The Construction Path of Artificial Intelligence Technology on Human Practice and Ethical Values

Ronald Deckert

Abstract

We apply linear programming to find the solutions for Nash equilibriums in two person zero-sum games. Linear programming has been shown to be a viable method for solving both pure and mixed strategy zero-sum games. We review this methodology and suggest classes of zero-sum games that are well suited for solving using linear programming. Rather than suggest the dual solution methodology for the opponent's strategies, we supply a single formulation for both players. We illustrate with MS-Excel and a Solver Macro template designed as a technology assistant.

Keywords

game theory, linear programming, zero-sum games, primal and dual solutions, linear program formulations, Excel, Excel Solver, Nash equilibrium

Introduction

In our interdisciplinary Department of Defense Analysis at the Naval Postgraduate School, we teach a three course sequence in mathematical modeling for decision making. In the first course, we teach basic linear programming both using the 2-variable graphical simplex technique and the Excel Solver using **SimplexLP**. In the third course, we teach *models of conflict* that concentrates on game theory.

In this 3rd course, we teach the basic concepts and solution techniques for game theory. In our class we use the Straffin text [21] as well as Chapter 10 from Giordano, Fox, and Horton [14]. We will not cover any of the basic solution techniques here in this paper.

Our students must complete a course project of their own choice using one of the modeling techniques from class. Students use the modeling process in their project: they identify the problem; they list the appropriate assumptions with justifications; they explain why their modeling technique is selected; they solve the model; interpret the solution, perform sensitivity analysis (if applicable); and they discuss strengths and weaknesses of their modeling approach. Here is short list of some of the game theory projects:

- Game Theory with US and Non-State actors
- Game Theory in Cameroon-Nigeria dispute
- Game Theory in PMI and US military tasks
- COIN Game
- The Somali Pirates game
- US-Afghanistan drug dilemma
- US-Afghanistan Regional Game
- US Coin Operations Game
- Dealing with Safe Havens as a Game
- IEDS and Counter-IEDS as a Game
- Game theory for Courses of Combat Action

In the past, our coverage did not cover much linear programming so our solution processes were limited to two-person, two strategy games using algebraic methods because of the complexity of the solution mechanics. Recently, we have added more applications of linear programming as a solution technique so students might add more reality to the number of possible strategies available to the players.

Zero-Sum Game Background

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 behavior in strategic situations where an individual's success in making choices depends on the choices of others. Game theory describes how players should play games if both the players are rational. By rational, we imply that each player desires the best outcomes that they can achieve playing the game. Here we analyze competitions in which one individual does better at another's expense i.e., zero sum games [1].

We begin with a discussion of zero-sum games, also known as total conflict games [14]. For 2-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. In most courses, we look for dominance first and then the movement diagram to look for pure strategy solutions. The saddle point method (minimax and maximin) is also a very useful technique for finding pure strategy solutions. If these methods fail to produce a pure strategy solution then we find a mixed strategy solution. We point out that linear programming always works with games having either mixed strategy solutions or 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 equilibrium [18]. At equilibrium, each player has adopted a strategy that they are unlikely to change. This equilibrium point (or points) is called the Nash equilibrium.

Many authors such as Straffin [21] and Winston [23] 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 short-cut procedures to solve a specific class of game theory problems. Winston [23] provides an example or two on applying linear programming to the zero-sum game for solving only mixed strategy games. Straffin [21] comments about linear programming being useful in large games but provides no examples. In his article on linear programming, Danzig [5, 6] discussed the historical foundations of linear programming as well as a meeting with Von Neumann where the latter felt that linear programming was an analog to the theory of games. Fox [9, 10] showed the use of linear programming in both zero-sum and non-zero sum games.

In a monograph edited by Koopman [16], several important theoretical discussions are presented concerning game theory and linear programming. Gale, Kuhn, and Tucker [12] 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 [5,6] presented the maximization of a linear function of variables subject to linear inequalities where he presented the replacement of a linear inequality with a linear equality in nonnegative variables.

Dorfman [8] then applied the simplex method of Danzig’s to a game theory problem. In his conference paper, he showed that, in solving zero-sum games with two opponents, optimal strategies could be found in accordance with the principles of game theory. In his example, he had 2 players with 6 strategies and 5 strategies, respectively. The conference papers were the foundation of linear programming being applied to game theory and were instrumental in linking linear programming to game theory.

In summary, the current methods to find the Nash equilibrium include Dominance, Minimax, Maximin, equalizing strategies, and William’s Method [23]. These techniques are found in many modern game theory textbooks such as Straffin [21] and Barron [3]. Again Straffin [21] only comments about linear programming but does not use linear programming as one of his techniques.

Zero-sum Games

Often game theory is taught in courses other than a mathematics' major elective. Other curricula might include economics, business management, business analytics, political sciences, or in interdisciplinary studies.

In the mathematics curriculum, we assume that the students have had a full course in linear programming that has covered the Primal-Dual methods. In the other disciplines, the students might have had only a cursory introduction to basic linear programming. The emphasis of the game theory course is on the game theory. However, linear programming, its game theory formulation, and solution via technology is accessible to all disciplines even those, like ours, whose mathematics requirement is only college algebra. We will illustrate how to do this.

Using the same conventions as Straffin, we call the row player, Rose, and the column player, Colin. Let's define the zero-sum game with the following payoff matrix that has components for both Rose and Colin where Rose has *m* strategies and Colin has *n* strategies:

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

In the special case of zero-sum games each pair sums to zero. For example, one such pair is $M_{11}+N_{11}=0$.

