2010

Teaching the applications of optimisation in game theory's zero sum and non-zero sum games

Fox, William P.

Inderscience

Fox, William P. "Teaching the applications of optimisation in game theory's zero sum and non-zero sum games." International Journal of Data Analysis Techniques and Strategies 2.3 (2010): 258-284.

http://hdl.handle.net/10945/62010

Teaching the applications of optimisation in game theory's zero sum and non-zero sum games

William P. Fox

Department of Defense Analysis,
Naval Postgraduate School,
589 Dyer Road, Room 103F,
Monterey, CA 93943 USA
Fax: 831-656-2649 E-mail: wpfox@nps.edu

Abstract: We apply linear and non-linear programming to find the solutions for Nash equilibriums and Nash arbitration in game theory problems. Linear programming was shown as a viable method for solving mixed strategy zero-sum games. We review this methodology and suggest a class of zero-sum game theory problems that are well suited for linear programming. We applied this theory of linear programming to non-zero sum games. We suggest and apply a separate formulation for a maximising linear programming problem for each player. We move on the Nash arbitration method and remodel this problem as a non-linear optimisation problem. We take the game's payoff matrix and we form a convex polygon. Having found the status quo point (x^*, y^*), we maximise the product $(x - x^*)(y - y^*)$ over the convex polygon using KTC non-linear optimisation techniques. The results give additional insights into game theory analysis.

Keywords: non-linear optimisation; game theory; linear programming; Nash equilibrium; Nash arbitration.

Reference to this paper should be made as follows: Fox, W.P. (2010) 'Teaching the applications of optimisation in game theory's zero sum and non-zero sum games', *Int. J. Data Analysis Techniques and Strategies*, Vol. 2, No. 3, pp.258–284.

Biographical notes: William P. Fox is a Professor at The Naval Postgraduate School in Monterey, California. He obtained his PhD in Industrial Engineering and Operations Research from Clemson University and his MS in Operations Research from the Naval Postgraduate School. His research interests include modelling, optimisation, game theory and simulation. He has many conferences presentations including: *INFORMS, Mathematical Association of America Joint Annual Conference, Military Application Society (MAS)* and *International Conference of Technology in Collegiate Mathematics (ICTCM)*. He has co-authored several books and over 100 articles. He has previously taught at West Point and Francis Marion University.

1 Introduction

Game theory is a branch of applied mathematics that is used in the social sciences (most notably economics), biology, decision sciences, engineering, political science, international relations, operations research, computer science and philosophy. Game theory attempts to mathematically capture behaviour in strategic situations, in which an individual's success in making choices depends on the choices of others. While initially developed to analyse competitions in which one individual does better at another's expense (zero sum games), it has been expanded to treat a wide class of interactions that are classified according to several criteria. Today, "game theory is a sort of umbrella or 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players" (Aumann, 1987).

Let us consider zero-sum games initially. For two-player finite zero-sum games, the different game theoretic solution concepts of Nash equilibrium, minimax and maximin all give the same solution. In zero sum games, many basic solution techniques exist and are taught to be used in a hierarchical approach. We look for dominance first and the movement diagram is a useful technique. We will point out now that linear programming works well with mixed strategy solutions and pure strategy solutions.

Traditional applications of game theory provide techniques to find an equilibrium value in zero-sum games. John F. Nash Jr. proved that every game has Nash (1951) equilibrium. At equilibrium, each player has adopted a strategy that they are unlikely to change and is called the Nash equilibrium. This methodology is not without criticism, and debates continue over the appropriateness of particular equilibrium concepts, the appropriateness of equilibrium altogether, and the usefulness of mathematical models. However, it is still widely used.

Many authors such as Straffin (2004) and Winston (1995) spend a great deal of effort to present traditional techniques of the zero-sum games and the Nash equilibrium. The minimax and maximin methods are presented as procedures to solve these game theory problems. Winston (1995) provides an example or two on applying linear programming to the zero-sum game for solving only mixed strategy games. Straffin (2004, p.19) commented about linear programming but provides no examples. In his article on linear programming, Danzig (2002) discussed the historical foundations of linear programming as well as a meeting with Von Neumann where the latter felt that linear programming was an analogue to the theory of games.

Ville (2009) presents a good discussion on the application of linear programming to the zero-sum game in terms of the primal and dual problem as well as their formulations. In game theory, the minimax theorem for a two player zero sum game relies on the fact that the two players strategies are always opposite, known as duals. He states that the strategy for one player is self-dual. Furthermore, Ville states that for more than two players or in non-zero sum games the indeterminacy should be removed by ethical rules, convention that exclude certain coalitions of types of coalitions. However, he does not provide examples of applications in these larger or non-zero sum games.

In the *Classics of Scientific Literature*, Klarrich (2009) provided a review of early game theory in which she states as Von Neumann was working on non-zero sum games... "He found his approach gave rise to complicated mathematics and intractable mathematics". This is where Nash came up with his proof concerning equilibrium for a specific set of strategies, one for each player, such that no player would benefit from unilaterally changing his strategy while the other players stick to their equilibrium strategies.

Although some developments occurred before it, the field of game theory came into being with the 1944 book *Theory of Games and Economic Behaviour* by John von Neumann and Oskar Morgenstern. This theory was developed extensively in the 1950s by many scholars. Game theory was later explicitly applied to biology in the 1970s, although similar developments go back at least as far as the 1930s. Game theory has been widely recognised as an important tool in many fields. Eight game theorists have won Nobel prizes in economics, and John Maynard Smith was awarded the Crafoord Prize for his application of game theory to biology.

In a monograph edited by Koopman (1951), several important theoretical discussions are presented concerning game theory and linear programming. Gale et al. (1951) presented and established theorem of duality and existence for 'general' linear programming problems and related these general problems to the theory of zero-sum two-person games. In his chapter, Danzig (1951) presented the maximisation of a linear function of variables subject to linear inequalities where he presented the replacement of a linear with a linear equality in non-negative variables. Dorfman (1951) then applied the simplex method of Danzig's to a game theory problem. He showed that in solving zero-sum games with two opponents that optimal strategies could be found in accordance with the principles of game theory. In his published example, he had two players with six strategies and five strategies, respectively. These papers were the foundation of linear programming being applied to game theory. This conference and its proceedings were instrumental in linking linear programming to game theory.

Crawford (1974) discussed an optimal strategy for zero-sum games. He stated by referring to the work done by Gale et al. (1951) that economists have known for a long time that finding optimal strategies in a zero-sum game is equivalent to finding a solution to an appropriately defined linear programme.

In summary, the current methods to find the Nash equilibrium include dominance, minimax, maximin, equalising strategies, William's method (1986) and these are found in many modern game theory textbooks such as Straffin (2004). Again, Straffin (2004, p.19) commented about linear programming but does not use linear programming as one of his techniques.

Through contacting Harold Kuhn about this work, we found the work of Daskalakis et al. (2008) who were concerned with the time until economic agents converge to Nash equilibrium. They applied the Brouwer function in order to find the approximate Nash equilibrium in the complex case PPAD and show that it will converge. They showed their method was in fact PPAD-complete. The importance here is the continued use of the Nash equilibrium to solve game theory problems.

