Deakin University

SIG718- Real World Analytics
Assessment Task 3, 2023

Submitted by

Neethu Sidhardhan
S223494027
Attempt # 1
Date: 24/12/2023

Problem Statement :1

A garment factory produces shirts and pants for Kmart chain. The contract is such that quality control is done before shipping and all products supplied to Kmart satisfying the quality requirements would be accepted by the chain. The factory employs 20 workers in the cutting department, 50 workers in the sewing department, and 14 workers in the packaging department. The garment factory works 8 productive hours a day (no idle time during these 8 hours). There is a daily demand for at most 180 shirts. The demand for pants is unlimited. Each worker can participate only in one activity- the activity to which they are assigned. The table below gives the time requirements (in minutes) and profit per unit for the two garments.

	Amount (minutes) per operation			
	Cutting	Sewing	Packaging	Profit per unit($)
Shirts	40	40	20	10
Pants	20	100	20	8

a) Explain why a Linear Programming (LP) model would be suitable for this case study. [5 marks]

Solution:

Linear Programming (LP) is a mathematical technique for the analysis of optimum decisions subject to certain constraints in the form of linear inequalities. Mathematically speaking, it applies to those problems which require the solution of maximization or minimization problems subject to a system of linear inequalities stated in terms of certain variables.

If x and y, the two variables, are the function of z, the value of r is maximized when any movement from that point results in a decreased value of z. The value of z is minimized when even a small movement results in an increased value of z.

The term "linear" indicates that the function to be maximized is of degree one and the corresponding constraints are represented by a system of linear inequalities. The word "programming" means that the planning of activities in a manner that leads to some optimum results with limited resources. A programme is optimal if it maximizes or minimizes output, profits or cost of a firm.

<u>Linear Programming Formula:</u>

A linear programming problem will consist of decision variables, an objective function, constraints, and non-negative restrictions. The decision variables, x, and y, decide the output of the LP problem and represent the final solution. The objective function, Z, is the linear function that needs to be optimized (maximized or minimized) to get the solution. The constraints are the restrictions that are imposed on the decision variables to limit their value. The decision variables must always have a non-negative value which is given by the non-negative restrictions. The general formula of a linear programming problem is given below:

Objective Function: $Z = ax + by$

Constraints: $cx + dy \le e$, $fx + gy \le h$. The inequalities can also be "≥"

Non-negative restrictions: $x \ge 0$, $y \ge 0$

<u>Linear Programming for this case study</u>:

Linear programming is the best suitable method for this case study because this problem statement involves optimizing a linear objective function subject to linear constraints. The goal is to maximize the profit (a linear objective) while considering constraints such as the availability of workers, time requirements for each garment, and the demand for shirts. LP models are well-suited for problems with linear relationships between decision variables and constraints, making them a suitable tool for optimizing resource allocation in manufacturing processes. Here in this case study we need to convert the times for a feasible result.

b) Formulate a LP model to help the factory management to determine the optimal daily production schedule that maximises the profit while satisfying all constraints.
[10 marks]

Decision Variable:

- If x is number of shirts produced daily
- If y is number of pants produced daily

Objective Function:

Profit $Z = 10x + 8y$

Constraints:

- Cutting department: $40x + 20y \leq 20*8*60$
- Sewing department: $40x + 100y \leq 50*8*60$
- Packaging department: $20x + 20y \leq 14*8*60$
- Daily demand for Shirt: $x \leq 180$
- Non-Negative Constraints: $x \geq 0$ and $y \geq 0$

The problem is to find x and y such that $10x + 8y$ is maximised

c) Use the graphical method to find the optimal solution. Show the feasible region and the optimal solution on the graph. Annotate all lines on your graph. What is the optimal daily profit for the factory?

