rsingav1_3

2023-10-13

Formulation of this problem:

From the table given, it seems like the supply(100+120=220) exceeds demand(80+60+70=210) by 10 AEDs. So, this is a unbalanced transportation problem. To balance we can create a new dummy warehouse 4 with the demand 10 AEDs and shipping cost to this warehouse is 0. With this, the problem becomes balanced and the constraints can be binding.

Step 1: Define Decision Variables

A1, B1: number of AEDs shipped from all Plants(A,B) to warehouse 1 A2, B2: number of AEDs shipped from all Plants(A,B) to warehouse 2 A3, B3: number of AEDs shipped from all Plants(A,B) to warehouse 3 A4, B4: number of AEDs shipped from all Plants(A,B) to warehouse 4(dummy warehouse) There are 8 in number:

Step 2: Set Objective Function

Minimize the total cost, which is the sum of production costs and shipping costs. Note: shipping cost is 0 for Warehouse 4(Dummy)

Objective Function: Minimize C = 600A1 + 600A2 + 600A3 + 600A4+ 625B1+ 625B2 + 625B3 + 625B4+ 22A1 + 14A2 + 30A3 + 16B1+ 20B2 + 24B3 and this can be further simplified as:

C = 622A1 + 614A2 + 630A3 + 600A4 + 641B1 + 645B2 + 649B3 + 625B4 (Combining both production and shipping costs)

Step 3: Set Constraints

1) Production Capacity Constraints: A1+A2+A3+A4 =100(Plant A Capacity) B1+B2+B3+B4 =120(Plant B Capacity)
2) Demand Constraints: A1+B1=80(Warehouse 1 Demand) A2+B2=60(Warehouse 2 Demand) A3+B3=70(Warehouse 3 Demand) A4+B4=10(Warehouse 4 Demand(Dummy))

Step 4: Non-Negativity Constraints

A1>=0, A2>=0, A3>=0, A4>=0, B1>=0, B2>=0, B3>=0, B4>=0

We can solve the above transportation problem in 2 ways:

1) using lpSolveAPI(lp.control())
2) using lpSolve(lp.transport())

First way: Using lpSolveAPI(lp.control())

Firstly installing and loading lpSolveAPI library

```
library(lpSolveAPI)
```

##Define Decision Variables A1, B1: number of AEDs shipped from all Plants(A,B) to warehouse 1 A2, B2: number of AEDs shipped from all Plants(A,B) to warehouse 2 A3, B3: number of AEDs shipped from all Plants(A,B) to warehouse 3 A4, B4: number of AEDs shipped from all Plants(A,B) to warehouse 4(dummy warehouse) There are 8 in number:

```
# Create a model with 9 decision variables
lprec_heart_start = make.lp(0, 8)
```

Objective Function: Minimize C = 622A1 + 614A2 + 630A3 + 600A4+ 641B1 + 645B2 + 649B3 + 625B4

Define the objective function coefficients. In this case, number of AEDs produced from each plant so as to minimize the cost

```
set.objfn(lprec_heart_start, c(622, 614, 630, 600, 641, 645, 649, 625))
```

Define the constraint matrix. This includes the production capacities of 2 plants, Monthly Demand for 3 warehouses and non-negativity constraints. As it is a balanced transportation problem, we can make the constraints binding

More details on Constraints:

1) Production Capacity Constraints: A1+A2+A3+A4 =100(Plant A Capacity) B1+B2+B3+B4 =120(Plant B Capacity)
2) Demand Constraints: A1+B1=80(Warehouse 1 Demand) A2+B2=60(Warehouse 2 Demand) A3+B3=70(Warehouse 3 Demand) A4+B4=10(Warehouse 4 Demand(Dummy))
3) A1>=0, A2>=0, A3>=0, A4>=0, B1>=0, B2>=0, B3>=0, B4>=0

```r
# Adding constraints and defining the constraint directions and right hand side of the constraints as p
add.constraint(lprec_heart_start, c(1, 1, 1, 1, 0, 0, 0, 0), "=", 100) # Plant A capacity
add.constraint(lprec_heart_start, c(0, 0, 0, 0, 1, 1, 1 ,1), "=", 120) # Plant B capacity
add.constraint(lprec_heart_start, c(1, 0, 0, 0, 1, 0, 0, 0), "=", 80) # Warehouse 1 Demand
add.constraint(lprec_heart_start, c(0, 1, 0, 0, 0, 1, 0, 0), "=", 60) # Warehouse 2 Demand
add.constraint(lprec_heart_start, c(0, 0, 1, 0, 0, 0, 1, 0), "=", 70) # Warehouse 3 Demand
add.constraint(lprec_heart_start, c(0, 0, 0, 1, 0, 0, 0, 1), "=", 10) # Warehouse 4 Demand(Dummy)
add.constraint(lprec_heart_start, c(1, 0, 0, 0, 0, 0, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 1, 0, 0, 0, 0, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 1, 0, 0, 0, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 0, 1, 0, 0, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 0, 0, 1, 0, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 0, 0, 0, 1, 0, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 0, 0, 0, 0, 1, 0), ">=", 0) # NonNegative constraint
add.constraint(lprec_heart_start, c(0, 0, 0, 0, 0, 0, 0, 1), ">=", 0) # NonNegative constraint
```

```r
# Set all variables to be integer as we do not want float values for the number of AEDs
for (i in 1:6) {
  set.type(lprec_heart_start, i, "integer")
}
```

