Chapter 6

The TRANS Procedure

Chapter Table of Contents

Chapter 6
The TRANS Procedure

Overview

The TRANS procedure is used to solve the *transportation problem*, which is a type of *network flow problem*. A node of a transportation problem is either a *source node* or a *destination node*. Each source node is able to supply a specified number of *flow units*; each destination node has a *demand* for a specified number of flow units. Each *arc* of a transportation problem originates at a source node and terminates at a destination node. Some arcs can have *capacities* (the maximum amount of flow that they can convey) and *lower flow bounds* (the minimum amount of flow that the arc can convey). Arcs also have per unit traversal costs, simply referred to as *costs* (for example, the cost incurred when one unit of flow is conveyed through an arc). Figure 6.1 shows a transportation problem network having three source nodes and two destination nodes.

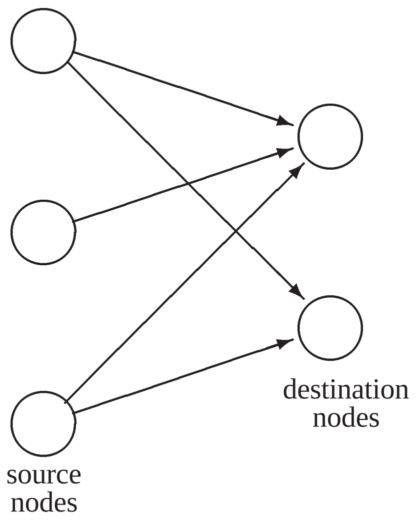

Figure 6.1. A Transportation Problem

Getting Started

The TRANS procedure accepts information about the transportation problem as data and produces a SAS data set that contains the flows that should be conveyed by each arc (the flow between each source and destination node) that minimizes the total cost of flow.

Introductory Example

Consider the SAS data set in Figure 6.2. The SAS code used to create the data set can be found in Example 6.1.

```
                      Uncapacitated Transportation Network

                              L           N     S            W
      A     C            H    o           e     a     S      a
      t     h     D      o    s           w     n     e      s     s
      l     i     e      u    _     M     _     _     a      h     u
      a     c     n      s    A     i     Y     F     t      i     p      c
 O    n     a     v      t    n     a     o     r     t      n     p      i
 b    t     g     e      o    g     m     r     a     l      g     l      t
 s    a     o     r      n    e     i     k     n     e      t     y      y

  1    50    75    89     8    27    39    64   100    50     8     .
  2    20    58   121    70   193    60    74   213   218    54    10   Atlanta
  3    58    20    92    94   174   118    71   185   173    57   150   Chicago
  4   121    92    20    87    83   172   163    94   102   149    90   Denver
  5    70    94    87    20   137    96   142   154   189   122    27   Houston
  6   193   174    83   137    20   223   245    34    95   230    80   Los_Ange
  7    60   118   172    96   233    20   109   259   273    92    26   Miami
  8    74    71   163   142   245   109    20   257   240    20    80   New_York
  9   213   185    94   164    34   259   257    20    67   244    25   San_Fran
 10   218   173   102   189    95   273   240    67    20   232     7   Seattle
 11    54    59   149   122   230    92    20   244   232    20    15   Washingt
```

Figure 6.2. Introductory Example Input Data Set

The first observation provides the number of units demanded at each destination node. The supply variable provides the number of units supplied at each source node. If you exclude the first observation and the supply variable, the remaining values are the cost of shipping one unit between nodes. For example, the per unit transportation cost between Miami and Houston is 96. The transportation problem is solved when the minimum total cost flow between supply points and destination points that satisfies the demand is found. PROC TRANS solves this problem and produces a SAS data set containing the number of units to ship between each supply point and demand point. It does not produce any output but does report the minimum cost on the log. See Example 6.1 for the results.

Syntax

> **PROC TRANS** *options* **;**
> **HEADNODE** *variables* **;**
> **SUPPLY** *variable* **;**
> **TAILNODE** *variable* **;**

The PROC TRANS statement invokes the procedure. The TAILNODE and SUPPLY statements are required.

Functional Summary

The following tables outline the options available for the TRANS procedure classified by function.

