An Algorithm and Its Computer Technique for Solving Game Problems Using LP method

H. K. Das\(^1\) and M. B. Hasan\(^2\)

Abstract—Game theory is the formal study of decision-making where several players must make choices that potentially affect the interests of the other players. In this paper, we develop a combined computer technique for solving game problems. In this technique, we implement Minimax-Maximin method, rectangular game, and convert the resulting game into Linear Programming (LP) problem. Finally, to find out the strategies of both players, we incorporate the usual Simplex method of Dantzig [11]. We develop our computer technique by using the programming language MATHEMATICA [16]. We will show the efficiency of our program for solving game problems by presenting a number of numerical examples.

Index Term—Game, Linear Programming, Minimax-Maximin, Simplex, Strategy.

I. INTRODUCTION

Game theory is a mathematical theory that deals with the general features of competitive situations like parlor games, military battles, political campaigns, advertising and marketing campaigns by competing business firms and so forth. It is a distinct and interdisciplinary approach to the study of human behavior. The disciplines most involved in game theory are mathematics, economics and the other social and behavioral sciences. Game theory (like computational theory and so many other contributions) was founded by the great mathematician John von Neumann [23]. The concepts of game theory provide a language to formulate structure, analyze, and understand strategic scenarios.

Game theory bears a strong relationship to LP, since every finite two-person zero sum game can be expressed as a LP and conversely every LP can be expressed as a game. If the problem has no saddle point, dominance is unsuccessful to reduce the game and the method of matrices also fails, then LP offers the best method of solution. So far several authors namely Bansal [2], Martin [3], Stephen [1], Theodor [4] and many other authors proposed different types of theoretical discussion of game problems with their strategies also. But they didn’t discuss computational procedure of game theory. Also none of them discussed the whole problem comprehensively. For this, there is a need to develop a technique which can address all type of game problems within a single framework. In this paper, we develop such a computer technique to find their strategies and the value of the game.

The rest of the paper is organized as follows. In Section II, we will discuss about different definitions of LP and game theory with some relevant theorems and propositions. In Section III, we will give a short discussion of simplex method and Minimax-Maximin method for solving game problems. A short discussion of rectangular 2x2 game will be given in section IV. In Section V, we will present our prime object by showing the construction of the generalized m x n computer technique with some numerical examples for the justification of our program. Also time comparison in second is given in Section VI. An over view focus on the above sections will be viewed in Section VII.

II. PRELIMINARIES

In this section, we briefly discuss some definitions of LP and game theory. We also include some propositions and theorems. For this, we first briefly discuss the general LP problems. Consider the standard LP problems as follows,

\[
\max \quad c^T x
\]

Subject to \quad \begin{align*}
Ax &\leq b \\
x &\geq 0
\end{align*}

Where \(A\) is an \(m \times n\) matrix and \(x=(x_1, x_2, x_3, \ldots, x_n)\), \(b=(b_1, b_2, b_3, \ldots, b_m)^T\) are column vectors. We shall consider any number of rows and columns, \(b\neq 0\) and the system of linear equations are given in equation (2). We shall also denote the \(i^{th}\) column of \(A\) by \(A^{(i)}\).

A. Objective function

The linear function \(z=\sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n\) which is to be maximized (or minimized) is called objective function of the general linear programming problem (GLPP).

B. Constraints

The set of equations or inequalities is called the constraints of the general linear programming problem. \(Ax \leq b\) is the set of constraints in the GLPP.

C. Solution of GLPP

An \(n\)-tuple \((x_1, x_2, \ldots, x_n)\) of real numbers which satisfies the constraints of a GLPP is called the solution of GLPP.