2 Linear programming and the zero-sum game

Consider a constant-sum two person game where X, the maximising player has m strategies and Y, the minimising player has n strategies. The entry a_{ij} from the ith row and jth column of the payoff matrix represents the value of the game. Without loss of generality, it can be assumed that every element of the matrix is greater than or equal to zero. If this is not true, then a constant can be added to every element in the payoff matrix to make all the entries positive. Suppose that player X plays a weighted mixed strategy defined by assigning a weight x_i to the ith strategy where $\Sigma\, x_i = 1$. Then, according to Dorfman (1951) and Danzig (1951) the value of the game will be:

$$v = \sum_{j=1}^{n} y_i (a_1 x_1 + a_2 x_2 + \ldots + a_m x_m)$$

where y_j is the jth strategy for player Y. They used this concept to construct the linear programming equivalent.

The Nash equilibrium for a two-player, zero-sum game can be found by solving a linear programming problem. Suppose a zero-sum game has a payoff matrix M where element $M_{i,j}$ is the payoff obtained when the minimising player chooses pure strategy i and the maximising player chooses pure strategy j (i.e., the player trying to minimise the payoff chooses the row and the player trying to maximise the payoff chooses the column). In their work, Danzig (1951) and Dorman (1951) assume every element of M is positive. A general payoff matrix would be:

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & . & . & . & M_{1,n} \\ M_{2,1} & M_{2,2} & . & . & . & M_{2,n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ M_{m,1} & M_{m,2} & . & . & . & M_{m,n} \end{bmatrix}$$

The game will have at least one Nash equilibrium. The Nash equilibrium can be found by solving the following linear programme to find a vector u as shown by Dorfman (1951) to solve for the solution to the column player's game:

$$\begin{aligned} &\text{Minimise } u_1 + u_2 + u_3 + \ldots + u_n \\ &\text{Subject to:} \\ &u_1 + u_2 + u_3 + \ldots + u_n \geq 0 \\ &M_{1,1}u_1 + M_{1,2}u_2 + \ldots + M_{1,n}u_n \geq 1 \\ &M_{2,1}u_1 + M_{2,2}u_2 + \ldots + M_{2,n}u_n \geq 1 \qquad (1) \\ &. \\ &. \\ &. \\ &M_{m,1}u_1 + M_{m,2}u_2 + \ldots + M_{m,n}u_n \geq 1 \end{aligned}$$

The first constraint says each element of the u vector must be non-negative, and the second constraint says each element of the Mu vector must be at least one. For the resulting u vector, the inverse of the sum of its elements is the value of the game. Multiplying u by that value gives a probability vector, giving the probability that the maximising player will choose each of the possible pure strategies.

An important concept in linear programming is non-negativity. If the game matrix does not have all positive elements, simply add a constant to every element that is large enough to make them all positive. That will increase the value of the game by that constant, and will have no effect on the equilibrium mixed strategies for the equilibrium.

The equilibrium mixed strategy for the minimising player can be found by solving the dual of the given linear programme. Or, it can be found by using the above procedure to solve a modified payoff matrix which is the transpose and negation of M (adding a constant so it is positive), then solving the resulting game.

If all the solutions to the linear programme are found, they will constitute all the Nash equilibrium for the game. Conversely, any linear programme can be converted into a two-player, zero-sum game by using a change of variables that puts it in the form of the above equations. So such games are equivalent to linear programmes, in general.

We can also consider the following formulation for the maximising player that provides results for the value of the game and the probabilities x_i as illustrated by Winston (1995, p.636). This formulation is a simpler modification to equation (1) and precludes any mathematics being used after the solution is found.

Again, we use the same format for the elements of the payoff matrix M but we no longer restrict the elements to be positive.

$$\begin{aligned}
&\text{Maximise } V\\
&\text{Subject to:}\\
&M_{1,1}x_1 + M_{1,2}x_2 + \ldots + M_{1,n}x_n - V \geq 0\\
&M_{2,1}x_1 + M_{2,2}x_2 + \ldots + M_{2,n}x_n - V \geq 0\\
&.\\
&.\\
&.\\
&M_{m,1}x_1 + M_{m,2}x_2 + \ldots + M_{m,n}x_n - V \geq 0\\
&x_1 + x_2 + \ldots + x_n = 1\\
&V, x_i \geq 0
\end{aligned} \tag{2}$$

where V is the value of the game, $M_{m,n}$ are payoff matrix entries, and x*'s* are the weights (probabilities to play the strategies).

For the zero-sum games only, we can take advantage of the fact the Rose is maximising and Colin ins minimising by using the primal (maximising linear programme) and the dual (minimising linear programme). If we solve for Rose's maximising solution in our primal linear programme, we find Colin's solution in the dual.

If we let our primal linear programme for Rose be equation (2), then the dual linear programme for Colin would be:

$$\begin{aligned}
&\text{Minimise } V \text{ (the dual model)}\\
&\text{Subject to:}\\
&M_{1,1}y_1 + M_{2,1}y_2 + \ldots + M_{n,1}y_n - V \leq 0\\
&M_{1,2}y_1 + M_{2,2}y_2 + \ldots + M_{n,2}y_n - V \leq 0\\
&.\\
&.\\
&.\\
&M_{1,n}y_1 + M_{2,n}y_2 + \ldots + M_{n,n}y_n - V \leq 0\\
&y_1 + y_2 + \ldots + y_n = 1\\
&V, y_i \geq 0
\end{aligned}$$

This formulation yields a solution to the probabilities $y_1, y_2 \ldots ym$ and V. We may read the dual solution directly from the primal output. Each dual variable $y_i = -$ coefficient of the slack variable (shown as dual prices in the software, LINDO).

In this paper, we will provide examples of using linear programming for the zero-sum game using equation (2) and we conclude that often the linear programming model is best when games are lager than 2 × 2. We illustrate this with a 3 × 3 example.

We then extend the optimisation illustrations to finding Nash equilibrium and Nash arbitration solutions to the non-zero sum games. The Nash equilibrium is found using linear programming formulations but now all players are maximising. A separate linear programming formulation is required for each player, which often is easier for solving for the Nash equilibrium than the alternative methods suggested in the literature. The reduced costs will be conjectured. The Nash arbitration is an example of constrained non-linear optimisation and will be illustrated as such.

We illustrate several examples of the applications of linear programming to zero-sum games in order to establish a working procedure. In the zero-sum games, because we only have Rose's game presented, the solution to Colin is found in the solution to the linear programming known as the dual problem. We use equation (2) to illustrate the procedure. We start by showing both formulations in the following simple example.

Example 1: baseball's the hitter-pitcher duel

Roger Clemons is facing Big Poppy. Roger is known for his fierce fast ball and his devastating split-finger. Big Poppy is a tremendous competitor and a great fast ball hitter. We have the following historical statistics on these players.

		Roger Clemons	
	His guess/ His pitch	Fast ball	Split-finger
Big Poppy	Fast ball	0.475	0.100
	Split-finger	0.125	0.250

Big Poppy wants to maximise our battling average by doing a better job at guessing the pitch. Roger Clemons wants Big Poppy to have his minimal batting average as he throws the unexpected pitch. We define the decision variables as follows:

BA = the batting average of Big Poppy

x_1 = the percentage of time to be looking for the fast ball and we let $(1 - x_1)$ be the percentage of time we are guessing the split-finger

We set up the linear programming problem for Big Poppy using equation (2).