Our knowledge of the zero-sum game and the Primal-Dual relations suggest formulating Rose's game and finding Colin's solution through the dual solution. This works well if you have a single problem to solve. But what if you have many games to consider and you only have Excel. How best to construct a technology assistant?

Treating the zero-sum game as above translates into two linear programming formulations, one for each maximizing player. We combine the formulation into a single formulation shown in equation (3) where the solution provides the values of the game and the probabilities that the players should play their strategies. Further, if any pairs (M_{mn}, $N_{m,n}$) are negative then there is a chance that the game solution can be negative. Since the game solution will be a decision variable in our formulation, we must account for that possibility. Our best recommendation is to use the method suggested by Winston [23, 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 0 and 1. Since this occurs only in the value of the game, we use as a substitute variable, $V=v_1-v_2$.

Maximize v_1-$v2$

Subject to:

$$\begin{aligned} &M_{1,1}x_1 + M_{2,1}x_2 + ... + M_{m,1}x_n - v_1 + v_2 \geq 0 \\ &M_{1,2}x_1 + M_{2,2}x_2 + ... + M_{m,2,}x_n - v_1 + v_2 \geq 0 \\ &... \\ &M_{1,m}x_1 + M_{2,m}x_2 + ... + M_{m,n}x_n - v_1 + v_2 \geq 0 \\ &x_1 + x_2 + ... + x_n = 1 \\ &\textit{Nonnegativity} \end{aligned}$$

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

Maximize v_3-v_4

Subject to:

$$N_{1,1}y_1 + N_{1,2}y_2 + ... + N_{1,m}y_n - v_3 + v_4 \geq 0$$
$$N_{2,1}y_1 + N_{2,2}y_2 + ... + N_{2,m}y_n - v_3 + v_4 \geq 0$$
...
$$N_{m,1}y_1 + N_{m,2}y_2 + ... + N_{m,n}y_n - v_3 + v_4 \geq 0$$
$$y_1 + y_2 + ... + y_n = 1$$
Nonnegativity

where the weights y_i yield Colin's strategy and the value of v_3-v_4 is the value of the game to Colin.

To accomplish this as one formulation, we combine as

Maximize v_1-v_2 + v_3-v_4 (1)

Subject to:

$$M_{1,1}x_1 + M_{2,1}x_2 + ... + M_{m,1}x_n - v_1 + v_2 \geq 0$$
$$M_{1,2}x_1 + M_{2,2}x_2 + ... + M_{m,2}x_n - v_1 + v_2 \geq 0$$
...
$$M_{1,m}x_1 + M_{2,m}x_2 + ... + M_{m,n}x_n - v_1 + v_2 \geq 0$$
$$x_1 + x_2 + ... + x_n = 1$$
Nonnegativity

$$N_{1,1}y_1 + N_{1,2}y_2 + ... + N_{1,m}y_n - v_3 + v_4 \geq 0$$
$$N_{2,1}y_1 + N_{2,2}y_2 + ... + N_{2,m}y_n - v_3 + v_4 \geq 0$$
...
$$N_{m,1}y_1 + N_{m,2}y_2 + ... + N_{m,n}y_n - v_3 + v_4 \geq 0$$
$$y_1 + y_2 + ... + y_n = 1$$
Nonnegativity

The algorithm for the LP template solver, saved as macro-enabled, is described as follows:

Step 1. Input the payoff matrix for Rose only
Step 2. Make all decision variables { x_1, x_2, …,x_{10},y_1,y_2,…,y_{10}, v_1,v_2,v_3,v_4} initially set to 0.
Step 3. Highlight the objective function cell, open the Solver, insure it has non-negativity for variables checked, and is set for SimplexLP. Then hit solve.
Step 4. The answers are updated automatically.

