

QMM_TRANSPORTATION

Venkateswara Rao Jammula

2023-10-15

```
#USING lpSolveAPI  
library(lpSolveAPI)
```

We have 6 decision variables, and 5 constraints (e.g., 3 warehouse and 2 supplier constraints).

Let,

X_{11} = total units shipped from plant A to warehouse 1
 X_{12} = total units shipped from plant A to warehouse 2
 X_{13} = total units shipped from plant A to warehouse 3
 X_{21} = total units shipped from plant B to warehouse 1
 X_{22} = total units shipped from plant B to warehouse 2
 X_{23} = total units shipped from plant B to warehouse 3

LP formulation for this problem:

We have only 2 plants, with 3 destinations or warehouses

$$Z = 22X_{11} + 14X_{12} + 30X_{13} + 16X_{21} + 20X_{22} + 24X_{23}$$

ST:

Demander Constraints $1X_{11} + 1X_{12} = 80$ $1X_{12} + 1X_{22} = 60$ $1X_{13} + 1X_{23} = 70$

Supplier Constraints $1X_{11} + 1X_{12} + 1X_{13} = 100$ $1X_{21} + 1X_{22} + 1X_{23} = 120$ $20X_7 + 15X_8 + 12X_9 = 5000$

$X_{ij} \geq 0$ The variables x_{ij} refer to units shipped from the plants (i) to the various warehouses (j).

The total supply is 220, while the total warehouse capacity is only 210. So, creating a dummy warehouse capacity

To determine the cost associated with producing and storing at the dummy installation site, we need to consider that the objective function coefficients include the cost of storage. Typically, for a dummy warehouse, there are no actual shipments, so the objective coefficient for such a variable would be zero. However, in linear programming, since we aim to minimize costs, we often assign a high cost to the dummy warehouses to discourage their use in the optimal solution. This approach is known as the "Big M method."

In this particular case, we can set the cost for X_{14} (representing shipments to the dummy warehouse) to \$600, and for X_{24} , which represents shipments to another dummy

warehouse, to \$625. This effectively puts a penalty on using these dummy warehouses as part of the optimal solution. The high costs make it less favorable to allocate resources to these dummy warehouses, encouraging the model to find solutions that prioritize the actual warehouses with lower associated costs for production and storage.

The complete formulation is :

X_{11} = total units shipped from plant A to warehouse 1 X_{12} = total units shipped from plant A to warehouse 2 X_{13} = total units shipped from plant A to warehouse 3 X_{14} (dummy) = total units shipped from plant A to warehouse 4 X_{21} = total units shipped from plant B to warehouse 1 X_{22} = total units shipped from plant B to warehouse 2 X_{23} = total units shipped from plant B to warehouse 3 X_{24} (dummy) = total units shipped from plant B to warehouse 4

LP formulation for this problem:

2 plants, with 3 warehouses

$$Z = 22X_{11} + 14X_{12} + 30X_{13} + 0X_{14} + 16X_{21} + 20X_{22} + 24X_{23} + 0X_{24}$$

ST:

Demand Constraints $1X_{11} + 1X_{12} = 80$ $1X_{12} + 1X_{22} = 60$ $1X_{13} + 1X_{23} = 70$ $1X_{14} + 1X_{24} = 10$

Supplier Constraint $1X_{11} + 1X_{12} + 1X_{13} = 100$ $1X_{21} + 1X_{22} + 1X_{23} = 120$ $20X_7 + 15X_8 + 12X_9 = 5000$