D. Feasible Solution

Any solution to a GLPP which also satisfies the nonnegative restrictions of the problem is called a feasible solution to the GLPP [8]. Or, A feasible solution to the LP problem is a vector \(x\)
(x₁, x₂, ..., xₙ) which satisfies the conditions \( \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \) \( i = 1, \ldots, m \) and \( j = 1, \ldots, n; \) \( x_j \geq 0. \)

### E. Matrix Form

Suppose we have found the optimal solution to (1). Let \( BV_i \) be the basic variable for row \( i \) of the optimal tableau. Also define \( BV = \{ BV_1, BV_2, \ldots, BV_m \} \) to be the set of basic variables in the optimal tableau, and define the \( m \times 1 \) vector as,

\[
x_{BV} = \begin{bmatrix} x_{BV_1} \\ x_{BV_2} \\ \vdots \\ x_{BV_m} \end{bmatrix}
\]

We also define \( NBV = \) the set of nonbasic variables in the optimal tableau

\[
x_{NBV} = (n - m) \times 1 \text{ vector listing the nonbasic variables (in any desired order)}
\]

Using our knowledge of matrix algebra, we can express the optimal tableau \( n \) terms of \( BV \) and the original LP (1). Recall that \( c_1, c_2, \ldots, c_n \) are the objective function coefficients for the variables \( x_1, x_2, \ldots, x_n \) (some of these may be slack, excess or artificial variables).

Here, \( c_{BV} \) is the \( 1 \times m \) row vector \([ c_{BV_1} \ c_{BV_2} \ \ldots \ c_{BV_m} \] \)

Thus the elements of \( c_{BV} \) are the objective function coefficients for the optimal tableau’s basic variables. \( c_{NBV} \) is the \( 1 \times (n - m) \) row vector whose elements are the coefficients of the non basic variables (in the order of \( NBV \)). The \( m \times m \) matrix \( B \) is the matrix whose \( j \)th column is the column for \( BV_j \) in (1), \( a_i \) is the column (in the constraints) for the variable \( x_j \) in (1). \( N \) is the \( m \times (n - m) \) matrix whose columns are the columns for the non-basic variables (in the \( NBV \) order) in (1). The \( m \times 1 \) column vector \( b \) is the right-hand side of the constraints in (1).

### F. Game

A game is a formal description of a strategic situation [25].

### G. Strategy

A strategy of a player ‘p’ is a complete enumeration of all the actions that he will take for every contingency that might arise [20].

### H. Payoff

The payoff is a connecting link between the sets of strategies open to all the players. Suppose that at the end of a play of a game, a player \( p_i \) (i=1,2,........,n) is expected to obtain an amount \( v_i \) called the payoff to the player \( p_i \).

#### I. Payoff matrix

A payoff matrix is the table that represents the payoff from player II to player I for all possible actions by players [22].

#### J. Fair game

A game is said to be fair game if the value of the game is zero.

### K. Pure strategy

A pure strategy for player I (or player II) is the decision to play the same row (or column) on every move of the game [20].

Consider the matrix game \( A = \{ a_{ij} \} \) for two players. If both players employ pure strategies, the outcome of each move is exactly the same and the game is completely predictable. For example, if player I always chooses the ith row and player II always chooses the jth column, then on every play of the game player I receive \( a_{ij} \) units from player II.

### L. Mixed strategy

A mixed strategy is an active randomization, with given probabilities that determine the player’s decision. As a special case, a mixed strategy can be the deterministic choice of one of the given pure strategies.

Suppose player I does not want to play each row on each play of the game with probability 1 or 0, as was the case with pure strategies. Instead, suppose he decides to play row \( i \) with probability \( x_i \), where more than one \( x_i \) is greater than zero, and \( \sum_{i=1}^{n} x_i = 1 \). This decision is denoted by

\[
X=\begin{bmatrix} x_1 & x_2 & \cdots & x_p & \cdots & x_m \end{bmatrix}
\]

is called a mixed strategy for player I [18]. In like manner, if player II decides to play column \( j \) with probability \( y_j \) with \( j=1,2,\ldots,\ldots,n \) where more than one \( y_j \) is greater than zero, and \( \sum_{j=1}^{n} y_j = 1 \).

Then

\[
Y=\begin{bmatrix} y_1 & y_2 & \cdots & y_p & \cdots & y_n \end{bmatrix}
\]

### M. Player

A player is an agent who makes decisions in a game.

