SIG718 Real World Analytics

Assessment Task 3

Linear Programming Models

Madhavi Joshi
Student Id # 224234405
Deakin University

Solution 1

a)

A Linear Programming (LP) model is suitable for this case study because it involves allocating limited resources to maximize a linear objective (profit) while satisfying certain constraints. The constraints involve the availability of workers in different departments, daily demand for shirts, and the fact that each worker can participate in only one activity. All these constraints can be expressed as linear inequalities. There are only two decision variables (shirts and pants) involved. Additionally, the profit from producing each unit of shirts and pants is also linear with respect to the production quantities. The objective function as well as all constraint functions can be plotted along linear coordinate planes. Therefore, LP is a suitable approach to find the optimal production schedule.

b)

Let x & y be the number of shirts and pants produced per day respectively.

Objective Function:

- Objective is to maximize profit (Profit per unit: \$10 for shirts, \$8 for pants):
  Maximize: $10x + 8y$

Production constraints:

- Cutting Department Constraint:
  $40x + 20y \leq 20 \times 8 \times 60$ (20 workers × 8 hours × 60 minutes)
  ➡ $2x + y \leq 480$
- Sewing Department Constraint:
  $40x + 100y \leq 50 \times 8 \times 60$ (50 workers × 8 hours × 60 minutes)
  ➡ $2x + 5y \leq 1200$
- Packaging Department Constraint:
  $20x + 20y \leq 14 \times 8 \times 60$ (14 workers × 8 hours × 60 minutes)
  ➡ $x + y \leq 336$

Demand Constraint:

- $x \leq 180$ (Daily demand for shirts)

Non-negativity constraint:

- $x \geq 0$, $y \geq 0$ (x and y should be non-negative)