Table 6.1. Input Data Set Options

Description	**Statement**	**Option**
data set containing arc capacities	TRANS	CAPACITY=
data set containing cost, supply, and demand data	TRANS	COST=
data set containing arc minimum flows	TRANS	MINFLOW=

Table 6.2. Output Data Set Options

Description	**Statement**	**Option**
data set containing the solution data set	TRANS	OUT=

Table 6.3. Other Options

Description	**Statement**	**Option**
the default arc capacities	TRANS	DEFCAPACITY=
the default arc lower flow bounds	TRANS	DEFMINFLOW=
the observation number in the COST= data set that contains the demands at the destination nodes	TRANS	DEMAND=
find maximum transportation cost	TRANS	MAXIMUM
discard any excess supply or demand	TRANS	NOTHRUNET
force excess supply or demand through the network	TRANS	THRUNET

The statements and options available in PROC TRANS are discussed in the order in which they appeared in the preceding list of syntax elements.

PROC TRANS Statement

PROC TRANS *options* **;**

Input Data Set Options

CAPACITY=*SAS-data-set*
names the SAS data set that contains the capacity on each arc in the transportation network. These data specify the maximum allowable flow on each network arc. If the CAPACITY= option is omitted, then the value of the DEFCAPACITY= option is used for each arc in the network.

COST=*SAS-data-set*
names the SAS data set that contains the cost, supply, and demand data for the transportation network. If the COST= option is omitted, the most recently created SAS data set is used.

MINFLOW=*SAS-data-set*
names the SAS data set that contains the minimum flow data for the transportation network. These data specify the minimum required flow on each arc in the network. If the MINFLOW= option is omitted, then the value of the DEFMINFLOW= option is used for each arc in the network.

Output Data Set Options

OUT=*SAS-data-set*
specifies a name for the output data set. If the OUT= option is omitted, the SAS System creates a data set and names it according to the DATA*n* convention. Refer to the section "SAS Statements Used in the Data Set" in base SAS documentation for more information.

Other Options

DEFCAPACITY=*c*
specifies the default capacity for the arcs in the network. The default value of *c* is $+\infty$.

DEFMINFLOW=*s*
specifies the default minimum flow for the arcs in the network. The default value of *s* is 0.

DEMAND=*d*
gives the number of the observation that contains the number of units demanded at each destination node. The default value of *d* is 1.0. See the "Demand" section on page 519.

MAXIMUM
tells the procedure to maximize rather than minimize the objective function.

NOTHRUNET
tells PROC TRANS to drain away any excess supply (the amount that total supply exceeds total demand) or excess demand (the amount that total demand exceeds total supply). If total supply exceeds total demand and if a destination node demands d

units of flow, then *at least* d units and maybe more will flow to that node. Similarly, if total demand exceeds total supply and if a source node supplies s units of flow, then *at least* s units and possibly more will flow from that node. If a transportation problem has unequal total supply and total demand, and you do not specify either the THRUNET or NOTHRUNET option, then PROC TRANS assumes that the problem is infeasible and does not proceed with the optimization. See the “Balancing Total Supply and Total Demand” section on page 520 for more information.

THRUNET | ADDSUPPLY

tells the procedure to force through the network any excess supply (the amount that total supply exceeds total demand) or any excess demand (the amount that total demand exceeds total supply), as is required. If a transportation problem has unequal total supply and total demand, and you do not specify either THRUNET or NOTHRUNET, then PROC TRANS assumes that the problem is infeasible and does not proceed with the optimization. See the “Balancing Total Supply and Total Demand” section on page 520 for more information.

HEADNODE Statement

HEADNODE *variables* **;**

The HEADNODE statement is a list of SAS variables in the COST= data set. Some or all of these variables are in the CAPACITY= data set and the MINFLOW= data set. The names of these SAS variables are also the names of the destination nodes in the transportation problem.

In the COST= data set, the value of a HEADNODE list variable in an observation that does not contain demand data is the cost of transporting a unit of flow from the source node named in that observation’s TAILNODE list to the destination node associated with the name of the HEADNODE list variable. HEADNODE variables in the MINFLOW= data set and CAPACITY= data set are similarly associated with destination nodes. In these data sets, the HEADNODE list variables have as values the upper and lower bounds on arc flows.

