The Computational Complexity of Nash Equilibria

Janet Rosenbaum

September 15, 2002

1 Abstract

Although the computation of Nash equilibria for general games is of unknown complexity, there exist many algorithms for specific game classes, some of which are a marked improvement of previous algorithms. This paper reviews general results on the computational complexity of Nash equilibria and discusses the major algorithms for specific game classes.

2 Introduction

Game theory has grown in importance as academics from all fields have realized the importance of strategy in domains of human behavior ranging from business to the Internet. The computational issues of game theory are important to both a game's participants and designers. A participant in a game, or their consultant, obviously wants to compute their optimal moves in anticipation of their opponents' likely moves. Likewise, someone who wishes to model a situation of conflict with a game needs to calculate its equilibrium in order to test their model's correctness. In addition, a game designer may wish to use computational intractability to their advantage; where players are unable to design a strategy by computing possible outcomes of the game, they have no incentive not to act according to their own preferences, as might be desired by the game's designer. For example, although no non-dictatorial voting scheme in an election of more than two candidates is strategy-proof,¹ in actuality, the computation needed to determine the ideal strategic vote may not be possible if the election is sufficiently large and the computation sufficiently intractable.

¹A dictatorial election is an election where one person's vote entirely determines the outcome of the election. The Gibbard-Satterthwaite theorem proves that in an election with more than two candidates, all other voting schemes are vulnerable to strategic voting — that is, voters can potentially gain by voting according to something other than their preferences.

Games are most commonly represented in normal or extensive form. Normal form is simply a matrix of payoffs used for static games, games in which players move simultaneously. Extensive form is a tree representing players' sequential moves in dynamic games, games in which players move in sequence.

The Nash equilibrium (NE) is a solution concept, defined as the strategy of best response for each player: $\vec{s} = (s_1, \dots, s_n) \in \mathbf{S}$ such that for each player i ,

$$P(s_1, \dots, s_i, \dots, s_n) \geq P(s_1, \dots, s'_i, \dots, s_n)$$

That is, the NE is the strategy from which no player would choose to deviate from their strategy even with knowledge of the other players' moves. Each player's strategy may be a pure strategy or a mixed (randomized) strategy. A mixed strategy is a probability distribution over a player's possible moves (those of non-zero probability are called the support of the mixed strategy). The pure strategy NE is found by determining, for each player, that player's strategy of best response to each of their opponents' strategies. The mixed strategy NE is found by determining which probability distribution of each player would cause their opponents to be indifferent between their own strategies — i.e., that none of the opponent strategies could do better than any other given the player's mixed strategy. Finding mixed strategy NE is thus an optimization subject to the (linear, in the two-player case) constraints determined by the opponents' strategies.

We define a subgame of a game in extensive form to be any subtree of the game in extensive form. The subgame perfect Nash equilibrium (SPNE) is the NE for which the players' strategies are a NE in every subgame. More intuitively, it is the NE found by backwards induction on the tree of payoffs; the player who moves last determines her preferred choice for each of her decision nodes, the player before determines his preferred decision given the last player's preferred moves, etc.

The Nash equilibrium, as defined above, has many variations. Harsanyi extended the notion of Nash equilibrium to Bayesian games, games in which each player has been assigned a "type" comprising private information. Players have information about the distribution of types, but do not know their opponents' types. The Nash equilibria of these games is called the Bayes-Nash equilibrium, the set of strategies from which no player has an incentive to deviate given each of the player's possible types. The Bayes-Nash equilibrium is the best

response to the opponents' average strategies, given the distribution of types.

A game may not have a pure Nash equilibrium if the strategies S_i are not convex sets, but it will always have one in mixed strategies, so this problem is not serious. The strategies involved in a mixed NE are called its support.

The concept of Nash equilibrium is not perfect. The Folk theorem reveals that a game repeated arbitrarily many times has infinitely many Nash equilibria, and even a finitely repeated game may have many Nash equilibria. It only reveals one or more “best” ways of playing a game, but does not provide any way of distinguishing between the strategies. Because the number of NE grows very quickly in the number of players and strategies per player, it is important to have some means of distinguishing between thousands of strategies. Refinements such as elimination of weakly dominated strategies may be added to Nash equilibria to make the idea more useful.

Even still, in some cases the obvious “solution” is not part of any Nash equilibrium, and so other solution concepts are necessary.² For example, take Rosenthal's centipede,³ an alternating game of two players where at each stage each player has the choice between Accepting (right arrow) and Defecting (down arrow).

