Overview of QM for Windows Decision Support Software (Version 2+)

Management Science 321: G. M. Claffie (October 2004)

QM for Windows decision support software is used to model and solve quantitative management problems. This software is written/supported by Howard J. Weiss and distributed by Prentice-Hall (www.prenhall.com/weiss, ISBN: 0-13-145066-2).

This primer provides an overview of:

- How to access **QM for Windows** (Version 2.1/Build 71) at Rutgers-Camden.
- How to use **QM for Windows** (Version 2.1/Build 71) to formulate (model) and solve linear programming, goal programming, transportation and PERT/CPM problems.

The computer assignments in this course should be much easier to complete if you **first** replicate and understand the related example in this primer and **then** model and solve your specific problem.

Access to QM for Windows

- **QM for Windows** is installed on the IBM PC compatible computers in the Business and Science Building computer rooms and in the Campus Center. You must have a computer account at Rutgers-Camden to access this software. Call 856-225-6326 if you are not sure when the computer rooms are open.
- When you login, leave "**mount home directory**" checked so that you can save problems on the networked **H**: drive. Save your work on the **H**: drive or on a portable disk in the **A**: drive.
- After you have logged in, double-click the **Business Packages** folder on the windows desktop to access the **Business Packages** window. Then double-click the **QM for Windows 2** icon (not the **QMWIN** icon) in the **Business Packages** window to access the **QM for Windows** screen.
    - <u>To begin a problem</u>: Select the desired **QM for Windows** module via the **Module** menu. Then select the **File** menu, **New** option.
    - <u>To continue working on a saved problem</u>: Access the **Open file** screen via the **File** menu, **Open** option. Then select the disk/directory location and the desired file with the appropriate file extension.
- To save a problem access the **Save File/Problem** screen via the **File** menu, **Save As** option. Then enter the desired disk/directory location and the file name. The file extension for each **QM for Windows** module is indicated on the **Save File/Problem** screen.

Solution of Linear Programming Problems

A linear programming model consists of a linear objective function to be maximized (or minimized) and linear constraints that limit the results that can be achieved. This model can be written in a matrix form with the rows corresponding to the constraints (the row index is i) and the columns corresponding to the variables (the column index is j). The objective function is the sum of coefficient (cj) and decision variable (xj) products in the form: Max (or Min) c1x1+c2x2+c3x3+.... The constraints consist of coefficients (aij), decision variables (xj) and resource or requirement limits (bi). There can be upper limits (≤ bi), exactly required limits (= bi) or lower limits (≥ bi). For example, the first constraint in row 1 of a linear programming model has the form:

$$a11x1+a12x2+a13x3+\ldots \leq b1 \text{ (or} = b1\text{, or} \geq b1)$$

The linear programming model is used (solved) to determine the optimum decision variable values that produce the best possible (optimal) solution. This course considers problems where the decision variables are continuous. The values of these variables may be zero or any positive, real number. It does **not** consider integer linear programming problems where the decision variable values must be integers.

Demonstration Example

Consider the Par, Inc. golf bag production problem in Chapters 2 and 3 of the textbook. The **QM for Windows** Linear Programming module will be used to determine the number of standard golf bags, and the number of deluxe golf bags, that should be produced to maximize contribution to profit.

The unit contribution to profit for each type of golf bag, the amount of time required by the four production operations that are used for each type of bag, and the amount of time available for these operations are shown in the following table.

	Production Time (hours)				Unit Profit
Product	Cut/Dye	Sewing	Finish	Inspect/Pack	Contribution
Standard	7/10	1/2	1	1/10	10
Deluxe	1	5/6	2/3	1/4	9
Avail. Hrs.	630	600	708	135	

QM for Windows (continued)

Let x1 = the number of standard golf bags to be made and let x2 = the number of deluxe golf bags to be made. For this case the linear programming model to optimize the value of these decision variables and maximize profit may be written:

$$
\begin{array}{lrrl}
\text{Max} & 10x1 + 9x2 & & \\
\text{s.t.} & & & \\
 & 7/10\ x1 + \ 1\ x2 & \leq 630 & \text{(Cutting and Dyeing)} \\
 & 1/2\ x1 + 5/6\ x2 & \leq 600 & \text{(Sewing)} \\
 & 1\ x1 + 2/3\ x2 & \leq 708 & \text{(Finishing)} \\
 & 1/10\ x1 + 1/4\ x2 & \leq 135 & \text{(Inspection and Packing)} \\
 & x1, x2 & \geq \ 0 & \text{(Decision variable non-negativity constraint)}
\end{array}
$$

PROBLEM SETUP: This problem can be entered into **QM for Windows** as follows:

- Select (double-click) the **Business Packages** window, **QM for Windows 2** icon to access the **QM for Windows** screen.
- Select the **Module** menu, **Linear Programming** option. Then select the **File** menu, **New** option to access the **Create data set for Linear Programming** screen. Use this screen to enter initial problem information as follows:
  - Title: Enter: **Par Demonstration Example**
  - Number of Constraints: Enter **4** using the scroll bar. There are four production constraints. The linear programming model non-negativity constraint, which is "understood" by the software, is not counted or entered.
  - Number of Variables: Enter **2** using the scroll bar. There are two decision variables: x1 for the number of standard golf bags to be made and x2 for the number of deluxe golf bags to be made.
  - Objective: Select **Maximize** to maximize profit.
  - Row names tab: Select **Constraint 1, Constraint 2,** …
  - Column names tab: Select **X1, X2, X3,** …
  - After the above entries and selections are made, click **OK** to access the **[Data Table]** screen as shown in Figure 1. (Figure 1 was printed via the **Print Screen** button at the bottom of this screen.)

Highlight and change (edit) the default decision variable names and the constraint names in the data entry table to more clearly describe the problem. Use S instead of x1 for the number of standard golf bags to be made and D instead of x2 for the number of deluxe golf bags to be made. Then edit the constraint names to identify the four manufacturing operations as shown in Figure 2.

QM for Windows (continued)

After editing, enter the model parameter values and the constraint relationships to complete the model as shown in Figure 2. Select the constraint ($\leq$, or =, or $\geq$) relationships via a drop-down menu that can be accessed from each constraint relationship cell. Use decimal equivalent values for all fractions.

PROBLEM SOLUTION: Click the **Solve** button at the top of the **[Data Table]** screen to obtain the **Linear Programming Results** screen shown in Figure 3. The optimized solution on this screen shows that:

- Profit is maximized at about $7668.00 if 540 standard golf bags and 252 deluxe golf bags are made.
- The positive numbers in the **Dua**l column indicates that profit will increase if additional time is made available in the CutDye operation or in the Finishing operation. Therefore, these are binding constraints that are limiting profit. The **Dual** values of zero for the Sew operation and for the Inspect/Pack operation indicate that these operations are not limiting profit. Some of the available hours in these production operations are not being used.

Additional solution information can be displayed via the **Window** menu or by selecting an icon at the bottom of the **Linear Programming Results** screen.