### N. Strategic form

A game in strategic form, also called normal form, is a compact representation of a game in which players simultaneously choose their strategies. The resulting payoffs are presented in a table with a cell for each strategy combination.

#### O. Two-person zero-sum game

A game is said to be zero-sum if for any outcome, the sum of the payoffs to all players is zero. In a two-player zero-sum game, one player’s gain is the other player’s loss, so their interests are diametrically opposed [21].

### P. Saddle point

A saddle point of a payoff matrix is that position in the payoff matrix where the maximum of row minima coincides with the minimum of the column maxima. The payoff at the saddle point is
called the value of the game and is obviously equal to the maximin
and minimax values of the game.

Theorem 2.1

If mixed strategies are allowed, the pair of mixed strategies that is
optimal according to the minimax criterion proves a stable
solution with \( V = \bar{V} = \ddot{V} \), so that neither player can do better by
unilaterally changing her or his strategy [10, 26].

Theorem 2.2

In a finite matrix game, the set of optimal strategies for each
player is convex and closed [14].

Theorem 2.3

Let, \( v \) be the value of an \( m \times n \) matrix game. Then if
\( Y=[y_1 \ y_2 \ \cdots \ y_p \ \cdots \ y_n] \) is an optimal strategy
for player II with \( y_i > 0 \), every optimal strategy \( x \) for player I
must have the property
\[
\sum_{i=1}^{m} a_{ij}x_i = \bar{v}
\]
Similarly, if the optimal strategy \( x \) has \( x_i > 0 \), then the optimal
strategy \( y \) must be that
\[
\sum_{i=1}^{n} a_{ij}y_i = \bar{v}
\]

Proposition 2.1: The set \( S= \{ x | Ax = b, x \geq 0 \} \) is convex.

Proposition 2.2: \( x \geq 0 \) is a basic nonnegative solution of (2) if and
only if \( x \) is a vertex of (1).

Proposition 2.3: If the system of equations (2) has a nonnegative
solution, then it has a basic nonnegative solution.

Proposition 2.4: \( S \) has only a finite number of vertices [15, 17].

III. TWO EXISTING METHODS [13, 15]

In this section, we briefly discuss of the usual Simplex method
and Minimax-Maximin method.

3.1. Simplex Method

The Simplex method is an iterative procedure for solving linear
programming problems expressed in standard form. In addition to
the standard form, the Simplex method requires that the constraint
equations be expressed as an economical system from which a
basic feasible solution can be readily obtained. If the standard
deficit of LP is not in canonical form, one has to reduce it to a
variable. Then we remove these artificial variables by applying
two- phase method or Big-M method. The Simplex method is
has a wide range of applications including agriculture, industry,
transportation and other problems in economics and management
science.

3.2. Minimax-Maximin pure strategies

Since each player knows that the other rational and same objective
that is, to maximize the pay off from the other player, each might
decide to us the conservative minimax criterion to select an action.
That is, player I examines each row in the payoff matrix and selects
the minimum element in each row, say \( p_{i0} \) with \( i=1,2 \)
\( \cdots \) \( m \). Then he selects the maximum of these minimum
elements, say \( p_{\ddot{v}} \).

Mathematically,
\( V = p_{\ddot{v}} = \max \{ \min (p_{ij}) \} \)
The element \( p_{\ddot{v}} \) is called the maximin value of the game, and
the decision to play row \( r \) is called the maximin pure strategy.
Likewise, player II examines each column in the payoff matrix to
the column with the smallest maximum loss. Let,
\( V = p_{\ddot{v}} = \min \{ \max (p_{ij}) \} \)
Then \( p_{\ddot{v}} \) is called the minimax value of the game and the decision
to play column \( u \) is called minimax pure strategy. It can be shown
that, the minimax value \( v \) represents a lower bound on a quantity
called the value of the game, and also \( v \) represents an upper bound
on the value of the game.

IV. RECTANGULAR 2×2 GAME

In this section, we present a short discussion about 2×2 particular
game problems [19].

First, consider an 2×2 game with the payoff matrix.