The payoffs for defecting at each stage expressed as (diamond payoff, heart payoff) are:

$$(1, 1), (0, 3), \dots (98, 98), (97, 100), (99, 99), (98, 101)$$

By backward induction, we can derive that the SPNE is to defect on the first move. Indeed, any Nash equilibrium of the above game involves defecting in the first move, and yet any sensible person would see that it is best to attempt to try to establish trust with one's opponent in order to get to the larger payoffs at the end of the centipede.

Likewise, the Nash equilibria of the following game runs against common sense of how the game ought to be played.

	t1	t2	t3	t4
s1	200,6	3,5	4,3	0,-1000
s2	0,-10000	5,-1000	6,3	3,20

²The following two examples are taken from Kreps, chapter 12.

³see Kreps for the picture form.

The solution concept which works better than the NE is playing in prudent strategies, in which each player chooses the strategy which guarantees them the maximum of their minimum payoffs.

Despite these flaws which emerge in strange cases, the Nash equilibrium is a useful solution concept for most games. In the majority of cases, it is not an unreasonable assumption that players of a game will achieve a NE by playing rationally.

3 General Complexity results

Given the importance of the computation in game theory for both participants and designers of games, it is vital to know the complexity of computation for finding central features of the games. Papadimitriou (2001) notes that finding the complexity of these problems is one of the two most important issues to computer science today. The complexity of computing Nash equilibria in the general case is unknown, but a number of results indicate that finding specific Nash equilibria, such as the NE which maximizes payoff, is intractable.

Gilboa (1988) found that finding the best response strategy (represented by a finite stage automaton), given a game and the opponents' strategies, is in P as long as the number of players is fixed. In spite of this result, Papadimitriou (2001) opines that it is fairly certain that the computation of a Nash equilibrium (the best response for all players) is not in P. At the same time, because Nash's theorem guarantees the existence of a solution for every game, Papadimitriou concludes that it is unlikely that computing the NE could be in NP. Megiddo (1988) concurs due to his proof that computing a Nash equilibrium for a two player game is in NP implies that $NP=coNP$, which is believed to be unlikely.

Gilboa (1988) also shows that for a fixed number of players, it is possible to verify in polynomial time whether a given strategy is a best response to the opponents' strategies. For an arbitrary number of players, he shows that if verifying that a given solution were a best response was solvable in polynomial time, it would be possible to find a polynomial time algorithm for showing the existence of a Hamiltonian cycle in a given graph (and hence any NP-complete problem, which is believed unlikely). He also shows that for an arbitrary (non-fixed) number of players, there is no polynomial time algorithm for finding a best response strategy.

Gilboa and Zemel (1989) proved that in the worst case, the standard methods of finding

Nash equilibria of n -player games with particular characteristics are NP-hard, such as finding the NE with the maximal payoff, finding the NE of maximal or minimal support, finding a NE consisting of a particular subset of strategies or containing a particular subset of strategies, and also proving uniqueness of a NE. In the two-player case, these problems are all in NP, except for uniqueness, which is in Co-NP. The complexity of these problems in the average case is unknown, however, as is the complexity of finding any NE for a game.

Conitzer and Sandholm have a number of general results: counting Nash Equilibria is $\#P$; showing the existence of NE subject to certain constraints is NP-hard; likewise, determining whether pure-strategy Bayes-Nash Equilibria exist is NP hard, and showing the existence of a pure strategy NE for a stochastic game is PSPACE-hard.⁴

These general results seem to show that finding NE for general games is likely intractable. Nonetheless, there are many methods for specific classes of games as well as improvements on classic solution methods which are more tractable.

4 Finding Nash Equilibria

There are many methods of finding NE which depend on how many equilibria are desired (one or all), the number of players in the game (two or more),⁵ the desired refinements on the Nash equilibrium (e.g., dominant, Pareto efficient, etc.)⁶, and whether the game is in strategic or extensive form.

Games are commonly represented in two forms: strategic/normal, where the game takes the form of a matrix of payoffs, and extensive, where the game takes the form of a tree where each layer of the tree is one player's move. Because the strategies in the normal form game must be entirely deterministic and so list player 2's moves in response to each possible move of player 1 in sequence, and vice-versa, transforming an extensive-form game into normal form results in a significantly larger game than the extensive form game itself. Consider a game where player 1 (who moves first) may move Up or Down and player 2 may move Left or

⁴Tangentially, a classic game theoretic result which is PSPACE-hard is that of Chandra (Chandra et al, Journal of the ACM, 28, 114–133 quoted in Feigenbaum et al 1995) who showed that the duration of a two-person perfect information games is a polynomial function of the description of the initial position. This game is used as an intuitive illustration for the meaning of PSPACE.