If the HEADNODE statement is not specified, PROC TRANS forms a HEADNODE list of all numeric variables in the COST= data set not included in the explicit or implicit list specifications.

SUPPLY Statement

SUPPLY *variable* **;**

The SUPPLY statement identifies the variable in the COST= data set that contains the number of units of supply at each supply node. The SUPPLY statement is required. Values of the SUPPLY variable must be numeric.

TAILNODE Statement

TAILNODE *variable* **;**

The TAILNODE statement identifies the variable in all input data sets that names each of the source nodes. This variable is included in the output data set. The TAILNODE statement is required, and the values of the TAILNODE variable must be character.

Details

Missing Values

A missing value in a HEADNODE list variable in the COST= data set indicates that the arc from that observation's TAILNODE list source node to the corresponding destination node does not exist. A missing supply or demand value is interpreted as zero.

A missing value in the MINFLOW= data set and CAPACITY= data set causes the procedure to assign the default capacity or default minimum flow to the corresponding arc, if it exists.

A missing value in the OUT= data set is used when the arc does not exist.

Output Data Set

The output data set contains the variables listed in the HEADNODE, TAILNODE, and SUPPLY statements, and an additional variable named _DUAL_. See the "Dual Variables" section on page 519. For each observation in the COST= data set that is associated with a source node, the output data set tells you

- the optimal flow between the source and destination nodes
- the name of the source node as given in the TAILNODE variable
- the value of the dual variable associated with that source node.

The demand data specified in the COST= data set are also included in the same observation as they are in the COST= data set. The TAILNODE variable has the value _DEMAND_ in this observation. An observation labeled with the TAILNODE variable taking the value _DUAL_ contains the dual variables at the destination nodes.

Objective Value

If the problem is infeasible (see the "Reasons for Infeasibility" section on page 520), a note to that effect is written to the SAS log. Otherwise, the value of the objective function, the total cost (the multiple of flow and cost, summed for all arcs) at optimality, is reported on the SAS log.

Demand

The demand at each destination node must be specified. Because there are the same number of destination nodes as there are HEADNODE statement variables, PROC TRANS assumes that the values of the first observation contain the number of units demanded at each destination node. If DEMAND=*d* is specified in the PROC TRANS statement, then observation *d* is assumed to contain the number of units demanded at each destination node.

Dual Variables

Let $\pi_i = 1, ..., n$ be the dual variable values of the source nodes, $\pi_j = 1, ..., m$ be the dual variable values of the destination nodes, and c_{ij} be the cost of unit flow on the arc between source *i* and destination *j* . Then

$$r_{ij} = \pi_i - \pi_j - c_{ij}$$

is the reduced cost of the arc between nodes *i* and *j*. This is the amount by which the total cost increases if flow through arc (*ij*) is increased by one unit. The total cost decreases by r_{ij} if the flow through arc (*i*,*j*) is decreased by one unit. Dual variables are saved in the OUT= data set.

Macro Variable _ORTRANS

On termination, the TRANS procedure defines a macro variable named _ORTRANS. This variable contains a character string that indicates the status of the procedure on termination and gives the objective value at termination. The form of the _ORTRANS character string is

STATUS=*xxx* **OBJECTIVE=***total-cost-of-the-solution*

where *xxx* can be one of the following:

- SUCCESSFUL
- INFEASIBLE
- INFEASIBLE_SUPPLY>DEMAND
- INFEASIBLE_SUPPLY<DEMAND
- MEMORY_ERROR
- IO_ERROR
- SYNTAX_ERROR
- SEMANTIC_ERROR
- BADDATA_ERROR
- UNKNOWN_ERROR

This information can be used when PROC TRANS is one step in a larger program that needs to identify how the TRANS procedure terminated. Because _ORTRANS is a standard SAS macro variable, it can be used in the ways that all macro variables can be used, (refer to the "*SAS Guide to Macro Processing*"). One way to write the _ORTRANS variable to the log is illustrated in Example 6.1.

Reasons for Infeasibility

By default, PROC TRANS assumes that all transportation problems have total supply equal to total demand. The THRUNET and NOTHRUNET options enable you to relax this assumption.

