Hardness of Computing Equilibria

Yiding Zhang
Institute for Interdisciplinary Information Sciences
zhangyd19@mails.tsinghua.edu.cn

Abstract

In this survey, we focus on some complexity-theoretic results related to equilibrium, an important concept in game theory. We summarize some results showing that it is computationally intractable to find an equilibrium in general. In addition, we also present some methods to deal with the intractability.

Introduction

Equilibrium concepts play a central role in game theory. When we consider a game or some real-world situations that can be formalized as a game, we usually want to find a stable state that will stay unchanged once it is reached. Such kind of states are usually called equilibria, for example, most famously the Nash equilibrium, or the subgame-perfect equilibrium in a extensive form game, a Bayesian Nash equilibrium in a game with incomplete information, etc. Equilibrium is an important concept since it can be regarded as a "solution" of the game, or as a prediction of the most likely outcomes of a long-rum system. However, whether we can find or reach an equilibrium in a reasonable amount of time is a problem, or in other words, equilibria may be computationally intractable. The hardness of finding an equilibrium is an interesting area for research since we still do not have algorithms for equilibrium computation in general - although we already have a proof for its existence. Also, if finding an equilibrium is generally hard, it is not reasonable to assume that all players can converge to an equilibrium with limited amount of effort, and thus the concept of equilibrium loses its credibility as a prediction of outcomes.

In the following sections, we present some complexity results for computing equilibria (mainly showing that they may be hard to compute). After that, we show some techniques that can bypass or (somewhat) overcome the computational intractability of computing equilibria.

Intractability results for equilibria

It has been proved that a mixed-strategy Nash equilibrium always exists in a normal form game (Nash et al. 1950). But the proof of this result is non-constructive, and cannot imply an efficient algorithm for finding an equilibrium.

The main hardness result of Nash equilibrium is its PPAD-completeness. In this section we summarizes some intuitions behind these results and why they show the hardness of this problem.

Why not NP-completeness

NP-completeness is the standard complexity theoretic approach to prove that a problem is computationally intractable. However, NP does not seem to be the right complexity class to capture Nash equilibrium.

Since our goal is to find a Nash equilibrium and it is not a decision problem, we need a generalized "function" version of NP, usually called FNP. The complexity class FNP consists of problems of the form "given a relation R and input x, output any y such that $(x, y) \in R$, or output no if such y does not exists". We require that the relation $R \subset \Sigma^* \times \Sigma^*$ must satisfy the following conditions:

1. polynomial-time verifiable: for any pair (x, y), we can check whether $(x, y) \in R$ in polynomial time.
2. polynomially bounded: there exists a polynomial p such that $|y| \leq p(|x|)$ for any $(x, y) \in R$.

Also we can define FP as the subset of FNP that is polytime solvable, which is a generalized version of P. Of course Nash equilibrium is in FNP, and is not known to be in FP. FNP and FP are just analogous to NP and P : we can prove that FNP $=$ FP if and only if P $=$ NP.

We also have a function version of NP $\cap$ coNP called TFNP ("T" for "total"). This class is a subset of FNP with an additional constraint:

- For any x, there always exists a y such that $(x, y) \in R$ (i.e., the relation R is a total relation).

Obviously, FP $\subseteq$ TFNP $\subseteq$ FNP, but neither inclusion is known to be strict, just like P $\subseteq$ NP $\cap$ coNP $\subseteq$ NP. Note that Nash's proof guarantees that every normal form game has a MSNE, and thus the Nash equilibrium problem is in TFNP. The following results shows an evidence that Nash equilibrium is not FNP-complete:

Lemma 1 *If a* FNP*-complete problem is in* TFNP*, then* NP $=$ coNP *(Megiddo and Papadimitriou 1991).*

NP $=$ coNP implies that PH collapses to the first level. Since PH does not collapse is a standard conjecture in complexity theory, we believe that Nash equilibrium does not

have NP-completeness. However, we will present that a subset of TFNP called PPAD can capture the hardness of Nash equilibrium.