Maximise BA

Subject to:

$0.475\,x_1 + 0.125\,x_2 - BA \geq 0$

$0.1\,x_1 + 0.250\,x_2 - BA \geq 0$

$x_1, x_2, V > 0$ (non-negativity)

The solution via LINDO is as follows:

```
LP OPTIMUM FOUND AT STEP        2

          OBJECTIVE FUNCTION VALUE
          1)    0.2125000

   VARIABLE          VALUE              REDUCED COST
         BA           0.212500              0.000000
         X1           0.250000              0.00000
         X2           0.750000              0.00000

        ROW    SLACK OR SURPLUS         DUAL PRICES
         2)           0.000000             -0.300000
         3)           0.000000             -0.700000
         4)           0.000000              0.212500

NO. ITERATIONS =          2
```

As an illustration we show the use of equation (1) and its procedure in the following steps.

Minimise $u_1 + u_2$

Subject to:

$u_1 + u_2 \geq 0$

$0.475\ u_1 + 0.1\ u_2 \geq 1$

$0.125\ u_1 + 0.25\ u_2 \geq 1$

The solution is found using LINDO as:

```
LP OPTIMUM FOUND AT STEP        2

          OBJECTIVE FUNCTION VALUE

1)              4.705883

 VARIABLE          VALUE           REDUCED COST
       U1           1.411765          0.000000
       U2           3.294118          0.00000

  ROW        SLACK OR SURPLUS      DUAL PRICES
  2)              4.705883            0.000000
  3)              0.000000           -1.176471
  4)              0.000000           -3.529412

NO. ITERATIONS=         2
```

Recall that we mentioned that using formulation (1) requires additional calculations. We set $S = u_1 + u_2 = 4.705883$. We find $x_1 = u_1/S = 0.30000$ and $x_2 = u_2/S = 0.70000$ for the pitch selection from Clemons and the dual solution $y_1 = 1.176471/4.705883 = 0.2500$ and $y_2 = 3.529412/4.705883 = 0.750000$ to indicate Big Poppy should look for the fast ball 25% of the time and for the split-finger 75% of the time. The value of the game is the inverse of S, $S^{-1} = 1.4705883 = 0.2125$, which is the batting average. Obviously, the procedures involved in using equation (1) is not straight forward as we introduced a new variable and is more work.

Example 2: Dorfman's original example (1951)

		Player A Strategy				
		1	2	3	4	5
	1	5.31	8.52	12.05	16	20.00
Player B	2	2.70	3.77	6.30	9.7	13.40
Strategy	3	3.64	2.70	3.60	5.91	8.99
	4	5.91	3.60	2.70	3.64	6.02
	5	9.70	6.30	3.77	2.70	4.04
	6	16	12.05	8.52	5.31	2.70

Dorfman used formulation (1) and found the solution of x_i from u_i and S using the $x_i = u_i/S$. He found $S = 0.166$ and $u_1 = 0.057$ and $u_5 = 0.109$ in the optimal solution. This lead to $x_1 = 0.343$ and $x_5 = 0.657$. The value of the game is $v = 5.982$ using:

$$v = \sum_{j=1}^{n} y_i(a_1x_1 + a_2x_2 + \ldots + a_mx_m)$$

Using equation (2), we set up to obtain the following linear programme and solve.

Maximise V

Subject to:

$$5.31x_1 + 8.52x_2 + 12.05x_3 + 16x_4 + 20x_5 - V \geq 0$$
$$2.7x_1 + 3.77x_2 + 6.3x_3 + 9.7x_4 + 13.4x_5 - V \geq 0$$
$$3.64x_1 + 2.7x_2 + 3.6x_3 + 5.91x_4 + 8.99x_5 - V \geq 0$$
$$5.91x_1 + 3.6x_2 + 2.7x_3 + 3.64x_4 + 6.02x_5 - V \geq 0$$
$$9.7x_1 + 6.3x_2 + 3.77x_3 + 2.7x_4 + 4.04x_5 - V \geq 0$$
$$16x_1 + 12.05x_2 + 8.52x_3 + 5.31x_4 + 2.7x_5 - V \geq 0$$
$$x_1 + x_2 + x_3 + x_4 + x_5 = 1$$
$$V, x_i \geq 0$$

The solution is directly found directly as $V = 5.982253$ when $x_1 = 0.343154$, $x_2 = x_3 = x_4 = 0$ and $x_5 = 0.656846$. The dual solution (for the minimising player) is $V = 5.982253$, $y_4 = 0.980936$ and $y_5 = 0.019064$.

Example 3: the 2 × 3 games

Consider the following example, which in the literature is used to show a short-cut graphical method introduced by Williams (1986) to reduce the 2 × 3 game to a 2 × 2 game. Rather than use his short cut method, we can use linear programming directly to obtain the solution.

			Colin	
		C	D	E
Rose				
	A	1	−1	3
	B	−2	1	−1

We note that there are negative values in the payoff matrix as we illustrate this procedure is valid the any values in the payoff matrix, *M*. Our best option is to use the method suggested by Winston (1995, p.172–178) to replace any variable that could take on negative values with the difference in two positive variables, $x_j - x'_j$. We assume that the value of the game could be positive or negative. The other values we are looking for are probabilities that are always between zero and one.

Maximise $V = v_1 - v_2$

Subject to:

$x_1 - x_2 - v_1 + v_2 \geq 0$

$-x_1 + x_2 - v_1 + v_2 \geq 0$

$3x_1 - x_2 - v_1 + v_2 \geq 0$

$x_1 + x_2 = 1$

$v_1, v_2, x_i \geq 0$

Note: we needed to replace V with $v_1 - v_2$ since the solution to V can be positive or negative.

Solving this linear programme using LINDO, we find:

$V = v_1 - v_2 = -0.2$

$x_1 = 0.60, x_2 = 0.40, v_1 = 0.00, v_2 = 0.20$ and the dual variables are

$y_1 = 0.40, y_2 = 0.60, y_3 = 0.00.$

Thus, for Rose she plays 60% A and 40% B to get a value of the game of −0.2 while Colin plays 40% C, 60% D and 0% E to obtain the value of −0.2.

The interpretation of the solution to the game is found as Colin plays strategy C, 40% of the time, strategy E, 60% of the time, and never plays strategy F. Colin's game is worth 0.2, Rose plays strategy A 40% of the time, and plays strategy B 60% of the time. The value of the game to Rose is −0.2.

In the previous two games it might have been easier to use the mixed strategy methods to find the solutions. However, these examples are nice as they provide a vehicle to set up and solve game as linear programming problems when we can obtain solutions via other methods. These examples show the direct application of linear programming to zero-sum games.

Example 4: a 3 × 3 game where equalising strategies does not work

In this 3 × 3 game we first checked the movement diagram and dominance, and then we attempted to employ equalling strategies. The method fails. Textbooks in game theory, such as Straffin's (2004) text, suggest trying every subgame to find the solution. The literature is hazy on which subgame solution we want to use and also states these methods can be very 'tedious'. Straffin (2004, p.19) confesses that linear programming is the most efficient method to solve larger games and his text contains no examples of the use of linear programming. Linear programming is the better choice to find the solutions to large game theory problems where there is no dominate solution or solutions found through the movement diagrams. In Winston's (1995) text, he provides a three step method to consider.

- *Step 1*: Check for a saddle point. If the game does not have a saddle point, go to Step 2.
- *Step 2*: Eliminate any of the row player's dominated strategies. Looking at the reduced payoff matrix, eliminate any column player's dominated strategies. Continue until all dominated strategies are removed. Go to Step 3.
- *Step 3*: If the game has been reduced is greater than to 2 × 2 then solve by linear programming.

We illustrate below.

