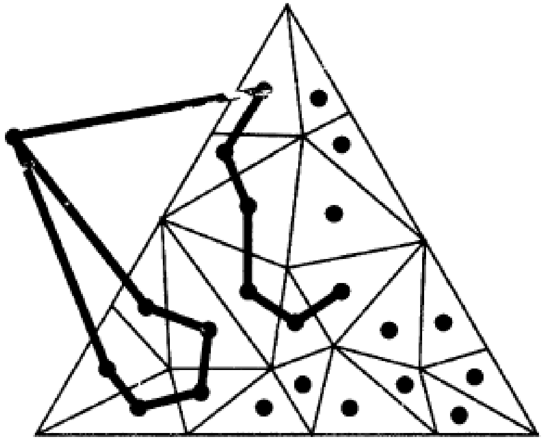
- **Theorem** [Sperner's Lemma (1928)]. Every properly colored simplicial subdivision has a fully-colored triangle.
- Build a "door" at those 0-1 edges.
- If we step in a "dead-end", we find a fully-colored triangle!
- Does there always exist a "dead-end"?



Proof of Sperner's Lemma

- Fix a proper coloring.
- Construct a graph $G = (V, E)$ where
  - V: the cells and one more vertex for the outer region
  - E: two vertices are adjacent if the corresponding two regions/cells share a boundary edge with endpoints colored 0 and 1.
- Let $s \in V$ be the vertex represent the outer region

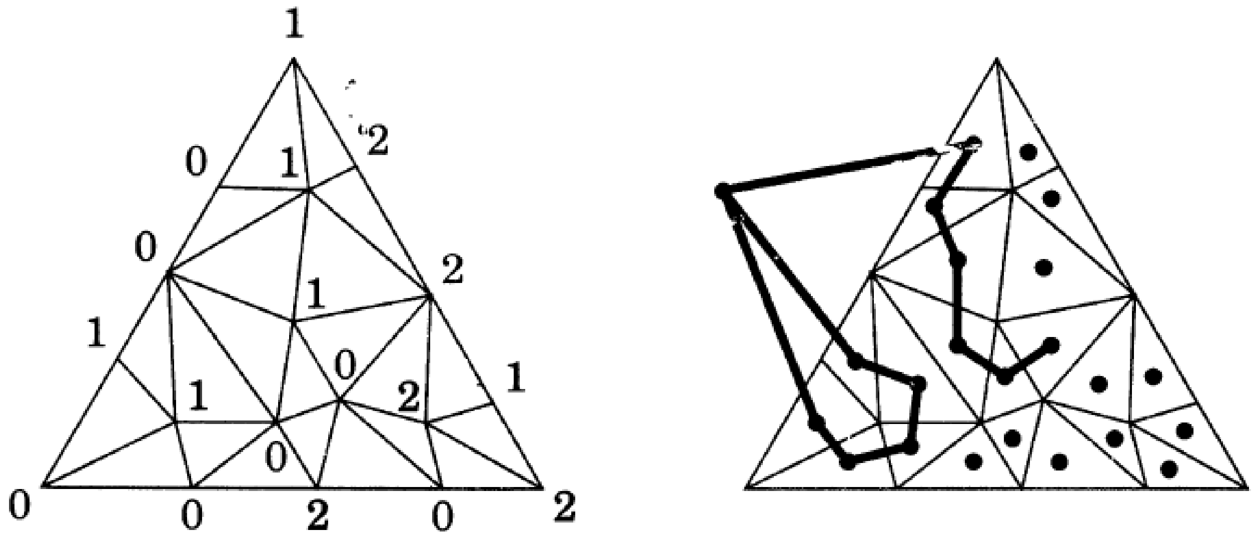




Proof of Sperner's Lemma

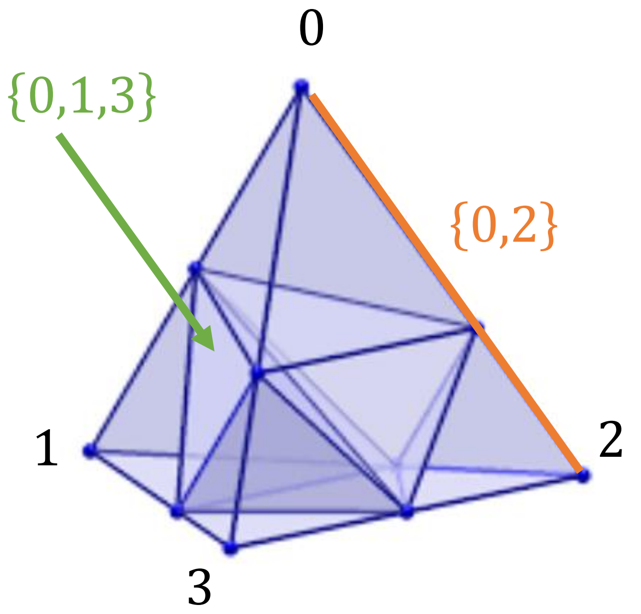
- Except for s, each vertex's degree is at most 2.
- A vertex $u \in V \setminus \{s\}$ with degree 1 corresponds to a fully-colored triangle.
- The degree of s is odd.
- Thus, the number of degree-1 vertices in $V \setminus \{s\}$ is odd.

Theorem. Every properly colored simplicial subdivision has an odd number of fully-colored triangles.



Generalization to Higher Dimensions

- A d-dimensional simplex is the convex hull of $d+1$ points $v_0, v_1, \ldots, v_d \in \mathbb{R}^d$ where the d vectors $v_1 - v_0, v_2 - v_0, \ldots, v_d - v_0$ are linearly independent.
- A simplicial subdivision of a d-dimensional simplex T is a partition of T into cells where
  - Each cell is a d-dimensional simplex
  - Two cells intersect in a common face (a simplex of any lower dimension) or not at all
- Proper coloring: assignments of colors from $\{0,1,\ldots,d\}$ to the nodes such that
  - $d+1$ corners of T have distinct color; assume w.l.o.g. that v_i is colored i
  - Nodes on a k-dimensional subface $v_{i_1} v_{i_2} \cdots v_{i_k}$ of T are colored only with the colors from $\{i_1, i_2, \ldots, i_k\}$.



Sperner's Lemma (Generalized)

- **Theorem** [Sperner's Lemma (1928)]. Every properly colored simplicial subdivision has an odd number of fully-colored cells.
- Proof. Induction on dimensions.

Applications

Brouwer Fixed Point Theorem

Brouwer Fixed Point Theorem

- **Theorem**. Every continuous function f maps from a d-dimensional simplex T to itself has a fixed point x_0 with $f(x_0) = x_0$.

- 2D version: Every continuous function f maps from a triangle T to itself has a fixed point x_0 with $f(x_0) = x_0$.

Proof for 2D Version

- T: convex hull of $v_0, v_1, v_2 \in \mathbb{R}^2$
- Each $v \in T$ can be expressed as $v = a_0 v_0 + a_1 v_1 + a_2 v_2$ with $a_0 + a_1 + a_2 = 1$.
- For each $i \in \{0,1,2\}$, define S_i such that it contains all $v = (a_0, a_1, a_2)$ with $a_i' \leq a_i$ for $v' = (a_0', a_1', a_2')$ where $v' = f(v)$
  - In words, S_i is the set of points whose i-th coordinates are mapped to weakly smaller values by f.
- $T = S_0 \cup S_1 \cup S_2$, and it suffices to prove $S_0 \cap S_1 \cap S_2 \neq \emptyset$.
  - Both are because $a_0 + a_1 + a_2 = 1$.
- Given a simplicial subdivision of T, color each node by i if the node is in S_i.
- Nodes on the edge on the opposite side of corner v_i have the i-th coordinate 0. Thus, these nodes can be colored without using color i.
  - $\Longrightarrow$ We have a proper coloring!
- Sperner's Lemma $\Longrightarrow$ There exists a fully-colored triangle.

