6.896 Topics in Algorithmic Game Theory

Lecture 7

Lecturer: Constantinos Daskalakis *Scribes: David Chen, Pablo Azar*

NOTE: The content of these notes has not been formally reviewed by the lecturer. It is recommended that they are read critically.

Continuing from the previous lecture, we finished looking at the proof of Sperner's Lemma in the multidimensional case. In this lecture, we adopt a more computational view of the lemma. We then define computational versions of `SPERNER`, `BROUWER`, and `NASH`, the complexity class FNP, and finish with a definition of the complexity class PPAD.

1 Sperner's Lemma

Recall from the previous lecture that Sperner's Lemma for many dimensions was proved:

Theorem 1 (Sperner 1928). *Suppose the vertices of the canonical simplicization of the hypercube $[0,1]^n$ are colored with colors $0, 1, \ldots, n$ so that the following property is satisfied by the coloring on the boundary:*

- (P_n)*: For all $i \in \{1, \ldots, n\}$, none of the vertices on the face $x_i = 0$ uses color i; moreover, color 0 is not used by any vertex on a face $x_i = 1$ for some $i \in \{1, \ldots, n\}$.*

Then, there exists a panchromatic simplex in the simplicization. In fact, there is an odd number of such simplices.

Proof and discussion of this theorem was given in the previous lecture.

2 Computational Problems

With the general version of Sperner's Lemma shown, we are now ready to move toward the computational view of these problems.

2.1 `SPERNER`

The coloring of an n-cube can be viewed in the following way. Say that the coordinate of a point in each direction can be represented by m bits; that is, the cube is broken up into 2^m pieces along each direction. Then, a point on the cube is defined by n coordinates of m bits, and a coloring is defined as a circuit C such that

$$C : \{0,1\}^{mn} \to \{0, 1 \ldots, n\}$$

That is, C operates on n coordinates $x_1, \ldots, x_n$ and outputs a color. We can see how having such a circuit is equivalent to a unique coloring of the cube: if we want to find the color at a vertex, just give the circuit the coordinates of the vertex, and use the output of the circuit. The Sperner problem then becomes the problem of finding a panchromatic simplex when given a coloring circuit C.

Notice that in the computational perspective, we meet additional complications. For one, it is fine to talk about what to do if we know a coloring fits the Sperner property. However, given a circuit C as above, we have no guarantee that the coloring it produces does fit the Sperner property! Indeed, as there are $\Omega(n2^{m(n-1)})$ different vertices on the facets of the cube, we cannot feasibly check if a circuit does in fact have the Sperner property! Any computational analogy of the Sperner problem must take this into account.

Definition 1 (SPERNER 1)**.** *Given a coloring circuit C, either find a panchromatic triangle, or find a point on the boundary that violates the Sperner coloring property.*

Definition 2 (SPERNER 2)**.** *Given a coloring circuit C, find a panchromatic triangle in the coloring produced by another circuit C' that:*

- *Agrees with C inside the hypercube.*
- *Produces the "envelope coloring" at the boundary.*

Note that both problems are *total problems*, in that one can find a solution for all inputs.

2.2 Search Problems, FNP and its Reductions

A *search problem* L is a language consisting of pairs (x, y). The first element x is an instance of the problem, and the second element y is a potential solution to the problem. Formally, a search problem is defined by a relation $R_L(x, y)$ such that

$$R_L(x, y) = 1$$

if and only if y is a solution to the instance x. A search problem is called *total* if for all x there exists y such that $R(x, y) = 1$.

A search problem L is in the class FNP if and only if there exists an efficient algorithm $A_L(x, y)$ and a polynomial function $p_L(\cdot)$ such that

1. if $A_L(x, z) = 1$, then $R_L(x, z) = 1$,
2. if there exists y such that $R_L(x, y) = 1$, then there exists z with $|z| \leq p_L(|x|)$ such that $A_L(x, z) = 1$

Informally, a search problem is in FNP if all instances x of the problem that have solutions, also have solutions that are polynomially small and efficiently verifiable. SPERNER is in FNP. Given a coloring circuit C and a panchromatic triangle induced by this coloring circuit, we can easily verify that the triangle is panchromatic. If we are given the circuit C and a point at the boundary that violates the Sperner coloring property, we can easily verify that the coloring property is violated.

A search problem L in FNP, associated with $A_L(x, y)$ and p_L, is *polynomial time reducible* to another search problem $L' \in$ FNP associated with $A_{L'}(x, y)$ if and only if there exist efficiently computable functions f, g such that [1]

1. if x is an input to L, then $f(x)$ is an input to L',
2. if $A_{L'}(f(x), y) = 1$ then $A_L(x, g(y)) = 1$,
3. if $R_{L'}(f(x), y) = 0$ for all y, then $R_L(x, y) = 0$ for all y.

