

# Module 6 — Transportation Model

Sonali Shankeshi

2025-10-14

We want to **decide how many AEDs to make at Plant A and Plant B and how many to ship to each warehouse (W1, W2, W3)** so that:

- All warehouse demands are **met exactly**, and
- The **total cost** is **as small as possible**.  
Total cost = **production cost + shipping cost**.

## Data

- **Plant A:** production cost = 600 per unit, capacity = 100
- **Plant B:** production cost = 625 per unit, capacity = 120
- **Shipping cost per unit**
  - $A \rightarrow W1$ : 22,  $A \rightarrow W2$ : 14,  $A \rightarrow W3$ : 30
  - $B \rightarrow W1$ : 16,  $B \rightarrow W2$ : 20,  $B \rightarrow W3$ : 24
- **Demand:**  $W1 = 80$ ,  $W2 = 60$ ,  $W3 = 70$  (total 210)

We add production + shipping to get **total unit cost** per route: -  $A \rightarrow (W1, W2, W3) = (622, 614, 630)$

-  $B \rightarrow (W1, W2, W3) = (641, 645, 649)$

## Formulation

### Decision variables

-  $(x_{A1}, x_{A2}, x_{A3})$ : units from A to W1, W2, W3

-  $(x_{B1}, x_{B2}, x_{B3})$ : units from B to W1, W2, W3

### Objective (minimize total cost)

[ min;  $622x_{A1} + 614x_{A2} + 630x_{A3} + 641x_{B1} + 645x_{B2} + 649x_{B3}$ .]

### Demand (must meet exactly)

[  $x_{A1} + x_{B1} = 80$ , quad  $x_{A2} + x_{B2} = 60$ , quad  $x_{A3} + x_{B3} = 70$ .]

### Capacity (cannot exceed)

[  $x_{A1} + x_{A2} + x_{A3} \leq 100$ , quad  $x_{B1} + x_{B2} + x_{B3} \leq 120$ .]

**Nonnegativity:**  $(x_{ij} \geq 0)$ .

# Solving

```
# install.packages("lpSolve") # run once if needed
library(lpSolve)
library(knitr)

# Costs per route = production + shipping
# Variable order: xA1, xA2, xA3, xB1, xB2, xB3
costs <- c(622, 614, 630, 641, 645, 649)

# Demand equalities (W1, W2, W3)
Aeq <- rbind(
  c(1,0,0, 1,0,0), # W1: xA1 + xB1 = 80
  c(0,1,0, 0,1,0), # W2: xA2 + xB2 = 60
  c(0,0,1, 0,0,1)  # W3: xA3 + xB3 = 70
)
beq <- c(80, 60, 70)

# Capacity inequalities (A, B)
Aineq <- rbind(
  c(1,1,1, 0,0,0), # Plant A ≤ 100
  c(0,0,0, 1,1,1)  # Plant B ≤ 120
)
bineq <- c(100, 120)

# Combine constraints for lp()
const.mat <- rbind(Aeq, Aineq)
const.dir <- c("=", "=", "=", "<=", "<=")
const.rhs <- c(beq, bineq)

# Solve
sol <- lp(direction = "min",
          objective.in = costs,
          const.mat = const.mat,
          const.dir = const.dir,
          const.rhs = const.rhs)

sol$status      # 0 = optimal
```

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

```
sol$objval      # minimum total cost
```

```
## [1] 132790
```

```
sol$solution    # xA1, xA2, xA3, xB1, xB2, xB3
```

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

## Interpretation

We created a small cost-minimization model for moving AEDs between two plants (A and B) to three warehouses (W1, W2, W3). First, we combined production and shipping into one “per-unit route cost” (this is the `costs` vector in the order `xA1`, `xA2`, `xA3`, `xB1`, `xB2`, `xB3`). We then told the solver two things: (1) each warehouse must get exactly its demand (those are the three “=” rows), and (2) each plant can not produce more than its capacity (the two “<=” rows). With these pieces, `lp()` finds the shipment plan with the “minimum total cost”.

The outputs we got was `sol$status = 0` means the solver found an “optimal” plan, and `sol$objval` is that “minimum total cost” (comes to “132,790” here), and `sol$solution` is the actual shipment quantities in the fixed above order. Judging by that solution we get one clear plan... Plant A ships “40 to W1” and “60 to W2” (a total of “100”, which is A’s total capacity), and Plant B ships “40 to W1” and “70 to W3” (a total of “110”, leaving B at 10 for unutilized capacity). All warehouse demand is exactly equal (W1=80, W2=60, W3=70). Plant A is less expensive for every destination, so we use Plant A entirely and send it to the lowest cost lanes first (W2 then W1) with Plant B covering the remaining shipments, mainly W3. There are lots of shipment splits with the same total cost, but this split is a valid optimal solution.