A transportation problem is infeasible if nodal flow conservation constraints cannot be met so that the flow into a node plus its supply does not equal the flow out of the node plus its demand. Instances of this type of infeasibility occur when

- the supply of a source node exceeds the total capacities of arcs leading out of it
- the supply of a source node does not meet the total lower flow bounds of arcs leading out of it
- the demand of a sink node exceeds the total capacities of arcs leading into it
- the demand of a sink node does not meet the total lower flow bounds of arcs leading into it

If a solution does not exist where all arcs have flow on or between their lower flow bounds or capacities, the problem is infeasible.

Balancing Total Supply and Total Demand

When Total Supply Exceeds Total Demand

When total supply of a transportation problem exceeds total demand, PROC TRANS can add an extra destination node (called the *excess node*) to the problem and set the demand at that node equal to the difference between total supply and total demand. There are two ways that this extra destination node can be added to the transportation network.

Figure 6.3 shows a network in which the NOTHRUNET option is specified. The sum of flows that reach the destination nodes equals the total demand of the transportation problem. For some source nodes, supply can be conveyed through *real* arcs but can also be drained away to the excess node. The supply of each source node is really an upper bound of the number of flow units such a node can actually supply.

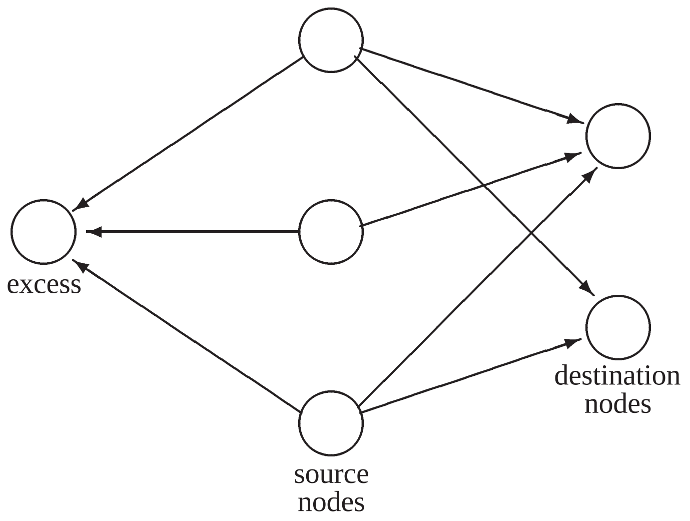

Figure 6.3. Using the NOTHRUNET Option when Total Supply Exceeds Total Demand

Figure 6.4 illustrates a network in which the THRUNET option is used. The sum of flows that reach the destination nodes equals the total supply of the transportation problem.

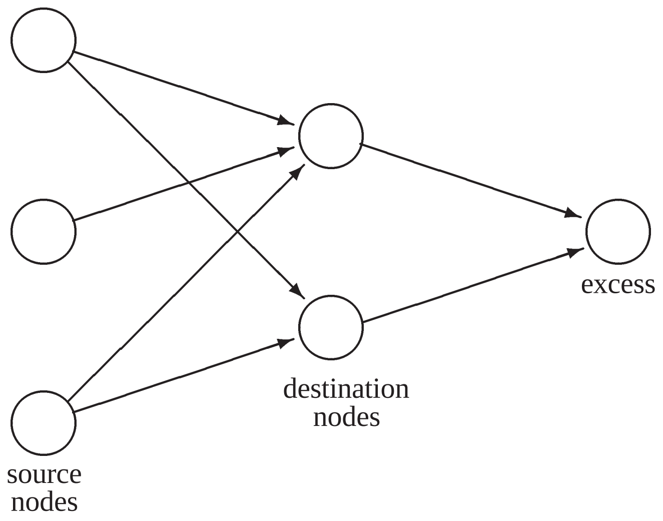

Figure 6.4. Using the THRUNET Option when Total Supply Exceeds Total Demand

For some destination nodes, the amount of flow can exceed that node's demand. The demand of destination nodes is a lower bound of the number of flow units a destination node can demand.

When Total Demand Exceeds Total Supply

When total demand exceeds total supply, PROC TRANS can add an extra source node (the excess node) to the problem. This node is able to supply to the network the difference between total demand and total supply. There are two ways this extra source node can be added to the transportation network.