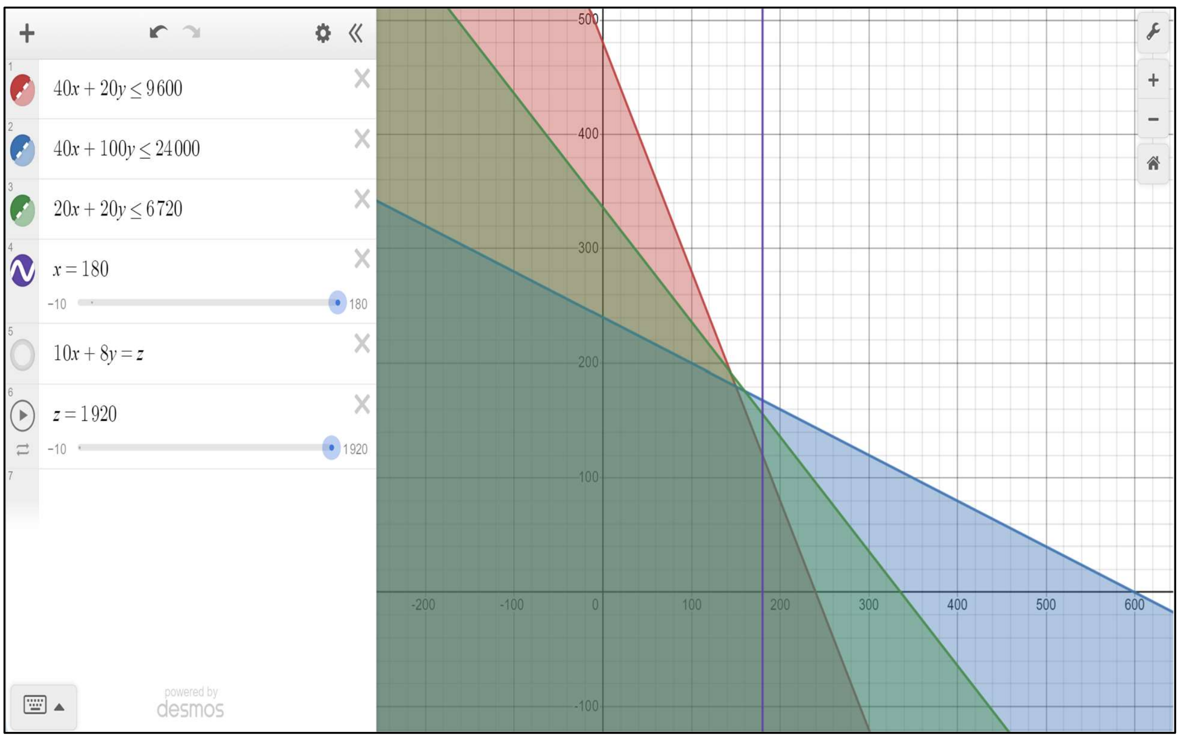


Fig 1 Graphical Representation

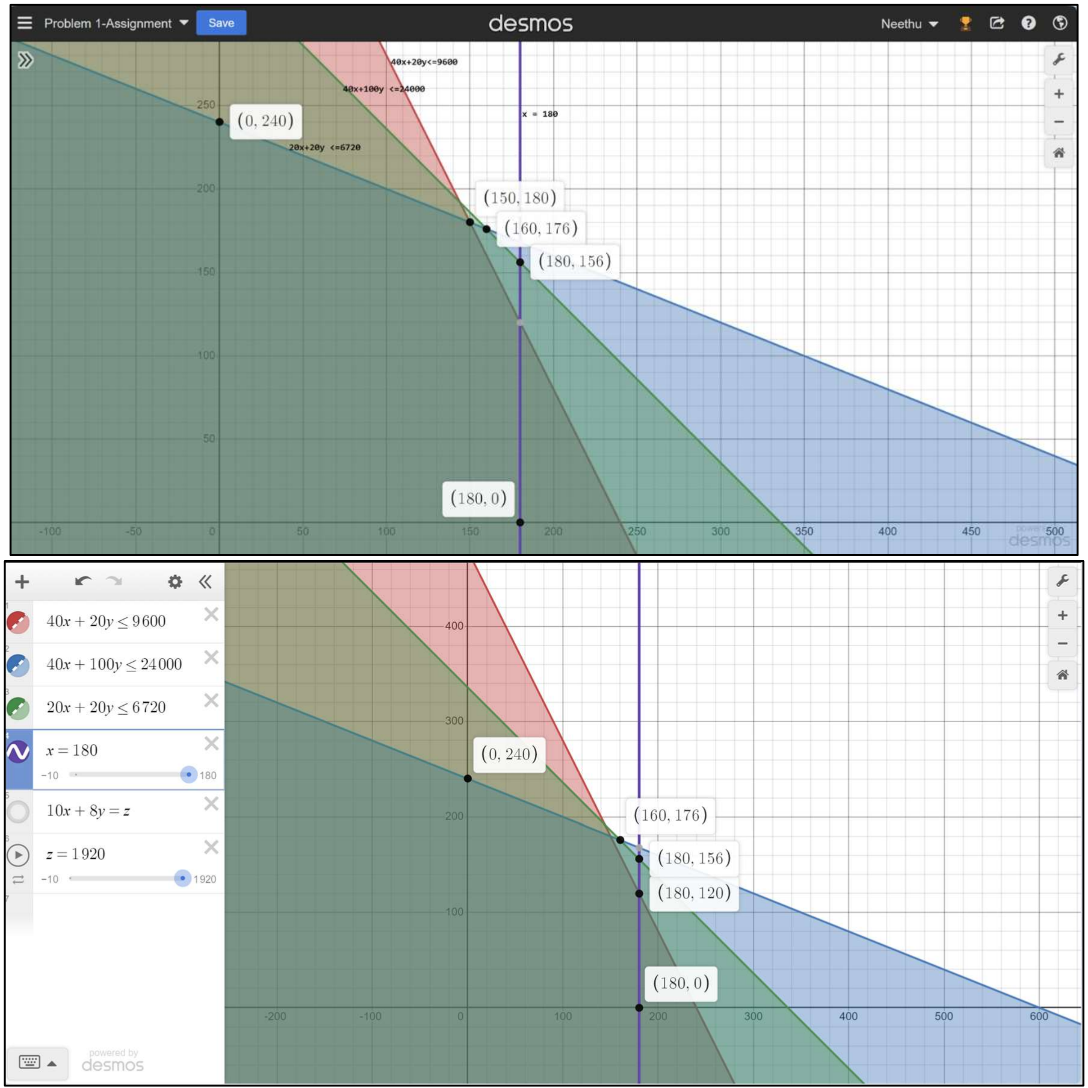


Fig 2: Feasible Region

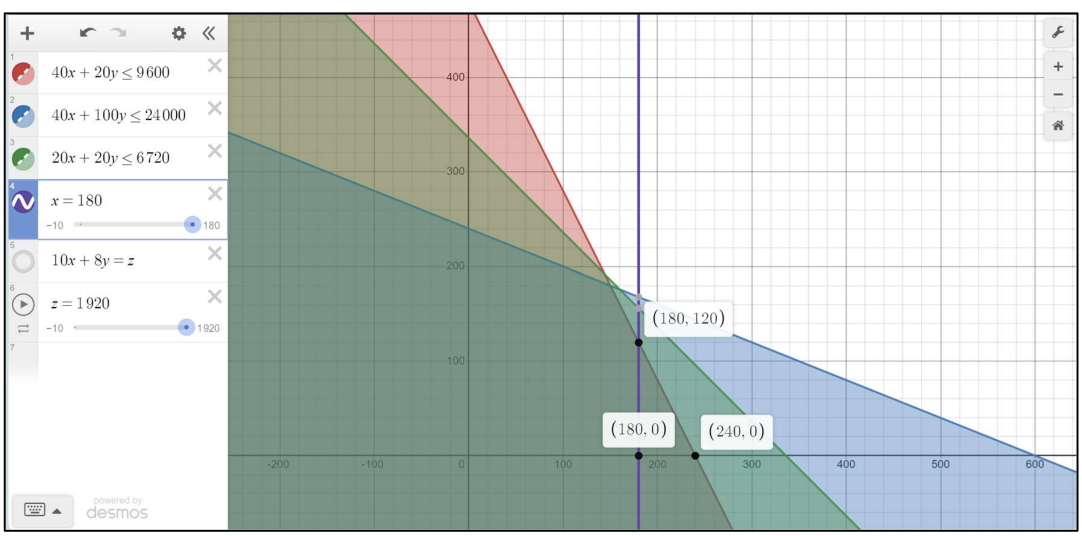


Fig 3 Optimal points

Substitute the coordinates in the equation:
10 x + 8 y = z

X	Y
0	0
240	0
180	0
180	120
150	180

10 (0) + 8 (0) = 0
10 (240) + 8 (0) = 2400
10 (180) + 8(0) = 1800
10 (180) + 8 (120) = 2760
10(150) + 8(180) = 2940

Since maximum optimization is our objective, then 2740 can be the maximum and 1800 can be the minimum

d) Find the range for the profit ($) per shirt (if any) that can be obtained without affecting the optimal point of part (c).
[5 marks]

