LOGWARE

SELECTED COMPUTER PROGRAMS FOR LOGISTICS/SUPPLY CHAIN PLANNING

Version 5.0

Ronald H. Ballou

Weatherhead School of Management
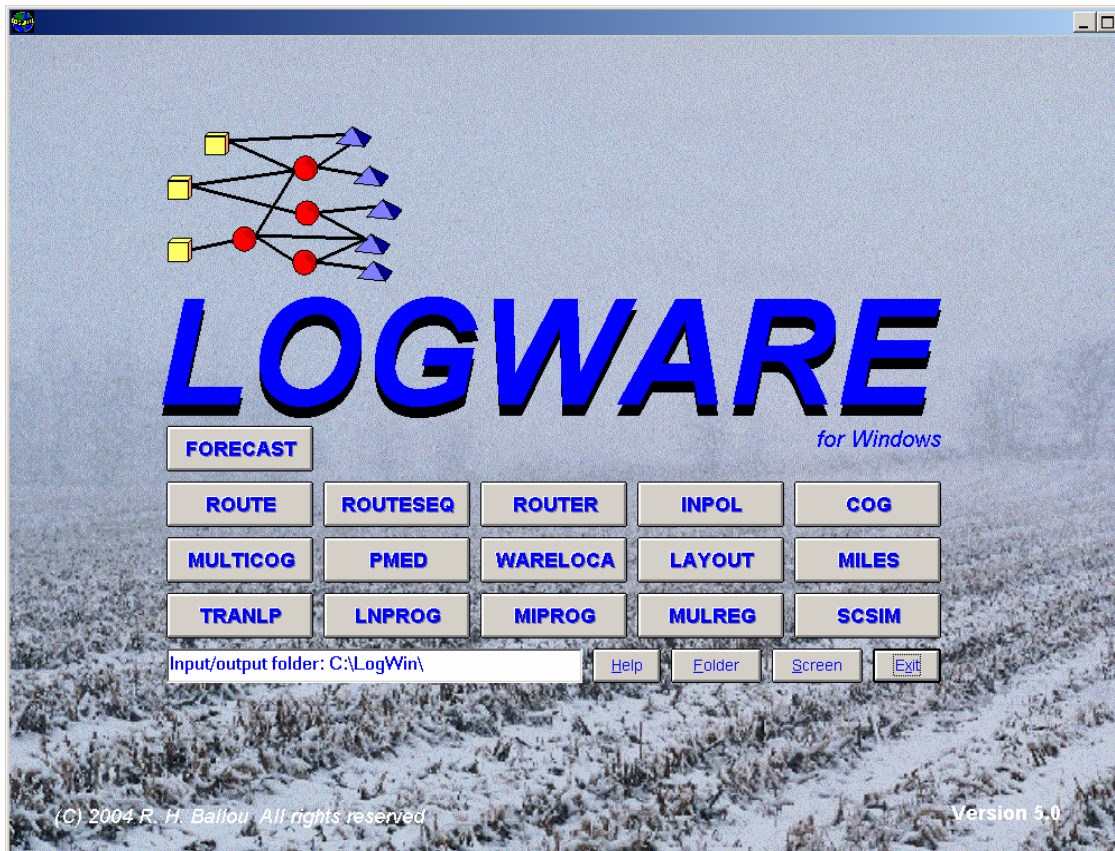**Case Western Reserve University**

LOGWARE

SELECTED COMPUTER PROGRAMS
FOR LOGISTICS/SUPPY CHAIN PLANNING

LOGWARE is a collection of selected software programs that is useful for analyzing a variety of logistics/supply chain problems and case studies. It contains the following modules.

Each module is selected from the following master screen by clicking on the appropriate button.

LOGWARE

for Windows

FORECAST

ROUTE	ROUTESEQ	ROUTER	INPOL	COG
MULTICOG	PMED	WARELOCA	LAYOUT	MILES
TRANLP	LNPROG	MIPROG	MULREG	SCSIM

Input/output folder: C:\LogWin\   Help   Folder   Screen   Exit

Version 5.0

HARDWARE REQUIREMENTS

LOGWARE is designed for microcomputers operating under WINDOWS 98, NT, 2000, or XP. At least 16MB of RAM should be installed. Hard drive space of at least 10MB should be available. A color monitor capable of producing at least 640x480 pixels resolution is needed, although 800x600 is better and 1024x768 is preferred. Resolutions greater than 1024x768 pixels are not supported. A laser printer is preferred. A mouse is needed. A 3½″ floppy drive and/or a compact disk reader are needed.

INSTALLING THE SOFTWARE ON A HARD DRIVE

Place the program compact disks in the appropriate drives. In WINDOWS, click on the **Start** button and then select the **Run** option from pop-up menu. Type "**X:Setup.exe**" ("X" being the letter designated for your CD drive) . The program may also be installed with Windows' **Start, Settings, Control Panel, Add/Remove Programs, Install** option. Change the subdirectory under which the program will be installed if the default subdirectory is not desired.

RUNNING THE PROGRAMS

After the program is installed, click on the **Start** button and select **Programs**. Choose the **Logware** icon to activate the program. Click on the desired program module. A shortcut icon on the Desktop may also be created.

EDITING THE DATA

In those modules where a screen data editor is present, the first action is to open a data file by clicking on the module's **Start** button. If a file is named that is not in the current list of files, a data shell will be created into which a new problem may be entered. The use of the editor is simple and transparent with a little practice; however, a few comments about its use may help to get started.

- Press the **Ins** key to start a new line of data in a matrix. The normal action is to insert a text row at the end of the matrix. The **Add** button may also be pressed. This will allow a row to be added at the end of the matrix as well as within the matrix. Position the cursor in the matrix row where the row is to be added.
- Pressing the **Esc** key clears a matrix cell.
- Pressing the **Delete** button deletes the row in a matrix highlighted by the current cursor position.
- If **Column arithmetic** is to be used, highlight the matrix column on which the action is to apply.

Alternatively, the data for each module except SCSIM may be created and edited with the use of Excel. It is expected that the user has a basic knowledge of Excel use.

COPYING THE INSTRUCTIONS AND THE SOFTWARE

This software and the associated instructions may be copied as long as they are used for educational purposes. All copied materials must display the following copyright notices.

Professor Ronald H. Ballou
Weatherhead School of Management
Case Western Reserve University
Cleveland, OH 44106 USA
Tel: (216) 368-3808
Fax: (216) 368-6250
E-mail: Ronald.Ballou@Weatherhead.CWRU.edu

Up to date information about the software may be found at www.PrenHall.com/Ballou.

INSTRUCTIONS FOR EXPONENTIAL SMOOTHING
AND TIME SERIES DECOMPOSITION FORECASTING
FORECAST

FORECAST is computer software that forecasts from time series data by means of exponential smoothing and/or time series decomposition methods. In logistics/SC, such time series may be product sales, lead times, prices paid for goods, or shipments. The philosophy of time series forecasting is to project an historical pattern of the data over time, and, if present, account for trend and seasonality. Exponential smoothing is a moving average approach that projects the average of the most recent data and adapts the forecast to changing data as they occur. On the other hand, the time series decomposition approach recognizes that major reasons for variation in data over time are due to trend and seasonal components. Each of these is estimated and combined to produce a forecast. For background information on the forecast models used in **FORECAST**, see Chapter 8 of the Business Logistics/Supply Chain Management 5e textbook.

To run **FORECAST**, select the appropriate module from the **LOGWARE** master menu. Open an existing file or select a new one. Prepare or change the database. Select the appropriate model type, which may be either some form of an exponential model (Level only, Level-Trend, etc.) or the time series decomposition model. Click on **Solve** to generate a forecast.

INPUT
The input to both forecasting modules consists of the time-ordered series of data, ranked from the most historic to the most recent observations, and various parameter values that guide the execution of the models. The dimensions of the models allow observations for up to 200 periods and a forecast of up to 50 periods. Both model types run from the same database although some of the parameters are not used in the time series decomposition model.

Parameters and Labels
This portion of the screen sets the parameters for both the exponential smoothing and time series decomposition models. These guide the overall action of the models. Consider each element on this screen.

Problem label. This is a label given to the problem you are solving. WARNING: Do not use commas (,) or double quotation marks (") in the label since this will cause an error in reading the data file.

Number of data points. Specify the number of data periods in the time series. Up to 200 points are allowed. Be sure that the number of points specified here matches the number of data points actually entered in the time series.

Initialization period. The initialization period is the number of the oldest data points used to determine starting values for the exponential smoothing model. A minimum of 3 periods of data should be declared for this purpose. If a seasonal model is to be used, at least the number of periods in one full seasonal cycle must be specified.

Error statistics. The number of data periods needed to compute forecast error statistics is referred to as the validation period. These error statistics are the mean

absolute deviation (*MAD*), the bias (*BIAS*), and the root mean square error (*RMSE*). The validation period is the last *N* periods of data. Enough data points should be used from this validation period to strike a reasonable average for these statistics.

MAD is defined as the average of the absolute differences between the actual values and the forecast values for the validation period. *BIAS* is the average of the differences between actual and forecast values for the validation period. *RMSE* is square root of the average of the squared differences between the actual and the forecast values for the validation period.

Model type. Selecting the model type refers to the exponential smoothing model or the time series decomposition model. There are four variations of the exponential smoothing model for best representing the character of the time series. These are the Level only, Level-Trend, Level-Seasonal, and Level-Trend-Seasonal. Select the type that best represents the data. Alternately, select the time series decomposition model.

Smoothing constant search. When one of the forms of the exponential model is selected, indicate whether a search for the smoothing constants is to be performed using **FORECAST**. If not, the smoothing constants for the model type selected must be specified.

If **FORECAST** is to search for the smoothing constants, indicate the smoothing constant increment for searching. **WARNING:** Significant running time may result from too small a search increment, especially for the more complicated model forms such as the Level-Trend-Seasonal model. An increment of 0.1 works reasonably well. Be prepared to wait when smaller increments are chosen.

Seasonal length. When using seasonality in either the exponential smoothing or time series decomposition models, indicate the number of data periods representing a full seasonal cycle. **WARNING:** The initialization period should be a seasonal cycle plus 2 data periods. If a model type without seasonality is used, specify a seasonal length of 0. However, be sure to use at least three data periods for initialization.

Forecast length. Up to 50 future periods may be specified for forecasting. However, remember that these are historical projection methods such that forecasting beyond 6 months or a full seasonal cycle can lead to a significant forecasting error.

Time Series Data

This screen section is for entering time series data by period. A period may be any time segment such as a day, week, month, or quarter.

Period label. A time segment, or period, may be given an identification label. Good practice is to keep the number of letters or numbers to less than or equal to 15. WARNING: Avoid using commas (,) or double quotation marks (") in the labels.

Observations. All time series data should be entered chronologically, with period 1 representing the oldest observation. Up to 200 data points are allowed. Although larger numbers are permitted, scaling data so that entries are not larger than 6 digits is good practice.

EXAMPLE

The prices for a certain purchased component have been observed over one year and a half. These prices are as follows.

Period, mo.	Price, $/unit	Period, mo.	Price, $/unit
2000, Jan.	19.36	2001, Jan.	32.52
Feb.	25.45	Feb.	31.33
Mar.	19.73	Mar.	25.32
Apr.	21.48	Apr.	27.53
May	20.77	May	26.38
June	25.42	June	23.72
July	23.79	July	29.14
Aug.	28.35	Aug	?
Sep.	26.80	Sep.	?
Oct.	25.32	Oct.	?
Nov.	25.22	Nov.	?
Dec.	27.14	Dec.	?

The prices are to be forecast to the end of 2001.

A full seasonal cycle is 12 months. Therefore, the initialization period is set at $12 + 2 = 14$ periods. Three periods are selected for validation. A Level-Trend-Seasonal exponential smoothing model is to be tested with a level smoothing constant (α) = 0.2, a trend smoothing constant (β) = 0.3, and a seasonal smoothing constant (γ) = 0.1. The data screen is prepared as shown in Figure FORECAST-1.

Once the data have been prepared, select the **Solve** button. A validation check of the data will be made and the data processed according to the selected model. The results are shown in Figure FORECAST-2. A graphical display of the data and the forecast can be seen in Figure FORECAST-3.

Figure FORECAST-1 Data Screen for the Example Problem.

```
                        TIME SERIES FORECASTING
                    Curve fitting and model validation

TREND EQUATION: 21.56 +.3958T

MODEL TYPE: Time Series Decomposition
Prd Period                                            Sea-
no. label----------     Actual    Forecast    Trend    sonal        Error
 15 MAR 2001             25.32     23.85       27.49    .92           1.47
 16 APR 2001             27.53     25.89       27.89    .99           1.64
 17 MAY 2001             26.38     24.96       28.28    .93           1.42
 18 JUNE 2001            23.72     30.47       28.68    .83          -6.75
 19 JULY 2001            29.14     28.43       29.08    1.00           .71


THE FORECAST
MAD = 2.96 BIAS = -1.54 RMSE = 4.00
The forecasts for periods 20 to 24 are:
Period          Forecast
    20           33.80
    21           31.87
    22           30.03
    23           29.84
    24           32.04
```
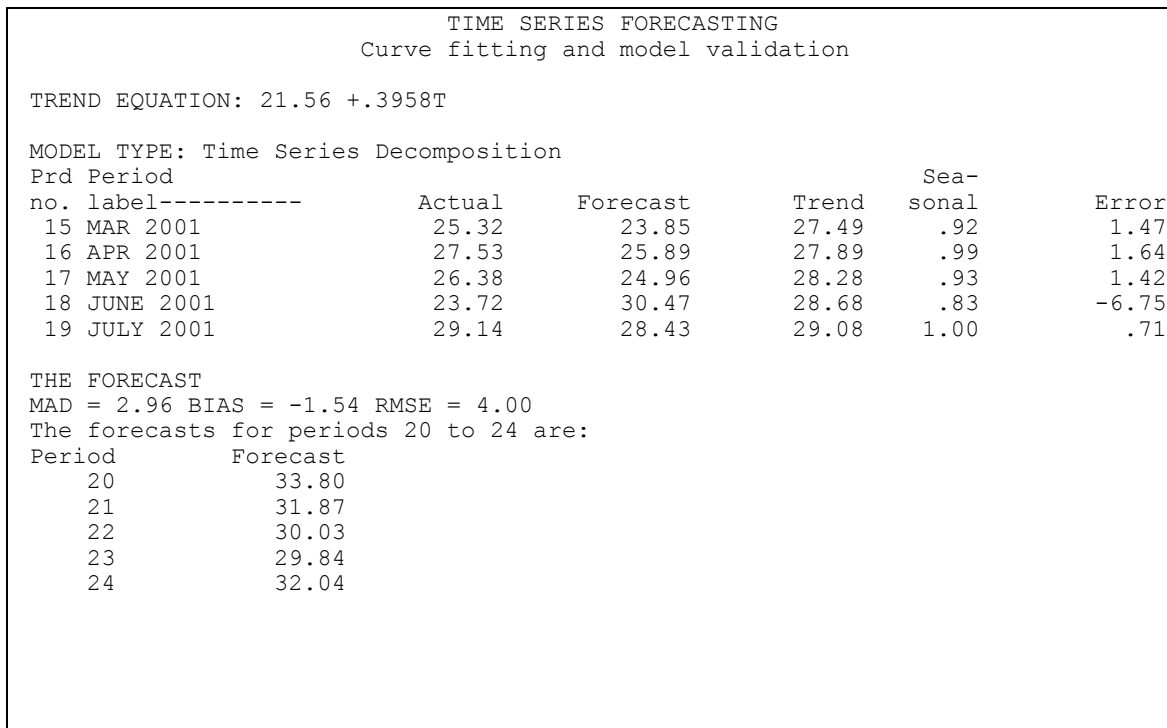
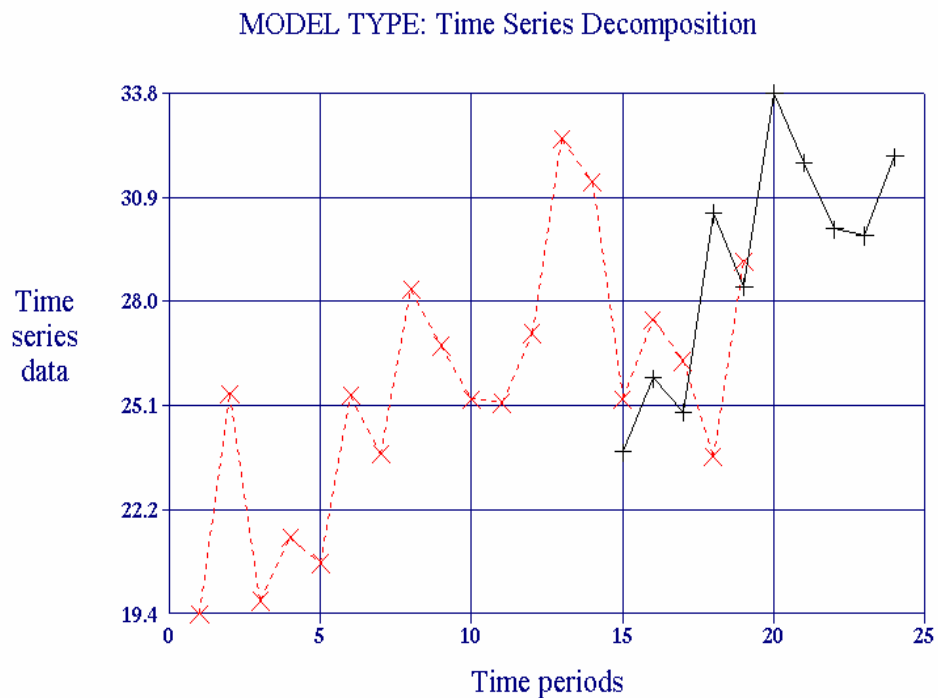**Figure FORECAST-2** Solution Report for the Example Problem.

Figure FORECAST-3 Time Series Data and the Forecast.

