Linear Programming Formulations

Formulations and Calculations using R programming

Aung Pyae

Asia Pacific University of Technology and Innovation

aungpyaeap@outlook.com

March 2019

Topics include

- Simple Linear Programming
- Transportation, Transshipment and Network Models
- Goal Programming
- Integer Linear Programming

Linear Programming

To achieve the best outcome (such as maximum profit or lowest cost)

Step 1: Define decision variables

Step 2: Set objective function

Step 3: Set constraints

Step 4: Non-negativity restriction

Exercise 1

Consider a chocolate manufacturing company which produces only two types of chocolate – A and B. Both the chocolates require Milk and Choco only. To manufacture each unit of A and B, following quantities are required:

- Each unit of A requires 1 unit of Milk and 3 units of Choco
- Each unit of B requires 1 unit of Milk and 2 units of Choco

The company kitchen has a total of 5 units of Milk and 12 units of Choco. On each sale, the company makes a profit of

- $ 6 per unit A sold
- $ 5 per unit B sold

Now, the company wishes to maximize its profit. How many units of A and B should it produce respectively?

Solution

	Milk	**Choco**	**Profit per unit**
A	1	3	$ 6
B	1	2	$ 5
Total	5	12	

Define decision variables

Number of units produced in A = X

Number of units produced in B = Y

Set objective function

Max Z = 6X + 5Y

Constraints

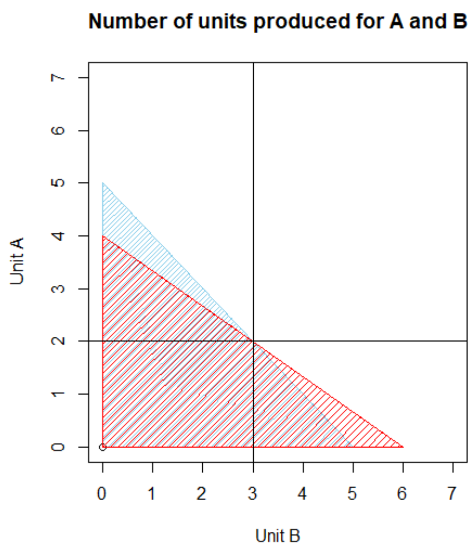
X + Y <= 5

3X + 2Y <= 12

X, Y >= 0 (non-negativity restriction)

Method 1 – graphical calculation

Let X = 0	**Let Y = 0**	
X + Y = 5 0 + Y = 5 Y = 5	X + Y =5 X + 0 = 5 X = 0	When X: 0, 5 Y: 5, 0
3X + 2Y = 12 3 * 0 + 2Y = 12 0 + 2Y = 12 Y = 6	3X + 2Y =12 3X + 2 * 0 =12 3X + 0 = 12 X = 12/3 = 4	When X: 0, 4 Y: 6, 0



Therefore, X = **2**, Y = **3**

Z = 6X + 5Y
Z = 6 * 2 + 5 * 3
Z = **27**

Method 2 – using R

```
library(lpSolveAPI)
lp.model = make.lp(0,2)
set.objfn(lp.model, c(6,5))
add.constraint(lp.model, c(1,1), "<=", 5)
add.constraint(lp.model, c(3,2), "<=", 12)

lp.control(lp.model, sense='max')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)

#plot visual presentation
plot(0,0, xlim = c(0,7), ylim = c(0,7), xlab = "Unit B", ylab = "Unit A", main = "Number of units produced for A and B")
polygon(c(0,5,0), c(5,0,0), col = "skyblue", density = 30)
polygon(c(0,6,0), c(4,0,0), col = "red", density = 20)
abline(h=2, v=3)
```

Exercise 2

A farmer has recently acquired a 110 hectares piece of land. He has decided to grow Wheat and barley on that land. Due to the quality of the sun and the region's excellent climate, the entire production of Wheat and Barley can be sold. He wants to know how to plant each variety in the 110 hectares, given the costs, net profits and labor requirements according to the data shown below:

Variety	**Cost (Price/Hec)**	**Net profit (Price/Hec)**	**Man-days/Hec**
Wheat	100	50	10
Barley	200	120	30

The farmer has a budget of US$10,000 and an availability of 1,200 man-days during the planning horizon. Find the optimal solution and the optimal value.

Solution

Decision variables

Number of wheat product = X

Number of barley produce = Y

Objective function

Max Z = 50X + 120Y

Constraints

100X + 200Y <= 10000

10X + 30Y <= 1200

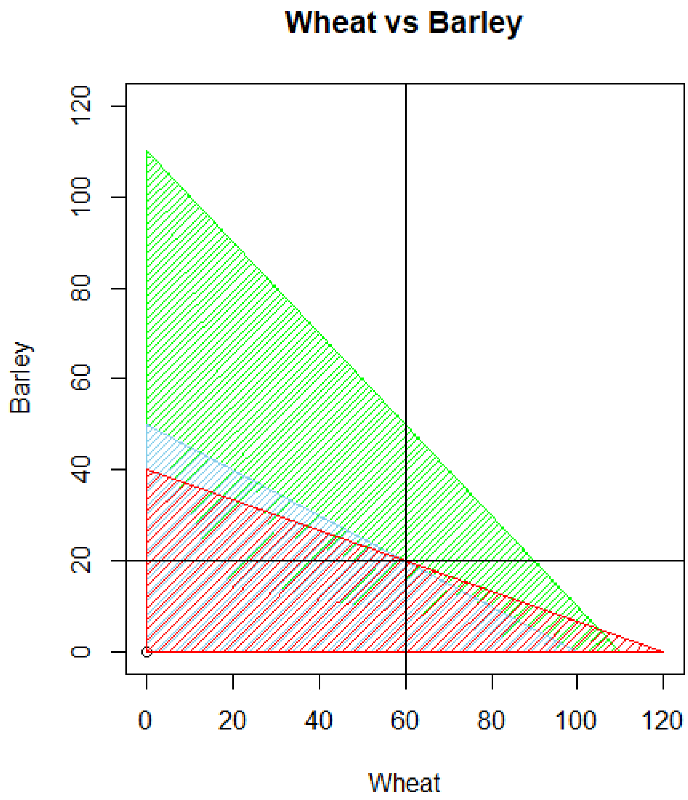
X + Y <= 110

X, Y >= 0 (non-negativity restriction)