⁵We will call a game with more than two players a multi-player game.

⁶The best summary of methods of finding NE is MM 1996, which is summarized in this section unless another source is otherwise noted.

Right. Player 2's strategies are not merely "left" and "right", but LL, RR, LR, RL — that is, always left, always right, left only if player 1 moved "up", and left only if player 1 moved "down". Giving player one another possible move so that he may move Up, Middle, or Down makes player 2's strategies 3 characters long. Since player 2 has only two possible moves (left or right), she only has 2^3 strategies, but if she had k moves and player 1 had n moves, she would have k^n strategies and the normal form matrix would be $n \times k^n$. By contrast, the extensive form tree would have only nk leaves — n branches at the first level, and k branches off each of the n . The discrepancy between extensive and normal form grows significantly in games of more than two players. While a game may be transformed from normal to extensive form in linear time, transforming from extensive to normal form is an exponential function of the size of the game tree (Gilboa 1988).

An algorithm for a game in extensive form could be applied to a game in normal form since the normal form matrix could be transformed to extensive form easily; the reverse is not true since according to Gilboa's result, the transformation from extensive form into normal form is already exponential. Thus, there are separate algorithms for games in the two forms, and the complexity results for the two forms may be different because matrices are not transformed easily between the two forms.

4.1 Two-player games in normal form

The NE of two-player games are much easier to find than for multi-player games in a number of respects. If the payoff matrix for a two-player game is rational (in $\mathcal{Q}$), the NE are a union of convex polyhedra with rational vertices. Even if the payoffs are not rational numbers, the NE is an optimization problem with linear constraints and the NE are still a union of convex polyhedra. By contrast, solutions to multi-player games are not necessarily connected sets or have rational coordinates, even with rational payoff matrices.

Many methods for finding all NE and exact NE have been developed for two-player games. Finding a single sample equilibrium has been studied for many years, and there are many known procedures. Recall from above that it was shown that finding a NE with specific characteristics seems hard, but the complexity of finding one NE is unknown. Although two-player games are far simpler than multi-player games, finding NE even for a two-player game could be hard. Since the NE is a best response to the opponent's mixed strategies,

the equations for finding the NE can be expressed as an LCP with constraints based on the opponent's payoffs. Thus, general methods of optimization in other branches of math can be applied to the problem of finding NE.

If only one NE is desired, it is possible to decrease the size of the game matrix by eliminating weakly dominated strategies.⁷ If all NE are desired, weakly dominated strategies cannot be eliminated without potentially losing NE of the original game. In either case, strongly dominated strategies can be eliminated from a game matrix because the NE of the reduced matrix are the NE of the original matrix.

If the payoffs of the two-player game are rational, the NE are convex polyhedra with rational vertices, so it's possible to find them exactly. Even without rational payoffs, the NE of a two-player game is a convex set, so it is possible to find all NE exhaustively. If there are more than two players, even with rational payoffs the Nash equilibria are not necessarily a connected set or have rational coordinates, so finding even a sample equilibrium is much more difficult.

The classical method of finding NE is due to Lemke and Howson (1964). They transformed a strategic-form game into a linear complementary problem. Their method for the solution of this LCP was the first to provide a constructive proof of the existence of NE in the two-player case. Prior to the Lemke-Howson algorithm, there was no numerical method to calculate NE for games other than the most simple. As mentioned above, for rational payoffs this algorithm gives exact solutions, so this algorithm gives a guaranteed solution. The computational complexity of the Lemke algorithm is unknown for the two-person case, but in the general case it is NP-complete (Murty 1988). In practice, the problem is only exponential in exceptional cases. If the game is zero-sum, however, it can be expressed as a linear program which can be solved in polynomial time (Murty 1988).

To be more concrete, an LCP is defined as following: Given an $n \times n$ matrix M and a vector $\vec{q} \in Re^n$, find vectors $\vec{w}, \vec{z} \in Re^n$ such that

$$w - Mz = q, w_i \geq 0, z_i \geq 0, w_i z_i = 0 \text{ for all } i$$

Two-player games generate equations linear in their support (the strategies played in the

⁷We say that a strategy is strongly dominated if its payoffs are strictly less than the payoffs of another of the player's strategies. A strategy is weakly dominated if its payoffs are less than or equal to the payoffs of another strategy.

mixed strategy with non-zero probability), so the mixed strategy will always be a rational combination of strategies, which provides a bound on finding the coefficients.