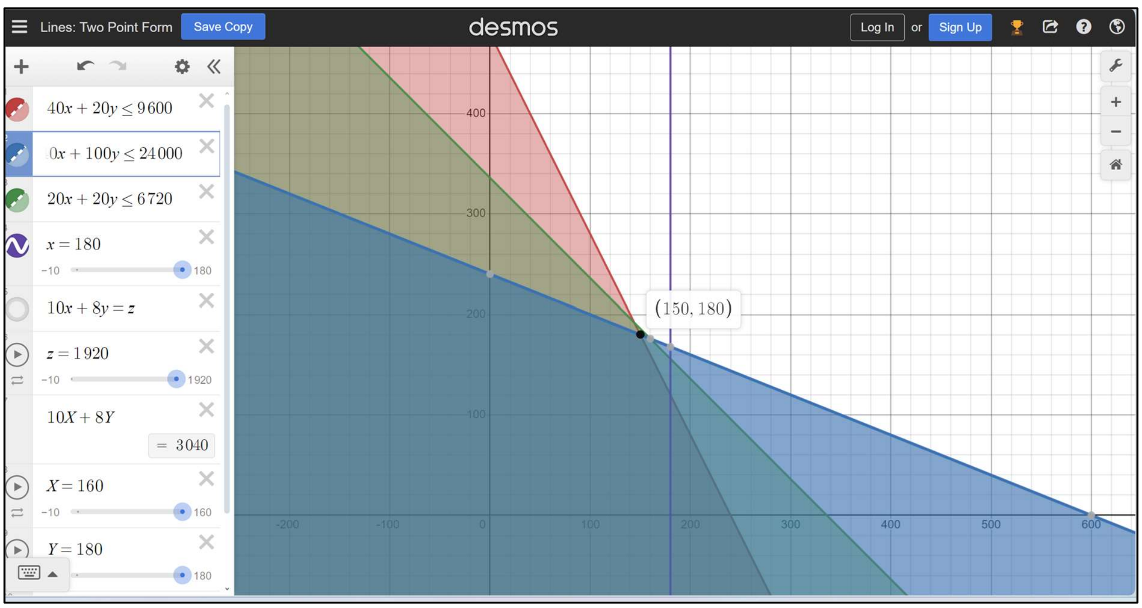


Fig 4 Range of Profit

We could find the range of profit at x = 150 and y =180, where
10x + 8y =z
10(150) + 8(180) = 2940

Therefore, Profit minimum can be 1800 at points x = 180 and y =0 and Profit maximum can be at 3040 where x = 150 and y =180

This shows that if 180 shirts are sold the minimum profit earned will be 1800. So minimum profit for one shirt can be $10 and the maximum profit for sale of 150 shirts and 180 pants is 2940. Hence the maximum profit for one shirt can be $20.

solve	Shirts	Pants			
Optimal points	150	180			
Cutting	40	20	9600	<=	9600
Sewing	40	100	24000	<=	24000
Packaging	20	20	6600	<=	6720
Demand Constrai	1	0	150	<=	180
Profit per unit($)	10	8	2940		

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
E6	Optimal points Shirts	150	0	10	6	6.8
F6	Optimal points Pants	180	0	8	17	3

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
G7	Cutting	9600	0.2125	9600	320	4800
G8	Sewing	24000	0.0375	24000	960	4800
G9	Packaging	6600	0	6720	1E+30	120
G10	Demand Constraint	150	0	180	1E+30	30

Problem 2

A factory makes three products called Bloom, Amber, and Leaf, from three materials containing Cotton, Wool and Nylon. The following table provides details on the sales price, production cost and purchase cost per ton of products and materials respectively.

	Sales price	Production cost		Purchase price
Bloom	$60	$5	Cotton	$40
Amber	$55	$4	Wool	$45
Leaf	$60	$5	Nylon	$30

The maximal demand (in tons) for each product, the minimum cotton and wool proportion in each product are as follows:

	Demand	min Cotton proportion	min Wool proportion
Bloom	4200	50%	40%
Amber	3200	60%	40%
Leaf	3500	50%	30%

a) Formulate an LP model for the factory that maximises the profit, while satisfying the demand and the cotton and wool proportion constraints.
[20 Marks]

Let's consider Cotton as C, Wool as W and Nylon as N

Let's consider Bloom as x1, Amber as x2, and Leaf as x3

Products Produced by a Factory:

- Bloom ► Cx1 +Wx1 +Nx1
- Amber ► Cx2 +Wx2 +Nx2
- Leaf ► Cx3 +Wx3 +Nx3

Materials used for production of each product in factory:

- Cotton► Cx1 +Cx2 +Cx3
- Wool ► Wx1+ Wx2+ Wx3
- Nylon ► Nx1+ Nx2+ Nx3

Daily revenue from products sales:

60 (xC1+xW1+ xN1) + 55 (xC2+xW2+ xN2) + 60 (xC3+xW3+ xN3)

Daily cost of purchasing each material:

40 (C1+C2+C3) + 45 (W1+W2+W3) + 30 (N1+N2+N3)

Daily cost of production of each product:

5 (xC1+xW1+ xN1) + 4 (xC2+xW2+ xN2) + 5 (xC3+xW3+ xN3)

Objective Function:

The objective is to maximize the profit. Profit is the difference between sales revenue and production cost. The profit function is expressed as given below:

Profit = Revenue – Purchasing Cost – advertising cost – production cost

Here, we consider Profit = Sales -Purchasing cost - Production cost

Optimal Profit = (60 -5) (xC1+xW1+ xN1) + (55-4) (xC2 + xW2+ xN2) + (60-5) (xC3 +xW3 + xN3) – 40(C1+C2+C3) – 45(W1+W2+W3) – 30 (N1+N2+N3)

i.e., 55 xC1 + 55 xW1 + 55xN1 + 51 x C2 + 51 xW2 + 51 xN2 + 55 XC3 + 55 XW3 + 55 XN3 -40 C1 - 40C2 - 40C3 – 45W1 – 45 W2 – 45W3 – 30N1 -30N2-30N3

This function is also written as Max 55 xC1 + 55 xW1 + 55xN1 + 51 x C2 + 51 xW2 + 51 xN2 + 55 XC3 + 55 XW3 + 55 XN3 -40 C1 - 40C2 - 40C3 – 45W1 – 45 W2 – 45W3 – 30N1 -30N2-30N3.

After simplifying the object function, we get:

Profit = 15 xC1+ 10xW1+ 25 xN1+ 11xC2+ 6xW2+ 21xN2+15xC3+ 10 xW3+ 25xN3

Demand constraints:

xC1 + xW1 + xN1 ≤ 4200

xC2 + xW2 + xN2 ≤ 3200

xC3 + xW3 + xN3 ≤ 3500

Material proportion constraints:

$$\frac{xc_1}{xc_1+xw_1+xN_1} \geq 0.5$$

$$\frac{xW1}{xc_1+xw_1+xN_1} \geq 0.4$$

$$\frac{xC2}{xc_2+xw2+xN2} \geq 0.6$$

$$\frac{xW2}{xc_2+xw2+xN2} \geq 0.4$$

$$\frac{xc_3}{xc_3+xw_3+xN_3} \geq 0.5$$

$$\frac{xW_1}{xc_3+xw_3+xN_3} \geq 0.3$$

Non-negative constraints:

xC1, xW1, xN1, xC2, xW2, xN2, xC3, xW3, xN3 ≥ 0

b) Solve the model using R/R Studio. Find the optimal profit and optimal values of the decision variables.

