Computational Game Theory, Fall 2009 December 14

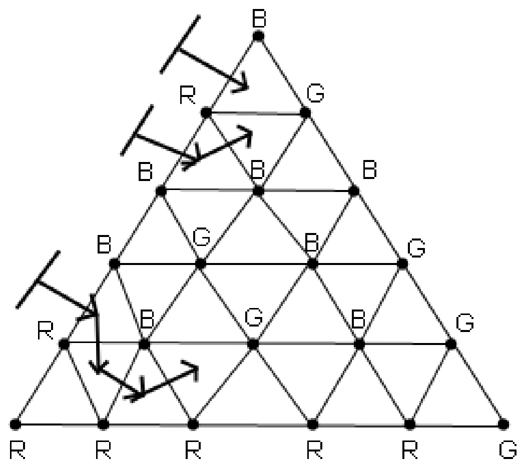
Lecture 12: Sperner's algorithm, PPAD, Lipton-Mehta-Markakis

Lecturer: Peter Bro Miltersen Scribe: Rok Grdina Marko Grgurovic

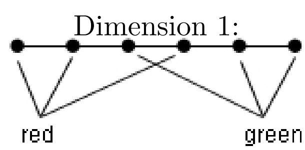
1 Sperner's Lemma

Let us begin by remembering Sperner's lemma from the previous lecture.

Lemma 1 (Sperner's lemma) *A Sperner coloring of a simplicial subdivision has an odd number of panchromatic subsimplices.*



We will prove Sperner's lemma by induction. The base case, dimension 1, is an easy exercise.



So assume that we have established the lemma for all smaller dimensions than k. Let a sperner colouring of a simplicial subdivision S be given and let P_k be its set of panchromatic subsimplices. Let S' be one of the $(k-1)$-dimensional facets of S (for $k = 2$, S' is just one of the three sides of the triangle). Let P_{k-1} be the set of panchromatic subsimplices of S'. By the induction hypothesis, $|P_{k-1}||$ is odd. We will define *perfect matching* in the set $P_{k-1} \cup P_k$. Clearly, the existence of a perfect matching implies that $|P_{k-1} \cup P_k|$ is even, and hence $|P_k|$ is odd.

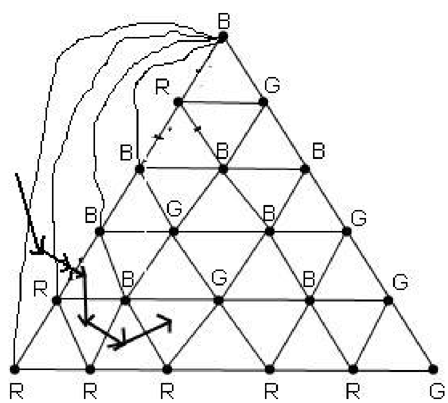
The perfect mactching is defined by the so-called *Sperner walk*. We start a Sperner walk either in a member of P_k or a member of P_{k-1} and end up in some other member. Let us first define a Sperner walk starting in a member x_0 of P_{k-1}. Note that x lacks one colour that a member of P_k has, let us call this colour "green". The first step of the Sperner walk goes to the k-dimensional subsimplex x_1 immediately adjacent to x_0. If this is panchromatic the walk ends here. Otherwise, there is another $(k-1)$-dimensional facet y of x_1 that has exactly the same colouring structure as

x_0. If this facet is at the boundary of S, it is a member of P_{k-1}, and the walk ends at y. Otherwise, the next subsimplex x_2 on the walk is the k-dimensional subsimplex sharing the facet y with x_1. If x_2 is panchromatic, the walk ends at x_2, otherwise we continue in exactly the same way. Let us now define a Sperner walk starting in a member x_0 of P_k. There is exactly one $(k-1)$-dimensional facet y of x_0 that has the same set of colours as membebrs of P_{k-1}. If y at the boundary of S, it is a member of P_{k-1}, and the walk ends at y. Otherwise, the next subsimplex x_1 on the walk is the k-dimensional subsimplex sharing the facet y with x_0. If x_1 is panchromatic, the walk ends at x_1, otherwise we continue. The walk indeed defines a perfect matching of $|P_k \cup P_{k-1}|$. To see this, we simply notice that if one step of the walk takes x to y then one step of the walk takes y to x. This completes the proof. This completes the proof of Sperner's lemma.

