Lecture 43 Scribe Notes

Instructor: Éva Tardos *Konstantinos Mamouras (km528)*

1 Lecture 43 – Wednesday 2 May 2012 – NP-completeness of deciding if a game has two Nash equilibria

1.1 Reminder

Recall that a *search problem* can be given as a multi-valued partial function or binary relation $R = \{(x, y)|x,y \text{ are objects encoded as strings}\}$. The search problem asks: Given an input x, return some y such that $(x, y) \in R$ or return "no" if there exists no such y. *Decision problems* can be seen as special cases of search problems, where the answer is either "yes" or "no".

In the previous lecture we defined the classes FNP and TFNP, which are classes that contain search problems. TFNP is the subclass of FNP that contains search problems that are *total*, i.e. problems for which there is a guarantee that an answer exists (equivalently, never is "no" a valid answer). We also saw three subclasses of TFNP, called PPA (polynomial-time parity argument), PPP (polynomial-time pigeonhole principle), and PLS (polynomial-time local search), that are defined based on the kind of proof used to show that an answer always exists.

One further subclass of PPA is PPAD (polynomial-time parity argument for directed graphs). The canonical problem that is complete for PPAD is the *End-of-Line* problem:

> End-of-Line Problem: We are given an implicitly represented directed graph that has an exponential number of vertices. Every vertex is represented by an n-bit string. The input includes two algorithms pred, and succ, which for a vertex u compute (in polynomial time) the predecessor $\mathsf{pred}(u)$ and the successor $\mathsf{succ}(u)$ respectively. This implies that each vertex has indegree and outdegree at most once, and the non-existence of a successor/predecessor can be indicated using a special symbol. The vertex represented by the all 0s string, which we denote by 0, is called the *standard source* and it has no predecessor.
>
> A *solution* is a vertex that is either a sink (i.e. a vertex with no outgoing edges) or a source (i.e. a vertex with no incoming edges) other than the standard one. Equivalently, a solution is a non-zero vertex u for which in-degree(u) + out-degree$(u) = 1$. The *goal* is to find a solution.

The graph described by an input for the above problem consists of disjoint directed lines, and simple cycles. We note that at least one solution exists, and it can be found by walking on the path starting from the standard source 0 until the endpoint of the path is reached. Because of the degree conditions, it is not possible for this path to form a cycle. This endpoint is one of the many possible solutions to the problem. The problem, however, does not require this particular solution.

***A harder variant of* End-of-Line**: If we were requiring the particular solution described in the previous paragraph instead, i.e. if our goal was "find the end of the path starting from the standard source 0", then the problem would become harder. It is known that this variant is complete for PSPACE. Containment is PSPACE is easy to see: Guess the length n of the path starting from 0 and walk n steps. If a sink is not reached, then reject. Otherwise, return the sink. This is a

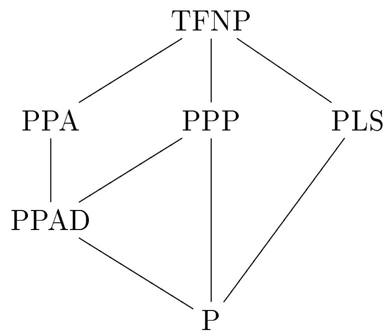

Figure 1: Known inclusions for the complexity classes TFNP, PPA, PPP, PLS, PPAD, P.

non-deterministic algorithm that requires polynomial space, since only n and the current node need to be stored. By Savitch's theorem NPSPACE = PSPACE and hence the problem is in PSPACE. Hardness for PSPACE is also not difficult to show.

In Figure 1 we give diagramatically known inclusions for the complexity classes TFNP, PPA, PPP, PLS, PPAD, as well as P. A class that is lower in the diagram and is connected to another class is contained in the latter.

Theorem 1 (Daskalakis & Papadimitriou, 2006)**.** Nash is complete for PPAD.

We have shown the easy part of the theorem, which is containment of the problem in the class PPAD, via Brower's Theorem and Sperner's Lemma. The hard part, which is showing hardness of the problem for PPAD, will not be covered in class.

