USING GROEBNER BASES TO FIND NASH EQUILIBRIA

WAYNE COOK

Abstract. Game theory is a field of study that has many applications in economics. In this paper we discuss how to apply algebraic geometry techniques to problems in game theory. In particular, we use Groebner bases to determine equilibrium points of specific types of games. We explain how to solve a given system of polynomial equations and show how these arise in game theory. We give specific examples of these techniques in small games.

1. Introduction

Game theory is a study of decision making which is often applied to economics. We look at game theory and how we can apply techniques of algebraic geometry. we show how systems of polynomial equations arise when looking for equilibrium points. In this paper we introduce concepts of algebraic geometry that we apply to game theory. We also introduce basic concepts in game theory.

2. Algebraic Geometry

We will be discussing algebraic geometry and its applications in game theory, but we begin by recalling some linear algebra. The goal is to solve systems of polynomial equations. First let us recall how to solve a system of linear equations. Consider the following system:

$$\begin{aligned} x - y - 1 &= 0 \\ 2x - 3y - 12 &= 0 \end{aligned}$$

We rewrite our system as an augmented matrix

$$\begin{pmatrix} 1 & -1 & 1 \\ 2 & -3 & 12 \end{pmatrix}$$

Date: December 10, 2014.

Through Gaussian elimination we arrive at the following.

$$\begin{pmatrix} 1 & 0 & -9 \\ 0 & 1 & -10 \end{pmatrix}$$

We have a unique solution: $x = -9$ and $y = -10$. A geometric interpretation is the intersection of the lines $x - y = 1$ and $2x - 3y = 12$ at the point $(-9, -10)$. Hence what we are doing is that we are showing that we can go from a solution that's hard to a solution that is easier to visualize. Thus we take the to equations $x - y - 1 = 0$ and $2x - 3y - 12 = 0$ and convert them to $x = -9$ and $y = -10$. Therefore we can easily see that our solution is the point $(-9, -10)$.

Now let us consider a situation in which we have a system with two equations in three unknowns:

$$x - y - z = 4$$

$$2x - 3y + 4z = -5$$

The augmented matrix for our system is

$$\begin{pmatrix} 1 & -1 & -1 & 4 \\ 2 & -3 & 4 & -5 \end{pmatrix}$$

which through Gaussian elimination gives us

$$\begin{pmatrix} 1 & 0 & -7 & 17 \\ 0 & 1 & -6 & 13 \end{pmatrix}$$

In this case we have an infinite number of solutions paramatrized by t:

$$x = 17 + 7t, y = 13 + 6t, z = t$$

Similar to our earlier example, the reason we use Gaussian elimination is to make our original equations easier to solve. In this case we go from $x - y - z - 4 = 0$ and $2x - 3y + 4z + 5 = 0$ to $x - 7z - 17 = 0$ and $y - 6z - 13 = 0$. This allows us to parametrize the intersection points of our planes.

Another important concept of linear algebra which we will be using is the bases of solutions. The bases for solutions in linear algebra, specifically for linear equations, are typically in the form of one solution for each variable, and at least on variable solution in terms of the other variables. The most common algorithm find the solutions

for linear equations that we will use is Gaussian elimination. Gaussian elimination is also known as row reduction, and this occurs when we convert a matrix into reduced row echelon form. Gaussian elimination is an algorithm to do this. Now with linear algebra and systems of linear equations we may notice that the solutions will be either a single point, where multiple lines or planes intersect, or infinitely many points where all of the lines or planes overlap, or no solution where the lines or planes are parallel. The solution space will always be a linear space

Now let's consider what happens when we have a system of polynomials. Similar to a system of linear equations, a system of polynomial equations is a set of polynomial equations, e.g.

$$x^2 - 2x + 3 = 1$$
$$2x^2 - x + 14 = 12$$

Geometrically, the solution set for a system of polynomial equations is either a finite set of points or a curve. We will work over $\mathbb{C}$, which is closed algebraically, so we will always have a solution. We want an algorithm that gives a basis for the solution space. [1]

First, we need to review the algebraic geometry that we will use.