[20 Marks]

Screen shorts from R for reference:

```
```{r}
obj <- c(15, 10, 25, 11, 6, 21, 15, 10, 25)

# Create an LP model with 9 decision variables and 9 constraints
lp_model <- make.lp(9, 9)

set.objfn(lp_model, obj)

lp.control(lp_model, sense = "max")

# Add demand constraints
#XC_(1 )+XW_(1 )+XN_(1 )≤4200
add.constraint(lp_model, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 4200)  # Bloom

#XC_(2 )+XW_(2 )+XN_2≤3200
add.constraint(lp_model, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 3200)  # Amber
```
```

```
#XC_(3 )+XW_(3 )+XN_3≤3500
add.constraint(lp_model, c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=", 3500)  # Leaf

# Add proportion constraints - Cotton
#XC_(1 )-XW_(1 )-XN_(1 )≥ 0
add.constraint(lp_model, c(1, -1, -1, 0, 0, 0, 0, 0, 0), ">=", 0)  # Bloom

#0.4XC_(2 )- 0.6XW_2- 0.6XN_(2 )≥ 0
add.constraint(lp_model, c( 0, 0, 0,0.4,-0.6, -0.6, 0, 0, 0), ">=", 0)  # Amber

#XC_(3 )-XW_3-XN_3≥ 0
add.constraint(lp_model, c(0, 0, 0, 0, 0, 0, 1, -1, -1), ">=", 0)  # Leaf

# Add proportion constraints - Wool
#0.6XW_(1 )-0.4 XC_(1 )-0.4XN_1≥ 0
add.constraint(lp_model, c(-0.4, 0.6, -0.4, 0, 0, 0, 0, 0, 0), ">=", 0)  # Bloom

#0.6 XW_(2 ) -0.4 XC_(2 )-0.4XN_(2 ) ≥ 0
add.constraint(lp_model, c(0, 0, 0, -0.4, 0.6, -0.4, 0, 0, 0), ">=", 0)  # Amber

#0.7XW_3-0.3 XC_(3 )-0.3XN_(3 )  ≥ 0
add.constraint(lp_model, c(0, 0, 0, 0, 0, 0, -0.3, 0.7, -0.3), ">=", 0)  # Leaf

# Add non-negativity constraints
set.bounds(lp_model, lower = rep(0, 9))

# Solve the linear programming problem
lp_solution <- solve(lp_model)

# Print the results
print(lp_solution)
```

Results from R

- Given the constraints, the optimal profit for products at $.141850
- > solution
- [1] 2100 1680 420 1920 1280   0 1750 1050 700
- Optimal values for decision variables-

```
x_C1=2100,          x_W1=1680,         x_N1=420
x_C2=1920 ,x_W2=1280,x_N2=0
x_C3=1750,x_W3=1050,x_N3=700
```

Based on Materials used:

```
Cotton:  5770
Wool:  4010
Nylon:  1120
```

Problem 3

Two construction companies, Giant and Sky, bid for the right to build in a field. The possible bids are $ 10 Million, $ 20 Million, $ 30 Million, $ 35 Million and $ 40 Million. The winner is the company with the higher bid.
The two companies decide that in the case of a tie (equal bids), Giant is the winner and will get the field.
Giant has ordered a survey and, based on the report from the survey, concludes that getting the field for more than $ 35 Million is as bad as not getting it (assume loss), except in case of a tie (assume win). Sky is not aware of this survey.

Solution:

(a) State reasons why/how this game can be described as a two-players-zero-sum game [5 Marks]

A two-player game is called a zero-sum game if the sum of the payoffs to each player is constant for all possible outcomes of the game. More specifically, the terms (or coordinates) in each payoff vector must add up to the same value for each payoff vector. Such games are sometimes called constant-sum games instead.

Game theory provides a mathematical framework for analysing the decision-making processes and strategies of adversaries (or players) in different types of competitive situations. The simplest type of competitive situations is two-person, zero-sum games. These games involve only two players; they are called zero-sum games because one player wins whatever the other player loses.

Basic Concepts of Two-player-zero-sum game: -

This game of odds and evens illustrates important concepts of simple games.

- A two-person game is characterized by the strategies of each player and the payoff matrix.
- The payoff matrix shows the gain (positive or negative) for player 1 that would result from each combination of strategies for the two players. Note that the matrix for player 2 is the negative of the matrix for player 1 in a zero-sum game.
- The entries in the payoff matrix can be in any units as long as they represent the utility (or value) to the player.
- There are two key assumptions about the behavior of the players. The first is that both players are rational. The second is that both players are greedy meaning that they choose their strategies in their own interest (to promote their own wealth).

Giant and Sky are the only two players in the game, so it is a two-player affair. Since the total gain for one player equals the total loss for the other, the game is zero-sum. In this instance, the bid amount is the payoff, and only one of the companies will be awarded the field. Giant loses that much if Sky wins, and vice versa. In the system, the total gain and total loss add up to zero.

(b) Considering all possible combinations of bids, formulate the payoff matrix for the game. [5 Marks]

A payoff matrix is a way to express the result of players' choices in a game. A payoff matrix does not express the structure of a game, such as if players take turns taking actions or a player has to make a choice without knowing what choice the other will make.

In game theory, a payoff matrix represents the outcomes for each player based on different combinations of strategies. The entries in the matrix represent the payoffs to the row player (Giant) and the column player (Sky).

Given the information you provided, let's use 1 to represent a win and -1 to represent a loss for Giant. The payoff matrix will be symmetric since the game is zero-sum.

Giant /Sky	10	20	30	35	40
10	0	-10	-20	-25	-30
20	10	0	-10	-15	-20
30	20	10	0	-5	-15
35	25	15	5	0	-5
40	30	20	10	5	40

Payoff Matrix based on L=-1 and U=1, the value of game is somewhere between -1 and 1.