## summary table (Shipments, costs, totals)

```
routes <- c("A->W1", "A->W2", "A->W3", "B->W1", "B->W2", "B->W3")
ship_units <- as.numeric(sol$solution)
unit_ship_cost <- c(22, 14, 30, 16, 20, 24) # shipping only
prod_cost_per_unit <- c(rep(600, 3), rep(625, 3))

df <- data.frame(
  Route = routes,
  Units = ship_units,
  Prod_Cost_per_Unit = prod_cost_per_unit,
  Ship_Cost_per_Unit = unit_ship_cost,
  Total_Unit_Cost = costs,
  Row_Total_Cost = ship_units * costs
)

kable(df, caption = "Optimal Shipments and Cost Breakdown (per route)")
```

Optimal Shipments and Cost Breakdown (per route)

Route	Units	Prod_Cost_per_Unit	Ship_Cost_per_Unit	Total_Unit_Cost	Row_Total_Cost
A->W1	600		22	622	0
A->W2	600		14	614	36840
A->W3	600		30	630	25200
B->W1	625		16	641	51280
B->W2	625		20	645	0
B->W3	625		24	649	19470

```
# Production by plant and cost breakdown
A_units <- sum(ship_units[1:3])
B_units <- sum(ship_units[4:6])

production_cost <- A_units*600 + B_units*625
shipping_cost    <- sum(ship_units * unit_ship_cost)
total_check      <- production_cost + shipping_cost

kable(data.frame(
  Plant = c("A","B"),
  Produced = c(A_units, B_units)
), caption = "Production by Plant")
```

Production by  
Plant

#### PlantProduced

```
A    100
B    110
```

```
kable(data.frame(
  Production_Cost = production_cost,
  Shipping_Cost = shipping_cost,
  Total_Cost = total_check,
  Solver_ObjVal = sol$objval
), caption = "Cost Summary (should match solver objective)")
```

Cost Summary (should match solver objective)

```
Production_CostShipping_CostTotal_CostSolver_ObjVal
128750          4040          132790    132790
```

## Interpretation

Here we are taking the solver's raw answer ( `sol$solution` ), and outputting it into a viewable report. The first table, “**Optimal Shipments and Cost Breakdown (per route)**”, includes all the routes (A → W1, A → W2, ..., B → W3). The **Units** column shows the actual number of units shipped per route. The two “per unit” columns breakdown the cost into **production** and **shipping**, and **Total\_Unit\_Cost** is a sum of them (this only uses the `costs` vector you constructed in the model). **Row\_Total\_Cost** is simply **Units x Total\_Unit\_Cost** per travel route. You'll see the list allows you to quickly identify which rows are being used (Units > 0) and how each route contributes to the total cost.

The second table, “**Production by Plant**”, just adds the shipments leaving each plant. It tells you how many units **Plant A** and **Plant B** actually shipped. You can then compare it to the capacity ( $A \leq 100$ ,  $B \leq 120$ ). In the ideal answer for this given problem, you should see **A at full capacity** being shipped and **B with a little leftover capacity**. This intuition makes sense since A is cheaper.

The last table “**Cost Summary (should match solver objective)**” provides the total cost broken down into its two parts, `Production_Cost` and `Shipping_Cost`, to obtain the `Total_Cost`. The solver's number `Solver_ObjVal` is also presented alongside it. These two totals should match perfectly

(allowing for tiny rounding issues). If they do match, this means your segregation of costs is consistent with what the solver optimized. You should also notice that production cost is a major part of the total cost and shipping cost is a relatively minor part—this gives the model reason to use Plant A (the least expensive producer) as much as it can and then the least expensive shipping lanes.

## Conclusion

We used Plant A as much as possible because A is cheaper than B for every warehouse.

A's cheapest lane is W2, then W1, so A ships there first.

The remaining demand (especially W3) is filled by Plant B.

The minimum total cost is `format(sol$objval, big.mark=",")`, which matches the cost breakdown table.

There are other shipment patterns that have the same minimum total cost; the plan shown is one valid optimal solution.