Factory Linear Programing Model

max $z = 10x + 8y$

s.t. $2x + y \leq 480$

$2x + 5y \leq 1200$

$x + y \leq 336$

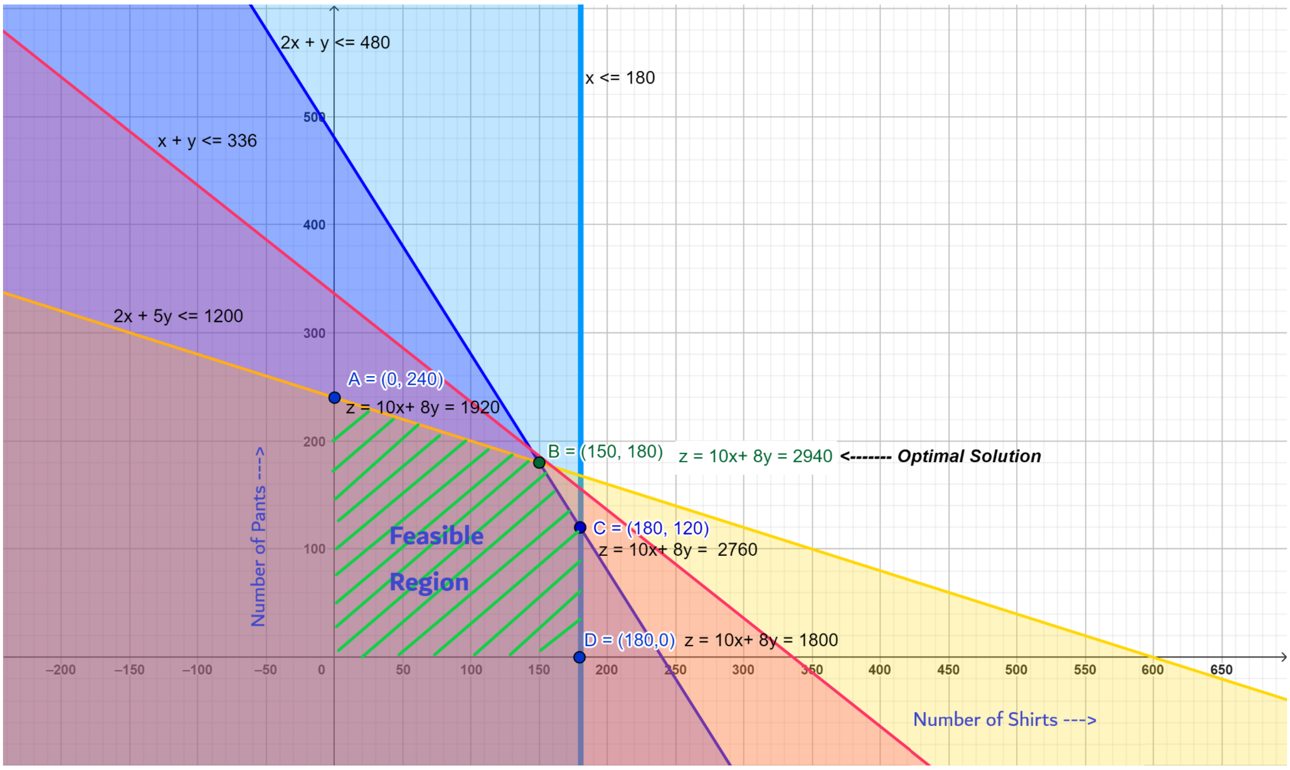
$x \leq 180$

$x \geq 0$, $y \geq 0$

c)

Plotting the 4 constraint lines, a feasible region is obtained and the vertices are marked as A, B & C. Objective function at each point is evaluated and tabulated in the table below. The point that maximizes objective function is selected as the optimal point.

Graphical Representation of LP model for factory



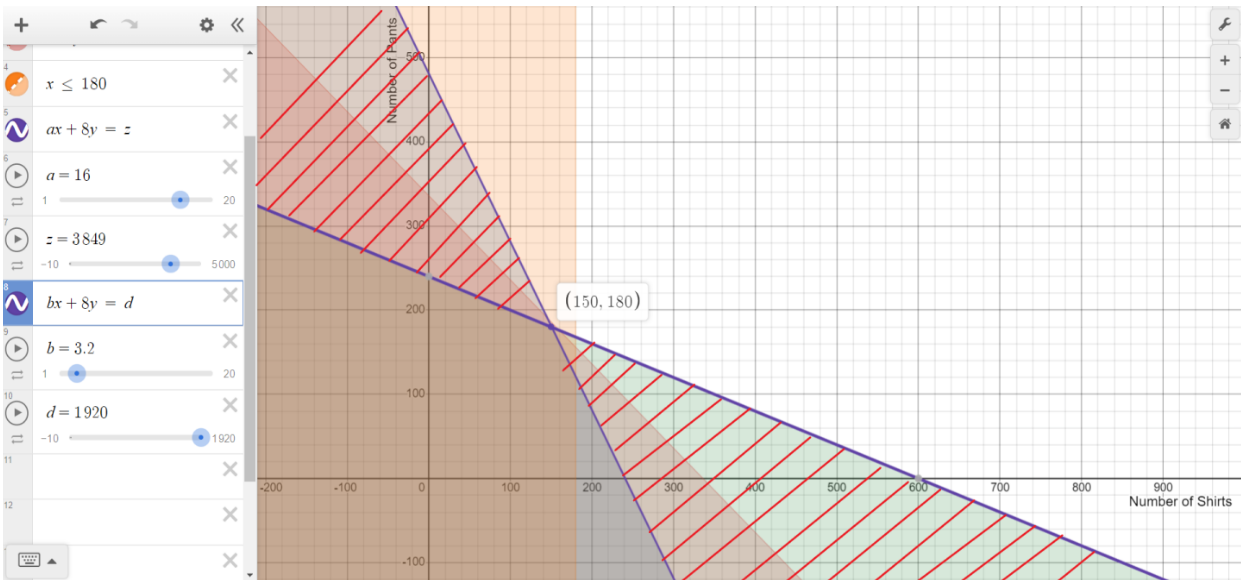
Optimal point of feasible region:

Point	A (0, 240)	B (150, 180)	C (180, 120)	D (180, 0)
Max Profit	z(A) = 10×0 + 8×240 z(A) = 1920	z(B) = 10×150 + 8×180 z(B) = 2940	z(C) = 10×180 + 8×120 z(C) = 2760	z(D) = 10×180 + 8×0 z(D) = 1800

Hence, the maximum profit z = 2940 is when x = 150, y = 180. Optimal daily profit for the factory is achieved when the number of shirts produced is 150 and the number of pants produced is 180.

d)

Range for the profit ($) per shirt without affecting the optimal point of (150, 180)



The range of the profit per shirt is from 3.2$ to 16$ without affecting the optimal point of (150, 180). Beyond [3.2, 16], the optimal point of (150,180) starts to deflect.