Proof for 2D Version

- We need: $S_0 \cap S_1 \cap S_2 \neq \emptyset$
- We have: for every simplicial subdivision of T, there is a cell/triangle xyz where $x \in S_0$, $y \in S_1$, and $z \in S_2$.
- Construct an infinite sequence of simplicial subdivisions where the area of the cells tends to 0.
- For each $t = 1,2,3, \ldots$, the t-th simplicial subdivision contains a cell $x_t y_t z_t$ where $x_t \in S_0$, $y_t \in S_1$, and $z_t \in S_2$.
- f is continuous $\Longrightarrow S_0$ is compact $\Longrightarrow \{x_1, x_2, x_3, \ldots\}$ has a convergent subsequence that converges to some $x \in S_0$.
- The same holds for the other two coordinates. Let $y \in S_1$ and $z \in S_2$ be the limits of the other two convergent subsequences.
- We must have $x = y = z$ as the area of the triangle $x_t y_t z_t$ tends to 0 as $t \to \infty$.

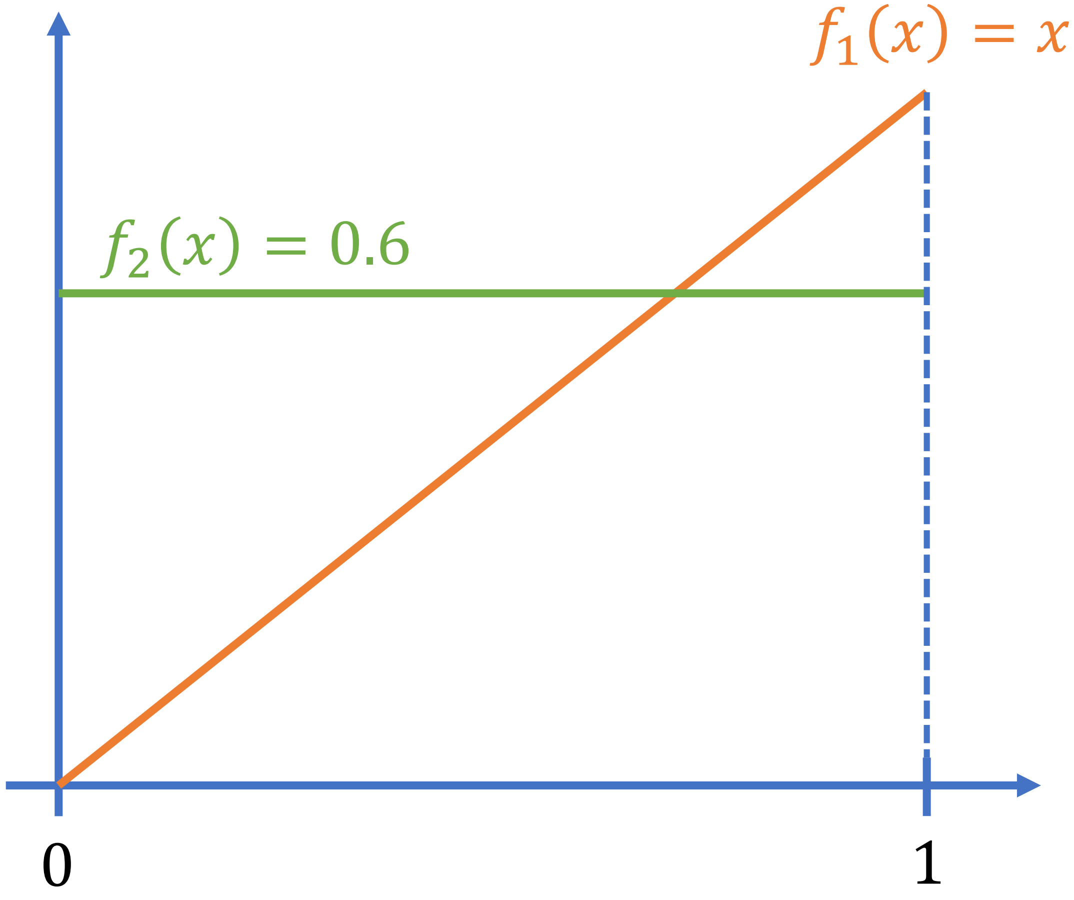
Applications

Envy-Free Cake Cutting

Cake-Cutting [Steinhaus 1948]

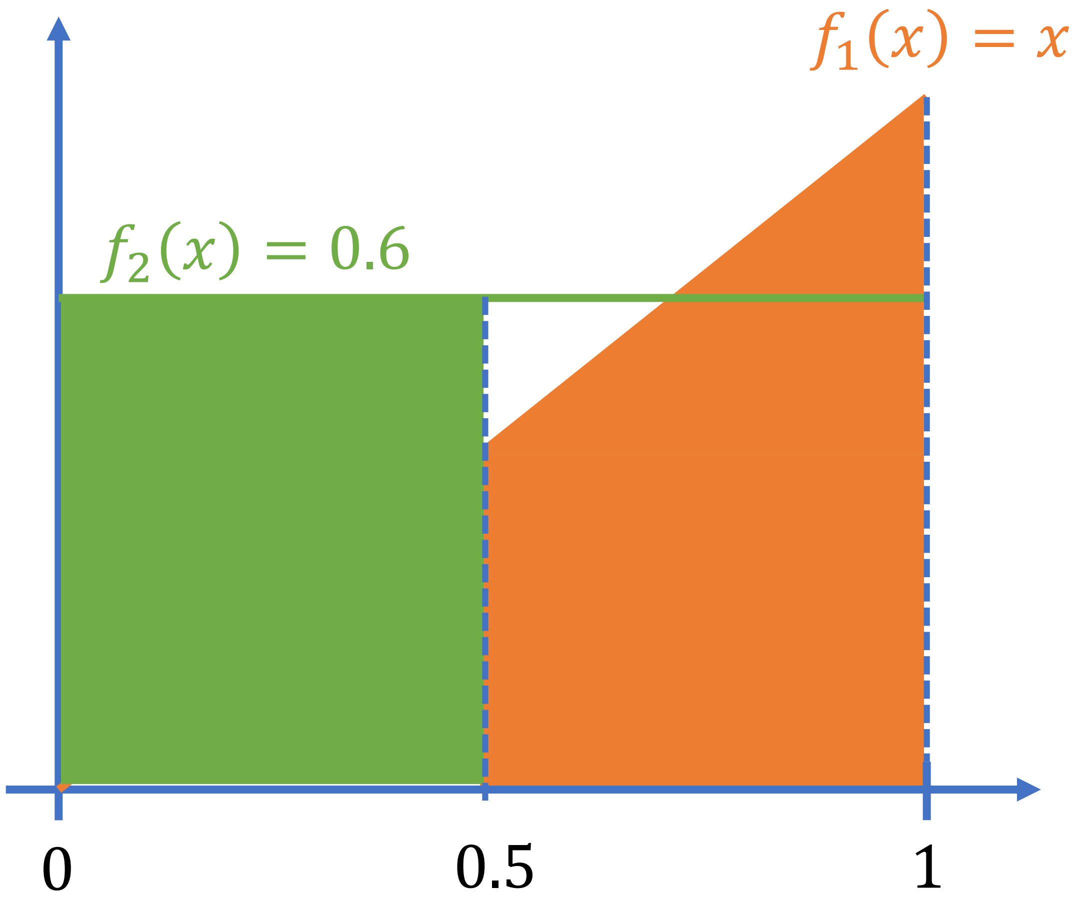
- Cake: interval $[0,1]$, to be allocated to n agents
- Allocation: $(A_1, A_2, \ldots, A_n)$
  - A_i: the piece allocated to agent i
  - Each A_i is an interval
  - A_i and A_j can only intersect at a single point
- Value density function for agent i: $f_i: [0,1] \rightarrow \mathbb{R}_{\geq 0}$.
  - Agent i values an interval $[a,b]$ by $v_i([a,b]) = \int_a^b f_i(x)dx$
  - For simplicity, assume each f_i is continuous
- Envy-Freeness: an allocation $(A_1, A_2, \ldots, A_n)$ is envy-free with respect to $(f_1, \ldots, f_n)$ if $\forall i, j: v_i(A_i) \geq v_i(A_j)$
  - In words, each agent i weakly prefer his/her own piece than any other's.

