Algorithmic Game Theory Assignment

Group work Component

Group 8

I. Introduction

This paper is a report on the complexity of Nash equilibria, summarising the reductions and results of [1] and [2]. We first discuss the background and motivation of these complexities, and then the results discovered and proof methods used in [1] and [2]. Additional research of interest in similar areas is also deliberated.

II. Relevant Background

Game Theory is the study of agent behaviour in models of competition. Each player chooses an action, which determines the payoff both players receive [3]. A famous example of a game is the Prisoner's Dilemma, which pits two prisoners against each other. A player's strategy determines what the player should do, for example, should they keep quiet, or should they fink, and with what probability?

A Nash equilibrium is a set of strategies for each player such that each player does not want to change their given strategy, i.e. each player would be more likely to receive a lower payoff by deviating from their current strategy. These strategies can be separated into two categories: pure Nash equilibria that only includes strategies that say to go for just one action, and mixed Nash equilibria where an action is chosen based on a probability distribution [3]. The only pure Nash equilibria for the Prisoner's Dilemma is for both players to fink, since if one of the players were to stay quiet instead, that player would spend longer in prison. An example of a mixed Nash equilibria is for the game rock-paper-scissors, where each player should play rock, paper, and scissors each with $\frac{1}{3}$ probability, since it maximises their expected payoff (which is 0) [1].

Nash's Theorem states that all games with a finite number of players and pure strategies must have at least one Nash equilibrium; it does not matter whether it is pure, or mixed. This has been proven using the Fixed Point theorem, where if a "best response" function is defined between a player's current strategy and a unilaterally deviating strategy, it can be said that there is always a strategy S such that the best response is to stick with the strategy S [4].

Within the two main complexity classes of **P** and **NP** there exist many sub-classes. A problem is said to be complete within a class when all other problems in its class can be reduced (i.e., transformed) to it with an appropriate reduction. A problem is said to be total when, for every possible instance, there is guaranteed to be at least one solution. The problem then moves from asking if there is a correct solution, to finding a correct solution. The class of all such problems is called Total Function Non-deterministic Polynomial Search (**TFNP**) [5]. **TFNP** does not have any complete problems of its own, but sub-classes of it do.

One important subclass of **TFNP** is Polynomial Local Search (**PLS**). A problem in **PLS** can calculate the cost of a solution in polynomial time, along with exploring the neighbourhood of the solution in polynomial time. This makes it possible to verify a point is a local optima in polynomial time. A problem can be shown to be **PLS**-complete if it can be reduced to CIRCUITFLIP [6].

Another key subclass of **TFNP** is Polynomial Parity Arguments on Directed graphs (**PPAD**). This class is defined in terms of its complete problems, with the main such problem being the END OF THE LINE problem. The END OF THE LINE problem works on a potentially exponentially large directed graph G that has no isolated vertices, and each vertex has at most one predecessor, and at most one successor. G is defined by a function f that takes a vertex as an input and returns the predecessor and successor of a vertex, if they exist. The END OF THE LINE problem asks, given G and a vertex s that has a successor but no predecessor, find a $t \neq s$ with either no successor or predecessors [7].

III. Results

This section summarises the findings of [1] and [2].

Paper [1] provides evidence that there are games in which convergence to a mixed Nash equilibrium takes prohibitively long. It also shows that finding a mixed Nash equilibrium is complete for a class of problems called **PPAD**, containing several other known hard problems. The results reported in this article indicate that there is not an efficient algorithm for computing a mixed Nash equilibrium. The main result in [1] is that the computational problem of finding mixed Nash equilibria (NASH) is **PPAD**-complete, and this is proven in two parts. First, that the problem is in **PPAD**, meaning that it can be reduced to the END OF THE LINE problem which is a total search problem. The second part is to prove that the problem is complete, achieved by the reverse reduction (**PPAD** back to NASH). Both parts here are established through Brouwer's fixed point theorem. The

results obtained in [1] also imply that finding an ε-Nash equilibrium is **PPAD**-complete, if ε is inversely proportional to an exponential function of the game size.