Method 1 – graphical calculation

- 100X + 200Y ≤ 10,000 can be simplified to X + 2Y ≤ 100 by dividing by 100.
- 10X + 30Y ≤ 1200 can be simplified to X + 3Y ≤ 120 by dividing by 10.
- The third equation is in its simplified form, X + Y ≤ 110.

<table>
<tr><th>Let X = 0</th><th>Let Y = 0</th><th></th></tr>
<tr><td>X + 2Y = 100
0 + 2Y = 100
Y = 100/2
Y = 50</td><td>X + 2Y =100
X + 0 =100
X = 100</td><td>When<table><tr><td>X</td><td>0</td><td>100</td></tr><tr><td>Y</td><td>50</td><td>0</td></tr></table></td></tr>
<tr><td>X + 3Y = 120
0 + 3Y = 120
Y = 120/3
Y = 40</td><td>X + 3Y =120
X + 0 = 120
X =120</td><td>When<table><tr><td>X</td><td>0</td><td>120</td></tr><tr><td>Y</td><td>40</td><td>0</td></tr></table></td></tr>
<tr><td>X + Y = 110
0 + Y = 110
Y =110</td><td>X + Y =110
X + 0 =110
X =110</td><td>When<table><tr><td>X</td><td>0</td><td>110</td></tr><tr><td>Y</td><td>110</td><td>0</td></tr></table></td></tr>
</table>



Therefore, X = **60**, Y = **20**

Z = 50X + 120Y
Z = 50 * 60 + 120 * 20
Z = **5400**

Method 2 – using R

```
library(lpSolveAPI)
lp.model = make.lp(0,2)
set.objfn(lp.model, c(50,120))
add.constraint(lp.model, c(100,200), "<=", 10000)
add.constraint(lp.model, c(10,30), "<=", 1200)
add.constraint(lp.model, c(1,1), "<=", 110)

lp.control(lp.model, sense='max')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)

#plot visual presentation
plot(0,0, xlim = c(0,120), ylim = c(0,120), xlab = "Wheat", ylab = "Barley", main = "Wheat vs Barley")
polygon(c(0,110,0), c(110,0,0), col = "green", density = 30)
polygon(c(0,100,0), c(50,0,0), col = "skyblue", density = 30)
polygon(c(0,120,0), c(40,0,0), col = "red", density = 20)
abline(h=20, v=60)
```

Exercise 3

You are taking a test in which items of Type A are worth 10 points and items of Type B are worth 15 points. It takes 3 minutes to complete each item of Type A and 6 minutes to complete each item of Type B. You are allowed 1 hour for this test, and you may not answer more than 16 questions. Assuming all answers are correct, how many items should you answer for each type to get the best score?

Solution

	Points	Minutes
Type A	10	3
Type B	15	6

Decision variables

Number of questions answered in Type A = X

Number of questions answered in Type B = Y

Objective function

Max Z = 10X + 15Y

Constraints

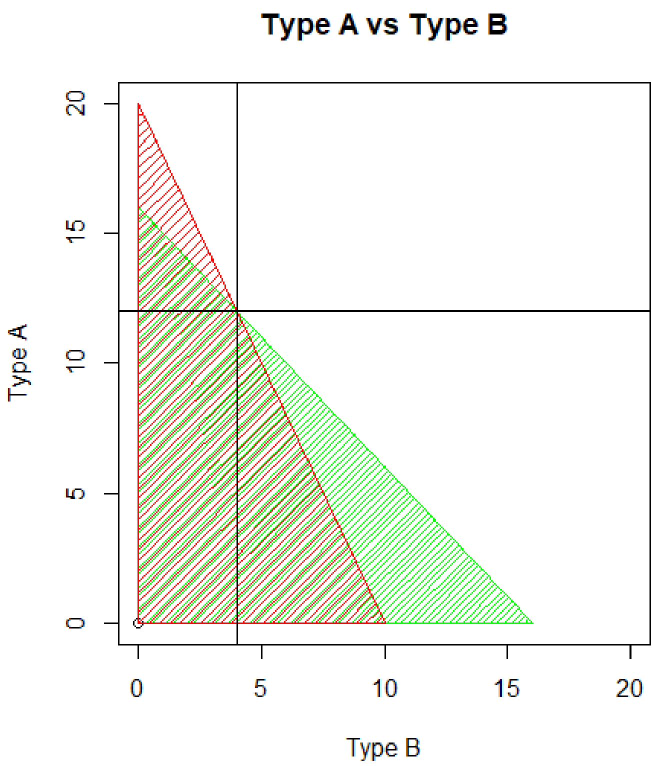
X + Y <= 16

3X + 6Y <= 60

X, Y >= 0 (non-negativity restriction)