Example



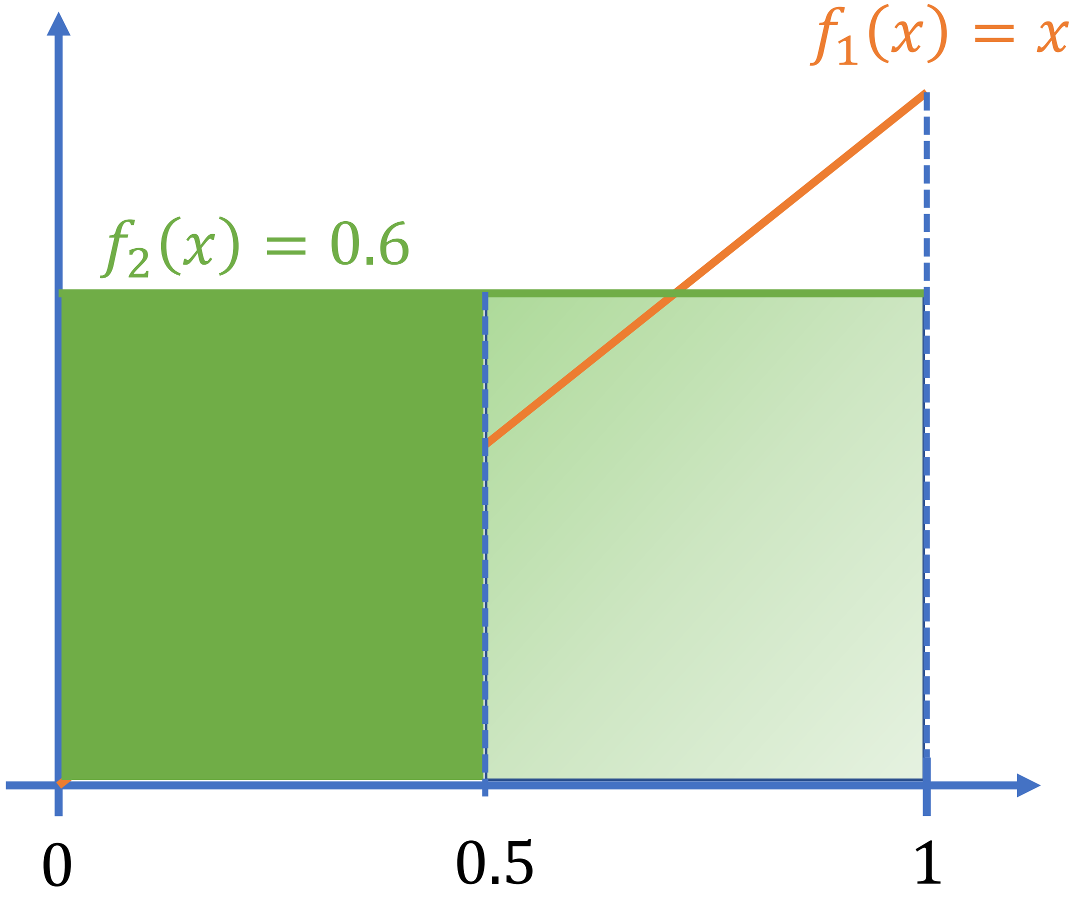
- Value density functions:
  - Agent 1: $f_1(x) = x$
  - Agent 2: $f_2(x) = 0.6$

Example



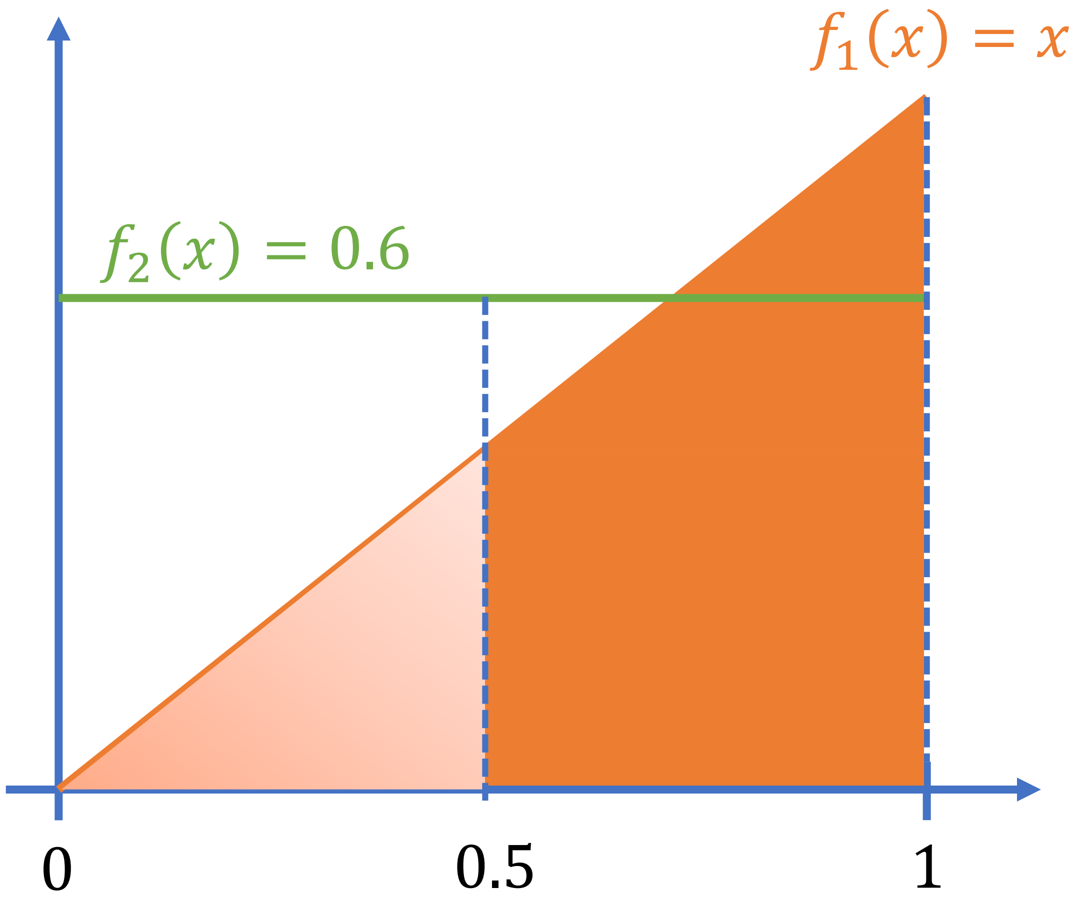
- Value density functions:
  - Agent 1: $f_1(x) = x$
  - Agent 2: $f_2(x) = 0.6$
- $(A_1 = [0,0.5], A_2 = [0.5,1])$ is an envy-free allocation