Software Error Note: The **Dual Value** (Dual Price) must always be positive for binding less-than or equal-to ($\leq$) constraints. Also, the **Dual Value** must always be negative for binding greater-than or equal-to ($\geq$) constraints. Some versions of the **QM for Windows** software will show an incorrect sign for the **Dual Value** of a binding constraint.

Solution Printing Note: The computer assignments in this course require that all printouts be generated by selecting options in the **Print Setup screen** as described below.

SOLUTION PRINTOUTS: Starting from the **Linear Programming Results** screen, access the **Print Setup** screen **Information** tab via the **File** menu, **Print** option. Then select all of the available **Information to Print** options, select the **Constraint style, Equations** option and click **Print** to obtain the printout shown in Figure 4. The **Information to Print** options are summarized below.

<u>**Results**</u>: This option prints the linear programming model that has been optimized as shown in Figure 4 (if the **Constraint style, Equations** option is selected). The **Results** label that is printed for this option is misleading since the linear programming model is printed, not the results.

QM for Windows (continued)

<u>Ranging</u> <u>(Sensitivity Analysis)</u>: This option shows the impact of changing the value of **one** objective function coefficient, **or** the value of **one** constraint limit (RHS value) on the value of the optimized decision variables and the optimal value of the objective function. All of the other model parameters must be unchanged from their original values.

- **Range of Optimality Information**: The optimized values of the decision variables S and D are shown. The current value of S continues to be the optimum number of standard bags to produce provided that the coefficient of S in the objective function (which is currently 10) lies anywhere between 6.3 and 13.50. This means that 540 standard golf bags should be made if the profit contribution from each standard bag is between \$6.30 and \$13.50. The Reduced Cost for decision variable S is zero, since S is greater than zero and contributes to the value of the solution. If the value of S were zero, the Reduced Cost would indicate the amount that the unit profit for a standard golf bag must increase before any standard golf bags should be manufactured. Similar comments can be made for decision variable D.
- **Range of Feasibility Information**: The CutDye constraint is binding since all of the available CutDye hours are being used (slack is 0). The Dual Value of 4.38 for this constraint indicates that profit will increase by \$4.38 for each additional hour of CutDry time that can be obtained. This Dual Value is unchanged provided that the number of CutDye hours (which is currently 630) lies anywhere between 495.6 and 682.4 hours. The Sew constraint is non-binding since 120 of the 600 available hours are not being used. The Dual Value remains 0 if the number of available sewing hours is above 479.99 hours. Similar comments can be made for the Finishing and Inspect/Pack constraints.

<u>Solution list</u>: This option shows the optimal value of the solution and the optimized value of all variables for this solution. For this case the optimal profit (Z) is \$7668.12 when the decision variables S and D and the slack variables for the four $\leq$ constraints have the listed values. There are six variables and four constraints (four independent equations). Therefore, two of the variables (slack 1 and slack 3) are set to zero and regarded as NONBasic as discussed in Chapter 5 of the Textbook. Since NONBasic variables are always zero, they do not contribute to the value of the solution.

<u>Iterations</u>: This option shows the initial simplex tableau and the successive simplex tableaus (iterations) that are required to reach the optimal solution as discussed in Chapter 5 of the textbook. These tableaus are very similar to the tableaus used in the textbook, but they do not have a cB column. This makes it a little more difficult to manually determine the values in the zj row.

QM for Windows (continued)

- The initial simplex tableau is labeled the Iteration 1 tableau. All of the decision variables are always set to zero in this tableau. Only the four slack variables are basic with non-negative values (must be ≥ 0). Since the coefficients of the slack variables are zero in the objective function, the value of this initial feasible solution is zero. The (cj-zj) net evaluation row values indicate that decision variable S will enter the solution to form the Iteration 2 tableau. Each unit increment of S will increase the value of the objective function by 10, the largest positive value in the (cj-zj) row.

- The Iteration 2 tableau shows that the decision variable S will have a value of 708. The solution will have a value of 7080 since the coefficient of S in the objective function is 10. The decision variable D is NONBasic with a value of zero. However, the (cj-zj) row values indicate that decision variable D will enter the basis to form the Iteration 3 tableau. Each unit of D that can be brought into the basis will increase the value of the objective function by 2.334.

- The Iteration 3 tableau is the optimized solution since all values in the (cj-zj) row are less than or equal to zero. This indicates that none of the NONBasic variables can be brought into the basis to improve the solution. This tableau shows that the optimal profit is about $7688.00 if 540 standard bags and 252 deluxe bags are produced.

Graph: This option is available only when the linear programming model has two decision variables. The graphical solution shows the constraints, the optimized objective function (the isoprofit line), the feasible region and the optimum values of decision variables S and C at the solution point. The optimal solution (the maximum possible profit) is determined by substituting the optimized values of S and D in the objective function.

SAVING THE PROBLEM: Save the **Par Demonstration Example** by accessing the **Save File/Problem** screen via the **File** menu, **Save As** option. Then enter the desired file name (e.g. ParDemoEx.lin) where .lin is the file extension for problems solved by the **QM for Windows** linear programming module.

QM for Windows (continued)

Solution of Goal Programming Problems (Chapter 15 of Textbook)

A goal-programming problem includes one or more linear goals. A goal may or may not be achieved as indicated by the value of a deviation variable. At times goal programming problems also contain constraints that must be satisfied to obtain a feasible solution. Therefore, if a problem contains both goals and constraints, the constraints must be satisfied before considering the goals.

The objectives of a goal-programming problem are to:

- Satisfy all constraints (if the problem has constraints).
- Achieve, or come as close as possible to achieving, all priority 1 goals (P1 goals), then all priority 2 goals (P2 goals), etc.
- Never compromise the achievement of a higher priority goal to better satisfy a lower priority goal since all priorities are preemptive.

Demonstration Example: Consider the Hub Properties, U.S. Oil problem in Chapter 15 of the textbook. Goal programming methodology will be implemented using the **QM for Windows** Linear Programming module (**not** the Goal Programming module).

A broker wants to determine the best possible investment allocation of \$80,000.00 (the available funds constraint), where the first priority (P1) goal is to limit investment risk and the P2 goal is to meet or exceed a specific investment return. For simplicity, the portfolio is limited to the following two stocks:

Stock	Price/Share	Est. Return/Share	Risk Index/Share
U.S. Oil	\$25.00	\$3.00	0.50
Hub Properties	\$50.00	\$5.00	0.25

The primary (first priority P1) goal is to incur a maximum risk index of 700. The second priority P2 goal is to obtain an annual return of at least \$9,000.00. If U = the number of shares of U.S. Oil to be purchased and H = the number of shares of Hub Properties to be purchased, the constraint and goals may be written:

$$25U + 50H \leq 80{,}000 \quad \text{(Available funds constraint)}$$

$$0.50U + 0.25H + d1m - d1p = 700 \quad \text{(Goal 1, the P1 Risk goal)}$$

$$3U + 5H + d2m - d2p = 9{,}000 \quad \text{(Goal 2, the P2 Return goal)}$$

$$U, H, d1m, d1p, d2m, d2p \geq 0 \quad \text{(non-negativity constraint)}$$

QM for Windows (continued)

The value of deviation variable d1m is the amount that 0.50U+0.25H is below the P1 goal of 700. The variable d1p is the amount that 0.50U+0.25H is above this P1 goal. Therefore, if one of these deviation variables is positive the other must be zero. Both deviation variables will have a value of zero if goal 1 is exactly achieved.