Let \( x_i \) be the probability player II plays row \( i \) with \( i = 1,2 \), and
let \( y_j \) be the probability player I plays column \( j \) with \( j = 1,2 \). Since

\[
\begin{bmatrix}
\text{Player I}
\end{bmatrix}
\begin{bmatrix}
p_{11} & p_{12} \\
p_{21} & p_{22}
\end{bmatrix} \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.1)
\]

\[
\sum_{i=1}^{2} x_i = 1 \quad \text{and} \quad \sum_{i=1}^{2} y_i = 1
\]
So we can write, \( x_1 = 1 - x_1 \) and \( y_1 = 1 - y_1 \).

The saddle point is necessarily the value of the game. If a saddle
point does not exist, then we have to follow the following
procedure. Let,

The optimal strategy of player I is \( \bar{y} = \left( \begin{array}{c}
y_1^* \\
y_2^*
\end{array} \right) \)

The optimal strategy of player II is \( \bar{x} = \left( \begin{array}{c}
x_1^* \\
x_2^*
\end{array} \right) \)

\[
y_i^* = \frac{p_{22} - p_{21}}{p_{11} + p_{22} - p_{12} - p_{21}} \ldots \ldots \ldots \ldots \ldots (4.2)
\]

\[
y_i = 1 - y_i^* \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.3)
\]
\[x_i^* = \frac{p_{22} - p_{12}}{p_{11} + p_{22} - p_{12} - p_{21}} \ldots \ldots \ldots \ldots \ldots (4.4)\]
\[x_i^* = 1 - x_i^* \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.5)\]

These will be optimal minimax strategies for player I and player II
respectively.

Finally the value of the game is
\[
V = y_i^* x_i^* p_{11} + y_i^* (1-x_i^*) p_{12} + (1-y_i^*) x_i^* p_{21} +
\]
V. SOLVING GAME PROBLEMS REDUCING INTO LP

In this section, we discuss generalized m x n game problems for converting it into LP to find the two players strategies with the help of LP method.

In many applications, one needs to compute basic solutions of a system of linear equations. For example, in dealing with many linear programming problems, especially degenerate and cycling problems [5], it is often more convenient to locate the extreme points by applying the usual simplex method. In this paper, we outline a procedure for finding two strategies of a system of m equations in n variables and develop a computer procedure using the computer algebra MATHMATICA [11, 16]. For this, we first apply Minimax-Maximin and Rectangular method to find game value in section 5.2.1. Finally, we will solve [5.a] by usual Simplex method in our code in section 5.3.1. On the other hand, our method yields the game value easily.

5.1 A Short Discussion of m x n Game [10]

Any Game with mixed strategies can be solved by transforming the problem to a linear programming problem. Let, the value of game is v. Initially, player I acts as maximize and player II acts as minimize. But after transforming some steps when we convert the LP then inverse the value of the game. For this objective function also changes.

First consider, the optimal mixed strategy for player II:

Expected payoff for player II = \( \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_j \) and the player II strategy \((x_1, x_2, \ldots, x_m)\) is optimal if \( \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_j \leq v \) for all opposing strategies i.e. player I is \((y_1, y_2, \ldots, y_n)\). After some necessary calculations we get the following two forms of player II and player I respectively.

Player II:
Maximize, \( \frac{1}{v} = x_1+x_2+\ldots+x_m \)

Subject to,
\[
p_{11}x_1 + p_{12}x_2 + \ldots + p_{1n}x_n \leq 1
\]
\[
p_{21}x_1 + p_{22}x_2 + \ldots + p_{2n}x_n \leq 1
\]
\[
\vdots
\]
\[
p_{m1}x_1 + p_{m2}x_2 + \ldots + p_{mn}x_n \leq 1
\]
\[x_1 + x_2 + \ldots + x_m = 1\]
\[x_j \geq 0, \text{for } j = 1, 2, \ldots, n\]

Player I:
Minimize, \( \frac{1}{v} = y_1+y_2+\ldots+y_m \)