Solution 2

a)

Let $x_{ij} \geq 0$ be a decision variable that denotes the number of tons of products i to be produced from materials j for i $\in$ {1 = Bloom, 2 = Amber, 3 = Leaf} and j $\in$ {1 = Cotton, 2 = Wool, 3 = Nylon}.

Total Revenue from sale of products:
$60(x_{11}+x_{12}+x_{13}) + 55(x_{21}+x_{22}+x_{23}) + 60(x_{31}+x_{32}+x_{33})$

Total Production Cost of all three products:
$5(x_{11}+x_{12}+x_{13}) + 4(x_{21}+x_{22}+x_{23}) + 5(x_{31}+x_{32}+x_{33})$

Total Purchase Cost of raw materials:
$40(x_{11}+x_{21}+x_{31}) + 45(x_{12}+x_{22}+x_{32}) + 30(x_{13}+x_{23}+x_{33})$

Objective function is to maximize daily profit:

Daily Profit = revenue - production cost - purchase cost

$$\begin{aligned}\max z &= 60(x_{11}+x_{12}+x_{13}) + 55(x_{21}+x_{22}+x_{23}) + 60(x_{31}+x_{32}+x_{33}) \\ &\quad - (5(x_{11}+x_{12}+x_{13}) + 4(x_{21}+x_{22}+x_{23}) + 5(x_{31}+x_{32}+x_{33})\ ) \\ &\quad - (40(x_{11}+x_{21}+x_{31}) + 45(x_{12}+x_{22}+x_{32}) + 30(x_{13}+x_{23}+x_{33})) \\ &= 15x_{11} + 10x_{12} + 25x_{13} + 11x_{21} + 6x_{22} + 21x_{23} + 15x_{31} + 10x_{32} + 25x_{33}\end{aligned}$$

Demand Constraints:

- $x_{11} + x_{12} + x_{13} \leq 4200$ (Max demand 4200 tons for Bloom)
- $x_{21} + x_{22} + x_{23} \leq 3200$ (Max demand 3200 tons for Amber)
- $x_{31} + x_{32} + x_{33} \leq 3500$ (Max demand 3500 tons for Leaf)

Supply Constraints:

- $\dfrac{x_{11}}{x_{11} + x_{12} + x_{13}} \geq 0.5 \Rightarrow 0.5x_{11} - 0.5x_{12} - 0.5x_{13} \geq 0$ (Min 50% cotton proportion in Bloom)
- $\dfrac{x_{21}}{x_{21} + x_{22} + x_{23}} \geq 0.6 \Rightarrow 0.4x_{21} - 0.6x_{22} - 0.6x_{23} \geq 0$ (Min 60% cotton proportion in Amber)
- $\dfrac{x_{31}}{x_{31} + x_{32} + x_{33}} \geq 0.5 \Rightarrow 0.5x_{31} - 0.5x_{32} - 0.5x_{33} \geq 0$ (Min 50% cotton proportion in Leaf)
- $\dfrac{x_{12}}{x_{11} + x_{12} + x_{13}} \geq 0.4 \Rightarrow -0.4x_{11} + 0.6x_{12} - 0.4x_{13} \geq 0$ (Min 40% wool proportion in Bloom)
- $\dfrac{x_{22}}{x_{21} + x_{22} + x_{23}} \geq 0.4 \Rightarrow -0.4x_{21} + 0.6x_{22} - 0.4x_{23} \geq 0$ (Min 40% wool proportion in Amber)
- $\dfrac{x_{32}}{x_{31} + x_{32} + x_{33}} \geq 0.3 \Rightarrow -0.3x_{31} + 0.7x_{32} - 0.3x_{33} \geq 0$ (Min 30% wool proportion in Leaf)