Figure 6.5 shows a network in which the NOTHRUNET option is specified. The sum of flows that leave the source nodes equals the total supply of the transportation problem. For some destination nodes, demand will be satisfied by flow through *real* arcs and can be satisfied by flows through arcs directed from the excess node. The demand of each destination node is really an upper bound of the number of flow units such a node can actually receive.

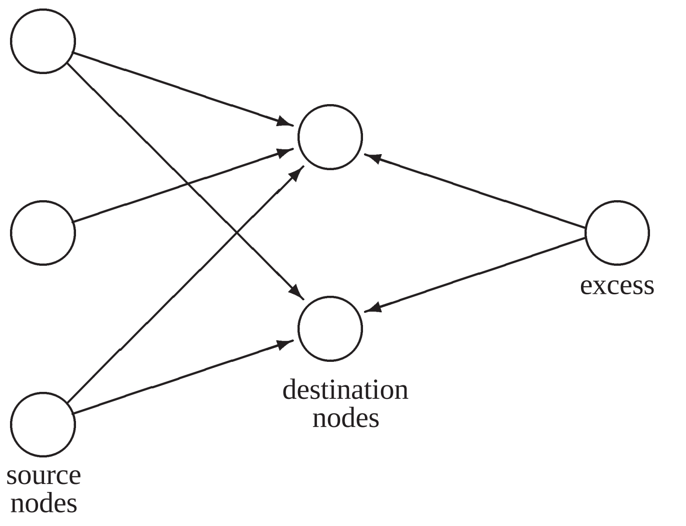

Figure 6.5. Using the NOTHRUNET Option when Total Demand Exceeds Total Supply

Figure 6.6 illustrates a network in which the THRUNET option is specified. The sum of flows that leave the source nodes equals the total demand of the transportation problem. For some source nodes, the sum of flow that is conveyed through arcs originating at that node can exceed the node's supply capability. The supply of source nodes is a lower bound of the number of flow units a source node is able to supply to the network.

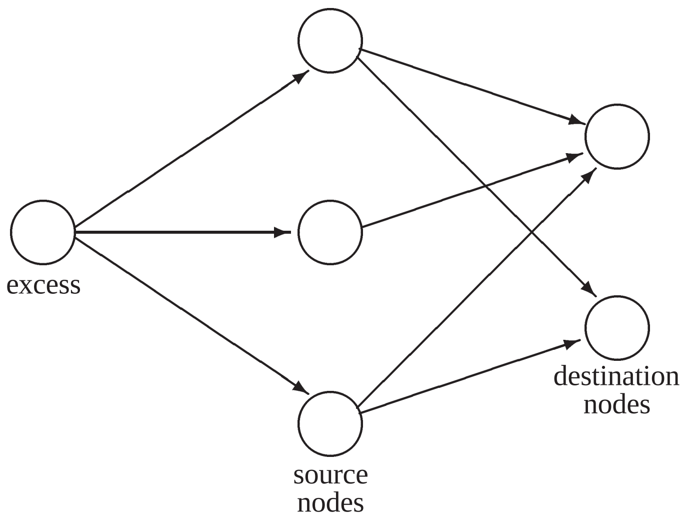

Figure 6.6. Using the THRUNET Option when Total Demand Exceeds Total Supply

If total supply does not equal total demand and either the THRUNET or NOTHRUNET option is specified, PROC TRANS automatically creates the excess node and the arcs incident to it. When the optimization is complete, these ancillary parts of the network are erased. Information about these parts will not be found in the OUT= data set.

Examples

Example 6.1. Uncapacitated Transportation Network

The transportation problem described in the “Getting Started” section is solved next. The cost data are stored in the following SAS data set. The solution is stored in a SAS data set as shown in Output 6.1.1 and displayed with PROC PRINT.

```
title 'Uncapacitated Transportation Network';

data cst;
   input Atlanta Chicago Denver Houston Los_Ange Miami
      New_York San_Fran Seattle Washingt supply city$;
   datalines;
 50  75  89   8  27  39  64 100  50   8    .        .
 20  58 121  70 193  60  74 213 218  54   10    Atlanta
 58  20  92  94 174 118  71 185 173  57  150    Chicago
121  92  20  87  83 172 163  94 102 149   90    Denver
 70  94  87  20 137  96 142 154 189 122   27    Houston
193 174  83 137  20 223 245  34  95 230   80    Los_Ange
 60 118 172  96 233  20 109 259 273  92   26    Miami
 74  71 163 142 245 109  20 257 240  20   80    New_York
213 185  94 164  34 259 257  20  67 244   25    San_Fran
218 173 102 189  95 273 240  67  20 232    7    Seattle
 54  59 149 122 230  92  20 244 232  20   15    Washingt
;

proc trans cost=cst;
   TAILNODE city;
   HEADNODE Atlanta--Washingt;
   SUPPLY supply;
run;

proc print;
run;
```