Each player's payoff matrix undergoes a linear transformation

$$A = \alpha - A', B = \beta - B'$$

so that the entries are non-negative and inversely proportional to the player's payoffs (Murty 1988). We say that an individual player's strategy is a probability distribution over all her strategies. If player one has strategy $\vec{x} \in [0, 1]^m$ and player two has strategy $\vec{y} \in [0, 1]^n$, we say that $(\vec{x}, \vec{y})$ is a Nash equilibrium if

$$\vec{x}^T A \vec{y} \leq \vec{x}^T A \vec{y} \vec{x}^T B \vec{y} \leq \vec{x}^T B \vec{y}$$

for all alternative strategies x and y . Since the elements of these strategies are all less than one, we can write these conditions as

$$\vec{x}^T A \vec{y} \leq A_i \vec{y} \vec{x}^T B \vec{y} \leq \vec{x}^T B_j$$

for all $i = 1 \dots m$ and $j = 1 \dots n$ or

$$\vec{x}^T A \vec{y} e_m \leq A \vec{y} \vec{x}^T B \vec{y} e_n \leq \vec{x}^T B$$

where the e s are appropriately sized vectors of ones. We create slack variables u, v for the inequalities and define

$$\zeta = \frac{\vec{x}}{\vec{x}^T B \vec{y}} \eta = \frac{\vec{y}}{\vec{x}^T A \vec{y}}$$

so that the NE of the game is the pair $(\vec{x}, \vec{y})$ such that

$$\begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \zeta \\ \eta \end{pmatrix} = \begin{pmatrix} -e_m \\ -e_n \end{pmatrix}$$

where

$$\begin{pmatrix} u \\ v \end{pmatrix} \geq 0, \begin{pmatrix} \zeta \\ \eta \end{pmatrix} \geq 0$$

$$\begin{pmatrix} u \\ v \end{pmatrix}^T \begin{pmatrix} \zeta \\ \eta \end{pmatrix} = 0$$

Once this equation is solved, we map back to our original $(\vec{x}, \vec{y})$ to get the NE.

If the matrix M of the game's LCP is positive semi-definite, the LCP can be solved in polynomial time. That is, if the determinants of the minors of M are all non-negative,⁸ the LCP can be solved in polynomial time. I have not been able to find a non-trivial description of the class of games for which this is true, so it is possible that these games don't share a common element other than positive semi-definiteness.

There are a number of software solutions for NE. Specialty software for solving linear programming programs as well as software such as Mathematica can be used for solving games once they have been transformed into LCPs. The C code of MM96 could solve an 8×8 game in 30 seconds on a 486 machine, and could work with games up to 12×12 . In addition, there is software designed specifically for solving games. Gambit⁹ is software which uses both classic and new algorithms for finding all NE of multi-player games in normal and extensive form. Gambit is very limited by the exponential complexity of existing algorithms for finding NE, and thus any user of the Gambit system must be careful to use the routines for eliminating dominated strategies and finding subgames (of an extensive-form game) before asking Gambit to solve the game. The Gambit documentation notes, however, that eliminating dominated strategies is itself computationally intensive. Gambit represents all games in matrices of $|S|^N$ elements¹⁰ and is not capable of exploiting sparsities and symmetries of payoff matrices by reducing the size of the representation. Thus, it is much slower in such cases than it would be if it would reduce the size of the representation — for even 5 players with 5 strategies each, Gambit is unable to find a solution after running for several days.

Finding a sample equilibrium for NE refinements (i.e., NE which satisfy particular criteria, such as dominance or maximizing payoff) is difficult; generally, NE refinements are found by first finding all NE and only then imposing refinements on the set of all NE.¹¹

4.2 N-player games in normal form

For games with more than two players, the equations are no longer linear in the support, so the Lemke-Howson algorithm on linear complementary problems is no longer applicable. However, results for LCPs can be generalized to methods for solving Brouwer fixed points,

⁸Alternatively if for non-zero vector t $t'Mt \geq 0$

⁹Available at <http://www.hss.caltech.edu/gambit>

¹⁰ $|S|$ = number of strategies, N = number of players

¹¹It is obviously possible to eliminate pure dominated strategies before finding the NE, but finding dominant mixed NE requires all NE to be found first.