Template Algorithm

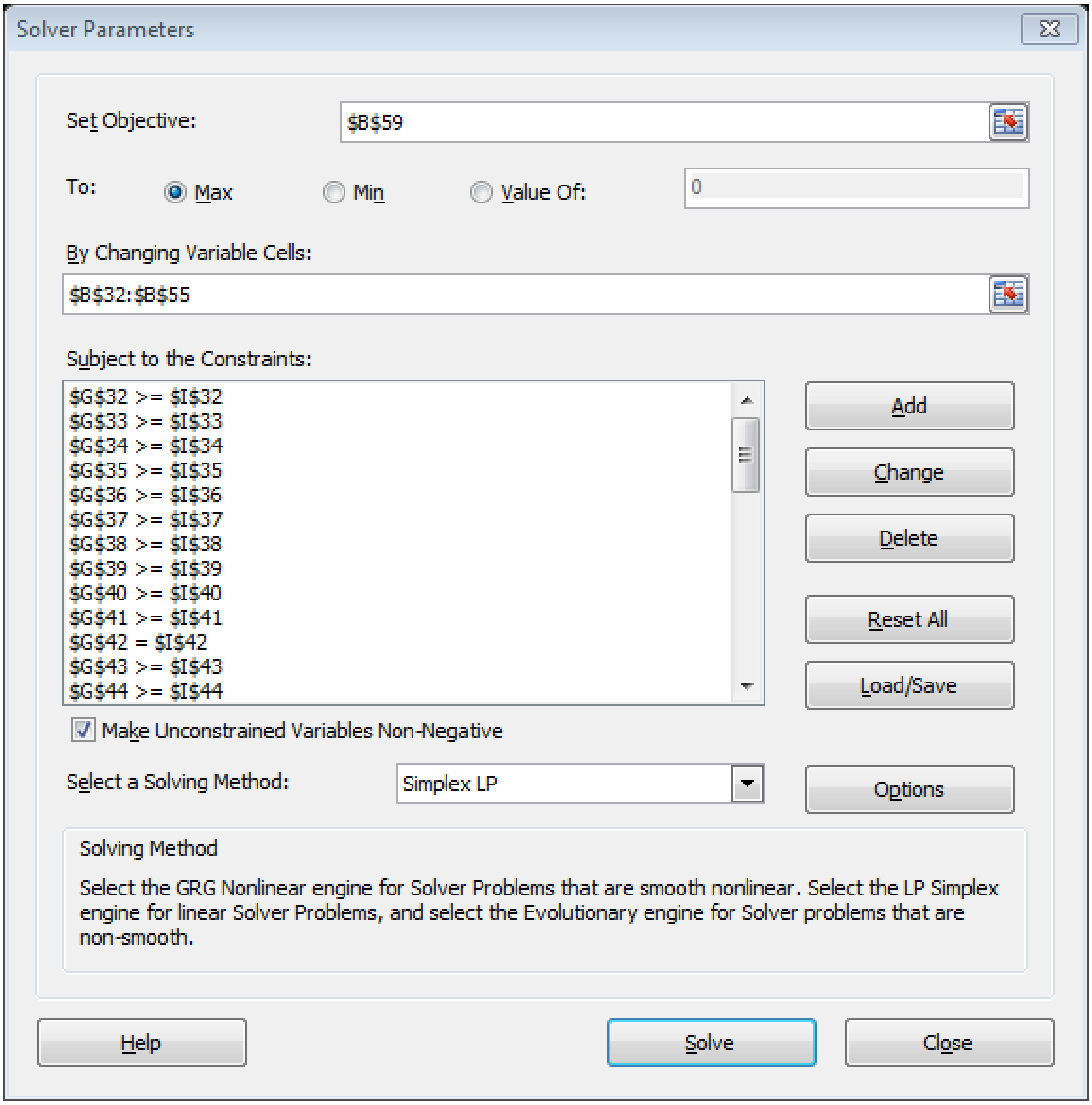
The algorithm for building the template for up to 10 strategies for Rose and Colin required several steps using *Logical If* statements that allow for including or not including constraints based upon the number of variables in the problem. The user input for the numbers of active rows and active columns is a critical user input. These changes in the formulation are essential because of the additional generic decision variables. In creating the constraints, the constraints that exceed the number of rows or column need to be zeroed out. *Logical If* statements are used to handle that condition within the constraints. Recall the template is set up to accept up to ten rows and ten columns. If our game theory problem has less than ten rows and/or less than ten columns then the used number of decision variables are eliminated if these exceed the number of row or columns as well. Again, a *Logical If* statement has been added to handle that eventuality.