INSTRUCTIONS FOR THE
SHORTEST ROUTE PROGRAM
ROUTE

ROUTE is a program to find the shortest route through a network of nodes connected by arcs, which usually are roads, rail lines, streets, pipes, or wires. A path, or arc, is a connection between nodal pairs over which travel may occur. A route is a set of arcs connected head to tail for traveling from one node on the network to another. Since there are usually many ways between points on a well-connected network, we seek the route minimizing travel distance or time between two selected points. **ROUTE** finds *all* of the minimum-cost routes from a specified origin node to all other nodes in the network. For background information on the shortest route method used in **ROUTE**, see Chapter 7 of the Business Logistics/Supply Chain Management 5e textbook.

INPUT

Input is managed by three sections of the data screen. The first calls for a title to the problem and identifies the origin node by number. There may be up to 500 nodes in the network.

The second section is used to create a master list of nodes and identify them with a descriptor. If a graph of the problem solution is desired, coordinate points for the nodes may be added. To determine these coordinate points, place a linear grid over a map of the network and read the *X,Y* coordinates for each node. The 0,0 coordinates must begin in the southwest corner of the grid.

The third section is used to define the available arcs between nodes and the cost (distance or time) to traverse the arcs. Nodes are numbered. Cost may also be expressed as a weighted index of both time and distance. The relative weights depend on the balance desired between the goals of shortest time and shortest distance. There may be up to 900 of these connecting arcs.

After completing the data entry, permanently save the data in a file by selecting the **Save** button and choosing a file name to receive the data. Use an input file of the form RFL*xx*.DAT, where *xx* refers to the problem number such 01, 05, 10, etc.

RUNNING *ROUTE*

After creating or retrieving data from a file, click the **ROUTE** button to solve the problem for the data as displayed on the screen. The resulting optimal paths to each destination point are shown.

You may also view a graph of the network by clicking on the **Plot** button. The problem solution may further be displayed for the current run. Select the **Solution path** button and indicate the destination point of interest. The optimal path will be highlighted.

EXAMPLE

Consider the problem as shown in Figure ROUTE-1. The major highways between Amarillo, Texas and Fort Worth, Texas are given. The approximate travel times between

nodal pairs are shown. Our task is to find the route that offers the lowest driving times to go from Amarillo, TX to Fort Worth, TX.
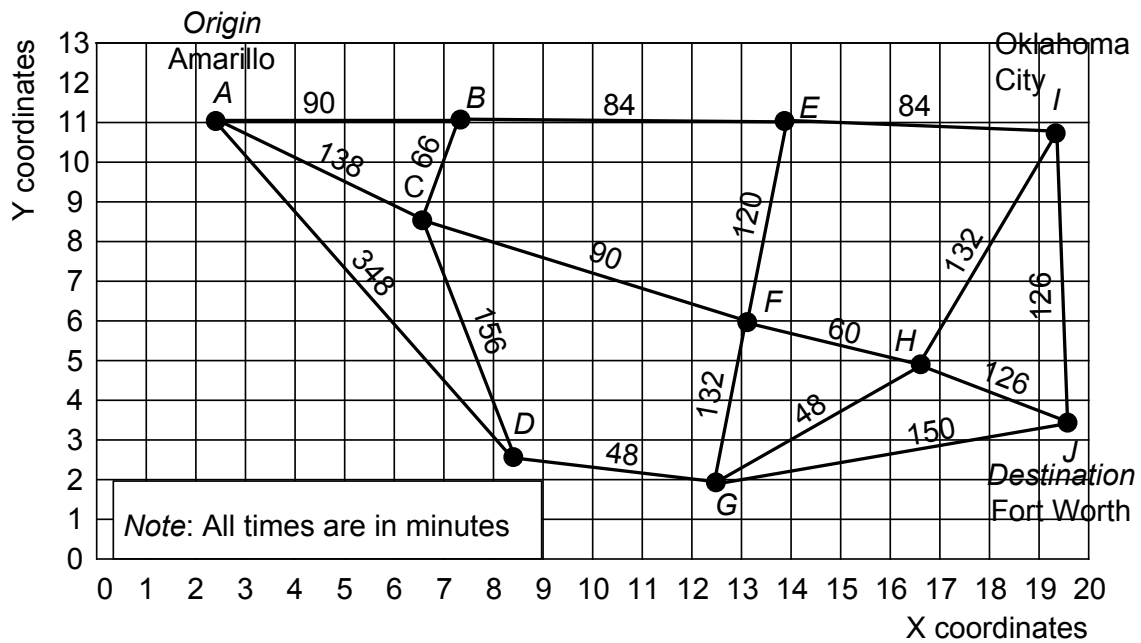
Figure ROUTE-1   Example Network of the Major Highways between Amarillo, TX and Ft. Worth, TX, with Driving Times.

We will label the nodes so that *A*=1, *B*=2, etc. The database screen segments are as follows.

Parameters and labels

```
Problem label- EXAMPLE PROBLEM

Origin node number:    1
```

Node identification and location

```
Point Node
 no.  no. Node name      X-coord  Y-coord
  1     1 A-AMARILLO        2.30    11.10
  2     2 B                 7.30    11.10
  3     3 C                 6.60     8.50
  4     4 D                 8.20     2.80
  5     5 E                13.90    11.00
  6     6 F                13.10     6.00
  7     7 G                12.50     1.90
  8     8 H                16.60     4.90
  9     9 I-OK CITY        19.20    10.70
 10    10 J-FT WORTH       19.60     3.30
```

11

Node connections

```
Point --------From----------    ----------To-----------
  no. Node no. Node name       Node no. Node name           Cost
  1          1 A-AMARILLO             2 B                    90.00
  2          1 A-AMARILLO             3 C                   138.00
  3          1 A-AMARILLO             4 D                   348.00
  4          2 B                      5 E                    84.00
  5          2 B                      3 C                    66.00
  6          3 C                      4 D                   156.00
  7          3 C                      6 F                    90.00
  8          4 D                      7 G                    48.00
  9          5 E                      9 I-OK CITY            84.00
 10          5 E                      6 F                   120.00
 11          6 F                      8 H                    60.00
 12          6 F                      7 G                   132.00
 13          7 G                      8 H                    48.00
 14          7 G                     10 J-FT WORTH          150.00
 15          8 H                      9 I-OK CITY           132.00
 16          8 H                     10 J-FT WORTH          126.00
 17          9 I-OK CITY            10 J-FT WORTH          126.00
```

```
EXAMPLE PROBLEM
Origin node number = 1    Number of nodes = 10    Number of arcs = 17

Shortest paths from origin node 1 to all destination nodes
   Cost Path
  90.00 1 -> 2
 138.00 1 -> 3
 294.00 1 -> 3 -> 4
 174.00 1 -> 2 -> 5
 228.00 1 -> 3 -> 6
 336.00 1 -> 3 -> 6 -> 8 -> 7
 288.00 1 -> 3 -> 6 -> 8
 258.00 1 -> 2 -> 5 -> 9
 384.00 1 -> 2 -> 5 -> 9 -> 10
```

The solution to this problem has the following output form.

It will take 384 minutes to drive the shortest route from Amarillo to Fort Worth. This is route $1 \Rightarrow 2 \Rightarrow 5 \Rightarrow 9 \Rightarrow 10$, or $A \Rightarrow B \Rightarrow E \Rightarrow I \Rightarrow J$ in the notation of Figure ROUTE-1. A plot of the problem with its solution is shown in Figure ROUTE-2.
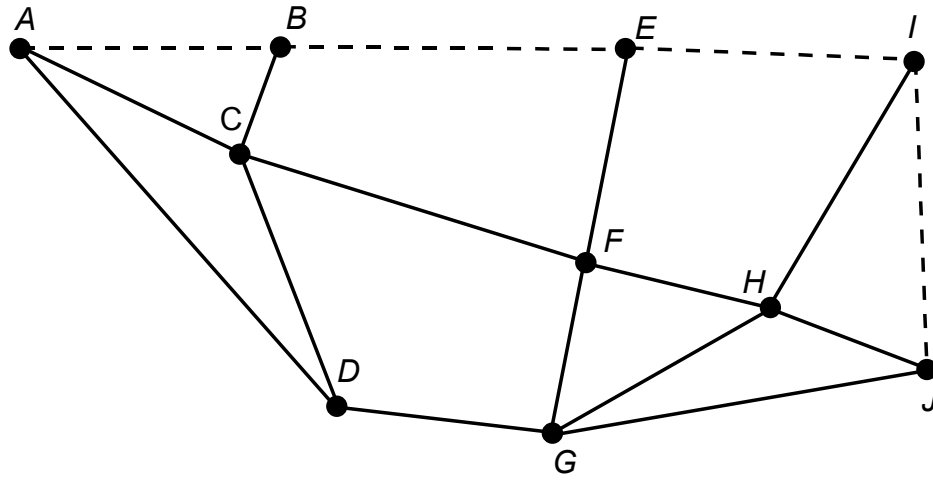
Figure ROUTE-2 Plotted Solution to Example Problem.

INSTRUCTIONS FOR THE
TRAVELING SALESMAN PROGRAM
ROUTESEQ

ROUTESEQ is a heuristic program for solving the traveling salesman problem. It will sequence up to 20 stops on a route plus an origin point. Stops and the origin point are identified with linear coordinate points. Euclidean (straight-line) distances are computed in terms of these coordinate points. For background information on the coincident origin-destination (traveling salesman) problem, see Chapter 7 of the Business Logistics/Supply Chain Management 5e textbook.

INPUT
An input file is prepared with the use of the data editor or Excel spreadsheet. There are four types of records: (1) the *X,Y* coordinates of the origin point, (2) a circuity factor, (3) a map scaling factor, and (4) the coordinate points for each stop. A typical data screen might look as follows:

Problem label:	Example		
Circuity factor: 1.21	Map scaling factor: 1		
Depot coordinates: X = 2	Y = 2		

STOP DATA

Point no.	Point label	X coordinate	Y coordinate
1	STOP 1	3	4
2	STOP 2	5	3
3	STOP 3	4	1
4	STOP 4	5	5
5	STOP 5	4	3

Figure ROUTESEQ-1 Typical Data Screen.

Problem label.   This is a problem descriptor.   Enter any convenient label. WARNING: Do not use commas (,) or double quotation marks (") in the label since this will cause an error in reading the data file.

Circuity factor. This is a multiplier greater than 1 to convert straight-line distance to road, rail, air, etc. actual travel distance. For example, if linear coordinates are used, a multiplier of 1.21 is a good average to convert coordinate-computed distance to road miles. Do not use a value less than 1.

Map scaling factor. A multiplier to convert coordinates to a desirable distance measure. Coordinate divisions on a map or grid may be arbitrary. The map-scaling factor typically converts these coordinates to miles or kilometers. For example, a map showing a scale of 1″=50 miles and line markers every inch would have a scaling factor of 50.

Depot coordinates. These are the *X,Y* coordinates for the point where the route starts and ends. *X* and *Y* are linear grid coordinates, although other coordinate systems may be used with care.

Stop data. The points to be visited on a route are identified by their *X,Y* coordinates. *X* and *Y* are linear grid coordinates, although other coordinate systems may be used with care. Up to 50 stops are allowed on a route.

RUNNING *ROUTESEQ*

After the data have been prepared on the data screen, click on the **Solve** button to find the best route. The stop sequence may be specified, or **LOGWARE** will design the route. Once a solution is available, it may be graphically presented by clicking on the **Plot** button.

EXAMPLE

A truck is to be routed from its depot to five stops. The data screen giving location of the depot and the stops was previously shown. A grid overlay of the depot and stops is shown in Figure ROUTESEQ-1. Note that a circuity factor of 1.21 is used. Also, note that the map-scaling factor is 1 in this case.

The computed results as they appear on the screen are as follows with the plotted route shown in Figure ROUTESEQ-2.
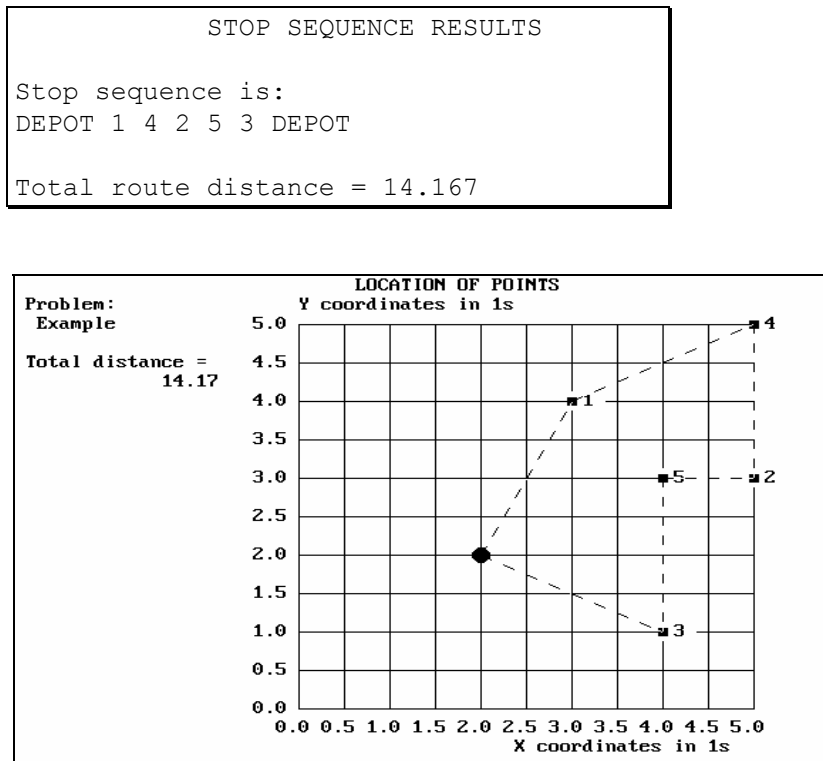
```
              STOP SEQUENCE RESULTS


Stop sequence is:
DEPOT 1 4 2 5 3 DEPOT


Total route distance = 14.167
```


Figure ROUTESEQ-2 Plot of Example Problem Solution.

INSTRUCTIONS FOR
VEHICLE ROUTING AND SCHEDULING
ROUTER

ROUTER is a software program to determine the best routes and schedules for a fleet of privately controlled vehicles. The typical problem is one where trucks are domiciled at a central depot, making deliveries or pickups to a number of stops, and returning to the same depot. Stops need to be assigned to vehicles and then sequenced on each vehicle route. The objective is to minimize the total distance traveled on all routes, and indirectly to minimize the total number of vehicles needed to serve the stops.

> *Example.* A food company makes daily deliveries from its warehouse to retail stores as shown in Figure ROUTER-1. The dispatcher must plan how many routes there should be, which stores should be assigned to the routes, and in which sequence the stores should be served. Certain of the stores can accept delivery only at specified times of the day, the drivers are to work 8 hours or be paid overtime, and the trucks are limited in their carrying capacity. These restrictions are to be respected in designing the routes.
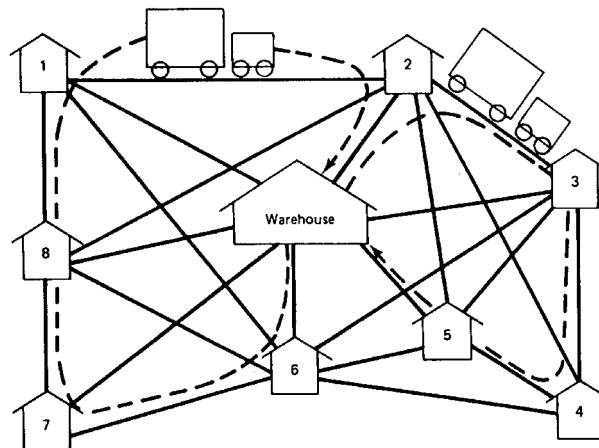
Figure ROUTER-1 Vehicle Routing from a Central Depot.

ROUTER is a commercial-grade model that has been installed in actual dispatching operations and has many capabilities that are not fully described in this set of abbreviated instructions; however, many of these become obvious by exploring the data screens. In addition, there are limits placed on the problem size of this educational version of the model. It handles up to 60 stops. The model's features include:

- Both pickup and delivery stops are permitted on the same route. Pickups may be mixed on the vehicle with deliveries, or they may only be permitted on the vehicle after all deliveries have been made.
- Mixed vehicle types are allowed.
- Loads on a vehicle may be controlled by weight, cube, or number of stops.
- A variety of coordinate systems for stop and depot location is allowed.

- Distances between depot and stops, or between stops, are computed from coordinate geometry, or they may be specified.
- The maximum time or distance on a route may be specified.
- Barriers may be defined to represent lakes, parks, rivers, etc. through which a route cannot penetrate.
- The earliest time for a vehicle to leave a depot and the latest time for it to return may be specified.
- Stop loading/unloading times may be calculated based on weight and cube, or they may be specified for each stop.
- Breaks such as for lunch and overnight may be specified.
- Speed zones are used to define the speed between groups of stops, or speeds may be specified between selected stop pairs.
- Time windows in which deliveries or pickups are to be made can be specified for each stop.
- Route design may be computed with the model using any one of three methods, or the user may specify the design.
- Route costs are determined based on vehicle fixed and variable rates, driver fixed and variable rates, and overtime rates.
- Incremental costs of serving a stop on a route are calculated which can be compared with an alternative transport method of serving the stop.

For background information on the routing and scheduling method (savings method) used in **ROUTER**, see Chapter 7 of the Business Logistics/Supply Chain Management 5e textbook.

INPUT

All data are entered by means of a screen editor. First, select a file from among existing ones or start a new one. Enter data in the appropriate screens as chosen from the various folders.

The data needed for vehicle routing consist primarily of data about stops, vehicles, costs, and constraints on route design. Each data element is discussed below.

Coordinate Points

The depot and all stops are located geographically with coordinate points. A variety of coordinate systems is permitted such as latitude-longitude or a simple linear grid. Coordinates are used to compute the approximate straight-line distance between point pairs and to locate them in relation to each other for mapping purposes.

Scaling Factors

Scaling factors are applied to the coordinate-computed distances to convert them to actual distances. These scaling factors, both horizontal and vertical, are made up of two parts. The first is to convert the coordinates to straight-line miles. Do this by multiplying the coordinate points by the map-scaling factor. However, if longitudes and latitudes are used, the scaling factors will depend on the approximate latitude of the stops

and depot. These conversion factors for latitude-longitude are given in Table ROUTER-1. Others may be obtained from the particular map used.