Method 1 – graphical calculation

<table>
<tr><td>Let X = 0</td><td>Let Y = 0</td><td></td></tr>
<tr><td>X + Y =15
0 + Y = 16
Y = 16</td><td>X + Y =16
X + 0 =16
X = 16</td><td>When<table><tr><td>X</td><td>0</td><td>16</td></tr><tr><td>Y</td><td>16</td><td>0</td></tr></table></td></tr>
<tr><td>3X + 6Y = 60
0 + 6Y = 60
Y = 60/6
Y = 10</td><td>3X + 6Y = 60
3X + 0 =60
X = 60/3
X = 20</td><td>When<table><tr><td>X</td><td>0</td><td>20</td></tr><tr><td>Y</td><td>10</td><td>0</td></tr></table></td></tr>
</table>



Therefore, X = **12**, Y = **4** Z = 10X + 15Y Z = 10 * 12 + 15 * 4 Z = **180**

Method 2 – using R

```
library(lpSolveAPI)
lp.model = make.lp(0,2)
set.objfn(lp.model, c(10,15))
add.constraint(lp.model, c(1,1), "<=", 16)
add.constraint(lp.model, c(3,6), "<=", 60)
lp.control(lp.model, sense='max')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)

#plot
plot(0,0, xlim = c(0,20), ylim = c(0,20), xlab = "Type B", ylab = "Type A", main = "Type A vs Type B")
polygon(c(0,16,0), c(16,0,0), col = "green", density = 30)
polygon(c(0,10,0), c(20,0,0), col = "red", density = 20)
abline(h=12, v=4)
```

Exercise 4

Stanley owns a small business that produces glass doors and windows. Stanley recently learned about linear optimization and wants to implement the framework to make his business more profitable. Currently, Stanley has 3 plants where he sources production. The production constraints and potential profits are outlined below:

	Production time per batch (hours)		
	Product		
	1	2	Production time available per week (hours)
Plant 1	1	0	4
Plant 2	0	2	12
Plant 3	3	2	18
Profit per batch	$ 3000	$ 5000	

Create and solve the LP. (Product mix problem)

Solution

Decision variables

X = Number of products produced for Product 1 (Doors)

Y = Number of products produces for Product 2 (Windows)

Objective function

Max Z = 3000X + 5000Y

Constraints

X + 0Y <= 4

0X + 2Y <= 12

3X + 2Y <= 18

X, Y >= 0 (non-negativity restriction)

```
library(lpSolveAPI)
lp.model = make.lp(0,2)
set.objfn(lp.model, c(3000,5000))
add.constraint(lp.model, c(1,0), "<=", 4)
add.constraint(lp.model, c(0,2), "<=", 12)
add.constraint(lp.model, c(3,2), "<=", 18)

lp.control(lp.model, sense='max')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)
```

Answer

X = **2**, Y = **6**, Z = **36000**

Exercise 5

Below there is a diet chart which gives me calories, protein, carbohydrate and fat content for 4 food items. Sara wants a diet with minimum cost. The diet chart is as follows:

	Food item 1	Food item 2	Food item 3	Food item 4
Calories	400	200	150	500
Protein (in grams)	3	2	0	0
Carbohydrates (in grams)	2	2	4	4
Fat (in grams)	2	4	1	5
Cost	$ 0.50	$ 0.20	$ 0.30	$ 0.80

The chart gives the nutrient content as well as the per-unit cost of each food item. The diet must be planned in such a way that it should contain at least 500 calories, 6 grams of protein, 10 grams of carbohydrates and 8 grams of fat.

Solution

Decision variables

Food item 1 = X_1, Food item 2 = X_2, Food item 3 = X_3, Food item 4 = X_4

Objective function

$$\text{Min } Z = 0.5X_1 + 0.2X_2 + 0.3X_3 + 0.8X_4$$

Constraints

$$400X_1 + 200X_2 + 150X_3 + 500X_4 >= 500$$

$$3X_1 + 2X_2 + 0X_3 + 0X_4 >= 6$$

$$2X_1 + 2X_2 + 4X_3 + 4X_4 >= 10$$

$$2X_1 + 4X_2 + X_3 + 5X_4 >= 8$$

$X_1, X_2, X_3, X_4 >= 0$ (non-negativity restriction)

```
library(lpSolveAPI)
lp.model = make.lp(0,4)
set.objfn(lp.model, c(0.5,0.2,0.3,0.8))
add.constraint(lp.model, c(400,200,150,500), ">=", 500)
add.constraint(lp.model, c(3,2,0,0), ">=", 6)
add.constraint(lp.model, c(3,2,4,4), ">=", 10)
add.constraint(lp.model.c(2,4,1,5), ">=", 8)

lp.control(lp.model, sense='min')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)
```

