Optimization Modeling in Spreadsheets

with

What'sBest!

June 2013
Preliminary version, comments welcome.

Published by

LINDO SYSTEMS INC.
1415 North Dayton Street
Chicago, Illinois 60622
Technical Support: (312) 988-9421
http://www.lindo.com
e-mail: tech@lindo.com

Contents

1

Introduction to Optimization in Spreadsheets

1.1 Introduction

Spreadsheets, combined with the optimization capability of the Excel add-in What's*Best*!, can be used to conveniently solve a variety of optimization problems in business, industry, and government.

For most optimization problems, one can think of there being *two important classes of objects.* The first of these is *limited resources*, such as land, plant capacity, and sales force size. The second is *activities*, such as "produce low carbon steel," "produce stainless steel," and "produce high carbon steel." *Each activity consumes* or possibly *contributes* additional amounts of the *resources*. The problem is to determine the best combination of activity levels that does not use more resources than are actually available.

In the following chapters we will illustrate how What's*Best*! can be used to solve the typical kinds of optimization problems found in practice. Additional details about advanced usage of What's*Best!* can be found in the What's*Best!* users manual, see www.lindo.com. Some of the material used herein is based on the text, *Optimization Modeling with LINGO*. That text is concerned with the use of the general purpose modeling language, LINGO, for formulating and solving optimization problems.

1.2 Example Applications of Optimization

Optimization has been applied in a wide range of industries. Some important examples are listed below.

- Petroleum Blending:

Some of the earliest applications of optimization occurred in the 1960's in gasoline refining. Gasoline must satisfy several major quality requirements, mainly octane, but also volatility, and vapor pressure. Gasoline is actually a blend of ingredients. Which ingredients are available, and their prices, vary from month to month based on political events, etc. Additionally, the volatility and vapor pressure requirements vary by time of year. Higher volatility and vapor pressure is required in the winter time. Octane requirements vary by location (e.g., lower at higher altitudes.) Given the costs of various ingredients and quality requirements today, what is the lowest cost acceptable ingredient mix?

- Electrical Generator Unit Commitment:

Many electricity generation companies use optimization to decide which generators to run which hours of the day. Given forecasted electricity demand over next 24 hours, week, etc., and cost structure of each generator, which generators should be run in which intervals?

- Financial Portfolios:

How much to invest in which assets given expected returns, interactions/correlations among investments, such as at a telecommunications company or a mutual fund firm.

- Auction of Electricity Transmission Capacity in a U.S. state:

Maximize the value of awards, subject to not selling more capacity than is available. Interesting feature: a bidder may bid on a combination of lines, e.g., if in series. The prices, so-called dual prices, generated as part of the optimization are the clearing prices.

- Plant Configuration Under Uncertainty at an Automobile Manufacturer:

At one point in time when it was clear the auto manufacturer had too much capacity for the demand coming from a slow economy, optimization was used to decide which plants to close, which to re-focus, given various demand scenarios and their probabilities.

- Gas contract selection under uncertainty at a natural gas supply company:

Which gas contracts to buy when, how much gas to store, when to draw it out, in the face of uncertainty(represented by about a various scenarios of possible weather and spot prices).

- Cutting stock in steel and paper industries:

Cutting long cables to consumer lengths at a cable manufacturer, paper rolls at a paper company. Metal bars in steel industry. Given length (width) of master or jumbo, and amount needed of the smaller f.g., lengths (widths), what cutting patterns should be used?

- Supply Chain Redesign/DC Location at a Consumer Goods Manufacturer:

After acquiring another company and merging in several new product lines and distribution centers, which DC's should be closed? Which DC's should serve which customers?

- Production Scheduling at a Tire Manufacturer:

Given daily demand schedule and which combinations of tires can be produced together in which heaters, which tire combinations should be run in which heaters?

- Gas Pipeline capacity auction (Midwestern U.S.)

Given pipeline capacity requested over what interval of days, and amount bid, which bids should be awarded, so as to maximize sales revenue and not exceed daily pipeline capacity.

- Quality Improvement via Matching of Components(electronics manufacturer).

Certain devices, e.g., cell phone mikes, blades in a jet engine turbine, should be closely matched to improve performance/quality. Solved a "matching" IP to increase yield to about 75% from 60%.

- Staffing and Rostering of maintenance personnel at a cell phone company.

Regular labor at metal fabrication firm, crew scheduling at airlines, telephone call center.

- Multiperiod Production Planning and Blending at Food Processing Company:

Meet demands each month at locations around the country from sources around the country, taking into account the required quality levels(mainly acidity) at the demand points, and available quality at each supply point.

1.3 The ABC's of Optimization in What's*Best*!

We assume the reader is familiar with setting up a conventional, so-called "What If" spreadsheet models in Excel. Converting a What-If model into an optimization model to be solved by What's*Best*! consists of three steps:

A) Identify the **A**djustable cells, i.e., the decision variables.

B) Specify a criterion for measuring a **B**est solution, i.e., specify a cell to minimize or maximize.

C) Provide the **C**onstraints,
i.e., the relationships limiting what values can be placed in the adjustable cells.

The typical variables identified in step A are: *How much do we buy, produce, ship, carry in inventory; - from a specific vendor of a specific product in a specific period.*

The typical objective in step B is either to *Maximize wealth or minimize cost.*

Typical constraints in step C are: *sources of a commodity* $\leq$ *uses of a commodity,*

where commodity could be cash, labor, capacity, product, etc.