Solve the linear programming problem to minimize the cost while meeting the above constraints

```r
# Set the objective function to minimize the cost
lp.control(lprec_heart_start, sense='min')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] -1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
```

```
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "minimize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
# Solve the linear programming model
solve(lprec_heart_start)
```

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

This will give you the optimal production amounts for each size at each plant to maximize profit, subject to the constraints.

A1 = 30 A2 = 60 A3 = 0 A4 = 10 B1 = 50 B2 = 0 B3 = 70 B4 = 0

```
# Get the solution
all_unit_sizes_for_all_plants = get.variables(lprec_heart_start)
print(all_unit_sizes_for_all_plants)
```

```
## [1] 30 60  0 10 50  0 70  0
```

Also the optimal cost or minimum cost is: 138980$

If we did not balance the problem, the minimum cost could have been just 132790$, but we have to account for the costs of excess supply to be consistent and accurate. So this is the final minimum cost that is achieved using the above optimization.

```
# Get the optimal cost
minimum_cost = get.objective(lprec_heart_start)
print(minimum_cost)
```

```
## [1] 138980
```

Writing the above lp model to a file:

```
write.lp(lprec_heart_start, filename = "heart_start_lp.lp")
```

#Second way: Using lpSolve(lp.transport())

Firstly installing and loading lpsolve library

```
library(lpSolve)
```

##Define Decision Variables A1, B1: number of AEDs shipped from all Plants(A,B) to warehouse 1 A2, B2: number of AEDs shipped from all Plants(A,B) to warehouse 2 A3, B3: number of AEDs shipped from all Plants(A,B) to warehouse 3 A4, B4: number of AEDs shipped from all Plants(A,B) to warehouse 4(dummy warehouse) There are 8 in number:

Costs are specified below for the above 8 decision variables

```
#set transportation costs matrix
costs = matrix(c(622, 614, 630, 600,
                 641, 645, 649, 625), nrow=2, byrow=TRUE)
```

Set warehouse names(demand) and Plant names(supply)

```
colnames(costs) = c("Warehouse 1", "Warehouse 2", "Warehouse 3", "Warehouse 4")
rownames(costs) = c("Plant A", "Plant B")
```

Set constraints in the below format. This includes the production capacities of 2 plants, Monthly Demand for 3 warehouses and non-negativity constraints

More details on Constraints:

1) Production Capacity Constraints: A1+A2+A3+A4 =100(Plant A Capacity) B1+B2+B3+B4 =120(Plant B Capacity)
2) Demand Constraints: A1+B1=80(Warehouse 1 Demand) A2+B2=60(Warehouse 2 Demand) A3+B3=70(Warehouse 3 Demand) A4+B4=10(Warehouse 4 Demand(Dummy))
3) A1>=0, A2>=0, A3>=0, A4>=0, B1>=0, B2>=0, B3>=0, B4>=0(these are automatically applied using lp.transport function)

```
# Set equality signs for suppliers
row.signs = rep("=", 2)

# Set right hand side coefficients for suppliers
row.rhs = c(100, 120)

# Set equality signs for Production capacities(Demands)
col.signs = rep("=", 4)

# Set right hand side coefficients for Production capacities(Demands)
col.rhs = c(80, 60, 70, 10)
```

```
lp_heart_start_solution = lp.transport(costs, "min", row.signs, row.rhs, col.signs, col.rhs)
print(lp_heart_start_solution$solution) # Calculate the number of AEDs to optimize the cost
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   30   60    0   10
## [2,]   50    0   70    0
```

```
print(lp_heart_start_solution$objval) # Minimum cost
```

```
## [1] 138980
```

This will give you the optimal production amounts for each size at each plant to maximize profit, subject to the constraints.

A1 = 30 A2 = 60 A3 = 0 A4 = 10 B1 = 50 B2 = 0 B3 = 70 B4 = 0

Also the optimal cost or minimum cost is: 138980$

If we did not balance the problem, the minimum cost could have been just 132790$, but we have to account for the costs of excess supply to be consistent and accurate.