A search problem L is *FNP-complete* if and only if $L \in$FNP and for all other L' in FNP, L' is polynomial-time reducible to L. For example SAT is FNP-complete.

2.3 BROUWER

We want to turn finding a Brouwer fixed point of a function into a combinatorial problem. Note that it is not immediately clear that such a thing is possible. For example, given a continuous function $f(x)$ on a compact, convex set, it could have a unique Brouwer fixed point that is irrational. Thus, the fixed point would be impossible to compute in finite time. To define a combinatorial problem, we introduce *approximation.* Let us say that a point x is an ϵ-approximate fixed point of f if $|f(x) - x| < \epsilon$. [2]

[1] We will see shortly that in order to compare total search problems with non-total search problems we will relax the last condition.

[2] There is a stronger notion of approximate fixed point, which we won't use in this class. In particular, we could say that x is an ϵ-approximate fixed point of f, if there exists y such that

$$|x - y| < \epsilon \text{ and } y = f(y).$$

For well-behaved functions, e.g. those having a polynomially bounded modulus of continuity the stronger notion of approximation always implies the weaker notion. For more information on the stronger notion and its computational nature, look at [1].

Informally, the Brouwer problem consists of finding an approximate fixed point of a continuous function $f : [0,1]^n \to [0,1]^n$. To guarantee that for a given approximation ϵ we can express the approximate fixed point with polynomial precision, we require that the continuous function have a well behaved modulus of continuity. For example, we can require that the function f be Lipschitz. That is, $|f(x) - f(y)| \le c|x - y|$ for some constant c, for all inputs x, y.

Formally, we can define `BROUWER` as a search problem in FNP. An instance of the problem is given by

1. an algorithm $A(x_1, ..., x_n)$ that claims to evaluate a continuous function $f : [0,1]^n \to [0,1]^n$;
2. an approximation parameter ϵ;
3. a Lipschitz constant c that f is claimed to satisfy.

A solution of such an instance is a pair of points $x, y \in [0,1]^n$ that satisfies any of the following conditions:

1. $|f(x) - x| \le \epsilon$
2. $|f(x) - f(y)| > c|x - y|$
3. $A(x)$ lies outside of $[0,1]^n$.

Remark 1. *The norms used for the approximation requirement and the Lipschitz condition in the definition of the problem are flexible.*

There are three different ways in which a `BROUWER` instance can have a solution. The reason behind this is that we want to guarantee that *any* instance of `BROUWER` has a solution, no matter what A, ϵ and c are given in the input. That is, `BROUWER` is a total problem.

Claim 1. `BROUWER` *is a total problem in FNP.*

Proof: We reduce the problem to `SPERNER`, which is total and in FNP. Start by defining a sufficiently fine canonical simplicization of $[0,1]^n$, using cells of size 2^{-m}. We will decide m later.

Use the direction of $f(x) - x$ to define a coloring of the vertices of the simplicization with $(n+1)$ colors. In particular, color i is allowed if $(f(x) - x)_i \le 0$. Color 0 is allowed if $(f(x) - x)_i \ge 0$ for all i. Ties are broken to avoid violating the coloring requirements of Sperner, if at all possible (if the value that A outputs at a point on the boundary is directed outside of the hypercube this may be impossible to avoid, but this is OK for our reduction).

Hence, we obtain a valid instance of `SPERNER`. Find a solution of this instance. We show how to obtain a solution to our original `BROUWER` instance.

- If our solution to `SPERNER` is a point x on the boundary violating the Sperner coloring requirements, then it must be that $A(x) \notin [0,1]^n$.
- If a panchromatic simplex is returned, let $z^0, z^1, ..., z^n$ be the vertices colored $0, 1, ..., n$. We can argue (as we did in the 2-d case in lecture 5), that

$$|(f(z^0) - z^0)_i| \le |(f(z^0) - f(z^i))_i| + |(z^0 - z^i)_i|$$

  for all $i \in \{1, ..., n\}$.
- If the Lipschitz condition is satisfied for all pairs of points (z^0, z^i), the above implies (in the infinity norm) that $|f(z^0) - z^0|_\infty < (c+1)2^{-m}$. We conclude that z^0 is an approximate fixed point of f when $(c+1)2^{-m} \le \epsilon$, assuming that our m was chosen appropriately in the beginning of the reduction. We can obtain a similar conclusion for other norms.
- If one of the pairs (z^0, z^i) violates the Lipschitz condition, this pair can be easily identified and returned as a solution to the `BROUWER` instance.

