Assessment Task 4: Problem solving task 3

Question 1

A factory produces dresses and coats for a chain of departmental stores in Victoria. The stores will accept all the production supplied by the factory. The production process includes Cutting, Sewing and Packaging, in this order. You can assume that each worker participates in one operation only (Cutting, Sewing, or Packaging). The factory employs 25 workers in the cutting department, 52 workers in the sewing department and 14 workers in the packaging department. The factory works 8 hours a day (these are productive hours). There is a daily demand for at least 120 dresses, and no specific demand for the coats. The table below gives the time requirements (in minutes) and profit per unit for the two garments to be produced.

Part a)

Explain why a linear programming model would be suitable for this case study

The pre-requisites for defining a problem as a linear programming model are: - A well-defined objective function. The objective must be sated and it can be a maximization or a minimization of costs. - There must be limited resources. There must be restrictions to the amount of resources used to attain the objective and those resources must be quantified economically and limited in supply. - Alternative cost of actions. Since a firm uses many alternatives or a combination of resources to achieve its chosen goals there must be alternative causes of actions. - Mathematical formulation. The organization objective equation and the inequalities must be mathematically expressed in order to obtain accurate means of measurement. - Linearity of the objective function and constraints. All the equations and the equalities must describe the problem in a linear form

The problem at hand, clearly fulfills all the requirements state above and therefore can use LP modelling for simulation and finding maximum profit.

Part b)

Formulate a Linear Programming model to help the management of the factory determine the optimal daily production schedule, that is, find the number of dresses and coats to be produced that would maximize the profit

Constraints: 1. There is a daily demand for at least 120 dresses

$$x_1 \geq 120$$

2. The factory employs 25 workers in the cutting department, 52 workers in the sewing department and 14 workers in the packaging department and the factory works 8 hours a day. The number of hours spent on performing different production processes should be less than the total available based on the number of hours the factory worked for multiply the number of workers available for that process.

$$\frac{25}{60}x_1 + \frac{12}{60}x_2 \leq 8 \times 25$$
$$\frac{25}{60}x_1 + \frac{55}{60}x_2 \leq 8 \times 52$$
$$\frac{15}{60}x_1 + \frac{15}{60}x_2 \leq 8 \times 14$$

Where,

$$x_1 = Number\ of\ Dresses$$
$$x_2 = Number\ of\ Coats$$

Objective:

$$\max_{x_1,x_2} \; 8x_1 + 15x_2$$

```
objective.in <- c(8, 15)

const.mat <- matrix(c(25/60, 12/60,
              25/60, 55/60,
              15/60, 15/60,
              1    , 0     ),
            ncol=2,
            byrow=TRUE)

const.rhs <- c(200, 3016, 112, 120)

constraints  <- c("<=", "<=", "<=", ">=")

solution <-  lp(direction="max",
               objective.in = objective.in,
               const.mat = const.mat,
               const.dir = constraints,
               const.rhs = const.rhs,
               all.int = T)

best.solution <- solution$solution
names(best.solution) <- c("Number of Dresses", "Number of Coats")
print(best.solution)
```

```
Number of Dresses   Number of Coats
              120               328
```

```
cat("\n\nTotal Profit:", solution$objval, sep=" ")
```

```
Total Profit: 5880
```