For ease of understanding, our initial examples will be small and simple, with perhaps a half dozen variables or constraints. Practical problems may have a hundred thousand variables and constraints.

1.4 Example: A Product Mix Problem

In a simple "product mix" problem, we want to decide upon what mix of products to produce, given our available resources. The Enginola Television Company produces two types of TV sets, the "Astro" and the "Cosmo". There are two production lines, one for each set. The Astro production line has a capacity of 60 sets per day, whereas the capacity for the Cosmo production line is only 50 sets per day. The labor requirements for the Astro set is 1 person-hour, whereas the Cosmo requires a full 2 person-hours of labor. Presently, there is a maximum of 120 man-hours of labor per day that can be assigned to production of the two types of sets. If the profit contributions are $20 and $30 for each Astro and Cosmo set, respectively, what should be the daily production?

A structured, but verbal, description of what we want to do is:

Maximize Profit contribution;

Subject to:

Units of Astro production is less than or equal to 60 units;

Units of Cosmo production is less than or equal to 50 units;

Labor hour usage by Astro and Cosmo production is less than or equal to 120 hours;

Until there is a significant improvement in artificial intelligence/expert system software, we will need to be more precise if we wish to get some help in solving our problem. We can be more precise if we define:

A = units of Astros to be produced per day,
C = units of Cosmos to be produced per day.

Further, we decide to measure:

Profit contribution in dollars,
Astro usage in units of Astros produced,
Cosmo usage in units of Cosmos produced, and
Labor in person-hours.

Then, a precise statement of our problem is:

$$
\begin{array}{lrl}
\text{Maximize} & 20A + 30C & \text{(Dollars)} \\
\text{subject to} & A \quad\quad \leq 60 & \text{(Astro capacity)} \\
 & C \leq 50 & \text{(Cosmo capacity)} \\
 & A + 2C \leq 120 & \text{(Labor in person-hours)}
\end{array}
$$

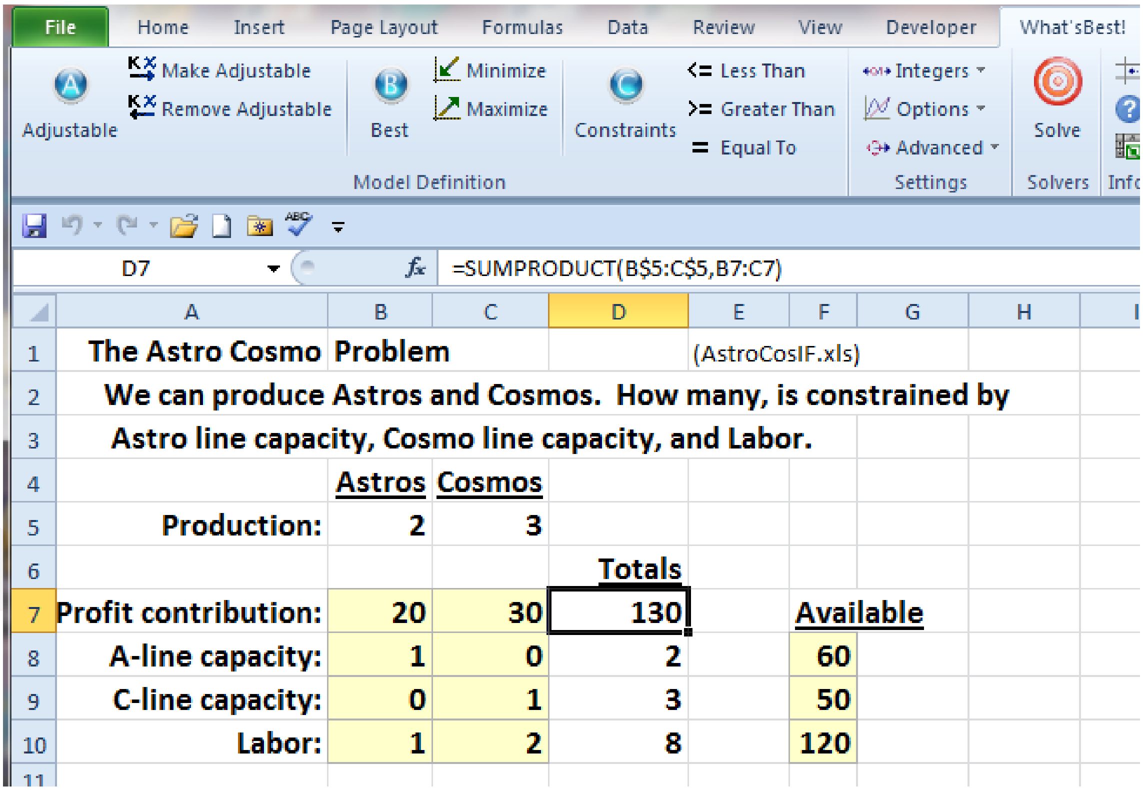
The first line, "Maximize 20A+30C", is known as the *objective function*. The remaining three lines are known as *constraints*. Most optimization programs, sometimes called "solvers", assume all variables are constrained to be nonnegative, so stating the constraints $A \geq 0$ and $C \geq 0$ is unnecessary. What's*Best*! would by default assume $A \geq 0$ and $C \geq 0$.

Using the terminology of resources and activities, there are three resources: Astro capacity, Cosmo capacity, and labor capacity. The activities are Astro and Cosmo production. It is generally true that, with each constraint in an optimization model, one can associate some resource. For each decision variable, there is frequently a corresponding physical activity.