Definition 2.1. *A* **monomial** *in $x_1, ..., x_n$ is a product of the form.*

$$x_1{}^{a_1} \cdot x_2{}^{a_2} \cdot ... \cdot x_n{}^{a_n}$$

where all of the exponents $a_1 ... a_n$ are nonnegative integers.

[1]

Definition 2.2. *A* **polynomial** *f in $x_1, ..., x_n$ with coefficients in k is a finite linear combination (with coefficients in k) of monomials. We will write a polynomial f in the form*

$$f = \sum_{\alpha} a_\alpha x^\alpha, a_\alpha \in k,$$

where the sum is over a finite number of n-tuples $\alpha = (a_1, ..., a_n)$. The set of all polynomials in $x_1, ..., x_n$ with coefficfients in k is denoted $k[x_1, ..., x_n]$.

[1]

Definition 2.3. *Given a nonzero polynomial $f \in k[x]$, let*

$$f = a_0 x^m + a_1 x^{m-1} + ... + a_m,$$

where $a_i \in k$ and $a_0 \neq 0$ (thus, $m = deg(f)$). Then we say that a_0x^m is the **leading term** *of f, written $LT(f) = a_0x^m$.*

The set of polynomials in $x_1, ..., x_n$ with coefficients in k is a commutative ring. We denote this by $k[x_1, ..., x_n]$
Let us notice that there are two notions of polynomials

(1) f is an element of the ring $k[x_1, ..., x_n]$
(2) f is a function $k^n \to k$

An example of a polynomial is $f = x^2 + 4x - 7$. Now we may notice that our two notions of polynomials agree by the following proposition:

Proposition 1. *Let k be an infinite field, and let $f \in k[x_1, ..., x_n]$. Then $f = 0$ in $k[x_1, ..., x_n]$ if and only if $f : k^n \to k$ is the zero function.*

[1]

Definition 2.4. *A subset I of a ring $k[x_1, ..., x_n]$ is an* **ideal** *if it satisfies these three criteria:*

(1) $0 \in I$
(2) *if $f, g \in I$, then $f + g \in I$*
(3) *if $f \in I$, and $h \in k[x_1, ..., x_n]$, then $hf \in I$*

[1]

Let us notice that all ideals $I \in k[x_1, ..., x_n]$ are finitely generated, i.e. $I = \langle f_1, ..., f_s \rangle$ for some $f_1, ..., f_s \in k[x_1, ..., x_n]$. We want to find a basis $\{f_1, ...f_s\}$ for a given I.

Definition 2.5. *Let $I \subset k[x_1, ..., x_n]$ be an ideal other than $\{0\}$.*

(1) *We denote by $LT(I)$ the set of leading terms of elements of I. Thus*
$$LT(I) = \{cx^a : \text{there exists } f \in I \text{ with } LT(f) = cx^a\}$$
(2) *We denote by $\langle LT(I) \rangle$ the ideal generated by the elements of LT(I)*

[1]
This is a concept that we must understand to truly work with Groebner bases.

Definition 2.6. *Given a field k and a positive integer n, we define the n-dimensional* **affine space** *over k to be the set*
$$k^n = \{(a_1, ..., a_n) : a_1, ..., a_n \in k\}$$

[1]
Now let us talk about varieties, and specifically affine varieties.

Definition 2.7. *An* **affine variety** $\mathbf{V}(f_1, ..., f_s) \subset k^n$*, and let*$f_1, ..., f_s \in k[x_1, ..., x_u]$ *is the set of all solutions of the system of equations* $f_1(x_1, ..., x_n) = ... = f_n(x_1, ..., x_s) = 0$.

[1]
Throughout the paper we will use the lexicographichic ordering. The reason we are specifying the ordering system we want to use is because, while in one variable there is a natural ordering system, problems can arise when using inconsistent ordering systems. This can lead to problems in the division algorithm. This is an issue because we use the division algorithm throughout our paper, so we need consistency within our ordering system for our division algorithm to be accurate.