methods which are applicable in solving N-player games. As mentioned above, the solutions are not necessarily rational, so exact solutions cannot be found. Multi-player games induce a non-linear complementary problem, for which there are approximation methods. The non-linear complementary problem is constructed the same way as the linear complementary problem is, but the extra interactions make the problem nonlinear. That is, in the case of three players, each player's payoff matrix is three-dimensional. If our three players have strategies x, y, z and payoff matrices A, B, C the initial inequalities are:

$$\bar{x}A\bar{y}\bar{z} \leq xA\bar{y}z\bar{x}A\bar{y}\bar{z} \leq \bar{x}A\bar{y}z\bar{x}A\bar{y}\bar{z} \leq \bar{x}A\bar{y}z$$

where the matrix multiplication occurs in three-dimensions.¹²

Mathiesen¹³ approximates each non-linear complementary problem with a sequence of LCPs beginning with the first-order Taylor expansion around an arbitrarily chosen point and continuing until it converges. Mathiesen's SLCP method has the same advantages and disadvantages of Newton's method (which it is based on) — it is not guaranteed to find a solution for every initial point, but it is a fast method of finding an approximate solution.

Another non-globally-convergent approximation method is that of Van den Elzen and Talman¹⁴ who approximate the non-linear stationary point problem by a series of linear stationary point problems.

There are solution methods for N-player games which are globally convergent. Scarf's simplicial subdivision algorithm¹⁵ returns to the beginnings of game theory, using Brouwer's fixed point theorem, as in Nash's original proof of the existence of NE; the algorithm is an algorithm for finding fixed points on a compact set. Scarf constructed a continuous function such that its fixed points are the NE, and the algorithm simply iterates until it finds the fixed points. Hirsch, Papadimitriou, and Vavasis (1989)¹⁶ showed that in the worst case, a method for computing Brouwer fixed points based on evaluating function values was exponential in

¹²This notation may well violate the conventions of multiplication of $n \times m \times l$ matrices.

¹³Mathiesen L, An algorithm based on a sequence of linear complementary problems applied to a Walrasian equilibrium model: an example, *Mathematical Programming*, 37 (1987), 1–18, cited in MM96

¹⁴van den Elzen AH and Talman AFF, Finding a Nash equilibrium in non-cooperative n-person games by solving a sequence of linear stationary point problems, *ZOR [Zeitschrift für Operations Research]: methods and models of Operations Research*, 35 (1994), 27–43, quoted in MM96

¹⁵Scarf H (1973) *Computation of Economic Equilibria*, New Haven: Yale University Press, first formulated in a 1967 paper, The Approximation of fixed points of a continuous mapping, *SIAM Journal of Applied Mathematics*, 15: 1328–1343, quoted in MM96

¹⁶Hirsch MD, Papadimitriou CH, Vavasis SA, Exponential lower bounds for finding Brouwer fixed points, *Journal of Complexity*, 5: 379–416, cited in MM96

the number of players and digits of accuracy. Megiddo and Papadimitriou (1989) proved that computing Brouwer fixed points for a continuous function f with Lipschitz constant 1 [i.e., $|f(x) - f(y)| \leq |x - y|$] is a problem in an intermediate class between P and NP, which they call TFNP. The HPV89 paper does not demonstrate an exponential lower bound for Brouwer fixed point evaluation. According to MM96, a continuously differentiable function which satisfies a Lipschitz condition can converge as a quadratic function in the number of players assuming the initial conditions are chosen appropriately. Thus, the Scarf method could be quadratic in the best cases.

Scarf's algorithm is highly dependent on the precision of the fixed points; if a higher degree of precision is required, the algorithm must be run again. There are a number of methods which solve this problem of precision, but are not otherwise a substantial departure from Scarf's algorithm.¹⁷ Scarf's algorithm does have the advantage that it is globally convergent.

Another method which uses geometric methods is that of McKelvey¹⁸ which views the NE as the minima of a function derived from the NE conditions on a polytope.¹⁹ They define a non-negative everywhere-differentiable function constructed so that its zeroes are NE, and use optimization methods to find the minimum closest to a user-provided starting point. This method is guaranteed to converge to an NE only if the starting point is placed sufficiently close to the NE, but the NE which is found is that closest to the starting point; other methods will not necessarily find the NE closest to the starting point. In theory, it would be possible to use this method to find all NE. Empirical tests have shown that this method is slower than the above, but they did not compute the theoretical complexity. Alternatively, it is possible to view the NE as solutions of differential equations based on the functions which generate the polytope.