Paper [2] shows that in congestion games a pure Nash equilibrium can be computed in polynomial time in the symmetric network case, while the problem is **PLS**-complete in general. This overall result is proven in several stages. [8] shows that every congestion game has a pure Nash equilibrium. Second, [2] proves that there is a polynomial algorithm for finding a pure Nash equilibrium in symmetric network congestion games. Third, it is shown that computing a Nash equilibrium in the general network case is **PLS**-complete. [2] then shows that under the necessary smoothness assumptions (delay functions satisfying the Lipschitz assumption with constant K), the ε-approximate Nash equilibria of non-atomic congestion games can be computed in strongly polynomial time. Where the non-atomic congestion game is the limit of the congestion game as n, the number of players, goes to infinity. This problem is reduced to the multicommodity min-cost flow problem in [2]. Finally [9] shows that any exact potential game is isomorphic to a congestion game, completing the overall result obtained by [2].

IV. Reductions

The first reduction described in [1] is a reduction from NASH to the END OF THE LINE problem given in [1] in order to prove NASH is in **PPAD**. This reduction uses Brouwer's fixed point theorem, which put simply means that a mapping in the Euclidean space of a compact and convex subset back onto itself must have some fixed point that does not change in the mapping F. This gives rise to the problem BROUWER, which is simply when given F and any subset, find a fixed point. Two inconveniences are discussed in [1] which are how to specify F and how to deal with fixed points that are not rational numbers. These are solved by first restricting the subset to the unit cube and by assuming that F is given by some tractable algorithm which can map from the unit cube to the correct value. To deal with irrational fixed points F is assumed to obey a simple Lipschitz condition that simply means that the distance between result for 2 points is bounded by K the Lipschitz constant of F and distance between the 2 points. This is done so the domain can be discretized. As with NASH this means approximate fixed points bounded by some constant ϵ can be found. A reduction exists from BROUWER to END OF THE LINE[1]. This is done by dividing the unit square into a mesh of small triangles (triangulation). The vertices of the triangles are coloured based on the angle F maps it in from the horizontal. This 3-colouring satisfies a property that red cannot appear on the lower side of the square, blue cannot appear on the left side of the square, and the other 2 sides cannot have yellow. Sperner's Lemma states that any colouring obeying this property must have a triangle with all 3 colours. By the Lipschitz condition the vertices of such a triangle are fixed points. By having all triangles with a red and yellow vertex form a directed graph in which an edge exists from a triangle to another if they share an edge from red to yellow clockwise in the first triangle. Such a graph consists only of paths and cycles [1]. An assumption can be made that the left of the square only has one edge between yellow and red. This edge is part of a triangle that may be 3-coloured or if not must form a path that ends at a 3-coloured triangle meaning at least one 3-coloured triangle must exist. This kind of graph is an instance of END OF THE LINE[1]. NASH can be trivially reduced to BROUWER as a function can be defined that maps from the set of strategies to itself, based on payoff improvement. It is clear that a fixed point under such a function must correspond to a Nash equilibrium as there is no way to improve from it. Therefore this function and set of strategies form an instance of BROUWER, consequently NASH is reducible to END OF THE LINE.

To prove NASH is **PPAD**-complete [1] presents a set of reductions from an instance of END OF THE LINE can be reduced to NASH. First END OF THE LINE is reduced to BROUWER; the unit cube is again used and it is subdivided into a mesh of smaller cubes (cubelets) each with a grid point in its centre. These grid points can then be assigned 1 of 4 colours $[0-3]$ based on the vector $F(x)-x$. The vectors determining what colour is chosen are set so $F(x)-x$ is only near 0 if x is near all 4 colours. To convert an instance G of END OF THE LINE to such a cube each vertex of the END OF THE LINE graph is placed either in bottom or top left edges of the cube. This gives rise to a special property that if an edge exists in G between 2 vertices then a sequence of grid points will be coloured 1, 2, or 3. It is possible to create an arrangement of colours such that all 4 are only adjacent grid points (corresponding to an approximate fixed point) in an area of the cube that corresponds to a solution to END OF THE LINE and is thus **PPAD**-complete.