Payoff matrix

			Colin		
		D	E	F	
	A	9	2	7	x
Rose	B	3	6	4	y
	C	5	3	1	z
		s	t	u	

For Rose, the decision variables are:

v = expected value of the game

x = probability for playing strategy A

y = probability for playing strategy B

z = probability for playing strategy C

We formulate the problem as:

Maximise V
Subject to:
$9x + 3y + 5z - v \geq 0$
$2x + 6y + 3z - v \geq 0$
$7x + 4y + 1z - v \geq 0$
$x + y + z = 1$
non-negativity $x, y, z, v \geq 0$

Our solution is $v = 4.8$ when $x = 0.03$, $y = 0.70$, $z = 0.0$ and the dual variables are 0.40, 0.60, 0.00 respectively.

Colin will play his strategies with probabilities 0.40, 0.60, 0.0 with the game yielding the same results ($V_c = -4.8$). The use of linear programme is quick, concise, and direct.

Example 5: a 3×3 game with a saddle point solution

Consider the game with payoff matrix, *M* and saddle point solution at pure strategy CE with value of the game as five to Rose.

Payoff matrix

		Colin			
		D	E	F	Max{RowMin}
	A	4	4	10	
Rose	B	2	3	1	
	C	6	5	7	*5*
Min{ColMax}			*5*		Saddle point

Let us treat this as a linear programming problem. Our formulation would be:

Maximise V

Subject to:

$4x_1 + 2x_2 + 6x_3 - V > 0$

$4x_1 + 3x_2 + 5x_3 - V > 0$

$10x_1 + x_2 + 7x2 - V > 0$

$x_1 + x_2 + x_3 = 1$

$x_i, V \geq 0$

Solving the linear programme yields $x_1 = x_2 = 0$, $x_3 = 1$ and $V = 5$ for Rose with a dual solution as $y_1 = y_3 = 0$, $y_2 = 1$ and $V = -5$ for Colin.

Example 6: consider the game with multiple saddle point solutions at strategies AF, AH, BE and CF with value of two

			Colin		
		E	F	G	H
Rose	A	4	2	5	2
	B	2	1	−1	−20
	C	3	2	4	2
	D	−16	0	16	1

We formulate the problem as:

Maximise V

Subject to:

$$4x_1 + 2x_2 + 3x_3 - 16x_4 - V > 0$$
$$2x_1 + x_2 + 2x_3 - V > 0$$
$$5x_1 - x_2 + 4x_3 + 16x_4 - V > 0$$
$$2x_1 - 20x_2 + 2x_3 + x_4 - V > 0$$
$$x_1 + x_2 + x_3 + x_4 = 1$$
$$x_i, V \geq 0$$

Using our LP software, we find a solution of $V = 2$ when $x_1 = 1$ and corresponding dual $y_4 = 1$. This is our solution for pure strategy AH at $V = 2$. The other solutions are found by alternate optimal techniques in linear programming. We suggest understanding a solid discussion of this technique such as in Winston (1995, p.142–144). We will find all the alternate optimal solutions to this game theory problem by applying the techniques to find such solutions in the optimal tableau.

So far in our examples and discussions, we have reviewed the concepts of applying linear programming to the zero-sum games. We showed that both pure strategy solutions as well as mixed strategy solutions can be found using linear programming. We make a strong recommendation that using linear programming more easily provides a solution to zero-sum game theory problems that are larger than 2 × 2. We have shown that linear programming can be used for finding solutions to pure strategy games as well although we believe that the application of movement diagrams and dominance to find pure strategy solutions might be easier. Now, we present some new ground and introduce the concept in applying the use of linear programming in game theory to non-zero sum games.

3 The non-zero sum game and optimisation

The literature for non-cooperative game solutions that are non-zero sum games is to find the Nash equilibrium. Recall, in a simple zero-sum game such as flipping and matching coins where if we match the Rose player loses $1 and the Colin player wins $1 and if we do not match the Rose player wins a dollar and the Colin player loses a dollar the total of the players entries always equally zero.

Zero-sum game

		Colin	
		Heads	Tails
Rose	Heads	(−1, 1)	(1, −1)
	Tails	(1, −1)	(−1, 1)

Non-zero sum games are games where if one player wins the other player does not have to lose. Both players could win something or lose something. The following payoff matrix is a simple example of a non-zero game between two players.

		Colin	
		C	D
Rose	A	(1, 1)	(−1, −1)
	B	(−1, −1)	(1, 1)

In a non-zero sum game solution methods include looking for dominance, movement diagrams and equalising strategies. Here, we present an extension to the application of linear programming from the zero-sum game to the non-zero sum game. The primal-dual relationship from zero-sum games does not hold for non-zero sum games since both players strategy is to maximise their game. However, the theory of Danzig (1951), Kuhn and Tucker (1951) that allowed for linear programming to be used in setting up a zero-sum game to maximise V subject to the constraints still holds for each individual player as we will show. Because of the nature of non-zero sum games, where both players are trying to maximise their outcomes, we can model each player's strategies as their own *maximising* linear programme. The dual results (reduced costs) will only provide us insights into the value of utility for the players but do not give insights into the other player's decisions or solutions. Therefore, the theory accompanying the application to the maximising player in a zero-sum game holds for each player maximising in a non-zero sum game. The analysis and articles from 1951 by Danzig, Dorfman and Gale et al. directly apply except for their discussions of duality. We treat each player as separate linear programming problems.

Again, let us define the following payoff matrix that has components for both Rose and Colin:

$$(M,N)=\begin{bmatrix}(M_{1,1},N_{1,1}) & (M_{1,2},N_{1,2}) & \cdots & (M_{1,n},N_{1,n})\\ (M_{2,1},N_{2,1}) & (M_{2,2},N_{2,2}) & \cdots & (M_{2,n},N_{2,n})\\ \cdot & \cdot & \cdots & \cdot\\ \cdot & \cdot & \cdots & \cdot\\ \cdot & \cdot & \cdots & \cdot\\ (M_{m,1},N_{m,1}) & (M_{m,2},N_{m,2}) & \cdots & (M_{m,n},N_{m,n})\end{bmatrix}$$

In non-cooperative non-zero sum games, we use similar concepts of pure strategies and equalising strategies (mixed strategy) to solve the game. We look for pure strategy solution using the *movement diagram*.

For *Rose*, she would maximise payoffs, so she would prefer the highest payoff at each *column*. Arrows in columns but values are Rose's. Similarly for Colin, he wants to maximise his payoffs, so he would prefer the high payoff at each row. We draw an arrow to the highest payoff in that row. Arrows are in rows but values are Colin's. If all arrows point in from every direction, then that or those points will be pure Nash equilibrium.

If all the arrows do not point at a value or values then we must use equalising strategies to find the weights (probabilities) for each player. Basically, we would proceed as follows:

- Rose's game: Rose maximising, Colin 'equalising' is a zero sum game which yields Colin's equalising strategy.
- Colin's game: Colin maximising, Rose 'equalising' is a zero-sum game which yields Rose's equalising strategy.
- Note: If either side plays their 'equalising strategy', the other side 'unilaterally' cannot improve their own situation (stymie the other player).

