Integer & network Linear Programming Assignment

Cesareo Giacomo 805716

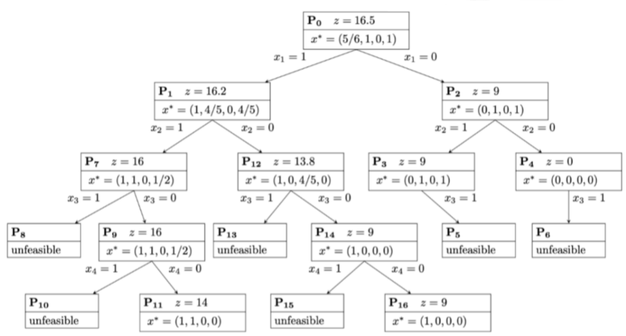
Contents

Problem 1

Consider the following ILP:

$$
\begin{aligned}
&\max \quad 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
&\text{s.t.} \\
&6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10, \\
&x_3 + x_4 \leq 1, \\
&-x_1 + x_3 \leq 0, \\
&-x_2 + x_4 \leq 0 \\
&x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{aligned}
$$

The following tree represents the solutions of all possible relaxations of the problem in which no sub-problem has been excluded (fathoming).



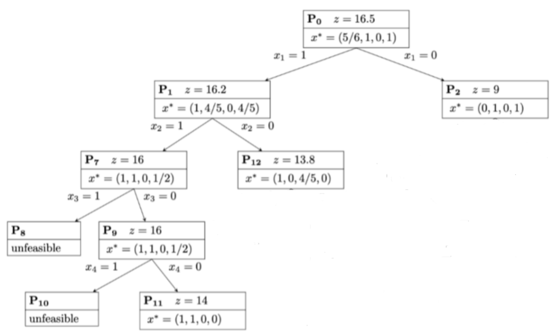
Suppose that the Branch and Bound (BB) algorithm applies to this problem. Also, let's suppose that the algorithm visits the sub-problems in the following order P0 , P1 , . . . , P16. Clearly, the algorithm does not visit all nodes.

1) Determine the nodes that will be visited by the BB algorithm and for each of them get the upper and lower limit deduced by the algorithm in the execution.

The algorithm starts from P_0 where the only non integer variable is X_1, since in an integer problem we can not accept variables solutions unless all of them are integer, the Branch and Bound algorithm (BB) will split the starting problem in two branches: P_1 by considering $X_1 = 1$ and P_2 in the case where $X_1 = 0$.
The Upper bound (Ub) and the Lower bound (Lb) for each node are computed in the following way: the Ub corresponds to the value of z that is the relaxed solution of the problem, instead the Lb is computed by solving z rounding down the non integer values for the decision variables to the closest integer value (0 in this case), therefore with the Lb is presented

the pessimistic minimal solution for the problem.
As presented above the first node explore by the BB is P_0 where $Ub = 16.5$ and $Lb = 9$.
Let's consider now P_2, where all the coefficients fot the decision variables are now integer, since $Ub = Lb = 9$ meaning that all the sub-poblem starting from P_2 will not find a better solution and it is completely worthless for the algorithm to procede further, then 9 can be considered as a possible canditate for the optimal solution of the problem.
Considering P_1, in the left side of the graph, it present $Ub = 16.2$ and $Lb = 9$, right after the BB will visit P_7 having the same Lb computed in the previous node. At this point P_8 is unfasible then the algorithm will look for annother optimal solution in P_9: this node present $Ub = 16$ and $Lb = 14$, therefore being $Lb_{(P_9)} > Lb_{(P_2)}$ the algorith will no longer consider $P_2 = 9$ as a candidate optimal solution for the problem because the algorith will surely find a better solution for the problem. P_{10} is unfeasible too then it will not be considered by BB, while P_{11} with $Ub = Lb = 14$ and all the coefficients found are integer so for now this can be considered as a canditate for an optimal solution.
Looking at P_{12} the $Ub = 13.8$, so being $Ub_{(P_{12})} < Ub_{(P_{11})}$ the algorith will stop here because there will not be a better solution for the problem. So the algorithm will find the optimal solution in the sub-problem P_{11} having $z^* = 14$ and $x^* = (1, 1, 0, 0)$

In the Immage below is represented the path chosen by the BB algorithm.



2) Solve the problem with an ILP solver and check the value of the objective function matches the one found at point 1.

```r
# Import packages.
library(lpSolve)
library(lpSolveAPI)
library(DiagrammeR)
```

```r
# Initializing the lp model with 0 constrains and 4 decision variables.
model <- make.lp(0, 4)

# Maximization problem.
lp.control(model, sense = 'max')
```

```r
# Objective function.
set.objfn(model, c(9, 5, 6, 4))

# Constrains.
add.constraint(model, c(6, 3, 5, 2), "<=", 10)
add.constraint(model, c(1, 1), "<=", 1, c(3, 4))
add.constraint(model, c(-1, 1), "<=", 0, c(1, 3))
add.constraint(model, c(-1, 1), "<=", 0, c(2, 4))

# Set decision variables value to be binary integer {0, 1}.
set.type(model, c(1:4), "binary")
```

```r
# 0 means that the model was solved correcty.
solve(model)
```

```
## [1] 0
```

```r
get.objective(model)
```

```
## [1] 14
```

```r
get.variables(model)
```

```
## [1] 1 1 0 0
```

As expected the maximazed value for the problem is 14 and the coefficients found are the same seen in the above graph.