1.5 What's*Best!* Spreadsheet Optimizer

Let us look at how we can solve optimization problems with What's*Best*!.We can set up the Astro/Cosmo problem in "What-if" form as in the figure below.

Figure 1.1 Enginola Problem in What-if Form



Notice that the cursor is located in cell D7. In the formula bar, notice that the formula in cell D7 is =SUMPRODUCT(B$5:C5,B7:C7). This is equivalent to D7 =B5*B7+C5*C7. Cells D8:D10 were filled with similar formulae by copying. The "$" sign in front of the 5 means that the 5 is not changed as the formula is copied. Similarly, the formula in cell D10 is =SUMPRODUCT(B$5:C5,B10:C10).

1.5.1 One-Click Formulation of What'sBest! Models

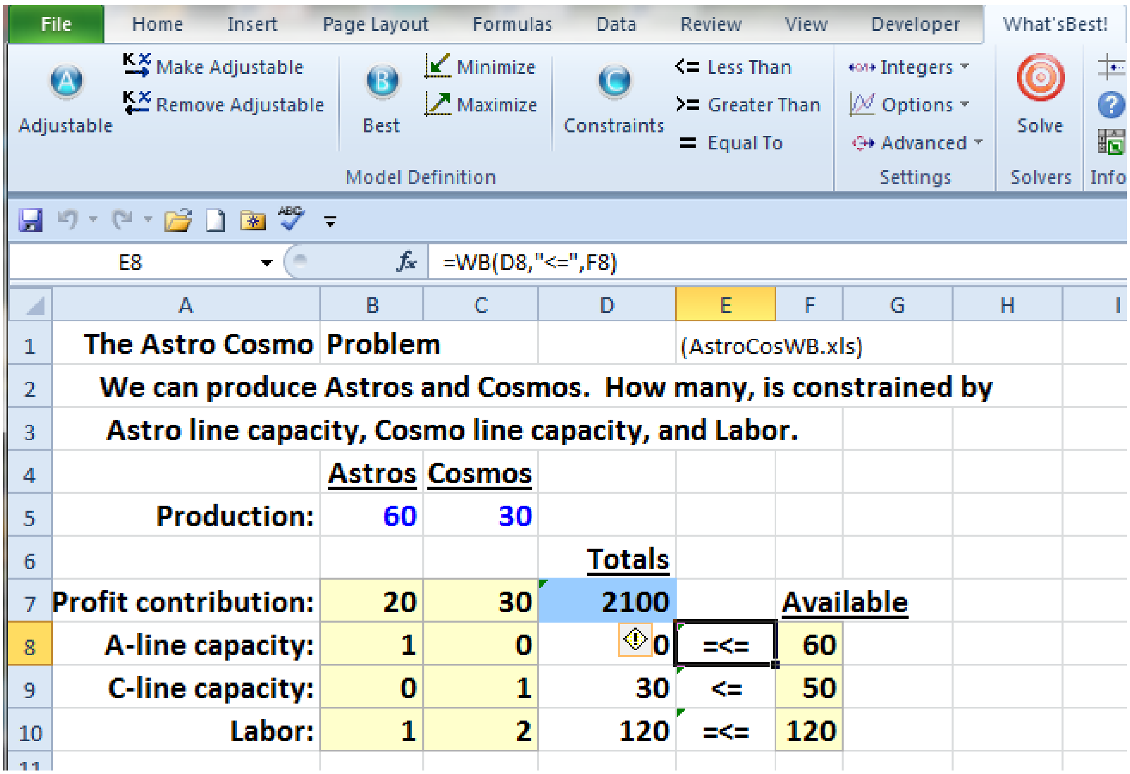
The above spreadsheet is set up in "what if" form. For appropriately formulated Excel models, each of the A, B, C, steps of What'sBest! can be done with one click.

Using the What'sBest! Tool bar in the upper left:

A) Mark B5:D5 as "Adjustable" (K⟶x)cells,

B) Mark D7 as the “Best” cell to be maximized,
C) Constraints are added by highlighting cells E8:E10, and then clicking on “<= Less Than”.
Optimize by clicking on the red bullseye.

Figure 1.2 Applying the ABC’s to the Enginola Problem



We see that the optimal solution is to produce 60 Astros and 30 Cosmos for a total profit contribution of 2100.

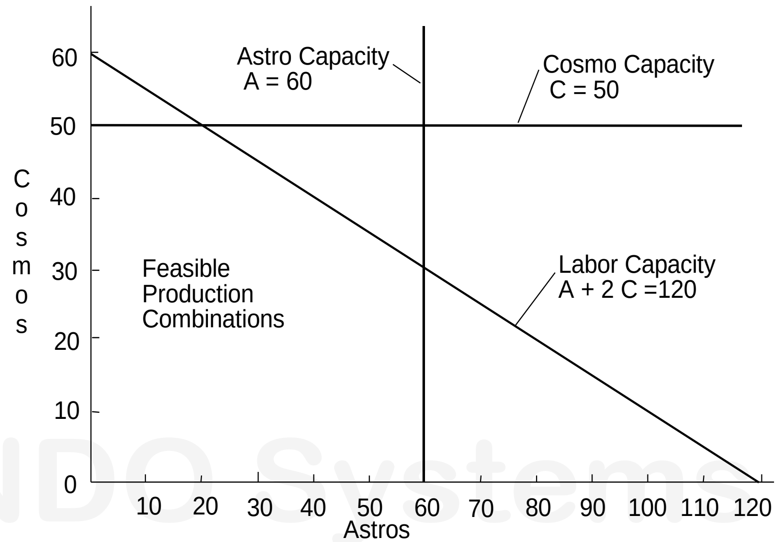
1.6 Graphical Analysis for Small Problems

An interesting exercise is to use our intuition to guess how much to produce of each of Astro and Cosmo. Some possibly useful observations are: Cosmo is more profitable per unit, however, Astro makes more $/hour of labor.