This translates into two additional linear programming formulations, one for each maximising player. We create two more formulations in the form of equation (2) that we call equations (3) and (4) that when used provide the values of the game and the probabilities that the players should play their strategies in the equalising strategy concept above.

$$\begin{aligned}
&\text{Maximise } V\\
&\text{Subject to:}\\
&N_{1,1}x_1 + N_{2,1}x_2 + \ldots + N_{m,1}x_n - V \geq 0\\
&N_{2,1}x_1 + N_{2,2}x_2 + \ldots + N_{m,2}x_n - V \geq 0\\
&\ldots\\
&N_{m,1}x_1 + N_{m,2}x_2 + \ldots + N_{m,n}x_n - V \geq 0\\
&x_1 + x_2 + \ldots + x_n = 1\\
&\text{Non-negativity}
\end{aligned} \tag{3}$$

where the weights x_i yield Rose strategy and the value of V is the value of the game to Colin.

$$\begin{aligned}
&\text{Maximise } V\\
&\text{Subject to:}\\
&M_{1,1}y_1 + M_{2,1}y_2 + \ldots + M_{m,1}y_n - v \geq 0\\
&M_{2,1}y_1 + M_{2,2}y_2 + \ldots + M_{m,2}y_n - v \geq 0\\
&\ldots\\
&M_{m,1}y_1 + M_{m,2}y_2 + \ldots + M_{m,n}y_n - v \geq 0\\
&y_1 + y_2 + \ldots + y_n = 1\\
&\text{Non-negativity}
\end{aligned} \tag{4}$$

where the weights y_i yield Colin's strategy and the value of v is the value of the game to Rose.

We also point out that looking for dominance and the movement diagrams are still critical as initial steps. If you find total dominance and pure strategy solutions by the movement diagram, you have found a pure strategy Nash equilibrium. According to Gillman and Housman (2009, p.189) non-zero sum games with pure strategy equilibrium also have non-pure strategy equilibriums suing Nash's equalising strategy. Linear programming provides a solution methodology for these non-pure strategy equilibriums.

Example 7: consider the following partial conflict mixed strategy game

		Colin	
		C	D
Rose	A	(2, 4)	(1, 0)
	B	(3, 1)	(0, 4)

The linear programming formulations for each of our player's in order to find the Nash equilibrium values for Rose and Colin are found as follows:

(a) Maximise V_c

Subject to:

$4x_1 + x_2 - V_c \geq 0$

$0x_1 + 4x_2 - V_c \geq 0$

$x_1 + x_2 = 1$

$x_i, V_c \geq 0$

(b) Maximise V_r

Subject to:

$2y_1 + y_2 - V_r \geq 0$

$3y_1 - Vr \geq 0$

$y_1 + y_2 = 1$

$y_1, V_r \geq 0$

The solutions are:

a $V_c = 2.285714286$ when $x_1 = 0.5714285714$ or 4/7 and $x_2 = 0.4285714286$ or 3/7

b $V_r = 1.5000$ when $y_1 = 0.50000$ and $y_2 = 0.50000$.

This game results in the Colin playing 1/2 C, 1/2 D and insuring a value of the game of 1.5000 for Rose while Rose plays 4/7 A, 3/7 B and yielding a value of the game of 2.285714286 for Colin. The solution is (1.5000, 2.285714286).

Example 8: consider the non-zero sum game with more than two strategies per player where there is no pure strategy equilibrium

		Colin		
		D	E	F
Rose	A	(9, 1)	(2, 2)	(7, 2)
	B	(3, 2)	(6, 1)	(4, 2)
	C	(5, 2)	(3, 2)	(5, 0)

The movement arrows reveal no pure strategy so we turn to linear programming to find our solutions using the equalising strategies.

Maximise V_c

Subject to:

$$x_1 + 2x_2 + 2x_3 - Vc \geq 0$$
$$2x_1 + x_2 + 2x_3 - Vc \geq 0$$
$$2x_1 + 2x_2 - Vc \geq 0$$
$$x_1 + x_2 + x_3 = 1$$
$$Vc, x_i \geq 0$$

Maximise V_r

Subject to:

$$9y_1 + 2y_2 + 7y_3 - Vr \geq 0$$
$$3y_1 + 6y_2 + 4y_3 - Vr \geq 0$$
$$5y_1 + 3y_2 + 5y_3 - Vr \geq 0$$
$$y_1 + y_2 + y_3 = 1$$
$$Vr, y_i \geq 0$$

Solving these two linear programming problems yield the following results:

$V_r = 4.5$, $Vc = 1.6$ when Rose plays 0.4A,0.4B,0.2C and Colin plays 0.0 D, 0.25 E, 0.75 F. Thus the Nash equilibrium is (4.5, 1.6) and the probabilities to play strategies are $x_1 = 0.4, x_2 = 0.4, x_3 = 0.2, y_1 = 0, y_2 = 0.25$ and $y_3 = 0.75$.

We have extended the application of linear programming in zero-sum games to finding the Nash equilibrium by finding the equalising strategies in non-zero sum games. The theory for linear programming use holds as each player is maximising so we treat each problem as a 'primal' maximising problem. The dual solution (the reduced costs) only provides us information about the utilities as resources. For example, assume we have a reduced cost of 0.5. Then an increase in the utility value of one unit of that constraint increases the objective function value by approximately 0.5 utility units.

Let us extend our optimisation applications to the Nash arbitration scheme as a non-linear programming problem using Kuhn-Tucker conditions (KTCs) (Kuhn, 1951; Bazarra et al., 1993).

4 Nash arbitration and non-linear programming formulation

We utilise the KTC to find the optimal solution to a non-linear optimisation problem as listed in equation (5):

$$\begin{aligned}
&\text{Max (or min)} f(x_1, x_2, \ldots, x_n) \\
&\text{Subject to:} \\
&g_1(x_1, x_2, \ldots, x_n) \le b_1 \\
&g_2(x_1, x_2, \ldots, x_n) \le b_2 \\
&. \\
&. \\
&. \\
&g_m(x_1, x_2, \ldots, x_n) \le b_m
\end{aligned} \tag{5}$$

Since we want to the Nash arbitration point, we desire the maximisation of the function. We want to find the values of $(x_1, x_2, \ldots, x_n)$ and multiplier $(\lambda_1, \lambda_2, \ldots, \lambda_m)$ that satisfy the following KTC conditions in equation (6):

$$\begin{aligned}
&\frac{\partial f(x)}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(x)}{\partial x_j} = 0 \\
&\lambda_i [b_i - g_i(x)] = 0 \\
&\lambda_i \ge 0
\end{aligned} \tag{6}$$

5 Nash arbitration as a non-linear programming problem

In the bargaining problem, Nash (1950) developed a scheme for producing a single fair outcome. The goals for the Nash arbitrations scheme are that the result will be at or above the stats quo point for each player and that the result must be 'fair'.

Nash introduced the following terminology:

- *Status quo point*: We will typically use the intersection of Rose's security level and Colin's security level; the Threat positions may also be used.
- *Negotiation set*: those points in the Pareto optimal set that are at or above the 'status quo' of both players.

We use Nash's four axioms that he believed that a reasonable arbitration scheme should satisfy rationality, linear invariance, symmetry and invariance. A good discussion of these axioms and can be found in Straffin (2004, p.104–105). Simply put, the Nash arbitration point is the point that follows all four axioms. This leads to Nash's theorem stated below:

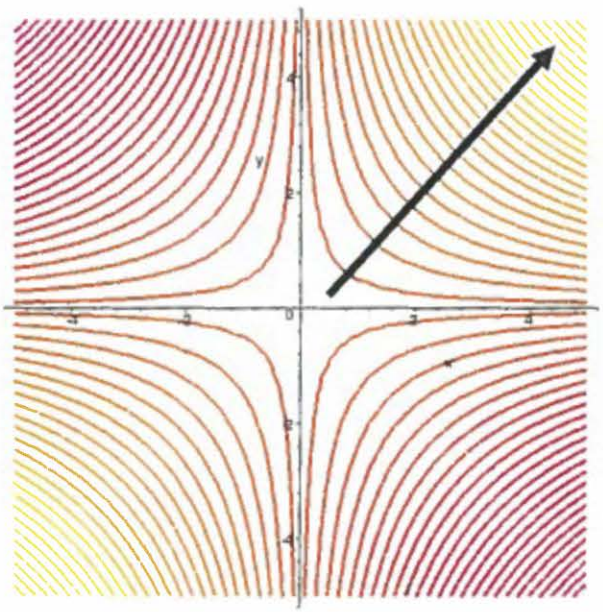
- *Nash's theorem (1950)*: There is one and only one arbitration scheme which satisfies axioms 1 through 4. It is this: if the *status quo* $SQ = x_0, y_0$, then the arbitrated solution point N is the point (x, y) in the polygon with $x \ge x_0$ and $y \ge y_0$ which *maximises the product*: $(x - x_0)(y - y_0)$.

