

Lecture 19

PPAD-hardness of Nash equilibrium

Instructor: Prof. Constantinos Daskalakis¹

We continue the discussion from the lecture on total search and TFNP by giving a glimpse of how the PPAD-hardness of finding ϵ -approximate Nash equilibria was shown by [Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. \[DGP09\]](#).

The proof can be broken down into two main steps:

- Reduction from the End-of-the-line problem to (approximate) Brouwer.
- Reduction from (approximate) Brouwer to (approximate) Nash equilibria.

The first step is relatively easy, and we will not cover it here. The second step is more involved and requires a careful construction of a reduction from Brouwer to Nash equilibria. This is the part we will focus on in this lecture.

The key idea is the following: in the reduction from END-OF-THE-LINE to Brouwer, we define a continuous function f for which we need to find an approximate fixed point. We now need to construct a game such that a Nash equilibrium of the game is the same as a fixed point of f (up to approximations). The issue is that it is not clear how we can have games “compute” functions. Can we construct games in such a way that their behavior at Nash equilibria can be seen as “computing something”? The answer is positive, as we see next.

L19.1 Arithmetic Circuit SAT

We show that given a function represented as an *arithmetic circuit*, it is possible to construct a game whose Nash equilibria correspond to computing a fixed point of the function. This is the key idea behind the reduction from Brouwer to Nash equilibria.

In particular, we will restrict our attention to functions constructed through circuits that are composed of the following:

- Variable nodes $v_1, \dots, v_n$;
- Gate nodes $g_1, \dots, g_m$ of six possible types:

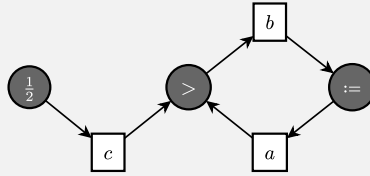
¹Some of the content of the lecture was adapted from material from Costis Daskalakis.

Gate	Symbol	Input-output relationship
Assignment	$x_1 \rightarrow \text{:=} \rightarrow y$	$y = x_1$
Constant	$a \rightarrow y$	$y = a$
Addition	$x_1, x_2 \rightarrow \text{+} \rightarrow y$	$y = \min\{1, x_1 + x_2\}$
Subtraction	$x_1, x_2 \rightarrow \text{-} \rightarrow y$	$y = \max\{0, x_1 - x_2\}$
Multiplication	$x_1 \rightarrow \text{\(\times a\}} \rightarrow y$	$y = \max\{0, \min\{1, a \cdot x_1\}\}$
Comparison	$x_1, x_2 \rightarrow \text{>} \rightarrow y$	$y = \begin{cases} 1, & \text{if } x_1 > x_2 \\ 0, & \text{if } x_1 < x_2 \\ \text{any}, & \text{if } x_1 = x_2. \end{cases}$ When the inputs are equal, this gate does not restrict the output.

- Directed edges connecting variables to gates and gates to variables (loops are allowed);
- Variable nodes have in-degree 1; gates have 0, 1, or 2 inputs depending on type as above; gates & nodes have arbitrary fanout.

Definition L19.1 (Arithmetic Circuit SAT problem). Given an arithmetic circuit satisfying the description above, output an assignment of values $v_1, \dots, v_n \in [0, 1]$ that satisfies all the gates.

Example L19.2. Consider the following diagram.



The only satisfying assignment is $a = b = c = 1/2$.

It is easy to see that this problem has the flavor of a Brouwer fixed point.

Theorem L19.3 ([DGP09]). The Arithmetic Circuit SAT problem always admits a solution, and it is PPAD-complete to find it.

L19.2 From gates to games

It is possible to convert an Arithmetic Circuit SAT instance into a Nash equilibrium computation problem in a *multiplayer game*. (The game can also be converted into a two-player [CDT09] or three-player game [DGP09], but we do not show how in this lecture).

The idea is to use *gadgets*: constructions that simulate the behavior of the gates in the circuit.

L19.2.1 Addition gate

Consider any game that contains the following interaction between four players x, y, z, w , each of which has two actions, denoted $\{0, 1\}$. With a slight abuse of notation, we will call x, y, z, w the probability of playing action 1; hence, $x, y, z, w \in [0, 1]$.

Example L19.4 (Addition gadget game). Consider any game that contains as a substructure the gadget shown in the diagram below, and payoffs set as follows.

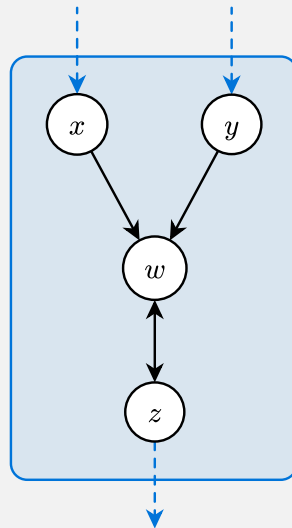


Figure 2: Addition gadget game. The dashed blue edges denote possible edges in the game, which do not affect the result in Theorem L19.5.

► *Payoffs of player w.* The payoff of player w is defined as follows. If w plays 0, her payoff does not depend on z 's strategy, but only on x and y , according to the payoff table

	$y = 0$	$y = 1$
$x = 0$	0	1
$x = 1$	1	2

If w plays 1, her payoff does not depend on x and y 's strategy and depends on z 's according to the table

	$z = 0$	$z = 1$
	0	1

► *Payoffs of player z .* The payoff of player z is defined according to the table

	$z = 0$	$z = 1$
$w = 0$	1/2	1
$w = 1$	1/2	0

► *Other payoffs and considerations.* The utilities of players x and y are independent of the strategies of w and z . Player w does not affect other players in the game.

Theorem L19.5. In all Nash equilibria of the game, $z = \min\{x + y, 1\}$.

Proof. Suppose that $z < \min\{x + y, 1\}$. Then, $z < x + y$. But then w will deterministically play $w = 0$, which will force z to play $z = 1$. This is a contradiction, since by hypothesis $z < \min\{x + y, 1\}$, which implies $z < 1$.

Suppose now that $z > \min\{x + y, 1\}$. In this case, $\min\{x + y, 1\} \neq 1$, as otherwise this would imply $z > 1$ which is impossible. Thus, $z > x + y$. This implies $w = 1$ and hence $z = 0$, which is again impossible since $z > x + y$, which implies $z > 0$.

The only remaining possibility is therefore $z = \min\{x + y, 1\}$, as we wanted to show. ◻

Bibliography

- [DGP09] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The Complexity of Computing a Nash Equilibrium,” *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009, doi: 10.1137/070699652.
- [CDT09] X. Chen, X. Deng, and S.-H. Teng, “Settling the complexity of computing two-player Nash equilibria,” *Journal of the ACM (JACM)*, vol. 56, no. 3, pp. 1–57, 2009.