$X_{ij} \geq 0$

2 new decision variables (dummies X_{14} and X_{24}) and an extra constraint (dummy warehouse). This gives 8 decision variables and 6 structural constraints.

```
# Create an LP model with 6 constraints (rows) and 8 decision variables (columns)
```

```
lptransmodel <- make.lp(6, 8)
```

```
# Set the objective function coefficients (default is minimization)
```

```
set.objfn(lptransmodel, c(22, 14, 30, 600, 16, 20, 24, 625))
```

```
# Change the optimization direction to minimization (optional, as it's already the default)
```

```
lp.control(lptransmodel, 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"

# Add constraints to the model:

# Plant A production and shipping capacity constraint
add.constraint(lptransmodel, c(1, 1, 1, 1, 0, 0, 0, 0), "=", 100)

# Plant B production and shipping capacity constraint
add.constraint(lptransmodel, c(0, 0, 0, 0, 1, 1, 1, 1), "=", 120)

# Warehouse 1 capacity requirement constraint
add.constraint(lptransmodel, c(1, 0, 0, 0, 1, 0, 0, 0), "=", 80)

# Warehouse 2 capacity requirement constraint
add.constraint(lptransmodel, c(0, 1, 0, 0, 0, 1, 0, 0), "=", 60)

# Warehouse 3 capacity requirement constraint
add.constraint(lptransmodel, c(0, 0, 1, 0, 0, 0, 1, 0), "=", 70)

# Warehouse 4 (dummy) capacity requirement constraint
add.constraint(lptransmodel, c(0, 0, 0, 1, 0, 0, 0, 1), "=", 10)

# Print the LP model
lptransmodel

## Model name:
##
##      C1      C2      C3      C4      C5      C6      C7      C8
## Minimize  22      14      30      600      16      20      24      625
## R1         0       0       0       0       0       0       0       0  free    0
## R2         0       0       0       0       0       0       0       0  free    0
## R3         0       0       0       0       0       0       0       0  free    0
## R4         0       0       0       0       0       0       0       0  free    0
## R5         0       0       0       0       0       0       0       0  free    0
## R6         0       0       0       0       0       0       0       0  free    0
## R7         1       1       1       1       0       0       0       0    = 100
## R8         0       0       0       0       1       1       1       1    = 120
## R9         1       0       0       0       1       0       0       0    = 80
## R10        0       1       0       0       0       1       0       0    = 60
## R11        0       0       1       0       0       0       1       0    = 70
## R12        0       0       0       1       0       0       0       1    = 10
## Kind      Std     Std     Std     Std     Std     Std     Std     Std

```

```

## Type      Real  Real  Real  Real  Real  Real  Real  Real
## Upper     Inf   Inf   Inf   Inf   Inf   Inf   Inf   Inf
## Lower      0     0     0     0     0     0     0     0

#solving lp Problem
solve(lptransmodel)

## [1] 0

# '0' indicates a solution was found.

get.objective(lptransmodel)

## [1] 9980

get.variables(lptransmodel)

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

get.constraints(lptransmodel)

## [1]  0  0  0  0  0  0 100 120  80  60  70  10

get.sensitivity.obj(lptransmodel)

## $objfrom
## [1] 1.00e+01 -1.00e+30  3.00e+01 -1.00e+30  1.60e+01  8.00e+00 -1.00e+30
## [8] 5.94e+02
##
## $objtill
## [1] 2.20e+01 2.60e+01 1.00e+30 6.31e+02 2.80e+01 1.00e+30 2.40e+01
1.00e+30

get.sensitivity.rhs(lptransmodel)

## $duals
## [1]  0  0  0  0  0  0 22 16  0 -8  8 578  0  0  0  0  0
12  0
## [20] 31
##
## $dualsfrom
## [1] -1.0e+30 -1.0e+30 -1.0e+30 -1.0e+30 -1.0e+30 -1.0e+30 1.0e+02
1.2e+02
## [9] -1.0e+30  6.0e+01  7.0e+01  1.0e+01 -1.0e+30 -1.0e+30 -5.0e+01 -
1.0e+30
## [17] -1.0e+30 -3.0e+01 -1.0e+30 -3.0e+01
##
## $dualstill
## [1] 1.0e+30 1.0e+30 1.0e+30 1.0e+30 1.0e+30 1.0e+30 1.0e+02 1.2e+02
1.0e+30
## [10] 6.0e+01 7.0e+01 1.0e+01 1.0e+30 1.0e+30 3.0e+01 1.0e+30 1.0e+30
5.0e+01
## [19] 1.0e+30 1.0e+01

```

Objective Value: The minimum cost associated with this transportation transshipment problem is \$9,980. This cost represents the most cost-effective way to move goods from production plants to warehouses, taking into account both production and shipping costs.