Note: Deviation variable identification labels correspond to the associated goal number, not the goal priority. For example, the deviation variables for a third goal (goal 3) would be d3m and d3p regardless of the priority of this goal. Also, there is no relationship between the goal number and the goal priority. For example, goal 2 could be a P1 goal and goal 1 could be a P2 goal.

The goal programming model objective function for this problem is written:

Min P1(d1p)+P2(d2m)

This notation clearly shows that the priority one (P1) goal is to be at or below a risk maximum level (risk target) and the P2 goal to be at or above a minimum return (return target).

The $80,000.00 available funds constraint must be satisfied as the broker attempts to determine how much of each stock should be purchased to best achieve the two prioritized goals. Since the first priority goal is to assume no more than 700 units of risk, the goal programming model attempts to make d1p (the deviation above 700) as small as possible, preferably zero. After the goal programming model meets (or comes as close as possible to meeting) the P1 goal to limit risk, a constraint is added to the model to assure that the results achieved for the P1 goal are not compromised when the broker attempts to achieve the P2 goal.

The **QM for Windows** Linear Programming module is used to optimize achievement of the P1 goal solution. Then, the model is modified and the Linear Programming module is used a second time to optimize achievement of the P2 goal as follows:

P1 Goal Solution using Linear Programming

- Select (double-click) the **Business Packages** window, **QM for Windows 2** icon to access the **QM for Windows** screen.
- Select the **Module** menu, **Linear Programming** option. Then select the **File** menu, **New** option to access the **Create data set for Linear Programming** screen. Use this screen to enter the initial problem information as follows:
  - Title: Enter: **U.S. Oil-Hub Properties (Priority 1 Goal Model)**

QM for Windows (continued)

- Number of Constraints: Enter **3**. There is one hard constraint and two goals (soft constraints). The **QM for Windows** Linear Programming module does not distinguish between goals and constraints. Therefore, the number of constraints is equal to the number of goals plus the number of constraints. The decision and deviation variable non-negativity constraint, which is "understood" by the software, is **not** entered or counted.
- Number of Variables: Enter **6.** There are two decision variables - one for the number of standard golf bags to be made and one for the number of deluxe golf bags to be made. The **QM for Windows** Linear Programming module does not distinguish between decision variables and deviation variables. Since there are also four deviation variables (two for each of the two goals) the total number of variables is 6.
- Objective: Select **Minimize** since we **always** want to meet, or minimize the deviation from, our goals.
- Row names tab: Select **Constraint 1, Constraint 2**, ...
- Column names tab: Select **X1, X2, X3**, ...
- After the above entries and selections are made, click **OK** to access the **[Data Table]** screen.

Change the default decision variable names and the constraint and goal names on the data entry table to more clearly represent this investment problem as shown in Figure 5. Then enter the model parameter values, the constraint/goal relationships and the objective function to complete the priority 1 model.

The objective of the P1 goal model is to minimize d1p [not to minimize P1(d1p)]. If there were more than one P1 goal, the objective would be to minimize the sum of the deviations from these goals. For example, if two P1 goals were to minimize the deviation above goal 1 and the deviation above goal 2, the objective function for the P1 goal model would be Min P1(d1p)+P1(d2p). This would be entered **without** the priority notation as Min d1p+d2p.

Click the **Solve** button at the top of the **[Data Table]** screen to obtain the **Linear Programming Results** screen for the P1 goal solution. Print the P1 goal solution shown in Figure 5 by accessing the **Print Setup** screen **Information** tab via the **File** menu, **Print** option. Select the **Information to print, Results** and **Solution list** options. Then select the **Constraint style, Equations** option and click **Print.**

QM for Windows (continued)

The optimum solution value of Min d1p = 0 in Figure 5 indicates that risk is not above 700 units if 1600 shares of Hub Properties are purchased and no US Oil is purchased. Therefore, the P1 goal of no more than 700 units of portfolio risk has been achieved.

Note: Artificial variable 2 and artificial variable 3 appear in the **Solution list** in Figure 5 with a value of 0. Artificial variables are required for the initial simplex iterations when a linear programming problem contains equality (=) or greater-than-or-equal to ($\geq$) constraints. However, the values of all artificial variables are driven to zero in the optimal solution of a feasible problem. Additional details about artificial variables are beyond the scope of this course.

Continue to the second priority (P2) solution, or save the P1 goal model by accessing the **Save File/Problem** screen via the **File** menu, **Save As** option. Then enter the desired file name (e.g. USHub1.lin).

P2 Goal Solution using Linear Programming

The P1 model can be modified immediately to form the P2 model by clicking the **Edit** button at the top of the **Linear Programming Results** screen to obtain the **[Data Table]** screen. Then edit the P1 goal model to form the P2 goal model as follows:

- Change the title of the linear programming model to: **U.S. Oil-Hub Properties (Priority 2 Goal Model)** via the **Title** icon at the top of the screen.
- Change the objective function to Min d2m since the priority 2 (P2) goal is to obtain a minimum investment return of $9,000.00.
- With the position cursor in the last row of the model, add a constraint via the **Edit** menu, **Insert Row** option. Label this constraint **P1 Goal Achievement**. This constraint, which is d1p = 0 for this example, maintains the results that were achieved for the priority 1 goal.

Note: If the P1 goal had not been achieved, the P1 goal achievement constraint would be written to ensure that the amount of this miss would not be increased. For example, if d1p = 15 for the P1 solution, the P1 goal achievement constraint added to the P2 goal model would be d1p = 15.

Click the **Solve** button at the top of the screen to obtain the P2 goal **Linear Programming Results** screen. Print the solution shown in Figure 6 by accessing the **Print Setup** screen **Information** tab via the **File** menu, **Print** option. Select the **Information to print, Results** and **Solution list** options. Then select the **Constraint Style, Equations** option and click **Print.**

The optimal P2 solution, where Z = d2m = 600, indicates that the P2 goal of a minimum return of $9,000.00 has been missed by $600.00. Therefore, 800 shares of U.S. Oil and 1200 shares of Hub Properties should be purchased to stay within the funding constraint and to best satisfy the two prioritized goals.

Solution of Transportation Problems

The transportation problem is a simplified version of the linear programming model since all of the coefficients on the left hand side of the constraints are equal to one. Also, each decision variable appears in only one supply constraint and in only one demand constraint. This allows the use of network modeling and a less complicated simplex tableau.

Demonstration Example: Consider the Foster Generators problem in Chapter 7 of the textbook. The inputs for this problem are as follows.