Answer

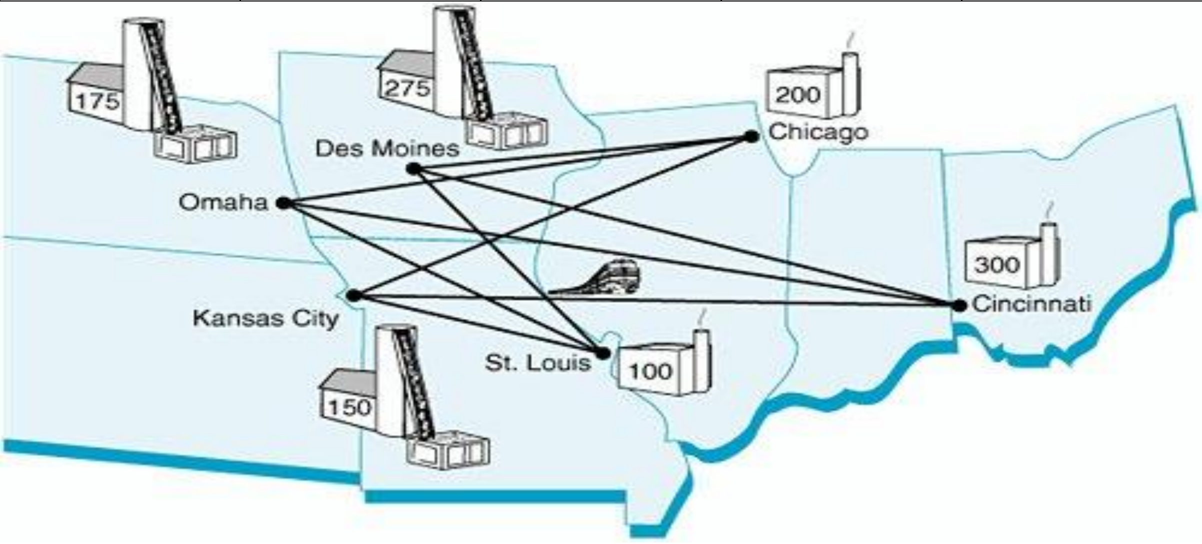
$X_1 = \mathbf{0}$, $X_2 = \mathbf{3}$, $X_3 = \mathbf{1}$, $X_4 = \mathbf{0}$

$Z = 0.5X_1 + 0.2X_2 + 0.3X_3 + 0.8X_4$
$Z = 0.5 * 1 + 0.2 * 3 + 0.3 * 1 + 0.8 * 0$
$Z = \mathbf{0.9}$

Transportation Models

Exercise 1

	Destinations	Chicago	St. Louis	Cincinnati
Origin	Kansas City	$ 6	$ 8	$ 10
	Omaha	$ 7	$ 11	$ 11
	Des Monies	$ 4	$ 5	$ 12



Solution

	Chicago - 4	St. Louis - 5	Cincinnati - 6		
Kansas City - 1	$ 6	$ 8	$ 10	**150**	Supply
Omaha - 2	$ 7	$ 11	$ 11	**175**	
Des Moines - 3	$ 4	$ 5	$ 12	**275**	
	200	**100**	**300**		
	Demand				

Decision variables

1. X_{14} = Number of products shipped from Kansas City to Chicago
2. X_{15} = Number of products shipped from Kansas City to St. Louis
3. X_{16} = Number of products shipped from Kansas City to Cincinnati
4. X_{24} = Number of products shipped from Omaha to Chicago
5. X_{25} = Number of products shipped from Omaha to St. Louis
6. X_{26} = Number of products shipped from Omaha to Cincinnati
7. X_{34} = Number of products shipped from Des Moines to Chicag0
8. X_{35} = Number of products shipped from Des Moines to St. Louis
9. X_{36} = Number of products shipped from Des Moines to Cincinnati

Objective function

$$\text{Min } Z = 6X_{14} + 5X_{15} + 10X_{16} + 7X_{24} + 11X_{25} + 11X_{26} + 4X_{34} + 5X_{35} + 6X_{36}$$

Constraints

Supply

$X_{14} + X_{15} + X_{16} <= 150$

$X_{24} + X_{25} + X_{26} <= 175$

$X_{34} + X_{35} + X_{36} <= 275$

Demand

$X_{14} + X_{24} + X_{34} = 200$

$X_{15} + X_{25} + X_{35} = 100$

$X_{16} + X_{26} + X_{36} = 300$

$X_{14}, X_{15}, X_{16}, X_{24}, X_{25}, X_{26}, X_{34}, X_{35}, X_{36} >= 0$ (non-negativity restriction)

```
library(lpSolveAPI)
lp.model <- make.lp(0,9)
set.objfn(lp.model, c(6,8,10,7,11,11,4,5,12))
add.constraint(lp.model, rep(1,3), indices = c(1,2,3), "<=", 150)
add.constraint(lp.model, rep(1,3), indices = c(4,5,6), "<=", 175)
add.constraint(lp.model, rep(1,3), indices = c(7,8,9), "<=", 275)
add.constraint(lp.model, rep(1,3), indices = c(1,4,7), "=", 200)
add.constraint(lp.model, rep(1,3), indices = c(2,5,8), "=", 100)
add.constraint(lp.model, rep(1,3), indices = c(3,6,9), "=", 300)

lp.control(lp.model, sense='min')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)

matrix(get.variables(lp.model),nrow = 3,byrow = TRUE, dimnames = list(c("Kansas City","Omaha","Des
Monies"),c("Chicago","St. Louis","Cincinnati")))
```

Answer

Z = **4245**

	Chicago	St. Louis	Cincinnati
Kansas City	**25**	**0**	**175**
Omaha	**0**	**0**	**100**
Des Moines	**125**	**175**	**0**

Exercise 2

Tri-County Utilities, Inc., supplies natural gas to customers in a three-county area. The company purchases natural gas from two companies: Southern Gas and Northwest Gas. Demand forecasts for the coming winter season are Hamilton County, 400 units; Butler County, 200 units; and Clermont County, 300 units. Contracts to provide the following quantities have been written: Southern Gas, 500 units; and Northwest Gas, 400 units. Distribution costs for the counties vary, depending upon the location of the suppliers. The distribution costs per unit (in thousands of dollars) are as follows:

From/To	Hamilton	Butler	Clermont
Southern Gas	10	20	15
Northwest Gas	12	15	18

a. Develop a linear programming model that can be used to determine the plan that will minimize total distribution costs.

b. Describe the distribution plan and show the total distribution cost.