The template uses linear programming in the Excel Solver (called *SimplexLP)* to solve for the game's solution and to identify the strategies to be played by each player to achieve the "best" solution either by pure or mixed strategies.

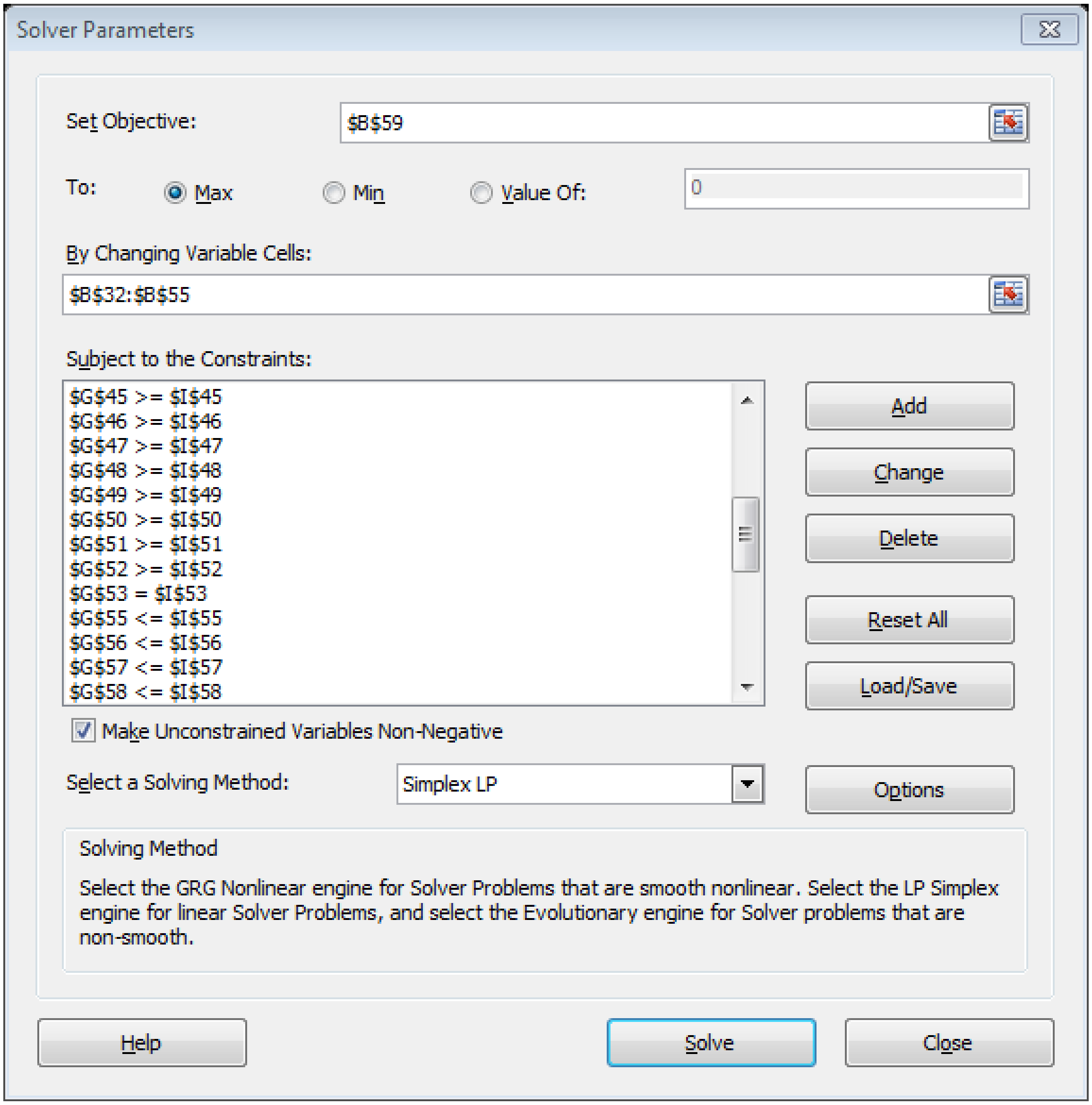
How do we deal with alternate optimal solution identification? If a pure strategy solution has alternate solutions this is identified by examining the Sensitivity Report. A basic decision variable, a variable whose value is 1, having either an increase or decrease of 0 in the report identifies a possible alternative optimal solution. To find the other solution, add a constraint that requires the optimal solution to equal the current value and change the objective function to Maximize another decision variable whose final value is 0 and also has an increase or decrease of 0. This requires some additional linear programming work beyond what is contained in this template..

