ETM540 - Homework #2

Mala Daryanani

October 10, 2018

1. Ex:Transportation Problem

Four manufacturing plants are supplying material for distributors in four regions. The four supply plants are located in Chicago, Beaverton, Eugene, and Dallas. The four distributors are in PDX (Portland), SEA (Seattle), MSP(Minneapolis), and ATL (Atlanta). Each manufacturing plant has a maximum amount that they can produce. For example, Chicago can produce at most 500. Similarly, the PDX region can handle at most 700 units. The cost to transport from Dallas to MSP is three times as high as the cost from Dallas to Atlanta.

Node	PDX	SEA	MSP	ATL	**Supply**
Chicago	20	21	8	12	**500**
Beaverton	6	7	18	24	**500**
Eugene	8	10	22	28	**500**
Dallas	16	26	15	5	**600**
Capacity	**700**	**500**	**500**	**600**	

Exericise:Formulate an Explict Transportation Model Formulate an explicit model for the above application that solves this transporation problem to find the lowest cost way of transporting as much as product as we can to distributors. Hint: You might choose to define variables based on the first letter of source and destination so XCP is the amount to ship from Chicago to PDX.

Implement and solve the model using ompr. Be sure to discuss the solution as to why it makes it sense.

Solution 1:

For this given problem, the total supply is less than the demand capacity $\sum_i S_i < \sum_j D_j$. Given the situation is *Supply Constrainted*, we will consider the below constraints:

$\sum_j x_{i,j} = S_i$

$\sum_i x_{i,j} \leq D_j$

Refer to the below table:

If...	Then Situation is:	Source Constraints	Demand Constraints
$\sum_i S_i < \sum_j D_j$	Supply Constrained	$\sum_j x_{i,j} = S_i$	$\sum_i x_{i,j} \leq D_j$
$\sum_i S_i > \sum_j D_j$	Demand Constrained	$\sum_j x_{i,j} \leq S_i$	$\sum_i x_{i,j} = D_j$
$\sum_i S_i = \sum_j D_j$	Balanced	$\sum_j x_{i,j} = S_i$	$\sum_i x_{i,j} = D_j$

Mathematical Representation for the given transportation problem:

$$
\begin{aligned}
\text{Minimize } & \sum_i \sum_j C_{i,j} x_{i,j} \\
\text{subject to } & \sum_i x_{i,j} \leq D_j \text{ , } j = 1,2,3,4 \\
& \sum_j x_{i,j} = S_i \text{ , } i = 1,2,3,4 \\
& x_{i,j} \geq 0 \; \forall \; i,j
\end{aligned}
\quad (\#Source\ Constraint - Transportation)
$$

Solving the problem in R:

```r
library (pander, quietly = TRUE)   # Used for nicely formatted tables
library (magrittr, quietly = TRUE) # Used for pipes/dplyr
library (dplyr, quietly = TRUE)    # Data management
library (ROI, quietly = TRUE)      # R Optimization Interface
library (ROI.plugin.glpk, quietly = TRUE) # Plugin for solving
library (ompr, quietly = TRUE)     # Allows specifying model algebraically
library (ompr.roi, quietly = TRUE) # Glue for ompr to solve with ROI
```

```r
MinCostModel0 <- MIPModel() %>%

  add_variable(XCP, type = "continuous", lb=0)%>%
  add_variable(XCS, type = "continuous", lb=0)%>%
  add_variable(XCM, type = "continuous", lb=0)%>%
  add_variable(XCA, type = "continuous", lb=0)%>%

  add_variable(XBP, type = "continuous", lb=0)%>%
  add_variable(XBS, type = "continuous", lb=0)%>%
  add_variable(XBM, type = "continuous", lb=0)%>%
  add_variable(XBA, type = "continuous", lb=0)%>%

  add_variable(XEP, type = "continuous", lb=0)%>%
  add_variable(XES, type = "continuous", lb=0)%>%
  add_variable(XEM, type = "continuous", lb=0)%>%
  add_variable(XEA, type = "continuous", lb=0)%>%

  add_variable(XDP, type = "continuous", lb=0)%>%
  add_variable(XDS, type = "continuous", lb=0)%>%
  add_variable(XDM, type = "continuous", lb=0)%>%
  add_variable(XDA, type = "continuous", lb=0)%>%

  set_objective(20*XCP + 21*XCS + 8*XCM + 12*XCA +
                6*XBP + 7*XBS + 18*XBM + 24*XBA +
                8*XEP + 10*XES + 22*XEM + 28*XEA +
                16*XDP + 26*XDS + 15*XDM + 5*XDA, "min") %>%

  add_constraint(XCP + XCS + XCM + XCA == 500) %>%
  add_constraint(XBP + XBS + XBM + XBA == 500) %>%
  add_constraint(XEP + XES + XEM + XEA == 500) %>%
  add_constraint(XDP + XDS + XDM + XDA == 600) %>%
```

```r
  add_constraint(XCP + XBP + XEP + XDP <= 700) %>%
  add_constraint(XCS + XBS + XES + XDS <= 500) %>%
  add_constraint(XCM + XBM + XEM + XDM <= 500) %>%
  add_constraint(XCA + XBA + XEA + XDA <= 600) %>%

solve_model(with_ROI(solver = "glpk"))
MinCostModel0
```

```
## Status: optimal
## Objective value: 14300
```

```r
results.unitsupply0 <- MinCostModel0$solution
pander(results.unitsupply0,
       caption ="Units to supply from Source to Destination for Minimum cost")
```

Table 3: Table continues below

XBA	XBM	XBP	XBS	XCA	XCM	XCP	XCS	XDA	XDM	XDP	XDS	XEA
0	0	200	300	0	500	0	0	600	0	0	0	0

XEM	XEP	XES
0	500	0

Interpretation of the model:

The cost of transporting units from each source to the destination depends on the distance in-between them. If the suppliers are located far from the distributors, the cost of transportation will increase significantly resulting into more money to the company.