Solution

	Hamilton - 3	**Butler - 4**	**Clermont - 5**		
Southern Gas - 1	10	20	15	500	Supply
Northwest Gas - 2	12	15	18	400	
	400	200	300		
	Demand				

Decision variables

1. X_{13} = Number of gas distributed from Southern Gas to Hamilton
2. X_{14} = Number of gas distributed from Southern Gas to Butler
3. X_{15} = Number of gas distributed from Southern Gas to Clermont
4. X_{23} = Number of gas distributed from Northwest Gas to Hamilton
5. X_{24} = Number of gas distributed from Northwest Gas to Butler
6. X_{25} = Number of gas distributed from Northwest Gas to Clermont

Objective function

$$\text{Min } Z = 10X_{13} + 20X_{14} + 15X_{15} + 12X_{23} + 15X_{24} + 18\,X_{25}$$

Constraints

Supply

$X_{13} + X_{14} + X_{15} <= 500$

$X_{23} + X_{24} + X_{25} <= 400$

Demand

$X_{13} + X_{23} = 400$

$X_{14} + X_{24} = 200$

$X_{15} + X_{25} = 300$

$X_{13}, X_{14}, X_{15}, X_{23}, X_{24}, X_{25} >= 0$ (non-negativity restriction)

```
library(lpSolveAPI)
lp.model <- make.lp(0,6)
set.objfn(lp.model, c(10,20,15,12,15,18))
add.constraint(lp.model, rep(1,3), indices = c(1,2,3), "<=", 500)
add.constraint(lp.model, rep(1,3), indices = c(4,5,6), "<=", 400)
add.constraint(lp.model, rep(1,2), indices = c(1,4), "=", 400)
add.constraint(lp.model, rep(1,2), indices = c(2,5), "=", 200)
add.constraint(lp.model, rep(1,2), indices = c(3,6), "=", 300)

lp.control(lp.model, sense='min')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)

matrix(get.variables(lp.model),nrow = 2,byrow = TRUE, dimnames = list(c("Southern Gas ","Northwest Gas
"),c("Hamilton","Butler","Clermont")))
```

Answer

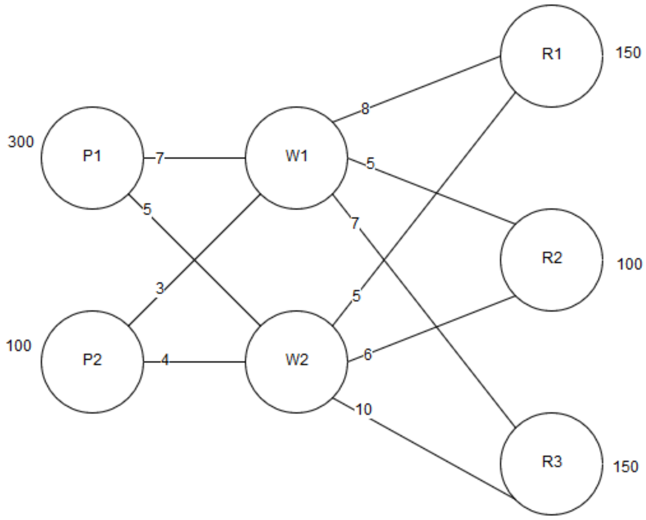
Z = **11900**

	Hamilton	Butler	Clermont
Southern Gas	**200**	**0**	**300**
Northwest Gas	**200**	**200**	**0**

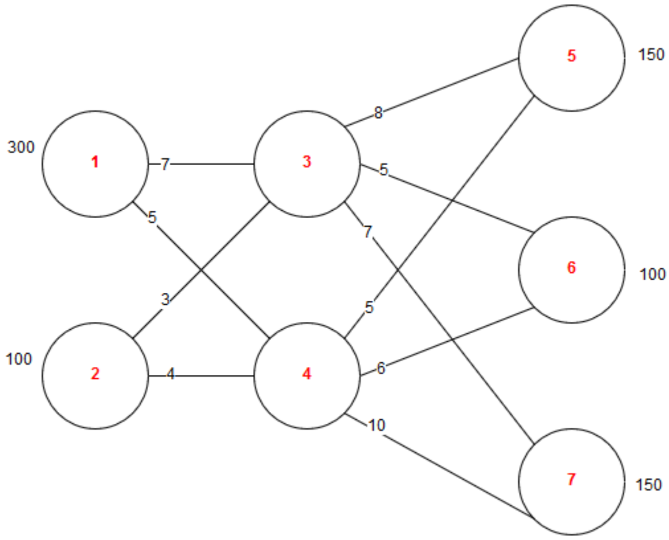
Network Model

Exercise 1

A company has two plants (P1 and P2), two regional warehouses (W1 and W2), and three retail outlets (R1, R2 and R3). The plant capacities, retail outlet demands, and per-unit shipping costs are shown in the following network. Formulate a linear programming model and determine the transportation plan to minimize the shipping costs.