After this program executes, the following message is written to the SAS log:

```
NOTE: Optimal Solution total = 22928.
```

Output 6.1.1. Uncapacitated Transportation Network Solution

```
                                         L        N    S         W
                    A   C        H       o        e    a    S    a
              s     t   h    D   o       s        w    n    e    s      _
              u     l   i    e   u       _   M    _    _    a    h      D
        c     p     a   c    n   s       A   i    Y    F    t    i      U
 O      i     p     n   a    v   t       n   a    o    r    t    n      A
 b      t     l     t   g    e   o       g   m    r    a    l    g      L
 s      y     y     a   o    r   n       e   i    k    n    e    t      _

 1  _DEMAND_    .  50  75   89   8   27   39  64  100   50   8     .
 2  Atlanta    10  10   0    0   0    0    0   0    0    0   0   -51
 3  Chicago   150  30  75    2   0    0    0   0    0   43   0   -13
 4  Denver     90   0   0   87   0    0    0   0    3    0   0   -85
 5  Houston    27   0   0    0   8   19    0   0    0    0   0   -28
 6  Los_Ange   80   0   0    0   0    8    0   0   72    0   0  -145
 7  Miami      26   0   0    0   0    0   26   0    0    0   0   -89
 8  New_York   80   0   0    0   0    0    8  64    0    0   8     0
 9  San_Fran   25   0   0    0   0    0    0   0   25    0   0  -159
10  Seattle     7   0   0    0   0    0    0   0    0    7   0  -166
11  Washingt   15  10   0    0   0    0    5   0    0    0   0   -17
12  _DUAL_      .  71  33  105  48  165  109  20  179  186  20     .
```

Because the first observation is associated with the demands, the TAILNODE value is _DEMAND_. If you had specified DEMAND=*d*, the *d*th observation would have contained demand information. The last observation has a TAILNODE value of _DUAL_, and the values of the HEADNODE variables are the dual variables at the destination nodes, or the marginal costs of increasing demand at a destination node.

The other observations in the OUT= data set contain the optimal flows. For example, the third observation contains information associated with the Chicago source node. The values of the HEADNODE variables in this observation give the optimal flow between Chicago and each destination node. To achieve the minimum cost, the Chicago supply must be sent to four destinations: 30 units to Atlanta, 75 units to Chicago, 2 units to Denver, and 43 units to Seattle. The SUPPLY variable has the supply capability of this source node (150 units).

The _DUAL_ variable value (-13) for Chicago is the amount the total solution cost would increase (because of the negative sign; in this case, the total cost would decrease by $13) if Chicago supplies an extra unit of flow. (If you were to increase the supply of Chicago to 151 and rerun PROC TRANS, specifying the NOTHRUNET option, the total cost would change from 22928 to 22915.)

The macro variable _ORTRANS, defined by PROC TRANS, contains information regarding the termination of the procedure. This information can be useful when PROC TRANS is part of a larger SAS program. This information can be written to the log using the macro language with the statement

```
   %put &_ORTRANS;
```

The following message is written to the SAS log:

```
   STATUS=SUCCESSFUL  OBJECTIVE=22928
```