Example



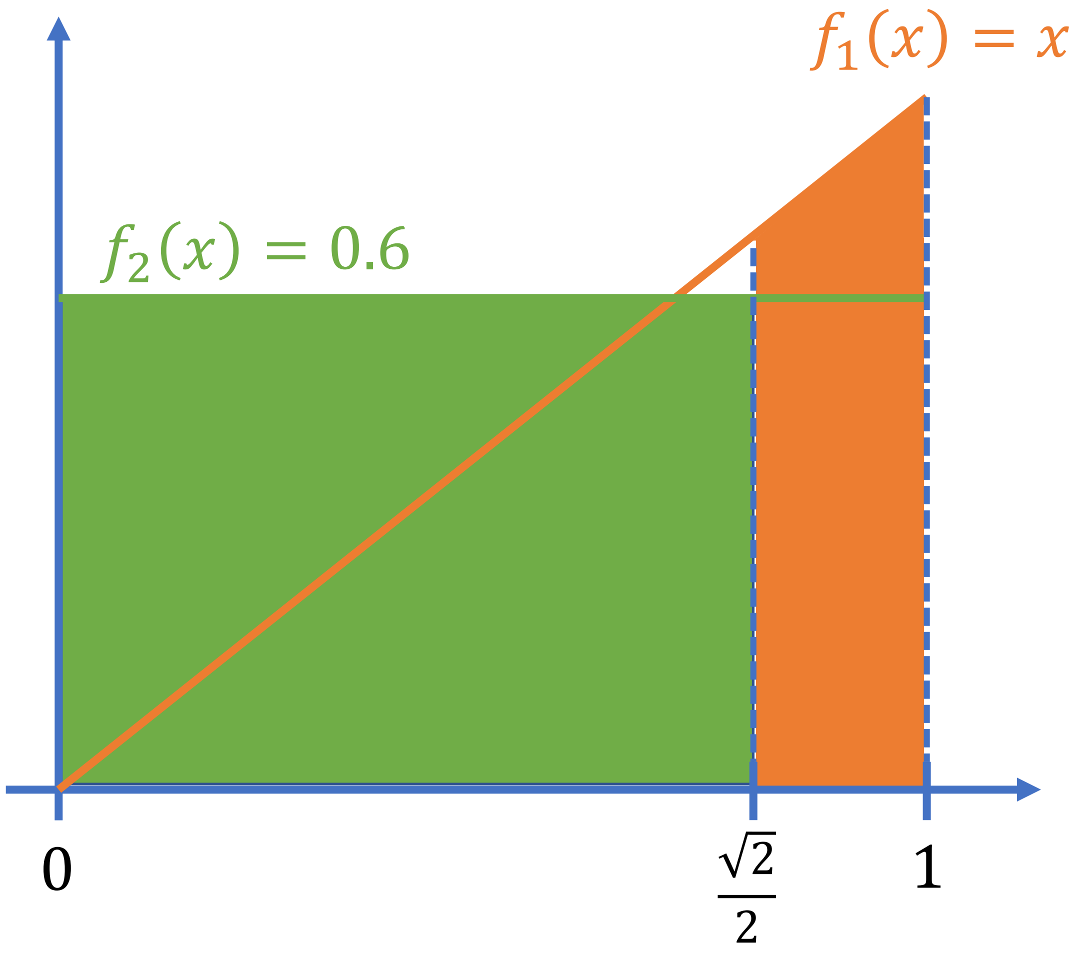
- Value density functions:
  - Agent 1: $f_1(x) = x$
  - Agent 2: $f_2(x) = 0.6$
- $(A_1 = [0,0.5], A_2 = [0.5,1])$ is an envy-free allocation:
  - $v_1(A_1) = 0.3 \geq v_1(A_2) = 0.3$

Example



- Value density functions:
  - Agent 1: $f_1(x) = x$
  - Agent 2: $f_2(x) = 0.6$
- $(A_1 = [0,0.5], A_2 = [0.5,1])$ is an envy-free allocation:
  - $v_1(A_1) = 0.3 \geq v_1(A_2) = 0.3$
  - $v_2(A_2) = \frac{3}{8} \geq v_2(A_1) = \frac{1}{8}$

Example



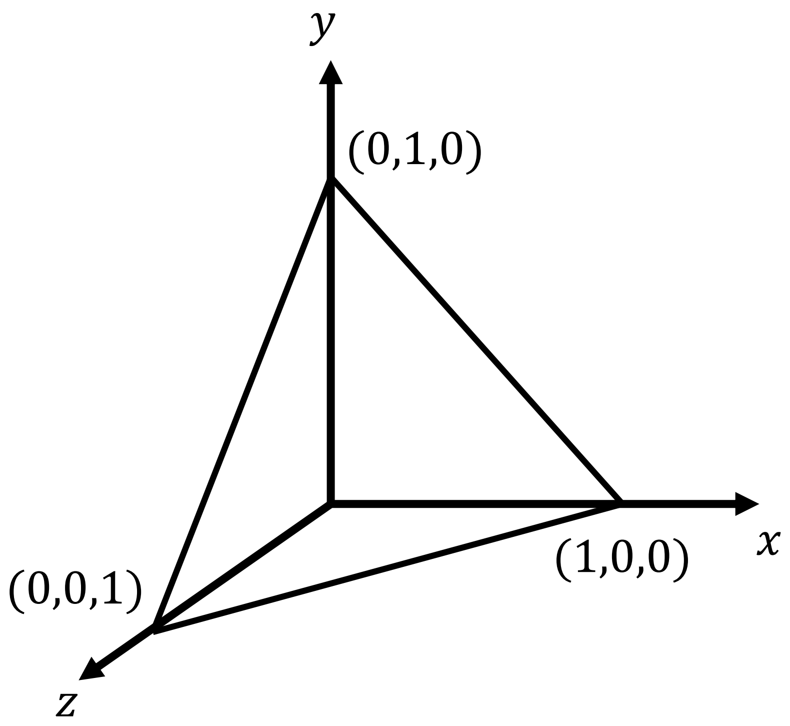
- Value density functions:
  - Agent 1: $f_1(x) = x$
  - Agent 2: $f_2(x) = 0.6$
- Similarly, $\left(A_1 = \left[0, \frac{\sqrt{2}}{2}\right], A_2 = \left[\frac{\sqrt{2}}{2}, 1\right]\right)$ is also an envy-free allocation:
  - $v_1(A_1) = \frac{3\sqrt{2}}{10} \geq v_1(A_2) = \frac{6-3\sqrt{2}}{10}$
  - $v_2(A_2) = \frac{1}{4} \geq v_2(A_1) = \frac{1}{4}$
- In fact, every cut between 0.5 and $\frac{\sqrt{2}}{2}$ yields an envy-free allocation.

Envy-Free Cake-Cutting

- **Theorem [Su 1999]**. For any value density function profile $(f_1, \dots, f_n)$, an envy-free allocation exists.

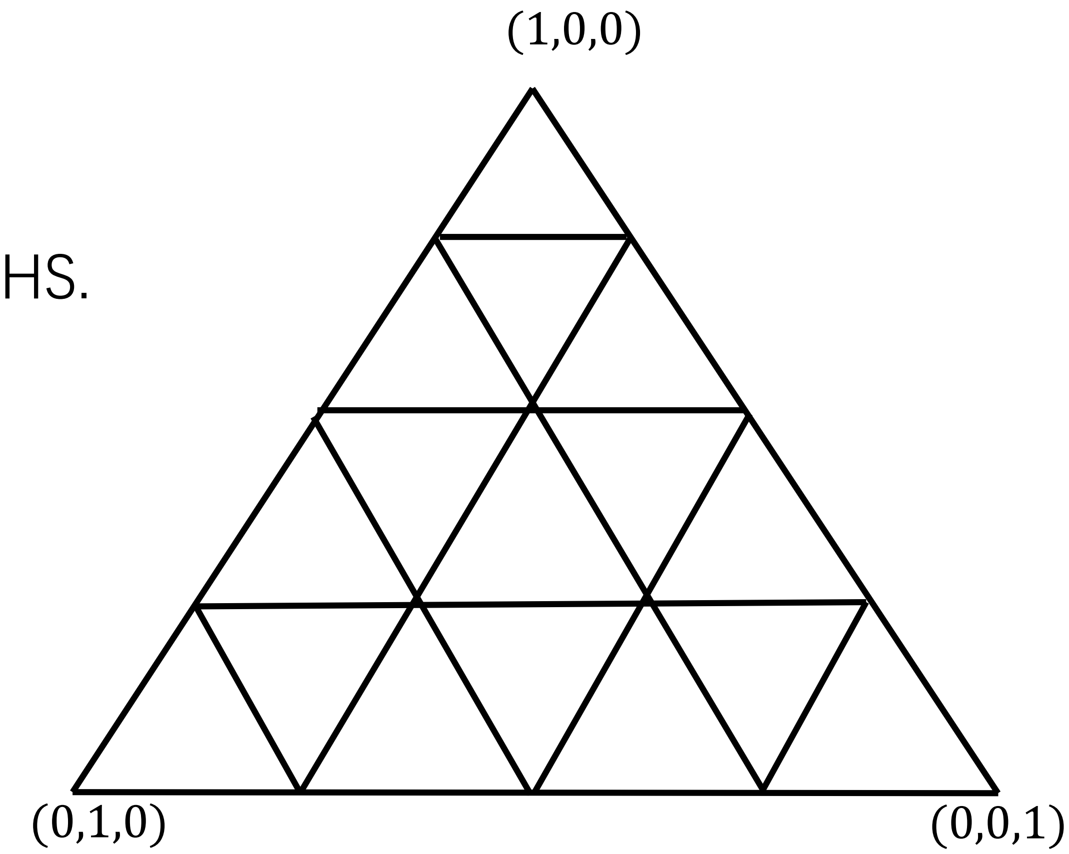
Proof (for 3 agents case)

- $\Delta_3 := \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 1\}$
  - The set of all possible partitions
  - x_1, x_2, x_3 are the length of the left, middle, right segments.
- Let each agent $\boldsymbol{i}$ color each $\boldsymbol{x} \in \Delta_3$ from the color-set {left, middle, right} indicating his/her favorite piece (break tie arbitrary).
- If three agents color a point $\boldsymbol{x} \in \Delta_3$ with three different colors, we find an envy-free allocation.
- It remains to show such a point exist!