2 Scarf's algorithm

Sperner's lemma tells us that panchomatic subsimplex exists. Scarf's algorithm is an algorithm for finding one. Scarf's algorithm is a variation of "doing the Sperner walk". Note that the Sperner walk is defined inductively and apparently tricky to turn into an algorithm for finding a panchromatic subsimplex as we would have to try multiple starting points which we would first have to find, inductively. Scarf's algorithm avoids this problem. It makes a *single* walk in a slightly augmented k-dimensional subdivision and ends up in a panchromatic subsimplex.

We illustrate how this "Scarf walk" appears for $k = 2$. The general definition is similar and is omitted. We augment the triangle as follows: We introduce a number of "pseudo-subsimplices" along a selected edge as illustrated here.



This augmentation results in a figure with no bondary: The entire plane has now been (topologically) triangulated. Seen from some point on the outside of the figure, we are already inside a panchromatic subsimplex, namely the "inverted" big simplex. We start the Sperner walk in that subsimplex, but with the walk defined exactly as before. As the figure no longer has a boundary, we are guaranteed to end up in another panchromatic subsimplex, and that one must be a real one. This completes the description of Scarf's algorithm.

Inspecting the proof of Brouwer's fixed point theorem, we see that Scarf's algorithm leads to an algorithm for finding an *approximate* fixed point of a continuous function on the ball, an approximate fixed point being a point x for which the distance between $f(x)$ and x is smaller than ϵ, where ϵ is part of the input. We simply define a Sperner colouring with an appropriate granularity (depending on ϵ and f) and perform Scarf's algorithm on these. And inspecting the proof of the Nash equilibrium theorem, we see that that algorithm leads to an algorithm for finding an approximate Nash equilibrium, i.e., solving Task II of the last lecture. By slight abuse of notation, we shall call all three algorithms Scarf's algorithm.

2.1 Time complexity of Scarf's algorithm.

Even though Scarf's algorithm works rather well in practice, there is no worst case bounds for its complexity better than the total number of subsimplices in the subdivision (corresponding to the walk that visits all of these). Thus, from a worst case complexty point of view, the clever path following algorithm is not better than an exhaustive search among the subsimpllices. The time bound is $(O(N))^k$, where N is the number of vertices on a 1-dimensional face (the "granularity"). When applied to computing an ϵ-approximate Brouwer fixed point of a function f, this becomes $\approx (O(1/\delta))^k$, where δ satisfies $|x-y| \leq \delta \Rightarrow |f(x)-f(y)| \leq \epsilon$. And, finally, for the ϵ-Nash case with l being the number of players and n_i the number of strategies of player i and assuming all payoffs between 0 and 1, we get a time bound of $(O(\frac{1}{\varepsilon}))^{\sum_{i=1}^{l} n_i}$. Note that the dependence on the number of strategies is exponential, but the dependence on the precision parameter ϵ is polynomial. We shall see a completely different algorithm with opposite behaivor at the end of these notes.

2.2 The complexity class PPAD

Even though Scarf's algorihtm has poor worst case complexity it puts the problems of finding panchromatic triangles, approximate Brouwer fixed points and approximate Nash equilibria in a complexity class PPAD. These problems were later shown to be complete for this class and hence they are polynomial-time equivalent to each other. This completeness result is in hindsight the ultimate motivation for defining PPAD and putting task II in it. We will do this (well, not really, but we shall understand how it can be done), but we unfortunately do not have enough time to do the completeness result.

To warm up to defining PPAD, we first consider an alternative definition of NP: NP is the class of decision problems that are polynomial-time many-one reducible to SAT. Here, "L polynomial-time many-one reduces to L" means that there exists a polynomial time computable r so that $x \in L_1 \Leftrightarrow r(x) \in L'$.