What do you think is the value of an additional hour of labor? Is it $20, $15, or $0?
The Astro/Cosmo problem is represented graphically in Figure 1.1. The feasible production combinations are the points in the lower left enclosed by the five solid lines. We want to find the point in the feasible region that gives the highest profit.

To gain some idea of where the maximum profit point lies, let's consider some possibilities. The point $A = C = 0$ is feasible, but it does not help us out much with respect to profits. If we spoke with the manager of the Cosmo line, the response might be: "The Cosmo is our more profitable product. Therefore, we should make as many of it as possible, namely 50, and be satisfied with the profit contribution of $30 \times 50 = \$1500$."

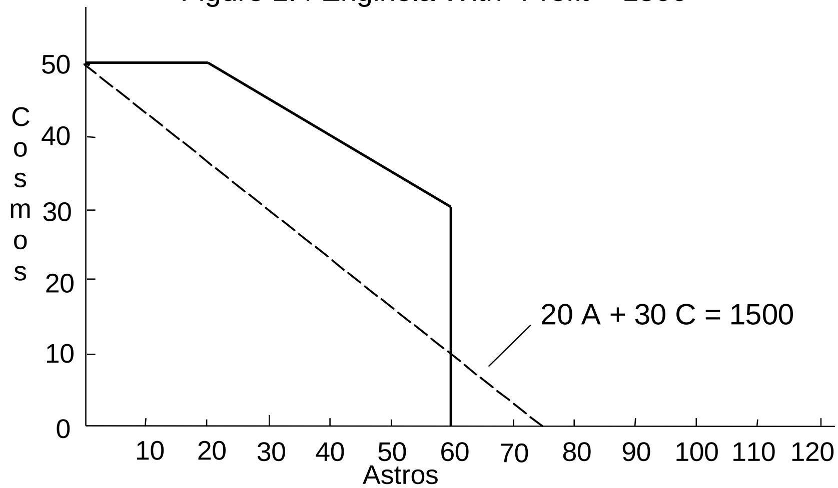
Figure 1.3 Feasible Region for Enginola



You might observe there are many combinations of A and C, other than just $A = 0$ and $C = 50$, that achieve \$1500 of profit. Indeed, if you plot the line $20A + 30C = 1500$ and add it to the graph, then you get Figure 1.2. Any point on the dotted line segment achieves a profit of \$1500. Any line of constant profit such as that is called an iso-profit line (or iso-cost in the case of a cost minimization problem).

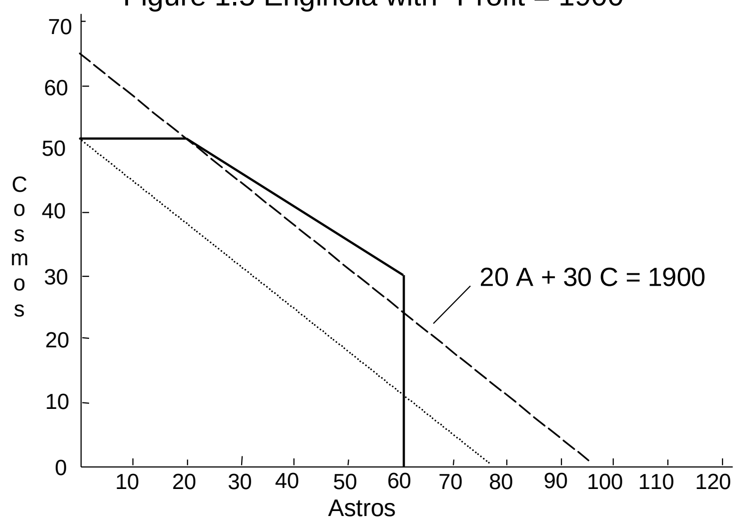
If we next talk with the manager of the Astro line, the response might be: "If you produce 50 Cosmos, you still have enough labor to produce 20 Astros. This would give a profit of $30 \times 50 + 20 \times 20 = \$1900$. That is certainly a respectable profit. Why don't we call it a day and go home?"

Figure 1.4 Enginola With "Profit = 1500"



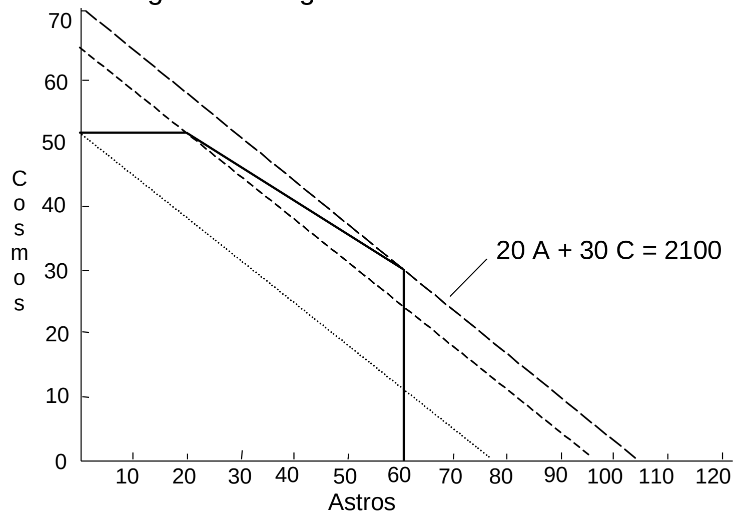
Our ever-alert reader might again observe that there are many ways of making $1900 of profit. If you plot the line 20*A* + 30*C* = 1900 and add it to the graph, then you get Figure 1.3. Any point on the higher rightmost dotted line segment achieves a profit of $1900.