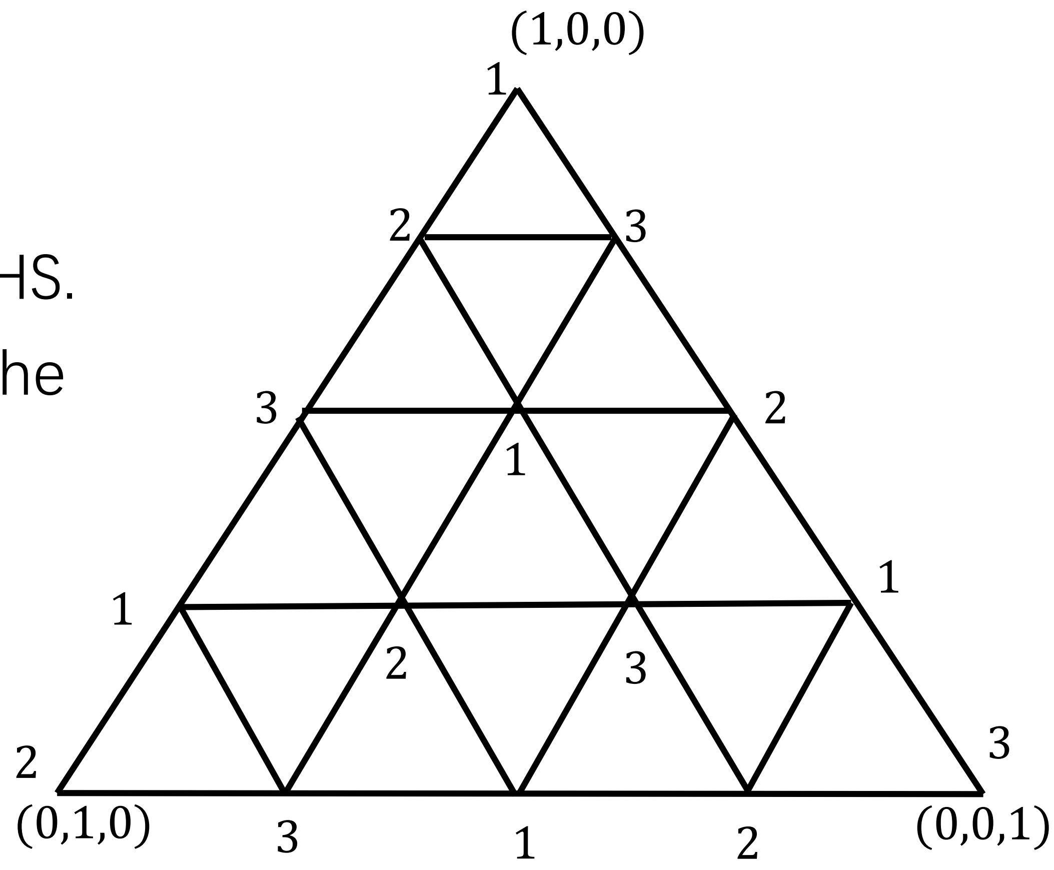
Proof (for 3 agents case)

- Consider the simplicial subdivision of Δ_3 on RHS.



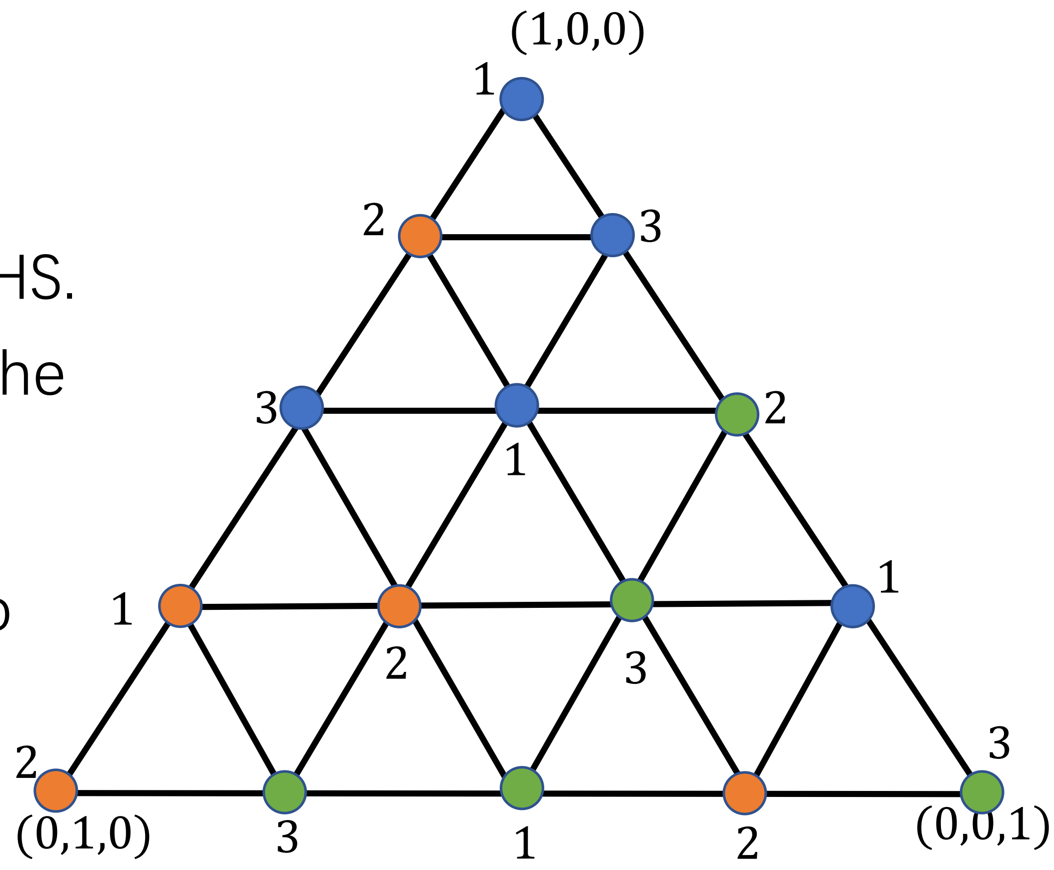
Proof (for 3 agents case)

- Consider the simplicial subdivision of Δ_3 on RHS.
- Before coloring, "label" each node by one of the three agents in a way shown RHS.
  - so that each cell is "fully-labeled"



Proof (for 3 agents case)

- Consider the simplicial subdivision of Δ_3 on RHS.
- Before coloring, "label" each node by one of the three agents in a way shown RHS.
  - so that each cell is "fully-labeled"
- For each node, let the agent corresponding to the label to color it from the color-set {left, middle, right}.
- This is a proper coloring (Why?)
- By Sperner's Lemma, there is a fully-colored cell.
- If the cell is small enough to be considered as a single point, we have an envy-free allocation!



Proof (for 3 agents case)

- We need: a point in Δ_3 where three agents color differently
- We have: for every "regular" simplicial subdivision, there is a cell/triangle xyz where agent 1's color on x_1, agent 2's color on y, and agent 3's color on z are all distinct.
- Construct an infinite sequence of "regular" simplicial subdivisions where the area of the cells tends to 0. Let $x_t y_t z_t$ be the fully-colored cell at the t-th subdivision.
- Let $\{a_1, a_2, \ldots\}$ be an infinite sequence where agent 1's color on $x_{a_1}, x_{a_2}, \ldots$ are the same.
  - Since $\{x_1, x_2, \ldots\}$ is infinite, at least one of the three colors is used by agent 1 for infinitely many times.
  - Assume w.l.o.g. agent 1's color on $x_{a_1}, x_{a_2}, \ldots$ is left.
- Let $\{b_1, b_2, \ldots\} \subseteq \{a_1, a_2, \ldots\}$ be an infinite subsequence where agent 2's color on $y_{b_1}, y_{b_2}, \ldots$ are the same.
  - Again, since $\{a_1, a_2, \ldots\}$ is infinite, at least one color is used by agent 2 for infinitely many times.
  - Moreover, this color cannot be left, due to the fully-colored property.
  - Assume w.l.o.g. agent 2's color on $y_{b_1}, y_{b_2}, \ldots$ is middle.
- Then, agent 3's color on $z_{b_1}, z_{b_2}, \ldots$ has to be right.