Giant /Sky	10	20	30	35	40
10	1	-1	-1	-1	1
20	1	1	-1	-1	1
30	1	1	1	-1	1
35	1	1	1	1	1
40	-1	-1	-1	-1	-1

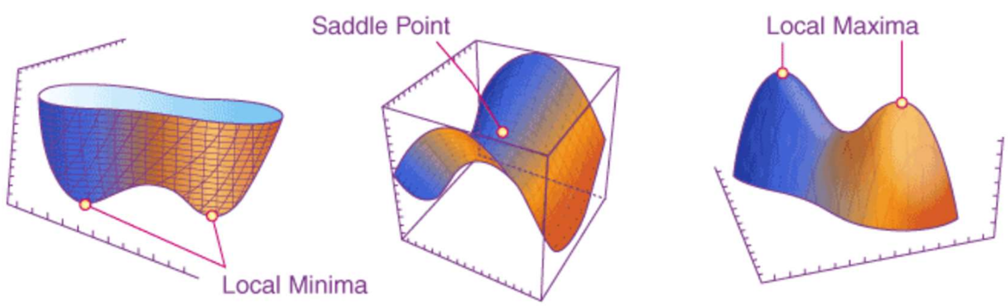
- The positive entries (1) represent a win for Giant.
- The negative entries (-1) represent a loss for Giant.
- The diagonal entries represent a tie, and in this case, Giant is considered the winner.
- The payoff to Giant is the opposite of the payoff to Sky. If Giant wins, Sky loses, and vice versa. If there is a tie, Giant is considered the winner.

(c) Explain what is a saddle point. Verify: does the game have a saddle point?
[5 Marks]

Saddle points of a multivariable function are those points in its domain where the tangent is parallel to the horizontal axis, but this point tends to be neither a local maximum nor a local minimum.

For a two-variable function f(x, y), its saddle point is defined as

If $z = f(x, y)$, then the point (x, y, z) is said to be a saddle point if both the partial derivatives $f_x(x, y)$ and $f_y(x, y)$ vanishes, but f does not attain any extremum values (maxima or minima) at (x, y).



Second Derivative Test for Saddle Points

Many times, saddle points are not identifiable with the general second derivative test which is used to find the concavity or convexity of a function at any point. To find the saddle points within the domain of a function we require mixed partial derivatives.

Let f(x, y) is a function in two variables for which the first and second-order partial derivatives are continuous on some disk that contains the point (a, b). If $f_x(a, b) = 0$ and $f_y(a, b) = 0$, we define $D = f_{xx}(a, b)\, f_{yy}(a, b) - [f_{xy}(a, b)]^2$ then:

- If $D > 0$ and $f_{xx}(a, b) > 0$, f has a local minimum at (a, b)
- If $D > 0$ and $f_{xx}(a, b) < 0$, has a local maximum at (a, b)
- If $D < 0$, f has a saddle point (a, b)
- If $D = 0$, the test fails.

Method of Finding of Saddle Points

Let f(x, y) be a function of two variables whose first and second-order partial derivatives exist and are continuous on a domain containing the point (a, b). Using the following steps apply the second derivative test to find the extreme points and saddle points.

- Determine the critical points (a, b) such that the partial derivatives fx(a, b) = fy(a, b) = 0. Discard those points for which any of the partial derivatives does not exist.
- Calculate the discriminant D = fxx(a, b) fyy (a, b) – [fxy(a, b)]2 for each critical point.
- Check for the cases:
  - If D > 0 and fxx(a, b) > 0, f has a local minimum at (a, b)
  - If D > 0 and fxx(a, b) < 0, has a local maximum at (a, b)
  - If D < 0, f has a saddle point (a, b)
  - If D = 0, the test fails.

Verify the Saddle point of the game:

We apply Min – Max principle to analyse the game

Giant /Sky	10	20	30	35	40	B_j (Row Minimum)	
10	1	-1	-1	-1	1	-1	
20	1	1	-1	-1	1	-1	
30	1	1	1	-1	1	-1	L = 1
35	1	1	1	1	1	1	
40	-1	-1	-1	-1	-1	-1	
A_i (Column Max)	1	1	1	1	1		
U = 1							

The game value range is between -1 and 1. There for pure strategy exist.

Let L the lower value of the game and call U the upper value of the game.

- If U= L , we call this the value of the game, and the optimal payoff for both players can be achieved by a pure strategy.
- If L < U, then a pure strategy will not result in an equilibrium and players must resort to mixed strategies.

(d) Construct a linear programming model for Company Sky in this game. [5 Marks]

Linear Programming Model for Player 2 (Sky)

Pay-off Matrix for Giant as Player 1 and Sky as Player 2

Giant (1) / Sky(2)	10	20	30	35	40
10	1	-1	-1	-1	-1
20	1	1	-1	-1	-1
30	1	1	1	-1	-1
35	1	1	1	1	-1
40	1	1	1	1	1

Here Player 1 (column wise) is Giant and player 2 (row wise) is Sky.

Using Linear Programming for Player I (Giant)

Suppose Player I chooses the mixed strategy LaTeX: (x1, x2, x3, x4, x5)

- If Player II chooses strategy 1, the expected payoff is (x1+x2+x3+ x4+x5)
- If Player II chooses strategy 2 , the expected payoff is (-x1+x2+x3+x4+x5)
- If Player II chooses strategy 3, the expected payoff is (-x1-x2+x3+x4+x5)
- If Player II chooses strategy 4, the expected payoff is (-x1-x2-x3+x4+x5)
- If Player II chooses strategy 5, the expected payoff is (-x1- x2-x3-x4+x5)

In mathematics, we can write the following LP optimization to find the minimum of three numbers a, b,c,d,e

$$\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad & v \leq a \Longrightarrow \; v - a \leq 0 \\
& v \leq b \Longrightarrow \; v - b \leq 0 \\
& v \leq c \Longrightarrow \; v - c \leq 0
\end{aligned}$$

Hence, Player I's game can be written as follows:

Max z = v

s.t. v-(x1+ x2+ x3+ x4-x5)≤ 0

v-(-x1+x2+x3+x4-x5)≤ 0

v - (-x1-x2+x3+x4-x5)≤ 0

v-(x1+x2+x3+x4+x5≤0

v-(-x1- x2-x3-x4+x5) ≤0

x1+x2+x3+x4+x5 = 1

xi ≥ 0, ∀i = 1, 2, 3,4,5

v u.r.s. (means – unrestricted sign)

Using Linear Programming for Player II (Sky):

Suppose Player II chooses the mixed strategy (y1, y2, y3, y4, y5)

- If Player I chooses strategy 1, the expected reward is (y1-y2-y3-y4-y5)
- If Player I chooses strategy 2, the expected reward is (y1+y2-y3-y4-y5)
- If Player I chooses strategy 3, the expected reward is (y1+ y2+ y3 – y4-y5)
- If Player I chooses strategy 4, the expected reward is (y1+y2+y3+y4-y5)
- If Player I chooses strategy 5, the expected reward is (y1 +y2 +y3 +y4+y5)

$$
\begin{aligned}
\text{min} \quad & v \\
\text{s.t.} \quad & v \geq a \Longrightarrow \quad v - a \geq 0 \\
& v \geq b \Longrightarrow \quad v - b \geq 0 \\
& v \geq c \Longrightarrow \quad v - c \geq 0
\end{aligned}
$$