Generic Solver formulation from the template with 43 constraints:

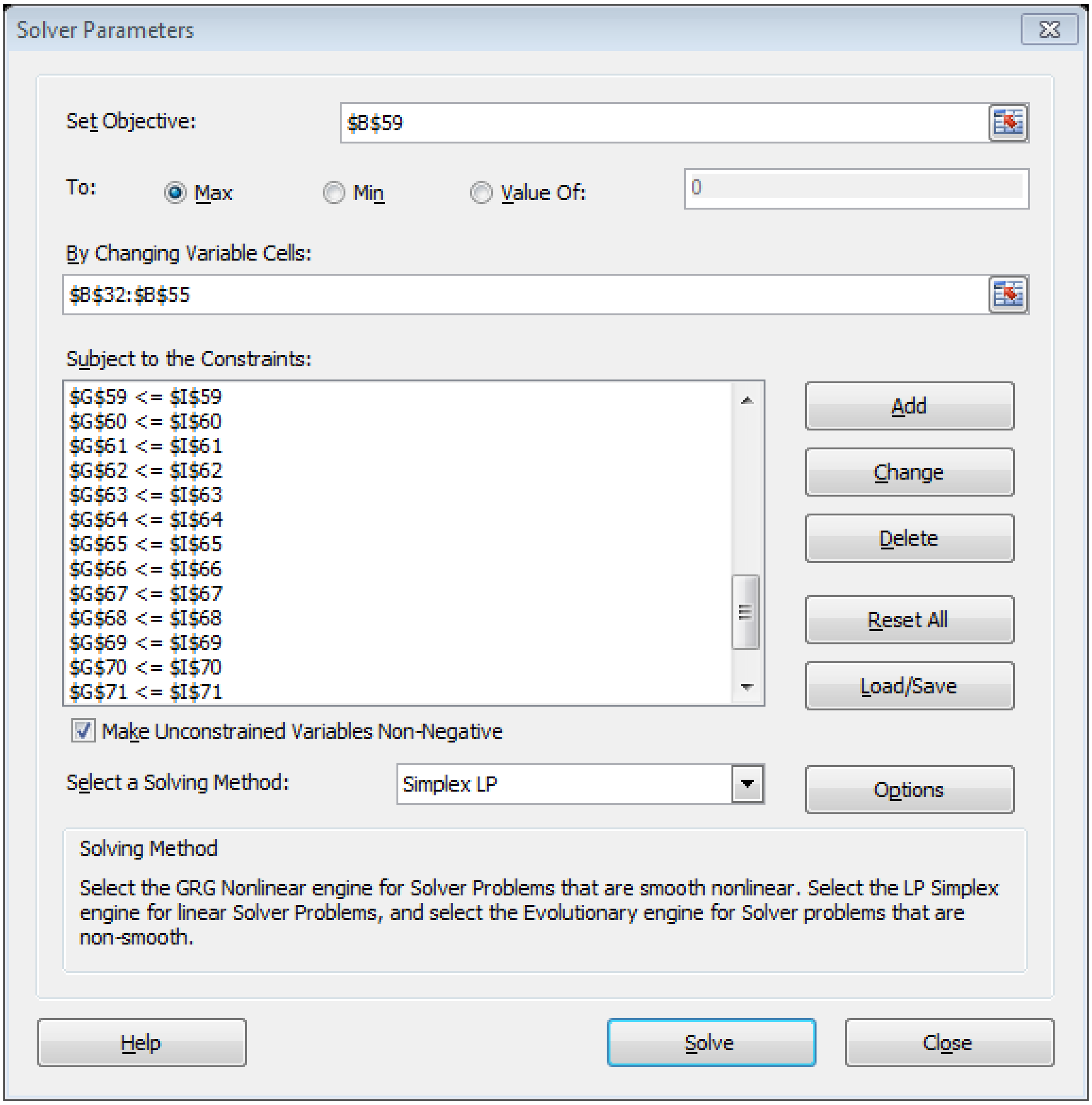
Used	Used	RHS
=IF(F32<=F$3,L32-B$42+B$43,	=G32	0
=IF(F33<=F$3,L33-B$42+B$43,	=G33	0
=IF(F34<=F$3,L34-B$42+B$43,	=G34	0
=IF(F35<=F$3,L35-B$42+B$43,	=G35	0
=IF(F36<=F$3,L36-B$42+B$43,	=G36	0
=IF(F37<=F$3,L37-B$42+B$43,	=G37	0
=IF(F38<=F$3,L38-B$42+B$43,	=G38	0
=IF(F39<=F$3,L39-B$42+B$43,	=G39	0
=IF(F40<=F$3,L40-B$42+B$43,	=G40	0
=IF(F41<=F$3,L41-B$42+B$43,	=G41	0
=H42	=B32+B33+B34+B35+B36+B37+B38+B39+B4(	1
=IF(F43<=F$2,L42-B$54+B$55,	=G43	0
=IF(F44<=F$2,L43-B$54+B$55,	=G44	0
=IF(F45<=F$2,L44-B$54+B$55,	=G45	0
=IF(F46<=F$2,L45-B$54+B$55,	=G46	0
=IF(F47<=F$2,L46-B$54+B$55,	=G47	0
=IF(F48<=F$2,L47-B$54+B$55,	=G48	0
=IF(F49<=F$2,L48-B$54+B$55,	=G49	0
=IF(F50<=F$2,L49-B$54+B$55,	=G50	0
=IF(F51<=F$2,L50-B$54+B$55,	=G51	0
=IF(F52<=F$2,L51-B$54+B$55,	=G52	0
=B44+B45+B46+B47+B48+B49	=G53	1
=B42-B43+B54-B55	=G54	0
=B32		=IF(F55<=F$2,1,0)
=B33		=IF(F56<=F$2,1,0)
=B34		=IF(F57<=F$2,1,0)
=B35		=IF(F58<=F$2,1,0)
=B36		=IF(F59<=F$2,1,0)
=B37		=IF(F60<=F$2,1,0)
=B38		=IF(F61<=F$2,1,0)
=B39		=IF(F62<=F$2,1,0)
=B40		=IF(F63<=F$2,1,0)
=B41		=IF(F64<=F$2,1,0)
=B44		=IF(F65<=F$3,1,0)
=B45		=IF(F66<=F$3,1,0)
=B46		=IF(F67<=F$3,1,0)
=B47		=IF(F68<=F$3,1,0)
=B48		=IF(F69<=F$3,1,0)
=B49		=IF(F70<=F$3,1,0)
=B50		=IF(F71<=F$3,1,0)
=B51		=IF(F72<=F$3,1,0)
=B52		=IF(F73<=F$3,1,0)
=B53		=IF(F74<=F$3,1,0)