	Unit Shipping Cost				
	Boston	**Chicago**	**St. Louis**	**Lexington**	**Supply**
Cleveland	3	2	7	6	5000
Bedford	7	5	2	3	6000
York	2	5	4	5	2500
Demand	6000	4000	2000	1500	

This transportation problem can be modeled in **QM for Windows** and solved as follows:

- Select (double-click) the **Business Packages** window, **QM for Windows 2** icon to access the **QM for Windows** screen.
- Select the **Module** menu, **Transportation** option. Then select the **File** menu, **New** option to access the **Create data set for Transportation** screen. Use this screen to enter the initial problem information as follows:
  - Title: Enter **Foster Generator Transportation Problem**
  - Number of Sources: Enter **3**. There are three sources of supply – Cleveland, Bedford and York.
  - Number of Destinations: Enter **4.** There four destinations – Boston, Chicago, St. Louis and Lexington.
  - Objective: Select **Minimize** to minimize the total cost of shipping generators from the three sources to the four destinations.
  - Row names: Select **Source 1, Source 2, ...**
  - Column names: Select **Destination 1, Destination 2, Destination 3, ...**
  - After the above entries and selections are made, click **OK** to access the **[Data Table]** screen.

QM for Windows (continued)

Starting Method: Select **Minimum Cost Method** option from the drop down menu on the **[Data Table]** screen. The minimum cost approach is also known as the intuitive approach. It will also allocate on the basis of maximum profit, sales, etc. for maximization problems.

Change (edit) the default source and destination names in the data table to more clearly represent the Foster Generator problem as shown in Figure 7. Then enter the model shipping cost coefficients (the unit cost of shipping a generator from each source to each destination), the supply constraints (the number of generators available at each source) and the demand constraints (the number of generators required by each destination/user).

Click the **Solve** button at the top of the **[Data Table]** screen to obtain the **Transportation Shipments** screen that shows the value of the minimum cost solution and the number of generators shipped from each source to each destination for this optimal solution.

Print the solution shown in Figure 7 by accessing the **Print Setup** screen **Information** tab via the **File** menu, **Print** option. Then select the **Data and Results, Final Solution Table, Iterations** and **Shipping list** options and click **Print.**

The **Iteration 1** tableau in Figure 7 is the initial allocation of traffic based upon the minimum cost method. The bracketed numbers indicate the unit cost of placing traffic in each unoccupied cell. Notice that placing traffic in the Bedford to Chicago route can reduce the total cost. This is done as shown in the **Iteration 2** tableau. This is the final, minimum cost tableau since the allocation of traffic to any of the unoccupied cells will increase cost.

Solution of PERT/CPM Problems

The PERT-CPM module solves project-scheduling problems using the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT). The features and capabilities of this module used n this course are demonstrated in the following example.

Demonstration Example: Consider the following meeting planning example.

Task	Description	Immediate Predecessors	Optimistic Time	M_Likely Time	Pessimistic Time
A	Plan Topic		1.5	2.0	2.5
B	Obtain Speakers	A	2.0	2.5	6.0
C	List Mtg. Locations		1.0	2.0	3.0
D	Select Location	C	1.5	2.0	2.5
E	Speaker Travel Plans	B, D	0.5	1.0	1.5
F	Final Check	E	1.0	2.0	3.0
G	Prepare/Mail Brochures	B, D	3.0	3.5	7.0
H	Take Reservations	G	3.0	4.0	5.0
I	Last Minute Details	F, H	1.5	2.0	2.5

This project scheduling problem with uncertain activity (task) time estimates can be modeled in **QM for Windows** and solved as follows:

- Select (double-click) the **Business Packages** window, **QM for Windows 2** icon to access the **QM for Windows** screen.
- Select the **Module** menu, **Project Management (PERT/CPM)** option. Then select the **File** menu, **New** option. A drop-down menu will appear. Select the **Triple time estimate** option from this menu since we are dealing with probabilistic time estimates. This will access the **Create data set for Project Management (PERT/CPM)** screen.

Note: For most business problems, a single time is estimated for each task. For this more common case the Single time estimate option is selected to access the **Create data set for Project Management (PERT/CPM)** screen.

QM for Windows(continued)

- Enter the initial problem information in the **Create data set for Project Management (PERT/CPM)** screen as follows:
  - Title: Enter **Project Management (PERT/CPM) Example**
  - Number of Tasks: Enter **9** using the scroll bar.
  - Table Structure: Select **Precedence list**.
  - Row name options: Select **A, B, C, ...**

After the above entries and selections are made, click **OK** to access the **[Data Table]** screen.

Enter the three time estimates and the task precedence relationships for each task.
For Task A, leave the precedence relationship cells empty. For Task B, enter A in the **Prec 1** cell. For Task E, enter B in the **Prec** 1 cell and enter D in the **Prec 2** cell, etc.

Note: The precedence relationship cells must be left blank for tasks that do not depend on the completion of other tasks (tasks that can be started immediately).

Click the **Solve** button at the top of the [**Data Table**] screen to obtain the **Project Management (PERT/CMP) Results** screen. This screen shows the project Activity time (the expected time to complete the project), the Activity time for each task and the standard deviation of these Activity times. This screen also shows the ES, LS, EF, LF and Slack times for each task.

Print the solution shown in Figure 8 by accessing the **Print Setup** screen **Information** tab via the **File** menu, **Print** option. Then select the **Information to Print, Results, Task time computations** and **Charts** options and click **Print**. The Gantt chart (Early times) and the Gantt chart (Late times) printouts that are generated have been discarded since this information is contained in the Gantt chart (Early and Late times) printout in Fig. 8.

The following probability of completion information can be obtained for this uncertain activity time problem since three time estimates have been used for each activity (task). Access the **Normal Distribution Calculator** screen from the **Project Management (PERT/CMP) Results** screen via the **Tools** menu, **Normal** option.

Probability of Completion Within a Specific Period of Time: The probability that the Casey Meeting planning will be completed in 15 weeks is 50 percent. The higher probability that the planning will be completed in 16 weeks can be determined by entering the following information in the **Normal Distribution Calculator** screen:

- Compute: Select **Probability given value(s)**
- Parameters: Do not change the mean and standard deviation values shown for the project completion date.
- Tail: Select **One-tailed** since we are interested in the area under the normal distribution curve to the left of the completion time.
- Value/Cutoff: Enter **16**

After the above selections and entries are made, click **Compute** to obtain a display of the results. Then click **Print** and **Yes** to print the graph shown in Figure 9. Notice that there is an 83.4 percent probability that the meeting planning process will be completed in 16 weeks.