Next by reducing BROUWER to NASH it can be shown NASH is also **PPAD**-complete. The reduction from BROUWER to END OF THE LINE is such that all Fs can be computed using circuits consisting of addition, multiplication, and comparison operators forming a dataflow graph [1]. Each node of this graph can be converted into its own game which can then be composed with the other nodes games forming a larger game representing the full circuit result. Each of the mini games has 4 players w, x, y, z. Players can take either a "go" or "stop" action such that z's action represents the result of performing the node operator on the probabilities of x and y's strategies, w acts as a mediating node between x, y, and z [1]. Payoffs are then defined for the players such that, for multiplication, at any Nash equilibrium the probability z picks "go" is the product of the x and y probabilities [1], payoffs can similarly be chosen for addition and multiplication by a constant. To represent a F using this method, the location of a point x in the unit cube can be represented by 3 players each choosing "go" with probability equal to the corresponding coordinate in the cube. Then by chaining together enough of the mini games described above F can be calculated as the 3 output players whose "go" probabilities are the coordinates of $F(x)$. It is possible to assign suitable payoffs such that any Nash equilibrium results in the input players and output players having the same "go"

probabilities which corresponds to a fixed point. In order to avoid contradicting Nash's theorem, F must be calculated as the average of a grid around the point in the cube due to **the brittle comparator problem** [1] giving an approximate fixed point. The game is, by construction, a graphical game [1] and can be simulated as a three-player normal form game by having the 3 players represent m nodes each that never play against each other in the dataflow graph or share a node that they both play against. Each of the players nodes don't compete with each other so there are no conflicts of interest, this means any Nash equilibrium in this 3 player game corresponds to one in the original game. A final necessary adjustment is to have the players compete in a rock-paper-scissors game to ensure the mixed-strategy profiles are balanced [1]. The resulting 3 player game is efficient to compute and completes the reductions from BROUWER to NASH so NASH is **PPAD**-complete.

The first result shown in [2] is that a symmetric network potential game has a pure Nash equilibrium that can be found in polynomial time. This is done by reducing the network N representing the game to an instance of MIN-COST FLOW by replacing all edges in N with a set of parallel edges each with capacity 1 and costs given by the corresponding delay function for that edge (resource). It follows simply that any integer solution to this problem minimises the potential function $\phi(s)$ of N [2]. Next it is shown that finding pure Nash equilibria of all other types of congestion game is **PLS**-complete. First it is proven for general congestion games by reducing a known **PLS**-complete problem POSNAE3FLIP [10] to a general congestion game. Each 3-clause in an instance of POSNAE3FLIP can be represented as 2 resources, with no delay or delay equal to their weight if there are more than 2 players (variables in the formula). Each player has 2 strategies one contains all the first resource for clauses containing the player and the second contains all the second resource for those clauses. It is easy to see that a Nash equilibrium in such a game is an optimum of POSNAE3FLIP. Symmetric games are easily proven by reducing a non-symmetric game to one. This is done by adding a new resource to all states in each action set in the game. These new resources have delay functions such that $d_e(j)$ is a very large integer for all $j > 1$ and 0 otherwise. Any equilibrium to such a game must involve a player using one of these new action sets as a strategy. Due to the choice of delay function for the new resources it is obvious that such an equilibrium when the new resources are removed from the strategy is simply the same as an equilibrium in the original game.

Asymmetric network congestion games require a more complex route. The problem POSNAE3FLIP must be extended as new kinds of clauses are needed to express this case properly; previously acceptable clauses are now incompatible as this type of game is more specific [2]. The new problem is called WITNESSED XPNAE3FLIP and is constructed from POSNAE3FLIP as another proxy problem to make the proof possible. It is shown that there is a **PLS** reduction from WITNESSED XPNAE3FLIP to NETWORK CONGESTION GAME by carefully creating edges in the network based on clauses in the instance of XPNAE3FLIP and setting delays to reflect the properties of these clauses [2]. It is further proved that WITNESSED XPNAE3FLIP is also **PLS**-complete. As WITNESSED XPNAE3FLIP is constructed from POSNAE3FLIP, it is natural that this proof leverages the known **PLS**-completeness of POSNAE3FLIP and so it follows similar steps to the reduction from CIRCUITFLIP used in that original proof [10].