Solution



Decision variables

1. X_{13} = Number of items shipped from P1 to W1
2. X_{14} = Number of items shipped from P1 to W2
3. X_{23} = Number of items shipped from P2 to W1
4. X_{24} = Number of items shipped from P2 to W2
5. X_{35} = Number of items shipped from W1 to R1
6. X_{36} = Number of items shipped from W1 to R2
7. X_{37} = Number of items shipped from W1 to R3
8. X_{45} = Number of items shipped from W2 to R1

9. X_{46} = Number of items shipped from W2 to R2
10. X_{47} = Number of items shipped from W2 to R3

Objective function

$$\text{Min } Z = 7X_{13} + 5X_{14} + 3X_{23} + 4X_{24} + 8X_{35} + 5X_{36} + 7X_{37} + 5X_{45} + 6X_{46} + 10X_{47}$$

Constraints

Supply

$X_{13} + X_{14} <= 300$

$X_{23} + X_{24} <= 100$

Demand

$X_{35} + X_{45} = 150$

$X_{36} + X_{46} = 100$

$X_{37} + X_{47} = 150$

Intermediate nodes

$\mathbf{X_{13} + X_{23}} = X_{35} + X_{36} + X_{37}$

$\mathbf{X_{14} + X_{24}} = X_{45} + X_{46} + X_{47}$

* **Flow in (+)**, Flow out (-) [place **center** the intermediate node]

$X_{13}, X_{14}, X_{23}, X_{24}, X_{35}, X_{36}, X_{37}, X_{45}, X_{46}, X_{47} >= 0$ (non-negativity restriction)

```
library(lpSolveAPI)
lp.model <- make.lp(0,10)
set.objfn(lp.model, c(7,5,3,4,8,5,7,5,6,10))
#supply
add.constraint(lp.model, rep(1,2), indices = c(1,2), "<=", 300)
add.constraint(lp.model, rep(1,2), indices = c(3,4), "<=", 100)
#intermediate nodes
add.constraint(lp.model, c(1,1,-1,-1,-1), indices = c(1,3,5,6,7), "=", 0)
add.constraint(lp.model, c(1,1,-1,-1,-1), indices = c(2,4,8,9,10), "=", 0)
#demand
add.constraint(lp.model, rep(1,2), indices = c(5,8), "=", 150)
add.constraint(lp.model, rep(1,2), indices = c(6,9), "=", 100)
add.constraint(lp.model, rep(1,2), indices = c(7,10), "=", 150)

lp.control(lp.model, sense='min')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)
```

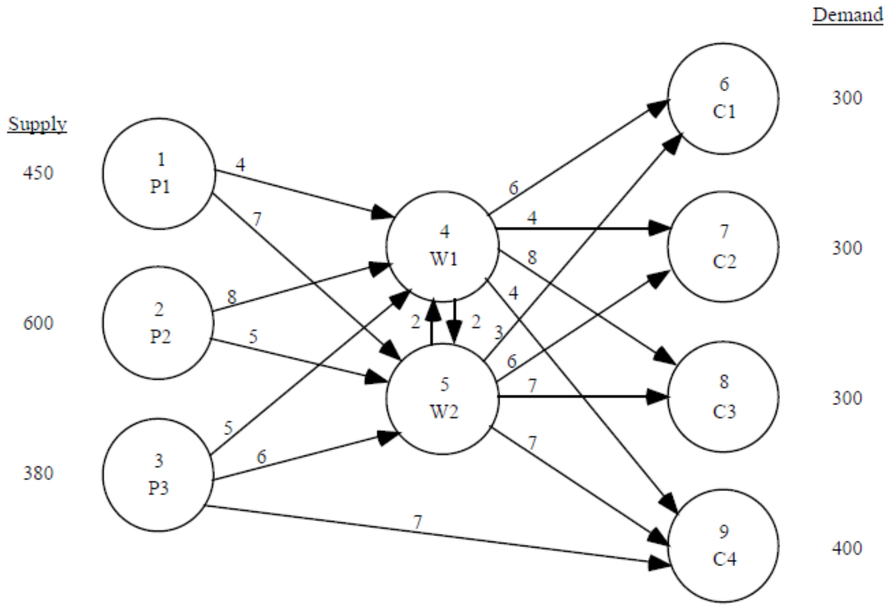
Answer

Z = **4300**

Decision variables = 50, 250, 100, 0, 0, 0, 150, 150, 100, 0

Exercise 2

A company has three plants (P1, P2 and P3), two regional warehouses (W1 and W2), and four retail outlets (C1, C2, C3 and C4). The plant capacities, retail outlet demands, and per-unit shipping costs are shown in the following network. Formulate a linear programming model to minimize the shipping costs and determine the optimal solution for the model.



Solution

Decision variables

1. X_{14} = Number of items shipped from P1 to W1
2. X_{15} = Number of items shipped from P1 to W2
3. X_{24} = Number of items shipped from P2 to W1
4. X_{25} = Number of items shipped from P2 to W2
5. X_{34} = Number of items shipped from P3 to W1
6. X_{35} = Number of items shipped from P3 to W2
7. X_{39} = Number of items shipped from P3 to C4
8. X_{45} = Number of items shipped from W1 to W2
9. X_{46} = Number of items shipped from W1 to C1
10. X_{47} = Number of items shipped from W1 to C2
11. X_{48} = Number of items shipped from W1 to C3
12. X_{49} = Number of items shipped from W1 to C4
13. X_{54} = Number of items shipped from W2 to W1
14. X_{56} = Number of items shipped from W2 to C1
15. X_{57} = Number of items shipped from W2 to C2
16. X_{58} = Number of items shipped from W2 to C3
17. X_{59} = Number of items shipped from W2 to C4

Objective function

Min $Z = 4X_{14} + 7X_{15} + 8X_{24} + 5X_{25} + 5X_{34} + 6X_{35} + 7X_{39} + 2X_{45} + 6X_{46} + 4X_{47} + 8X_{48} + 4X_{49} + 2X_{54} + 3X_{56} + 6X_{57} + 7X_{58} + 7X_{59}$