Let us examine this geometrically first as it will provide insights into using non-linear optimisation methods. We produce the contour plot of our non-linear function: $(x - x_0)(y - y_0)$ when our status quo point is assumed to be (0, 0). It is obvious that the

NE corner of quadrant 1 is where this function is maximised. This is illustrated in Figure 1.

In his theory for the arbitration and cooperative solutions, Nash (1950) stated the 'reasonable' solution should be Pareto optimal and will be at or above the security level. The set of outcomes that satisfy these two conditions is called the negotiation set. The line segments that join the negotiation set must form a convex region as shown in Nash's proof. Methodologies for solving for this point use basic calculus, algebra and geometry.

Figure 1 Contour plot for (x^*y) (see online version for colours)



Note: We note that the direction of maximum increase is NE as indicated by the arrow.

We present another methodology. Since we have used linear programming to solve for the Nash equilibrium values earlier, we show that we can use non-linear programming to solve for the arbitration point. It is non-linear because of the choice of the Nash function, $(x - x^*)$ $(y - y^*)$ that we want to maximise. The constraints are linear and must form a convex set.

For any game theory problem, we next overlay the convex polygon onto our contour plot (Figure 1). The most NE point in the feasible region is our optimal point and the Nash arbitration point. This will be where the feasible region is tangent to the hyperbola. Without generalisation, maximising a non-linear concave function over a linear convex region produces a maximum directly from the KTCs (Bazarra et al., 1993).

In our examples, we will use the security value as the status quo point to use in the Nash arbitration procedure. We additionally define the procedure to find the security value as follows:

In a non-zero-sum game, Rose's optimal strategy in Rose's game is called Rose's *prudential strategy* and the value is called Rose's *security level*. Colin's optimal strategy in Colin's game is called Colin's *security level*. We will illustrate this during the solution to find the Nash arbitration point in example 9.

Example 9: Nash arbitration example from a non-zero sum game

		Colin	
		C	D
Rose	A	*(2, 6)*	*(10, 5)*
	B	*(4, 8)*	*(0, 0)*

To find the security level (status quo point) we look at the following two separate games extracted from the original game and use movement diagrams, dominance, or our linear programming method to solve each game for those players' values.

In a prudential strategy, we allow a player to find their optimal strategy in their own game. For Rose, she would need to find her optimal solution in her own game. Rose's game below has a mixed strategy solution; $V = 10/3$.

		Colin	
		C	D
Rose	A	*2*	*10*
	B	*4*	*0*

For Colin, he would need to find his optimal solution in his own game. Colin's game below has a pure strategy solution, $V = 6$.

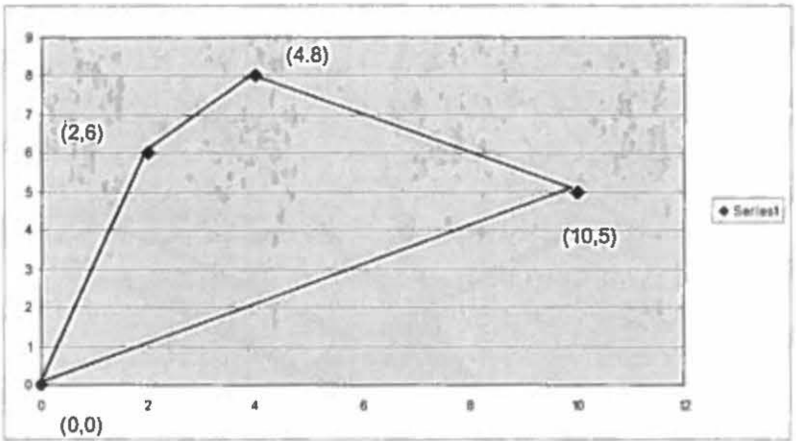
		Colin	
		C	D
Rose	A	*6*	*5*
	B	*8*	*0*

The status quo point or security level from the prudential strategy is found to be (10/3, 6). We will use this point in the formulation of the non-linear programme.

We set up the convex polygon (constraints) for the function that we want to maximise, which is $(x - \frac{10}{3})\ (y - 6)$. The convex polygon is the convex set from the values in the pay-off matrix. Its boundary and interior points represent all possible combinations of strategies. Corner points represent *pure* strategies. All other points are mixed strategies. Occasionally, a pure strategy is an interior point. Thus, we start by plotting the strategies from our payoff matrix set of values {(2, 6), (4, 8), (10, 5), (0, 0)}, see Figure 2.

We note that our convex region has four sides whose coordinates are our pure strategies. We use the point-slope formula to find the equations of the line and then test points to transform the equations to inequalities. For example, the line form (4, 8) to (10, 5) is $y = -0.5x + 10$. We rewrite as $y + 0.5x = 10$. Our test point (0, 0) show that are inequality is $0.5x + y \leq 10$. We use this technique to find all boundary lines as well as add our security levels as lines that we need to be above.

Figure 2 Payoff polygon, example 9 (see online version for colours)



The convex polygon is bounded by the following in equalities:

$$.5x + y \le 10$$
$$-3x + y \le 0$$
$$0.5x - y \le 0$$
$$-x + y \le 4$$
$$x \ge x^*$$
$$y \ge y^*$$

where x^* and y^* are the security levels (10/3,6).

The NLP formulation to find the Nash arbitration value following the format of equation (5) is as follows:

$$\begin{aligned} &\text{Maximise } Z = (x - \tfrac{10}{3}) \cdot (y - 6) \\ &\text{Subject to:} \\ &0.5x + y \le 10 \\ &-3x + y \le 0 \\ &0.5x - y \le 0 \\ &-x + y \le 4 \\ &x \ge \frac{10}{3} \\ &y \ge 6 \end{aligned} \quad (7)$$

We use MAPLE, as our software, to both provide the graphs and the solution outputs using programmes previously written (Fox, 2000). We display the feasible region graphically in Figure 2. The feasible region is the solid region. From the figure we can approximate the solution as the point of tangency between the feasible region and the hyperbolic contours in the NE region.

We use the conditions of equation (6) to solve our NLP as shown in equation (7) to find the point indicated by the arrow in Figure 3.