Figure 1.5 Enginola with "Profit = 1900"



Now, our ever-perceptive reader makes a leap of insight. As we increase our profit aspirations, the dotted line representing all points that achieve a given profit simply shifts in a parallel fashion. Why not shift it as far as possible for as long as the line contains a feasible point? This last and best feasible point is *A* = 60, *C* = 30. It lies on the line 20*A* + 30*C* = 2100. This is illustrated in Figure 1.4. Notice, even though the profit contribution per unit is higher for Cosmo, we did not make as many (30) as we feasibly could have made (50). Intuitively, this is an optimal solution and, in fact, it is. The graphical analysis of this small problem helps understand what is going on when we analyze larger problems.

Figure 1.6 Enginola with "Profit = 2100"



1.6.1 Linearity

We have now seen one example. We will return to it regularly. This is an example of a linear mathematical program, or LP for short. Solving linear programs tends to be substantially easier than solving more general nonlinear mathematical programs. Therefore, it is worthwhile to dwell for a bit on the linearity feature.

Linear programming applies *directly* only to situations in which the effects of the different activities in which we can engage are linear. For practical purposes, we can think of the linearity requirement as consisting of three features:

1. *Proportionality.* The effects of a single variable or activity by itself are proportional (e.g., doubling the amount of steel purchased will double the dollar cost of steel purchased).
2. *Additivity*. The interactions among variables must be additive (e.g., the dollar amount of sales is the sum of the steel dollar sales, the aluminum dollar sales, etc.; similarly, the amount of electricity used is the sum of that used to produce steel, aluminum, etc).
3. *Continuity.* The variables must be continuous (i.e., fractional values for the decision variables, such as 6.38, must be allowed). If both 2 and 3 are feasible values for a variable, then so is 2.51.

A model that includes the two decision variables “price per unit sold” and “quantity of units sold” is probably not linear. The proportionality requirement is satisfied. However, the interaction between the two decision variables is multiplicative rather than additive (i.e., *dollar sales* = *price* × *quantity*, not *price* + *quantity*).

If a supplier gives you quantity discounts on your purchases, then the cost of purchases will not satisfy the proportionality requirement (e.g., the total cost of the stainless steel purchased may be less than proportional to the amount purchased).

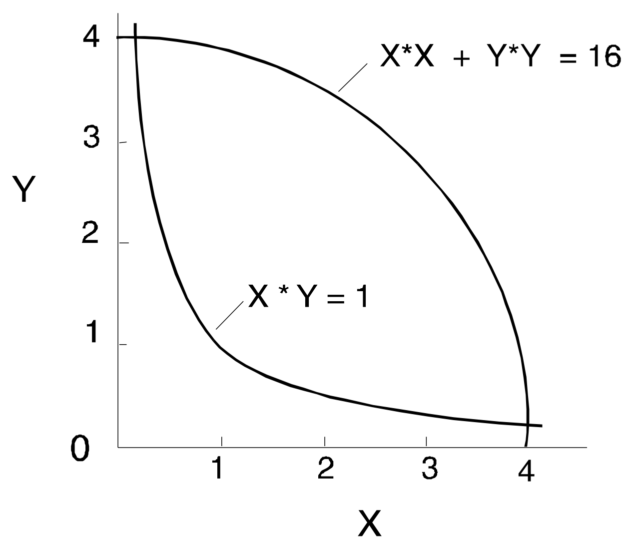
A model that includes the decision variable "number of floors to build" might satisfy the proportionality and additivity requirements, but violate the continuity conditions. The recommendation to build 6.38 floors might be difficult to implement unless one had a designer who was ingenious with split level designs. Nevertheless, the solution of an LP might recommend such fractional answers.

The possible formulations to which LP is applicable are substantially more general than that suggested by the example. The objective function may be minimized rather than maximized; the direction of the constraints may be ≥ rather than ≤, or even =; and any or all of the parameters (e.g., the 20, 30, 60, 50, 120, 2, or 1) may be negative instead of positive. The principal restriction on the class of problems that can be analyzed results from the linearity restriction.

Fortunately, as we will see later in the chapters on integer programming and quadratic programming, there are other ways of accommodating these violations of linearity.

Figure 1.5 illustrates some nonlinear functions. For example, the expression $X \times Y$ satisfies the proportionality requirement, but the effects of X and Y are not additive. In the expression $X^2 + Y^2$, the effects of X and Y are additive, but the effects of each individual variable are not proportional.