Constraints

Supply

$X_{14} + X_{15} <= 450$

$X_{24} + X_{25} <= 600$

$X_{34} + X_{35} + X_{39} <= 380$

Demand

$X_{46} + X_{56} = 300$

$X_{47} + X_{57} = 300$

$X_{48} + X_{58} = 300$

$X_{49} + X_{59} + X_{39} = 400$

Intermediate nodes

$\mathbf{X_{14} + X_{24} + X_{34}} = X_{45} + X_{46} + X_{47} + X_{48} + X_{49}$

$\mathbf{X_{15} + X_{25} + X_{35}} = X_{54} + X_{56} + X_{57} + X_{58} + X_{59}$

* **Flow in (+)**, Flow out (-) [place **center** the intermediate node]

X_{14}, X_{15}, X_{24}, X_{25}, X_{34}, X_{35}, X_{39}, X_{45}, X_{46}, X_4, X_4, X_{49}, X_{54}, X_{56}, X_{57}, X_{58}, X_{59} >= 0 (non-negativity restriction)

```
library(lpSolveAPI)
lp.model <- make.lp(0,17)
set.objfn(lp.model, c(4,7,8,5,5,6,7,2,6,4,8,4,2,3,6,7,7))
#supply
add.constraint(lp.model, rep(1,2), indices = c(1,2), "<=", 450)
add.constraint(lp.model, rep(1,2), indices = c(3,4), "<=", 600)
add.constraint(lp.model, rep(1,3), indices = c(5,6,7), "<=", 380)
#intermediate nodes
add.constraint(lp.model, c(1,1,1,-1,-1,-1,-1,-1), indices = c(1,3,5,6,7), "=", 0)
add.constraint(lp.model, c(1,1,1,-1,-1,-1,-1,-1), indices = c(2,4,8,9,10), "=", 0)
#demand
add.constraint(lp.model, rep(1,2), indices = c(9,14), "=", 300)
add.constraint(lp.model, rep(1,2), indices = c(10,15), "=", 300)
add.constraint(lp.model, rep(1,2), indices = c(11,16), "=", 300)
add.constraint(lp.model, rep(1,3), indices = c(12,17,7), "=", 400)

lp.control(lp.model, sense='min')
print(lp.model)
solve(lp.model)
get.objective(lp.model)
get.variables(lp.model)
```

Answer

Z = **5800**

Decision variables = 0, 0, 0, 0, 0, 0, 0, 0, 0, 300, 0, 400, 0, 300, 0, 300, 0

Goal Programming

Exercise 1

Company produces two popular products with home renovators – old-fashioned chandeliers and ceiling fan. Both the chandeliers and fans require a two-step production process, which as wiring and assembly. It takes about 2 hours to wire each chandelier and 3 hours to wire a ceiling fan. Final assembly of the chandeliers and fans requires 6 and 5 hours respectively. The production capability is such that 12 hours of wiring time and 30 hours of assembly time are available. The profits for each chandelier and fan are $7 and $6 respectively.

Besides, the management of the company also set up several goals of equal in priority to be achieved:

Goal 1: To produce at least a profit of $30,

Goal 2: To use all the available wiring department hours,

Goal 3: To avoid overtime in the assembly department,

Goal 4: To meet a contract requirement to produce at least 7 ceiling fans.

Formulate a goal programming model for this problem.

Solution

Decision variables

X = Number of chandeliers produced

Y = Number of ceiling fans produced

Objective function

Max Z = 7X + 6Y

Constraints

2X + 3Y <= 12

6X + 5Y <= 30

X, Y >= 0 (non-negativity restriction)

Goal 1 $\mathbf{7X + 6Y = 30 + d1^+ - d1^-}$

$7X + 6Y = \mathbf{30} + d1^+ - d1^-$

Goal 2 $\mathbf{2X + 3Y = 12 + d2^+ - d2^-}$

$2X + 3Y = \mathbf{12} + d2^+ - d2^-$

Goal 3 $\mathbf{6X + 5Y = 30 + d3^+ - d3^-}$

$6X + 5Y = \mathbf{30} + d3^+ - d3^-$

Goal 4 $\mathbf{Y = 7 + d4^+ - d4^-}$

$Y = \mathbf{7} + d4^+ - d4^-$

$$\text{Min } Z = P\,(d1^- + d2^+ + d2^- + d3^+ + d4^-)$$

Note

- Values in Red are selected minimization variables to achieve goals. Numbers in Blue are given coefficients.
- Put all min variables in P (Priority)
- Perform minimization = **1**, No need = **0**

```
library(lpSolve)
library(goalprog)
coefficient <- matrix(c(7,6,
                2,3,
                6,5,
                0,1), nrow = 4, byrow = TRUE)
target <- c(30,12,30,7)
goals <- data.frame(matrix(c(1,1,0,1,
                  2,2,1,1,
                  3,3,1,0,
                  2,4,0,1), nrow = 4, byrow = TRUE))
names(goals) <- c("objective","priority","positive","negative")
solution <- llgp(coefficient, target, goals)
llgpout(solution$tab, coefficient, target)
```

Answer

X = **2.000000e+00**, Y = **2.666667e+00**

Another style of coding

```
library(lpSolve)
library(goalprog)

coef <- matrix(c(7,6,2,3,6,5,0,1), nrow = 4, byrow = TRUE)
target <- c(30,12,30,7)
goals <- data.frame("objective"=c(1,2,3,4),"priority"=c(1,2,3,4),"positive"=c(0,1,1,0),"negative"=c(1,1,0,1))
solution <- llgp(coef, target, goals)
llgpout(solution$tab, coef, target)
```

Matrix explanation

Coefficients [X Y]

$$\begin{bmatrix} 7 & 6 \\ 2 & 3 \\ 6 & 5 \\ 0 & 1 \end{bmatrix}$$

Goals [#Objectives #Priority Positive Negative]

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 2 & 2 & 1 & 1 \\ 3 & 3 & 1 & 0 \\ 4 & 4 & 0 & 1 \end{bmatrix}$$

Integer Linear Programming

Produces only 1 and 0. 1 for positive and 0 is negative.