Figure 3 Convex polygon and function contour plot (see online version for colours)

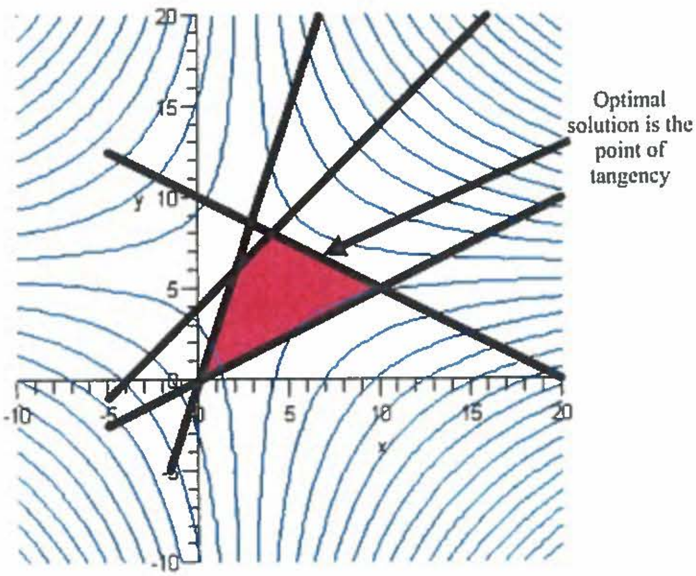
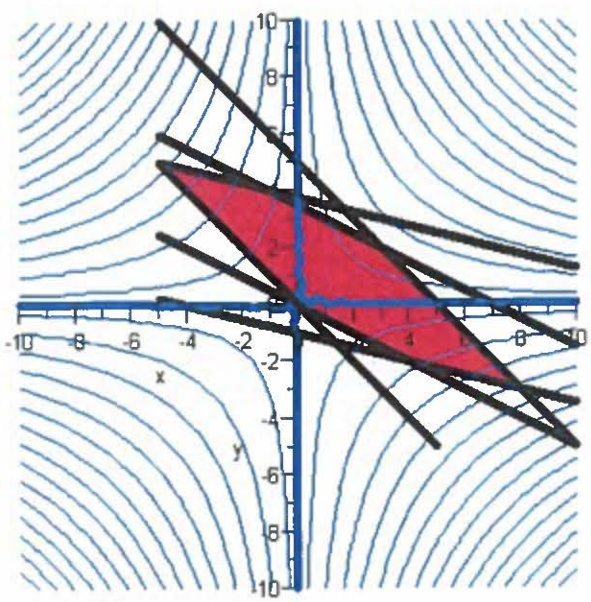


Figure 4 The graphical NLP problem for the management-labour arbitration (see online version for colours)



Our optimal solution, the Nash arbitration point is found to be x = 5.667 and y = 7.167 and the value of the objective function is 2.72 with shadow prices: $\lambda_2 = 2.3333$ and all other $\lambda_i = 0$.

The value is 2.72. The interpretation of the Lagrange multiplier, λ_i, would be an increase in the utility associated with strategy AD of one unit would yield an increase in the value of the game solution of 2.3333 units.

We choose a more complicated example next, the labour management game from Straffin (2004). This game has two players and the Rose player has four strategies and the Colin player has four strategies.

Example 10: management-labour arbitration (from Straffin, p.115–117)

				Labour concedes	
		Nothing	C	A	CA
	Nothing	(0, 0)	(4, −1)	(4, −2)	(8, −3)
Management concedes	P	(−2, 2)	(2, 1)	(2, 0)	(6, −1)
	R	(−3, 3)	(1, 2)	(1, 1)	(5, 0)
	PR	(−5, 5)	(−1, 4)	(-1, 3)	(3, 2)

The convex polygon is graphed from the constraints below (see the plots in Figure 3 and Figure 4):

$$x + y \geq 0$$
$$0.5x + y \geq 0$$
$$0.25x + y \geq -1$$
$$x + y \geq 5$$
$$0.5x + y \leq 3.5$$
$$0.25x + y \leq \frac{15}{4}$$

The status quo point (our security level) is (0, 0), making the function to maximise simply $x * y$.

Our formulation is:

Maximise $x * y$

Subject to:

$$x + y \geq 0$$
$$0.5x + y \geq 0$$
$$0.25x + y \geq -1$$
$$x + y \geq 5$$
$$0.5x + y \leq 3.5$$
$$0.25x + y \leq \frac{15}{4}$$

We use Maple to assist in finding the results.

> *sol* := *NLPSolve* (*objective*, *constr*, *maximise* = *true*);

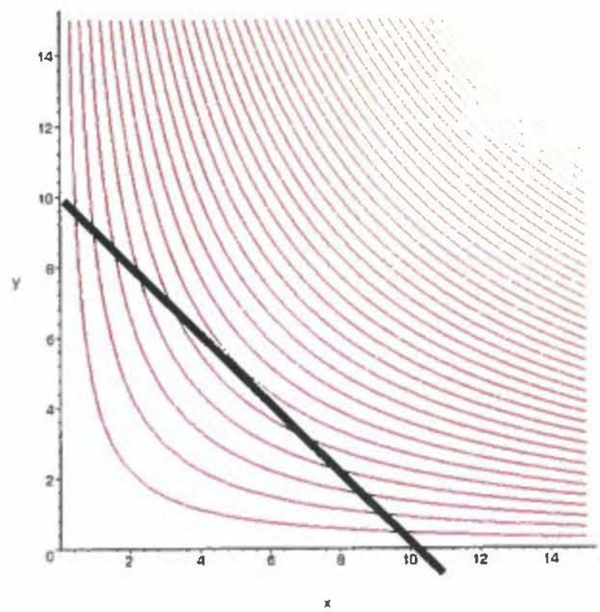
sol := [5.99999999999999912, [$x = 3.00000000000000088$,
$y = 1.99999999999999912$]]

The product is taken as $xy = 6.0$ and the values are taken as $x = 3$ and $y = 2$.

With our function, we are concerned with a point of the boundary of the constraint. The constraint region is the Pareto optimal region (the northeast boundary).

In particular, we are looking for:

Figure 5 The Nash function's contours with status quo (0, 0) and the NE boundary line (see online version for colours)



The optimal point is the point on the line that is tangent to the contours in the direction of the NE increase.

Example 11: the writer's guild strike (Fox, 2008)

A payoff matrix consisting of cardinal utilities is presented in Figure 6.

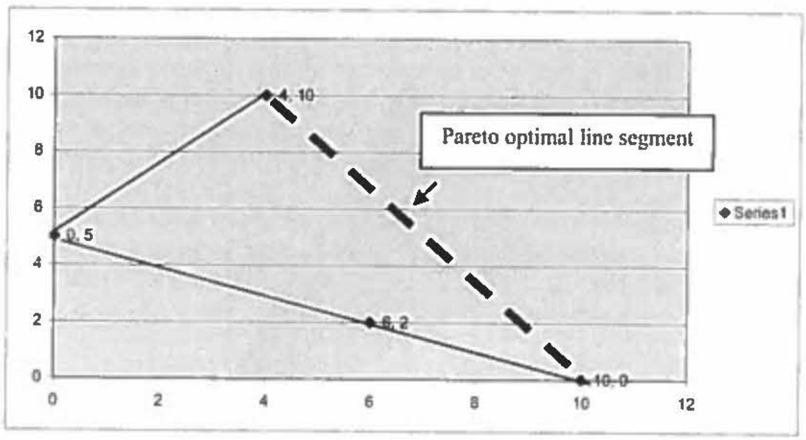
Payoff matrix

		Management (Colin)	
		SQ	*IN*
Writer's (Rose)	*S*	(0, 5)	(6, 2)
	NS	(4, 10)	(10, 0)

Using the movement diagram, we can easily find that (4, 10) is the pure Nash equilibrium. We also note that this result is not satisfying to the Writer's guild and that they would like to have a better outcome. Both (6, 2) and (10, 0) provide a better outcome to the writers. We plot these coordinates from the payoff matrix to determine if any points are Pareto optimal, see Figure 6.