Non-negativity constraint:

- $x_{ij} \geq 0$ (i = 1,2,3 & j = 1,2,3) (x and y should be non-negative)

Garment Factory Linear Programming Model

max $z = 15x_{11} + 10x_{12} + 25x_{13} + 11x_{21} + 6x_{22} + 21x_{23} + 15x_{31} + 10x_{32} + 25x_{33}$

s.t.
$x_{11} + x_{12} + x_{13} \leq 4200$
$x_{21} + x_{22} + x_{23} \leq 3200$
$x_{31} + x_{32} + x_{33} \leq 3500$
$0.5x_{11} - 0.5x_{12} - 0.5x_{13} \geq 0$
$0.4x_{21} - 0.6x_{22} - 0.6x_{23} \geq 0$
$0.5x_{31} - 0.5x_{32} - 0.5x_{33} \geq 0$
$-0.4x_{11} + 0.6x_{12} - 0.4x_{13} \geq 0$
$-0.4x_{21} + 0.6x_{22} - 0.4x_{23} \geq 0$
$-0.3x_{31} + 0.7x_{32} - 0.3x_{33} \geq 0$

b)

Refer to R Code (MadhaviJoshi-code.R)

Optimal profit of **$141850** is achieved when the products are produced as indicated in the table below.

Product	Optimal Quantity (in tons)	Use of Cotton (in tons)	Use of Wool (in tons)	Use of Nylon (in tons)
Bloom	4200	2100	1680	420
Amber	3200	1920	1280	0
Leaf	3500	1750	1050	700

Inference:

In order to achieve the maximum profit of $141850, 4200 tons of Bloom needs to be produced using 2100 tons of cotton, 1680 tons of cool and 420 tons of nylon. The Optimum quantity for Amber is 3200 tons which needs to be produced using 1920 tons of cotton & 1280 tons of wool only. Similarly, an optimal quantity of Leaf of 3500 tons needs to be produced using 1750 tons of cotton, 1050 tons of wool and 700 tons of nylon.