Subject to,
\[
p_{11}y_1 + p_{12}y_2 + \ldots + p_{1m}y_m \geq 1
\]
\[
p_{21}y_1 + p_{22}y_2 + \ldots + p_{2m}y_m \geq 1
\]
\[
\vdots
\]
\[
p_{m1}y_1 + p_{m2}y_2 + \ldots + p_{mn}y_m \geq 1
\]
\[y_j \geq 0, \text{for } j = 1, 2, \ldots, m\]

We can solve (5.a) and (5.b) by suitable LP method such as usual simplex method or big M simplex method or Primal-dual simplex method [7]. In this paper we will develop a computer technique incorporate with usual Simplex method.

5.1.1 Numerical Example

All game problems can be solved by our procedure. Here we consider a real life problem, which illustrates the implementation and advantage of the above procedure.

Two oil companies, Bangladesh Oil Co. and Caltex, operating in a city, are trying to increase their market at the expense of the other. The Bangladesh (B. D.) Oil Co. is considering possibilities of decreasing price, giving free soft drinks on Rs. 40 purchases of oil or giving away a drinking glass with each 40 litter purchase. Obviously, Caltex cannot ignore this and comes out with its own program to increase its share in the market. The payoff matrix forms the viewpoints of increasing or decreasing market shares is given in table below.

<table>
<thead>
<tr>
<th></th>
<th>Caltex</th>
<th>Decrease</th>
<th>Free soft drinks on Rs.40 purchase</th>
<th>Free drinking glass on 40 liters or so</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.D. Oil Co.</td>
<td></td>
<td>Decrease</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4%</td>
<td>1%</td>
<td>-3%</td>
</tr>
<tr>
<td></td>
<td>Free soft drinks on Rs.40 purchase</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Free drinking glass on 40 liters or so</td>
<td>-3</td>
<td>4</td>
<td>-2</td>
</tr>
</tbody>
</table>

Determine the optimum strategies for the oil company [12].

5.1.2 Solution


5.2 Algorithm

In this section, we first discuss the algorithm of the game by Minimax-Maximin, 2x2 strategies and for the modified matrix of the game problems.

Step (1): If the pay-off matrix is 2x2 then find the game value.
**Sub step (I):** Search the maximum element from each row of the payoff matrix of equation (4.a).

**Sub step (II):** Search the minimum element from each column of the payoff matrix of equation (4.a).

**Sub step (III):** If they coincide then the value of the game is \( V = \text{Maximin element}=\text{Minimax element.} \) Then Stop .If we fail to get such value, go to Sub step (IV).

**Sub step (IV):** Find the mixed strategies for player I using (4.b) and (4.c).

**Sub step (V):** Find the mixed strategies for player II using (4.d) and (4.e).

**Sub step (VI):** Finally, we get value of the game by (4.f). Otherwise go to Step (2) for \( m, n > 2 \).

**Step (2):** Search the minimum element from each row of the reduced payoff matrix and then find the maximum element of these minimum elements.

**Step (3):** Search the maximum element from each column of the reduced payoff matrix and then find the minimum element of these maximum elements.

**Step (4):** For the player I if the Maximin less than zero then find \( k \) which is equal to addition of one and absolute value of Maximin.

**Step (5):** For the player II if the Minimax less than zero then find \( k \) which is equal to addition of one and absolute value of Minimax.

**Step (6):** If Maximin and Minimax both are greater than zero then \( k \geq 0 \).

**Step (7):** Finally to get the modified payoff matrix adding \( k \) with each payoff elements of the given payoff matrix.