Figure 1.7: Nonlinear Relations

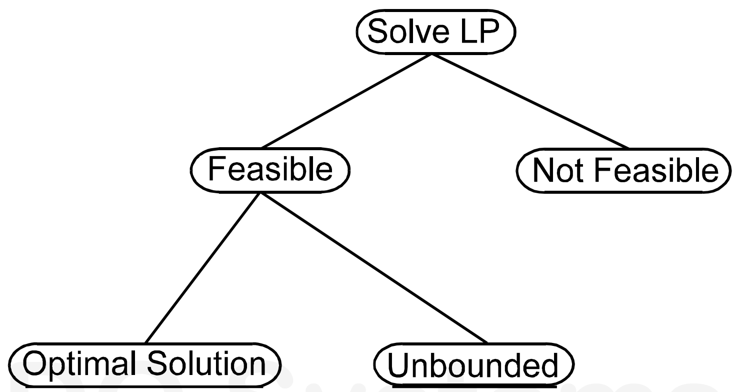


1.7 Analysis of Solutions

When you direct the What'sBest! to solve an optimization problem, the possible outcomes are indicated in Figure 1.8.

For a typical case, the leftmost path will be taken. The solution procedure will first attempt to find a feasible solution (i.e., a solution that simultaneously satisfies all constraints, but does not necessarily maximize the objective function). The rightmost, “No Feasible Solution”, path will be taken if the formulator has been too demanding. That is, two or more constraints are specified that cannot be simultaneously satisfied. A simple example is the pair of constraints $x \le 2$ and $x \ge 3$. The nonexistence of a feasible solution does not depend upon the objective function. It depends solely upon the constraints. In practice, the “No Feasible Solution” outcome might occur in a large complicated problem in which an upper limit was specified on the number of productive hours available and an unrealistically high demand was placed on the number of units to be produced. An alternative message to “No Feasible Solution” is “You Can’t Have Your Cake and Eat It Too”.

Figure 1.8 Solution Outcomes



If a feasible solution has been found, then the procedure attempts to find an optimal solution. If the “Unbounded Solution” termination occurs, it implies the formulation admits the unrealistic result that an infinite amount of profit can be made. A more realistic conclusion is that an important constraint has been omitted or the formulation contains a critical typographical error.

1.7.1 Sensitivity Analysis, Dual Prices, and Reduced Costs

A user of a model should be concerned with how the recommendations of the model are altered by changes in the input data. Sensitivity analysis is the term applied to the process of answering this question. Fortunately, an optimization solution report can provide supplemental information that is useful in sensitivity analysis. This information falls under two headings, reduced costs and dual prices.

Sensitivity analysis can reveal which pieces of information should be estimated most carefully. For example, if it is blatantly obvious that a certain product is unprofitable, then little effort need be expended in accurately estimating its costs. The first law of modeling is "do not waste time accurately estimating a parameter if a modest error in the parameter has little effect on the recommended decision".

1.7.2 Dual Prices

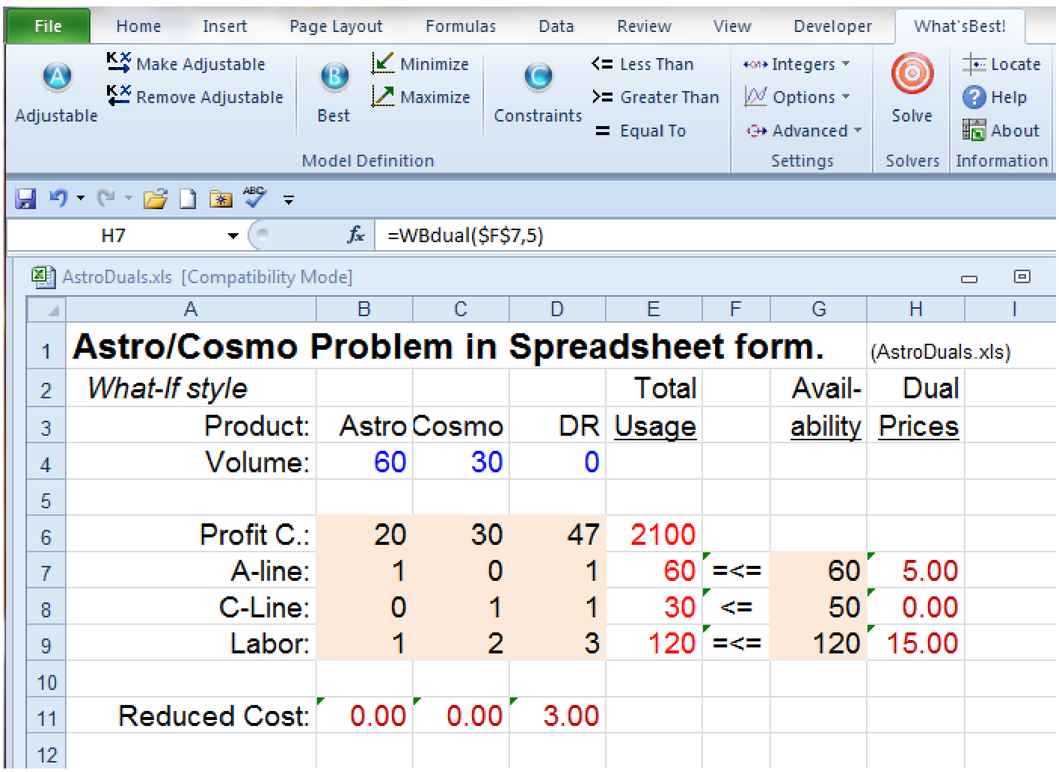
Associated with each constraint is a quantity known as the *dual price*. The dual price of a constraint is the rate at which the objective function value will improve as the right-hand side or constant term of the constraint is increased a small amount. If the units of the objective function are dollars and the units of the constraint in question are kilograms, then the units of the dual price are dollars per kilogram.