<u>R Code for Solution 2</u>

```
# Loading the necessary Packages
library(lpSolveAPI)

# Create a LP model to hold 9 variables and 9 constraints
lpfactory <- make.lp(9, 9)

# Objective function is to maximize the profit
lp.control(lpfactory, sense= "maximize")

# set objective function
set.objfn(lpfactory, c(15, 10, 25, 11, 6, 21, 15, 10, 25))

# set Demand constraint of Max demand 4200 tons for Bloom
set.row(lpfactory, 1, c(1,1,1), indices = c(1,2,3))

# set Demand constraint of Max demand 3200 tons for Amber
set.row(lpfactory, 2, c(1,1,1), indices = c(4,5,6))

# set Demand constraint of Max demand 3500 tons for Leaf
set.row(lpfactory, 3, c(1,1,1), indices = c(7,8,9))

# set constraint of Min 50% cotton proportion in Bloom
set.row(lpfactory, 4, c(0.5,-0.5,-0.5), indices = c(1,2,3))

# set constraint of Min 60% cotton proportion in Amber
set.row(lpfactory, 5, c(0.4,-0.6,-0.6), indices = c(4,5,6))

# set constraint of Min 50% cotton proportion in Leaf
set.row(lpfactory, 6, c(0.5,-0.5,-0.5), indices = c(7,8,9))

# set constraint of Min 40% wool proportion in Bloom
set.row(lpfactory, 7, c(-0.4,0.6,-0.4), indices = c(1,2,3))

# set constraint of Min 40% wool proportion in Amber
set.row(lpfactory, 8, c(-0.4,0.6,-0.4), indices = c(4,5,6))

# set constraint of Min 30% wool proportion in Leaf
set.row(lpfactory, 9, c(-0.3,0.7,-0.3), indices = c(7,8,9))

# Set the RHS values for all above constraints in same order
set.rhs(lpfactory, c(4200, 3200, 3500, 0, 0, 0, 0, 0, 0))

# Set the comparison for all above constraints in same order
set.constr.type(lpfactory, c("<=", "<=", "<=", ">=", ">=", ">=", ">=", ">=", ">="))
```

```
# All values have to be real numbers
set.type(lpfactory, c(1:9),"real")

# set Non-negativity Constraints for all 9 variables where  value range is [0, Inf]
set.bounds(lpfactory, lower = rep(0, 9), upper = rep(Inf, 9))

# Solve the LP problem of the model
solve(lpfactory)
# Get the maximum objective function value (profit)
objvalue<-get.objective(lpfactory)

# Print the maximum objective function value (profit)
cat("The maximum profit for factory given the constraints is $",objvalue,"\n")

# Get the optimal solution for each decision variable
solution<-get.variables(lpfactory)

# Print the optimal solution for each decision variable solution
# Get the Optimal Quantity of Bloom to be produced by adding first 3 elements
bloomQty <- sum(solution[c(1,2,3)])
cat("The Optimal Quantity of Bloom to be produced is ",bloomQty,"tons using:\n",
   solution[c(1)], "tons of cotton\n",
   solution[c(2)], "tons of Wool\n ",
   solution[c(3)], "tons of Nylon\n")

# Get the Optimal Quantity of Amber to be produced by adding second 3 elements
amberQty <- sum(solution[c(4,5,6)])
cat("The Optimal Quantity of Amber to be produced is ",amberQty,"tons using:\n",
   solution[c(4)], "tons of cotton\n",
   solution[c(5)], "tons of Wool\n ",
   solution[c(6)], "tons of Nylon\n")

# Get the Optimal Quantity of Leaf to be produced by adding last 3 elements
leafQty <- sum(solution[c(7,8,9)])
cat("The Optimal Quantity of Leaf to be produced is ",leafQty,"tons using:\n",
   solution[c(7)], "tons of cotton\n",
   solution[c(8)], "tons of Wool\n ",
   solution[c(9)], "tons of Nylon\n")
```

Solution 3

a)

Two-players zero-sum game

A two-players zero-sum game is a game where one player's gain is exactly equal to the other player's loss, so the total outcome is zero. This is a two-players game because there are only two players Giant and Sky in this bidding process. The gain of one player is equal to the loss of the other player. For example, if Giant wins the bid at $35 Million, Sky loses the opportunity to build in the field, and vice versa. Also, there is no cooperation between the two players, and each player acts solely in their own self-interest to win the bid and build in the field.

Thus, the problem can be described as a two-players-zero-sum game because there are only two players involved and their gains are equal to the losses of the other player.

b)

Understanding the Game Structure

In a zero sum game case like this, whatever amount one company bids and wins the contract for, the other company loses that potential income. The interests of the two players (Giant and Sky) are completely opposed. In such games, one player's gain is the other player's loss, and the total payoff to all players in the game adds up to zero for any outcome. This is because the 'prize' is the right to build in a field with a fixed value; it's a contest over who gets it, not about creating additional value.

Payoff Matrix Criteria

To formulate the payoff matrix, we must consider every possible bid combination between Giant and Sky. Let us consider the conditions mentioned in the problem statement:

- The winner is the company with the higher bid.
- If Giant and Sky tie, Giant wins (as per the agreement) and Sky loses.
- If Giant bids more than $35M, Giant loses except in case of a tie.

Constructing the Payoff Matrix

Let's denote the bids by $B = \{10,20,30,35,40\}$ million dollars. Considering all possible combinations of bids, the payoff matrix for the game can be formulated as below. Each cell represents an outcome from a possible pair of bids. Giant's payoff is the value of the contract minus $35 million (since they view any price above that as a loss), and Sky's payoff is the negative of Giant's because it's a zero-sum game.

