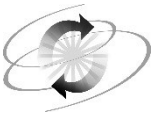

Exercise Solution

Formulating a Transportation Problem

Consider the transportation problem with unit shipping costs, supply, and demand in the table below.

DESTINATIONS

ORIGINS		Boston	Newark	Toronto	Supply
	Chicago	40	25	15	200
	Detroit	30	15	10	150
	Demand	50	150	100	

a. Complete the parameter declaration statements in part **4a** of the program **ch2ex.sas**. Replace each *INDEX-SET* with the appropriate (declared) index set and *INITIALIZERS* with the appropriate data items from the table. This assigns values to a two-dimensional array **unit_cost** of unit costs, an array **supply** of supplies, and an array **demand** of demands. How does PROC OPTMODEL format the output of the PRINT statement in the SAS program?

The following are the declaration statements:

```
proc optmodel;
   /* declare sets and parameters */
   set ORIGINS = /Chicago Detroit/;
   set DESTINATIONS = /Boston Newark Toronto/;
   num supply {ORIGINS} = [200 150];
   num demand {DESTINATIONS} = [50 150 100];
   num unit_cost {ORIGINS, DESTINATIONS} =
       [40 25 15 30 15 10];

   /* print parameter arrays */
   print unit_cost supply demand;

quit;
```

In the declaration statement for **unit_cost**, the values can be divided into two lines to make it easier to parse the initializers.

PROC OPTMODEL Output

```
                    The OPTMODEL Procedure

                              unit_cost
                        Boston    Newark    Toronto

              Chicago       40        25         15
              Detroit       30        15         10

                   [1]        supply    demand
```

```
                                Boston                   50
                                Chicago       200
                                Detroit       150
                                Newark                  150
                                Toronto                 100
```

The PRINT statement combines **supply** and **demand** as a listing, whereas **unit_cost** is formatted as a separate table because it is a two-dimensional array.

b. Complete the variable declaration statements in part **4b** of the program **ch2ex.sas**. Replace each *INDEX-SET* with the appropriate (declared) index set and *EXPRESSION* with the appropriate variable expression. (The decision variables are initialized to 50 in order to print values for the implicit variables.)

The declaration statements are as follows:

```
proc optmodel;
   /* declare sets and parameters */
   set ORIGINS = /Chicago Detroit/;
   set DESTINATIONS = /Boston Newark Toronto/;
   num supply {ORIGINS} = [200 150];
   num demand {DESTINATIONS} = [50 150 100];
   num unit_cost {ORIGINS, DESTINATIONS} =
       [40 25 15 30 15 10];

   /* declare variables */
   var NumShip {ORIGINS, DESTINATIONS} >= 0 init 50;

   impvar FlowOut {i in ORIGINS} =
      sum {j in DESTINATIONS} NumShip[i,j];

   impvar FlowIn {j in DESTINATIONS} =
      sum {i in ORIGINS} NumShip[i,j];

   expand FlowOut['Chicago'];
   expand FlowIn['Boston'];

   expand / impvar;

   print FlowOut;
   print FlowIn;

quit;
```

PROC OPTMODEL Output

```
                              The OPTMODEL Procedure

Impvar FlowOut[Chicago] = NumShip[Chicago,Boston] + NumShip[Chicago,Newark] +
NumShip[Chicago,Toronto]
Impvar FlowIn[Boston] = NumShip[Chicago,Boston] + NumShip[Detroit,Boston]

                                                 Flow
```

```
                                  [1]         Out

                                  Chicago     150
                                  Detroit     150

                                              Flow
                                  [1]          In

                                  Boston      100
                                  Newark      100
                                  Toronto     100
```

The (individual) implicit variables are expanded, but the EXPAND/IMPVAR statement triggers the current model to be built. It does not produce any output because the implicit variables do not enter into the constraints or objective function of the optimization problem.

c. Complete the constraint and objective declaration statements in part 4c of the program ch2ex.sas and solve the transportation problem using PROC OPTMODEL. Use the EXPAND statement to check your formulation.