Proof (for 3 agents case)

- We need: a point in Δ_3 where three agents color differently
- We have: for every "regular" simplicial subdivision, there is a cell/triangle xyz where agent 1's color on x_1, agent 2's color on y, and agent 3's color on z are all distinct.
- We further have: an index sequence $b_1, b_2, \dots$ where
  - Agent 1 colors $x_{b_1}, x_{b_2}, \dots$ left
  - Agent 2 colors $y_{b_1}, y_{b_2}, \dots$ middle
  - Agent 3 colors $z_{b_1}, z_{b_2}, \dots$ right
- The set of points where agent 1 colors left is compact.
- We can find a subsequence of $x_{b_1}, x_{b_2}, \dots$ that converges to some x where agent 1 colors left.
- Similarly, a subsequence of $y_{b_1}, y_{b_2}, \dots$ converges to some y where agent 2 colors middle.
- The same for z where agent 3 colors right.
- Finally, x, y, z must be the same point, since the area of the cell tends to $\mathbf{0}$.

Computational Complexity Aspect

The Complexity Class PPAD

Computational Complexity

- We have seen:
  - There exists a fully-colored cell in a proper coloring for the nodes of a simplicial subdivisions.
  - There exists a fixed point in a continuous function mapping from a simplex to itself.
  - There exists an envy-free cake cutting allocation.
- What if we take a computational complexity aspect?
  - Can we find such an object in polynomial time?
  - for the “discrete versions” of these problems…

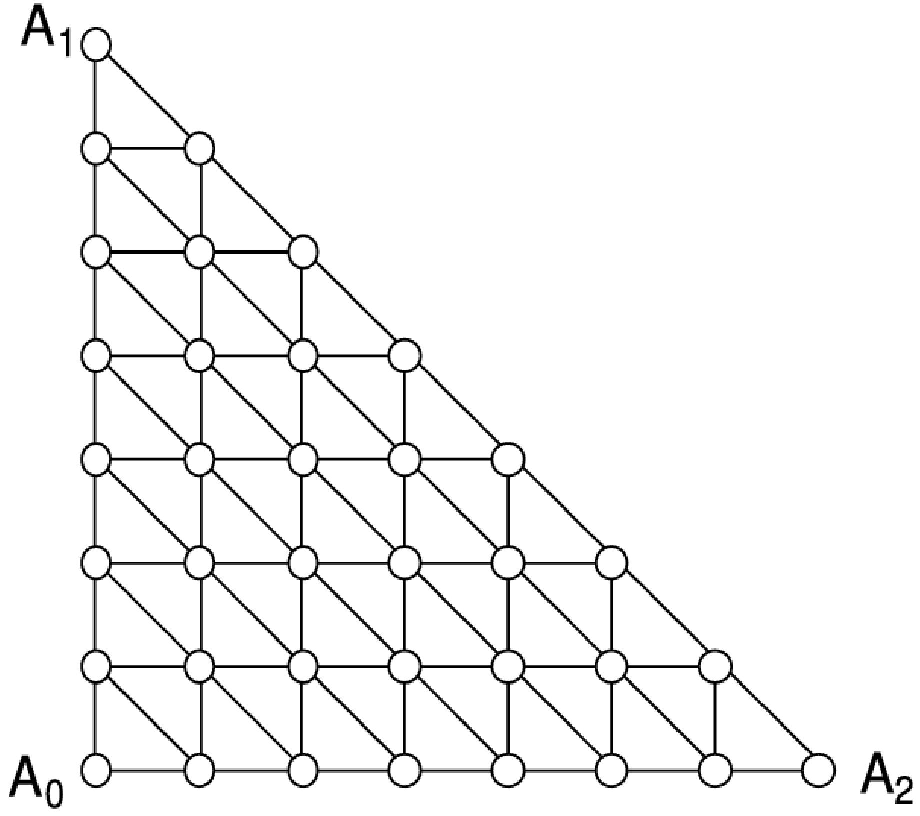
2D-SPERNER

Given:

- Set of lattice points $S = \{(x, y) \in \mathbb{Z}^2 : x + y \leq n\}$
- A polynomial-time computable function $f: S \to \{0,1,2\}$ that gives a proper coloring.

Find:

- A fully-colored triangle.



Approximate Envy-Free Cake Cutting

Given:

- a cake-cutting instance where each query

$$v_i([a,b]) = \int_a^b f_i(x)dx$$

can be computed in polynomial time

- a parameter ϵ

Find:

- an ϵ-envy-free allocation $(A_1, \ldots, A_n)$:

$$\forall i, j: v_i(A_i) \geq v_i(A_j) - \epsilon$$

2D-BROUWER

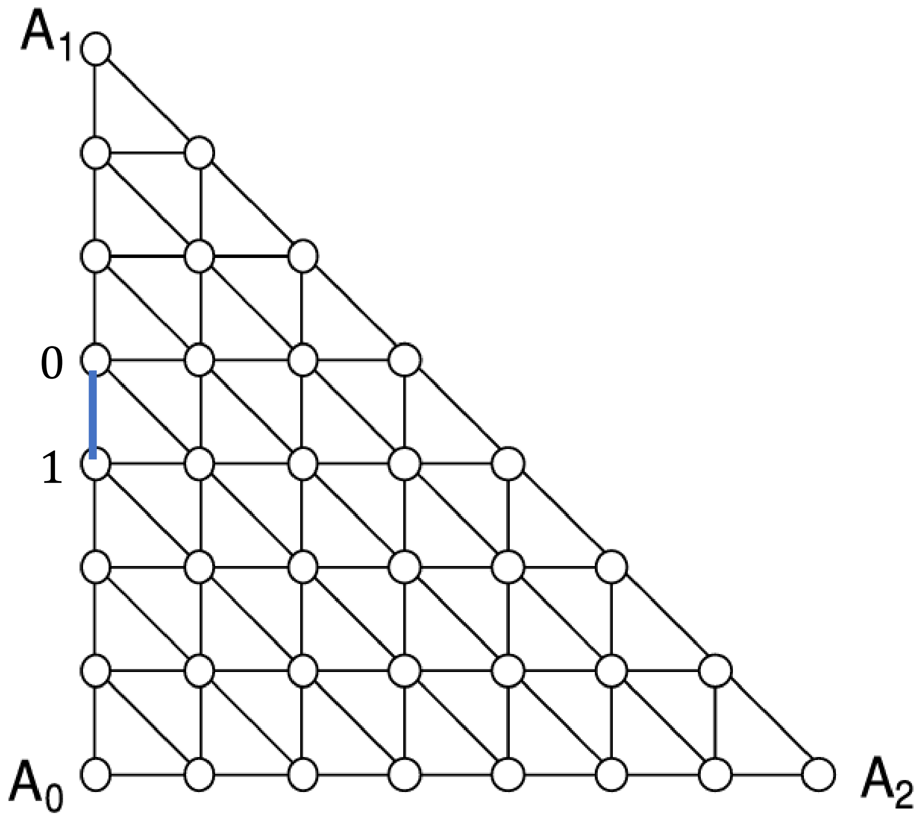
- Consider a unit 2D square subdivided into n^2 equal subsquares, each of size $\epsilon = 1/n$
- a function ϕ defined only on centers of subsquares : for each center x, $\phi(x)$ can only take 3 values: $x + \delta_i$, $i = 0,1,2$
  - $\delta_1 = (\epsilon, 0)$, $\delta_2 = (0, \epsilon)$
  - $\delta_0 = (-\epsilon, -\epsilon)$
- and $\phi(x)$ does not go outside the boundary…
- A fixed point: a subsquare cornet point such that, among its eight adjacent subsquares, all 3 possible displacements δ_i, $i = 0,1,2$ are presented
- There always exists fixed points (Sperner’s Lemma)

Computational Complexity?