PPAD can be analogously defined as follows: It is the class of *search* problems that are Turing-reducible in polynomial time to END-OF-A-LINE.

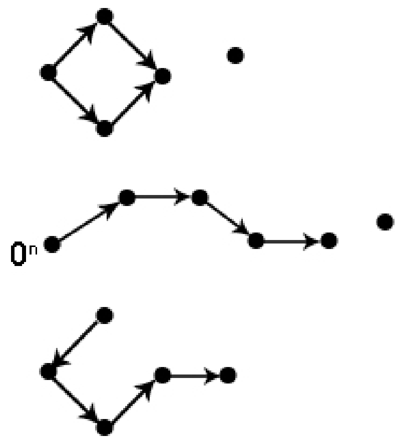
What is a search problem? In a search problems, for each input x, we have a non-empty set of allowed answers S_x. An algorithm solving a search problem returns one element (and one element only) of S_x. For example, one such problem is to determine the Nash equilibrium of a given 2-player game or to compute an approximate Nash equilibrium of a given 3-player game. It is not known how to capture these search problems exactly by decision problems. In particular, we do not know of any language L (no matter how convoluted a definition we allow), so that we can prove that L can be decided in polynomial time if and only if Nash equilibria for two-player games can be found in polynomial time.

A search problem S_1 reduces to a search problem S_2, written as $S_1 \leq S_2$, if S_1 can be solved in polynomial time, assuming a magic black box solving S_2 in linear time.

We now define the search problem END-OF-A-LINE Each input instance to the problem is given by two boolean circuits S, P
$S : \{0,1\}^n \Rightarrow \{0,1\}^n$

$P : \{0,1\}^n \Rightarrow \{0,1\}^n$

To explain what an allowed output is we look at a graph defined by the input. Let $G = (V, E)$ be the directed graph
$V = \{0,1\}^n$
$(x, y) \in E \Leftrightarrow S(x) = y \wedge P(y) = x \wedge y \neq 0^n$
The graph clearly has the property that each vertex has at most one incoming and one outgoing arc. Also 0^n has no incoming arc.



This means the graph consists of directed paths, directed cycles, and isolated points. An allowed output of the search problem is defined as follows:

- If 0^n is an isolated point, then 0^n is the only allowed output.
- If 0^n is not an isolated point, then the end point of a path (either beginning or end) but not 0^n must be returned.

Note that the problem satisifes the condition that every input has an allowed output: If 0^n is not an isolated point it must be the end point (the beginning) of a path and hence there must be another end point (the end). Note however, that we do not require *that* end point to be returned. This would make the problem much harder (in fact PSPACE-hard). It is easy to see that the correctly stated END-OF-A-LINE problem is NP-easy, but not NP-hard, unless NP=coNP.

Note that Scarf's algorithm is a path following procedure. It can be verifed (straightforwardly, but somewhat tediously) that Scarf's algorithm and the reductions puts task II, defined in the last lecture, in PPAD. The reduction establishing this constructs circuits P and S that correspond to taking one step of the Sperner walk.

3 Lipton-Mehta-Markakis algorithm

The Lipton-Mehta-Markakis algorithm is based on Hoeffding's inequality:

Lemma 2 (Hoeffding's inequality) *Let X_1, X_3, ..., X_k be independent random variables with X_i taking values in the interval $[0,1]$. Let $A = \frac{\sum_{i=1}^{e} x_i}{n}$. Then,*

$$\Pr[|A - E[A] \geq t] \leq 2e^{-2kt^2}$$

For simplicity, we describe the Lipton-Mehta-Markakis algorithm for the 2-player case. Assume both players have $n \geq 2$ strategies. Consider any fixed, pure or mixed, strategy y of one of the players, say Player 2. Consider a mixed strategy x of Player 1. Sample $k = 9 \ln n/\epsilon^2$ times from x and consider the mixed strategy x' corresponding to the uniform distribution on the samples. Look at the expected payoff to either Player 1 or Player 2 when Player 1 plays x' instead of x.