Second, a circuity factor is applied. This factor adjusts straight-line miles to approximate actual road miles. A commonly accepted factor (used by the Department of Transportation) is 1.21, if linear grid coordinates are used. This means that road miles are about 21 percent longer than straight-line miles where distance is calculated from linear grid coordinates. If latitude-longitude coordinates are used, a factor of 1.20 is more accurate for road distances in the U.S. See Table 14-4 of Business Logistics/Supply Chain Management 5e for circuity factors for other parts of the world.

Therefore, the scaling factor input into the **ROUTER** database is the product of the above two factors. For example, if the map scaling factor is 0.3 and the circuity factor is 1.21, the combined factor is $0.3 \times 1.21 = 0.363$. The final value may be altered to reflect any adjustments that may have been made in the coordinate points. If linear coordinates are used, horizontal and vertical scaling factors are likely to be the same, but they will be different for latitude-longitude coordinates.

Depot Time Restrictions

Depots frequently have time restrictions on them reflecting their open hours. It may be desirable to have vehicles not leave the depot before a certain time (earliest starting time) and to return no later than a certain time (latest return time).

Route Restrictions

It is often necessary or desirable to restrict routes by time and or distance. A time restriction is the maximum number of hours that a vehicle is allowed on a route. The distance restriction is the maximum number of miles that a vehicle may travel before it must return to the depot. *Be sure that these restrictions do not conflict with the depot time restrictions.*

Speed Zones

The depot and the stops may be grouped into boxes called speed zones and the speeds then defined between the zones. This allows speed differences to be recognized between rural and city areas, between congested and non-congested regions, or between any groups of stops.

A speed zone is a rectangular box into which stops are assigned. These boxes are defined with the coordinates of their sides, i.e., left, right, top, and bottom. The coordinates of each box must use the same coordinate system as for the depot and the stops.

Once the speed zones have been defined, the speeds between the boxes and within the boxes should be specified. For example, speeds between stops within a city zone might be 25 mph, while 50 mph might be appropriate between stops in a rural zone. However, speeds between the two city zones crossing a rural zone would be represented by a combined speed, possibly reflecting the proportion of the distance spent in each zone. If the approximations seem crude, a greater number of smaller boxes may be created to improve accuracy.

Table ROUTER-1   Lengths of One Degree Latitude and
One Degree Longitude

Latitude (Degrees)	Latitude		Longitude	
	Statute miles	Kilo-meters	Statute miles	Kilo-meters
0	68.704	110.569	69.172	111.322
5	68.710	110.578	68.911	110.902
10	68.725	110.603	68.129	109.643
15	68.751	110.644	66.830	107.553
20	68.786	110.701	65.026	104.650
25	68.829	110.770	62.729	100.953
30	68.879	110.850	59.956	96.490
35	68.935	110.941	56.725	91.290
40	68.993	111.034	53.063	85.397
45	69.054	111.132	48.995	78.850
50	69.115	111.230	44.552	71.700
55	69.175	111.327	39.766	63.997
60	69.230	111.415	34.674	55.803
65	69.281	111.497	29.315	47.178
70	69.324	111.567	23.729	38.188
75	69.360	111.625	17.960	28.904
80	69.386	111.666	12.051	19.394
85	69.402	111.692	6.049	9.735
90	69.407	111.700	0.000	0.000

For our problem, there would be 6 speed combinations, assuming that the speed between say zone 1 and zone 2 is the same as between zone 2 and zone 1. Listing the speed combinations, we would have:

Origin zone	Destination zone	Speed mph
1	1	25
1	2	45
1	3	35
2	2	25
2	3	45
3	3	25

If a speed for a zone combination that is needed has not been specified and there is no specific speed for the stop combination, the default or the specific speed between stops is used.

Speeds may also be specified for selected combinations of stops, or depot and stops. Finally, a default speed is given.

ROUTER prioritizes its choices of speeds. First, the computer searches for a specific stop-to-stop speed. If none is available, an attempt is made to find a speed from the Speed Zones list. If none is available, the default speed is used. A default speed should always be defined. *Be sure that speeds are sufficiently large so that all stops can be served within the time restrictions.*

Distances

Specified distances may be used in place of the coordinate point computed distances. Such specified distances always take precedence over the approximated distances. They are used typically on a selected basis where greater accuracy is needed, such as when the route is to represent a drive-time standard. In addition, they may be needed to accurately represent distances when one-way streets, lakes, mountains, or other barriers on the route make distance approximations unacceptable. These distances can be specified by depot-to-stop or by stop-to-stop pairs.

Break Times

Some route designs may require the driver to take a break during the tour, such as for lunch or a rest. Up to two break times are permitted, and they are expressed in minutes. They are to take place after a specified number of minutes into the planning horizon.

Time Windows

Any stop may have certain times within which deliveries or pickups can be made. These are referred to as time windows. They are specified as a beginning time and an ending time in minutes. If the first of two time windows is non-constraining, then the times should be set as wide as possible. A good choice would be to use the earliest start and latest return times for the depot. A non-constraining second time window is to set the beginning and ending times to a number beyond the planning horizon, say 9999.

Stop Volumes

Stops are designated as either a delivery (*D*) or a pickup (*P*). A delivery stop is one where goods originate from the depot and are destined for the stop. A pickup stop is one where goods originate from the stop and are destined for the depot.

Stop volumes may be expressed in one or two measures, typically weight and cube. Weight may be hundredweight, cases, units, kilograms, or other similar measure. It uses the *weight* carrying capacity of the vehicle. Cube, on the other hand, uses the *space* carrying capacity of the vehicle. It is expressed as cubic feet, cubic meters, or other appropriate space measure. Cube may also be used as a surrogate for stops. That is, by declaring cube to be 1 and then specifying cube capacity to be the number of stops allowed on a route, the number of stops on a route may be controlled.

It is not always necessary to control both weight and cube. Weight is the primary measure such that cube may be inactive at times. If so, use zeros for all stops and use a vehicle cube capacity of any size.

Pickup Policy

When pickups are to enter a delivery route, there will be the question of whether pickups should be made *while* delivery volume is present on the vehicle or whether they should only be allowed *after* all deliveries have been made. This policy is expressed in percentage terms of the available vehicle capacity. A vehicle loaded value of 0 percent means that pickups are allowed only after all of the deliveries have been made, or the vehicle is empty. A 35 percent value means that pickups are allowed when the vehicle load has dropped to 35 percent of capacity or less. Pickups are not allowed as long as the

pickup and delivery volume is more than 35 percent of capacity. A value of 100 percent means that pickups are allowed any time on a route. Percentage values near 100% allow the most flexibility in route design.

Vehicle Capacity

Vehicles are categorized first by unique type and then by the number of vehicles and their characteristics within that type. The various vehicle types are numbered consecutively. The number of vehicles within that type is then declared. Each type is given a capacity by weight and by cube. Weight and cube units should match those given as stop volumes.

It is a good idea to declare more vehicles than actually available in the fleet. **ROUTER** will only use enough of them to form the routes, which may be more than currently being used due to meeting the restrictions placed on route design. The current fleet may not be meeting these restrictions and therefore can operate with fewer vehicles.

Costs

Costs of the routes are determined by summing the costs associated with the vehicle and the driver. Vehicle costs are based on two rates. The first is the fixed rate per vehicle. This is the fixed charge associated with owning, or leasing, and maintaining the vehicle for the period represented by route planning, i.e., a day, a week, or a month. Next is the variable rate per mile for operating the vehicle. This rate represents such costs apportioned on a mileage basis as fuel, tires, and oil. These rates can vary with the vehicle type.

Driver costs are based on fixed and variable rates plus an overtime rate. The fixed driver rate is the charge associated with fringe benefits, salary minimums, and other charges that do not vary with the time on the route. The variable rate is associated with the costs that are dependent on the hours spent on the route such as the wage rate. The overtime rate is the per-hour rate that takes effect after a specified number of hours have been spent on a route.

Route Barriers

There may be certain areas where the vehicles must go around rather than penetrate them. Such areas may be lakes, rivers, parks, etc. In **ROUTER**, a vehicle comes up to the barrier and runs around the *shortest side*. The distance around the barrier is added to the route distance.

Barriers are always expressed as a 4-sided figure. This figure may be an irregular shape or a rectangle. Each corner of the figure is expressed as horizontal and vertical coordinates. The coordinates must be from the same coordinate system as for the stops. The coordinates of each barrier should be expressed as the northwest, northeast, southwest, and southeast corners of the barrier. More than one barrier may be used in any one problem.

Note: The shape of the barrier must always bulge outward.

RUNNING *ROUTER*

After preparing the data screens, the solver may be activated by clicking on the **Solve** button. A submenu will appear from which you may select to have **ROUTER** determine

the route design, or you may specify the route configuration. The latter is useful in establishing the costs associated with a current route design.

ROUTER-Designed Routes

When **ROUTER** designs the routes, the solution procedure is a heuristic method based on the savings method of Clarke and Wright. Solution takes place on the database as shown on the current data screens.

> *NOTE:* If you see the message that there is an INADEQUATE NUMBER OF VEHICLES AVAILABLE TO COVER ROUTES, add more vehicle capacity (large vehicles or more vehicles), even if the current total vehicle capacity exceeds total current stop volume. The additional capacity is needed for intermediate computations, but will not necessarily appear in the final solution.

User-Designed Routes

The user has control over the route design through the route designer/editor. This editor serves two purposes. First, it allows the user to create route designs with the aid of graphics, statistics about developing routes, and whatever principles of good route design may be available. Second, it allows the user to specify routes, vehicles, and stop sequences, and to edit routes that have been developed. This procedure may be used as an alternative to a **ROUTER**-designed route.

OUTPUT

The output is presented in two forms: a report of the routes and associated statistics, and a graphical display of the routes. The route is displayed graphically upon completion of the solution process. For the Regals Metals database, the resulting route design is displayed in Figure ROUTER-2.
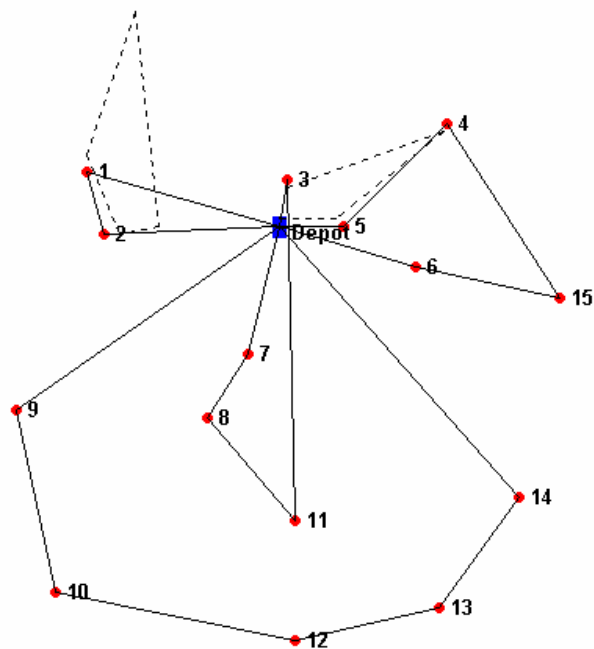
Figure ROUTER-2 Graphical Display of the Route Design for Regals Metals

A report may be obtained by clicking on the **Report** button after a solution run is completed. The report provides summary information for all routes as well as detailed costs and time statistics on individual routes. A sample report is shown in Figure ROUTER-3.

```
                           ROUTER SOLUTION REPORT
Label- EXAMPLE
Date- 5/7/97
Time- 9:20:40 AM


                        *** SUMMARY REPORT ***

TIME/DISTANCE/COST INFORMATION


        Route   Run  Stop  Brk  Stem
Route   time, time, time, time, time,   Start  Return No of   Route      Route
  no      hr    hr    hr    hr    hr      time    time stops dist,Mi    cost,$
   1     23.5  15.8   7.7    .0  13.8 12:00AM 11:29PM     2     791       .00
   2     29.6  22.6   7.0    .0   5.5 12:00AM 05:37AM     4    1131       .00
   3     28.2  21.8   6.3    .0   6.2 12:00AM 04:09AM     4    1092       .00
   4     52.3  42.4   9.8    .0  20.7 12:00AM 04:15AM     5    2122       .00
Total  133.6 102.7  30.8    .0  46.3                     15    5136       .00

VEHICLE INFORMATION

Route Veh Weight Delvry Pickup Weight   Cube Delvry Pickup   Cube Vehicle
  no  typ capcty weight weight   util capcty   cube   cube   util description
   1    1    300    230      0  76.7%   9999      0      0    .0% TRUCK - 1
   2    2    210    210      0 100.0%   9999      0      0    .0% TRUCK - 2
   3    2    210    190      0  90.5%   9999      0      0    .0% TRUCK - 2
   4    1    300    295      0  98.3%   9999      0      0    .0% TRUCK - 1
Total      1020    925      0  90.7%  39996      0      0    .0%

DETAILED COST INFORMATION
```

```
          --------Vehicle-----------  -----------Driver----------------
 Route    Total   Fixed  Mileage   Total   Fixed  Regular Overtime
  no     cost,$  cost,$   cost,$   cost,$  cost,$   time,$   time,$
   1       .00     .00      .00      .00     .00      .00      .00
   2       .00     .00      .00      .00     .00      .00      .00
   3       .00     .00      .00      .00     .00      .00      .00
   4       .00     .00      .00      .00     .00      .00      .00
Total      .00     .00      .00      .00     .00      .00      .00

              *** DETAIL REPORT ON ROUTE NUMBER 1 ***

A TRUCK - 1 leaves at 12:00AM on day 1 from the depot at Toledo OH


                                     Stop   Drive Distance Time
    Stop            Arrive    Depart time to stop to stop wind
 No description      time Day    time Day  Min     Min   Miles met?
  2 Chicago      05:20AM  1 12:20PM  1  420   320.8     267  YES
  1 Milwaukee    02:20PM  1 03:00PM  1   40   119.7     100  YES
   Depot         11:29PM  1 ------- --  ---   509.2     424

    Stop              Stop volume  Inc cost to serve stop  Capacity in use
 No description      Weight   Cube    In $  In $/unit      Weight   Cube
                                                            76.7%    .0%
  2 Chicago            210      0     .00        .0          6.7     .0
  1 Milwaukee           20      0     .00        .0           .0     .0
Totals Weight: Del = 230 Pickups = 0 Cube: Del = 0 Pickups = 0

Route time:                   Distance:
  Driving            15.8 hr    To 1st stop              267 mi
  Load/unload         7.7       From last stop           424
  Break                .0       On route                 100
    Total            23.5 hr      Total                   791 mi

  Max allowed       168.0 hr    Max allowed       9999 mi

Route costs:
  Driver (reg time)          $.00
  Driver (over time)          .00
  Vehicle (mileage)           .00
  Fixed                       .00
          Total             $.00
```

Figure ROUTER-3 Excerpts from a Solution Report for the Regals Metals Company

INSTRUCTIONS FOR
INVENTORY CONTROL SOFTWARE
INPOL

INPOL is a software program to compute inventory policies under the *reorder point* (fixed order quantity, variable order interval) and *periodic review* (variable order quantity, fixed order interval) methods of economic order inventory control. Economic order quantity principles are used to find the optimal policies. These policies are a result of answering two questions:

- How much to order of a product?
- When should the product be ordered?

The policy variables can include replenishment quantity, reorder-point quantity, time between inventory level review, and the target quantity for orders.

A single echelon, single inventory is assumed, which schematically can be represented as follows.
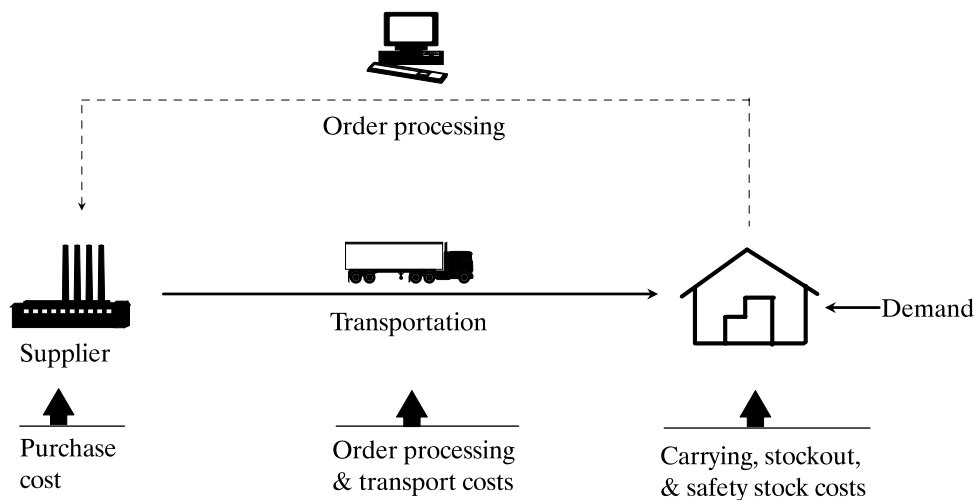
INPOL will compute, and optimize if called upon, the following total cost expression.

Total cost = Purchase cost + Transport cost + Carrying cost
+ Order processing cost + Out-of-stock cost + Safety stock cost

There are a number of options available to determine inventory policy. These are:

- Reorder point or periodic review methods of inventory control may be selected.
- The customer service level may be specified, or it may be computed if out-of-stock costs are known.
- The order quantity may be specified or computed.
- The products may be ordered singly or jointly.

- An average inventory level may be specified to represent existing conditions and the costs determined.

Thus, a variety of practical situations may be represented. The inventory control methods used in the **INPOL** module are discussed in Business Logistics/Supply Chain Management 5e.

INPUT

The input data are entered by means of a data screen editor or Excel spreadsheet. Four data sections need to be prepared. Each section is illustrated using the data from the test problem shown as product 1, and the data elements on the screens are defined.