V. Related Research

Papers [1] and [2] are far from the only research that has been conducted into the complexity of finding Nash equilibria in recent years. A number of these have focused on "proper" equilibria, which were first defined in [11]. Proper equilibria are essentially a subset of Nash equilibria, where for some small value ε, the probability assigned by a player to the worse of two mixed strategies is no greater than ε times the probability they assign to playing the better one. One of these pieces of research is [12], which shows that the problem of determining, given the payoff matrices, whether or not a given pure Nash Equilibrium for a 2-player game is a proper equilibrium is **NP**-complete. This same paper also looks into the complexity of Polymatrix games, which are games where there are separate payoff matrices for each possible pair of players, and a player's payoff is the sum of their payoffs for each individual matrix that they feature in. This paper shows that for an n-player Polymatrix game, that the problem of calculating a proper equilibrium for such a game is in the **PPAD** complexity class

Another variant of Nash Equilibria are relative ε-Nash Equilibria, which are strategy profiles where the incentive to deviate from that profile is no greater than some factor ε. One piece of research on this topic is found in [13], which shows that the problem of calculating a relative ε-Nash Equilibrium in a 2-player game is **PPAD**-complete. Another piece of research in this field that did not focus on Proper Equilibria is found in [14], which proves that for any 2-player game, where each player has n available actions, the problem of finding an ε-Nash equilibrium is in the **PPAD** complexity class, and can be computed in quasi-polynomial time.

Research more directly related to [1] and [2] has also been made, for example [15], which at least partially answers a question posed at the end of [1], which asked whether or not there was a polynomial time algorithm to calculate approximate Nash equilibria. Paper [15] finds that this is the case for any "anonymous" game, which is one where the payoff is not dependent on which other players perform various actions, only on how many do so, provided that the number of actions stays constant (i.e. it does not grow as the number of players increases).

Nash equlibiria are not the only solution concepts for games. Due to limited bandwidths on the Internet, it is desirable to use game theory to model the traffic on the Internet, with each user being represented by a selfish agent in the game (each user wants the quickest browser which will act to gain as much bandwidth as possible) [16]. A congestion game is the perfect

setup for this, and as discussed already, the general case of finding a pure Nash equilibrium in such games is **PLS**-complete. This complexity is unfortunate when aiming to use this in real-time to direct network traffic. There are also other issues with Nash equilibria in this setting which make finding an alternative worthwhile.

When there is limited information provided to the players of the game, Nash equilibria alone cannot completely capture the decision process agents must make [16, 17]. Source [17] presents a solution to this when the players are not aware of all the moves called an augmented game. This involves modelling the game in different "views" so that the unawareness can be captured. However, this is of limited use in the domain of Internet traffic; it becomes increasingly complex as more players and layers to the game build up, and it requires the viewpoint of an omniscient modeler, which would be computationally difficult to create in a distributed setting [16]. Another way to mitigate this issue is to abandon the traditional idea of a Nash Equilibrium entirely, and instead study the convergence behaviours of learning agents [16]. The simplest of these, called a "stage learner", learns in separate phases and plays the actions which had the highest average in the last phase with a higher probability in the current one. This key nondeterministic property allows the agent to be responsive to changes in the environment, which is vital for changing network traffic. Such learners are shown to have good convergence behaviour which leads to a new equilibrium notion as the convergence of a set of learners. More recently, this line of research has been joining with advances in reinforcement learning such as in [18], and some researchers have even proposed computing Nash equilibria at every step to determine the optimal action strategy for learning agents [19].

Nash equilibria can often give unfavourable solutions. As already outlined, a well-known instance of this is the Prisoners Dilemma. The only Nash equilibrium is for both players to fink, which results in a lower payoff for both players than cooperation. This was first conceptualised as Braess' paradox [20], where adding more roads to a network can in fact slow the traffic flow. This kind of behaviour is extremely undesirable when attempting to improve the speed of network traffic. A strategy which can mitigate this is tit-for-tat, which improves the cooperability of agents in repeated games, which the transmission period of internet traffic can be modelled as. Here the agent starts out cooperating and then plays the action played by the other agent in subsequent stages [17]. However, this is not a Nash equilibrium unless the game is infinitely repeated. This is not true for any real-life game, so the Nash concept is not enough. Tit-for-tat was shown to be successful in Axelrod's tournament [21], and is even used as part of BitTorrent's file sharing to optimise download speeds [22]. Therefore, it can be useful to analyse strategies which do not directly result in a Nash equilibrium.