Different optimization programs may use different sign conventions with regard to the dual prices. What'*sBest*! uses the convention that a positive dual price means increasing the right-hand side constant term in question will improve the objective function value, whereas a negative dual price means an increase in the right-hand side constant term will cause the objective function value to worsen. A zero dual price means changing the right-hand side a small amount will have no effect on the solution value.

It follows that, under this convention, ≤ constraints will have nonnegative dual prices, ≥ constraints will have nonpositive dual prices, and = constraints can have dual prices of any sign. Why?

In order to illustrate dual prices, we have generalized the Enginola problem by adding a third product, Digital Recorders, or DR for short. A DR is a little more complicated than the other two products. It requires one unit of A-line capacity, one unit of C-line capacity and three units of Labor.

Figure 1.9 Solution with Dual Prices



Understanding Dual Prices. It is instructive to analyze the dual prices in the solution to the Enginola problem. The dual price on the constraint $A \leq 60$ is \$5/unit. At first, one might suspect this quantity should be \$20/unit because, if one more Astro is produced, the simple profit contribution of this unit is $20. An additional Astro unit will require sacrifices elsewhere, however. Since all of the labor supply is being used, producing more Astros would require the production of Cosmos to be reduced in order to free up labor. The labor tradeoff rate for Astros and Cosmos is ½.. That is, producing one more Astro implies reducing Cosmo production by ½ of a unit. The net increase in profits is $20 – (1/2)* $30 = $5, because Cosmos have a profit contribution of $30 per unit.

Now, consider the dual price of $15/hour on the labor constraint. If we have 1 more hour of labor, it will be used solely to produce more Cosmos. One Cosmo has a profit contribution of $30/unit. Since 1 hour of labor is only sufficient for one half of a Cosmo, the value of the additional hour of labor is $15.

1.7.3 Reduced Costs

Associated with each variable in any solution is a quantity known as the *reduced cost*. If the units of the objective function are dollars and the units of the variable are gallons, then the units of the reduced cost are dollars per gallon. The reduced cost of a variable is the amount by which the profit contribution of the variable must be improved (e.g., by reducing its cost) before the variable in question would have a positive value in an optimal solution. Obviously, a variable that already appears in the optimal solution will have a zero reduced cost.

It follows that a second, correct interpretation of the reduced cost is that it is the rate at which the objective function value will deteriorate if a variable, currently at zero, is arbitrarily forced to increase a small amount. Suppose the reduced cost of x is $2/gallon. This means, if the profitability of x were increased by $2/gallon, then 1 unit of x (if 1 unit is a “small change”) could be brought into the solution without affecting the total profit. Clearly, the total profit would be reduced by $2 if x were increased by 1.0 without altering its original profit contribution.

1.7.4 Unbounded Formulations

If we forget to include the labor constraint and the constraint on the production of Cosmos, then an unlimited amount of profit is possible by producing a large number of Cosmos. This is illustrated here:

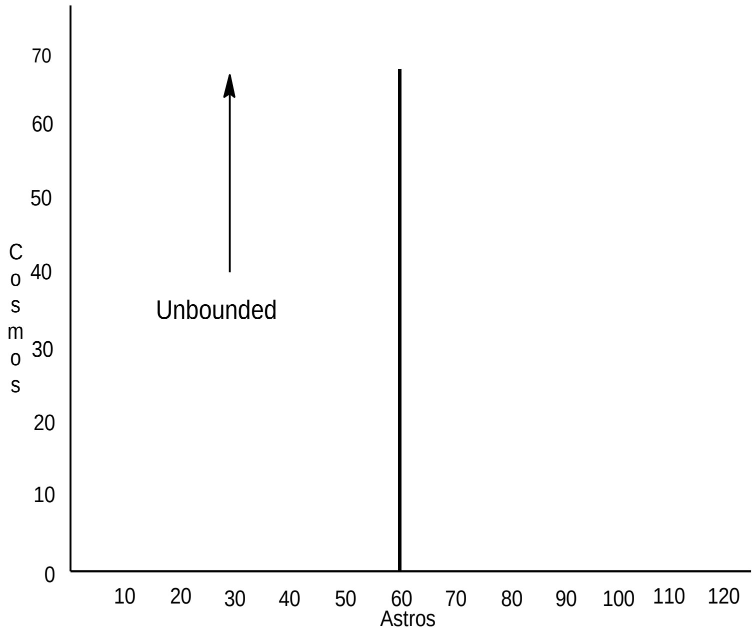
```
Maximize 20 * A + 30 * C;
A ≤ 60;
```

This generates an error window with the message:

```
UNBOUNDED SOLUTION
```

There is nothing to prevent C from being infinitely large. The feasible region is illustrated in Figure 1.7. In larger problems, there are typically several unbounded variables and it is not as easy to identify the manner in which the unboundedness arises.