The class PPAD

To capture the hardness of finding Nash equilibria, the main idea is to check the non-constructive step in Nash's existence proof, and check whether it is hard to simulate computationally. Recall the proof of existence we talked about in class. For each strategy profile, we define the best response correspondence and directly apply Kakutani's fixed point theorem to show that a MSNE (i.e., a fixed point) always exists. The non-constructive step is Kakutani's fixed point theorem, which is a generalized version of Brouwer's theorem, and further more, Brouwer's theorem can be proved based on Sperner's lemma. So let's start from Sperner's lemma. Here we ignore the description of this lemma since we have talked about it in class, and exactly the same settings are used in the rest of this section. Recall the proof of Sperner's lemma, in which we abstract a graph based on the coloring of vertices, and then apply the following lemma to the graph to prove the result:

Lemma 2 (Parity lemma) *(Undirected) Any finite graph has an even number of odd-degree nodes.*

(Directed) If a directed graph has an unbalanced node (i.e., in-degree not equal to out-degree), then it must have another one.

The parity lemma can be proved by a simple counting. Based on the lemma and some "starting points" with odd degree on the three edges, it is guaranteed that a trichromatic triangle always exists. Now going back to the complexity class FNP: for a given relation R with its corresponding problem in FNP, if we want to show that this problem is in TFNP, we need to show that R is a total relation. Notice that the above parity lemma is just used to give a proof of existence and can further show the totality of R, we define a subclass of TFNP by unifying the way of proving totality as parity arguments.

Definition 3 (Polynomial Parity Argument) *A problem P is in* PPA *if and only if it can be defined in the form "on input x, given a standard leaf of $G(x)$, output another leaf", where the graph $G(x)$ is defined on a configuration set $C(x) \subseteq \Sigma^{\mathsf{poly}(|x|)}$ with edges defined in terms of a polytime algorithm $M_x(c)$ that on input $c \in C(x)$, output all the neighbours of c on the graph (at most two) (Papadimitriou 1994).*

Intuitively, PPA is just the subclass of TFNP that are guaranteed to have a solution because of the parity lemma. Remark that there is something worth noticing in the definition:

1. Graphs with degree at most 2 are just enough: if we allow poly-bounded number of inputs as the settings in the parity lemma, it is just an equivalent definition, since we can decompose the graph into a new graph of degree at most 2 such that previous odd-degree nodes coincide with leaves in the new graph.
2. The graph may have exponential size: this is the main reason that such problems may be hard. For example, in Sperner's lemma, we can define a triangle with edge length 2^n, and on a given coordinate (x, y, z) (each of x, y, z uses n bits), output the color on that point in polynomial time (i.e., a succinct description of colors).

By doing the same thing on a directed graph and trying to find a source or a sink, we can similarly define PPAD as follows:

Definition 4 (Polynomial Parity Argument, Directed) *A problem in* PPAD *is defined by modifying the definition of* PPA *such that $M_x(c)$ outputs an ordered pair of nodes (c_1, c_2) denoting two directed edges (c_1, c) and (c, c_2) respectively. Then the problem is that given a node that is either a source or a sink (indegree+outdegreee=1), try to find another source or sink.*

This definition forms a trivially complete problem of PPAD. From a graph-theoretic perspective, the problem is to find another end of a line in the directed graph. So this PPAD-complete problem is called END OF THE LINE.

By adapting the theorems into search problems, we can get some problems like SPERNER, BROUWER and KAKUTANI that are in PPAD. (problems like BROUWER has some kind of approximation since a continuous function in the theorem cannot be precisely characterized by Turing machines). Also we have NASH $\in$ PPAD since Nash's proof (Nash et al. 1950) essentially gives a reduction from NASH to BROUWER. Another more direct approach to prove this result is to use Lemke-Howson algorithm (Lemke and Howson 1964), which computes a Nash equilibrium for two-player games by following a similar END OF THE LINE path.

PPAD-completeness results

We have defined a subclass of TFNP that contains NASH and some other related fixed-point problems. If the trivial complete problem END OF LINE can be reduced to these problems in polynomial time, then a Nash equilibrium can be found in polynomial time if and only if all the hard problems in PPAD mentioned above can be efficiently solved. Therefore, Nash equilibrium is somewhat computationally intractable.