1.2 NP-completeness of deciding if a game has two Nash equilibria

Many natural ways of turning the problem of finding a Nash equilibrium into a non-trivial decision question give us NP-complete problems. The main theorem of today's lecture is a result of this sort. First, we define the problem Nash-Two: Given a game, is it the case that it has two Nash equilibria?

Theorem 2. Nash-Two is NP-complete.

Before we start with the proof of the theorem, we note that the following problems are also hard:

- "Given a game, find a socially optimal Nash equilibrium". This is a search problem, not a decision problem. The corresponding decision problem would be: "Given a game and a threshold, is there a Nash equilibrium with social welfare above the threshold?" The latter problem is known to be NP-complete.
- "Given a game, is there a Nash equilibrium where Player 1 plays a pure strategy?" This problem is NP-complete.
- "Given a game and a strategy of Player 1, is there a Nash equilibrium where Player 1 plays this strategy with positive probability?" This problem is NP-complete.

1 \ **2**	x_1 $\neg x_1$ $\dots$ x_n $\neg x_n$	c_1 $\dots$ c_m	d
x_1 $\neg x_1$ $\vdots$ x_n $\neg x_n$			$(0,0)$
c_1 $\vdots$ c_m			$(5,0)$
d	$(0,0)$	$(5,0)$	$(6,6)$

Table 1: (Incomplete) payoff matrix for the game constructed from 3SAT instance.

Proof. We now prove Theorem 2. We know that 3SAT is NP-complete. We will reduce 3SAT to NASH-TWO. Consider an instance of 3SAT that involves n variables $x_1, x_2, \dots, x_n$, and clauses $C_1, \dots, C_m$, where each C_i is of the form $C_i = \lambda_1^i \vee \lambda_2^i \vee \lambda_3^i$. Each λ_j^i is a literal, i.e. either a variable x or the negation $\neg x$ of a variable. The formula is $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$. The question we need to answer for 3SAT is: "Does there exist a truth assignment for the variables $x_1, \dots, x_n$ such that ϕ is true?".

Note: The game we will construct from the 3SAT instance will involve only 2-players, which makes the theorem stronger: already in 2-player games, the problem is NP-complete.

We consider two symmetric players with available strategies $x_1, \neg x_1, \dots, x_n, \neg x_n, C_1, \dots, C_m, d$, where d stands for "default". So, each player has $2n + m + 1$ strategies. It remains to specify the payoffs for each pure strategy profile. We want to ensure that the "default" strategy profile (d, d) is a Nash equilibrium. So, in this case we will assign a payoff vector with high payoffs.

- If the players play (d, d), then the payoff vector is $(6, 6)$.
- If one player plays d and the other plays a clause c_j, then the first gets 5 and the second gets 0.
- If ones of them plays d and the other plays a literal, then both get 0.

From the above it follows that (d, d) is a Nash equilibrium. Table 1 contains the payoff vectors we have defined so far. It now remains to define the rest of the payoffs in the $(2n + m) \times (2n + m)$ submatrix so that a second Nash equilibrium exists iff ϕ is satisfiable.

We consider a generalization of the familiar *rock-paper-scissors* game (RPS) to n-objects. We call this game $\mathbf{RPS}_n$ or (generalized) rock-paper scissors on n objects. RPS_n is a two-player game. The available pure strategies for the players are n objects $1, 2, \dots, n$, which we arrange in a cycle

$$1 \to 2 \to \dots \to (n-1) \to n \to 1 \ .$$

This cycle indicates which object beats another. An arrow from i to j ($i \to j$) indicates that i beats j. In other words, an object beats the neighbor on the right and loses to the neighbor on the left. The payoffs are $+1$ for the winner, -1 for the loser, and 0 for both players in the remaining cases where there is a tie (i.e. no object beats the other).

It is not hard to show that the mixed strategy profile where each player chooses an object uniformly at random is a Nash equilibrium. For this equilibrium, the expected payoff for both players is 0. If n is even, we notice that the mixed strategy profile given by

– Player 1 chooses an odd object uniformly at random
– Player 2 chooses an even object uniformly at random
is also a Nash equilibrium and the expected payoff for both players is again 0. We want to exclude this equilibrium, so we restrict ourselves to those RPS_n games with n odd.