References

[1] Constantinos Daskalakis, Goldberg, Paul, W., and Papadimitriou, Christos, H. "The Complexity of Computing a Nash Equilibrium". In: *SIAM Journal on Computing* 39.1 (2009), pp. 195–259.

[2] Alex Fabrikant, Papadimitriou, Christos, H., and Kunal Talwar. "The Complexity of Pure Nash Equilibria". In: (2004), pp. 604–612.

[3] Osborne, Martin, J. and Ariel Rubinstein. *A Course in Game Theory*. Cambridge, Massachusetts: The MIT Press, 2011. ISBN: 0-262-15041-7.

[4] Nash, John, F. "Equilibrium points in n-person games". In: *Proceddings of National Academy of Sciences of the United States of America* 36 (1950), pp. 48–49.

[5] Avi Wigderson. *Mathematics and Computation: Ideas Revolutionizing Technology and Science*. Princeton University, 2019. ISBN: 978-0-691-18913-0.

[6] Michaela Borzechowski. "The Complexity Class Polynomial Local Search (PLS) and PLS-complete problem". PhD thesis. Berlin, 2016.

[7] Papadimitriou, Christos, H. *On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence*. 1994.

[8] Robert W Rosenthal. "A class of games possessing pure-strategy Nash equilibria". In: *International Journal of Game Theory* 2.1 (1973), pp. 65–67.

[9] Dov Monderer and Lloyd S Shapley. "Potential games". In: *Games and economic behavior* 14.1 (1996), pp. 124–143.

[10] Alejandro Schaffer and Mihalis Yannakakis. "Simple Local Search Problems That are Hard to Solve." In: *SIAM J. Comput.* 20 (Feb. 1991), pp. 56–87. DOI: 10.1137/0220004.

[11] Myerson, R., B. "Refinements of the Nash equilibrium concept". In: International Journal of Game Theory (1978), pp. 73–80.

[12] Hansen, Kristoffer, Arnsfelt and Lund, Troels, Bjerre. "Computational Complexity of Proper Equilibrium". In: *Conference on Economics and Computation* (2018).

[13] Constantinos Daskalakis. "On the Complexity of Approximating a Nash Equilibrium". In: *ACM Transactions on Algorithms* 9.3 (2013), pp. 1–33.

[14] Aviad Rubinstein. "Settling the complexity of computing approximate two-player Nash equilibria". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 258–265.

[15] Constantinos Daskalakis and Papadimitriou, Christos, H. "Approximate Nash Equilibria in Anonymous Games". In: *Journal of Economic Theory* 156 (2015), pp. 207–245.

[16] Eric J Friedman. *Learning and Implementation on the Internet*. 1998.

[17] Joseph Halpern. *Beyond Nash Equilibrium: Solution Concepts for the 21st Century | SpringerLink*. 2011.

[18] Ann Nowe, Peter Vrancx, and Yann-Michaël De Hauwere. "Game Theory and Multi-agent Reinforcement Learning". In: *Adaptation, Learning, and Optimization*. Journal Abbreviation: Adaptation, Learning, and Optimization. Jan. 2012, p. 30. ISBN: 978-3-642-27645-3. DOI: 10.1007/978-3-642-27645-3_14.

[19] Euijong Lee, Young-Duk Seo, and Young-Gab Kim. "A Nash equilibrium based decision-making method for internet of things". en. In: *Journal of Ambient Intelligence and Humanized Computing* (June 2019). ISSN: 1868-5145. DOI: 10.1007/s12652-019-01367-2.

[20] Dietrich Braess, Anna Nagurney, and Tina Wakolbinger. "On a Paradox of Traffic Planning". en. In: *Transportation Science* 39.4 (Nov. 2005), pp. 446–450. ISSN: 0041-1655, 1526-5447. DOI: 10.1287/trsc.1050.0127.

[21] Robert M. Axelrod and W. D. Hamilton. *The evolution of cooperation*. New York: Basic Books, 1984.

[22] Bram Cohen. "Incentives build robustness in BitTorrent". In: *Workshop on Economics of PeertoPeer systems* 6 (June 2003).