$\square$

2.4 NASH

We now define the FNP problem `NASH`. Unlike `SPERNER` and `BROUWER`, we do not need many conditions to ensure that this is a total problem. The reason is that Nash's theorem guarantees that any game has an equilibrium.

An input to `NASH` is described by

1. the number of players n,
2. an enumeration of the strategy set S_p for every player p,
3. a utility function $u_p : \times_p S_p \to \mathbb{R}$ for each player p.

The `NASH` problem is the task of finding an ϵ-Nash equilibrium of the game specified in the input. Recall that in an ϵ-Nash equilibrium $\sigma = (\sigma_1, ..., \sigma_n)$, every pure strategy in the support of σ_i gives a payoff that is ϵ-close to optimum given the strategy profile σ_{-i} of the other players.

Remark 2. *We use approximation in the definition of the problem for a reason. Already in 1951, Nash showed a three-player game whose unique equilibrium was irrational. The single exception to such counterexamples are two-player games that always have rational equilibria as we will establish in following lectures. Thus, for two player games we can define an analogue of the* `NASH` *problem that is exact.*

Remark 3. *As we mentioned before, the problem is total. Nash's theorem guarantees that any game has a Nash equilibrium, hence an ϵ-Nash equilibrium as well.*

Remark 4. *We could define our problem in terms of an ϵ-approximate Nash equilibrium. Recall that this weaker notion of approximate equilibrium is a mixed strategy profile $\sigma = (\sigma_1, ..., \sigma_n)$ satisfying the following property: the mixed strategy σ_i of each player i is an approximate best response to the other players' mixed strategies, but there is no requirement that everything in the support of this strategy is an approximate best response. The following theorem shows that the two versions of the* NASH *problem with the different kinds of approximation are equivalent.*

Theorem 2 (Daskalakis-Goldberg-Papadimitriou 2009 [2])**.** *Given an ϵ-approximate Nash equilibrium of an n-player game, we can efficiently compute a $\sqrt{\epsilon}(\sqrt{\epsilon} + 1 + 4(n-1)u_{max})$ well supported Nash Equilibrium, where u_{max} is the maximum absolute value of a payoff in the game.*

We now reduce `NASH` to `BROUWER`. For this we make use of Nash's fuction $y = f(x)$, which given a mixed strategy profile x computes another mixed strategy profile y. Recall from Lecture 5 that this function is defined as follows

$$y_p(j) = \frac{x_p(j) + \max(0, u_p(j; x_{-p}) - u_p(x))}{1 + \sum_{j \in S_p} \max(0, u_p(j; x_{-p}) - u_p(x))} \quad , \text{ for all } j \in S_p.$$

The following theorem, which can be found in [2], shows that Nash's function is Lipschitz.

Theorem 3 (Daskalakis-Goldberg-Papadimitriou 2009 [2])**.** *For any pair of mixed strategy profiles x, y*

$$|f(x) - f(y)|_\infty \leq [1 + 2u_{max} n \cdot m \cdot (m+1)]|x - y|_\infty,$$

where m is an upper bound on the number of strategies of a player and n is the number of players.

Given the above we can solve `BROUWER` to obtain an approximate fixed point of f. To conclude our reduction, we need to establish that the function f is approximation preserving; that is, if we are given an ϵ-approximate fixed point of f, this corresponds to an approximate Nash equilibrium.

Theorem 4 (Daskalakis-Goldberg-Papadimitriou 2009 [2])**.** *Let f be Nash's function defined above. Let also x satisfy $|f(x) - x|_\infty < \epsilon$. Then x is a δ-approximate Nash equilbrium of the game, where*

$$\delta = m\sqrt{\epsilon(1 + m \cdot u_{max})}(1 + \sqrt{\epsilon(1 + m \cdot u_{max})}) \max(u_{max}, 1),$$

m is an upper bound on the number of strategies of a player, and u_{max} is the maximum absolute value of a payoff in the game.

The above theorems (which we do not prove here) show that `NASH` can be reduced to `BROUWER`. However, there is a fine point. `BROUWER` is defined for functions in the hypercube, while Nash's function is defined on the product of simplices. For this to be a proper reduction, we have to embed the product of simplices in a hypercube, then extend Nash's function to points outside the product of simplices, making sure that given an approximate fixed point of the extended function we can compute an approximate fixed point of Nash's function. This can be done and is a 2-point exercise.

2.5 Poly-time Reductions, FNP, Total Problems, and PPAD

We can ask ourselves if `SPERNER` is FNP-complete. A way to show this would be to find a reduction from `SAT` to `SPERNER`.

This is impossible (in the present definition of poly-time reduction given above), since an instance of `SPERNER` always has a solution, while an instance of `SAT` may or may not have a solution.