**Step (8):** Then to find the mixed strategies with game value of the two players, follow the algorithm of section 5.3.

5.2.1 Program

In this section, we construct a computer technique for finding the modified matrix. Also we use Minimax-Maximin for finding value of the game and Rectangular 2x2 game also find value of the game with their strategies using MATHEMATICA [11, 16].

```mathematica
Modify[AA_] := 
  Block[{m, n, d, dd, cc, ccc, pp, qg, r, s, xx, yy, mm, 
    x1, x2, y1, y2, v, m5, n5, var4, var5, r1, e, k, hh, AAA}, 
  If[m == n == 2, 
    d = TakeRows[AA, {1}]; dd = TakeRows[AA, {2}]; 
    cc = TakeColumns[AA, {1}]; ccc = TakeColumns[AA, {2}]; 
    pp = Min[d]; qg = Min[dd]; r = Max[cc]; s = Max[ccc]; 
    xx = {pp, qg}; yy = {r, s}; 
    mm = Max[xx]; mn = Min[yy]; 
    If[mm == mn, Print["value of the game is "; mm, 
      "\(\text{Min in this case we can find strategies by the following code.}\)"]; 
      x1 = (Part[AA, 2, 2] - Part[AA, 2, 1]) / 
        (Part[AA, 1, 1] + Part[AA, 2, 2] - Part[AA, 1, 2] - Part[AA, 2, 1]); 
      y1 = (Part[AA, 2, 2] - Part[AA, 1, 2]) / 
        (Part[AA, 1, 1] + Part[AA, 2, 2] - Part[AA, 1, 2] - Part[AA, 2, 1]); 
      v = x1 + y1; 
      \(\text{Print["value of the game is "; v \(/\) /\];}\) 
      \(\text{Print["Optimal solution of player A strategies = "; x1, y1];}\) 
      \(\text{Print["Optimal solution strategies B strategies = "; y1, v];}\) 
      \("nIn this case no need to go in the next code because this conclude all the necessary things.\)];}, 
    \(\text{nn}5, n5 = \text{Dimensions}[AA];\) 
    \(\text{var4 = Table}[R[1], \{1, 1, n5\}];\) 
    \(\text{var5 = Table}[T[1], \{1, 1, n5\}];\) 
  ] 
  \(\text{r1 = Table}[AA[[i]][[1]], \{i, 1, m5\}, \{j, 1, n5\}];\) 
  \(\text{For}[i = 1, i \leq m5, i = i + 1,\) 
    \(\text{k}[[i]] = \text{Min}[r1[[i]]];\) 
  \(\text{Max}[var4] = \text{e} = \text{Table}[AA[[i]][[1]], \{i, 1, n5\}, \{j, 1, m5\}];\) 
  \(\text{For}[i = 1, i \leq n5, i = i + 1,\) 
    \(\text{y1}[[i]] = \text{Max}[e[[i]]];\) 
  \(\text{Min}[var5] = \text{Max}[[\text{var4} - \text{Min}[\text{var5}] = 0,\) 
    \(\text{Print["The value of the game is "; \text{Max}[\text{var4}],\) 
    \(\text{If}[\text{Max}[\text{var4}] < 0,\) 
    \(\text{k} = \text{Abs}[\text{Max}[\text{var4}]] + 1,\) 
    \(\text{If}[\text{Min}[\text{var5}] < 0,\) 
    \(\text{k} = \text{Abs}[\text{Min}[\text{var5}]] + 1,\) 
    \(\text{k} = 0]\) 
    \(\text{Print["The require value of k is "; k,}\) 
    \(\text{f}[[i]] = \text{Array}[f, \{m5, n5\}];\) 
    \(\text{AAA} = \text{f} + \text{hh} + \text{AA} + \text{f};\) 
    \(\text{Print["The require modified matrix is "; MatrixForm[AAA],\) 
      "\nWe take input this modified matrix and k to find their strategies and game value by Linear Programming method." ];\)]
  ]
]
```

5.2.1.1 Input of Example 5.1.1

\[ AA = \{(4, 1, -3), (3, 1, 6), (-3, 4, -2)\}; \]

5.2.1.2 Output of Example 5.1.1

The require value of k is 0

The require modified matrix is

\[
\begin{pmatrix}
4 & 1 & -3 \\
3 & 1 & 6 \\
-3 & 4 & -2
\end{pmatrix}
\]

We take input this modified matrix and k to find their strategies and game value by Linear Programming method.

5.3 Algorithm for player I and player II

In this section, we present our computational procedure incorporated with simplex method in terms of some steps for finding their strategies with the game value from the modified matrix for \( m \times n \) game problems.