Claim 3. For n odd, the game RPS_n has a unique Nash equilibrium, where both players choose an object among $1, 2, \ldots, n$ uniformly at random.

Note: We would like to think of the n objects in RPS_n as the n variables of our 3SAT instance. If the instance contains an even number of variables, then without loss of generality we can introduce one extra variable that does not appear in any clause in order to make n odd.

We also consider a further generalization of the rock-paper-scissors game. We call this game $\textbf{BWRPS}_n$ (black and white RPS on n objects). Now, we think that have two colored versions of each object, one black and one white. So, each player has $2n$ pure strategies available. We think of each object as having two attributes: color (black/white) and index $(1, \ldots, n)$. The payoffs are now defined in the following way:

- If the players play objects of *different index*, then the payoffs are given as in RPS_n disregarding the colors.
- If the players play objects of the *same index*, then:
  - If the colors agree, then both players receive 0 payoff.
  - If the colors differ, then both players receive a huge negative payoff.

This game has at least 2^n different Nash equilibria: For every index i choose a color c_i. This can be done in 2^n ways and thus we obtain a *color assignment*. The mixed strategy profile where each player chooses an object (i, c_i) uniformly at random is a Nash equilibrium.

Remark: BWRPS_n has also Nash equilibria other than these 2^n equilibria described in the previous paragraph, since by the parity argument of previous lectures we know that a game has an odd number of Nash equilibria.

- We will now define the payoff vectors for the $2n \times 2n$ submatrix of the payoff matrix of Table 1, i.e. for the region where both players play a literal. Let this $2n \times 2n$ submatrix be given by BWRPS_n with the only difference being that both players receive an added bonus of 2. That is, the payoffs are increased from -1 to 1, from 0 to 2, and from 1 to 3. Now, we think of the color assignment we described in the previous paragraph as a *variable assignment* (black corresponds to a negative literal and white to a positive literal, and the index of the variable in the literal is the index of the object).

- If both players play a clause, then the payoff vector is $(2, 2)$.

- We have to define the payoffs in the case where one of the players p plays a literal λ and the other player q plays a clause C.

  - If $\neg\lambda$ is in C, then p gets 0 and q gets n.
  - Otherwise, both players receive 0.

  As we will see shortly, the reason for these payoffs is to create a profitable deviation from a truth assignment that does not satisfy the formula to the clause that is falsified by it.

($\Rightarrow$) Assume that ϕ is satisfiable and let $\lambda_i, \ldots, \lambda_n$ be a selection of literals corresponding to a truth assignment to the variables that satisfies ϕ. We argue that the mixed strategy profile where both players choose a literal among $\lambda_i, \ldots, \lambda_n$ uniformly at random is a Nash equilibrium. The expected payoff is 2 for both players. Clearly, no player wants to unilaterally shift probability to the default strategy, because the payoff there is 0. Moreover, no player want to shift probability to a clause, because a clause can only be falsified by at most two literals (the assignment is satisfying) and hence the corresponding payoff is at most $2 \cdot \frac{1}{n} \cdot n = 2$, which is no better.

($\Leftarrow$) Conversely, suppose that the game possesses at least two Nash equilibria. We know that (d, d) is one of them. First, we claim that every Nash equilibrium other than (d, d) has its support in $x_1, \neg x_1, \ldots, x_n, \neg x_n$. Assume not. If a player chooses a clause or d, then the other player will shift to d and then the first player will shift to d as well. Now, consider this equilibrium whose support is a truth assignment. There is no collision (disagreement of the assignments of the players) possibility, because then a player would prefer to default. Moreover, as have already seen in the RPS games, the distributions will be uniform. Assume for the sake of contradiction that the assignment does not satisfy ϕ. Then, there is a clause C that is falsified by 3 literals and therefore a player has an incentive to deviate to C, because this would give him payoff $3 \cdot \frac{1}{n} \cdot n = 3 > 2$. This contradicts the fact that we have an equilibrium, and the proof is complete. $\square$