Only one of the above geometric methods is globally convergent — that is, finding Nash equilibria requires strategic choice of initial points since not every initial point converges to a NE. Herings and Peeters (2001) created an algorithm to implement Harsanyi's tracing proce-

¹⁷See MM96 section 3.2.2 for details.

¹⁸McKelvey RD (1992), A Liapunov function for Nash equilibria, mimeo, California Institute of Technology, cited in MM96

¹⁹A polytope $\mathcal{Q}$ is the convex hull of a finite set of points.

$$CH(x_i) = \bar{x} : \bar{x} = \sum_i \alpha_i x_i, \sum_i \alpha_i = 1, \alpha_i \geq 0$$

dure (1975)²⁰. for an everywhere-differentiable homotopy, avoiding computational problems of the Harsanyi-Selten algorithm²¹ which required that the homotopy be updated at each step. Herings and Peeters create a differentiable everywhere homotopy by transforming the variables of Harsanyi's approach, and use prepackaged software to trace the curve of the homotopy until it reaches an equilibrium. The computational trials of this algorithm shows that computation time increases more in the number of players than in the number of strategies; this result is not surprising given that the expected number of NE increases dramatically with the number of players, as we see below.²²

strategies\players	2	3	4	5
2	1.3	2.2	4.5	7
3	1.5	3.8	18	82
4	1.8	12.7	82.5	879
5	2.6	27.2	440	
6	2.6	65.7	2037	
7	4.6	152		
8	5.5	356		
9	6.7			
10	8.6			

The hardest case they tried was for five players with four strategies each, which took 519 seconds (with standard deviation 333.) They did not derive the theoretical complexity of their procedure.

5 Extensive Form Games

While games in normal form can be transformed to extensive form in polynomial time, the reverse transformation from extensive to normal is exponential in the number of players and strategies. Thus, the normal-form algorithms discussed above cannot be directly applied to extensive-form games, and it is necessary to develop algorithms which solve extensive-form games.

In particular, the possibility of transforming a normal form games into linear optimization

²⁰Harsanyi JC, The tracing procedure: a Bayesian approach to defining a solution for n-person non-cooperative games, *International Journal of Game Theory*, 4:61–94, quoted in HP01

²¹Harsanyi JC, Selten R: *A general theory of equilibrium selection in games*, Cambridge: MIT Press, 1988

²²McLennan A: The expected number of Nash equilibria of a normal form game. Mimeo (1999), quoted in Herings and Peeters (2001)

problems makes it very easy to apply standard algorithms and software packages to normal form games. By contrast, the literature on extensive form games has only developed recently, so there are no standard software or algorithms

For zero-sum extensive-form games, Koller and Megiddo (1992) find an algorithm polynomial in the size of the game tree for computing saddlepoints²³ for a zero-sum two-person extensive-form game with perfect recall; with imperfect recall computing any prudent strategy is NP-hard. They compute the maxmin behavior strategies in terms of a linear programming problem with exponentially many constraints. However, it is possible in polynomial time to test whether a given point violates one of the constraints, and in linear time to find the best response to the mixed strategy of the other player. Thus, a polynomial time algorithm exists to compute the saddlepoints of a two-person zero-sum game.

Koller, Megiddo, and von Stengel (KMS) (1994) create a way to apply Lemke and Howson's result to zero-sum extensive form games. They transform an extensive form game to the sequence form, a listing of the moves in order, which can be done in linear time (see von Stengel 1996). The sequence form then induces an LCP which can then be solved with Lemke's algorithm. In our above example with player 1 moving Up or Down and player 2 moving Left or Right, the sequences for player 1 would be u, d, and the empty set, and while player 2 has the sequences L, R, and the empty set. Von Stengel (1996) shows that a zero-sum game in extensive form can be put into sequence form, which can be made into an LP linear in the size of the game tree and then solved in polynomial time. The Lemke algorithm isn't guaranteed to be polynomial — in fact, it is NP-complete in general (Murty 1988) — but the sequence form is generally more compact than the normal form and so the Lemke algorithm applied to the sequence form is more efficient.

The above algorithms only apply to zero-sum games, but there are also algorithms for non-zero-sum extensive-form games. Von Stengel et al (2001) provide an algorithm for the solution of two-person non-zero-sum games in extensive form with perfect recall. They use the sequence form, but use a path-tracing algorithm similar to that of Herings and Peeters above for computing NE for normal form games. As above, this algorithm requires the starting

²³i.e., the pure or mixed strategies such that the minimum payoff for both players is simultaneously maximized. Every zero-sum game has at least one saddlepoint in either pure or mixed strategies, and this saddlepoint is a Nash equilibrium. The strategy for each player which maximizes their minimum payoff is called a prudent or maxmin strategy.

point to be chosen appropriately. In addition, they do computational experiments and show that their algorithm is more tractable than the normal form, as predicted by the relative sizes of the sequence and normal representations.

