CMSC 858M: Fun with Hardness
Spring 2021

Instructor: Mohammad T. Hajiaghayi
Scribe: Erika Melder

April 30, 2021

1 Overview

In this chapter, we look at graph arguments where the search space may be intractable. We use parity arguments to deduce features of these graphs without having to scan the search space. We then use this to define the class PPAD and prove that certain game theory problems (notably Nash equilibrium) are PPAD-complete, via a long chain of reductions through many problems. Finally, we conclude by analyzing other graph parity arguments and describing the classes that they define.

2 End of the Line Problem

Two elementary results on a directed graph are that the sum of all indegrees equals the sum of all outdegrees, and if there exists an unbalanced vertex then there must exist another one. These results give us that the problem UNBALANCED, where we are given a graph and one of its unbalanced nodes and asked if there is another, is trivially in P since the answer is always Yes. If we want to find the node, then the complexity depends on what we consider the input size. If the input to the problem is a graph, then this takes polynomial time; however, it is possible for us to require exponential time with respect to a logarithmic input. This is the complexity captured by the End of the Line problem:

Definition 1 *Consider two injective circuits* P *and* N *that map* $\{0,1\}^n \to \{0,1\}^n \cup \{NO\}$ *that satisfy for all bitstrings* $\mathsf{P}(x) = y \Leftrightarrow \mathsf{N}(y) = x$*. These circuits describe a graph: if* $\mathsf{P}(x) = y$ *where* y *is a bitstring, there is an edge* xy*, and if* $\mathsf{N}(x) = y$*, there is an edge* yx*.*
You are promised exactly one of $\mathsf{P}(0^n)$ *and* $\mathsf{N}(0^n)$ *is unbalanced.*
The problem End of the Line (EOL) *is to find another unbalanced vertex, which is guaranteed to exist.*

It is unlikely this is an NP-hard problem - if SAT $\leq_p$ EOL then NP = coNP. The exact hardness is not known, but it is assumed to be hard due to the exponential search space.

3 Game Theory

3.1 The Prisoner's Dilemma and the Penalty Shot Game

The Prisoner's Dilemma is a classic example in game theory of finding a Nash equilibrium, often used to show that it may not maximize social welfare in a simplistic setting. The Nash equilibrium (NE) of a game is a pair of strategies where neither player has an incentive to alter their strategy, independent of what they predict the other player to do. In this case, for the Prisoner's Dilemma, the social welfare maximizing solution is for both players to stay silent, which will give them each two years of time. However, if either player confesses against the other, then they improve their situation regardless of what the other does - if the other stays silent, then the confessor gets one year instead of two, and if the other confesses, then the confessor gets four years instead of five. This leads to a stable solution where both players confess and each gets four years - the NE.

Some games do not have an NE restricted to pure strategies. For instance, there is the Penalty Shot Game, which is summarized simply: Two players each choose 0 or 1 and reveal simultaneously. If they match, Player 1 wins a point and Player 2 loses a point. If they don't match, then Player 2 wins a point and Player 1 loses a point. This game has no pure NE because the players can always improve by deviating in some scenario depending on what the other player will do. However, there is a mixed strategy equilibrium - if both players flip a fair coin. This will ensure that expectation is maximized by sticking to the fair coin result.

3.2 Formal Game Theory

Definition 2 *A* finite game *has a set* P *of* r *players, and for each player, it has both a set of pure strategies and a real-valued payoff function for each combination of strategies of all players.*

Definition 3 *Define a* mixed strategy *as a probabilistic distribution over a player's strategy set.*
A Nash equilibrium *is a collection of mixed strategies, one for each player, such that the expected payoff for each player is maximized by sticking to their mixed strategy assuming that the others do too.*

Finding the Nash equilibrium of an r-player game is a complexity problem. Typically we input an ε and accept an approximate answer within ε. It is known that 2-player Nash equilibrium is in P in the zero-sum case, but it is an unknown complexity otherwise. If generalized 2-player NE is NP-hard, then NP=coNP, similarly to EOL.

4 Other Problems

4.1 Brouwer Fixed Point Theorem

Theorem 1 *Any continuous function* f *from a convex, compact subset* D *of Euclidean space to itself has at least one* $x \in D$ *satisfying* $f(x) = x$.

The Fixed-Point Theorem is a key result in topology that also enabled Nash to prove the existence of Nash equilibrium, by finding a fixed point of the vector field representing willingness to deviate from a strategy. The problem of finding a fixed point in general is hard because f is continuous over the reals. However, one could analyze a discrete version of this problem, BROUWER [**?**].