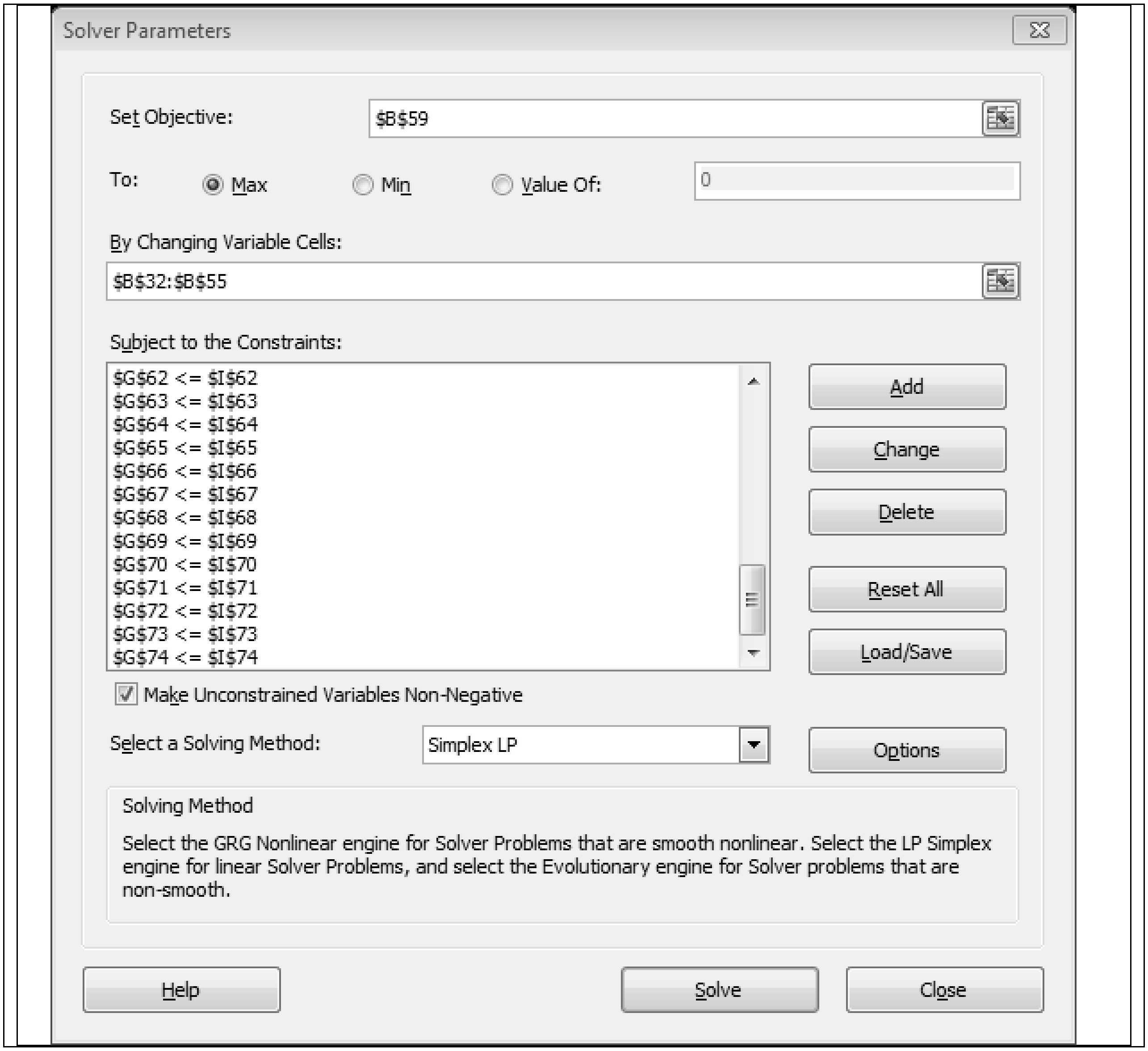

Solver Parameters
Set Objective:
$B$59
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$B$32:$B$55
Subject to the Constraints:
$G$32 >= $I$32
$G$33 >= $I$33
$G$34 >= $I$34
$G$35 >= $I$35
$G$36 >= $I$36
$G$37 >= $I$37
$G$38 >= $I$38
$G$39 >= $I$39
$G$40 >= $I$40
$G$41 >= $I$41
$G$42 = $I$42
$G$43 >= $I$43
$G$44 >= $I$44
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close

The remaining constraints are shown here:



Finally,


Solver Parameters
Set Objective:
$B$59
To:
Max
Min
Value Of:
0
By Changing Variable Cells:
$B$32:$B$55
Subject to the Constraints:
$G$59 <= $I$59
$G$60 <= $I$60
$G$61 <= $I$61
$G$62 <= $I$62
$G$63 <= $I$63
$G$64 <= $I$64
$G$65 <= $I$65
$G$66 <= $I$66
$G$67 <= $I$67
$G$68 <= $I$68
$G$69 <= $I$69
$G$70 <= $I$70
$G$71 <= $I$71
Add
Change
Delete
Reset All
Load/Save
Make Unconstrained Variables Non-Negative
Select a Solving Method:
Simplex LP
Options
Solving Method
Select the GRG Nonlinear engine for Solver Problems that are smooth nonlinear. Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver problems that are non-smooth.
Help
Solve
Close



In the template created for student use, Figure 1, the student enters only what is highlighted in within the template. This consists of the number of rows (cell F2), the number of columns (cell F3), the payoff matrix for Rose (cells B2:K16), and initial values for the decision variables usually set at 0 (cells B32: B55). The cells L2 and L3 are the values of the game and cells L18:L27 & B28:K28 are the strategies played by Rose and Colin, respectively. We illustrate the template with two examples one with pure strategy solution and the other with mixed strategy solutions.

Example 1. Pure Strategy *4 x 4* Game

Consider the following zero sum game displayed in the payoff matrix for Rose.

		C1	C2	C3	C4
	R1	4	3	2	5
Rose	R2	-10	2	0	-1
	R3	7	5	2	3
	R4	0	0	-4	-5

There is a pure strategy solution at R1C3 and R3C3, with value of 2 to Rose. Since this is a zero sum game we also know Colin's game value of -2. We may use the template to insert this game and obtain a solution.

	A	B	C	D	E	F	G	H	I	J	K	L
1	This template will allow you to solve up to 10 strategies for each player in a two-person zero-sum game										Game Values	
2	Enter the number of Strategies for Rose					4					Rose	2
3	Enter the number of Strategies for Colin					4					Colin	-2
4												
5	Enter the payoff to the Row player only											
6	R/C	1	2	3	4	5	6	7	8	9	10	
7	1	4	3	2	5	0	0	0	0	0	0	
8	2	-10	2	0	-1	0	0	0	0	0	0	
9	3	7	5	2	3	0	0	0	0	0	0	
10	4	0	0	-4	-5	0	0	0	0	0	0	
11	5	0	0	0	0	0	0	0	0	0	1	
12	6	0	0	0	0	0	0	0	0	0	0	
13	7	0	0	0	0	0	0	0	0	0	0	
14	8	0	0	0	0	0	0	0	0	0	0	
15	9	0	0	0	0	0	0	0	0	0	0	
16	10	0	0	0	0	0	0	0	0	0	0	
17	R/C	1	2	3	4	5	6	7	8	9	10	Rose's strategies
18	1	4	3	2	5	0	0	0	0	0	0	1.0000
19	2	-10	2	0	-1	0	0	0	0	0	0	0.0000
20	3	7	5	2	3	0	0	0	0	0	0	0.0000
21	4	0	0	-4	-5	0	0	0	0	0	0	0.0000
22	5	0	0	0	0	0	0	0	0	0	0	0.0000
23	6	0	0	0	0	0	0	0	0	0	0	0.0000
24	7	0	0	0	0	0	0	0	0	0	0	0.0000
25	8	0	0	0	0	0	0	0	0	0	0	0.0000
26	9	0	0	0	0	0	0	0	0	0	0	0.0000
27	10	0	0	0	0	0	0	0	0	0	0	0.0000
28	Colin's	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
29	Strategies											

Figure 1. Screen capture of a proportion of the student template.