Example 6.2. Capacitated Transportation Network

In this example, the optimal flow is found on a capacitated transportation network. Suppose that there are upper bounds on the amount that can be shipped within each city. The following SAS program and output show how this capacity constraint is included in the model:

```
title 'Capacitated Transportation Network';

data capcty;
   input Atlanta Chicago Denver Houston Los_Ange Miami
      New_York San_Fran Seattle Washingt city$;
   datalines;
10  .   .  .  .  .  .  .  .  . Atlanta
 . 60   .  .  .  .  .  .  .  . Chicago
 .  . 100  .  .  .  .  .  .  . Denver
 .  .   . 10  .  .  .  .  .  . Houston
 .  .   .  . 30  .  .  .  .  . Los_Ange
 .  .   .  .  . 20  .  .  .  . Miami
 .  .   .  .  .  . 75  .  .  . New_York
 .  .   .  .  .  .  . 25  .  . San_Fran
 .  .   .  .  .  .  .  . 10  . Seattle
 .  .   .  .  .  .  .  .  . 10 Washingt
;

proc trans cost=cst capacity=capcty;
   HEADNODE Atlanta--Washingt;
   TAILNODE city;
   supply supply;
run;

proc print;
run;
```

After this program executes, the following message is written to the SAS log:

```
NOTE: Optimal Solution Total = 24036.
```

The preceding statements produce the SAS data set in Output 6.2.1

Output 6.2.1. Capacitated Transportation Network Solution

```
                Capacitated Transportation Network with MINFLOW

                                                  L          N     S           W
                          A    C         H        o          e     a     S     a
                     s    t    h    D    o        s          w     n     e     s      _
                     u    l    i    e    u        _    M     _     _     a     h      D
       c             p    a    c    n    s        A    i     Y     F     t     i      U
 O     i             p    n    a    v    t        n    a     o     r     t     n      A
 b     t             l    t    g    e    o        g    m     r     a     l     g      L
 s     y             y    a    o    r    n        e    i     k     n     e     t      _

 1   _DEMAND_        .   50   75   89    8       27   39    64   100    50     8      .
 2   Atlanta        10    0    0    0    0        0   10     0     0     0     0    -53
 3   Chicago       150   44   60    3    0        0    0     0     0    43     0     -2
 4   Denver         90    0    0   86    0        0    0     0     4     0     0    -74
 5   Houston        27    0    0    0    8       18    1     0     0     0     0    -17
 6   Los_Ange       80    0    0    0    0        9    0     0    71     0     0   -134
 7   Miami          26    6    0    0    0        0   20     0     0     0     0      0
 8   New_York       80    0    8    0    0        0    0    64     0     0     8     -9
 9   San_Fran       25    0    0    0    0        0    0     0    25     0     0   -148
10   Seattle         7    0    0    0    0        0    0     0     0     7     0   -155
11   Washingt       15    0    7    0    0        0    8     0     0     0     0    -21
12   _DUAL_          .   60   80   94   37      154  113    29   168   175    29      .
```

Note that the optimal objective value is greater in the capacitated network (24036) than in the uncapacitated network (22928). Additional constraints can never decrease the objective value of a minimization problem at optimality. Also observe that the flow within Chicago, Miami, and San_Fran are at their limits. The rerouting of flow within these cities accounts for the increase in cost.

Example 6.3. Capacitated Transportation Network with MINFLOW

Suppose you place a minimum on the flow within each city. Just as capacity restrictions can be interpreted as limits on available transportation, minimum flow restrictions can be interpreted as requirements to ship minimum quantities on certain routes, perhaps as a result of contractual agreements. The following program adds minimum flow requirements on four routes. Because the MINFLOW= data set contains many missing values, named input mode is used to input the data. The solution is displayed following the program in Output 6.3.1.

```
title 'Capacitated Transportation Network with MINFLOW';

data minflw;
   input Chicago= Denver=  San_Fran= Seattle= city= $;
   datalines;
city=Chicago Chicago=30 San_Fran=40 Seattle=50
city=Denver  Denver=40
;

proc trans cost=cst capacity=capcty minflow=minflw;
   HEADNODE Atlanta--Washingt;
   TAILNODE city;
   supply supply;
run;
```

```
proc print;
run;
```

The SAS log contains the following message:

```
NOTE: Optimal Solution Total = 31458.
```