> *Example.* The test problem involves two products that are to be jointly ordered. The inventory level review time is set at 1.5 weeks for both products. The service indices are specified to be 0.90 and 0.80 respectively. Other data are shown on the illustrations below.

The PARAMETERS AND LABELS Section

This section of the data screen has the following elements.

```
Problem label-TEST PROBLEM


Number of products: 2
Time frame (1=Calendar days, 2=Working days, 3=Weeks,
          4=Months, 5=Year): 3
Are the items jointly ordered (Y/N)? Y
Are the service indices specified (Y/N)? Y
Do you want a sensitivity analysis (Y/N)? Y

Joint-Order Parameters
  Joint-order procurement cost: 100
  Order cycle time (in the unit of time frame): 1.5

Individual-Order Parameter
  Are the order quantities to be specified (Y/N)? N
```

A description of each item in this section is as follows.

Number of Products

Specify the number of products that are to be analyzed. A maximum of 50 product items may be processed at one time. If the items are to be jointly ordered, data for at least two items must be entered in the database.

Time Frame (1=Calendar days, 2=Working days, 3=Weeks, 4=Months, 5=Year):

Select the time dimension for the demand, lead-time, and carrying cost data. Be sure that the data are expressed in the same time units throughout. Calendar days represent 365 days per year. Working days represent 250 days per year, or about 5 working days per week. Weeks, months, and year have their usual definition.

Are the Items Jointly Ordered (Y/N)?
If items are to be ordered separately, select No (*N*). If two items or more are to be jointly ordered, select Yes (*Y*). Do not respond with *Y* and select that order quantities are to be specified at the same time.

Are Service Indices Specified (Y/N)?
If you are specifying the customer service levels for the product(s), select Yes (*Y*). If the optimum service levels are to be computed, select No (*N*). Computing the service levels requires that you know the out-of-stock costs.

Do You Want a Sensitivity Analysis (Y/N)?
A sensitivity analysis computes various inventory related costs and parameters for incremental changes in service levels. Do not use this option if the average inventory is given values greater than zero (0).

JOINT-ORDER PARAMETERS

Joint-Order Procurement Cost
This is a common cost as incurred for processing all jointly ordered items simultaneously. It is expressed as $/order.

Order Cycle Time
You may specify a common order cycle time, or lead-time, for all jointly ordered items. It must be the dimension as specified in the Time Frame. **INPOL** will compute the remaining parameters. You must first have declared that items are to be jointly ordered. Then, if order cycle time is specified as zero (0), the optimum review time will be computed. Specify an order cycle time greater than zero (0) if you want to fix the review time at a particular value. Use this option only if items are to be jointly ordered.

INDIVIDUAL-ORDER PARAMETER

Are the Order Quantities to be Specified (Y/N)?
This selection applies only to separately ordered items. You may assign the order quantity for an item and **INPOL** will complete the remaining calculations. Respond with Yes (*Y*). If No (*N*) is selected, **INPOL** will compute the optimum order quantity. You must select No (*N*) if you have declared No (*N*) to items being jointly ordered.

The DEMAND/LEAD TIME DATA Section
This section has the following layout:

```
 Prd                     Average    Std. dev.   Average    Std. dev.
 no.  Product label---   demand    of demand  lead time  of lead time
   1  PRODUCT 1            2000          100      1.50         0.00
   1  PRODUCT 2             500            7      1.50         0.00
```

A description of each item in the section is as follows.

Product Label

Give an identifying label to the data. Good practice is to use no more than 15 characters, however, do not use a comma (,) or a double quotation mark (") in the label. Retain the same product sequence throughout all data screens.

Average Demand

This is the item's average demand for the period that you selected in **PARAMETERS AND LABELS** section. It may be the item demand forecast or the average demand over the period. For example, average demand is 2000 lb. per year. Be sure the same time dimension is used as for lead-time and carrying cost.

Standard Deviation of Demand

This is the standard deviation of the demand. It may be the standard error of the forecast or the standard deviation as calculated from the distribution of demand. Demand is assumed normally distributed. For example, the standard deviation of demand is 100 lb. per year.

Average Lead-Time

This is the average lead-time as calculated from the distribution of lead-times. The period must be the same as for demand and carrying cost. For example, the average lead-time is 0.0096 years (5 weeks), but it is expressed in years because demand is in yearly time units.

Standard Deviation of Lead-Time

This is the standard deviation of the lead-time distribution. It is assumed that demand and lead-time distributions are independent of each other, which may not be the case. A large standard deviation of lead-time can produce very high safety stock levels and, therefore, high in-stock probabilities. If the service level for the item is very high, reduce this value. For example, the standard deviation of lead-time is 0.00005 lb. per year. Lead times are assumed normally distributed.

The PRICE/COST DATA Section

This screen has the following layout.

```
                          Trans-  Carry-  Order   Out-of-
 Prd                 Unit port    ing     proc.   stock
 no. Product label--- price rate  cost    cost    cost
   1 PRODUCT 1        2.25 0.0000 0.0058  0.0000   1.00
   2 PRODUCT 2        1.90 0.0000 0.0058  0.0000   0.75
```

A description of each item on the screen is:

Product Label

Give an identifying label to the data. Good practice is to use no more than 15 characters and do not use a comma (,) or a double quotation mark (") in the label. Retain the same product sequence throughout all data screens.

Unit Price

This is the price paid for the item in inventory. It may be a delivered price or an f.o.b. factory price. Care must be taken in how the transportation rate is specified. Price should be expressed in $/unit. For example, the unit price is $2.25 per lb.

Transport Rate

Price plus transport rate represent the landed price of the item in inventory. If the transportation charge is already included in the price, as it would be for a delivered price, no transport rate needs to be included. With an f.o.b. factory price, the transport rate should be included. It should be expressed in $/unit. For example, the transport rate is $0.25 per lb.

Carrying Cost

Carrying cost represents such components as capital tied up in inventory, insurance on the inventory, personal property taxes, obsolescence, and any other costs that are incurred due to the level of inventory held. It is expressed as a fraction of the item value per unit of time. For example, a carrying cost of 30% per year would be inputted as a fraction, i.e., 0.30. *Be sure carrying cost is expressed in the same time dimension as demand and lead-time, i.e., if demand is in weeks, carrying cost should also be in per-week units*.

Order Processing Cost

This refers to the cost to process a specific item on an order. For jointly ordered items, there may be a common cost in addition to this item cost or instead of it. The common cost is assigned in the **PARAMETERS AND LABELS** section. An example of the item order processing cost would be $1.35 per order.

Out-of-Stock Cost

This is the cost associated with being out of stock. It refers to lost profit, lost future sales, or additional processing costs due to backordering. If a customer service level is specified, you do not necessarily have to provide this cost. If no customer service level is

specified, this cost is required to find the service level and the optimal policy. The value may be set at zero (0) if the service level is given. A value greater than zero (0) must be used if no service level is selected in the **PARAMETERS AND LABELS** section. It is a cost expressed as $/unit, for example, $0.55 per lb.

The MISCELLANEOUS DATA Section
This screen has the following layout:

```
 Prd                     Avg initial      Order      Service
 no. Product label-       inventory      quantity      index
   1 PRODUCT 1                    0            0        0.90
   2 PRODUCT 2                    0            0        0.80
```

A description of each item in this section is as follows.

Product Label
Give an identifying label to the data. Good practice is to use no more than 15 characters and do not use a comma (,) or a double quotation mark (") in the label. Retain the same product sequence throughout all data screens.

Average Initial Inventory
If you wish to find the cost of a particular inventory level, you may specify the average initial inventory. The level is expressed in the same units as demand. For example, the average initial inventory is 1689 lb. The value is usually set at zero (0). Set the sensitivity option in the **PARAMETERS AND LABELS** section to No (*N*).

Order Quantity
This is used to specify a particular order quantity rather than have **INPOL** calculate it for you. Otherwise, it is left as zero (0). The units are the same as those used for demand. For example, order quantity is 125 lb. Do not use for jointly ordered items.

Service Index
This is the probability of being in stock during an order cycle. This index is given a value when there is no out-of-stock cost provided in the database. It presets the service level and **INPOL** minimizes cost based on it. It is expressed as a fraction of 1, e.g., 0.90. Use when service levels are to be specified as a selected option in the **PARAMETERS AND LABELS** section.

RUNNING *INPOL*
After the data have been prepared on the data screen, click on **Solve** to compute the inventory policy. The results are shown as an output report and graphical displays for various combinations of output variables.

Output Report

Using the example problem shown under **INPUT** above, the following will be presented in report form. The first portion of the report shows the values of the policy variables. There are no values shown under the reorder point policy since this is a joint-order problem and only a periodic review policy is appropriate. Hence, we should see.

```
              COMPUTED STOCKING POLICIES

 Periodic Review Policy
                     Product 1    Product 2
 Average inventory      1,722          385
 Order quantity         3,000          750
 Max Level              6,222        1,510
 Order review time      1.50wk       1.50wk
 Turnover ratio            60           68
 Investment            $3,874         $732
 Demand in-stock       99.73%       99.82%
```

and annual costs for various cost categories by product:

```
                ESTIMATED ANNUAL COSTS

 Periodic Review Policy
                     Product  1    Product  2
 Purchase cost    $    234,000        49,400
 Transport cost             0             0
 Carrying cost          1,018           215
 Order proc. cost       1,733         1,733
 Out-of-stk cost          285            35
 Safety stk cost          150             6
```

and a summary of the cost and investment in all products in the database:

```
                  SUMMARY DATA

 Periodic Review Policy
 Purchase cost      $   283,400
 Transport cost               0
 Carrying cost            1,228
 Order proc. cost         3,466
 Out-of-stk cost            320
 Safety stk cost            156
 Total cost             288,570
 Total investment   $     4,606
```

If a sensitivity analysis is selected, a report of the following type is displayed.

```
 Sensitivity Analysis Results For Periodic Review System
Product 1

 Service   Service      Max    Review     Average     Total
   index   level, %    level  time, wk   inventory   cost, $
    0.50     97.70     6,000     1.50       1,500    239,141
    0.51     97.78     6,005     1.50       1,505    239.056
    0.52     97.84     6,009     1.50       1,509    239,000
     .         .         .         .          .         .
     .         .         .         .          .         .
     .         .         .         .          .         .
    0.98     99.96     6,355     1.50       1,855    237,030
    0.99     99.98     6,404     1.50       1,904    237,038
    1.00*   100.00 1,738,051     1.50   1,733,551  1,406,036
*High values for a service index of 1.00 represent infinity.
```

Graphical Displays
Sensitivity plots can be displayed. The four choices are:

- Total costs against service level
- Total costs against average inventory
- Average inventory against max level
- Service level against average inventory

These are plots whose data are taken from the sensitivity results in a solution run. An example is shown in Figure INPOL-1.
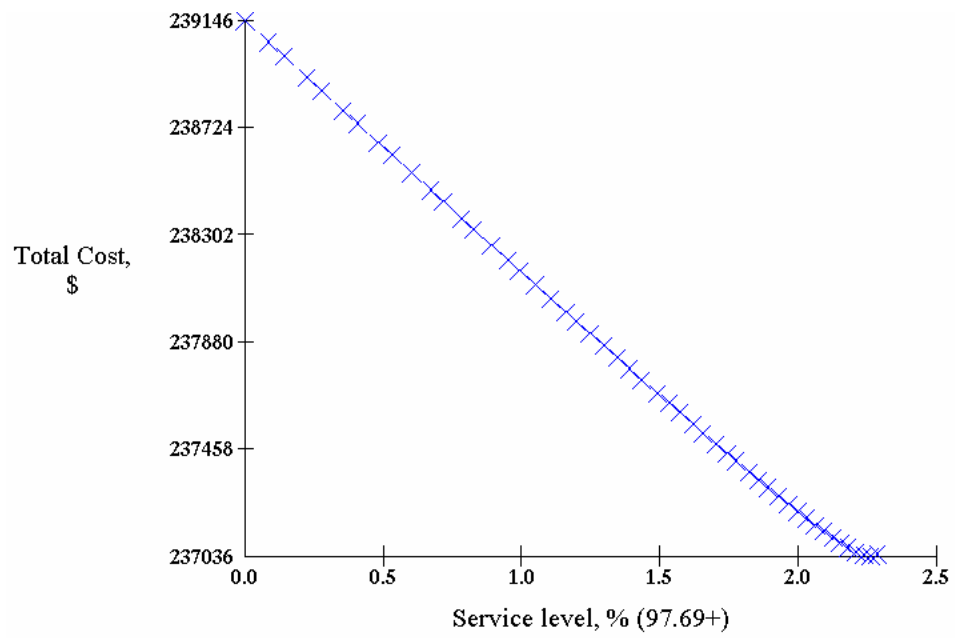
Figure INPOL-1 Plot of Total Cost Against Service Level

INSTRUCTIONS FOR
CENTER-OF-GRAVITY FACILITY LOCATOR
COG

COG is computer software to locate a single facility by means of the exact center-of-gravity method. The problem is one where a single facility, such as a warehouse, is to serve (or be served by) a number of demand (or supply) points with known locations and volumes. The objective is to find a location such that total transportation cost, as represented in the following expression, is minimized:

$$TC = \sum_{i=1}^{N} V_i R_i K \left[\left(X_i - \overline{X} \right)^2 + \left(Y_i - \overline{Y} \right)^2 \right]^T$$

where

TC = total transportation cost

N = the number of origin/destination points in the problem. Up to 500 points may be used.

X_i, Y_i = the geographical locations of the origin/destination points represented using linear X, Y coordinate points.

T = a power factor in the distance computation formula. Distances are computed from coordinate points using the following formula.

$$\text{Distance} = K \left[\left(X_i - \overline{X} \right)^2 + \left(Y_i - \overline{Y} \right)^2 \right]^T$$

where X_i, Y_i represent the origin/destination points and $\overline{X}, \overline{Y}$ represent the facility. The power factor T controls the linearity of the distance between points. The value of T is usually 0.5, which is a straight line between points.

K = a scaling factor to convert coordinate distances to miles.

V = the volume of an origin/destination point in any appropriate demand units.

R = the transportation rate between the facility to be located and the origin/destination points, expressed in a monetary unit per unit of volume per unit of distance, such as $/unit/mile.

For background on the center-of-gravity method used in the **COG** module, see Chapter 13 of Business Logistics/Supply Chain Management 5e.

INPUT
The data inputs consist of coordinates for locating origin/destination points, volumes of origin/destination points, transportation rates between the facility and the origin/destination points, and miscellaneous factors.

EXAMPLE

Suppose that we have a small problem as shown in Figure COG-1. The product is Chemicals. There are 10 markets to be served from a single warehouse location. The warehouse is supplied from a single plant. The total amount of product shipped by the plant is the sum of the volume demanded by the markets. The product is shipped over road networks. The annual volumes of the markets and the transportation rates are given as follows:

Point, i	Volume, lb.	Rate, $/lb./mile
M_1	3,000,000	0.0020
M_2	5,000,000	0.0015
M_3	17,000,000	0.0020
M_4	12,000,000	0.0013
M_5	9,000,000	0.0015
M_6	10,000,000	0.0012
M_7	24,000,000	0.0020
M_8	14,000,000	0.0014
M_9	23,000,000	0.0024
M_{10}	30,000,000	0.0011
	147,000,000	
P_1	147,000,000	0.0005

Locate the single warehouse so that transportation costs are minimized.

The inputs to **COG** would look like:

```
 Problem label: Example
 Power factor (T): .5
 Map scaling factor (K): 50

Point   Point      X coor-   Y coor-              Transport
  no.   label----   dinate    dinate    Volume      rate
    1   M1            2.00      1.00    3000000     0.0020
    2   M2            5.00      2.00    5000000     0.0015
    3   M3            9.00      1.00   17000000     0.0020
    4   M4            7.00      4.00   12000000     0.0013
    5   M5            2.00      5.00    9000000     0.0015
    6   M6           10.00      5.00   10000000     0.0012
    7   M7            2.00      7.00   24000000     0.0020
    8   M8            4.00      7.00   14000000     0.0014
    9   M9            5.00      8.00   23000000     0.0024
   10   M10           8.00      9.00   30000000     0.0011
   11   P11           9.00      6.00  147000000     0.0005
```
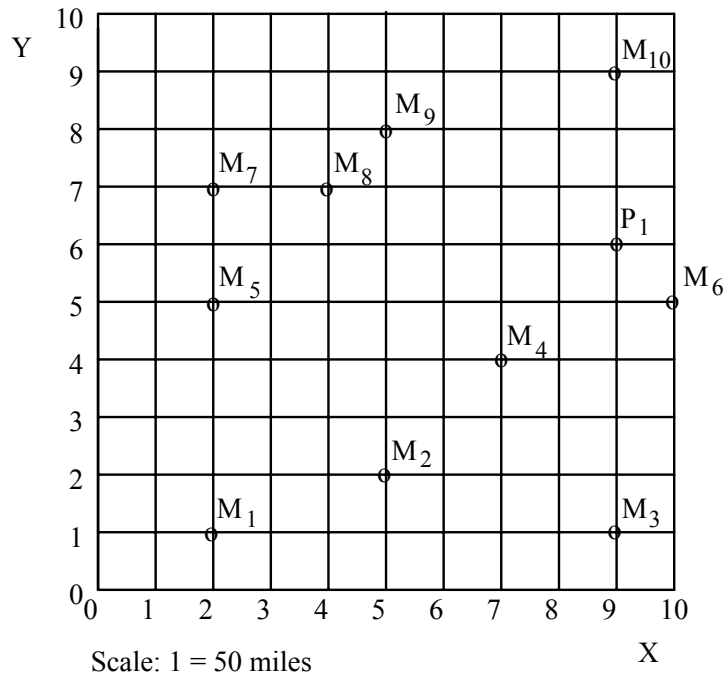
Scale: 1 = 50 miles

Figure COG-1 Locations of Market Points (M_i) and Plant (P_1) on a Linear Grid.