**Step (1):** First, take the modified payoff matrix for the player II and player I and the value of k.
Step (2): We will get equations (5.a) and (5.b) for the player II and player I respectively.

Step (3): We take input for player II from the equation (5.a).

Step (4): Define the types of constraints. If all are of “≤” type goes to step (6).

Step (5): We follow the following sub-step.

Sub-step (I): Express the problem in standard form.

Sub-step (II): Start with an initial basic feasible solution in canonical form and set up the initial table.

Sub-step (III): Use the inner product rule to find the relative profit factors \( \bar{C}_j \) as follows \( \bar{C}_j = c_j - Z_j = c_j - ( \text{inner product of } c_B \text{ and the column corresponding to } x_j \text{ in the canonical system}) \).

Sub-step (IV): If all \( \bar{C}_j \leq 0 \), the current basic feasible solution is optimal and stop. Otherwise select the non-basic variable with most positive \( \bar{C}_j \) to enter the basis.

Sub-step (V): Choose the pivot operation to get the table and basic feasible solution.

Sub-step (VI): Go to Sub-step (III).

Step (6): At first express the problem in standard form by introducing slack and surplus variables. Then express the problem in canonical form by introducing artificial variables if necessary and form the initial basic feasible solution. Go to Sub-step (III).

Step (7): If any \( \bar{C}_j \) corresponding to non-basic variable is zero, the problem has alternative solution, take this column and go to Sub-step (V).

Step (8): Finally, we find all the strategies for player II is in corresponding their right hand side (RHS) and strategies of player I is in corresponding the \( \bar{C}_j = c_j - Z_j \) of the slack variables.

Step (9): Calculate the value of the object functions for each feasible solution.

5.3.1 Program

In this section, we construct a computer technique for finding the player I and player II strategies with the game value.
...
The program shown in the Section 5.3.1 with an algorithm in Section 5.3 is the general program for finding the strategies of \( m \times n \) game problems.

To take input as in matrix form, we convert the modified matrix as a form of the equation (5.a) in section 5.1.

**5.3.1.1 Input for the player II of Example 5.1.1**

**5.3.1.2 Output for the player I and player II of Example 5.1.1**

Hence the output of both player strategies and the game value coincide with the given example 5.1.1.

Moreover, to justify our technique we present another numerical example in this section.

### 5.1.2 Numerical Example

The payoff matrix of a game is given below.

\[
\begin{array}{ccccccc}
 & I & II & III & IV & V & VI \\
A & 1 & 4 & 2 & 0 & 2 & 1 & 1 \\
B & 4 & 2 & 3 & 1 & 3 & 2 & 2 \\
C & 3 & 4 & 3 & 7 & -5 & 1 & 2 \\
D & 4 & 4 & 3 & 4 & -1 & 2 & 2 \\
E & 5 & 4 & 3 & 3 & -2 & 2 & 2 \\
\end{array}
\]

Find the best strategy for each player, and the value of the game to A and B [12].

### 5.1.3 Solution

The Solution can be found step by step in [12].

The value of the game is 13/7.

The strategy for player A is \([0, 6/7, 1/7, 0, 0]\).

The strategy for player B is \([0, 0, 4/7, 3/7, 0, 0]\).

\[
\begin{array}{ccccccc}
 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \\
C_1 & 1 & 1 & 1 & 0 & 0 & 0 \\
C_2 & 0 & 0 & 0 & 1 & 0 & 0 \\
C_3 & 0 & 0 & 0 & 0 & 1 & 0 \\
C_4 & 0 & 0 & 0 & 0 & 0 & 1 \\
C_5 & 1 & 1 & 1 & 0 & 0 & 0 \\
\end{array}
\]
5.1.2.1 Programming Input of Example 5.1.2 in section 5.2.1

\[ AA = \{(4, 2, 0, 2, 1, 1), (4, 3, 3, 2, 2), (4, 3, 7, -5, 1, 2), \\
(4, 3, 4, -1, 2, 2), (4, 3, 3, -2, 2)\}; \]