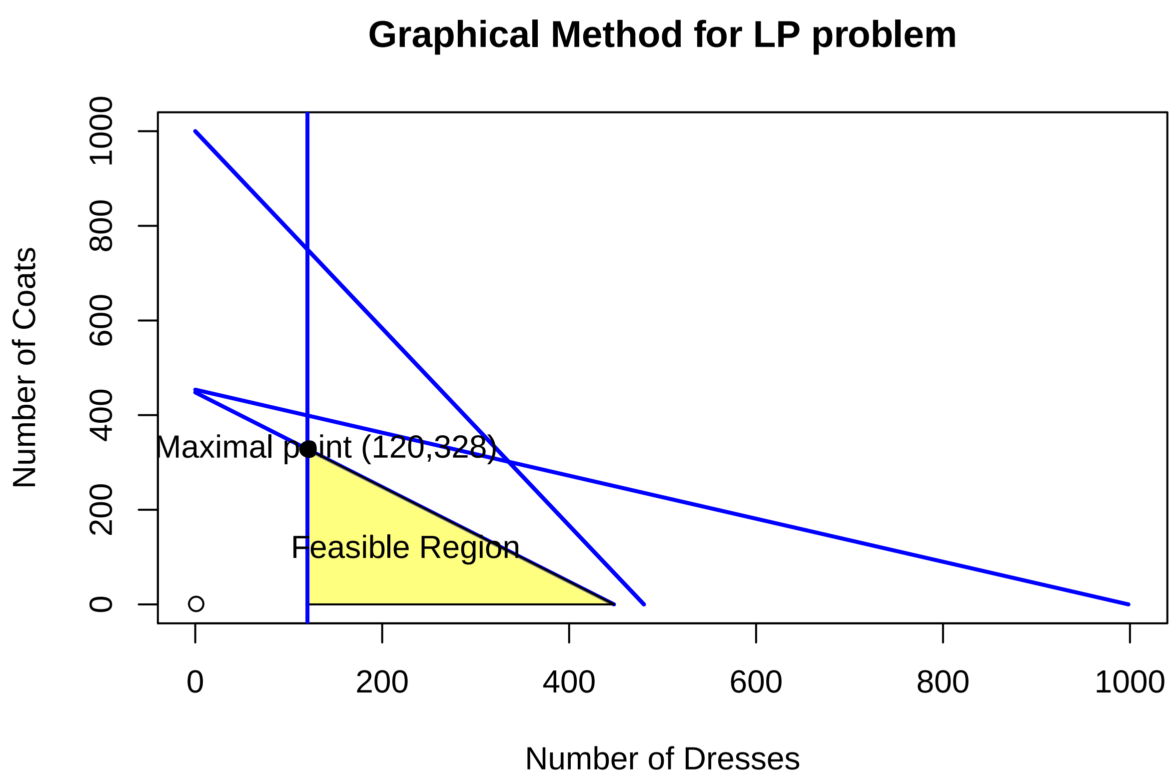
Part c)

Use the graphical method to find the solution solution. Show the feasible region and the optimal solution on the graph. Annotate your graph. What is the optimum profit?

```
x <- c(120,448,120)
y <- c(328,0,0)

plot(1,xlim=c(0,1000),
     ylim=c(0,1000),
     xlab="Number of Dresses",
     ylab="Number of Coats",
     main="Graphical Method for LP problem")
lines(c(480,0),
      c(0,1000),
      lwd=2,
      col="blue")
lines(c(480,0),
      c(0,1000),
      lwd=2,
```

```
       col="blue")
lines(c(448,0),
      c(0,448),
      lwd=2,
      col="blue")
lines(c(998.4, 0),
      c(0, 453.82),
      lwd=2,
      col="blue")
polygon(x,
        y,
        col=rgb(1,1,0,0.5))
abline(v=120,
       lwd=2,
       col="blue")
points(120,
       328,
       pch=19)
text(140,
     328,
     "Maximal point (120,328)")
text(225,
     115,
     "Feasible Region")
```



Part d)

Find a range for the profit ($) of a dress that can be changed without affecting the optimum solution obtained above.

```
profit <- 8
while (TRUE) {

  profit <- profit + 1

  objective.in <- c(profit, 15)

  const.mat <- matrix(c(25/60, 12/60,
                25/60, 55/60,
                15/60, 15/60,
                1    , 0     ),
              nrow=4,
              byrow=TRUE)

  const.rhs <- c(8*25, 8*52, 8*14, 120)

  constraints  <- c("<=", "<=", "<=", ">=")

  solution <-  lp(direction="max",
                 objective.in = objective.in,
                 const.mat = const.mat,
                 const.dir = constraints,
                 const.rhs = const.rhs,
                 all.int = T)

  if (sum(solution$solution==c(120, 328))==2) {
    next
  } else {
    break
  }
}
cat("",
    "8 - ",
    profit - 1,
    sep="$")
```

```
$8 - $14
```

Question 2

Part a)

Choose appropriate decision variables. Formulate a linear programming (LP) mod-el to determine the optimal production mix of cereals that maximises the profit, while satisfying the constraints. Then compute the associated amounts of ingredients for each cereal.