Calculating and Interpreting the Payoffs

For each pair of bids (G,S), where G is Giant's bid and S is Sky's bid:

- If $G \geq S$, Giant wins and the payoff is $(G-35)$ for Giant and $(35-G)$ for Sky.
- If $G < S$, Sky wins and the payoff is $(35-S)$ for Giant and $(S-35)$ for Sky.

Payoff matrix representing Giant's gains and Sky's losses in millions of dollars

Company		Sky				
Giant	**Bid Value**	**$10M**	**$20M**	**$30M**	**$35M**	**$40M**
	$10M	-25	15	5	0	-5
	$20M	-15	-15	5	0	-5
	$30M	-5	-5	-5	0	-5
	$35M	0	0	0	0	-5
	$40M	5	5	5	5	5

c)

Finding the Saddle Point for the given model

A saddle point is a solution where the minimum value in a row is equal to the maximum value in a corresponding column. It is a position which, once reached, neither player benefits from moving away from it. Saddle point represents the optimal strategy for both players, as neither player can improve their outcome by changing their strategy.

Finding Saddle Point from the Matrix

Company		Sky					Minimum of each row
Giant	**Bid**	**$10M**	**$20M**	**$30M**	**$35M**	**$40M**	
	$10M	-25	15	5	0	-5	-25
	$20M	-15	-15	5	0	-5	-15
	$30M	-5	-5	-5	0	-5	-5
	$35M	0	0	0	0	-5	-5
	$40M	$\boxed{5}$	5	$\boxed{5}$	$\boxed{5}$	$\boxed{5}$	5
Maximum of each column		5	15	5	5	5	

← **Maximum of min of row**

↑ ↑ ↑ ↑

Minimum of maximum of column

The saddle points in the payoff matrix occur at the positions (4,0) , (4,2), (4,3) , (4,4). These positions correspond to Giant's bid of $40 million against Sky's bids of $10 million, $30 million, $35 million, and $40 million, respectively. The presence of multiple saddle points suggests multiple optimal strategies for Giant, with a bid of $40 million being the dominant strategy. In order to get the maximum game value, companies must go for mixed strategy.

d)

Linear programming model for Company Sky

The LP model for Company Sky in this game can be constructed by setting up constraints and the objective function. The constraints include keeping the bid amount within the available options and ensuring that the outcome is not worse than not getting the field. The objective is to maximize the outcome for Sky. The linear programming model for Company Sky can be formulated as below.

Suppose Company Sky chooses the mixed strategy with y_1, y_2, y_3, y_4, y_5 being the probabilities for selecting a bid, then

- Expected Payoff for Sky when Giant bids for \$10M is $(-25y_1+15y_2+5y_3-5y_5)$
- Expected Payoff for Sky when Giant bids for \$20M is $(-15y_1-15y_2+5y_3-5y_5)$
- Expected Payoff for Sky when Giant bids for \$30M is $(-5y_1-5y_2-5y_3-5y_5)$
- Expected Payoff for Sky when Giant bids for \$35M is $(-5y_5)$
- Expected Payoff for Sky when Giant bids for \$40M is $(5y_1+5y_2+5y_3+5y_4+5y_5)$

Giant will choose a strategy to ensure that they obtain an expected reward of max $(-25y_1+15y_2+5y_3-5y_5, -15y_1-15y_2+5y_3-5y_5, -5y_1-5y_2-5y_3-5y_5, -5y_5, 5y_1+5y_2+5y_3+5y_4+5y_5)$. Therefore Sky should choose $(y_1, y_2, y_3, y_4, y_5)$ to make Max $(-25y_1+15y_2+5y_3-5y_5, -15y_1-15y_2+5y_3-5y_5, -5y_1-5y_2-5y_3-5y_5, -5y_5, 5y_1+5y_2+5y_3+5y_4+5y_5)$ as small as possible.