Definition 2.8. *Let* $\alpha = (a_1, ..., a_n)$*and*$\beta = (b_1, ..., b_n) \in \mathbb{Z}_{\geq 0}{}^n$*. We say* $\alpha >_{lex} \beta$ *if in the vector difference* $\alpha - \beta \in \mathbb{Z}^n$*, the leftmost entry is positive. We will write* $x^\alpha >_{lex} x^\beta$ *if* $\alpha >_{lex} \beta$.

[1]
Simply put, will write from left to right in decending order of the leading variable, and we keep the same leading variable throughout. For example if we have the lex ordering of $x > y > z$ with the terms $x^2y^3, x^4y^2, 2z, 4x^3z^2, x^2y^4z$ we would have the equation

$$x^4y^2 + 4x^3z^2 + x^2y^4z + x^2y^3 + 2z$$

.

Definition 2.9. *Fix a monomial order. A finite subset* $G = \{g_1, ..., g_t\}$ *of an ideal* I*is said to be a* **Groebner Basis** *if*

$$\langle LT(g_1), ..., LT(g_t)\rangle = \langle LT(I)\rangle$$

[1]
A Groebner basis is a preferred generating set of an ideal. It has the property that when the division algorithm is applied to our bases by a ring, then no term of r, our remainder, is divisible by any leading term of our basis, and there are two elements of our ring such that one is always the sum of the other and the remainder.

We may note that every $I \subset \mathbb{C}[x_1, ..., x_n]$ has a Groebner basis and there exists an algorithm to find the basis. Buchberger's algorithm is an algorithm in which we can take a normal set of generators for an ideal into a Groebner basis. Let us note that we can always write $f = q_1f_1 + ... + q_sf_s + r$. If $r = 0$, then $f \in \langle f_1, ..., f_s\rangle$, but we could have $r \neq 0$ and still have $f \in \langle f_1, ..., f_s\rangle$. But if we find a

Groebner basis $\{g_1, ..., g_t\}$, then $f \in \langle g_1, ..., g_t \rangle = I$ if and only if $r = 0$ for $f = p_1 g_1 + ... + p_t g_t + r$. Groebner bases often have polynomials with fewer variables than the originals. In the enxt section we give an example to illustrate why Groebner bases are desirable.

3. Game Theory

We will discuss Game theory specifically, in reference to the algebraic geometry we discussed earlier. A game has n players, which will be denoted $1, ..., n$. A game is a sequence of moves by the players, in which a move is simply a choice between multiple alternatives by one or more of the players. An important theme to remember throughout working with game theory is the understanding that the rules of a game should not be confused with the strategies of the game. Each player will freely choose there own strategy. We will work primarily with personal moves, which are choices made by a specific player depending solely upon their decision and nothing else. [4]

Definition 3.1. Game theory *is the study of strategic decision making through the use of mathematical models of conflict and cooperation between intelligent rational decision-makers.*

There are several types of games such as cooperative, symmetrical, non cooperative, and asymmetical games. We will work specifically with non cooperative games. [4]

Our paper is very strongly influenced by the notion of an equilibrium point. According to John Nash the set of equilibrium points in a two person zero sum game is the set of all pairs of opposing "good" strategies. Nash continues on to show that a finite non-cooperative game will always have at least one equilibrium point, we show how to find this using algebraic geometry techniques. [2]

Definition 3.2. *A* **non-cooperative game** *is a game where players are unable to form binding commitments.*

Within non cooperative games we have pure and mixed strategies.

