BT1101

Lab 9: Linear Optimisation

Optimisation

A component of business analytics that uses optimisation to identify the best alternatives to minimise or maximise some objective, given a set of constraints.

The process of selecting values of decision variables that minimise or maximise some quantity of interest.

- Linear
- Integer

Key Terms

Objective function

Our goal of linear optimization is to maximize (or minimize) an objective function which is a linear function of both our decision variables x, given parameters θ.

Constraints

The restrictions or limitations on the decision variables. They usually limit the value of the decision variables.

Binding & non-binding

Non-negativity

For some linear programs, the decision variables should take non-negative values. This means the values for decision variables should be greater than or equal to 0

Decision variables

The variables controlled by decision maker. For example: production quantity, goods to consume, stocks to buy in, etc.

Shadow price

The marginal change in the optimal objective function value that occurs if the right-hand side of a constraint is changed.

Feasible soln

Any solution that satisfies all the given constraints.

Feasible region

The set or region of decision variable satisfying all constraints

Installing and loading packages

```
# load required packages
# install any package below if it's first time loaded in your computer.
library(lpSolve)
```

Part One: Lab Session Completion and Discussion

Question 1

The examples we discussed in lecture were all maximization problems (specifically, to maximize profit). In this question we shall explore minimization.

FunToys is famous for three types of toys: Cars, Animals, and Robots. Each year, near the holiday season, it receives large bulk orders for these items. To meet these orders, FunToys operates three small toy-making factories, A, B and C.

- Factory A costs $1000 per day to operate, and can produce 30 cars, 20 animals and 30 robots per day.
- Factory B costs $1200 per day to operate, and can produce 40 cars, 50 animals and 10 robots per day.
- Factory C costs $1500 per day to operate, and can produce 50 cars, 40 animals and 15 robots per day.

This Christmas, FunToys is required to deliver 5000 cars, 3000 animals and 2500 robots. You are tasked with finding out what is the most cost-efficient way to meet the order.

(1a) Formulating the optimisation problem

- First, write down what you want to minimise.
- Second, write down your decision variables. What are you actually choosing?
- Third, write your objective function in terms of your decision variables.
- Fourth, write down the constraints: what are the contractual requirements you need to fulfil. What other constraints are there? Write them down in terms of your decision variables.
- Summarise them nicely in a table.

Minimize total cost using decision variables X_1, X_2, X_3 = number of days to run A, B, C respectively.	**Cost = 1000 X_1 + 1200 X_2 + 1500 X_3**
Subject to	
Contract for cars	$30X_1 + 40X_2 + 50X_3 \geq 5000$
Contract for animals	$20X_1 + 50X_2 + 40X_3 \geq 3000$
Contract for robots	$30X_1 + 10X_2 + 15X_3 \geq 2500$
Non-Negativity Constraint 1	$X_1 + \quad + \quad \geq 0$
Non-Negativity Constraint 2	$\quad + X_2 + \quad \geq 0$
Non-Negativity Constraint 3	$\quad + \quad + X_3 \geq 0$

(1b)

Write code to solve this optimisation problem. Report the optimal solution, and the value of the objective function at that solution. Interpret the solution: what do these numbers mean?

```
# Toy factory

#defining parameters
objective.fn <- c(1000, 1200, 1500)
const.mat <- matrix(c(30, 40, 50, 20, 50, 40, 30, 10, 15) , ncol=3 , byrow=TRUE)
const.dir <- c(">=", ">=", ">=")
const.rhs <- c(5000, 3000, 2500)

#solving model
lp.solution <- lp("min", objective.fn, const.mat, const.dir, const.rhs, compute.sens=TRUE)
lp.solution$solution #decision variables values
```

```
## [1] 47.61905  0.00000 71.42857
```

```
lp.solution
```

```
## Success: the objective function is 154761.9
```

Optimal solution is: X1=47.6, X2=0.00, X3=71.43. That is, run Factory A for 47.6 days, Factory B for 0 days and Factory C for 71.4 days. The Minimum cost is $154761.90

With the current constraints, the optimal solution involves not operating Factory B at all.

(1c)

What if we impose an additional constraint that FunToys only has 60 days to complete the order? (Note that we can run all three factories *simultaneously*).

What happens now?

Re-produce a new table summarizing the optimisation problem (including the existing and new constraints) and write R code to solve it. What is the new solution, and what is the objective function value?