Modify \[ AA \]

5.1.2.2 Programming Output of 5.1.2 in Section 5.2.1

The require value of \( k \) is 0

\[
\begin{pmatrix}
4 & 2 & 0 & 2 & 1 & 1 \\
4 & 3 & 1 & 3 & 2 & 2 \\
4 & 3 & 7 & -5 & 1 & 2 \\
4 & 3 & 4 & -1 & 2 & 2 \\
4 & 3 & 3 & -2 & 2 & 2
\end{pmatrix}
\]

We take input this modified matrix and \( k \) to find their strategies and game value by Linear Programming method.

5.1.2.4 Programming Input of 5.3.1 for player I & II from section 5.1.2.2

Clear \( [u, v, y] \) main[=twoBasic];

5.1.2.4 Programming Output of 5.1.2 for player I & II in section 5.3.1

Therefore we get from the output from Section 5.1.2.4

Optimal strategy for player A is: \((0, \frac{6}{7}, \frac{1}{7}, 0, 0)\) and

Optimal strategy for player B is: \((0, 0, \frac{4}{7}, \frac{3}{7}, 0, 0)\).
Game value: =13/7.

Hence the output of two player strategies and the game value coincides with that of the given example in Section 5.1.2.

Using the combined program, we have to ‘Local Kernel Input’ 5, 6, 0, 0 respectively as to indicate the number of rows, number of columns, number of greater than type constraints and the value of k from the modified matrix. Also no greater type constraints, ‘l’ five time to indicate five less than type constraints and all other inputs as prompt requirement. The program showed the simplex table iteration by iteration. And the strategies obtained which is identical with that of the Simplex method.

To solve 3×3 or higher Games also, the first step is to look for a saddle point for, if there is one, and then Game is readily solved. If the Game is 2×2 it can easily solved by the methods described in section 4. If the pay-off matrix is 2xn (or mx2 or 3x3 size), it can be solve by algebraic method, method of matrices and iterative method. But for the higher dimensions it cannot be solved successfully by the prescribed methods. But our computer technique can be solved m × n Game problems easily by method of LP.

VI. TIME COMPARISN

In this section, we compare the time required to solve the game problems by our method with that of the manual solution. Here, we use the command “TimeUsed[]” for calculating the required time in our code to find the output.

<table>
<thead>
<tr>
<th>Example No</th>
<th>Manual Time</th>
<th>1st Code Time</th>
<th>2nd Code Time</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.11</td>
<td>Much times</td>
<td>0.124 sec</td>
<td>0.202 sec</td>
<td>0.326 sec</td>
</tr>
<tr>
<td>5.12</td>
<td>Much Time</td>
<td>0.124 sec</td>
<td>0.343 sec</td>
<td>0.467 sec</td>
</tr>
</tbody>
</table>

Hence, we can say that our code is highly powerful for saving time.

VII. CONCLUSION

In this paper, we developed a combined computer technique for solving game problems. This computer technique incorporated Minimax-Maximin Method, Rectangular 2x2 Game and LP Method. Our technique can address any game problem within a single framework. We demonstrated our program and discussed the changes step by step throughout our paper. The program developed by us is a powerful computer technique. Hand calculation is very tough and time consuming for analyzing the game problems with large number of variables and constrains, where we can do the same problems by our program very easily. Nowadays, the world is being ruled by the fastest. So, we must try to finish our job as fast as we can. Finally, we can say that our computer oriented technique with MATHEMATICA for analyzing game problems is more efficient than any other methods.

VIII. ACKNOWLEDGEMENT

The author wishes to thank reviewers of IJBAS-IJENS for their important suggestions which helped to improve the presentation.

REFERENCES


***

1 Haridas Kumar Das , Department of Mathematics. University of Dhaka, Dhaka-1000, Bangladesh, E-mail:hkdas.rohit@gmail.com.

2. Dr. Md. Babul Hasan , Assistant professor, Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh, E-mail: nbabulhasan@yahoo.com.