Objective function is to minimize the game value:

- Minimize $z = v$

Strategy Constraint by Giant:

- $v \geq -25y_1 + 15y_2 + 5y_3 - 5y_5$
  ➡ $v + 25y_1 - 15y_2 - 5y_3 + 5y_5 \geq 0$ (Giant chooses \$10M Bid value)
- $v \geq -15y_1 - 15y_2 + 5y_3 - 5y_5$
  ➡ $v + 15y_1 + 15y_2 - 5y_3 + 5y_5 \geq 0$ (Giant chooses \$20M Bid value)
- $v \geq -5y_1 - 5y_2 - 5y_3 - 5y_5$
  ➡ $v + 5y_1 + 5y_2 + 5y_3 + 5y_5 \geq 0$ (Giant chooses \$30M Bid value)
- $v \geq -5y_5$
  ➡ $v + 5y_5 \geq 0$ (Giant chooses \$35M Bid value)
- $v \geq 5y_1 + 5y_2 + 5y_3 + 5y_4 + 5y_5$
  ➡ $v - 5y_1 - 5y_2 - 5y_3 - 5y_4 - 5y_5 \geq 0$ (Giant chooses \$40M Bid value)

Sum of Probability Constraint:

- $y_1 + y_2 + y_3 + y_4 + y_5 = 1$ (sum of probabilities equal to 1)

Non-negativity constraint:

- $y_i \geq 0$ $(i = 1,2,3,4,5)$ (y should be non-negative)

Linear Programming Model for Construction Company Sky

min $z = v$

s.t $v + 25y_1 - 15y_2 - 5y_3 + 5y_5 \geq 0$

$v + 15y_1 + 15y_2 - 5y_3 + 5y_5 \geq 0$

$v + 5y_1 + 5y_2 + 5y_3 + 5y_5 \geq 0$

$v + 5y_5 \geq 0$

$v - 5y_1 - 5y_2 - 5y_3 - 5y_4 - 5y_5 \geq 0$

$y_1 + y_2 + y_3 + y_4 + y_5 = 1$

$y_i \geq 0$ $(i = 1,2,3,4,5)$

v u.r.s (means - unrestricted sign)

e) Refer to section "Solution 3 For Construction Company Sky" in the file *MadhaviJoshi-code.R*

```
# Loading the necessary Packages
install.packages('lpSolveAPI')
library(lpSolveAPI)
install.packages('hash')
library(hash)

# Create a LP model to hold 6 variables (y1, y2, y3, y4, y5 and v)
lpsky <- make.lp(0, 6)

# Set objective function (z = v)
set.objfn(lpsky, c(0, 0, 0, 0, 0, 1))

# Objective function is to minimize the game value
lp.control(lpsky, sense="min")

# Add the strategy constraint of Giant choosing $10M Bid value
add.constraint(lpsky, c(25,-15,-5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $20M Bid value
add.constraint(lpsky, c(15,15,-5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $30M Bid value
add.constraint(lpsky, c(5,5,5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $35M Bid value
add.constraint(lpsky, c(0,0,0,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $40M Bid value
add.constraint(lpsky, c(-5,-5,-5,-5,-5,1), ">=", 0)

# Add the constraint of sum of probabilities be equal to 1
add.constraint(lpsky, c(1, 1, 1, 1, 1, 0), "=", 1)

# set Non-negativity Constraints for all 5 probability variables where
# value range is [0, Inf]. v is unrestricted sign.
set.bounds(lpsky, lower=c( 0, 0, 0, 0, 0, -Inf))

# solve the LP problem
solve(lpsky)

# Get the value of the game which is given by objective function
get.objective(lpsky)

# Get the solution of the game which tells the probability with which
# Sky should consider the bidding value
val <- get.variables(lpsky)

# The first 5 variables denote the probability of choosing the bid value. Get
# the probability values returned by the LP model.
probability_list <- val[1:5]

# Store the 6th variable denoting the game value given by LP model.
GameVal <- val[6]

# Create Map table to retrieve the bid value from Probability list
BiddingValue <- hash()
BiddingValue[["1"]] <- "$10M"                    # Key 1 denotes $10M
BiddingValue[["2"]] <- "$20M"                    # Key 2 denotes $20M
BiddingValue[["3"]] <- "$30M"                    # Key 3 denotes $30M
BiddingValue[["4"]] <- "$35M"                    # Key 4 denotes $35M
BiddingValue[["5"]] <- "$40M"                    # Key 5 denotes $40M

# Retrieve the probability which is non-zero
prob = which(probability_list != 0, arr.ind = T)

# Print the Bid value that Sky must choose as optimal solution
cat("The construction company Sky must choose the bid value of",
    values(BiddingValue, keys=prob), "as optimal strategy to maximise its
    chances of winning against Giant.")
```

f)