All the variables are restricted to be integers.

Exercise 1

The Ice-Cold Refrigerator Company is considering investing in several projects that have varying capital requirements over the next four years. Faced with limited capital each year, management would like to select the most profitable projects. The estimated net present value (net cash flow) for each project, the capital requirements, and the available capital over the four-year period is given.

	Project				
	Plane expansion	Warehouse expansion	New machinery	New product research	Total capital available
Present Value	\$90,000	\$40,000	\$10,000	\$37,000	
Year 1	\$15,000	\$10,000	\$10,000	\$15,000	\$40,000
Year 2	\$20,000	\$15,000		\$10,000	\$50,000
Year 3	\$20,000	\$20,000		\$10,000	\$40,000
Year 4	\$15,000	\$5,000	\$4,000	\$10,000	\$35,000

Solution

Decision variables

P = 1 if the plant expansion project is accepted

W = 1 if the warehouse expansion project is accepted

M = 1 if the new machinery project is accepted

R = 1 if the new product research project is accepted

Objective function

Max Z = 90000P + 40000W + 10000M + 37000R

Constraints

15000P + 10000W + 10000M + 15R <= 40000 (Year 1 capital)

20000P + 15000W + 10000R <= 50000 (Year 2 capital)

20000P + 20000W + 10000R <= 40000 (Year 3 capital)

15000P + 50000W + 40000M + 10000R <= 35000 (Year 4 capital)

P, W, M, R = 0, 1

```r
library(lpSolveAPI)
rm(lps.model)
lps.model=make.lp(0,4)
set.type(lps.model,1:4,"binary") #make lps model
set.objfn(lps.model,c(90000,40000,10000,37000))

add.constraint(lps.model,c(15000,10000,10000,15000),"<=",40000)
add.constraint(lps.model,c(20000,15000,10000,15000),"<=",50000)
add.constraint(lps.model,c(20000,20000,0,10000),"<=",40000)
add.constraint(lps.model,c(15000,5000,4000,10000),"<=",35000)

lp.control(lps.model,sense='max')
print(lps.model)
solve(lps.model)
get.variables(lps.model)
get.objective(lps.model)
```

Answer

Max Z = **140000**

Variables

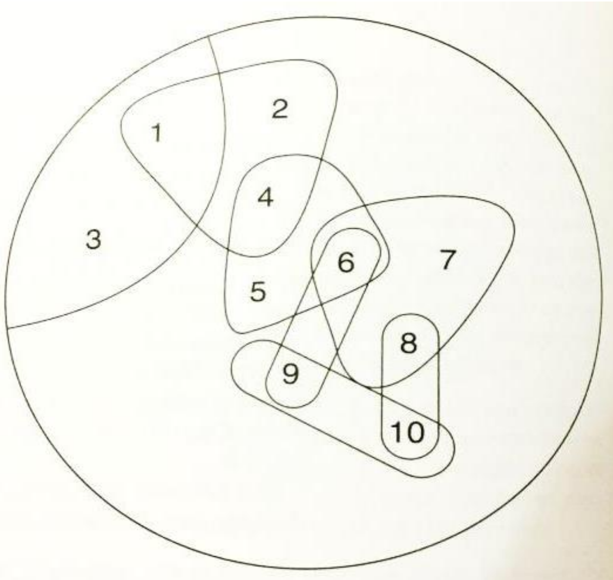
P = **1**, W = **1**, M = **1**, R = **0**

Exercise 2

Company ABC has brought out a competing grocery store chain. However, it now has too many stores near each other in certain city. In Cheras, the chain has 10 stores and it does not want any stores closer than 2 miles to each other. Following are the monthly revenue (in thousands) from each store and a map showing the general proximity of the stores. Stores within 2 miles of each other are circled.

Develop and solve an integer linear programming model to determine which stores should keep open in Cheras.

Store	Monthly revenue
1	$ 127
2	$ 83
3	$ 165
4	$ 96
5	$ 112
6	$ 88
7	$ 135
8	$ 141
9	$ 117
10	$ 94



Solution

```
library(lpSolveAPI)
rm(lps.model)
lps.model=make.lp(0,10)
set.type(lps.model, 1:10,"binary")
set.objfn(lps.model,c(127,83,165,96,112,88,135,141,117,94))

add.constraint(lps.model,rep(1,2), indices = c(1,3),"=",1)
add.constraint(lps.model,rep(1,3), indices = c(1,2,4),"=",1)
add.constraint(lps.model,rep(1,3), indices = c(4,5,6),"=",1)
add.constraint(lps.model,rep(1,3), indices = c(6,7,8),"=",1)
add.constraint(lps.model,rep(1,2), indices = c(6,9),"=",1)
add.constraint(lps.model,rep(1,2), indices = c(8,10),"=",1)
add.constraint(lps.model,rep(1,2), indices = c(9,10),"=",1)

lp.control(lps.model,sense='max')
print(lps.model)
solve(lps.model)
get.variables(lps.model)
get.objective(lps.model)
```

Answer

Max Z = **618**

Variables = **0 1 1 0 1 0 0 1 1 0**