- We have seen that
  - 2D-BROUWER polynomial-time reduces to 2D-SPERNER
  - Approximate Envy-Free Cake Cutting polynomial-time reduces to 2D-SPERNER
- But what is the computational complexity for 2D-SPERNER?

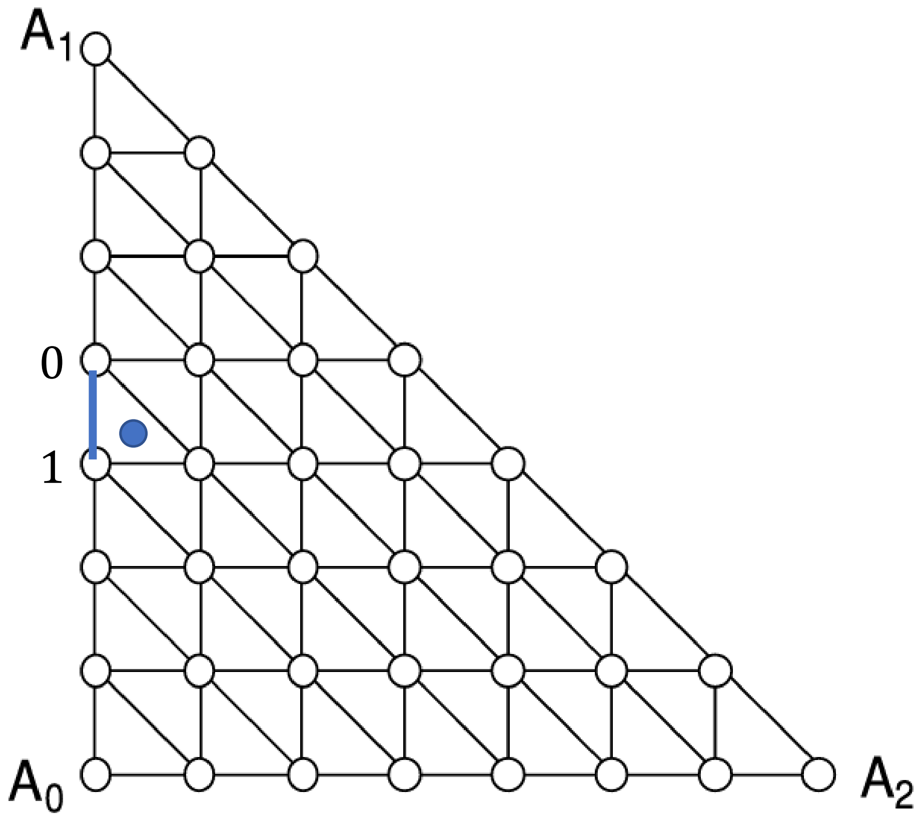
An algorithm for 2D-SPERNER

- Find a 0-1 edge on the side A_0A_1.



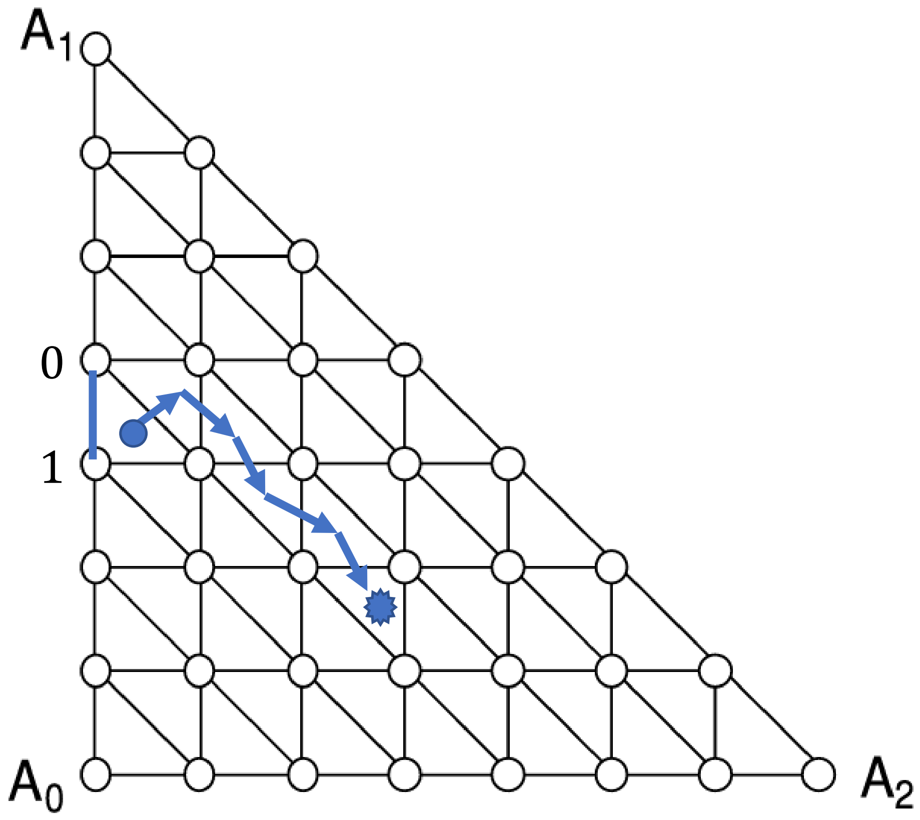
An algorithm for 2D-SPERNER

- Find a 0-1 edge on the side A_0A_1.
- This locates a triangle with at least one “door”.



An algorithm for 2D-SPERNER

- Find a 0-1 edge on the side A_0A_1.
- This locates a triangle with at least one "door".
- Move along the "path" and search for "the end of the line".
- Time Complexity: $O\left(n^2\right)$
- Not polynomial time!
  - The input length is only $\Theta(\log n)$
- It doesn't seem that we can do better without some assumptions on the coloring function f
  - We need to "teleport" from one point to another