(1c)

Minimize total cost using decision variables X_1, X_2, X_3 = number of days to run A, B, C respectively.	**Cost = 1000 X_1 + 1200 X_2 + 1500 X_3**
Subject to	
Contract for cars	$30X_1 + 40X_2 + 50X_3 \geq 5000$
Contract for animals	$20X_1 + 50X_2 + 40X_3 \geq 3000$
Contract for robots	$30X_1 + 10X_2 + 15X_3 \geq 2500$
Time Constraint 1	$X_1 + \quad + \quad \leq 60$
Time Constraint 2	$+ X_2 + \quad \leq 60$
Time Constraint 3	$+ \quad + X_3 \leq 60$
Non-Negativity Constraint 1	$X_1 + \quad + \quad \geq 0$
Non-Negativity Constraint 2	$+ X_2 + \quad \geq 0$
Non-Negativity Constraint 3	$+ \quad + X_3 \geq 0$

(1c)

```
# Toy factory

#defining parameters
objective.fn <- c(1000, 1200, 1500)
const.mat <- matrix(c(30, 40, 50, 20, 50, 40, 30, 10, 15, 1, 0, 0, 0, 1, 0, 0, 0, 1) , ncol=3 , byrow=TRUE)
const.dir <- c(">=", ">=", ">=", "<=", "<=", "<=")
const.rhs <- c(5000, 3000, 2500, 60, 60, 60)

#solving model
lp.solution <- lp("min", objective.fn, const.mat, const.dir, const.rhs, compute.sens=TRUE)
lp.solution$solution #decision variables values
```

```
## [1] 48.88889 13.33333 60.00000
```

```
lp.solution
```

```
## Success: the objective function is 154888.9
```

The solution is now: $X_1=48.89$, $X_2=13.33$, $X_3=60$. That is, run Factory A for 48.89 days, Factory B for 13.33 days and Factory C for 60 days. The Minimum cost is $154888.90

(1d)

For the solution in 1c, which of the constraints are binding, and which are non-binding?

```
lp.solution$solution #decision variables values
```

```
## [1] 48.88889 13.33333 60.00000
```

```
# Toy factory
num_cars <- sum(lp.solution$solution*c(30, 40, 50)) # 5000
num_animals <- sum(lp.solution$solution*c(20, 50, 40)) # 4044
num_robots <- sum(lp.solution$solution*c(30, 10, 15)) # 2500
```

`lp.solution$solution` is the vector containing the optimal values of our decision variables — the number of days we want to run Factory A, B and C.

- Cars = 30X1 + 40X2 + 50X3 = 5000 bound by 5000
- Animals = 20X1 + 50X2 + 40X3 = 4044.4444444 > 3000, not bound by 3000
- Robots = 30X1 + 10X2 + 15X3 = 2500 bound by 2500

Binding constraints:

- Contract for cars
- Contract for robots
- Time constraint on Factory C

Nonbinding constraints:

- Contract for animals
- Time constraints on Factories B and C
- All non-negativity constraints

(1e)

Using your solution in 1c, print out the Shadow Prices. Interpret these values — make sure you can explain why each shadow price is zero or why it is positive/negative! Your answer from part d) should also help you explain.

```
lp.solution$duals
```

```
## [1]  28.888889   0.000000   4.444444   0.000000   0.000000 -11.111111   0.000000
## [8]   0.000000   0.000000
```

Minimize total cost using decision variables X_1, X_2, X_3 = number of days to run A, B, C respectively.	**Cost = 1000 X_1 + 1200 X_2 + 1500 X_3**
Subject to	
Contract for cars	$30X_1 + 40X_2 + 50X_3 \geq 5000$
Contract for animals	$20X_1 + 50X_2 + 40X_3 \geq 3000$
Contract for robots	$30X_1 + 10X_2 + 15X_3 \geq 2500$
Non-Negativity Constraint 1	$X_1 + \quad + \quad \geq 0$
Non-Negativity Constraint 2	$+ X_2 + \quad \geq 0$
Non-Negativity Constraint 3	$+ \quad + X_3 \geq 0$

Recall the definition for shadow price: the **marginal change** in the optimal objective function value when the RHS of a constraint is increased by 1.

(1e)

Shadow Prices in order:

Contract for cars : $30X1 + 40X2 + 50X3 \geq 5000$ is BINDING so shadow price is positive
Contract for animals : $20X1 + 50X2 + 40X3 \geq 3000$ is non-binding so shadow price is zero.
Contract for robots : $30X1 + 10X2 + 15X3 \geq 2500$ is BINDING so shadow price is positive.

Time Constraint 1 : X1 + + ≤ 60 is non-binding so shadow price is zero.
Time Constraint 2 : + X2 + ≤ 60 is non-binding so shadow price is zero.
Time Constraint 3 : + + X3 ≤ 60 is BINDING. Shadow price is **negative** because if we allow Factory 3 (the most cost-efficient factory) to have more than 60 days, we can reduce our overall cost.

All of the non-negativity constraints are non-binding so the last three shadow prices are zero.