Variable Values: The solution provides values for each of the decision variables, which indicate how many units of a product are shipped from each source (plants) to each destination (warehouses).

The variable values are as follows:

- **X11:** 30 units are shipped from Plant A to Warehouse 1.
- **X12:** 60 units are shipped from Plant A to Warehouse 2.
- **X13:** 0 units are shipped from Plant A to Warehouse 3.
- **X14:** 10 units are shipped from Plant A to Warehouse 4 (dummy). These units do not actually get shipped but remain in storage as a result of the optimization.
- **X21:** 50 units are shipped from Plant B to Warehouse 1.
- **X22:** 0 units are shipped from Plant B to Warehouse 2.
- **X23:** 70 units are shipped from Plant B to Warehouse 3.
- **X24:** 0 units are shipped from Plant B to Warehouse 4 (dummy).

The solution demonstrates that to minimize transportation costs while meeting capacity and demand constraints, it's optimal to ship 30 units from Plant A to Warehouse 1, 60 units from Plant A to Warehouse 2, and so on. The 10 units shipped from Plant A to Warehouse 4 are not actually shipped but rather remain in storage (hence, the "dummy" designation).

Production Costs: The production costs, which are irrelevant to the total shipping costs, are as follows: 100 units are produced at Plant A with a cost of \$60,000, and 120 units are produced at Plant B with a cost of \$75,000. The total production cost is \$135,000.

The solution found by the LP model indicates the optimal way to transport goods from the production plants to the warehouses while minimizing total costs, which include both shipping and production costs. The variable values show how many units are shipped from each source to each destination. The total cost is \$9,980, considering the penalty for items left in storage.

#may also be done as

Minimized cost of production and shipping : Objective Function $Z_{min} = 622X_{11} + 614X_{12} + 630X_{13} + 0X_{14} + 641X_{21} + 645X_{22} + 649X_{23} + 0X_{24}$ 6 decision variables and 2 dummy variables are considered to equalize supply and demand. Constraints: Supply Constraints $X_{11} + X_{12} + X_{13} + X_{14} = 100$ $X_{21} + X_{22} + X_{23} + X_{24} = 120$ Demand Constraints $X_{11} + X_{21} = 80$ $X_{12} + X_{22} = 60$ $X_{13} + X_{23} = 70$ $X_{14} + X_{24} = 10$ Where, $X_{ij} \geq 0$ (i (Plant) = 1,2 and j (warehouses) = 1,2,3,4)

```

library(lpSolveAPI)
lprec<-make.lp(0,8)
lp.control(lprec,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"

set.objfn(lprec,c(622,614,630,0,641,645,649,0))
add.constraint(lprec,rep(1,4),"=",100,indices=c(1,2,3,4))
add.constraint(lprec,rep(1,4),"=",120,indices=c(5,6,7,8))
add.constraint(lprec,rep(1,2),"=",80,indices=c(1,5))
add.constraint(lprec,rep(1,2),"=",60,indices=c(2,6))
add.constraint(lprec,rep(1,2),"=",70,indices=c(3,7))
add.constraint(lprec,rep(1,2),"=",10,indices=c(4,8))
solve(lprec)

## [1] 0

get.objective(lprec)

## [1] 132790

get.constraints(lprec)

## [1] 100 120 80 60 70 10

get.variables(lprec)

## [1] 0 60 40 0 80 0 30 10

```