Output 6.3.1. Capacitated Transportation Network Solution with MINFLOW

```
                                        L         N    S         W
                     A    C        H    o         e    a    S    a
                s    t    h   D    o    s         w    n    e    s     _
                u    l    i   e    u    _    M    _    _    a    h     D
       c        p    a    c   n    s    A    i    Y    F    t    i     U
 O     i        p    n    a   v    t    n    a    o    r    t    n     A
 b     t        l    t    g   e    o    g    m    r    a    l    g     L
 s     y        y    a    o   r    n    e    i    k    n    e    t     _

 1 _DEMAND_     .   50   75  89    8   27   39   64  100   50    8     .
 2 Atlanta     10   10    0   0    0    0    0    0    0    0    0  -138
 3 Chicago    150    0   60   0    0    0    0    0   40   50    0  -100
 4 Denver      90   11    8  71    0    0    0    0    0    0    0   -37
 5 Houston     27    0    0   0    8    0   19    0    0    0    0   -91
 6 Los_Ange    80    0    0  18    0   27    0    0   35    0    0    26
 7 Miami       26    6    0   0    0    0   20    0    0    0    0   -98
 8 New_York    80    8    0   0    0    0    0   64    0    0    8   -84
 9 San_Fran    25    0    0   0    0    0    0    0   25    0    0    12
10 Seattle      7    0    7   0    0    0    0    0    0    0    0    44
11 Washingt    15   15    0   0    0    0    0    0    0    0    0  -104
12 _DUAL_       .  158  129  57  111   -6  187  104    8  -24  104     .
```

Note that the optimal objective value is greater in the minimum flow capacitated network than in the capacitated network. Additional constraints can never decrease the objective value of a minimization problem at optimality.

Example 6.4. An Infeasible Problem

This example shows what happens when the total demand exceeds the total supply. The data from Example 6.1 are used with the demand at Atlanta increased to 100 units. Consequently, the demand exceeds the supply by 50 units. When the statements

```
proc trans cost=cst;
   TAILNODE city;
   HEADNODE Atlanta--Washingt;
   supply supply;
run;
```

are executed, the following message is written to the SAS log:

```
ERROR: Infeasible network. Total supply 510 does not equal
       total demand 560.
       Use either the THRUNET or the NOTHRUNET option
       on the PROC TRANS statement.
```

However, if the THRUNET option is specified in the PROC TRANS statement, the procedure distributes the supply optimally among the source nodes. In that case, the statements

```
title 'Using the THRUNET Option';

proc trans data=cst THRUNET;
   TAILNODE city;
   HEADNODE Atlanta--Washingt;
   supply supply;
run;

proc print;
run;
```

produce the following message on the SAS log and display the solution in Output 6.4.1.

```
NOTE: Optimal Solution Total = 18302.
```

Output 6.4.1. Using the THRUNET Option

```
                            Using the THRUNET Option

                                             L       N    S         W
                      A   C        H    o       e    a    S    a
                  s   t   h    D   o    s       w    n    e    s     _
                  u   l   i    e   u    _   M   _    _    a    h     D
      c           p   a   c    n   s    A   i   Y    F    t    i     U
 O    i           p   n   a    v   t    n   a   o    r    t    n     A
 b    t           l   t   g    e   o    g   m   r    a    l    g     L
 s    y           y   a   o    r   n    e   i   k    n    e    t     _

 1  _DEMAND_    .  50  75   89   8   27   39  64  100   50   8      .
 2  Atlanta    10  10   0    0   0    0    0   0    0    0   0    -51
 3  Chicago   150  30  75    2   0    0    0   0    0   43   0    -13
 4  Denver     90   0   0   87   0    0    0   0    3    0   0    -85
 5  Houston    27   0   0    0   8   19    0   0    0    0   0    -28
 6  Los_Ange   80   0   0    0   0    8    0   0   72    0   0   -145
 7  Miami      26   0   0    0   0    0   26   0    0    0   0    -89
 8  New_York   80   0   0    0   0    0    8  64    0    0   8      0
 9  San_Fran   25   0   0    0   0    0    0   0   25    0   0   -159
10  Seattle     7   0   0    0   0    0    0   0    0    7   0   -166
11  Washingt   15  10   0    0   0    0    5   0    0    0   0    -17
12  _DUAL_      .  71  33  105  48  165  109  20  179  186  20      .
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/OR® User's Guide: Mathematical Programming, Version 8*, Cary, NC: SAS Institute Inc., 1999. 566 pp.

SAS/OR® User's Guide: Mathematical Programming, Version 8

ISBN 1–58025–491–8

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999