Hence, Player II's game can be written as follows:

Min w=v

s.t. v-(y1-y2-y3-y4-y5) ≥ 0

v-(y1+y2-y3-y4-y5) ≥0

v-(y1+ y2+ y3 – y4-y5) ≥0

v-(y1+y2+y3+y4-y5) ≥ 0

v-(y1 +y2 +y3 +y4+y5≥0

y1 +y2 +y3 + y4 + y5 =1

yi ≥ 0, ∀i = 1, 2,3,4,5

If Sky is considered as Player 1 and Giant as player 2, then the pay off matrix will be as given below:

		GIANT – PLAYER 2					
SKY – PLAYER 1	Sky \ Giant	10	20	30	35	40	B_j (Row Minimum)
	10	1	-1	-1	-1	-1	-1
	20	1	1	-1	-1	-1	-1
	30	1	1	1	-1	-1	-1
	35	1	1	1	1	-1	-1
	40	1	1	1	1	1	1
	Ai (Column Max)	1	1	1	1	1	U =1 L=1

Using Linear Programming for Player I (Sky)

Suppose Player I chooses the mixed strategy LaTeX: (x1, x2, x3, x4, x5)

- If Player II chooses strategy 1, the expected payoff is (x1 +x2 +x3 +x4+x5)

- If Player II chooses strategy 2 , the expected payoff is $(-x1+x2+x3+x4+x5)$
- If Player II chooses strategy 3, the expected payoff is $(-x1-x2+x3+x4+x5)$
- If Player II chooses strategy 4, the expected payoff is $(-x1-x2-x3+x4+x5)$
- If Player II chooses strategy 5, the expected payoff is $(-x1-x2-x3-x4+x5)$

Hence, Player I's game can be written as follows:

Max $z = v$

s.t. $v-(x1 +x2 +x3 +x4+x5) \le 0$

$v-(-x1+x2+x3+x4+x5) \le 0$

$v - (-x1-x2+x3+x4+x5) \le 0$

$v-(-x1-x2-x3+x4+x5)) \le 0$

$v-(-x1-x2-x3-x4+x5) \le 0$

$x1+x2+x3+x4+x5 = 1$

$xi \ge 0, \forall i = 1, 2, 3,4,5$

v u.r.s. (means – unrestricted sign)

<u>Using Linear Programming for Player II (Giant):</u>

Suppose Player II chooses the mixed strategy (y1, y2, y3, y4, y5)

- If Player I chooses strategy 1, the expected reward is $(y1 -y2-y3-y4-y5)$
- If Player I chooses strategy 2, the expected reward is $(y1+y2-y3-y4-y5)$
- If Player I chooses strategy 3, the expected reward is $(y1+y2-+y3 – y4-y5)$
- If Player I chooses strategy 4, the expected reward is $(y1+y2+y3+y4-y5)$
- If Player I chooses strategy 5, the expected reward is $(y1 +y2 +y3 +y4+y5)$

Hence, Player II's game can be written as follows:

Min $w=v$

s.t. $v-(y1 -y2-y3-y4-y5) \ge 0$

$v-(y1+y2-y3-y4-y5) \ge 0$

$v-(y1+y2-+y3 – y4-y5) \ge 0$

$v-(y1+y2+y3+y4-y5) \ge 0$

$v-(y1 +y2 +y3 +y4+y5) \ge 0$

$y1 +y2 +y3 + y4 + y5 =1$

$yi \ge 0, \forall i = 1, 2,3,4,5$

R Code for Sky as Player II

```
skygame<-make.lp(0,6) #initializing 0 constraints and 6 decision variables

lp.control(skygame, sense="maximize") #setting control parameters
set.objfn(skygame, c(0,0,0,0,0,1)) #y1,y2,y3,y4,y5,v
#adding constraints
add.constraint(skygame, c(1, -1, -1, -1, -1, 1), "<=", 0)#(y1-y2-y3-y4-y5)
add.constraint(skygame, c(1, 1, -1, -1, -1, 1), "<=", 0) #(y1+y2-y3-y4-y5)
add.constraint(skygame, c(1, 1, 1, -1, -1, 1), "<=", 0) #(y1+ y2+ y3 – y4-y5)
add.constraint(skygame, c(1, 1, 1, 1, -1, 1), "<=", 0) # (y1+y2+y3+y4-y5
add.constraint(skygame, c(1, 1, 1, 1, 1, 1), "<=", 0)#(y1 +y2 +y3 +y4+y5)
add.constraint(skygame, c(1 , 1, 1, 1, 1, 1), "=", 1)

 #updating the lower and upper bounds of decision variables

set.bounds(skygame, lower = c(0, 0, 0, 0, 0, -Inf))

RowNames <- c("C1", "C2", "C3","C4", "C5","C6")

ColNames <- c("Y1", "Y2", "Y3", "Y4", "Y5", "v")

dimnames(skygame) <- list(RowNames, ColNames)

skygame #Display the LP model
solve(skygame)

valueofgame<-get.objective(skygame)
valueofgame  #value of the game

optimal_solution<-get.variables(skygame)
optimal_solution #optimal strategies for Sky

constraints1<-get.constraints(skygame)
constraints1  #values of the constraints
```

```
Model name:
            Y1    Y2    Y3    Y4    Y5     v
Maximize     0     0     0     0     0     1
C1           1    -1    -1    -1    -1     1  <=  0
C2           1     1    -1    -1    -1     1  <=  0
C3           1     1     1    -1    -1     1  <=  0
C4           1     1     1     1    -1     1  <=  0
C5           1     1     1     1     1     1  <=  0
C6           1     1     1     1     1     1   =  1
Kind       Std   Std   Std   Std   Std   Std
Type      Real  Real  Real  Real  Real  Real
Upper      Inf   Inf   Inf   Inf   Inf   Inf
Lower        0     0     0     0     0  -Inf
[1] 2
[1] -1e+30
[1] 1.379617e-306 5.562820e-308 4.450351e-308 1.246115e-306 7.788225e-308 1.056937e-307
[1] 1.112615e-306 4.450576e-308 1.691192e-306 9.346084e-307 9.456836e-308 1.112589e-306
```

(e) Produce an appropriate code to solve the linear programming model in part (d).