The *profit* is given as:

$$Profit = Sales\ Price - Production\ Cost - Purchase\ Price$$

Objective:

$$\max_{x_1,x_2,x_3} \; Profit$$

Where,

$$x_1 = Amount\ of\ Cereal\ A$$
$$x_2 = Amount\ of\ Cereal\ B$$
$$x_3 = Amount\ of\ Cereal\ C$$

Constraints:

1. Minimum Demand (in boxes)

$$x_1 \geq 1000$$
$$x_2 \geq 800$$
$$x_3 \geq 750$$

2. Maximum availability of Ingredients (in boxes)

$$0.80x_1 + 0.60x_2 + 0.45x_3 \leq 10 \times \frac{1000}{2}$$
$$0.10x_1 + 0.25x_2 + 0.15x_3 \leq 5 \times \frac{1000}{2}$$
$$0.05x_1 + 0.05x_2 + 0.10x_3 \leq 2 \times \frac{1000}{2}$$
$$0.05x_1 + 0.10x_2 + 0.30x_3 \leq 2 \times \frac{1000}{2}$$

Now, re-writing the profit in terms of the decision variables that is the quantity/amount of all cereal types,

$$Profit = 2.60x_1 + 2.30x_2 + 3.20x_3 - \left(\frac{4.20}{\frac{1000}{2}}x_1 + \frac{2.60}{\frac{1000}{2}}x_2 + \frac{3.00}{\frac{1000}{2}}x_3\right)$$
$$- \left(0.80\frac{100}{\frac{1000}{2}} + 0.10\frac{90}{\frac{1000}{2}} + 0.05\frac{110}{\frac{1000}{2}} + 0.05\frac{200}{\frac{1000}{2}}\right)x_1$$
$$- \left(0.60\frac{100}{\frac{1000}{2}} + 0.25\frac{90}{\frac{1000}{2}} + 0.05\frac{110}{\frac{1000}{2}} + 0.10\frac{200}{\frac{1000}{2}}\right)x_2$$
$$- \left(0.45\frac{100}{\frac{1000}{2}} + 0.15\frac{90}{\frac{1000}{2}} + 0.10\frac{110}{\frac{1000}{2}} + 0.30\frac{200}{\frac{1000}{2}}\right)x_3$$

Or, *Objective:*

$$\max_{x_1,x_2,x_3} \; 2.383x_1 + 2.079x_2 + 2.935x_3$$

Part b)

Find the optimal solution using R/R studio.

```r
objective.in <- c(2.383, 2.079, 2.935)

const.mat <- matrix(c(1.00, 0.00, 0.00,
              0.00, 1.00, 0.00,
              0.00, 0.00, 1.00,
              0.80, 0.60, 0.45,
              0.10, 0.25, 0.15,
              0.05, 0.05, 0.10,
              0.05, 0.10, 0.30),
            nrow=7,
            byrow=TRUE)

const.rhs <- c(1000, 800, 750, 5000, 2500, 1000, 1000)

constraints  <- c(rep(">=", 3), rep("<=", 4))

solution <-  lp(direction="max",
               objective.in = objective.in,
               const.mat = const.mat,
               const.dir = constraints,
               const.rhs = const.rhs,
               all.int = T)

best.solution <- solution$solution
names(best.solution) <- c("Amount of Cereal A",
                          "Amount of Cereal B",
                          "Amount of Cereal C")
print(best.solution)
```

```
Amount of Cereal A Amount of Cereal B Amount of Cereal C
              4326                808               2343
```

```r
cat("\n\nTotal Profit:", solution$objval, sep=" ")
```

```
Total Profit: 18865.4
```