Problem 2

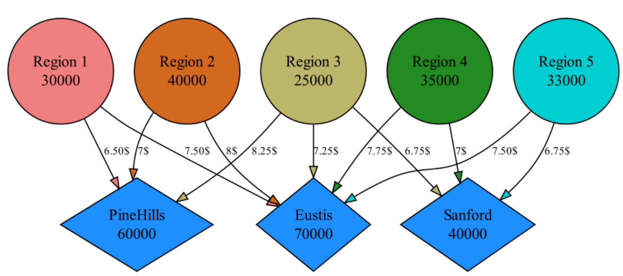
SunNet is a residential Internet Service Provider (ISP) in the central Florida area. Presently, the company operates one centralized facility that all of its clients call into for Internet access.

To improve service, the company is planning to open three satellite offices in the cities of Pine Hills, Eustis, and Sanford. The company has identified five different regions to be serviced by these three offices. The following table summarizes the number of customers in each region, the service capacity at each office, and the monthly average cost per customer for providing the service to each region from each office. Table entries of "n.a." indicate infeasible region-to-service center combinations.

SunNet would like to determine how many customers from each region to assign to each service center to minimize the total cost.

Region	Pine Hills	Eustis	Sanford	Customers
1	$6.50	$7.50	n.a.	30,000
2	$7.00	$8.00	n.a.	40,000
3	$8.25	$7.25	$6.75	25,000
4	n.a.	$7.75	$7.00	35,000
5	n.a.	$7.50	$6.75	33,000
Capacity	60,000	70,000	40,000	

1) Draw a network flow model to represent this problem.



The goal about the problem is to find the optimal way to assign the customers between the $i-th$ office and the $j-th$ region. The decision variable can be expained in this way.

Region	Pine Hills	Eustis	Sanford	Customers
1	$x_{1,1}$	$x_{2,1}$	n.a.	$30,000$
2	$x_{1,2}$	$x_{2,2}$	n.a.	$40,000$
3	$x_{1,3}$	$x_{2,3}$	$x_{3,3}$	$25,000$
4	n.a.	$x_{2,4}$	$x_{3,4}$	$35,000$
5	n.a.	$x_{2,5}$	$x_{3,5}$	$33,000$
Capacity	$60,000$	$70,000$	$40,000$	

The goal is to minimize the average cost of the services for each customer. The objective function is presented below since the average costs coefficients $c_{i,j}$ are given in the first table:

$$MIN : \sum_{i=1}^{3}\sum_{j=1}^{5} c_{i,j}x_{i,j}$$

2) Implement your model and solve it.

```
model <- make.lp(0, 11)
lp.control(model, sense = 'min')
```

```
# Objective function.
set.objfn(model, c(6.5, 7.5, 7, 8, 8.25, 7.25, 6.75, 7.75, 7, 7.5, 6.75))

# Capacity constrains.
add.constraint(model, c(1, 1, 1), "<=", 60000, c(1, 3, 5))
add.constraint(model, c(1, 1, 1, 1, 1), "<=", 70000, c(2, 4, 6, 8, 10))
add.constraint(model, c(1, 1, 1), "<=", 40000, c(7, 9, 11))

# Customer cosntrains.
add.constraint(model, c(-1, -1), "<=", -30000, c(1, 2))
add.constraint(model, c(-1, -1), "<=", -40000, c(3, 4))
add.constraint(model, c(-1, -1, -1), "<=", -25000, c(5, 6, 7))
add.constraint(model, c(-1, -1), "<=", -35000, c(8, 9))
add.constraint(model, c(-1, -1), "<=", -33000, c(10, 11))

# Integer programming
set.type(model, c(1:11), "integer")
solve(model)
```

```
## [1] 0
```

3) What is the optimal solution?

```
get.objective(model)
```

```
## [1] 1155000
```

```r
get.variables(model)
```

```
##  [1] 20000 10000 40000     0     0 25000     0     0 35000 28000  5000
```

```r
get.primal.solution(model)[2:8]
```

```
## [1]  60000  63000  40000 -30000 -40000 -25000 -35000
```

From the outputs it is possible to discover that the minimized cost is equal to 1155000\$, and the values for each variable of the objective function can be seen below.

$x_{1,1}$	$x_{2,1}$	$x_{1,2}$	$x_{2,2}$	$x_{1,3}$	$x_{2,3}$	$x_{3,3}$	$x_{2,4}$	$x_{3,4}$	$x_{2,5}$	$x_{3,5}$
20000	10000	40000	0	0	25000	0	0	35000	28000	5000

In the last output it's easy to notice that all the right hand side of the constraints are maximized except for the second capacity constrain where over 70000 only 63000 ($x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5}$) were selected.