Refering to the table above, the quantity of material to be supplied is proportionally distributed based on the minimum distance between the suppliers and distributors. For example, Dallas to Atlanta is the shortest distance to cover with the cost of $6/unit - this explains why the major distribution of 600 units are being supplied this way. Alternately, if we observe the longest route Eugene to Minneapolis ($22/unit), the model recommends if anything supplied on this route will definately cost more to the company and so supply quantity=0.

2. Ex: Formulate a Generalized Transportation Model

Formulate a generalized model for the above application that solves this transporation problem to find the lowest cost way of transporting as much as product as we can to distributors. Implement and solve the model using ompr. Be sure to discuss the solution as to why it makes it sense.

Solution 2:

let's consider the below terms to generalize the transportation model:

$x_{i,j}$: the amount of product to ship from node i to node j

$C_{i,j}$: the cost per unit to ship from node i to node j

S_i, D_j: the supply available from each supply node and the maximum demand that can be accommodate from each destination node respectively

Since this problem is supply constraint, we need to ensure that the we don't supply more than the capacity from each supply node.

Mathematical Representation for generalized transportation problem:

$$\begin{split}
\text{Minimize } & \sum_i \sum_j C_{i,j} x_{i,j} \\
\text{subject to } & \sum_i x_{i,j} \leq D_j \; \forall \; j \\
& \sum_j x_{i,j} = S_i \; \forall \; i \\
& x_{i,j} \geq 0 \; \forall \; i, j
\end{split} \quad (\#Source\ Constraint - Transportation)$$

Solving the problem in R:

```
library (pander, quietly = TRUE)   # Used for nicely formatted tables
library (magrittr, quietly = TRUE) # Used for pipes/dplyr
library (dplyr, quietly = TRUE)    # Data management
library (ROI, quietly = TRUE)      # R Optimization Interface
library (ROI.plugin.glpk, quietly = TRUE) # Plugin for solving
library (ompr, quietly = TRUE)     # Allows specifying model algebraically
library (ompr.roi, quietly = TRUE) # Glue for ompr to solve with ROI
library(pander)
```

```
jDestCount <- 4
iSourceCount <- 4

jDestNames <- lapply(list(rep("Dest",jDestCount)),paste0,1:jDestCount)
iSourceNames<- lapply(list(rep("Source",iSourceCount)),paste0,1:iSourceCount)
UnitCost <- matrix(c( 20, 6, 8, 16,
                         21, 7, 10, 26,
                         8, 18, 22, 15,
                         12, 24,28, 5),
              ncol=jDestCount,dimnames=c(iSourceNames,jDestNames))
SourceSupplyAvail <- matrix(c(500, 500, 500, 600),
              ncol=1,dimnames=c(iSourceNames,"Supply"))
```

```r
DestDemandCapacity <- matrix(c(700, 500, 500, 600),
              nrow = 1,dimnames=c("Capacity",jDestNames))

MinCostModel <- MIPModel() %>%
  add_variable (x[i,j], i=1:iSourceCount, j=1:jDestCount,
               type="continuous", lb=0) %>%
  set_objective (sum_expr(sum_expr(UnitCost[i,j]*x[i,j],j=1:jDestCount),
                 i=1:iSourceCount),"min") %>%
  add_constraint (sum_expr(x[i,j], i=1:iSourceCount) <= DestDemandCapacity[1,j],
                 j=1:jDestCount) %>%
  add_constraint (sum_expr(x[i,j], j=1:jDestCount) == SourceSupplyAvail[i,1],
                 i=1:iSourceCount) %>%

  solve_model(with_ROI(solver = "glpk"))
MinCostModel
```

```
## Status: optimal
## Objective value: 14300
```

```r
results.unitsupply <- matrix (rep(-1.0,jDestCount), nrow = jDestCount, ncol=1,
                            dimnames=c(jDestNames,c("x")))
temp <- get_solution (MinCostModel, x[i,j])
results.unitsupply <- t(temp [,4] )
results.unitsupply <- matrix (results.unitsupply, nrow = iSourceCount, ncol=jDestCount,
                              dimnames=c(iSourceNames,jDestNames))
pander(results.unitsupply,
       caption ="Units to supply from Source to Destination for Minimum cost")
```

Table 5: Units to supply from Source to Destination for Minimum cost

	Dest1	Dest2	Dest3	Dest4
Source1	0	0	500	0
Source2	200	300	0	0
Source3	500	0	0	0
Source4	0	0	0	600