The Nash equilibrium value (4, 10) lies along the Pareto optimal line segment. But the writers can do better by going on strike and forcing arbitration, which is what they did.

Figure 6 Payoff polygon for writer's guild strike (see online version for colours)



The status quo point is the security levels of each side. We find these values using prudential strategies as (4, 5). The function for the Nash arbitration scheme is *Maximise* $(x-4)(y-5)$.

Our formulation is:

Maximise $(x-4)(y-5)$

Subject to:

$$\frac{5}{3}x + y \le \frac{50}{3}$$

$$\frac{-5}{4}x + y \le 5$$

$$\frac{1}{2}x + y \le 5$$

We can find the convex polygon from the payoff values and plot in Figure 6. We graph the convex polygon in Figure 7 with the contours of our function, $(x-4)(y-5)$.

Using Maple, we find the desired solution to our NLP as:

$x = 5.5$

$y = 7.5$

$l_1 = 1.5$

$l_2 = 0$

$l_3 = 0$

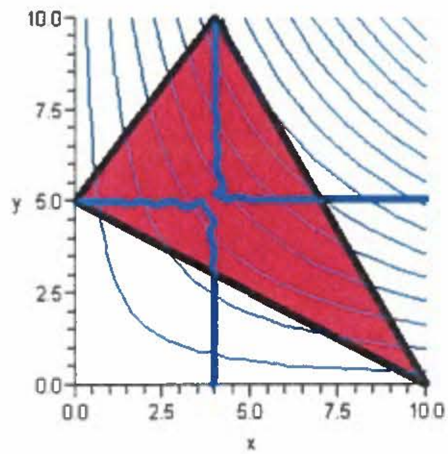
$u_1 = 0$

$u_2 = 2.09165$

$u_3 = 2.29128$

We have the x and y coordinate (5.5, 7.5) as our arbitrated solution. We also have obtain some information about λ_1. It's value of 1.5 means that an increase in utility of one unit for this constraint provides an increase in value of approximately 1.5 units.

Figure 7 Convex polygon and Nash arbitration contours (see online version for colours)



6 Conclusions

We have presented numerous ideas from optimisation into game theory. We provided an example to support Straffin's (2004, p.19) comment that linear programming is most efficient for solving large zero-sum games. We also show how linear programming can be used to find pure strategy solutions as well as alluded to finding multiple solutions using alternate optimal solution analysis. Both Straffin and Winston recommend using a computer software package to use linear programming if you cannot solve the game in a reasonable time. We found that LINDO and Excel are both good software packages to solve the linear programming problems. We recommend using linear programming for zero-sum games that are 3 × 3 and larger after checking and reducing the game in you have dominance. Furthermore, you do not need to be an expert in linear programming to formulate the game theory into a linear programme or to use the commercial software.

We broke new ground and showed how and when linear programming can be used to obtain the Nash equilibrium (equalising strategies) in solving non-zero sum games. We apply that same theory to separate linear programmes for each player maximising their strategy in a non-zero game where equalising strategies are used. Again commercial software is available to solve these formulated linear programmes. For large, non-zero sum games linear programmes are both efficient and a time saver.

Having applied linear programming to both zero sum games and non-zero sum games, we illustrate how we can employ non-linear optimisation to find the Nash arbitration point. Again, for large games the non-linear programming methodology is fast and efficient. We also graphically illustrated the notion of non-linear optimisation through obtaining a graphical display of the result.

In summary, we recommend using linear programming with equation (2) for any zero sum game that is larger than 2 × 2. We recommend using equations (3) and (4) for any non-zero sum game larger than 2 × 2 which requires equalising strategies to solve for the Nash equilibrium. We also recommend for larger games involving the Nash arbitration using non-linear optimisation methods using equations (5) and (6).

The impetus for looking into the optimisation techniques in game theory followed a visit of John Nash to our university and to my game theory class (Nash, 2009). After working our preliminary findings to apply optimisation to various aspects of game theory these finding were sent to both John Nash and H. Kuhn, who both wished success in the endeavour but their new interests kept them from more involvement.

In our own classes in mathematical modelling for decision making, we have antidotal evidence of the success of using linear programming techniques. Now that we have introduced the use of linear programming into the course, the student projects and current thesis work involving applied modelling and analysis include much larger games.

References

Aumann, R.J. (1987) 'Game theory', *The New Palgrave: A Dictionary of Economics*, Vol. 2, pp.460–82.

Bazarra, M.S., Sherali, H.D. and Shetty, C.M. (1993) *Nonlinear Programming*, Chapter 4, pp.149–173.Wiley, New York.

Crawford, V. (1974) 'Learning the optimal strategy in a zero-sum game', *Econometrica*, Vol. 42, No. 5, pp.885–891.

Danzig, G. (1951) 'Maximization of a linear function of variables subject to linear inequalities', in Koopman, T. (Ed.): *Activity Analysis of Production and Allocation Conference Proceeding*, John Wiley Publishers, Chap. 21, pp.339–347.

Danzig, G. (2002) 'Linear programming', *Operations Research*, Vol. 50, No. 1, pp.42–47.

Daskalakis, C., Goldberg, P.W. and Papadimirtriou, C.H. (2008) 'The complexity of computing a Nash equilibrium', to appear in *SICOMP*.

Dorfman, R. (1951) 'Application of the simplex method to a game theory problem', in Koopman, T. (Ed.): *Activity Analysis of Production and Allocation Conference Proceeding*, Chap. 22, pp.348–358, John Wiley Publishers.

Fox, W.P. (2000) 'Nonlinear optimization', Course notes and student study guide at Francis Marion University Print Plant, Florence, SC.

Fox, W.P. (2008) 'Mathematical modeling of conflict and decision making "the Writers Guild strike 2007–2008"', *Computers in Education Journal*, Vol. 18, No. 3, pp.2–11.

Gale, D., Kuhn, H. and Tucker, A. (1951) 'Linear programming and the theory of games', in Koopman,T. (Ed.): *Activity Analysis of Production and Allocation Conference Proceeding*, Chapter 19, pp.317–329, John Wiley Publishers.

Gillman, R. and Housman, D. (2009) 'Models of conflict and cooperation', *Providence*, pp.189–195, American Mathematical Society.

Klarrich, E. (2009) 'The mathematics of strategy', *Classics of the Scientific Literature*, October, available at http://www.pnas.org/site/misc/classics5.shtml.

Kuhn, H.W. and Tucker, A.W. (1951) 'Nonlinear programming', in Newman, J. (Ed.): *Proceedings 2nd Berkley Symposium on Mathematical Statistics and Probability*, University of California Press, CA.

Nash, J. (1950) 'The bargaining problem', *Econometrica*, Vol. 18, pp.155–162.

Nash, J. (1951) 'Non-cooperative games', *Annals of Mathematics*, Vol. 54, pp.289–295.

Nash, J. (2009) 'Lecture at NPS', Feb 19.

Straffin, P.D. (2004) 'Game theory and strategy', Mathematical Association of America, Washington.

Ville, J.A. (2009) 'Game theory, duality, economic growth', *Electronic Journal for Probability and Statistics*, Vol. 5, No. 1, pp.5–17.

Williams, J.D. (1986) *The Compleat Strategyst*, Dover Press, New York, original edition by RAND Corporation, 1954.

Winston, W.L. (1995) *Introduction to Mathematical Programming*, 2nd ed., Duxbury Press, Belmont.