	A	B	C	D	E	F	G	H	I	J	K	L
1	This template will allow you to solve up to 10 strategies for each player in a two-person zero-sum game										Game Values	
2	Enter the number of Strategies for Rose					4					Rose	2
3	Enter the number of Strategies for Colin					4					Colin	-2
4												
5	Enter the payoff to the Row player only											
6	R/C	1	2	3	4	5	6	7	8	9	10	
7	1	4	3	2	5	0	0	0	0	0	0	
8	2	-10	2	0	-1	0	0	0	0	0	0	
9	3	7	5	2	3	0	0	0	0	0	0	
10	4	0	0	-4	-5	0	0	0	0	0	0	
11	5	0	0	0	0	0	0	0	0	0	1	
12	6	0	0	0	0	0	0	0	0	0	0	
13	7	0	0	0	0	0	0	0	0	0	0	
14	8	0	0	0	0	0	0	0	0	0	0	
15	9	0	0	0	0	0	0	0	0	0	0	
16	10	0	0	0	0	0	0	0	0	0	0	
17	R/C	1	2	3	4	5	6	7	8	9	10	Rose's strategies
18	1	4	3	2	5	0	0	0	0	0	0	0.0000
19	2	-10	2	0	-1	0	0	0	0	0	0	0.0000
20	3	7	5	2	3	0	0	0	0	0	0	1.0000
21	4	0	0	-4	-5	0	0	0	0	0	0	0.0000
22	5	0	0	0	0	0	0	0	0	0	0	0.0000
23	6	0	0	0	0	0	0	0	0	0	0	0.0000
24	7	0	0	0	0	0	0	0	0	0	0	0.0000
25	8	0	0	0	0	0	0	0	0	0	0	0.0000
26	9	0	0	0	0	0	0	0	0	0	0	0.0000
27	10	0	0	0	0	0	0	0	0	0	0	0.0000
28	Colin's	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
29	Strategies											

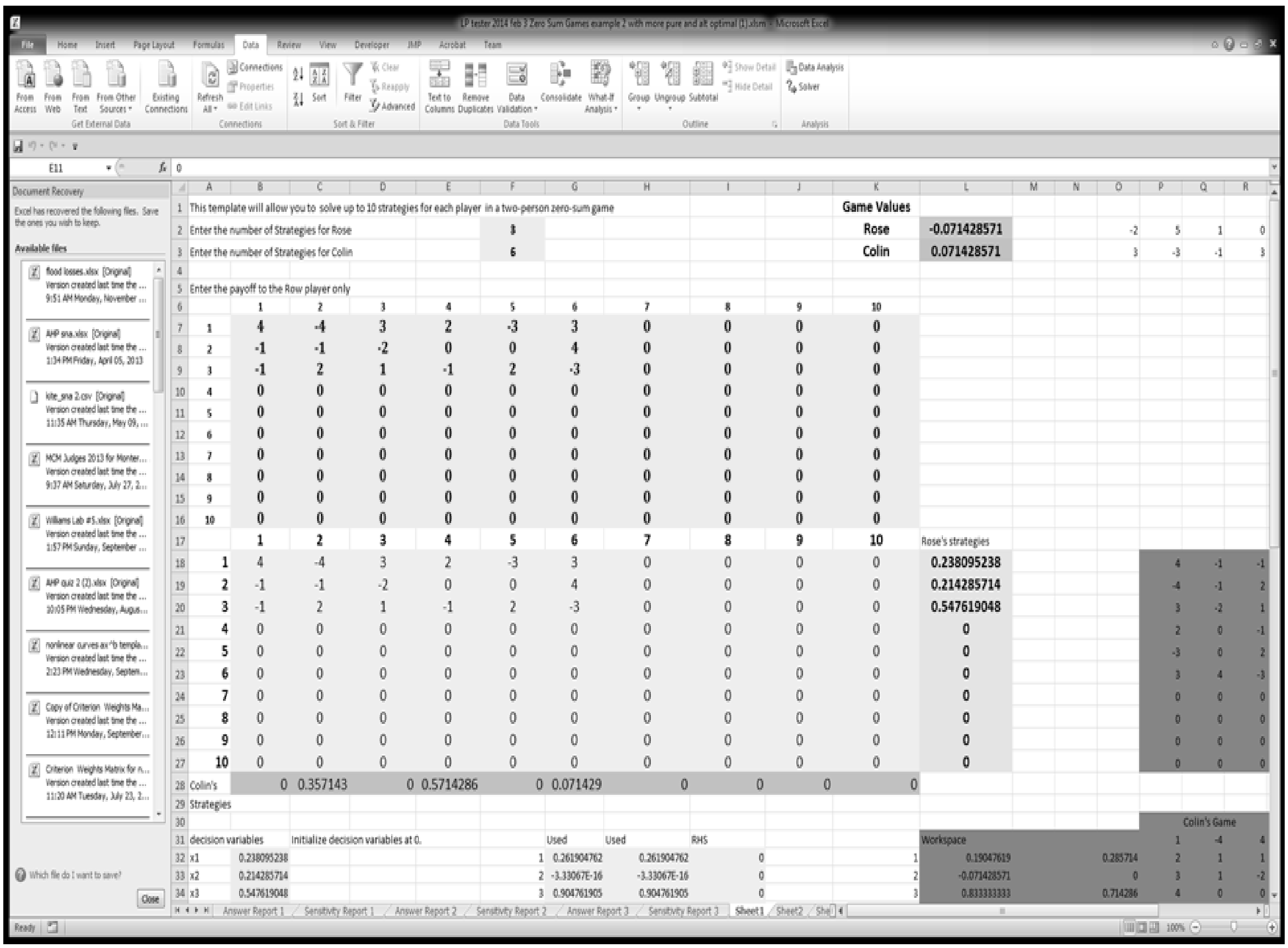
With the template we have found the pure strategy solution at R1C3. So how do we know if there are alternate optimal solutions? Fox [11] provides a methodology with examples to show how to use MS-Excel for this. Using the steps provided by Fox [11] to find an alternate optimal solution, our solution is now at R3C3.