RUNNING *COG*

Running **COG** requires you to first create the database for a particular location problem. Then click on the **Solve** button to compute the center-of-gravity coordinates. You may choose to have **COG** compute the facility location coordinates, or you may specify these coordinates. If you choose have the coordinates computed, the simple center-of-gravity is found. To see if this initial location can be improved upon, ask for additional computational cycles. When there is little or no change in costs between successive cycles, ask for no further computations. Read the results from the screen or print them. For the example problem, the results will appear as shown in Figure COG-2. After 50 computational cycles, the best location for the facility is

$$\overline{X} = 6.298, \quad \overline{Y} = 6.484$$

for a total annual transportation cost of $55,015,057. At this point, you may ask for the points and the facility location to be plotted on a linear grid by selecting this option from the master menu. Such output is shown in Figure COG-3.

```
                              COG
      LOCATES A FACILITY BY THE EXACT CENTER OF GRAVITY METHOD

      Iteration      _              _
      number        X coord        Y coord        Total cost
              0      6.252          5.969          55,469,593   <- COG
              1      6.360          6.306          55,061,186
              2      6.370          6.409          55,024,105
              3      6.358          6.444          55,018,749
              4      6.344          6.459          55,016,953
              5      6.332          6.467          55,016,068
              6      6.323          6.472          55,015,599
              7      6.317          6.475          55,015,348
              8      6.312          6.477          55,015,214
              9      6.308          6.479          55,015,141
             10      6.306          6.480          55,015,102
             11      6.304          6.481          55,015,081
             12      6.302          6.482          55,015,070
             13      6.301          6.483          55,015,064
             14      6.300          6.483          55,015,061
             15      6.300          6.483          55,015,059
             16      6.300          6.483          55,015,058
             17      6.299          6.484          55,015,058
             18      6.299          6.484          55,015,058
             50      6.298          6.484          55,015,057
```

Figure COG-2 Computational Results for the Example Problem.
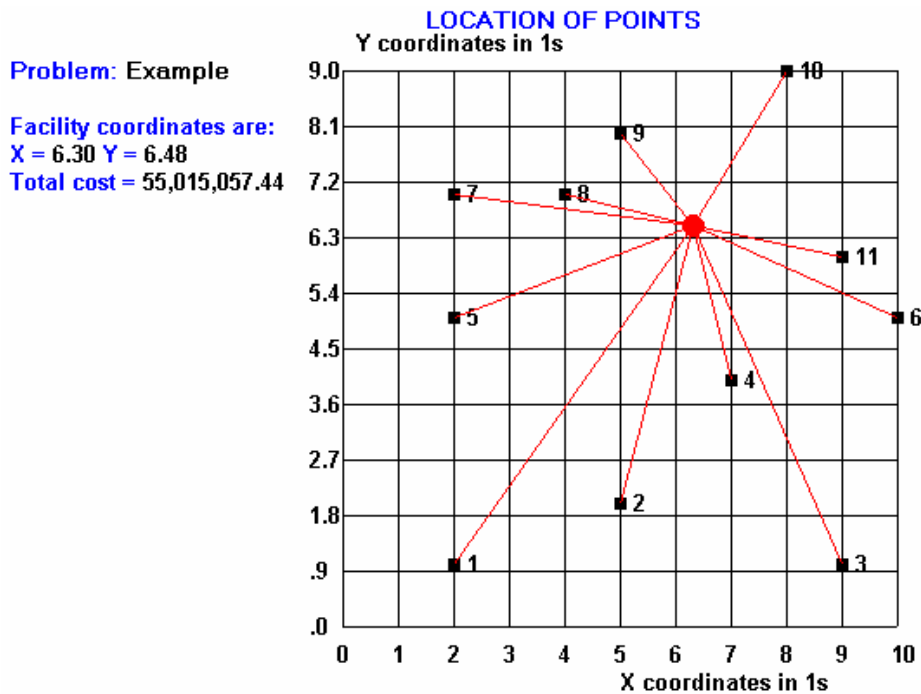

Figure COG-3 Plot of Optimized Facility Location for Example Problem.

INSTRUCTIONS FOR
CENTER-OF-GRAVITY MULTIPLE FACILITY LOCATOR
MULTICOG

MULTICOG is computer software to locate multiple facilities by means of the exact center-of-gravity method. The problem is to locate one or more facilities (sources), such as warehouses, to serve a number of demand points (sinks) of known locations, volumes, and transportation rates. The number of facility locations is specified. The objective is the find the coordinates of the facilities such that the following expression is minimized.

$$TC = \sum_{j=1}^{M} \sum_{i=1}^{N} V_{ij} R_{ij} K \sqrt{\left(X_i - \overline{X}_j\right)^2 + \left(Y_i - \overline{Y}_j\right)^2}$$

where

TC	=	total transportation cost
i	=	demand (sink) point number up to a total of N
j	=	facility (source) number up to a total of M
V_{ij}	=	volume associated with a demand point
R_{ij}	=	transport rate to a demand (from a supply) point
X_i, Y_i	=	coordinates of a demand (or supply) point
$\overline{X}_j, \overline{Y}_j$	=	coordinates for the facility location j.
K	=	scaling factor to convert coordinates to distance units. Multiply K by 1.21 to approximate road distance and 1.24 to approximate rail distance.

For background on the center-of-gravity method used in the **MULTICOG** module, see Chapter 13 of Business Logistics/Supply Chain Management 5e.

INPUT

The inputs to the program consist of (1) a problem descriptor, (2) coordinates for demand points, (3) volumes of demand points, (4) transportation rates between the facility and the demand points, and (5) a scaling factor. Specifically:

- A problem descriptor using any combination of letters and numbers
- The number of demand points in the problem. Up to 500 may be used.
- The geographical locations of the demand points represented by linear *X,Y* coordinate points. Any linear grid system may be used.
- The number of facility locations to be analyzed. Up to 20 locations may be used.
- A scaling factor to convert coordinate distances to miles
- The volume of a demand point in any appropriate units
- The transportation rate between the facility to be located and the demand points expressed in $/unit/mile or other distance measure

All input can be prepared from the screen editor or using an Excel spreadsheet.

EXAMPLE

Suppose we have a small problem as shown in Figure MULTICOG-1. The product is Chemicals. There are 10 markets that being served from two warehouse locations. The product is shipped over road networks. The annual volumes of the markets and the transportation rates are given as follows:

Point, i	Volume, lb.	Rate, $/lb./mile
M_1	3,000,000	0.0020
M_2	5,000,000	0.0015
M_3	17,000,000	0.0020
M_4	12,000,000	0.0013
M_5	9,000,000	0.0015
M_6	10,000,000	0.0012
M_7	24,000,000	0.0020
M_8	14,000,000	0.0014
M_9	23,000,000	0.0024
M_{10}	30,000,000	0.0011
	147,000,000	

Locate the two warehouses so that annual transportation costs are minimized.

The input to **MULTICOG** would look like:

Parameters and Labels

```
          --PARAMETERS AND LABELS--
Problem label: Example
Map scaling factor (K):    50.00
```

Location Data

```
                    --LOCATION DATA--

Point Point          X coor-   Y coor-              Transport
  no. label--------- dinate    dinate    Volume        rate
   1   M1              2.00      1.00    3000000      0.0020
   2   M2              5.00      2.00    5000000      0.0015
   3   M3              9.00      1.00   17000000      0.0020
   4   M4              7.00      4.00   12000000      0.0013
   5   M5             10.00      5.00   10000000      0.0012
   6   M6              2.00      5.00    9000000      0.0015
   7   M7              2.00      7.00   24000000      0.0020
   8   M8              4.00      7.00   14000000      0.0014
   9   M9              5.00      8.00   23000000      0.0024
  10   M10             8.00      9.00   30000000      0.0011
```
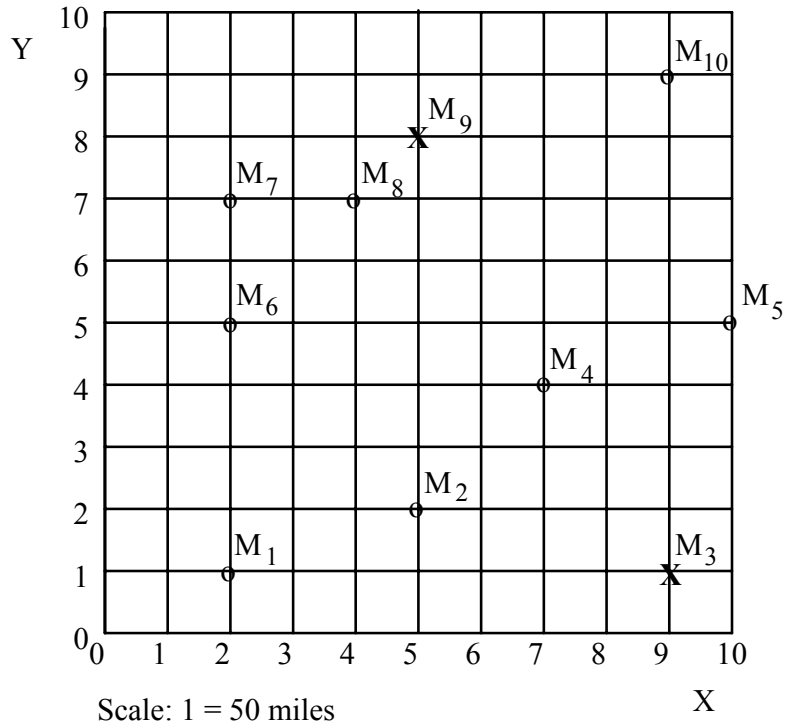
Figure MULTICOG-1 Locations of Market Points (M_i) on a Linear Grid.

RUNNING *MULTICOG*

After preparing a database, click on **Solve** to find the facility coordinate points. Then, select how many facilities are to be located. Select whether you wish the program to find the locations or whether you want to specify the location coordinates. If the latter is chosen, you will need to assign each demand point to a facility location. Read the results from the screen or direct them to a printer. The results may be displayed graphically by clicking on the **Plot** button.

A computed solution for two warehouses in the example problem is shown in Figure MULTICOG-2. The first location is to be located at X_1=5.0, Y_1=8.0, should serve markets 6, 7, 8, 9, and 10. Warehouse 2, which is to be located at X_2=9.0, Y_2=1.0, is to serve markets 1, 2, 3, 4, and 5. The suggested locations are plotted in Figure MULTICOG-3.

```
            ** PROBLEM SOLUTION **

 Title: EXAMPLE PROBLEM
 Source  X-Coordinate  Y-Coordinate      Volume        Cost
    1   4.99983       7.999873     100,000,000  17,057,023
    2   8.997027      1.004134      47,000,000   8,932,740


 Source  Allocated demand points to source points
    1   6 7 8 9 10
    2   1 2 3 4 5
 Total cost = 25,989,764.00
```

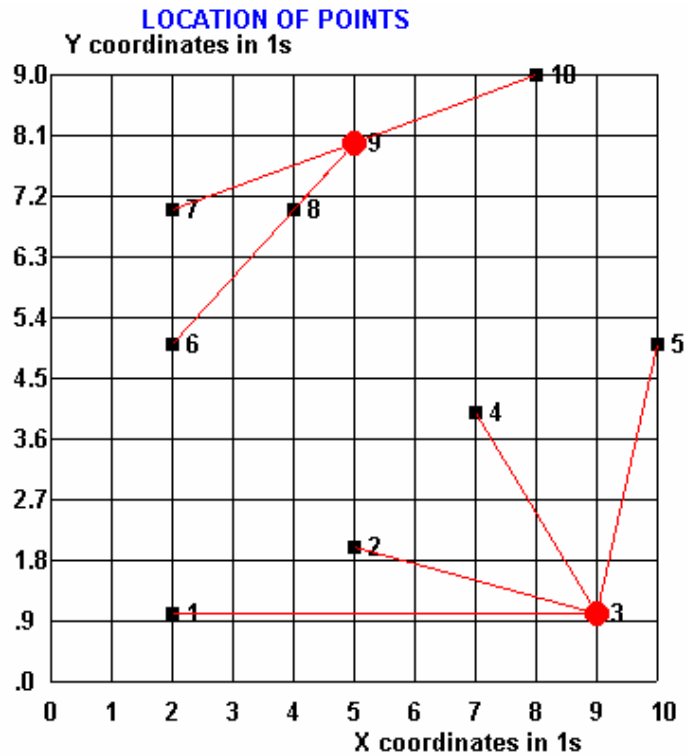**Figure MULTICOG-2** Report of Solution Results.

Figure MULTICOG-3   A Plot of the Solution for Two Warehouses for the Example
Problem.

INSTRUCTIONS FOR
P-MEDIAN MULTIPLE FACILITY LOCATOR
PMED

PMED is computer software to locate multiple facilities by means of the P-median approach. The problem is to locate one or more facilities (sources), such as warehouses, to serve a number of demand points (sinks) of known locations, volumes, and transportation rates. Fixed costs of a candidate set of facilities may also be known. The candidate set of facilities is selected from the demand points. The objective is to find the best *P* locations from the *M* candidate sites where *P* is less than or equal to *M*. The P-median problem is expressed as:

$$Minimize \ TC = \sum_i \sum_j V_i R_i d_{ij} X_{ij} + \sum_j F_j Y_j$$

subject to:

$$\sum_j X_{ij} = Y_j \text{ (for all } i)$$

$$X_{ij} \leq X_{ij} \text{ (for candidate } i, j \text{ pairs)}$$

$$\sum_j X_{ij} = p$$

$$X_{ij} = (0,1) \text{ (for all } i, j \text{ pairs)}$$

$$Y_j = \begin{cases} 1 \text{ if facility is open} \\ 0 \text{ if closed} \end{cases}$$

where:

$$X_{i,j} = \begin{cases} 1, \text{ if demand or supply node } i \text{ is assigned to facility } j \\ 0, \text{ if otherwise} \end{cases}$$

$$X_{j,j} = \begin{cases} 1, \text{ if a facility is sited at node } j \\ 0, \text{ if otherwise} \end{cases}$$

$$Y_j = \begin{cases} 1, \text{ if } j \text{ is open} \\ 0, \text{ if not open or a candidate site} \end{cases}$$

$TC =$ total transportation and facility fixed costs

$i =$ demand or supply point (sink) number up to a total of N

$j =$ candidate facility (source) number up to a total of M

$V_i =$ volume of demand or supply point i

$R_i =$ transportation rate associated with demand or supply point i

$d_{ij} =$ distance between demand or supply point i and facility j

$p =$ number of facilities to locate

For more discussion on the P-median method used in the **PMED** module, see Chapter 13 of Business Logistics/Supply Chain Management 5e.

INPUT

The inputs to the program consist of (1) a problem descriptor, (2) coordinates for demand and supply points, (3) volumes of demand and supply points, (4) transportation rates associated with a demand or supply point, (5) a candidate list of facilities from which the number of locations are selected for analysis, (6) fixed costs for candidate facilities, (7) type of coordinate system used, and (8) map scaling factors. Specifically:

- A problem descriptor using any combination of letters and numbers
- The number of demand and supply points in the problem. Up to a maximum of 65 are allowed.
- The geographical locations of demand and supply points are represented as linear grid coordinates or as latitude-longitude coordinates. If latitude-longitude coordinates are used, they should be expressed in **_degrees_**. Specify the type of coordinate system to be used.
- The particular demand or supply points representing a candidate set of potential facility locations. Up to 15 points may be in the candidate list. These sites are identified with an **X** in the database. Non-candidate sites are left blank.
- Scaling factors for the linear coordinates to convert them to a distance measure such as miles. Scaling factors may include a correction for actual road distance as a multiplier on the computed straight-line distance. Try a multiplier of 1.21 for linear coordinates for intercity travel. A circuity factor can be applied to latitude-longitude coordinate distances to account for actual traveled distance. Try a factor of 1.2 for intercity travel and 1.44 for travel within cities. See Table 14-4 of Business Logistics/Supply Chain Management 5e for additional circuity factors.
- The volume of a demand or supply point in any appropriate dimension
- The transportation rate for a demand or supply point expressed in $/unit/mile or other distance measure.
- Fixed cost for a candidate facility

All input data are prepared on the screen data editor or using an Excel spreadsheet.

EXAMPLE

Suppose that we have a small problem as shown in Figure PMED-1. There are 12 markets being served from up to 5 candidate warehouse locations. The product is shipped over a road network. The annual volumes of the markets, the transportation rates, and the candidate sites with their fixed costs are shown in the figure. Be sure to select the type of coordinates being used to locate the points in the problem.

RUNNING _PMED_

After preparing a problem database, specify the _number_ of facilities to locate. This number must be less than or equal to the number of candidate sites indicated by an **X** in the problem database. Next, click on the **Solve** button to find a problem solution. The run results are shown in Figure PMED-2. A graphical display of the solution is shown in Figure PMED-3.

Figure PMED-1 A Small Location Problem with Five Candidate Sites

```
SOLUTION RESULTS FOR P-MEDIAN PROBLEM

No. Facility name-        Volume Assigned node numbers
 1  M1                     3,000 1
 2  M4                    48,000 2 3 4 5 11
 3  M7                    33,000 6 7
 4  M9                    69,000 8 9 10 12
       Total             153,000

Total cost: $3,349.20
```
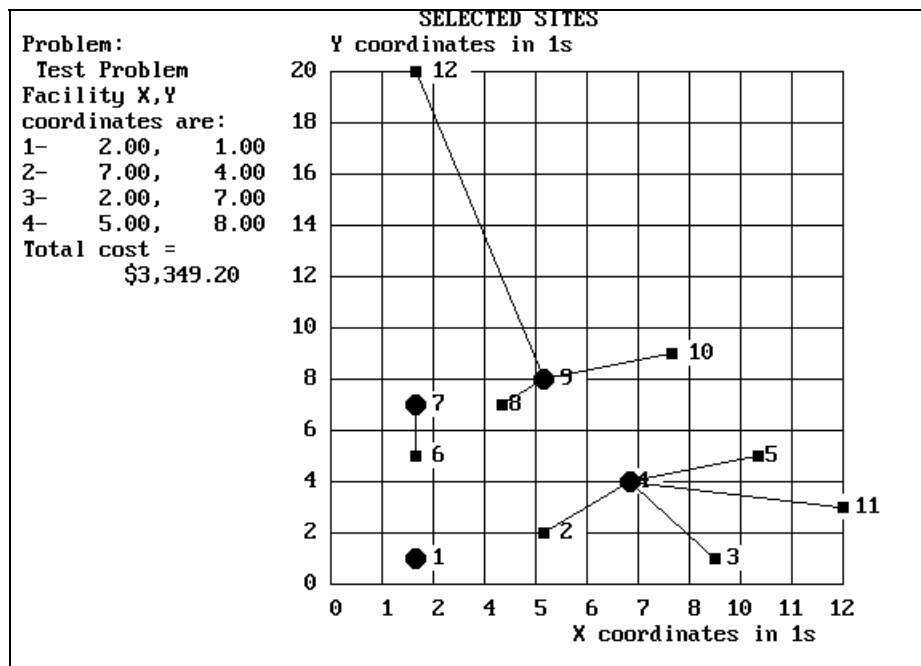
Figure PMED-2 Report on Run Results.

