SETS:

The simple/primitive sets - these sets will eventually be the parameters;
Unlike the previous problem, each of these sets have two defining characteristics: Total number of shipments and resource constraints;

Cannery: CanProduce, Output;
Warehouse: WarProduce, Allocation;

A derived set maps the warehouses to the canneries, creating the matrix given in the original problem;
notice that each matrix term has both a cost and an amount;
Links(Cannery, Warehouse): ShipCost, Ship;

ENDSETS

DATA:

Input the Canneries and their constraints;
Cannery, Output =
  C1 75
  C2 125
  C3 100;

Input the Warehouses and their constraints;
Warehouse, Allocation =
  W1 80
  W2 65
  W3 70
  W4 85;

Input the shipping costs per truckload as given in the original shipping cost matrix;
ShipCost = 464 513 654 867
          352 416 690 791
          995 682 388 685;

ENDDATA

Minimize total cost;
MIN = @SUM(Links: ShipCost*Ship);

Cannery restraints;
For each cannery i;
@FOR(Cannery(i):
  sum the warehouse truckloads for every warehouse and set them equal to the output constraint (output);
@SUM(Warehouse(j): Ship(i,j)) = Output(i));
Warehouse constraints;

For each machine i;

@FOR(Warehouse(j):

! sum the cannery truckloads for every cannery and set them equal to the
output constraint (allocation);

@SUM(Cannery(i): Ship(i,j)) = Allocation(j));