Example 2. Mixed Strategy *3 x 6* Game

Consider the following zero sum game displayed in the payoff matrix for Rose.

		C1	C2	C3	C4	C5	C6
	R1	4	-4	3	-2	-3	3
Rose	R2	-1	-1	-2	0	0	4
	R3	-1	2	1	-1	2	-3

This game does not have a pure strategy solution so we need to find the mixed strategy solution. Our linear programming template will find either pure or mixed strategy solutions. We put the payoff matrix into the template and let the template know there are 3 rows and 6 columns as asked in cells F2:F3.



The template output solution states that the values of the game are:

Rose -0.71428571

Colin 0.71428571

Rose plays R1 with 23.8095238%, R2 with 21.4285714%, and R3 with 54.7619048% to achieve her best solution and Colin never plays C1, C4, or C6 and plays C2 with 35.7143%, C3 with 57.14286%, and C5 with 7.1429% to achieve his best solution.

Conclusions

Using linear programming has added both greater flexibility and reality for our students to set up and solve more complex game theory models. Our students, who use game theory for their projects, use the templates to obtain the solutions. Copies of this template can be obtained by email request to wpfox@nps.edu.

References

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

2. Bazarra, Mokhtar S., Hanif D. Sherali, and C. M . Shetty. (1993). *Nonlinear Programming*. New York: Wiley.

3. Barron, E.N. (2013) Game theory: An Introduction, New York: Wiley.

4. Crawford, Vincent. (1974). Learning the optimal strategy in a zero-sum game. *Econometrica*, 42(5), pp. 885-891.

5. Danzig, George. (2002). Linear programming, *Operations Research* 50(1), pp.42-47.

6. Danzig, George (1951). Maximization of a linear function of variables subject to linear inequalities. Chapter XXI from *Activity Analysis of Production and Allocation Conference Proceeding*. T. Koopman (ed) John Wiley Publishers. pp.339-347.

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

8. Dorfman, Robert. (1951). Application of the simplex method to a game theory problem. Chapter XXII from *Activity Analysis of Production and Allocation Conference Proceeding*. T. Koopman (ed) John Wiley Publishers. pp.348-358.

9. Fox, William P. (2008) Mathematical modeling of conflict and decision making "the Writers Guild strike 2007-2008", *Computers in Education Journal*, XVIII(3), pp.2-11.

10. Fox, W.P. (2010). Teaching the applications of optimization in game theory's zero-sum and non-zero sum games, *International Journal of Data Analysis Techniques and Strategies (IDATS)*, **2**(3), pp. 258-284.

11. Fox, William P. (2014), Finding alternate optimal solutions in a zero-sum game with MS- Excel, *Computers in Education Journal*, accepted Vol. 6 (3), pp 10-19.

12. Gale, David, Harold Kuhn, and Albert Tucker. (1951). Linear programming and the theory of games. Chapter XIX from *Activity Analysis of Production and Allocation Conference Proceeding*. T. Koopman (ed) John Wiley Publishers. pp. 317-329.

13. Gillman, Rick and David Housman. (2009). *Models of Conflict and Cooperation*. Providence: American Mathematical Society. pp.189-195.

14. Giordano, F., W. Fox and S. Horton (2014). *A first course in mathematical modeling*. Boston: Cengage Publishers.

15. Klarrich, Erica (2009). The mathematics of strategy. *Classics of the Scientific Literature*. Oct 2009 (http://www.pnas.org/site/misc/classics5.shtml)

16. Koopman, T. (1951). *Activity Analysis of Production and Allocation Conference Proceeding*. John Wiley Publishers..

17. Kuhn, H.W. and A.W. Tucker. (1951). *Nonlinear Programming*. Proceedings 2nd Berkley Symposium on Mathematical Statistics and probability. J Newman (ed). University of California Press, CA.

18. Nash, John (1950). The bargaining problem, *ECONOMETRICA* 18, pp.155-162.

19. Nash, John. (1951). Non-Cooperative Games, *Annals of Mathematics*, 54, pp.289-295.

20. Nash, John (2009). Lecture at NPS. Feb 19, 2009.

21. Straffin, Philip D. (2004). *Game Theory and Strategy*. Washington: Mathematical Association of America.

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

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