Definition 3.3. *A* **pure strategy** *is one in which a player uses strategies which are only one unit or zero.*

Definition 3.4. *A* **mixed strategy** *of a player will be a collection of non negative numbers which have unit sum.*

[2]

Pure strategies are all or nothing bets, whereas mixed strategies are when someone spreads there bets throughout their options. Let us notice that by using these concepts of pure and mixed strategies we can

then find equilibrium points of the players involved in the game. This means that we can, when working with non-cooperative games, determine the best strategies to use to achieve the best possible outcomes. One way in which we could determine the strategies with the most positive possible outcomes. This can happen, as we will see later, when we are given the results of when the players each use pure strategies. The way we find our optimal strategies is with the following functions. Let us look at the payouts for three players: A, B, and C, with the respective payouts α, β, γ respectively we find each payout by the following equations:

$$\begin{aligned}
\alpha &= x_1 \sum A_{1jk} y_j z_k + x_2 \sum A_{2jk} y_j z_k + ... + x_n \sum A_{njk} y_j z_k \\
\beta &= y_1 \sum B_{i1k} x_i z_k + y_2 \sum B_{i2k} x_i z_k + ... + y_n \sum A_{ink} x_i z_k \\
\gamma &= z_1 \sum C_{ij1} x_i y_j + z_2 \sum C_{ij2} x_i y_j + ... + z_n \sum A_{ijn} x_i y_j
\end{aligned}$$

Also we may notice that since α, β, γ are the maximum payouts, then they are larger than every individual element of the summations as follows:

$$\begin{aligned}
\alpha &\geq x_1 \sum A_{1jk} y_j z_k, x_2 \sum A_{2jk} y_j z_k, ..., x_n \sum A_{njk} y_j z_k \\
\beta &\geq y_1 \sum B_{i1k} x_i z_k, y_2 \sum B_{i2k} x_i z_k, ..., y_n \sum A_{ink} x_i z_k \\
\gamma &\geq z_1 \sum C_{ij1} x_i y_j, z_2 \sum C_{ij2} x_i y_j, ..., z_n \sum A_{ijn} x_i y_j
\end{aligned}$$

Let us address a problem by John Nash. In this example we have two players, both using two pure strategies. In this example the first player has the pure strategies a and b, and the second player has the pure strategies c and d. Thus when using these respective pure strategies the payouts for our players are as follows:

$$\begin{array}{rcr}
5 & ac & -3 \\
-4 & ad & 4 \\
-5 & bc & 5 \\
3 & bd & -4
\end{array}$$

[2]

Let us call the first player player X and the second player, player Y. P_X is the payout for player X and P_Y is the payout for player Y. Let us observe that

$$\begin{aligned}
S_1 &= xa + (1-x)b \\
S_2 &= yc + (1-c)d
\end{aligned}$$

where S_1 is the mixed strategy for X and S_2 is a mixed strategy for Y. Our goal of using S_1 and S_2 is to determine the maximum payout for each strategy and as such, determine which is the best strategy for our players. Let us take α to be the maximum payout for S_1 and β to be the maximum payout for S_2.

Definition 3.5. *An n-tuple* $\mathfrak{s}$ *is an* **equilubrium point** *if and only if for every i*

$$p_i(\mathfrak{s}) = max_{allr_is}[p_i(\mathfrak{s}; r_i)]$$

[2]

Simply put, an equilibrium point is an n-tuple $\mathfrak{s}$ such that each player's mixed strategy maximizes his payoff if the strategies of the other players are held fixed.

The set of Nash equilibria in our game is given by our payoff matrices X, Y consisting of the common zeros of the following four polynomials subject to x, y.

$$\begin{array}{lcl} x(\alpha - X_{11}(y) - X_{12}(1-x)) & = & 0 \\ (1-x)(\alpha - X_{21}(y) - X_{22}(1-y)) & = & 0 \\ y(\beta - Y_{11}(x) - Y_{12}(1-x)) & = & 0 \\ (1-y)(\beta - Y_{21}(x) - Y_{22}(1-x)) & = & 0 \end{array}$$

[3]

Thus when we substitute the values of our payoff function into our Nash equilibrium we notice that our Nash equilibrium becomes

$$\begin{array}{lcl} x(\alpha - 5y + 4(1-y)) & = & 0 \\ (1-x)(\alpha + 5y - 3(1-y)) & = & 0 \\ y(\beta + 3x - 4(1-x)) & = & 0 \\ (1-y)(\beta - 5x + 4(1-x)) & = & 0 \end{array}$$

These simplify to be

$$\begin{array}{lcl} x\alpha - 9xy + 4x & = & 0 \\ \alpha - x\alpha - 8xy + 3x + 8y - 3 & = & 0 \\ y\beta + 7xy - 4y & = & 0 \\ \beta - y\beta + 9xy - 9x - 4y - 4 & = & 0 \end{array}$$

Where α isw the optmial strategy for the first player and β is the optimal strategy for the second player.