Time Required for a Specific Probability of Completion: The time required for a 99 percent probability that the meeting planning process will be completed can be determined by entering the following information in the **Normal Distribution Calculation** screen:

- Compute: Select **Value(s) given probability**
- Parameters: Do not change the mean and standard deviation values shown for the project completion date.
- Tail: Select **One tailed.** We are interested in the area under the normal distribution curve to the left of the completion time. However, this area is specified indirectly by specifying the area under the normal distribution curve to the right of the completion time.
- Probability to RIGHT of the tail: Enter or select: 1% or **0.01**

After the above selections and entries are made, click **Compute** to obtain a display of the results. Then click **Print** and **Yes** to print the graph shown in Figure 10. Notice that it is 99 percent certain that the project will be completed in 17.39 weeks since 99 percent of the area under the normal curve is to the left of this completion date.

Note: Sometimes the **QM for Windows** software prints an **incorrect** probability number on the normal distribution plots shown in Figures 9 and 10. However, the probabilities printed below the graph are always correct.

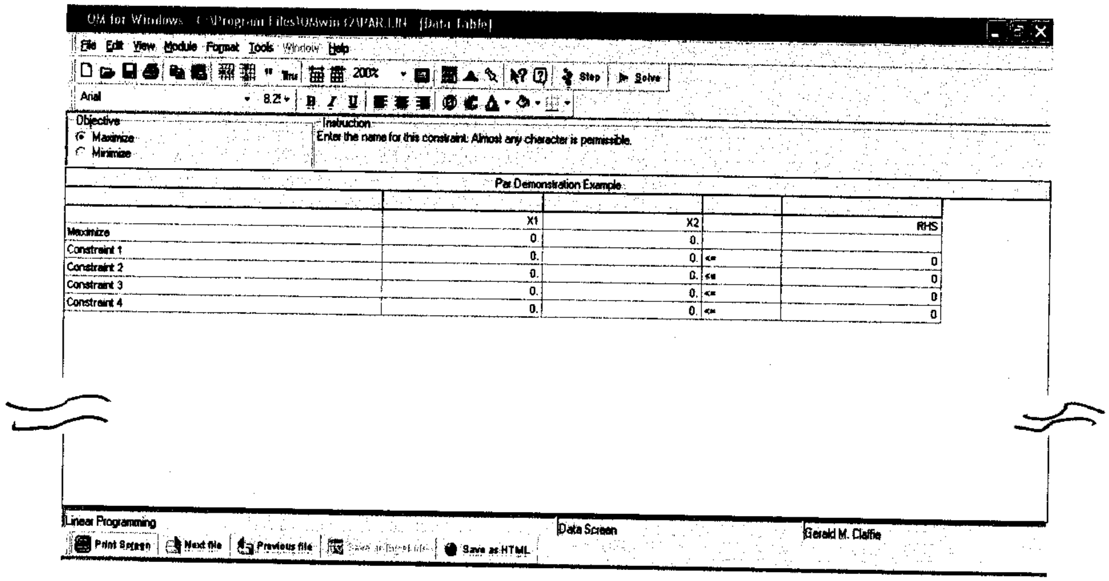


FIG-1.

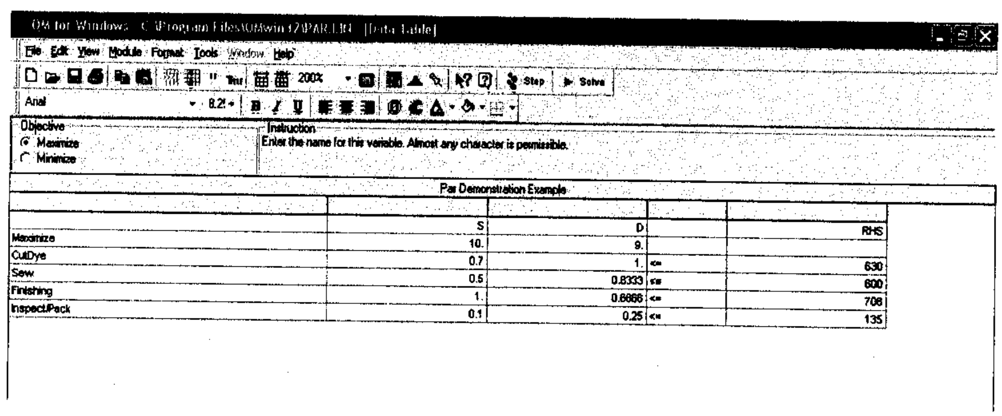


FIG-2.

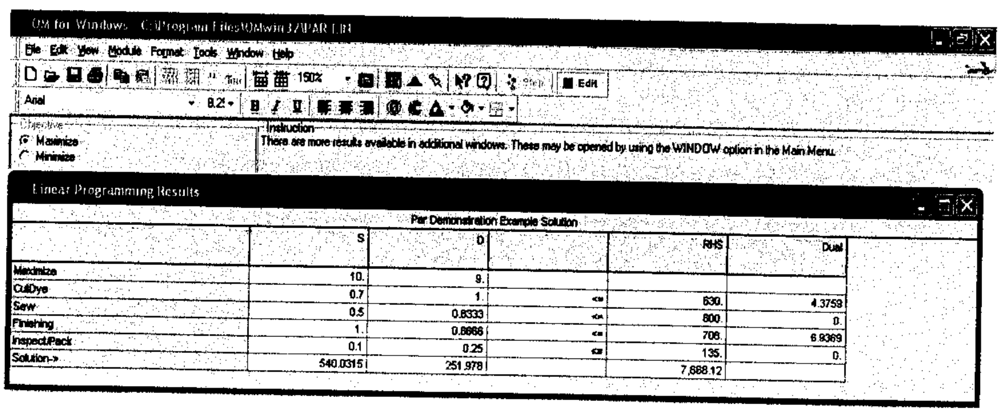


FIG-3.

Module/submodule: Linear Programming
Problem title: Par Demonstration Example
Objective: Maximize

FIG-4.

Results ----------

OPTIMIZE: 10S + 9D
CutDye:+ .7S + 1D <= 630
Sew:+ .5S + .8333D <= 600
Finishing:+ 1S + .6666D <= 708
Inspect/Pack:+ .1S + .25D <= 135

Ranging ----------

RANGE OF OPTIMALITY

RANGE OF COEFFICIENTS C_1 & C_2

Variable	Value	Reduced Cost	Original Value	Lower Bound	Upper Bound
S	540.0315	0	10	6.3	13.5014
D	251.978	0	9	6.666	14.2857

Constraint	Dual Value	Slack/ Surplus	Original Value	Lower Bound	Upper Bound
CutDye	4.3759	0	630	495.6	682.3732
Sew	0	120.011	600	479.989	Infinity
Finishing	6.9369	0	708	579.972	900
Inspect/Pack	0	18.0024	135	116.9976	Infinity

RANGE OF CONSTRAINT RHS VALUES (LIMITS)

RANGE OF FEASIBILITY

Solution list ----------

Variable	Status	Value
S	Basic	540.0315
D	Basic	251.978
slack 1	NONBasic	0
slack 2	Basic	120.011
slack 3	NONBasic	0
slack 4	Basic	18.0024
Z	Optimal	7,668.117