Koller and Pfeffer (1997) created a system (Gala) to encode imperfect information games. Gala has two aspects. Gala can be used as a Prolog-based language which encodes game parameters, such as the players, moves, and move sequence. This encoding can be used to generate a game tree which can then be passed to the game-solving software GAMBIT. Alternatively, Gala can encode the tree into sequence form and solve it on its own using the KMS algorithm; this function is perhaps obviated by the fact that the new release of GAMBIT has begun using the KMS algorithm for solving extensive-form games. Their computational trials using poker and the arm-control inspection game show that the KMS algorithm is exponentially faster than the previous algorithms for these games in both theory and practice.

Koller and Milch (2001) introduced multi-agent influence diagrams (MAIDs), directed graphs which show relationships between decisions, variable dependencies, and utility payoffs. Their algorithm gives only the sub-game perfect Nash equilibrium, which is only a subset of all the Nash equilibria. The MAIDs can be transformed into extensive form games, but can also be solved directly by breaking them into their component parts. Solving several smaller component games of a multi-level game is vastly faster than solving the entire multi-level game at once. Koller and Milch give an example game whose tree is exponential in the number of players. Previously, this game couldn't be feasibly solved for more than three players; however, the MAID representation and computational time is linear in the number of players, and even the case of 40 players can be computed in under nine minutes.

As impressive as these results are, decomposition requires that the game's MAID is decomposable. Kim (2002) proves a number of theorems about the characteristics a MAID must contain in order to be decomposed: a MAID works best for a game of many agents with few interactions, so that the graph is sufficiently sparse; likewise, MAIDs cannot be decomposed for games with few players and many strategies. MAIDs do not offer significant computational advantages for games with characteristics such as perfect recall, which are common in economic games. Kim also points out that MAIDs are designed for dynamic games, in which players move in sequence, but do not work for static games.

The model of Kearns, Littman, and Singh (2001a) creates another graph-based repre-

sentation; their model uses a nondirected graph to model interactions between players in a static game. Players are represented by vertices with corresponding payoff matrices, and their interactions form the edges of the graph. They define two algorithms applicable to games which can be expressed as trees. The first approximates every NE in polynomial time of the tree and payoff matrices. The second finds exact solution and is exponential in the size of the graph. Recall that because the solutions of multi-player games are irrational, there are no finite-time algorithms which can find exact solutions for general games. Like the MAID model, the KLS model requires that the graph be sufficiently sparse to admit decomposition, and so works only for a large number of players with few dependencies.

In contrast to their above paper, Kearns, Littman, and Singh (2001b) describe an algorithm which computes a single exact NE for multi-player games where each player has two strategies. They model a game as a non-directed graph in which the players are vertices and their interactions are edges. Associated with each vertex is that player's payoff matrix. For games which can be represented as trees, their algorithm gives an exact solution as a polynomial function of the number of players; however, the algorithm is exponential in the number of neighbors each player interacts with. This algorithm is applicable to cases where, for example, players are geographically constrained, as in computer networks or salespeople. This algorithm would not apply to a tournament of iterated prisoners' dilemma, a game with wide applicability, as described in Axelrod (1984).

Kearns and Mansour (2002) also attempt to simplify computation by introducing a more compact representation for certain games. Their compact representation requires no player has overriding influence on the outcome and that the games' outcome can be represented by linear summary function, i.e., where the players care about the overall outcome, but not their opponents' individual actions. Their example of such a game is voting; another possible game would be a cartel of many agents. They create an algorithm which can find approximate pure NE as a polynomial function of the number of players and approximation precision. While this polynomial time algorithm is definitely an important advance, it's applicability is not clear; Kearns and Mansour do not give examples other than voting.