R Code for Sky as Player 1

```
# Sky as Player 1

```{r}
skygame<-make.lp(0,6) #initializing 0 constraints and 6 decision variables

lp.control(skygame, sense="maximize") #setting control parameters
set.objfn(skygame, c(0,0,0,0,0,1)) #x1,x2,x3,x4,x5,v
#adding constraints
add.constraint(skygame, c(1, 1, 1, -1, 1, 1), "<=", 0) # (x1 +x2 +x3 +x4+x5),If Player II chooses
strategy 1
add.constraint(skygame, c(-1, 1, 1, 1, 1, 1), "<=", 0) #  (-x1+x2+x3+x4+x5),If Player II chooses
strategy 2
add.constraint(skygame, c(-1,-1, 1, -1, 1, 1), "<=", 0) # (-x1-x2+x3+x4+x5),If Player II chooses
strategy 3
add.constraint(skygame, c(-1, -1, -1, 1, 1, 1), "<=", 0) # (-x1-x2-x3+x4+x5),If Player II chooses
strategy 4
add.constraint(skygame, c(-1, -1, -1, -1, 1, 1), "<=", 0) # (-x1-x2-x3-x4+x5),If Player II chooses
strategy 5
add.constraint(skygame, c(1 , 1, 1, 1, 1, 1), "=", 1)

 #updating the lower and upper bounds of decision variables

set.bounds(skygame, lower = c(0, 0, 0, 0, 0, -Inf))

RowNames <- c("C1", "C2", "C3","C4", "C5","C6")

ColNames <- c("x1", "x2", "x3", "x4", "x5", "v")

dimnames(skygame) <- list(RowNames, ColNames)

skygame #Display the LP model
solve(skygame)
```

```
Model name:
              x1    x2    x3    x4    x5     v
Maximize       0     0     0     0     0     1
C1             1     1     1    -1     1     1  <=  0
C2            -1     1     1     1     1     1  <=  0
C3            -1    -1     1    -1     1     1  <=  0
C4            -1    -1    -1     1     1     1  <=  0
C5            -1    -1    -1    -1     1     1  <=  0
C6             1     1     1     1     1     1   =  1
Kind         Std   Std   Std   Std   Std   Std
Type        Real  Real  Real  Real  Real  Real
Upper        Inf   Inf   Inf   Inf   Inf   Inf
Lower          0     0     0     0     0  -Inf
[1] 0
[1] 0
[1] 0.5 0.0 0.0 0.5 0.0 0.0
[1]  0  0 -1  0 -1  1
```
```

(f) Solve the game for Sky using the linear programming model and the code you constructed in parts (d) and (e). Interpret your solution.

R Codes:

Sky as Player 1

```
skygame<-make.lp(0,6) #initializing 0 constraints and 6 decision variables

lp.control(skygame, sense="maximize") #setting control parameters
set.objfn(skygame, c(0,0,0,0,0,1)) #x1,x2,x3,x4,x5,v
#adding constraints
add.constraint(skygame, c(1, 1, 1, -1, 1, 1), "<=", 0) # (x1 +x2 +x3 +x4+x5),If Player II chooses strategy 1
add.constraint(skygame, c(-1, 1, 1, 1, 1, 1), "<=", 0) #  (-x1+x2+x3+x4+x5),If Player II chooses strategy 2
add.constraint(skygame, c(-1,-1, 1, -1, 1, 1), "<=", 0) # (-x1-x2+x3+x4+x5),If Player II chooses strategy 3
add.constraint(skygame, c(-1, -1, -1, 1, 1, 1), "<=", 0) # (-x1-x2-x3+x4+x5),If Player II chooses strategy 4
add.constraint(skygame, c(-1, -1, -1, -1, 1, 1), "<=", 0) # (-x1-x2-x3-x4+x5),If Player II chooses strategy 5
add.constraint(skygame, c(1 , 1, 1, 1, 1, 1), "=", 1)

 #updating the lower and upper bounds of decision variables

set.bounds(skygame, lower = c(0, 0, 0, 0, 0, -Inf))

RowNames <- c("C1", "C2", "C3","C4", "C5","C6")

ColNames <- c("x1", "x2", "x3", "x4", "x5", "v")

dimnames(skygame) <- list(RowNames, ColNames)

skygame #Display the LP model
solve(skygame)

valueofgame<-get.objective(skygame)
valueofgame  #value of the game
```

```
Model name:
             x1    x2    x3    x4    x5     v
Maximize      0     0     0     0     0     1
C1            1     1     1    -1     1     1  <=  0
C2           -1     1     1     1     1     1  <=  0
C3           -1    -1     1    -1     1     1  <=  0
C4           -1    -1    -1     1     1     1  <=  0
C5           -1    -1    -1    -1     1     1  <=  0
C6            1     1     1     1     1     1   =  1
Kind        Std   Std   Std   Std   Std   Std
Type       Real  Real  Real  Real  Real  Real
Upper       Inf   Inf   Inf   Inf   Inf   Inf
Lower         0     0     0     0     0  -Inf
[1] 0
[1] 0
[1] 0.5 0.0 0.0 0.5 0.0 0.0
[1]   0   0 -1   0 -1   1
```

Giant As Player 1

```
```{r}
skygame<-make.lp(0,6) #initializing 0 constraints and 6 decision variables

lp.control(skygame, sense="minimize") #setting control parameters
set.objfn(skygame, c(0,0,0,0,0,1)) #x1,x2,x3,x4,x5,v
#adding constraints
add.constraint(skygame, c(1, 1, 1, 1, 1, 1), "<=", 0) # (x1+x2+x3+ x4+x5),If Player II chooses strategy 1
add.constraint(skygame, c(-1, 1, 1, 1, 1, 1), "<=", 0) #   (-x1+x2+x3+x4+x5),If Player II chooses strategy 2
add.constraint(skygame, c(-1,-1, 1, 1, 1, 1), "<=", 0) #  (-x1-x2+x3+x4+x5),If Player II chooses strategy 3
add.constraint(skygame, c(-1, -1, -1, -1, 1, 1), "<=", 0) #  (-x1-x2-x3+x4+x5),If Player II chooses strategy 4
add.constraint(skygame, c(-1, -1, -1, -1, 1, 1), "<=", 0) #  (-x1- x2-x3-x4+x5),If Player II chooses strategy 5
add.constraint(skygame, c(1 , 1, 1, 1, 1, 1), "=", 1)

 #updating the lower and upper bounds of decision variables

set.bounds(skygame, lower = c(0, 0, 0, 0, 0, -Inf))

RowNames <- c("C1", "C2", "C3","C4", "C5","C6")