Iterations ----------

Iteration 1

MISSING C_B COLUMN

Cj--> Basic		10 S	9 D	0 slack 1	0 slack 2	0 slack 3	0 slack 4	Quantity
slack 1	0	0.7	1	1	0	0	0	630
slack 2	0	0.5	0.8333	0	1	0	0	600
slack 3	0	1	0.6666	0	0	1	0	708
slack 4	0	0.1	0.25	0	0	0	1	135
zj		0	0	0	0	0	0	0
cj-zj		10	9	0	0	0	0	

Iteration 2

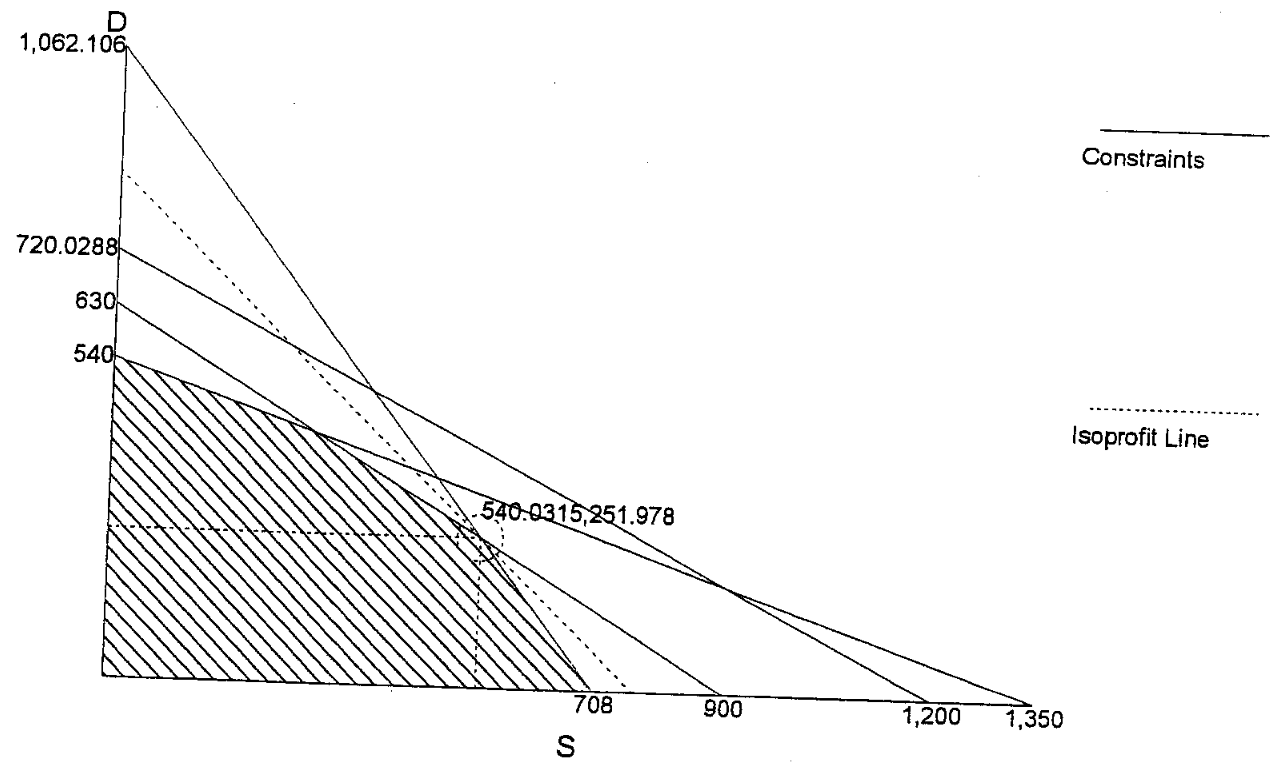
Cj--> Basic	C_B	10 S	9 D	0 slack 1	0 slack 2	0 slack 3	0 slack 4	Quantity
slack 1	0	0	0.5334	1	0	-0.7	0	134.4
slack 2	0	0	0.5	0	1	-0.5	0	246
S	10	1	0.6666	0	0	1	0	708
slack 4	0	0	0.1833	0	0	-0.1	1	64.2
zj		10	6.666	0	0	10	0	7,080
cj-zj		0	2.334	0	0	-10	0	

FIG-4 (CONTINUED)

Iteration 3

Cj--> Basic	C_B	10 S	9 D	0 slack 1	0 slack 2	0 slack 3	0 slack 4	Quantity
D	9	0	1	1.8748	0	-1.3124	0	251.978
slack 2	0	0	0	-0.9374	1	0.1562	0	120.011
S	10	1	0	-1.2498	0	1.8748	0	540.0315
slack 4	0	0	0	-0.3437	0	0.1406	1	18.0024
zj		10	9	4.3759	0	6.9369	0	7,668.1166
cj-zj		0	0	-4.3759	0	-6.9369	0	

Par Demonstration Example



Module/submodule: Linear Programming
Problem title: U.S. Oil-Hu Properties (Priority 1 Goal Model)
Objective: Minimize

Results ----------

Multiple optimal solutions exist

OPTIMIZE: 1d1p
Available Funds:+ 25U + 50H <= 80000
Risk Goal (G1:P1):+ .5U + .25H + 1d1m - 1d1p = 700
Return Goal (G2:P2):+ 3U + 5H + 1d2m - 1d2p = 9000

Solution list ----------

Variable	Status	Value
U	NONBasic	0
H	Basic	1,600
d1m	Basic	300
d1p	NONBasic	0
d2m	Basic	1,000
d2p	NONBasic	0
slack 1	NONBasic	0
artfcl 2	NONBasic	0
artfcl 3	NONBasic	0
Z	Optimal	0

FIG-5

Module/submodule: Linear Programming
Problem title: U.S. Oil-Hu Properties (Priority 2 Goal Model)
Objective: Minimize

Results ----------

OPTIMIZE: 1d2m
Available Funds:+ 25U + 50H <= 80000
Risk Goal (G1:P1):+ .5U + .25H + 1d1m - 1d1p = 700
Return Goal (G2:P2):+ 3U + 5H + 1d2m - 1d2p = 9000
P1 Goal Achievement:+ 1d1p = 0

Solution list ----------

Variable	Status	Value
U	Basic	800
H	Basic	1,200
d1m	NONBasic	0
d1p	Basic	0
d2m	Basic	600
d2p	NONBasic	0
slack 1	NONBasic	0
artfcl 2	NONBasic	0
artfcl 3	NONBasic	0
artfcl 4	NONBasic	0
Z	Optimal	600

FIG-6

Module/submodule: Transportation
Problem title: Foster Generator Transportation Problem
Starting method: Minimum Cost Method
Objective: Minimize

FIG-7.