One way around this would be to change our definition of a reduction to a more robust one, requiring that a solution to an instance of `SPERNER` should inform us on whether the corresponding instance of `SAT` has a solution or not.

However, it is unlikely that we have such a reduction from `SAT` to `SPERNER`. Suppose such a reduction did exist. Then we could just guess the solution to the `SPERNER` instance and then check whether the `SAT` instance reduced to it does or does not have a solution. Hence, we could obtain verifiable certificates both for the fact that a formula is satisfiable and for the fact that it is not satisfiable. This would mean that $\texttt{NP} = \texttt{co} - \texttt{NP}$, which most complexity theorists do not believe is true.

Another approach would be to turn `SPERNER` into a non-total problem by removing the boundary conditions. In this case `SAT` can be reduced to `SPERNER`, but for trivial reasons that do not reflect the structure of Sperner's problem.

In view of the difficulty to characterize the complexity of total search problems tightly using `FNP`, we wish to introduce new complexity classes capturing the hardness of solving `SPERNER`, as well as `BROUWER` and `NASH`. We do this by studying carefully the combinatorial structure in our proof of Sperner's lemma.

Recall that in our (directed-walk version of the) proof of Sperner's lemma, we constructed a directed graph in the space of simplices, where every node had at most one incoming and at most one outgoing edge. Moreover, there was a node in this graph (corresponding to our "starting simplex") that had no incoming edge, but it did have an outgoing edge. So it was a source of our graph. An easy parity argument establishes then that somewhere in our graph there must be a sink node. In fact, all sink nodes of our graph, and all sources different than the starting simplex are panchromatic simplices.

The `END OF THE LINE` problem defined next is the problem of finding a sink, or a source different than a pre-specified one, in a graph with in-degree and out-degree at most one. This is non-trivial since the graph could be exponentially large, but can a succinct description.

> `END OF THE LINE`: A graph on $\{0,1\}^n$ is defined by two circuits P, N that take as input an n-bit string (a node identity) and output an n-bit string (a node identity). P is interpreted as the "possible Previous" circuit, while N as the "possible Next" circuit. Given a pair of nodes v_1, v_2 there is a directed edge from v_1 to v_2 if and only if $P(v_2) = v_1$ and $N(v_1) = v_2$. That is, v_2 claims that v_1 is its previous node AND v_1 claims that v_2 is its next node.
>
> Given a pair of circuits P, N the `END OF THE LINE` problem is this: If 0^n is an unbalanced node (that is, its indegree is different from its outdegree), find another unbalanced node. Otherwise say "yes".

The class $PPAD$ is the class of all search problems in FNP reducible to `END OF THE LINE`.

Proposition 1. *1.* $\texttt{PPAD} \subseteq \texttt{FNP}$

2. $\texttt{SPERNER} \in \texttt{PPAD}$

Proof: To see the first item, note that it is easy to verify whether a node is unbalanced in an instance of `END OF THE LINE`.

To see the second item, it is sufficient to map a `SPERNER` instance to a pair of circuits P and N, so that a solution to the resulting `END OF THE LINE` instance corresponds to a solution of the original `SPERNER` instance.

To obtain our reduction, we define an efficiently computable correspondence between the simplices in the simplicization of the hypercube and the n-bit strings, for an appropriate choice of n. We also assume that the starting simplex in the instance of `SPERNER` is mapped to 0^n.

The circuits P, N are defined as follows:

- The value of $P(0^n)$ is set to 0^n, while the value of $N(0^n)$ is the simplex sharing the colorful facet with the starting simplex. Thus, the node 0^n is unbalanced.
- If a simplex S is neither colorful nor panchromatic (this is efficiently checkable), $P(S) = S$, and $N(S) = 0^n$. This guarantees that the node associated with simplex S is isolated.
- If a simplex S has a colorful facet shared with another simplex S', then if the sign of the facet is -1 for S, we set $N(S) = S'$. Otherwise (if the sign of the facet is $+1$ for S), we set $P(S) = S'$.

It is clear that the graph of the `END OF THE LINE` instance thus created is isomorphic to the graph that we would construct for showing the existence of a panchromatic simplex in the given instance of `SPERNER`. Hence, the solutions of `END OF THE LINE` are in one-to-one correspondence with the solutions of `SPERNER`. This completes the reduction and the proof of the proposition. Notice that it is important that the sign of a facet is locally computable and consistent along a path in the proof of Sperner's lemma. Otherwise, our graph won't satisfy that all in- and out-degrees are bounded by one. □

References

[1] K. Etessami and M. Yannakakis. On the Complexity of Nash Equilibria and Other Fixed Points (Extended Abstract). *FOCS*, 2007.

[2] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing,* 39(1): 195-259, 2009.