Question 3

John and Alice are playing a game by putting chips in two piles (each player has two piles P1 and P2), respectively. Alice has 4 chips and John has 5 chips. Each player place his/her chips in his/her two piles, then compare the number of chips in his/her two piles with that of the other player's two piles. Note that once a chip is placed in one pile it cannot be moved to another pile. There are four comparisons including John's P1 vs Alice's P1, John's P1 vs Alice's P2, John's P2 vs Alice's P1, and John's P2 vs Alice's P2. For each comparison, the player with more chips in the pile will score 1 point (the opponent will loose 1 point). If the number of chips are the same in the two piles, then nobody will score any points from this comparison. The final score of the game is the sum score over the four comparisons. For example, if Alice puts 4 and 0 chips in her P1 and P2, John puts 1 and 4 chips in his P1 and P2, respectively. Then Alice will get 1 (4 vs 1) + 0 (4 vs 4) - 1 (0 vs 1) - 1 (0 vs 4) = -1 as her final score, and John will get his final score of 1.

Part a)

Give reasons why/how this game can be described as two-players-zero-sum game

In game theory and economic theory, a zero-sum game is a mathematical representation of a situation in which each participant's gain or loss of utility is exactly balanced by the losses or gains of the utility of the other participants. Therefore, this game can be described as the zero-sum game for two-players.

If a two-person zero-sum game has saddle points, the best each player can do (assuming both to be rational) is to choose the strategy (i.e., the row or column of the game matrix) which contains a saddle point.

Part b)

Formulate the payoff matrix for the game.

```
payoff <- data.frame(
  "1" = c(-1,-1,-1,-1,0),
  "2" = c(0,-2,-2,0,0),
  "3" = c(-1,-1,-1,-1,0),
  "4" = c(-1,0,0,-1,0)
)
colnames(payoff) <- 1:4
row.names(payoff) <- 1:5
payoff
```

```
   1  2  3  4
1 -1  0 -1 -1
2 -1 -2 -1  0
3 -1 -2 -1  0
4 -1  0 -1 -1
5  0  0  0  0
```

Part c)

Explain what is a saddle point. Verify: does the game have a saddle point?

The joint choice $(row_{number}, column_{number})$ appears as a sort of equilibrium or "balance of power" when the payoff for both the plsyers corresponding to that cell is equal. The payoff in that entry is the best that either of the players can do, given that he is playing against a rational opponent. An entry such as $(row_{number}, column_{number})$ is called a saddle point of the game matrix. The term is suggestive. The center of a saddle is the lowest point on the horse's back in the horse's longitudinal plane (i.e., as one moves from front to back) and at the same time the highest point in the plane perpendicular to the horse's motion (i.e., as one slides from side to side). Hence the saddle point is at the same time a minimum and a maximum. The term "minimax" (or "maximin"), frequently used in game theory, is in this context synonymous with saddle point.

Part d)

Construct a linear programming model for this game;

Constraints:

$$x_1, x_2, x_3, x_4 \geq 0$$

$$for\ i\ in\ 1:5 \Rightarrow \sum_{j=1}^{4} p\left[i, j\right] * x\left[j\right] <= 1$$

And,

$$p = Payoff\ Matrix$$

Objective:

$$\max_{x_1, x_2, x_3, x_4} x_1 + x_2 + x_3 + x_4$$

Part e)

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

```
objective.in <- c(1, 1, 1, 1)

const.mat <- matrix(rbind(diag(4),
                  as.matrix(payoff)),
            nrow=9)

const.rhs <- c(rep(0, 4), rep(1, 5))

constraints  <- c(rep(">=", 4), rep("<=", 5))
```

Part f)

Solve the game for Alice using the linear programming model you constructed in part (c). Interpret your solution

```
solution <-  lp(direction="max",
               objective.in = objective.in,
               const.mat = const.mat,
               const.dir = constraints,
               const.rhs = const.rhs,
               all.int = T)

best.solution <- solution$solution
names(best.solution) <- c("1 Chip", "2 Chips", "3 Chips", "4 Chips")
print(best.solution)
```

```
 1 Chip 2 Chips 3 Chips 4 Chips
      0       0       0       0
```

```
cat("Objective:", solution$objval, sep=" ")
```

```
Objective: 0
```