6 References

Axelrod R (1984), *Evolution of Cooperation*, New York: Basic Books

- Conitzer V and Sandholm T (2002), Complexity Results about Nash Equilibria, CMU-CS-02-135, May 2002
- Feigenbaum J, Koller D, and Shor P (1995), Game-Theoretic Classification of Interactive Complexity Classes, Proceedings of the 10th Annual IEEE Conference on Structure in Complexity Theory, Minneapolis, Minnesota, June 1995, 227–237.
- Gilboa I (1988), The Complexity of Computing Best-Response Automata in Repeated Games, *Journal of Economic Theory*, 45:342–352.
- Gilboa I and Zemel E (1989), Nash and Correlated Equilibria: Some Complexity Considerations, *Games and Economic Behavior* 1, 80–93.
- Govindan S and Wilson R (1998, revised 2001), A Global Newton method to Compute Nash Equilibria, Working Paper
<http://faculty-gsb.stanford.edu/wilson/pdf>
- Herings PJJ and Peeters JAP (2001), A differentiable homotopy to compute Nash equilibria of n-person games, *Economic Theory*, 18: 159–185
- Kalai E and Stanford W (1988), Finite Rationality and Interpersonal Complexity in Repeated Games, *Econometrica*, 56(2): 397–410
- Kalai E (1995), Games, Computers, and O.R., Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, 468–473.
- Kearns M, Littman S, Singh S (2001a), Graphical Models for Game Theory, Proceedings of UAI 2001.
<http://www.cis.upenn.edu/~mkearns/papers/graphgames.pdf>
- Kearns M, Littman M, Singh S (2001b), An Efficient Exact Algorithm for Singly Connected Graphical Games, To appear, NIPS 2001.
<http://www.cis.upenn.edu/~mkearns/papers/gg-exact.pdf>
- Kearns M and Mansour Y (2002), Efficient Nash Computation in Large Population Games with Bounded Influence. To appear, Proceedings of UAI 2002.
<http://www.cis.upenn.edu/~mkearns/papers/summgames.pdf>
- Kim BT (2002), Modeling Games with Multi-Agent Influence Diagrams, senior thesis, Harvard College, Cambridge, MA, April 8, 2002.
- Koller D, Megiddo N, and von Stengel B (1994), Fast Algorithms for Finding Randomized Strategies in Game Trees, Proc of the 26th ACM Symposium on the Theory of Computing, 1994, 750–759
- Koller D and Megiddo N (1992), Complexity of Two-Person Zero-Sum Games in Extensive Form, *Games and Economic Behavior*, 4, 528–552

- Koller D and Megiddo N (1994), Finding Mixed Strategies with Small Supports in Extensive Form Games, later published in *International Journal of Game Theory*, 25:1, March 1996, 73–92.
- Koller D, Megiddo N, von Stengel B (1996), Efficient Computation of Equilibria for Extensive Two-person games, *Games and Economic Behavior*, 14(2): 247–259.
- Koller D and Milch B (2001), Multi-Agent Influence Diagrams for Representing and Solving Games, *Games and Economic Behavior*. (Also in Seventeenth International Joint Conference on Artificial Intelligence, Aug 2001.)
- Koller D and Pfeffer A (1997), Representations and Solutions for Game-Theoretic Problems, *Artificial Intelligence* 94(1), 167–215
- Kreps D (1990), *A Course in Microeconomic Theory*. Princeton, NJ: Princeton University Press.
- Lemke CE and Howson JT (1964), Equilibrium points in bimatrix games. *J of the Society for Industrial and Applied Mathematics* 12, 413–423.
- McKelvey, RD and McLennan A (1996), Computation of Equilibria in Finite Games in *Handbook of Computational Economics, Vol I*, eds. HM Aumann, DA Kendrick and J Rust, Elsevier, Amsterdam, p 87–142.
- Megiddo N (1988), A Note on the Complexity of P-Matrix LCP and Computing an Equilibrium, May 1988, Research Report RJ 6439, IBM Almaden Research Center, San Jose, CA
- Megiddo N and Papadimitriou CH (1989), A Note on total functions, existence theorems, and computational complexity, later published in *TCS* 81(2) 1991, 317–324
- Murty KG (1988) *Linear complementarity, linear and nonlinear programming*, Berlin: Heldermann Verlag. Reproduced at http://ioe.engin.umich.edu/books/murty/linear_complementarity_w ebbook/
- Papadimitriou CH (2001) Algorithms, Games, and the Internet, *Proc. 33rd Annual ACM Symp on the theory of Computing*, 749–753
- von Stengel B (1996), Efficient Computation of Behavior Strategies, *Games and Economic Behavior* 14, 220–246
- von Stengel B, van den Elzen A, Talman D (2001), Computing Normal Form Perfect Equilibria for Extensive Two-Person Games, draft of May 10, to appear in *Econometrica*, <http://www.maths.lse.ac.uk/Personal/stengel/TEXTE/comptrace.pdf>