4.2 Sperner's Lemma

Definition 4 *Consider a 3-coloring of the lattice points of an* $n \times n$ *grid. The coloring is* valid *if the bottom row is all RED, the left column is all YELLOW, and all other boundary nodes are BLUE.*
If this grid is triangularized, then a trichromatic triangle *is a triangle where all three vertices have different colors.*

Theorem 2 *For all valid 3-colorings of an* $n \times n$ *grid there exists an odd number of trichromatic triangles.*

One proof of this theorem is to enter the grid from the lower left and walk only across edges that are yellow-red and have yellow on the left side of the path. The construction ensures that there are no cycles, since any triangle has at most one such edge that is satisfying from the exterior of the triangle, and it cannot leave the grid, since there are no edges that are crossable on the boundary. This means that the walk is finite and so there must be some trichromatic triangle, since we can enter but not exit a trichromatic triangle and no other structure. Another proof takes the same structure and turns it into a digraph, where we add one unbalanced input node to the lower left. This guarantees the existence of another unbalanced node in the graph, which is our trichromatic triangle.

The problem of 2-SPERNER is a subcase defined as follows.

Definition 5 *Consider a circuit mapping points on an* $n \times n$ *grid to one of three colors. (We don't test if the coloring is valid.)*
Then 2-SPERNER is the problem of either finding and outputting a trichromatic triangle, or stating that the coloring is invalid. (Even on an invalid coloring it is okay to output a trichromatic triangle if one is successfully found.)

One may define n-SPERNER via a more technical generalization to an n-dimensional lattice [**?**].

5 Total Search Problems in NP

Definition 6 *A* search problem *is one where, given a relation* R *on a set* D *and a target* $x \in \mathsf{D}$*, find* $y \in \mathsf{D}$ *with* $x\mathsf{R}y$ *or output that none exists.*
A total search problem *is a search problem where you are promised that a short satisfying* y *exists and must find it.*

Search problems are NP-hard in general, but total search problems being NP-hard would imply NP = coNP. All four of EOL, NASH, BROUWER, and SPERNER are total search problems. There are analogous classes to P and NP for these problems. These definitions are due to Rich [**?**].

Definition 7 *A relation* R *is in* FP *if* $x\mathsf{R}y$ *can be computed in polynomial time, the decision problem of whether there exists a* y *satisfying* $x\mathsf{R}y$ *for an input* x *is in P, and the function which finds such a* y *guaranteed one exists takes polynomial time.*

Note that we often consider the set associated with such a relation; that is, $\{x : (\exists y)[x\mathsf{R}y]\}$.

Definition 8 *A relation* R *is in* FNP *if* $x\mathsf{R}y$ *can be computed in polynomial time, and there is a polynomial* p *where any* x *that has a* y *satisfying* $x\mathsf{R}y$ *has some* y' *where* $x\mathsf{R}y'$ *and* $|y'| \leq p(x)$*.*

Like in FP, we can consider the set associated with such a relation. We also don't need to be able to find y for membership in FNP. There is a subclass of FNP for total search problems, known as TFNP:

Definition 9 *A relation* R *is in* TFNP *if it is in FNP and, for every* x*, there is a (short)* y *such that* $x\mathsf{R}y$*.*

Unlike the prior two sets, it does not make sense to consider the set associated with such a relation because the set is just all x.

6 Reductions in FNP and the Class PPAD

Definition 10 *Let* R *and* R' *be in FNP. Let* L *and* L' *be the respective associated sets. Then* R *is poly time reducible to* R' *if:*

- *There is some function* f *computable in polynomial time such that* $x \in \mathsf{L} \Leftrightarrow f(x) \in \mathsf{L}'$*.*
- *There is some function* g *computable in polynomial time such that* $f(x)\mathsf{R}'y$ *then* $x\mathsf{R}g(y)$*.*

This is how to carry out an FNP reduction. However, the four total search problems seem unlikely to be FNP-hard due to the NP = coNP corollary of such a result. To capture this hardness, Papadimitriou defined the class PPAD (**P**olynomial **P**arity **A**rguments on **D**igraphs) [**?**].

Definition 11 *Consider* R *in FNP.*

- R *is in* PPAD *if* R *is reducible to EOL.*
- R *is PPAD-hard if EOL is reducible to R.*
- R *is PPAD-complete if it is in PPAD and PPAD-hard.*

Immediately we can determine that SPERNER, NASH, and BROUWER are all in PPAD. SPERNER has a simple reduction to EOL, BROUWER can be proven from SPERNER which leads to a reduction, and NASH can be proven from BROUWER. These problems are also PPAD-complete, by a series of reductions.

7 2-NASH is PPAD-Complete

It was known that 3-NASH is PPAD-complete [**?**]. While 2-NASH seems much easier and is in P for the zero-sum case, it is also PPAD-complete [**?**, **?**]. This comes from reducing a PPAD problem to one in the Euclidean unit cube, then reducing that to 3D-Sperner, then Arithmetic Circuit SAT, then PolyMatrix Nash, then 2-Nash.