Figure PMED-3 Graphical Display of Solution Results

If latitude-longitude coordinates are specified, a map of the U.S. will be displayed instead of the linear grid. An example is shown in Figure PMED-4. Other geographic maps are not available.
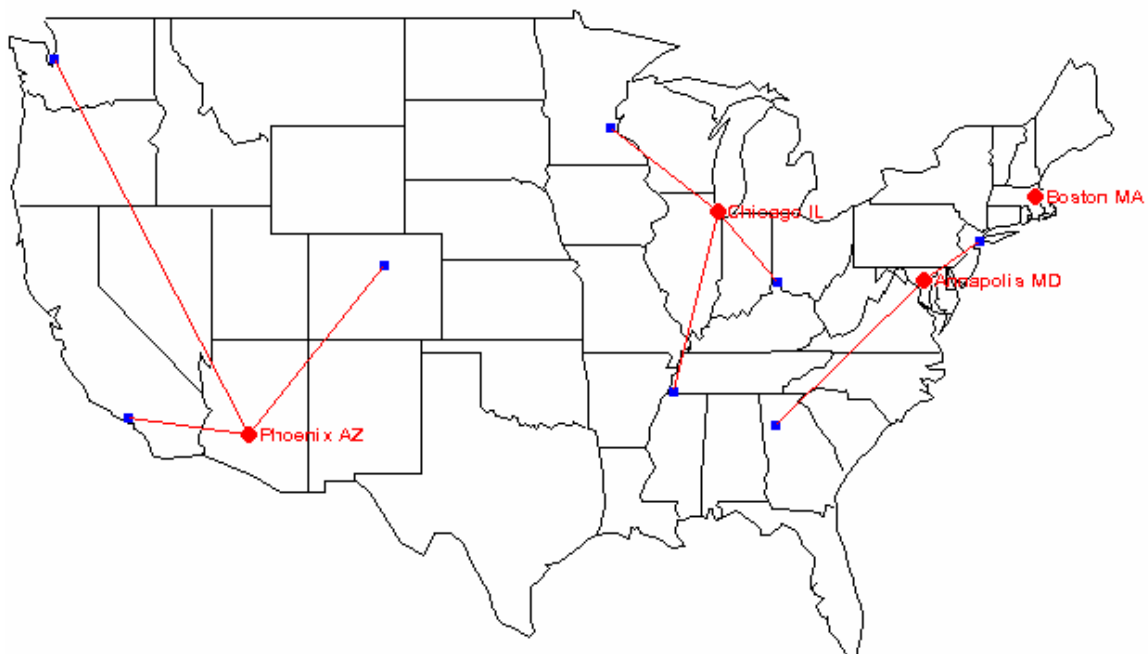

Figure PMED-4 Graphical Display of Results when Using Latitude-Longitude Coordinates.

INSTRUCTIONS FOR SOFTWARE TO ACCOMPANY
THE USEMORE SOAP COMPANY CASE STUDY
WARELOCA

WARELOCA is a computer program specifically designed to assist in the analysis of the Usemore Soap Company case study. Users should refer to the case study in Chapter 14 of Business Logistics/Supply Chain Management 5e for background and data.

The program solution procedure is based on linear programming. The user selects the particular scenario of plants, warehouses, and customer service level to be considered, and the program optimally finds the best allocation of demand to the warehouses and plants. Variable costs are minimized, subject to customer service and plant capacity constraints. Fixed costs are *not* handled within the solution process and must be added to the solution results.

INPUT

The database for the case study has been prepared and stored in a file called Uma01.Dat. Call it by using the **Open file** button. The data should only be temporarily changed during a computer run. If it is desired to save data changes, it is good practice to save them to a file of a different name from Uma01.Dat to preserve the original data.

The number of customers, warehouses, and plants are limited to those in the database; however, the data associated with each may be altered as desired. Remember that demand and capacities are expressed in hundredweight (cwt.).

RUNNING *WARELOCA*

To make a solution run, it is first necessary to select the plants and warehouses to be evaluated. Do this by placing an **X** beside the appropriate facility on the data screen. Be sure that there is adequate capacity among the selected facilities to satisfy all demand.

Next, indicate how far a customer is allowed to be from a warehouse. For example, if the distance is 300 miles, which approximates 1-day delivery service, **WARELOCA** will look for all the warehouses in your list marked with an **X**. From among those that are within 300 miles of the customer, it will select the warehouse that can *least expensively* serve the customer. If no warehouse is within 300 miles of the customer, **WARELOCA** will find the warehouse that is *closest* to the customer and assign the customer to it, irrespective of cost. That is, service considerations will override cost when the desired service distance cannot be met.

You may change the demand by region of the country. The demand data in the Uma01.Dat database is for the current year. The region in which each customer resides is shown in the customer data. For example, region 1 refers to the Northeast section of the country. Demand growth factors may be applied to customer demand by region.

Previous run scenarios are retained, and you may alter them incrementally. Once the scenario to be evaluated is prepared on the screen, the selected list of plants, warehouses, and service level is saved.

To find a solution, click on the **Solve** button.

OUTPUT
The results of a solution run can be obtained in two forms. The first is a report showing costs, facility throughput, and product flow paths through the logistics network. The second is a graphical display of the customer and facility locations, and the run results.

INSTRUCTIONS FOR THE
PRODUCT LAYOUT PROGRAM
LAYOUT

LAYOUT is a program for laying out products in warehouses and other facilities. The methods used are (1) by popularity, (2) by cube, and (3) by cube-per-order index. Products are allocated to limited space so that total travel cost to retrieve items is minimized. A picking trip originates at the outbound dock and returns to the same point. Only one product type is picked per trip, but more than one item may be retrieved on a trip. **LAYOUT** rank orders products in the way they should be allocated to the available space beginning at the outbound dock. For background on the layout methods used in the **LAYOUT** module, see Chapter 12 of Business Logistics/Supply Chain Management 5e.

INPUT

An input file is prepared with the use of the data editor. Open a previous file or start a new one. First, label the problem, avoiding the use of commas (,) and double quotation marks ("). Then, select the method for layout. Finally, enter (1) the annual sales of each product in units per year, (2) the size of a product unit in cubic feet, (3) the inventory turnover for each product, and (4) the number of orders per year on which the product appears. A typical input file might look like that in Figure LAYOUT-1.

Problem label: Test problem

LAYOUT BY
- ⊙ Order frequency (popularity)
- ○ Item size (cube)
- ○ Cube-per-order index

LAYOUT DATA

Product no.	Product label	Item sales, units/yr.	Item size, cu. ft.	Inventory turns	No. of orders/yr.
1	PRODUCT 1	10000	4.5	5.6	90
2	PRODUCT 2	20000	3.2	8.9	100
3	PRODUCT 3	40000	5.5	3.4	200
4	PRODUCT 4	50000	3.5	5.4	300
5	PRODUCT 5	3500	10.6	11.3	430
6	PRODUCT 6	5000	11.5	5	5000

Figure LAYOUT-1 Typical Data Editor Screen

RUNNING *LAYOUT*

To execute a run, click on the **Solve** button. The results report will show the products rank ordered with those products being allocated to space closest to the outbound dock listed first. The required cubic space for each product is also computed to aid in allocating the products to the warehouse space.

EXAMPLE

A warehouse contains six storage bays as shown in Figure LAYOUT-2. Each bay has 1,400 sq. ft. of storage area, and product is stacked 16 feet high. Data on five products was given previously as an example of input data (Figure LAYOUT-1). The five products are to be allocated to the available space by means of the *cube-per-order index* method. Products with small values of the index are to be located nearest the outbound dock. The computed results are shown in Table LAYOUT-1. The results are used to develop the layout as shown in Figure LAYOUT-2.

Table LAYOUT-1 Layout of the Example Problem by Cube-Per-Order Index.

LAYOUT BY CUBE-PER-ORDER INDEX

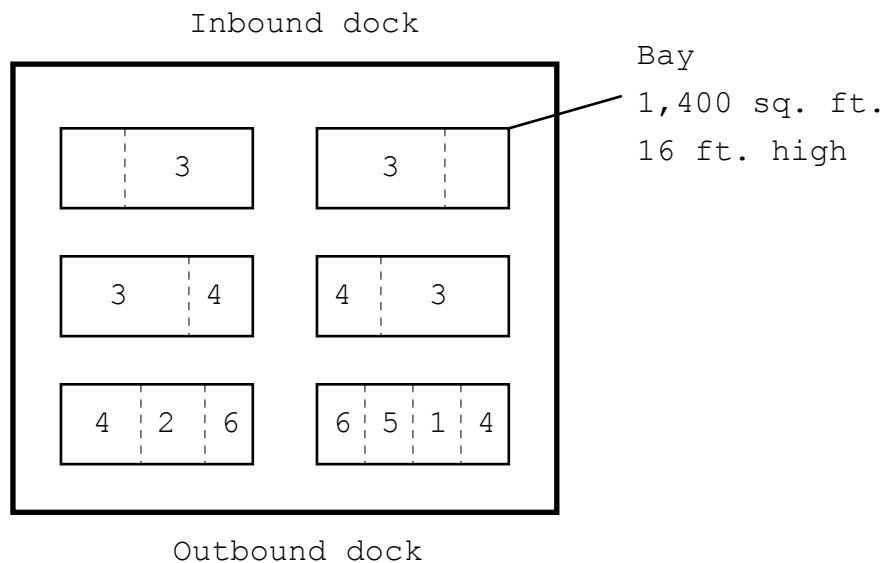| Rank	Product name-----	No. of orders/yr.	Item sales, units	Item size, cu. ft.	Cube-per- order index	Req. space, cu. ft.
1	PRODUCT 6	5,000	5,000	11.50	840	11,500
2	PRODUCT 5	430	3,500	10.60	2,787	3,283
3	PRODUCT 2	100	20,000	3.20	26,247	7,191
4	PRODUCT 1	90	10,000	4.50	32,589	8,036
5	PRODUCT 4	300	50,000	3.50	39,429	32,407
6	PRODUCT 3	200	40,000	5.50	118,088	64,706

Inbound dock

Bay
1,400 sq. ft.
16 ft. high

Outbound dock

Figure LAYOUT-2 Layout of the Five Products in the Example Problem.

INSTRUCTIONS FOR
COMPUTING DISTANCES FROM COORDINATE POINTS
MILES

MILES is a computer program to compute distances from latitude-longitude coordinates using the great circle formula and from linear coordinate points using the Pythagorean Theorem. For background on distance computation, see Chapter 14 of Business Logistics/Supply Chain Management 5e.

INPUT
Inputs are in the form of (1) a map scaling factor, (2) circuity factor, (3) distance dimension, (4) coordinate type, and (5) origin and destination coordinate. An example of a problem setup is shown in Figure MILES-1.

Figure MILES-1 An Example of the Typical Data for a Run Setup.

The map-scaling factor is used to convert coordinate points to miles or kilometers. Use a scaling factor of 1 for latitude-longitude coordinates. The circuity factor is a multiplier that converts coordinate-calculated distance to estimated road, rail, or other distance. Typical circuity factors are: 1.21 for road and 1.24 for rail, if linear coordinates are used. A factor of 1.20 is appropriate for road distances if latitude-longitude coordinates used. A circuity factor for any situation can be found by averaging a sample of the ratio of actual distance between point pairs to the calculated distance from coordinate points. Additional circuity factors may be found in Table 14-4 of Business Logistics/Supply Chain Management 5e.

RUNNING *MILES*

Click on **Calculate** to see the computed distance displayed on the screen. You may then elect to change the origin point, the destination point, or both, and compute again. The distances for subsequent calculations are accumulated. See Figure MILES-1 for an example of calculation results.

INSTRUCTIONS FOR RUNNING THE
TRANSPORTATION METHOD OF LINEAR PROGRAMMING
TRANLP

TRANLP is a software module that solves the standard "transportation problem" of linear programming. It will handle a problem of up to 30 rows and 30 columns.

Prepare a data matrix by opening an existing file or choosing a new file name for entering data. File names are of the form Tran01.dat. The data editor of the type shown in Figure TRANLP-1 will be displayed. Adjust the matrix size by entering the number of rows or columns to create the desired problem size. Click on the appropriate row, or column, for the label to be changed. Enter the desired descriptor in the label box.

Problem label: Example				Row label	
No. of rows 3		No. of columns 4		Column label	

From\To	1	2	3	Supply
A	5	4	7	1000
B	2	1	3	2000
Demand	900	700	1100	

Figure TRANLP-1 Data Shell on Editor Screen with Example Problem

The problem structure follows the standard form of the transportation method of linear programming. No dummy row or column is required. The program will add one or the other when supply and demand are not equal. To prepare a database, you will need to define the following elements.

- Number of sources (rows)
- Number of sinks (columns)
- Available supply associated with each source point
- Requirements for each sink
- Cell costs by source and sink

Refer to any good book on management science for a discussion of the transportation method.

EXAMPLE

A problem with two sources of supply and three sinks might have the matrix setup as shown in Figure TRANLP-1. Supply and requirements values should be entered as whole (integer) numbers. Cell costs may be entered as decimal values, if desired. Avoid extremely large (or small) values in the matrix, since this may exceed the computer capabilities. Rescale the data, if necessary.

RUNNING *TRANLP*

After the data have been prepared, click on the **Solve** button to execute. If errors are detected in the input data, error messages will appear.

OUTPUT

Once the program has executed, the results will appear on the screen as shown in Figure TRANLP-2. Allocation quantities are shown in the body of the matrix. A line report may also be obtained by clicking on the **Report** button. This report for the example problem is shown in Figure TRANLP-3.

Problem label: Example

Results   Total cost: 7900

From\To	1	2	3	Supply
A	700	0	0	1000
B	200	700	1100	2000
Demand	900	700	1100	

Figure TRANLP-2 Solution Screen

```
Problem label: Example
                  OPTIMUM SUPPLY SCHEDULE
----------- Cell ------------       Unit         Cell          Units
Source name    Sink name            cost         cost        allocated
A              1                     5.00      3,500.00           700
A              2                     4.00          .00             0
A              3                     7.00          .00             0
  Totals                                       3,500.00           700
Source capacity =      1,000
Slack capacity  =        300

B              1                     2.00        400.00           200
B              2                     1.00        700.00           700
B              3                     3.00      3,300.00         1,100
  Totals                                       4,400.00         2,000
Source capacity =      2,000
Slack capacity  =          0

                                      Total allocated =        2,700
                                      Slack required  =        2,700

Total cost = 7,900.00
```

Figure TRANLP-3 Computational Results of the Example Problem

INSTRUCTIONS FOR
LINEAR PROGRAMMING
LNPROG

LNPROG is computer software to solve linear programming problems by means of the two-phase simplex method. Problems of up to 50 constraints and 100 variables can be solved. Refer to any good book on management science for a discussion of the linear programming method.

INPUT
Open a previously prepared file or start a new one. The inputs to the program consist of (1) the number of constraints and variables; (2) the cost coefficients; and (3) the Right Hand Side (RHS) values. The problem setup for an example problem is shown in Figure LNPROG-1. *Note:* To reconfigure the matrix for a problem size different than appears on the screen, first click on the **Resize** button and then change the number of constraints and number of variables.

Problem label: EXAMPLE PROBLEM

Number of constraints: 4

Number of variables: 3

COEFFICIENTS

Variable\Contraint	X1	X2	X3	Type, <, =, >	RHS Value
1	10	18	8	<	5000
2	2.5	3	3	<	1200
3	1	1	1	<	600
4		1		>	100
Obj. coef.	-90	-70	-84		

Figure LNPROG-1 Data Setup Screen for an Example Problem.

Before preparing the linear programming problem data, it is necessary to convert any problem into a standard form. That is,

(1) State the objective function as a *minimizing* problem. If the objective is to maximize, multiply the objective function coefficients by a minus one (-1)
(2) Make all right-hand-side values (RHS values or "requirements") of constraints positive or zero
(3) Note whether each constraint is of the $\leq$, $=$, or $\geq$ type

EXAMPLE
Suppose we have the following problem.

```
Maximize    90X₁ + 70X₂ + 84X₃
```

Maximize $90X_1 + 70X_2 + 84X_3$

subject to:

$$10X_1 + 18X_2 + 8X_3 \leq 5000$$
$$2.5X_1 + 3X_2 + 3X_3 \leq 1200$$
$$1X_1 + 1X_2 + 1X_3 \leq 600$$
$$- 1X_2 \leq -100$$
$$\text{All } X_i \geq 0$$

We first standardize the problem. The fourth constraint is transformed to $X_2 \geq 100$ to make a positive right-hand side value. The objective function is multiplied through by a -1 to convert it to a *minimizing* problem. Thus, the problem statement would now appear as:

$$\text{Min} \quad -90X_1 - 70X_2 - 84X_3$$

subject to:

$$10X_1 + 18X_2 + 8X_3 \leq 5000$$
$$2.5X_1 + 3X_2 + 3X_3 \leq 1200$$
$$1X_1 + 1X_2 + 1X_3 \leq 600$$
$$1X_2 \geq 100$$

In matrix form, the problem now appears as setup in Figure LNPROG-1.