Data and Results ----------

Original Data

	Boston	Chicago	St. Louis	Lexington	SUPPLY
Cleveland	3	2	7	6	5,000
Bedford	7	5	2	3	6,000
York	2	5	4	5	2,500
DEMAND	6,000	4,000	2,000	1,500	

Shipments

	Boston	**Chicago**	**St. Louis**	**Lexington**
Cleveland	**3,500**	**1,500**		
Bedford		**2,500**	**2,000**	**1,500**
York	**2,500**			

Total cost = 39,500

Final Solution Table ----------

	Boston	Chicago	St. Louis	Lexington
Cleveland	**3500**	**1500**	**(8)**	**(6)**
Bedford	**(1)**	**2500**	**2000**	**1500**
York	**2500**	**(4)**	**(6)**	**(6)**

Iterations ----------

Iteration 1

	Boston	Chicago	St. Louis	Lexington
Cleveland	**1,000**	**4,000**	**(9)**	**(7)**
Bedford	**2,500**	**(-1)**	**2,000**	**1,500**
York	**2,500**	**(4)**	**(7)**	**(7)**

Iteration 2

	Boston	Chicago	St. Louis	Lexington
Cleveland	**3,500**	**1,500**	**(8)**	**(6)**
Bedford	**(1)**	**2,500**	**2,000**	**1,500**
York	**2,500**	**(4)**	**(6)**	**(6)**

Shipping list ----------

From	To	Units	$/Unit	Total Cost
Cleveland	Boston	3,500	3	10,500
Cleveland	Chicago	1,500	2	3,000
Bedford	Chicago	2,500	5	12,500
Bedford	St. Louis	2,000	2	4,000
Bedford	Lexington	1,500	3	4,500
York	Boston	2,500	2	5,000

Total shipping cost =				39500

FIG-8.

Module/submodule: Project Management (PERT/CPM)/Triple time estimate
Problem title: Project Management (PERT/CPM) Example
Method: Triple time estimate
Network type: Precedence list

Results ----------

Task	Precedences
A	
B	A
C	
D	C
E	B, D
F	E
G	B, D
H	G
I	F, H

Project completion time = 15
Project standard deviation = 1.02740234672036

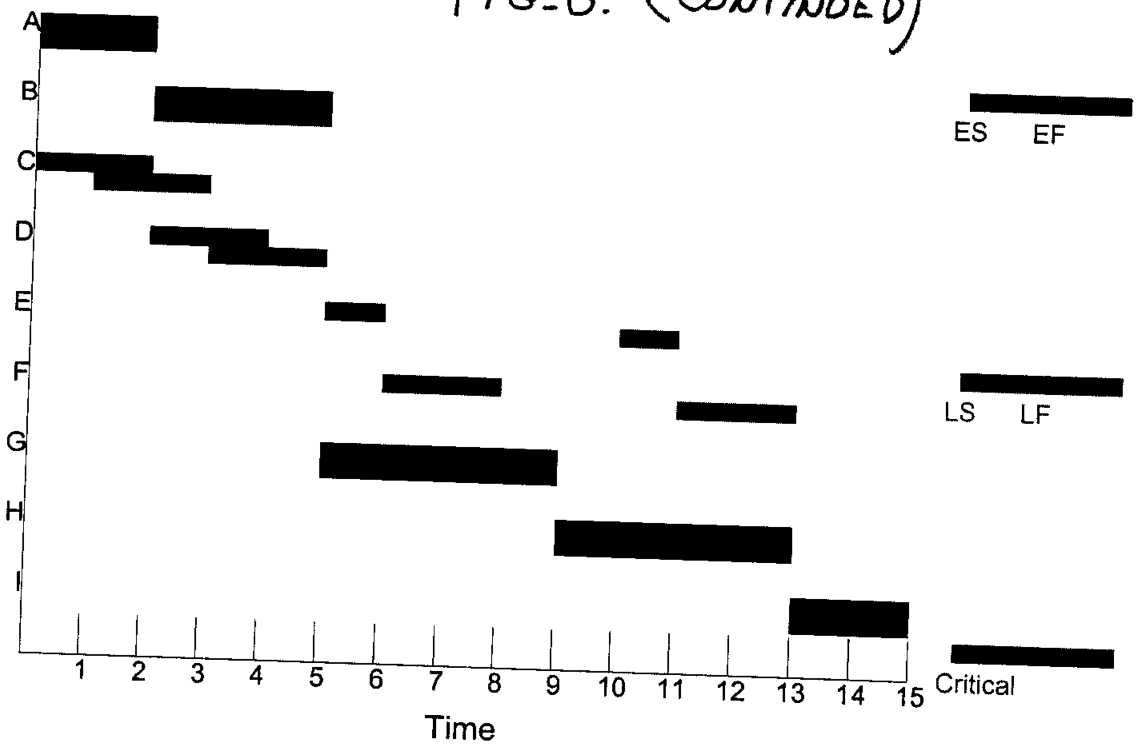
	Activity time	Early Start	Early Finish	Late Start	Late Finish	Slack	Standard Dev
A	2	0	2	0	2	0	0.1667
B	3	2	5	2	5	0	0.6667
C	2	0	2	1	3	1	0.3333
D	2	2	4	3	5	1	0.1667
E	1	5	6	10	11	5	0.1667
F	2	6	8	11	13	5	0.3333
G	4	5	9	5	9	0	0.6667
H	4	9	13	9	13	0	0.3333
I	2	13	15	13	15	0	0.1667

Task time computations ----------

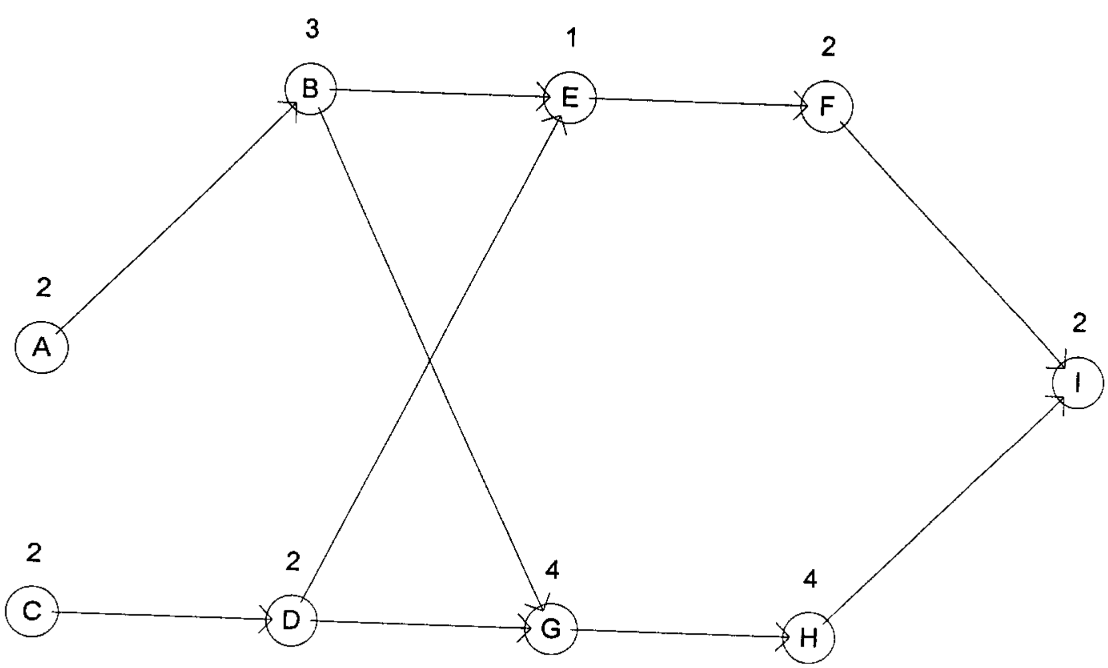
Activity	Optimistic time	Most Likely time	Pessimistic time	Activity time	Standard deviation	Variance
A	1.5	2	2.5	2	0.1667	0.0278
B	2	2.5	6	3	0.6667	0.4444
C	1	2	3	2	0.3333	0.1111
D	1.5	2	2.5	2	0.1667	0.0278
E	0.5	1	1.5	1	0.1667	0.0278
F	1	2	3	2	0.3333	0.1111
G	3	3.5	7	4	0.6667	0.4444
H	3	4	5	4	0.3333	0.1111
I	1.5	2	2.5	2	0.1667	0.0278

