```julia
warehouse = 4
consumer = 3

costs = [25 35 125;
        10 30 50;
        20 40 30;
        70 75 65]

supply = [30 30 30 45]
demand = [45 50 40];

using JuMP, HiGHS

ship = Model(HiGHS.Optimizer)

#Create a variable xij for each pair of warehouse and consumer that
represents
#the amount of shipment sent from warehouse i to consumer j
@variable(ship, x[1:warehouse,1:consumer] >= 0)

#Each warehouse i can provide at most supply[i] of stuff
@constraint(ship, supplyconstraint[i in 1:warehouse], sum(x[i,j] for j
in 1:consumer) <= supply[i])

#Each consumer j requires at least demand[j] of stuff
@constraint(ship, demandconstraint[j in 1:consumer], sum(x[i,j] for i
in 1:warehouse) >= demand[j])

#Cost of sending from warehose i to consuemr j is given by costs[i,j]
@objective(ship, Min, sum(sum(costs[i,j]*x[i,j] for j in 1:consumer)
for i in 1:warehouse))

print(ship)
```

```
Min 25 x[1,1] + 35 x[1,2] + 125 x[1,3] + 10 x[2,1] + 30 x[2,2] + 50
x[2,3] + 20 x[3,1] + 40 x[3,2] + 30 x[3,3] + 70 x[4,1] + 75 x[4,2] +
65 x[4,3]
Subject to
 demandconstraint[1] : x[1,1] + x[2,1] + x[3,1] + x[4,1] ≥ 45
 demandconstraint[2] : x[1,2] + x[2,2] + x[3,2] + x[4,2] ≥ 50
 demandconstraint[3] : x[1,3] + x[2,3] + x[3,3] + x[4,3] ≥ 40
 supplyconstraint[1] : x[1,1] + x[1,2] + x[1,3] ≤ 30
 supplyconstraint[2] : x[2,1] + x[2,2] + x[2,3] ≤ 30
 supplyconstraint[3] : x[3,1] + x[3,2] + x[3,3] ≤ 30
 supplyconstraint[4] : x[4,1] + x[4,2] + x[4,3] ≤ 45
 x[1,1] ≥ 0
 x[2,1] ≥ 0
 x[3,1] ≥ 0
 x[4,1] ≥ 0
 x[1,2] ≥ 0
 x[2,2] ≥ 0
 x[3,2] ≥ 0
 x[4,2] ≥ 0
 x[1,3] ≥ 0
 x[2,3] ≥ 0
 x[3,3] ≥ 0
 x[4,3] ≥ 0
```

```
optimize!(ship)
@show objective_value(ship)
@show value.(x)
```

```
Running HiGHS 1.7.2 (git hash: 5ce7a2753): Copyright (c) 2024 HiGHS under MIT licence terms
Coefficient ranges:
  Matrix [1e+00, 1e+00]
  Cost   [1e+01, 1e+02]
  Bound  [0e+00, 0e+00]
  RHS    [3e+01, 5e+01]
Presolving model
7 rows, 12 cols, 24 nonzeros  0s
7 rows, 12 cols, 24 nonzeros  0s
Presolve : Reductions: rows 7(-0); columns 12(-0); elements 24(-0) - Not reduced
Problem not reduced by presolve: solving the LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 3(135) 0s
          7     5.2250000000e+03 Pr: 0(0) 0s
Model   status      : Optimal
Simplex   iterations: 7
Objective value     :  5.2250000000e+03
HiGHS run time      :          0.00
objective_value(ship) = 5225.0
value.(x) = [0.0 30.0 0.0; 15.0 15.0 0.0; 30.0 0.0 0.0; 0.0 5.0 40.0]
```

```
4×3 Matrix{Float64}:
  0.0  30.0   0.0
 15.0  15.0   0.0
 30.0   0.0   0.0
  0.0   5.0  40.0
```