Definition 12 Arithmetic Circuit SAT *(ARITHCIRCSAT) is the problem of, given a circuit with variable nodes, assignment operators, binary addition and subtraction, scalar multiplication, variable comparison, and equality checking to constants as possible actions at nodes (an* arithmetic circuit*), find a 0-1 satisfying assignment to the variable nodes.*

Definition 13 *A* graphical game *is a game that can be modeled by a digraph of players, where the payoff of each player depends only on their own strategy and the strategies of in-neighboring players.*
A polymatrix game *is a graphical game where the utility function of a player is simply the sum of the utilities of the two-player subgames on each incoming edge.*

We can reduce ARITHCIRCSAT to POLYMATRIX NASH by constructing polymatrix games whose Nash equilibria rely on computing that arithmetic operation from the probabilities of incoming edges. There is one gadget for each operation - add, subtract, copy, assignment, scalar multiplication, and comparison. This, combined with PPAD-completeness of ARITHCIRCSAT and the earlier result that NASH is in PPAD, proves that POLYMATRIX NASH is PPAD-complete.

From this, there is a complex reduction to 2-NASH using a union argument that involves viewing every player in the POLYMATRIX NASH instance as a client, and having two players as lawyers competing to assign strategies to these clients. This game can solve the POLYMATRIX NASH instance if and only if it can approximate the Nash equilibrium of the 2-NASH instance. The fact that this is an approximation ends up not mattering because ARITHCIRCSAT is still PPAD-hard to approximate.

8 Other Arguments of Existence

8.1 PPA

Several other theorems give nonconstructive existence of graph elements by parity. One is that, if a graph has an odd-degree node, then it must have a second odd-degree node. This invokes the problem of ODG:

Definition 14 *Consider a circuit* C *on* $\{0,1\}^n$ *that maps bitstrings to sets of bitstrings. You are given that* 0^n *has an odd cardinality in its mapped set. The problem of* Odd Degree Node *(ODG) is to find a second node of odd degree.*

This evokes the following class, known as PPA (**P**olynomial **P**arity **A**rguments):

Definition 15 *Consider* R *in FNP.*

- R *is in* PPA *if* R *is reducible to ODG.*
- R *is PPA-hard if ODG is reducible to* R*.*
- R *is PPA-complete if it is in PPA and PPA-hard.*

Another problem trivially in P is the problem HAM-3reg of finding a second Hamiltonian cycle in a 3-regular graph when given one. A sequence of theorems (as seen in chapter 13 of the book) asserts that such a second cycle always exists, so the decision problem is in P. The problem of actually finding the cycle is in PPA.

PPA also contains the problem CHEVALLEY, which is a problem of, given L polynomials on n variables over $\mathbb{Z}_2$ with degree-sum at most n, and given one solution, find another solution. In terms of real-world problems, the consensus-halving problem is PPA-complete; finding square roots mod n and quadratic residues mod n is in PPA with unknown hardness; and integer factoring has a randomized reduction to PPA (deterministic assuming the Generalized Riemann Hypothesis) with unknown hardness.

8.2 FS

Every DAG has at least one sink, meaning a node with no outdegree. The problem of *Find Sink* (FS) is to find this sink. Formally:

Definition 16 *Consider a circuit* C *on* $\{0,1\}^n$ *that maps bitstrings to sets of bitstrings. Define the neighbors of a bitstring to be the bitstrings in its output set that are strictly larger than itself (the resultant graph is acyclic). The problem of FS is to find the sink of the graph.*

This evokes the following class, known as PLS (**P**olynomial **L**ocal **S**earch)

Definition 17 *Consider* R *in FNP.*

- R *is in* PLS *if* R *is reducible to FS.*
- R *is PLS-hard if FS is reducible to* R*.*
- R *is PLS-complete if it is in PLS and PLS-hard.*

A notable problem in PLS is finding a locally optimal max cut of a weighted graph (a cut where no one-vertex swaps will increase the cut size).

8.3 PPP

Any function from a set of size n to a set of size $n-1$ is necessarily non-injective by the pigeonhole principle. Similarly, any subset of $[2^n - 1]$ with n elements whose sum is less than $2^n - 1$ must have two subsets with the same sum. These both evoke problems:

Definition 18 *Consider a circuit* C *on* $\{0,1\}^n$ *that maps bitstrings to bitstrings. The problem of* COLLISION *is to find either an* x *that maps to* 0^n*, or two distinct* x, y *where* $C(x) = C(y)$*.*

Definition 19 *Consider a subset* A *of* $[2^n - 1]$ *where* $|A| = n$ *and the sum of* A*'s elements is smaller than* $2^n - 1$*. The problem of* Distinct Subsets *(DS) is to find two distinct subsets of* A *with the same sum.*

COLLISION is used to evoke a new class, PPP (**P**olynomial **P**igeonhole **P**rinciple) in which DS lies [**?**].