RUNNING *LNPROG*

After the data input screen has been prepared, click on the **Solve** button to solve the problem. A data validation check will be performed, so there may be some error messages. The solution is for the data as they appear on the screen and not necessarily the data as may be saved in a file. The results for the example problem are given in Figure LNPROG-2.

```
Problem label: EXAMPLE PROBLEM

SUMMARY OF RESULTS

 Basis          Activity Nonbasis      Opportunity
 variables      level    variables     cost
 X(1)           240      --            --
 X(2)           100      --            --
 X(3)           100      --            --
 --             --       X(4)          6
 --             --       X(5)          12
 X(6)           160      --            --
 --             --       X(7)          74
 --             --       X(8)          -74

Objective function value (Z) = 37000
```

Figure LNPROG-2 Solution Results for the Example Problem.

INSTRUCTIONS FOR
MIXED INTEGER PROGRAMMING
MIPROG

MIPROG is computer software to solve mixed integer linear programming problems by means of the branch and bound method. Problems of up to 50 constraints and 100 variables can be solved. Integer programs are sensitive to problem configurations and even slight changes to a problem can cause a long running time or an infeasible solution to occur. Refer to any good book on management science for further discussion of the integer linear programming method. An example of an integer programming application is shown in Figure 13-5 and the Technical Supplement to Chapter 13 of Business Logistics/Supply Chain Management 5e.

INPUT

Open a previously prepared file or start a new one. The inputs to the program consist of (1) constraints and variables; (2) the cost coefficients; and (3) the Right Hand Side (RHS) values, (4) constraint type, and (5) variable type. The problem setup for an example problem is shown in Figure MIPROG-1. *Note:* To reconfigure the matrix for a problem size different than appears on the screen, first click on the **Resize** button and then change the number of constraints and number of variables.

Problem label: TEST PROBLEM

Number of constraints: 2

Number of variables: 3

OBJECTIVE
- ○ Minimize
- ◉ Maximize

Variable label:

Constraint label:

COEFFICIENTS

Variable\Constraint	Var #1	Var #2	Var #3	Type, <, =, >	RHS Value
Cons #1	7	4	3	<	28
Cons #2	4	7	2	<	28
Obj. coef.	1	2	1		
Var. type	INT	INT			

Figure MIPROG-1 Data Setup Screen for an Example Problem.

Before preparing the integer programming problem data, it is necessary to convert any problem into a standard form. That is,

(1) Declare the objective function to be a *minimizing* or *maximizing* problem
(2) Make all right-hand-side values (RHS values or "requirements") of constraints *positive* or *zero*
(3) Note whether each constraint is of the $\leq$, $=$, or $\geq$ type
(4) Declare whether each variable is of the type BLANK, INT, or 0,1. A BLANK refers to the variable taking on values greater than or equal to zero, but being unrestricted in size. The variable may take on fractional values. A INT refers to

the variable having a value greater than or equal to zero but being unrestricted in size, however, the variable must be an integer (no fraction allowed.) A 0,1 refers to the variable being allowed to take on only two possible values—0 or 1.

EXAMPLE

Suppose we have the following problem.

$Maximize \quad 1X_1 + 2X_2 + 1X_3$

subject to:

$$7X_1 + 4X_2 + 3X_3 \leq 28$$
$$4X_1 + 7X_2 + 2X_3 \leq 28$$
$$\text{All } X_i \geq 0, \text{ and } X_1 \text{ and } X_2 \text{ are integer}$$

The problem now is entered into the editor screen as shown in Figure MIPROG-1. The Excel spreadsheet editor also may be used.

RUNNING *MIPROG*

After the data input screen has been prepared, click on the **Solve** button to solve the problem. The solution is for the data as they appear on the current screen and not necessarily the data as may be saved in a file. The results for the example problem are given in Figure MIPROG-2.

```
              OPTIMAL SOLUTION

 Variable        Value        Rate            Cost Variable label
 X(1)    =       .0000       1.0000          .0000 Var #1
 X(2)    =      2.0000       2.0000         4.0000 Var #2
 X(3)    =      6.6667       1.0000         6.6667 Var #3

 Objective function value =       10.67
```

Figure MIPROG-2 Solution Results for the Example Problem.

INSTRUCTIONS FOR THE
MULTIPLE REGRESSION PROGRAM
MULREG

MULREG is a stepwise multiple regression program that is designed to handle up to 5 variables and 999 observations. Multiple regression analysis as a technique is not described in these instructions. The reader is referred to any good book on statistics covering multiple regression analysis.

INPUT
Open a previously prepared file or start a new one. A data screen of the type shown in Figure MULREG-1 is to be prepared. Each data item is discussed.

| Problem label | TEST PROBLEM | | | Number of observations | 29 |

VARIABLE IDENTIFICATION

	Label	Type (D/I)
Variable #1	RATE	D
Variable #2	DISTANCE	I
Variable #3		
Variable #4		
Variable #5		

Want a stepwise regression (Y/N)? N

Incremental F-level for entering variable, e.g., 0.05 — 0

Incremental F-level for leaving variable, .e.g., 0.05 — 0

Want an error analysis (Y/N)? Y

OBSERVATIONS

Obs no.	Data for RATE	Data for DISTANCE	Data for var. #3	Data for var. #4	Data for var. #5
1	22.34	822	0	0	0
2	11.95	92	0	0	0
3	20.02	557	0	0	0
4	15.48	237	0	0	0
5	17.67	214	0	0	0
6	30.97	1576	0	0	0
7	31.31	1639	0	0	0
8	23.54	1008	0	0	0
9	17.9	362	0	0	0
10	26.52	1169	0	0	0
11	14.16	197	0	0	0
12	18.34	465	0	0	0

Buttons: Open, Save data, Transform, Plot data, Print Data, Solve, Exit

Figure MULREG-1 Example of Data Input Screen.

Parameters and Labels
Problem label. The problem label is an identifier for the problem. Do not use commas (,) or double quotation marks (") in the label. Keep the descriptor to 15 characters or less.

Number of observations. Entering the number of observations in a problem changes the size of the data input matrix.

Variable label. Each variable may be identified with a label. Keep the descriptor to 10 characters or less. Do not use commas (,) or double quotation marks (") in the labels.

Variable type. Up to five variables are allowed. Each variable is identified as (1) a dependent variable (*D*) or (2) an independent variable (*I*). Only one variable can be declared an dependent variable. The remaining four variables may be independent or left blank.

Stepwise regression. Invoking the stepwise procedure allows the independent variables to be selected based on their contribution to reducing the error in the dependent variable. If the stepwise procedure is not used, the regression analysis is conducted simultaneously on all independent variables.

When the stepwise option is selected, entering and leaving variables may be controlled according to their statistical contribution. If a zero (0) F-level value is given for both entering and leaving variables, all variables will be brought into the regression equation. Incremental F-level values greater than zero may cause some variables to be omitted from the regression equation.

Transformed input data. Invoking the transform data option allows specified variables to be transformed according to several predefined functions. The particular variable to be transformed is selected by clicking on the appropriate column OBSERVATIONS matrix. Then, click on the **Transform** button and select the desired transform function.

Print data. A listing of the problem setup and data can be obtained by clicking on the **Print data** button.

Plot data. Clicking on the **Plot data** button invokes plotting capability. A plot of the dependent variable against the selected independent variable can be seen. The plot is made after any transformation of a variable has been carried out.

Error analysis. Provides additional information about how well the regression line fits the data. This invokes a plot of residuals in the output report.

Observation Data

The observations for each variable are entered on the screen. There may be a maximum of 999 entries. *Data for at least two variables must be provided.* Scaling of the data can be important to keep it within the range of the program's plotting and presentation capabilities. A good rule of thumb is to scale the data so that it has a range of approximately *xxx.x*. If the output presentation is not satisfactory, try re-scaling the data with the editor or the transformation procedure.

RUNNING *MULREG*

After the data are prepared with data editor, execute the analysis by clicking on the **Solve** button. A report of the solution results will appear.

EXAMPLE

Suppose that we wish to develop a curve to estimate class transportation rates as a function of distance. From rate data collected for 5000 lb. shipments over varying distances, we have 29 observations. A listing of the input data is shown in Table MULREG-1 and its graphical presentation is given in Figure MULREG-2. The results of analysis are illustrated in Figures MULREG-3 to MULREG-5.

Table MULREG-1 Observation Data for Example problem

RATE	DISTANCE
22.340	822
11.950	92
20.020	557
15.480	237
17.670	214
30.970	1576
31.310	1639
23.540	1008
17.900	362
26.520	1169
14.160	197
18.340	465
46.050	2803
29.230	1278
43.120	2471
46.960	2785
36.610	1880
47.870	2944
22.740	852
20.020	656
26.520	1269
26.570	1351
46.960	2818
29.230	1237
15.390	336
34.310	1829
46.050	2832
47.870	2965
21.340	862

From running **MULREG,** the rate estimating equation that we seek is:

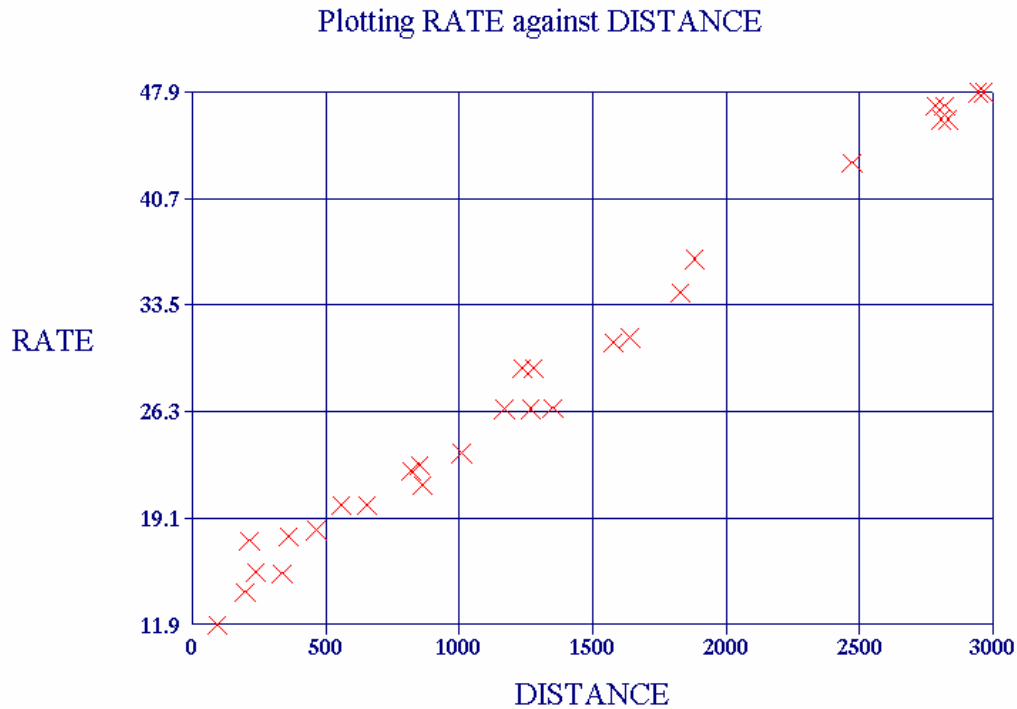$$Rate = 12.36 + .012 \times Distance$$

with an R-square of 0.9911.

Figure MULREG-2 A plot of the input data for the example problem.

```
ANALYSIS OF VARIANCE

Source                SS   df           MS      F-value
Regression    3794.70508    1   3794.70508   3020.22241
Residual        33.92358   27      1.25643
     Total    3828.62866   28



SUMMARY

The estimating equation is:
    Y (RATE) =    12.3570     +    0.0121 x DISTANCE

with a standard error of estimate =     1.1209
and an R-square of  0.9911
```

Figure MULREG-3 Computational results for the example problem.

```
ERROR ANALYSIS

Obs    Actual Estimated   Dif-         Expected range    Out of
no.       Y         Y     ference         2xSEE          range
 1    22.340    22.317     0.023      20.075 -  24.559
 2    11.950    13.472    -1.522      11.230 -  15.714
 3    20.020    19.106     0.914      16.864 -  21.348
 4    15.480    15.229     0.251      12.987 -  17.471
 5    17.670    14.950     2.720      12.708 -  17.192      Y
 6    30.970    31.453    -0.483      29.211 -  33.695
 7    31.310    32.216    -0.906      29.975 -  34.458
 8    23.540    24.571    -1.031      22.329 -  26.813
 9    17.900    16.743     1.157      14.501 -  18.985
10    26.520    26.522    -0.002      24.280 -  28.763
11    14.160    14.744    -0.584      12.502 -  16.986
12    18.340    17.991     0.349      15.750 -  20.233
13    46.050    46.320    -0.270      44.079 -  48.562
14    29.230    27.842     1.388      25.600 -  30.084
15    43.120    42.298     0.822      40.056 -  44.539
16    46.960    46.102     0.858      43.861 -  48.344
17    36.610    35.137     1.473      32.895 -  37.378
18    47.870    48.029    -0.159      45.787 -  50.271
19    22.740    22.681     0.059      20.439 -  24.922
20    20.020    20.306    -0.286      18.064 -  22.547
21    26.520    27.733    -1.213      25.491 -  29.975
22    26.570    28.727    -2.157      26.485 -  30.969
23    46.960    46.502     0.458      44.260 -  48.744
24    29.230    27.346     1.884      25.104 -  29.587
25    15.390    16.428    -1.038      14.186 -  18.670
26    34.310    34.519    -0.209      32.277 -  36.760
27    46.050    46.672    -0.622      44.430 -  48.914
28    47.870    48.283    -0.413      46.042 -  50.525
29    21.340    22.802    -1.462      20.560 -  25.044
```

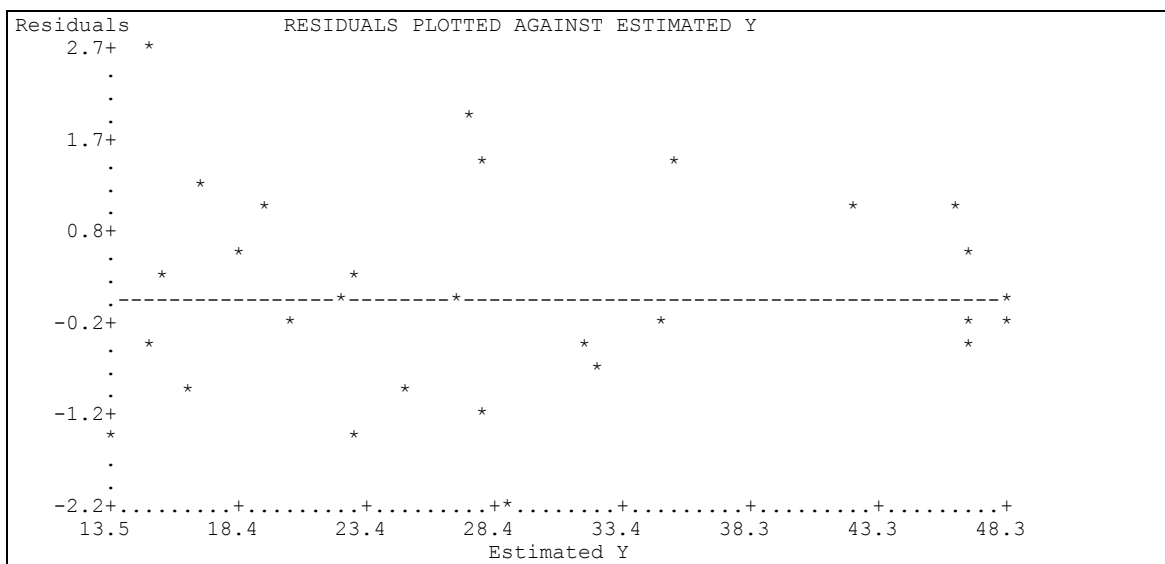**Figure MULREG-5** Error analysis for the example problem.

Figure MULREG-6 Plot of residuals for the estimating equation.

INSTRUCTIONS FOR THE
SUPPLY CHAIN SIMULATION PROGRAM
SCSIM

SCSIM is computer software to mimic the actual flows of product through a supply channel having four echelons ranging from factory to customers, as illustrated in Figure SCSIM-1. The product flows are replicated with a Monte Carlo-type simulation. A single, or aggregate, product is used and a single facility, or aggregation of facilities, is assumed at each echelon. The objective is to simulate supply channel performance and costs when various forecasting methods, inventory policies, transportation services, production lot sizes, order-processing costs, and the like are used throughout the supply channel. Reports and graphs are used to portray supply chain performance for different simulation runs.
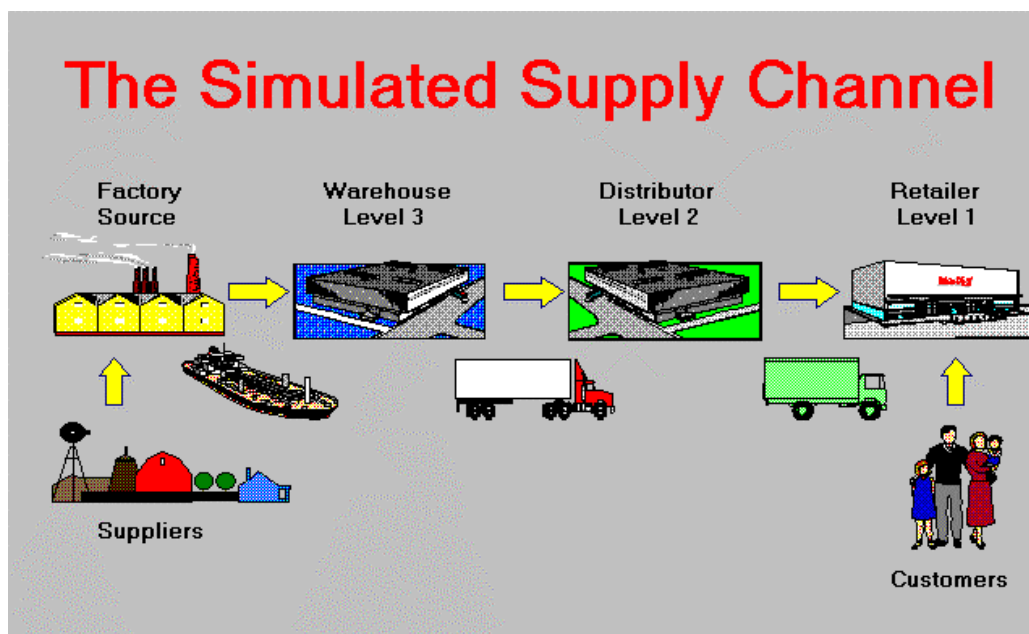

Figure SCSIM-1 The General Supply Channel for SCSIM

ABOUT SIMULATIONS
A computer simulation is an experimental methodology where events and occurrences, such as product demand and transit times, are replicated by obtaining their values from a probability distribution. Random numbers distributed according to a probability distribution of the event being represented are used to mimic the variability in the event, such as shipping times. A seed number is used to initiate the random processes. Starting with the same seed number will produce the same results if the data inputs are always identical. However, different seed numbers produce separate sample observations without changing the input data. To obtain a correct picture of channel performance, a number of runs should be made using different seed numbers to create a reasonable sample size. The results in the sample can then be averaged. Comparisons of the

samples having different data sets are best made by performing hypothesis tests on the averaged results to see if there are statistically significant differences. A comparison of single runs is risky since the observed differences may simply be due to random variations that are not statistically significant.