The completeness result of NASH has been proved, and such problem is PPAD-hard even in 2-player games:

Theorem 5 *Finding MSNE in 2-player normal form games are* PPAD*-complete (Chen and Deng 2006).*

The detailed proof is quite complex and here we omit it for simplicity. The main idea is to reduce 3-DIMENSIONAL BROUWER to 2-NASH, where 3-DIMENSIONAL BROUWER is proved to be PPAD-complete. BROUWER's completeness result is proved by embedding the END OF LINE graph into a 3-dimensional cube, which is used to define a continuous function with its approximate fixed points corresponding to the unbalanced nodes of the END OF LINE graph (Daskalakis, Goldberg, and Papadimitriou 2009).

Besides Nash equilibria, we also have PPAD-hardness results for other kind of equilibria:

1. Arrow-Debreu equilibria (Chen, Paparas, and Yannakakis 2013): finding equilibria in markets with complements is PPAD-hard;
2. Almost zero-sum games (Mehta 2014): finding equilibria of a bimatrix game (A, B) with $rank(A + B) \geq 6$ is PPAD-hard ($rank(A + B) = 0$ corresponds to zero-sum games);
3. Anonymous games (Chen, Durfee, and Orfanou 2015): finding an approximate Nash equilibrium (with exponentially small error) in an anonymous game is PPAD-hard;

Ways to overcome the intractability

Based on the hardness result we have shown, finding an efficient algorithm that can in general compute Nash equilibria seems hopeless. However, it does not mean that games in real-worlds are wholly unpredictable and we cannot do anything to deal with them. In fact, We can try to bypass the computational intractability by some other methods.

Possible method 1: approximation

Sometimes an approximate result is enough for real-world applications. So we may also be satisfied if approximate equilibria are tractable. However, the following result shows that relative-approximated Nash equilibria are also hard to find:

Theorem 6 (approximation hardness of NASH) *In a 2-player games, finding a mixed strategy profile such that neither player can individually improve his current payoff by more than an ϵ-fraction is PPAD-hard for some $\epsilon > 0$ (Daskalakis 2013).*

But for absolute error approximation, we know that the problem is *unlikely* to be PPAD-hard, because there has already been an algorithm that can find ϵ-Nash of 2-player n-strategy games in $O(n^{\log n/\epsilon^2})$ time (Lipton, Markakis, and Mehta 2003). However, a polynomial-time algorithm is still missing.

Another positive example of approximation is anonymous games. We have mentioned that exact equilibria are intractable in anonymous games (Chen, Durfee, and Orfanou 2015). However, we can get arbitrarily good approximations in polynomial time if the number of strategies does not scale to infinity (Daskalakis, Kamath, and Tzamos 2015).

Possible method 2: games with special properties

Although finding MSNE in an arbitrary normal form game is (believed) hard, we still have games with special properties that can be solved easily. For example, we can find a MSNE in the 2-player zero-sum games efficiently by a linear programming.

However, zero-sum games are not always simple if more players are involved.

Theorem 7 *k-player zero-sum games are PPAD-hard if $k \geq 3$.*

Proof This result can be proved easily by reducing arbitrary 2-player games to 3-player zero-sum games. For any 2-player game, suppose that on any outcome σ, the payoff of each player if $u_1(\sigma)$ and $u_2(\sigma)$. Then we add a third player that has only one possible action, and the payoff he receives is $u_3(\sigma) = -u_1(\sigma) - u_2(\sigma)$. Since the third player has only one possible action, of course he has no intention to deviate. So a MSNE in the 3-player zero-sum game implies a MSNE in the original 2-player game, which is PPAD-hard to compute. ■

In the proof we use some kind of 3-way interactions to prove its hardness. By limiting such interactions, we can still extend to a wider range of games with some good properties of 2-player zero-sum games. For example, we have the following zero-sum **polymatrix** games:

Definition 8 (Zero-sum polymatrix games) *(Cai et al. 2015)*

A polymatrix game G consists of the following components:

- *a finite set of players $V = \{1, 2, \ldots, n\}$, and a finite set of strategies S_i for each player i;*
- *a finite set of undirected edges E between players, and for each edge $[i, j] \in E$ a 2-player game (p^{ij}, p^{ji}) with players i, j, strategy sets S_i, S_j, and payoffs p^{ij}, p^{ji} respectively (p^{ij} and p^{ji} are functions from $S_i \times S_j$ to $\mathcal{R}$);*
- *the payoff of player i is $p_i(s) = \sum_{[i,j]\in E} p^{ij}(s_i, s_j)$ for each strategy profile $s = (s_1, s_2, \ldots, s_n) \in \prod_{j\in V} S_j$.*

A polymatrix game G is zero-sum if for all $s \in \prod_{j\in V} S_j$, we have $\sum_{i\in V} p_i(s) = 0$.