Project Management (PERT/CPM) Example
Gantt chart (Early and Late times)

FIG-B. (CONTINUED)



Project Management (PERT/CPM) Example
Precedence Graph



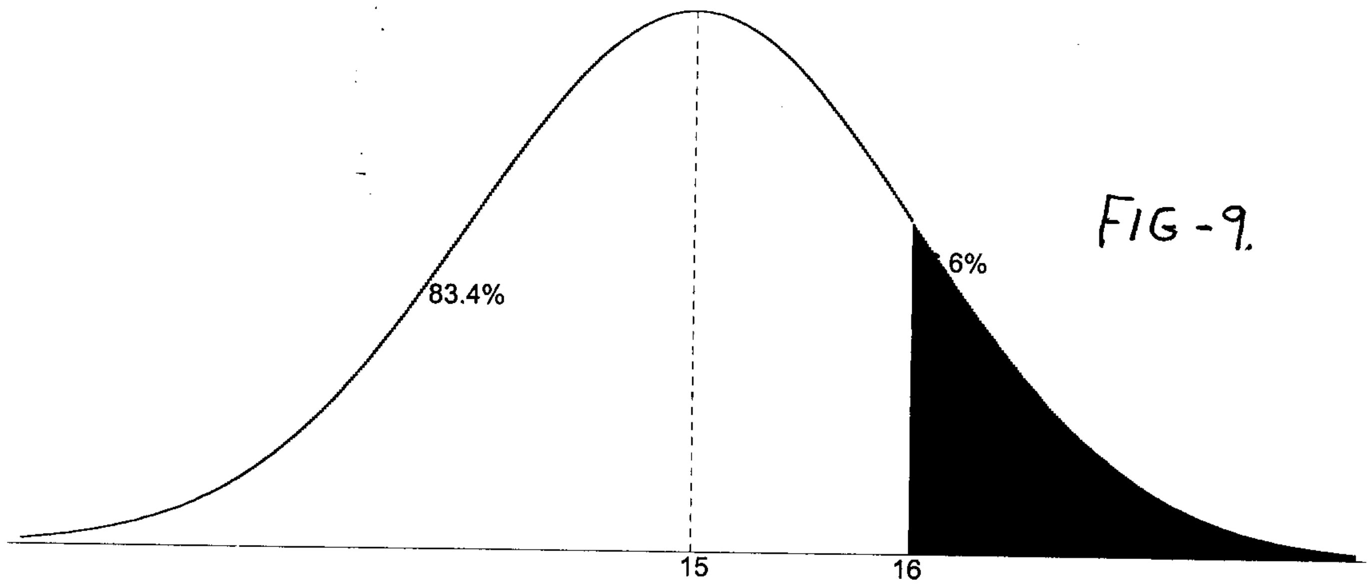


FIG-9.

Mean = 15
Standard deviation = 1.027402: (Variance = 1.055
One tailed
Cutoff = 16

---- Results ----

Probability left of the cutoff = 83.4%
Probability right of the cutoff = 16.6%

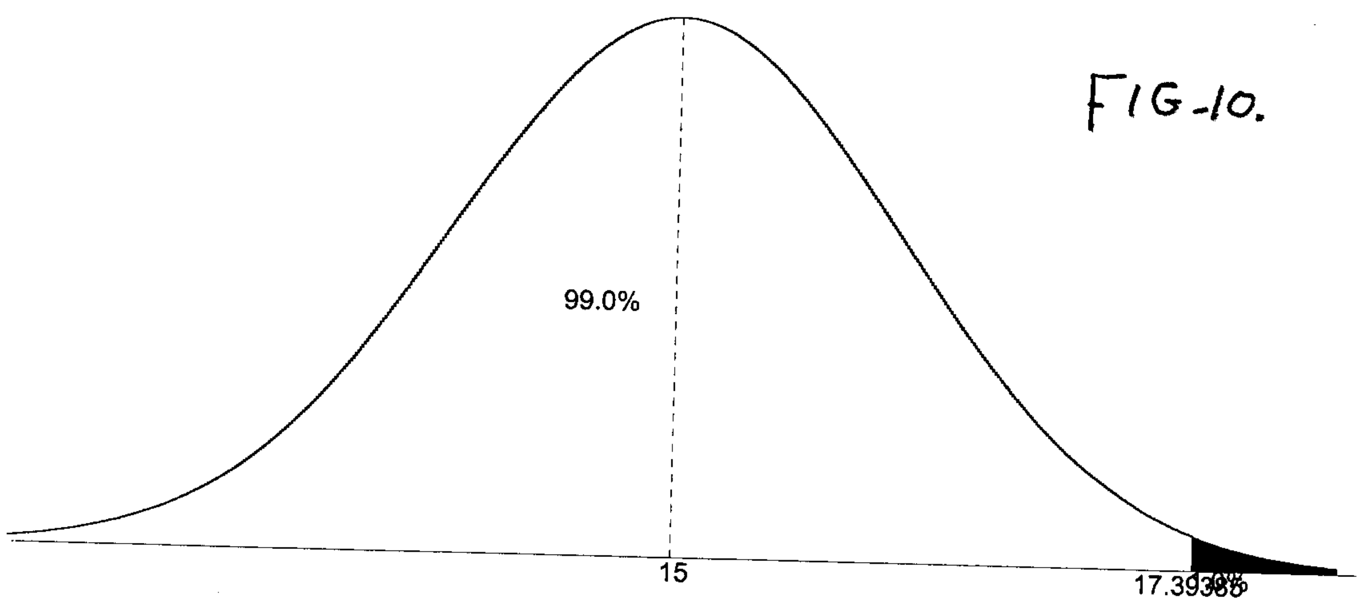


FIG-10.

Given

Mean = 15
Standard deviation = 1.027402: (Variance = 1.055554869604)
One tailed
Probability to the right = 0.01
(Probability to the left = .99)
---- Results ----

The cutoff is 17.39385