Hint: The optimal objective value is $5,750.

The formulation using index sets is as follows:

```
proc optmodel;
   /* declare sets and parameters */
   set ORIGINS = /Chicago Detroit/;
   set DESTINATIONS = /Boston Newark Toronto/;
   num supply {ORIGINS} = [200 150];
   num demand {DESTINATIONS} = [50 150 100];
   num unit_cost {ORIGINS, DESTINATIONS} =
      [40 25 15 30 15 10];

   /* declare variables */
   var NumShip {ORIGINS, DESTINATIONS} >= 0;

   impvar FlowOut {i in ORIGINS} =
      sum {j in DESTINATIONS} NumShip[i,j];

   impvar FlowIn {j in DESTINATIONS} =
      sum {i in ORIGINS} NumShip[i,j];

   /* declare constraints */
   con Supply_con {i in ORIGINS}:
      FlowOut[i] <= supply[i];

   con Demand_con {j in DESTINATIONS}:
      FlowIn[j] >= demand[j];

   /* declare objective */
   min TotalCost = sum {i in ORIGINS, j in DESTINATIONS}
```

```
      unit_cost[i,j] * NumShip[i,j];

   expand;

   solve;

   print NumShip;

quit;
```

PROC OPTMODEL Output

```
                                The OPTMODEL Procedure

Var NumShip[Chicago,Boston] >= 0
Var NumShip[Chicago,Newark] >= 0
Var NumShip[Chicago,Toronto] >= 0
Var NumShip[Detroit,Boston] >= 0
Var NumShip[Detroit,Newark] >= 0
Var NumShip[Detroit,Toronto] >= 0
Impvar FlowOut[Chicago] = NumShip[Chicago,Boston] + NumShip[Chicago,Newark] +
NumShip[Chicago,Toronto]
Impvar FlowOut[Detroit] = NumShip[Detroit,Boston] + NumShip[Detroit,Newark] +
NumShip[Detroit,Toronto]
Impvar FlowIn[Boston] = NumShip[Chicago,Boston] + NumShip[Detroit,Boston]
Impvar FlowIn[Newark] = NumShip[Chicago,Newark] + NumShip[Detroit,Newark]
Impvar FlowIn[Toronto] = NumShip[Chicago,Toronto] + NumShip[Detroit,Toronto]
Minimize TotalCost=40*NumShip[Chicago,Boston] + 25*NumShip[Chicago,Newark] +
15*NumShip[Chicago,Toronto] + 30*NumShip[Detroit,Boston] + 15*NumShip[Detroit,Newark] +
10*NumShip[Detroit,Toronto]
Constraint Supply_con[Chicago]: FlowOut[Chicago] <= 200
Constraint Supply_con[Detroit]: FlowOut[Detroit] <= 150
Constraint Demand_con[Boston]: FlowIn[Boston] >= 50
Constraint Demand_con[Newark]: FlowIn[Newark] >= 150
Constraint Demand_con[Toronto]: FlowIn[Toronto] >= 100

                                   Problem Summary

                          Objective Sense           Minimization
                          Objective Function           TotalCost
                          Objective Type                  Linear

                          Number of Variables                  6
                          Bounded Above                        0
                          Bounded Below                        6
                          Bounded Below and Above              0
                          Free                                 0
                          Fixed                                0

                          Number of Constraints                5
                          Linear LE (<=)                       2
                          Linear EQ (=)                        0
                          Linear GE (>=)                       3
                          Linear Range                         0

                          Constraint Coefficients             12
```

Solution Summary

Solver	LP
Algorithm	Dual Simplex
Objective Function	TotalCost
Solution Status	Optimal
Objective Value	5750
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0
Iterations	7
Presolve Time	0.00
Solution Time	0.00

NumShip

	Boston	Newark	Toronto
Chicago	50	0	100
Detroit	0	150	0

The output of the PRINT statement for **NumShip** is formatted as a table because it is a two-dimensional array. PROC OPTMODEL formats sparse two-dimensional arrays as lists. (The PMATRIX option controls the tolerance level.)