This definition is just a multiplayer generalization of 2-player zero-sum games such that the entire game can be decomposed into several 2-player games (i.e., the edges E). In other words, only 2-way interactions are allowed. In zero-sum polymatrix games, we have the following properties similar as 2-player zero-sum games:

1. A Nash equilibrium can be found in poly-time by a linear programming.
2. The Nash equilibria form a convex set.
3. If each individual uses a no-regret learning algorithm, the game converges to a Nash equilibrium.

Possible method 3: tractable alternatives of MSNE

Since Nash equilibria are computationally intractable, we can try to find some other plausible definitions of equilibria that are easier to compute. Correlated equilibrium is just an example. In Nash equilibrium, we assume that the strategies of the players are mutually independent. If we remove this constraint, we get **correlated equilibrium** (intuitively, the choice of players can be correlated after removing the independence constraint). We can show that finding a correlated equilibrium is tractable:

Theorem 9 *The problem of finding a correlated equilibrium is in P.*

Proof Here we just give a simple sketch. The main idea is that equilibrium conditions can now be expressed as linear

constraints on the joint action distribution. For each normal form game, we maintain a variable for every pure strategy profile denoting its probability in an equilibrium, and then construct the equilibrium constraints on these variables and solve it by linear programming. Since the game description needs to give the payoffs of each pure strategy profile, the linear program has polynomial size in the game description. ■

Correlated equilibrium is in P, while Nash equilibrium is PPAD-hard. So we may sometimes use correlated equilibrium instead of Nash equilibrium as a credible prediction of a game.

Acknowledgments

The author thanks the talk about complexity and algorithmic game theory from Simons Institute (https://simons.berkeley.edu/talks/constantinos-daskalakis-2015-08-26) for the introduction of the complexity-theoretic approaches in game theory.

References

Cai, Y.; Candogan, O.; Daskalakis, C.; and Papadimitriou, C. 2015. A multiplayer generalization of the minmax theorem. *Mathematics of Operation Research, Forthcoming* .

Chen, X.; and Deng, X. 2006. Settling the complexity of two-player Nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, 261–272. IEEE.

Chen, X.; Durfee, D.; and Orfanou, A. 2015. On the complexity of nash equilibria in anonymous games. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 381–390.

Chen, X.; Paparas, D.; and Yannakakis, M. 2013. The complexity of non-monotone markets. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 181–190.

Conitzer, V.; and Sandholm, T. 2008. New complexity results about Nash equilibria. *Games and Economic Behavior* 63(2): 621–641.

Daskalakis, C. 2013. On the complexity of approximating a Nash equilibrium. *ACM Transactions on Algorithms (TALG)* 9(3): 1–35.

Daskalakis, C.; Goldberg, P. W.; and Papadimitriou, C. H. 2009. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing* 39(1): 195–259.

Daskalakis, C.; Kamath, G.; and Tzamos, C. 2015. On the structure, covering, and learning of poisson multinomial distributions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 1203–1217. IEEE.

Lemke, C. E.; and Howson, Jr, J. T. 1964. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics* 12(2): 413–423.

Lipton, R. J.; Markakis, E.; and Mehta, A. 2003. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, 36–41.

Megiddo, N.; and Papadimitriou, C. H. 1991. On total functions, existence theorems and computational complexity. *Theoretical Computer Science* 81(2): 317–324.

Mehta, R. 2014. Constant rank bimatrix games are PPAD-hard. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 545–554.

Nash, J. F.; et al. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* 36(1): 48–49.

Papadimitriou, C. H. 1994. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences* 48(3): 498–532.

Roughgarden, T. 2010. Algorithmic game theory. *Communications of the ACM* 53(7): 78–86.