Inference by using the linear programming model for Sky

From the results obtained from the solution of LP model for Sky indicates that the optimal strategy for Sky is to bid for $30 Million. This strategy will maximize Sky's chances of winning the field, as it will only lose in the case of a tie, which is a relatively unlikely outcome. The presence of multiple saddle points suggests multiple optimal strategies for Giant. For Sky, the survey information that Giant possesses is unknown, but the analysis here indicates that to minimize losses, Sky should never bid above $30million. This strategic interaction emphasizes the importance of information in competitive bidding situations. Since Giant is aware due to survey that getting the field for more than $ 35 Million is as bad as not getting it, best bet for Giant would be to bid for <= $35M. The outcome is favorable for Sky by bidding for $30 Million increasing the chances of winning the bid as well as minimizing the loss to a maximum of $5 Million.

References

1. Taha H (2017) Operations Research an Introduction, EBook, Global Edition, Pearson Education Limited, Harlow, UK
2. https://www.torontomu.ca/content/dam/tedrogersschool/success/resources/ITM107_LP_Sensitivity_Analysis.pdf
3. https://iopscience.iop.org/article/10.1088/1757-899X/245/6/062047/pdf

Solution 3 For Construction Company Sky

```
# Loading the necessary Packages
install.packages('lpSolveAPI')
library(lpSolveAPI)
install.packages('hash')
library(hash)

# Create a LP model to hold 6 variables (y1, y2, y3, y4, y5 and v)
lpsky <- make.lp(0, 6)

# Set objective function (z = v)
set.objfn(lpsky, c(0, 0, 0, 0, 0, 1))

# Objective function is to minimize the game value
lp.control(lpsky, sense="min")

# Add the strategy constraint of Giant choosing $10M Bid value
add.constraint(lpsky, c(25,-15,-5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $20M Bid value
add.constraint(lpsky, c(15,15,-5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $30M Bid value
add.constraint(lpsky, c(5,5,5,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $35M Bid value
add.constraint(lpsky, c(0,0,0,0,5,1), ">=", 0)

# Add the strategy constraint of Giant choosing $40M Bid value
add.constraint(lpsky, c(-5,-5,-5,-5,-5,1), ">=", 0)

# Add the constraint of sum of probabilities be equal to 1
add.constraint(lpsky, c(1, 1, 1, 1, 1, 0), "=", 1)
```

```
# set Non-negativity Constraints for all 5 probability variables where value range is [0, Inf]. v is unrestricted sign.

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


# solve the LP problem

solve(lpsky)


# Get the value of the game which is given by objective function

get.objective(lpsky)


# Get the solution of the game which tells the probability with which Sky should consider the bidding value

val <- get.variables(lpsky)


# The first 5 variables denote the probability of choosing the bid value. Get the probability values returned by the LP model.

probability_list <- val[1:5]


# Store the 6th variable denoting the game value given by LP model.

GameVal <- val[6]


# Create Map table to retrieve the bid value from Probability list

BiddingValue <- hash()

BiddingValue[["1"]] <- "$10M"                    # Key 1 denotes $10M

BiddingValue[["2"]] <- "$20M"                    # Key 2 denotes $20M

BiddingValue[["3"]] <- "$30M"                    # Key 3 denotes $30M

BiddingValue[["4"]] <- "$35M"                    # Key 4 denotes $35M

BiddingValue[["5"]] <- "$40M"                    # Key 5 denotes $40M


# Retrieve the probability which is non-zero

prob = which(probability_list != 0, arr.ind = T)


# Print the Bid value that Sky must choose as optimal solution

cat("The construction company Sky must choose the bid value of",

        values(BiddingValue, keys=prob), "as optimal strategy to maximise its

        chances of winning against Giant.")
```