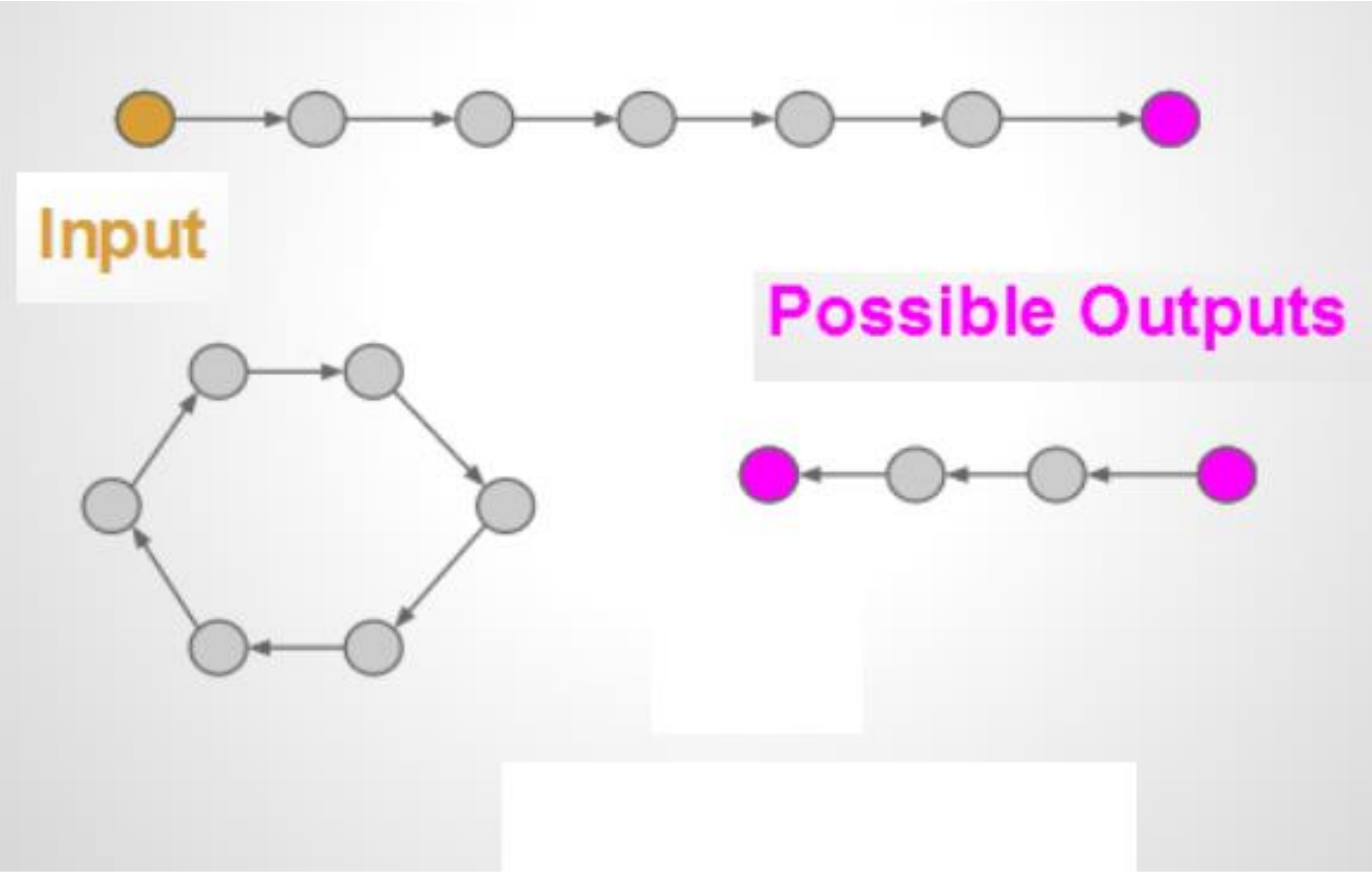
The Complexity Class PPAD

- Proposed by Papadimitriou in 1990
- Name short for "**P**olynomial **P**arity **A**rgument for **D**irected graphs"
- **Definition**: every search problem that reduces to END_OF_THE_LINE

END_OF_THE_LINE Problem

- $G = (V, E)$ is a directed graph of exponential size, every vertex having at most one predecessor and one successor
- For each $v \in V$, poly-time computable $f(v)$ returns its predecessor and successor
- Given function f and a source $s \in V$, find a sink or another sink

A Typical $PPAD$ problem

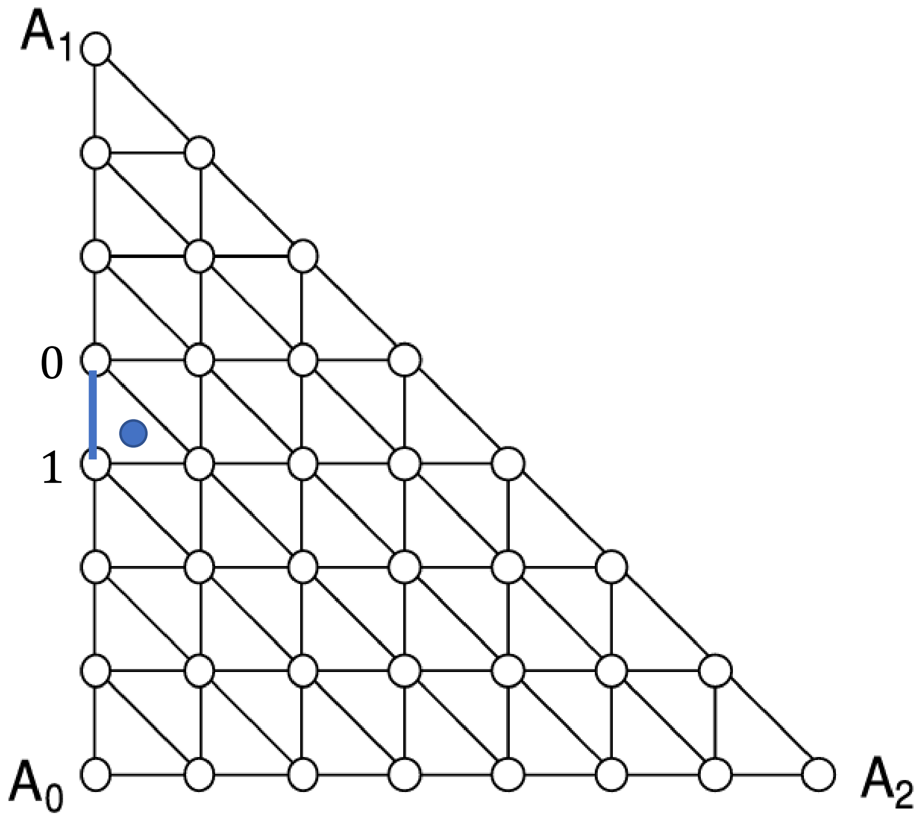


2D-SPERNER is in PPAD

- Find a 0-1 edge on the side A_0A_1.
- This locates a triangle with at least one “door”.
- Move along the “path” and search for “the end of the line”.

2D-SPERNER ∈ PPAD

- An initial source can be found by binary search in polynomial time.
- The predecessor and successor can be computed in polynomial time by computing the colors of the three vertices of the cell.



Problems in PPAD

- SPERNER
- BROUWER
- Approximate Envy-Free Cake Cutting
- Finding a Nash equilibrium

PPAD-Completeness

- **Theorem**. SPERNER is PPAD-complete. [Papadimitriou 1990]
- **Theorem**. BROUWER is PPAD-complete. [Papadimitriou 1990] [Chen&Deng, 2009]
- **Theorem**. Approximate envy-free cake cutting is PPAD-complete. [Deng, Qi, Saberi, 2009]
- **Theorem**. Finding a Nash equilibrium is PPAD-complete. [Daskalakis, Goldberg, Papadimitriou, 2006] [Chen, Deng, Teng, 2007]

PPAD vs NP?

PPAD vs NP

- PPAD: search problem
- NP: decision problem
- To compare them, we need “search version” of NP.

A typical NP problem, which is also well-known to be NP-complete.

[SAT] Given a Boolean formula ϕ in conjunctive normal form, decide if it has a satisfying assignment.

$$\phi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_4 \vee x_5 \vee \neg x_6) \wedge (\neg x_1 \vee \neg x_4 \vee x_6)$$

FNP (Functional NP)

- "searching version" of NP: for a "yes" instance, a solution is expected
- A binary relation $P(x, y)$, where y is at most polynomially longer than x, is in FNP if and only if there is a deterministic polynomial time algorithm that can determine whether $P(x, y)$ holds given both x and y.

[FSAT] Given a Boolean formula ϕ in conjunctive normal form,

- output a satisfying assignment if there exists one,
- output "no" otherwise.

Relationship between PPAD and (F)NP

- If END_OF_THE_LINE is FNP-complete, then $NP = coNP$.

Proof. Assume there is a reduction from SAT to END_OF_THE_LINE:

- algorithm A mapping every SAT formula ϕ to a END_OF_THE_LINE instance $A(\phi)$
- algorithm B mapping every sink t of $A(\phi)$ to
  - a satisfying assignment $B(t)$ of ϕ, if exist;
  - the string “no”, otherwise

Then t is also a certificate for a unsatisfiable ϕ:

- compute $A(\phi)$ and verify if t is a sink
- verify if $B(t)$ maps to “no”

What's really going on?

- If END_OF_THE_LINE is FNP-complete, then $NP = coNP$.

- A mismatch:
  - FNP-complete problem (like FSAT): an instance may be "yes" or "no"
  - $PPAD$: all instances are "yes" instances (Nash's Theorem)

Take-Home Message

- PPAD-complete search problem does not seem to admit polynomial-time algorithms.
- But it is easier than NP-complete problems.

Reference

[Su 1999] Rental Harmony: Sperner's Lemma in Fair Division

[Papadimitriou 1990] On Graph-theoretic Lemmata and Complexity Classes

[Chen&Deng 2009] On the Complexity of 2D Discrete Fixed Point Problem

[Deng, Qi, Saberi, 2009] On the Complexity of Envy-Free Cake Cutting

[Daskalakis, Goldberg, Papadimitriou, 2006] The Complexity of Computing a Nash Equilibrium

[Chen, Deng, Teng, 2007] Settling the Complexity of Computing Two-Player Nash Equilibria