Lemma 3 *For $j = 1, 2$,* $\Pr[|u_j(x, y) - u_j(x', y)| \geq \epsilon/3] \leq 2n^{-2}$

Proof Note that the probability space in the lemma is the space of the sampling, not the strategy space (which is probability space for the mixed strategies). Apply Hoeffding, and note that $2e^{-2(9 \ln n/\epsilon^2}(\epsilon/3)^2 = 2n^{-2}$ □

Of course, by symmetry we can also apply the lemma with the roles of Player 1 and Player 2 switched. We will do so below.

Lemma 4 *Suppose (x, y) is a Nash equilibrium and we sample from $k = 12 \ln n/\epsilon$ times from both x and y to make x' and y'. Then* $\Pr[(x', y')\text{is } \epsilon\text{-Nash}] > 0$.

Proof We have that $\Pr[(x', y')\text{is } \epsilon\text{-Nash}]$ is equal to

$$\Pr[\forall i, u_1(e_i, y') \leq u_1(x', y') + \epsilon \wedge \forall j, u_2(x', e_j) \leq u_2(x', e_j) + \epsilon]$$

where e_i is the mixed strategy which is really pure as it puts all probability mass on i. Look at the first condition

$$\forall i, u_1(e_i, y') \leq u_1(x', y') + \epsilon. \tag{1}$$

Since (x, y) is a Nash equilibrium, we have $u_1(e_i, y) \leq u_1(x, y)$. Now, if we also have $u_1(e_i, y') \leq u_1(e_i, y) + \epsilon/3$ and $u_1(x, y) \leq u_1(x', y) + \epsilon/3$ and $u_1(x', y) \leq u_1(x', y') + \epsilon/3$, we have (1), by combining the 4 inequalities. Arguing similarly about the second condition, we have that a sufficient condition for (x', y') to be ϵ-Nash is that all the following inequalities are satisfied:

- For all i, $u_1(e_i, y') \leq u_1(e_i, y) + \epsilon/3$
- $u_1(x, y) \leq u_1(x', y) + \epsilon/3$
- $u_1(x', y) \leq u_1(x', y') + \epsilon/3$,
- For all j, $u_2(x', e_j) \leq u_2(x, e_j) + \epsilon/3$
- $u_2(x, y) \leq u_2(x, y') + \epsilon/3$
- $u_2(x, y') \leq u_2(x', y') + \epsilon/3$

There are $2n + 4$ inequalities. By Lemma 4, the probability that any particular one of them is *not* satified is at most $2n^{-2}$. By the union bound, there is positive probability that they are all satisfied, if $n \geq 2$. □

Since the sampling results in an ϵ-Nash equilibrium with positive probability, we in particular have that an ϵ-Nash equilibrium exists where both players use a mixed strategy which is the uniform distribution on a multiset of size k. This leads to the Lipton-Mehta-Markakis algorithm (to find the ε-Nash equilibrium)

- Do an exhaustive search through all subsets of size $9 \ln n/\epsilon^2$ for each of the two players.

- Check the ε-Nash condition

- Output the first that works

The complexity of the algorithm is $n^{O(\log n/\epsilon^2)}$, i.e., not polynomial, but quasi-polynomial in n, which is much better than exponential dependence (but with worse dependence on the precision than Scarf's algorithm).

We remark that a very similar proof gives the algorithm with the same complexity for finding an ϵ-minimax strategy of a 3-player game that we alluded to earlier.

Returning to ϵ-Nash computation, if we want very crude precision ϵ, we can acheive polynomial time. Suppose all payoffs are in [0, 1], then

$$\varepsilon = 1/2$$

ε-Nash equilibrium for 2-players can be found in polynomial time.

- Let i_1 be an arbitrary strategy of player 1
- Let j be the best reply to i_1
- Let i_2 be the best reply to j

The profile where Player 1 plays i_1 with probability 1/2 and i_2 with probability 1/2 and Player 2 plays j with probability 1 is easily seen to be a 1/2-equilibrium. It is known how to achieve ϵ-Nash equilibria in polynomial time for slightly smaller values of ϵ, but it is not known how to achieve, say, $\epsilon = 0.25$ in polynomial time for the two-player case.