Simulation results can be sensitive to start up conditions. That is, certain assumptions are made when starting the simulation, such as inventories are zero, that are not true after the simulation output has stabilized. The simulation needs to run for a number of periods before stabilization can occur. It cannot be known in advance how long this initialization will take, so watch the graphs to see when steady state (repeatable patterns) in the output has occurred. **SCSIM** uses a one-year period for initialization. If steady state has not occurred, run the simulation for additional years, and then use the results only from the later years where steady state has been observed.

For further information about the simulation process, see a good book on management science covering Monte Carlo simulation or a book that is specifically devoted to simulation.

ABOUT THIS SIMULATION
A number of assumptions have been made in the operation of this simulation. These are not under the control of the user, but they are noted here for better understanding of the action of the simulator.

- All distributions are normal with a minimum average value of 1
- Any draw on a normal distribution producing a number less than 1 is truncated to 1
- The time period is one day with a week being 7 days, a month being 30 days, and a year being 360 days
- Quantities arriving at an echelon in the channel for a particular day are added to inventory before deductions from inventory due to sales are made
- The minimum value for demand, sales, and production lot size is 1
- Product volumes are in whole numbers
- Numbers as input data or as computed values greater than 1 billion should be avoided as they may cause overflow errors
- All unfilled orders become back orders, not lost sales, until adequate inventory is available to eliminate the back orders

Although the results can be seen in the graphs and reports of the simulation, the day-to-day quantity flows throughout the supply channel can be observed in the file named RESULTS. It is updated each time the simulation is run and is a comma-delimited file that can easily be read in a spreadsheet program such as Excel. The simulation results are shown for all periods after the initialization year. *Note*: The RESULTS file can become quite large if long simulation runs are requested. Be sure to have adequate file storage space available in the output folder.

INPUT
The data inputs consist of simulation control information, demand projection data, sales and cost data, forecasting and inventory control methods, and transportation choices with statistical performance data. Any units of product flow may be used, but control the data

scaling so that the numbers are neither too large nor too small. Time is always expressed in days. Costs, revenues, and profits are expressed in $. Although any monetary unit may be used, only $ will show on the input data screens and the output reports. Each data element is briefly discussed as follows.

Initialization
Seed number. A seed number needs to be specified to provide an entry point to the random number generator. Different seed numbers will provide different simulation results and, therefore, different sample observations. *Note*: Changing a seed number after a run has already been made will produce a different result than when the same number is used for the first run.

Length of simulation. The length of a simulation run is specified. Usually just a few simulated years are needed to produce steady-state results. A maximum of 20 simulated years is allowed, but the RESULTS file becomes quite large and may not fit on a 1.44 MB floppy disk, if a floppy disk has been selected for the location of **LOGWARE** output.

Selling price. The selling price is the price charged to the end customer in the supply channel, in $/unit.

Customer Demand Pattern
End customer demand can be generated randomly or it can be specified.

Generate demand pattern. Selecting to generate demand will produce a daily demand pattern from a normal distribution. Inputs require a daily demand average and a standard deviation, in units. Any demand generated that is less than one is set to one unit.

An annual growth rate is a percentage of the average demand. Seasonal indices may be specified on a monthly basis. Each monthly seasonal index is nominally a value of 1. Since it is a multiplier to demand, an index of 1.25 means that demand for that month is 1.25 times the monthly average demand. An index of 0.5 means actual demand is one-half of the average monthly demand.

Specify demand pattern. Specific demand values may be specified for each day of a 360-day year. This pattern is repeated for each simulated year. An annual growth percentage may be used as a multiplier to these values.

Inputs for Levels 1, 2, and 3
The inputs are the same for all three echelons, so they are discussed jointly.

Product item data
Item value. Data relating to inventory include item value, carrying cost, in-stock probability, and back-order cost. The item value refers to the worth of the product at the particular level in the channel. The item value is the accumulation of purchase, production, transportation, and order-processing costs to the point in the channel where inventory is being held.

Inventory carrying cost. Inventory carrying cost is the percent per year of the item value that it costs to hold a product as inventory. It is the total of the cost of money,

insurance on inventory, personal property tax, value loss due to obsolescence, and storage costs. Twenty to forty percent per year is a common range.

In-stock probability. In-stock probability is the desired percentage of sales that is being filled on request. A probability less than 100% permits some back ordering to occur, but setting in-stock probability at 100% may force exceedingly high inventory levels to occur, depending on the inventory control method selected.

Back-order cost.   The back order cost refers to the cost for handling an order outside of the normal processes when the order cannot be filled on request due to an inventory shortage. The cost can be a result of extra paperwork, premium transportation, extra communication efforts, and the like.

Customer/retailer/distributor/warehouse order filling.   The cost to fill an order received at a facility refers to the cost associated with credit checking, inventory availability checking, stock picking, and shipping preparation. The cost is expressed in $/unit. The time to accomplish the order filling is expressed as the average time and the associated standard deviation, both in days.

Purchase order processing. This is the cost incurred by the buyer to prepare a purchase order being *sent* to an upstream supplier. It should not be confused with the previous cost for *processing* a purchase order being received by an upstream seller. The purchase order cost generally will be a clerical cost associated with supplier communication, item checking, order transmission, and the like. It is expressed in terms of $/order.

Forecasting method

Three forecasting methods used to project future sales at a facility are allowed. Two are forecasting from a time series and the third is a user's specified forecast. (See Chapter 8 of Business Logistics/Supply Chain Management for more information on forecasting methods.)

Exponential smoothing. Exponential smoothing forecasting is the level-only type. That is, the forecasting model has the general form:

$$F_{t+1} = \alpha A_t + (1-\alpha)F_t$$

The smoothing constant (α) is a value between 0 and 1. Values between 0.05 and 0.3 are common. High values give more rapid response to a changing time series, whereas lower values cause the forecast to be less affected by random variations in the time series. A_t are the sales in period t, and F_t is the previous forecast in period t.

Moving average. Moving average forecasting is a simple type of the form

$$F_{t+1} = \frac{\sum_{t}^{t-N} A_t}{N}$$

where A_t is the sales in period t, and N is the number of periods (days) over which the most recent N periods of sales are to be averaged. Smaller N values give a more

responsive forecast to shifting trends and seasonality but can be overly sensitive to random variations. Higher N values give the opposite effect.

Specified forecast. The specified forecast allows the user to provide the average daily sales value. This value remains in effect during the simulation until changed by the user. The interval is specified as an input value. The user is prompted for a sales value at the specified interval.

Reorder policy

Inventory replenishment can be controlled by three methods: reorder point, periodic review, and stock-to-demand.

Reorder point. The reorder point method (also known as the trigger point method, $S s$ policy, Q-system, and fixed quantity-variable order interval method) controls inventory levels through two values: the reorder size (Q) and the quantity (ROP) that triggers when to place a replenishment order. That is, when the inventory level of an item drops to ROP, replenish inventory with an amount Q. The user may select the option to have these values calculated or may choose to specify them.

Periodic review. The periodic review method (also known as the P-system or fixed order interval-variable order quantity method) controls inventory levels through two values: the review interval (T) and a target level (MAX). The inventory level is observed every T days, and a replenishment order is placed for the difference in the MAX level and the quantity on hand. The user may select the option to have these values calculated or may choose to specify them.

Stock-to-demand. Stock-to-demand control is similar to the periodic review method except that the MAX level is determined as a multiple of the *number of days* in the forecast. For example, the MAX level might be set at 1½ times the monthly forecast, or 45 days. A higher multiple forces more safety stock into inventory. The review interval T might be a convenient review interval such as 30 days. The user sets these values.

Factory/Source

Production cost and lot size. The cost to produce a unit of product at the factory is the standard cost associated with a particular product. It is expressed in $/unit. The lot size in units of sales is the minimum quantity to be produced at one time. Lot size is related to the standard cost of production.

Production time. The time required to produce a lot quantity is expressed as the average time and the standard deviation, both in days.

Purchase cost. This is the cost in $/unit to purchase the materials to build the product.

Order-filling cost. The cost to fill an order received at the factory refers to the cost associated with credit checking, production scheduling, and shipping preparation. The cost is expressed in $/unit. The time to accomplish the order filling is included in the production time.

Transportation

Shipping between supply channel echelons can be accomplished by a number of transportation modes and their service offerings. Each transportation service can be

represented by its cost per unit and its delivery time. Speed and reliability are expressed as the average transit time and the standard deviation of transit time, both in days.

RUNNING *SCSIM*
Example Problem
To illustrate the running of **SCSIM**, consider the database shown in Table SCSIM-1. This problem shows a variety of options throughout the supply channel. Once the database is prepared, click on the **Solve** button to execute the simulation. A 2-year simulation is run, and some of the results are shown below without judging their "quality."

Table SCSIM-1 Database for example problem
SIMULATION DATABASE

Title: Sample problem

Initialization
123456	Seed value
2	Length of simulation, years
75	Selling price, $/unit

Customer demand pattern
Generate daily demand
100	Average daily demand, units
20	Standard deviation of daily demand, units
0	Annual demand growth increment, %

Monthly seasonal indices
Month	Index	Month	Index	Month	Index	Month	Index
1	1	4	1	7	1	10	1
2	1	5	1	8	1	11	1
3	1	6	1	9	1	12	1

Retailer/Level 1
Product item data
40	Item value in inventory, $/unit
5	Customer order filling cost, $/unit
3	Purchase order processing cost, $/order
21	Inventory carrying cost, %/year
1	Average customer order fill time, days
0.1	Customer order fill time standard deviation, days
95	In-stock probability, %
5	Back order cost, $/unit

Forecasting method
 Exponential smoothing
0.3	Smoothing constant

Reorder policy
 Reorder point control method
 Calculate inventory control parameters (Q, ROP)

Distributor/Level 2
Product item data
35	Item value in inventory, $/unit
0.7	Retailer order filling cost, $/unit

10	Purchase order processing cost, $/order
25	Inventory carrying cost, %/year
2	Average retailer order fill time, days
0.2	Retailer order fill time standard deviation, days
95	In-stock probability, %
20	Back order cost, $/unit

Forecasting method
  Moving average
  120   Number of periods
Reorder policy
  Stock-to-demand control method
   45   Target days of inventory
    0   Review time in days

Warehouse/Level 3
Product item data
30	Item value in inventory, $/unit
0.5	Distributor order filling cost, $/unit
50	Purchase order processing cost, $/order
20	Inventory carrying cost, %/year
2	Average distributor order fill time, days
0.1	Distributor order fill time standard deviation, days
90	In-stock probability, %
10	Back order cost, $/unit

Forecasting method
  Moving average
  360   Number of periods
Reorder policy
  Periodic review control method
   30   Order review interval (T), days
  6500   Target level (Max), units

Factory/Source
Product item data
12	Production cost, $/unit
1000	Minimum production lot size, units
0.3	Warehouse order filling cost, $/unit
5	Average production time, days
0.5	Production time standard deviation, days
40	Purchase cost, $/unit

Transportation
Transport between Distributor and Retailer
1.25	Transport cost, $/unit
2	Average time in-transit, days
0.2	Transit time standard deviation, days

Transport between Warehouse and Distributor
2.09	Transport cost, $/unit
3	Average time in-transit, days
0.3	Transit time standard deviation, days

Transport between Factory and Warehouse
2.45	Transport cost, $/unit
5	Average time in-transit, days
0.5	Transit time standard deviation, days

Simulation Results

After running a simulation, click the **Results** button, which will be enabled as a result of making a run. The results are shown in both graphical and report forms. Various combinations of graphs and reports can be selected. You have the option of displaying the graphs in daily, weekly, or monthly periods. Whereas the daily period plots display the minimal time unit of the simulation, weekly and monthly periods average the results for the selected period. When entering a graph or report, you have the option of presenting all of the simulated years or a portion thereof. This is useful for presenting the simulated period(s) that are most stable and representative of the steady state condition and for controlling the display resolution for the graphs.

An example of graphical output is shown in Figure SCSIM-2 for the retailer/level 1 in the channel. A number of such graphs can be selected. Select the period for display.

Table SCSIM-2 shows a sample performance/financial report for the warehouse/level 3 echelon of the channel. A variety of reports may be selected. Within the report, results are shown as a yearly average and for the simulated period. The yearly average is the *average* of a result for the selected period of the report, whereas the result for the simulated period is the *sum* of the result values over the simulated period.

The reported financial items have the following meaning.

Revenue. The selling price per unit multiplied by the retailer sales.

Cost of purchased goods. The unit purchase cost times retailer unit sales.

Gross margin. The difference between retailer sales revenue and the associated purchased goods cost.

Production cost. Retailer sales multiplied by production cost per unit.

Transportation costs. Transportation rate times the shipping volume moving between echelons within the channel.

Sales order handling cost. The cost for a seller to process the sales orders received. It is the unit order filling cost times the unit sales volume.

Order-processing cost. The cost for a facility to prepare a purchase order placed on an upstream seller. It is the purchase order-processing cost times the number of purchase orders placed.

Inventory cost. The cost of an item in inventory times the inventory carrying cost times the average inventory level at a particular echelon.

Back-order cost. The volume of units back ordered at an echelon times the unit back-order cost.

Total cost. The sum of the costs for a particular echelon.

Net profit contribution. The difference between revenue and the sum of all supply channel costs.

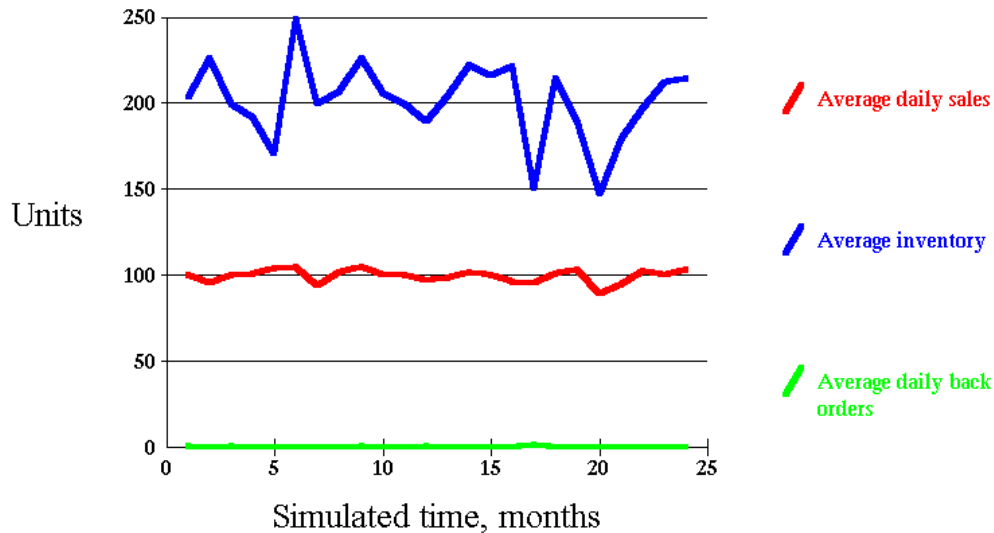## RETAILER PERFORMANCE FOR YEARS 1 THROUGH 2

Figure SCSIM-2 An Example of a Graphical Display

The reported performance items have the following meaning.

Sales forecast. The expected sales for a particular echelon as projected by the selected forecasting method.

Sales to channel member. The actual sales received by a channel member. This will usually be less than customer demand due to the inability to fill all of demand on request.

Average inventory on hand. The average inventory level for the selected reporting period.

Inventory turnover ratio. The ratio of a channel member's sales to the average inventory on hand.

Daily back orders. A channel member's average daily back-order volume.

Back-order occurrences. The number of back-order occurrences that took place within the reporting period.

Average demand filled on request. A service measure expressed as a percent of echelon member demand that could be filled at the time the demand occurred. Demand is either customer demand, in the case of the retailer, or the purchase orders placed on a distributor, warehouse, or factory.

Daily quantity on order. The average daily units placed as orders on an upstream seller that have not yet been received.

Number of orders placed. Number of purchase orders placed on the upstream echelon for the selected period.

Table SCSIM-2 An Example of an Output Report

WAREHOUSE REPORT FOR SIMULATED YEARS 1 TO 2
Forecasting method: Moving average
Inventory control method: Periodic review

Yearly average	Simulated period	PERFORMANCE STATISTICS
36,411	72,822	Sales Forecast, units
35,748	71,496	Retailer sales to retailer, units
2,397.32		Average inventory on hand, units
14.91		Inventory turnover ratio
26.62		Daily back orders, units
73.20%		Average demand filled on request
1,236.00	2,472	Back order occurrences
1,291		Daily quantity on order, units
12	24	Number of orders placed
		FINANCIAL PERFORMANCE
600.00	1,200	Cost to process factory orders
17,874.00	35,748	Cost for handling distributor orders
14,383.90	28,768	Cost for carrying inventory
12,360.00	24,720	Cost for back orders
74,713.32	149,427	Cost for transport to distributor
$119,931.22	$239,862	Total cost