Definition 20 *Consider* R *in FNP.*

- R *is in* PPP *if* R *is reducible to COLLISION.*
- R *is PPP-hard if COLLISION is reducible to* R*.*
- R *is PPP-complete if it is in PPP and PPP-hard.*

It is known that DS lies in PPP, but it is unknown whether it is PPP-hard. It is also known that PPP contains the entire class PPAD, as well as a variant of Shortest Lattice. Similarly to PPA, integer factoring has a randomized reduction to a variant of COLLISION which is deterministic assuming GRH.

9 Open Problems

- From the book - It is unknown whether the current algorithm for approximate Nash equilibrium on zero-one matrices is optimal. For approximation factor ε, the current optimal algorithm takes time $n^{\mathcal{O}(\log n/\varepsilon^2)}$.
- Also from the book - HAM-3_{REG} and CHEVALLEY are in PPA, but it is unknown whether they are PPA-hard.
- The intent behind defining these classes is that they represent a sort of analogous intermediary hardness hierarchy that isn't quite NP-hard but is harder than P. It is still an open problem to find where exactly these classes lie with respect to FP and FNP, and if any of the known inclusions are strict.
- One notable PPAD-complete problem is envy-free cake cutting with polynomial utility [**?**]. This is an open problem in the field of mechanism design that is of major interest to finding optimal auction solutions.
- There are two newer results for subsets of FNP: End Of Potential Line and Unique End Of Potential Line. These both evoke new classes defined in the paper. Trivially, Unique EOPL is a subset of EOPL, but it is unknown whether the two classes are equal.
- Another recent result is that the class CLS is the intersection of PPAD and PLS, it contains FP, and it is where gradient descent is contained. It is uncertain whether FP is a strict subset of CLS or whether CLS equals one or both of PPAD and PLS.

10 Important Problems

- EOL - PPAD-complete [**?**]

  Given two injective circuits which allow forward and backward traversal of a bitstring graph, and the promise that either the predecessor or successor of 0^n is unbalanced, find the other unbalanced vertex.
- NASH - PPAD-complete [**?**, **?**]

  Given a finite game of r players, find a collection of mixed strategies that is a Nash equilibrium (assuming no player deviates from the equilibrium, no player can improve utility by deviating).
- BROUWER - PPAD-complete [**?**, **?**, **?**]

  Given a convex set and a continuous function, find a fixed point. (Note that this problem needs to be adapted in order to be tractable, as seen in [**?**].)

- 2-SPERNER - PPAD-complete [?, ?]

  Given a circuit that three-colors an $n \times n$ grid, either find and output a trichromatic triangle, or find that the coloring is invalid. (n-SPERNER is defined in [?].)

- ODG - PPA-complete [?]

  Given a circuit mapping bitstrings to sets of bitstrings, and a promise that 0^n has an odd cardinality, find a second node of odd degree.

- FS - PLS-complete [?]

  Given a circuit mapping bitstrings to sets of bitstrings, define the neighbors of a bitstring to be those that are strictly larger than it. Find a sink of the graph.

- COLLISION - PPP-complete [?]

  Given a circuit mapping bitstrings to bitstrings, find either a bitstring which maps to 0^n or two distinct bitstrings that map to the same output.

References

[1] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings, pages 261–272. IEEE Computer Society, 2006. https://doi.org/10.1109/FOCS.2006.69

[2] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. SIAM Journal of Computing, 39(1): 195–259, 2009. https://doi.org/10.1137/070699652.

[3] Deng, X.; Qi, Q.; Saberi, A. (2012). "Algorithmic Solutions for Envy-Free Cake Cutting". Operations Research. 60 (6): 1461. doi:10.1287/opre.1120.1116.

[4] Fearnley, John; Goldberg, Paul W.; Hollender, Alexandros; Savani, Rahul (11 November 2020). "The Complexity of Gradient Descent: CLS = PPAD ∩ PLS". arXiv:2011.01929 [cs.CC].

[5] Fearnley, John; Gordon, Spencer; Mehta, Ruta; Savani, Rahul (December 2020). "Unique end of potential line". Journal of Computer and System Sciences. 114: 1–35. doi:10.1016/j.jcss.2020.05.007.

[6] Emil Jérabek. Integer factoring and modular square roots. Journal of Computing and System Sciences, 82(2):380–394, 2016. https://arxiv.org/pdf/1207.5220.pdf.

[7] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? Journal of Computer and System Sciences, 37(1): 79–100, 1988. https://doi.org/10.1016/0022-0000(88)90046-3

[8] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. Journal of Computer and Systems and Science, 48(3): 498–532, 1994.https://doi.org/10.1016/S0022-0000(05)80063-7.

[9] Elaine Rich. Automata, computability, and complexity: theory and application. Prentice Hall, 2008.