Let us notice that our equations are elements of $\mathbb{R}[\alpha, \beta, x, y]$

Now we can use the program Maple to find the basis for the ideal defined by these polynomials. Now through the use of Maple we can

find the reduced groebner basis, with the lex ordering of $\alpha > \beta > x > y$ to be:

$$\begin{aligned}
g_1 &= 16\alpha + 20 + 17\beta - 41x - 8y \\
g_2 &= 16xy + 4 + \beta - 9x - 8y \\
g_3 &= 16y\beta - 28 - 7\beta + 63x - 8y \\
g_4 &= 280x^2 - 289y^2 - 280x + 289y \\
g_5 &= 280\beta x - 560 - 697y^2 - 140\beta - 140x + 1817y \\
g_6 &= 280\beta^2 - 5600 - 19601y^2 - 280\beta + 21841y \\
g_7 &= 17y^3 - 24y^2 + 7y
\end{aligned}$$

Now we may notice that equation g_7 is in only one variable. This is one of the positive features of using Groebner bases. Thus when we solve for $17y^3 - 24y^2 + 7y = 0$ we obtain that $y \in \{0, 1, \frac{7}{17}\}$. We must use $y = \frac{7}{17}$ and $(1-y) = \frac{10}{17}$ because if we substitute the pure strategies we may notice that $y = 0$ results in either $x = 0$ or $x = 1$ and $y = 1$ results in $x = 0$ or $x = 1$ as well. Thus since we are looking at solutions that are not 0 or 1 we know that $y = \frac{7}{17}$ and $(1 - y) = \frac{10}{17}$. We are not going to use the pure strategies because we are looking for equilibrium points, which will give us our maximum payout. The pure strategies will not give us a max payout, so they will not give us an equilibrium point. Now let us substitute the values for our non pure strategies into our prior equations we may note that $x = \frac{9}{16}$ and $(1 - x) = \frac{7}{16}$.

Thus we may notice that the solution is a vector $(\frac{9}{16}, \frac{7}{16}, \frac{7}{17}, \frac{10}{17})$. Also Nash brings up the idea that this example is from one large solution set of abstract algebra. What this means is that every example similar to this exists within one vector set and the correct answer is contained within this vector. Therefore our Nash equilibrium is the set of strategies:

$$S_1 = \tfrac{9}{16}a + \tfrac{7}{16}b$$

$$S_2 = \tfrac{7}{17}c + \tfrac{10}{17}d$$

Thus our maximum payouts are:

$$\alpha = -0.49$$

$$\beta = 0.64$$

4. Conclusion

Thus we have shown that algebraic geometry can be used to find the maximum payouts for games and game theory. Through the use of algebraic geometry and Groebner bases we can quickly and efficiently

do this. Thus algebraic geometry can be used on a frequent basis within game theory.

References

[1] Cox, D, Little, J., and O'Shea, D., *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, (2008)

[2] Nash, J., *Non-Cooperative Games* Annals of Mathematics, 54 (1951), 286-295.

[3] Sturmfells, B., *Solving Systems of Polynomial Equations*, Regional Conference Series in Mathematics, 97 (2000)

[4] Von Neumann, J., Morgenstern, O., *Theory of Games and Economic Behavior*(1944)

Wayne Cook, Mathematics Department, Georgia College, Milledgeville, GA 31061, U.S.A.

E-mail address: barry.cook@bobcats.gcsu.edu