ColNames <- c("x1", "x2", "x3", "x4", "x5", "v")

dimnames(skygame) <- list(RowNames, ColNames)

skygame #Display the LP model
solve(skygame)

valueofgame<-get.objective(skygame)
valueofgame  #value of the game
```

```
Model name:
              x1    x2    x3    x4    x5     v
Minimize       0     0     0     0     0     1
C1             1     1     1     1     1     1  <=  0
C2            -1     1     1     1     1     1  <=  0
C3            -1    -1     1     1     1     1  <=  0
C4            -1    -1    -1    -1     1     1  <=  0
C5            -1    -1    -1    -1     1     1  <=  0
C6             1     1     1     1     1     1   =  1
Kind         Std   Std   Std   Std   Std   Std
Type        Real  Real  Real  Real  Real  Real
Upper        Inf   Inf   Inf   Inf   Inf   Inf
Lower          0     0     0     0     0  -Inf
[1] 2
[1] 1e+30
[1] 0 0 0 0 0 0
[1] 0 0 0 0 0 0
```

Interpretation:

The optimal solution (lp_result$solution) will provide the probabilities for each bid that maximize Company Sky's expected profit. The optimal expected profit is given by is in form of payoff matrix as 0 0 0 1 0 1
```

Interpret the results by examining the optimal probabilities. These probabilities indicate the optimal strategy for Company Sky, specifying the likelihood of choosing each bid. Additionally, the optimal expected profit represents the maximum expected profit that Company Sky can achieve using this strategy.

Keep in mind that this analysis assumes that Company Sky uses a mixed strategy, randomizing among its bids based on the probabilities found in the solution. If you have specific values for the probabilities or if you need further assistance, feel free to provide additional information.

This means that the optimal strategy for Sky is to bid $10 Million or $20 Million with equal probability (0.5 each), and never bid $30 Million, $35 Million, or $40 Million. The expected payoff for Sky is $10 Million. This is also the value of the game, which means that Sky cannot expect to do better than this, regardless of what Giant does.

References

1. (n.d.). Retrieved from https://www.cuemath.com/algebra/linear-programming/
2. (n.d.). Retrieved from https://www.geeksforgeeks.org: https://www.geeksforgeeks.org/saddle-point-matrix/
3. (n.d.). Retrieved from https://calculator-online.net/saddle-point-calculator/
4. (n.d.). Retrieved from Atozmaths: https://cbom.atozmath.com/CBOM/GameTheory.aspx?q=matrix&q1=0%2C10%2C20%2C15%2C30%3B-10%2C0%2C10%2C15%2C20%3B-20%2C-10%2C0%2C5%2C10%3B-25%2C-15%2C-5%2C0%2C5%3B-30%2C-20%2C-10%2C5%2C0%6010%2C20%2C30%2C35%2C40%6010%2C20%2C30%2C35%2C40%60matrix%60true&dm=D&dp
5. (n.d.). Retrieved from https://home.ubalt.edu/ntsbarsh/business-stat/otherapplets/Game.htm
6. (2023, 12). Retrieved from https://math.libretexts.org/: https://math.libretexts.org/Bookshelves/Calculus/Map%3A_University_Calculus_(Hass_et_al)/13%3A_Partial_Derivatives/13.7%3A_Extreme_Values_and_Saddle_Points
7. (2023, 12). Retrieved from https://mathworld.wolfram.com: https://mathworld.wolfram.com/SaddlePoint.html
8. (2023, 12). Retrieved from https://byjus.com: https://byjus.com/maths/saddle-points/
9. (2023, 12). Retrieved from https://www.desmos.com/matrix
10. (2023). Retrieved from https://video.deakin.edu.au/media/t/1_oi84ad51: https://video.deakin.edu.au/media/t/1_oi84ad51
11. Arsham, P. H. (n.d.). *Introduction to Game Theory:Wining Business in A Competitive Environment.*
12. Arsham, P. H. (n.d.). *Introduction to Game Theory:Wining Business in A Competitive Environment.* Retrieved from https://home.ubalt.edu/ntsbarsh/business-stat/opre/partVI.htm
13. Giacomo Bonanno, U. o. (n.d.). *Game theory: Parts I and II - with 88 solved exercises. An open access textbook.* Retrieved from 2015 Giacomo Bonanno: https://www.econstor.eu/bitstream/10419/173660/1/847845893.pdf
14. *https://cbom.atozmath.com/Menu/CBomMenu.aspx*. (n.d.). Retrieved from https://cbom.atozmath.com/Menu/CBomMenu.aspx
15. https://olympus.mygreatlearning.com/courses/90778/pages/6-dot-9-solving-2-person-zero-sum-games-using-lp?module_item_id=3749082. (2023, 12). Retrieved from https://olympus.mygreatlearning.com/courses/90778/pages/6-dot-9-solving-2-person-zero-sum-games-using-lp?module_item_id=3749082

16. *https://plus.maths.org/content/game-theory-and-cuban-missile-crisis*. (n.d.). Retrieved from https://plus.maths.org/content/game-theory-and-cuban-missile-crisis
17. *https://video.deakin.edu.au/media/t/1_oi84ad51*. (n.d.). Retrieved from https://video.deakin.edu.au/media/t/1_oi84ad51
18. *https://video.deakin.edu.au/media/t/1_qfpruapw*. (n.d.). Retrieved from https://video.deakin.edu.au/media/t/1_qfpruapw
19. *https://www.chegg.com*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/2-factory-makes-three-products-called-bloom-amber-leaf-three-materials-containing-cotton-w-q107822173
20. *https://www.chegg.com*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/2-factory-makes-three-products-called-bloom-amber-leaf-three-materials-containing-cotton-w-q107822173
21. *https://www.chegg.com/homework-help/questions-and-answers/3-two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million--q92187971*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/3-two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million--q92187971
22. *https://www.chegg.com/homework-help/questions-and-answers/factory-makes-three-products-called-spring-autumn-winter-three-materials-containing-cotton-q51458861*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/factory-makes-three-products-called-spring-autumn-winter-three-materials-containing-cotton-q51458861
23. *https://www.chegg.com/homework-help/questions-and-answers/two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million-35-q107104660*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million-35-q107104660
24. *https://www.chegg.com/homework-help/questions-and-answers/two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million-35-q91775222*. (n.d.). Retrieved from https://www.chegg.com/homework-help/questions-and-answers/two-mining-companies-red-blue-bid-right-drill-field-possible-bids-15-million-25-million-35-q91775222
25. *https://www.desmos.com*. (n.d.). Retrieved from https://www.desmos.com/calculator/7ukbv8ec9r
26. *Payoff Matrices*. (2023, 12). Retrieved from https://www.math.uci.edu/~mathcircle/materials/2014.06.03MSMC.pdf
27. *www.economicsdiscussion.net*. (2023, 12). Retrieved from www.economicsdiscussion.net: https://www.economicsdiscussion.net/linear-programming/linear-programming-lp-meaning-and-limitations/18785