Figure 1.10 Graph of Unbounded Formulation



1.7.5 Infeasible Formulations

An example of an infeasible formulation is obtained if the right-hand side of the labor constraint is made 190 and its direction is inadvertently reversed. In this case, the most labor that can be used is to produce 60 Astros and 50 Cosmos for a total labor consumption of $60 + 2 \times 50 = 160$ hours. The formulation and attempted solution are:

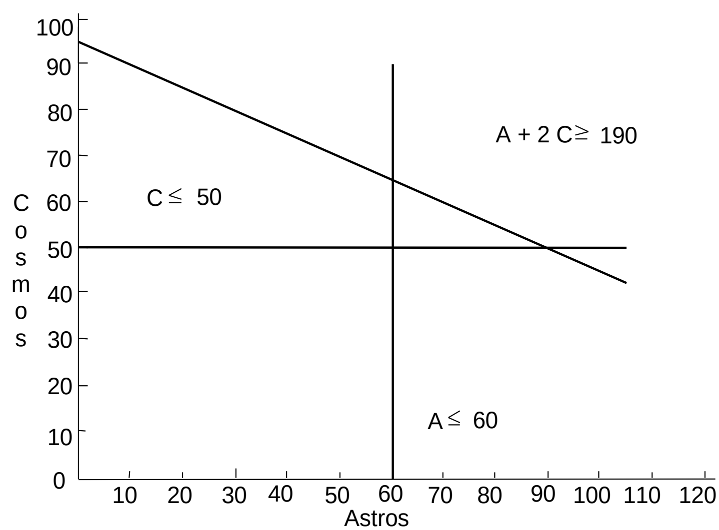
```
MAX = (20 * A) + (30 * C);
A <= 60;
C <= 50;
A + 2 * C >= 190;
```

If you solve it you may get a display as follows:

	A	B	C	D	E	F	G	H
1	The Astro Cosmo Problem							
2	We can produce Astros and Cosmos. How many, is constrained by							
3	Astro line capacity, Cosmo line capacity, and Labor.							
4		Astros	Cosmos					
5	Production:	90	50					
6				Totals				
7	Profit contribution:	20	30	3300		Available		
8	A-line capacity:	1	0	90	Not <=	60		
9	C-line capacity:	0	1	50	=<=	50		
10	Labor:	1	2	0	=>=	190		
11								
12								

Notice that one of the constraints says “Not <=”. The displayed “solution” is feasible to the labor constraint but violates the A-line capacity constraint. Figure 1.8 illustrates the constraints for this formulation.

Figure 1.11 Graph of Infeasible Formulation

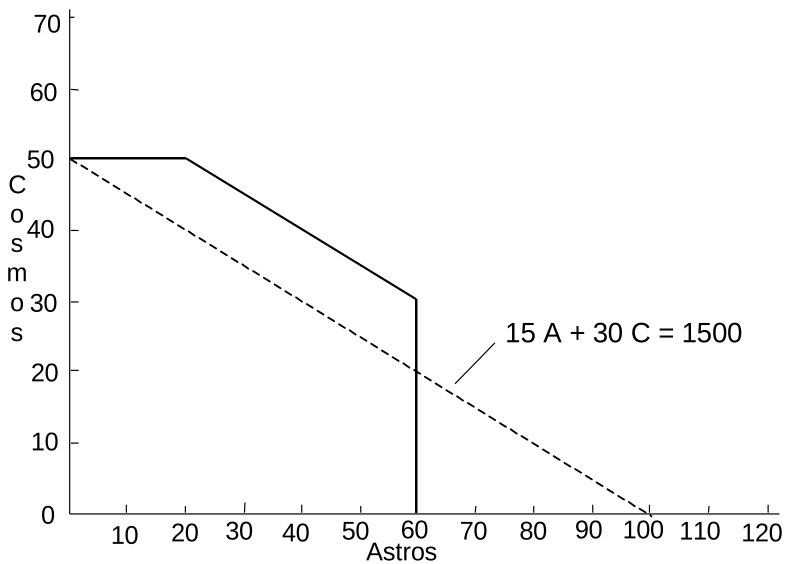


1.8 Multiple Optimal Solutions and Degeneracy

For a given formulation that has a bounded optimal solution, there will be a unique optimum objective function value. However, there may be several different combinations of decision variable values (and associated dual prices) that produce this unique optimal value. Such solutions are said to be degenerate in some sense. In the Enginola problem, for example, suppose the profit contribution of *A* happened to be $15 rather than $20. The problem is:

```
MAX = 15 * A + 30 * C;
A <= 60;
C <= 50;
A + 2 * C <= 120;
```

Figure 1.12 Model with Alternative Optima



The feasible region, as well as a “profit = 1500” line, are shown in Figure 1.9. Notice the lines $A + 2C = 120$ and $15A + 30C = 1500$ are parallel. It should be apparent that any feasible point on the line $A + 2C = 120$ is optimal. The maximum profit possible in this case is 1800. Thus, if you tradeoff Astros for Cosmos along the 15A + 30 C = 1800 line, you will not change the profit, even though you are changing the recommended solution. Two such extreme points are: 1) A = 60, C = 30, and 2) A = 20, C = 50. Below is a solution you may get from What’s*Best*!.

	A	B	C	D	E	F	G	H
1	**Astro/Cosmo Problem in Spreadsheet form.**							
2				Total		Avail-	Dual	
3	Product:	Astro	Cosmo	Usage		ability	Prices	
4	Volume:	60	30					
5								
6	Profit C.:	15	30	1800				
7	A-line:	1	0	60	=<=	60.0	0.00	
8	C-Line:	0	1	30	<=	50.0	0.00	
9	Labor:	1	2	120	=<=	120.0	15.00	
10								
11	Reduced Cost:	0.00	0.00					
12								

If you want to discover the alternate optimum that favors Cosmo production, you can solve the problem:

```
MAX = 15 * A + 30.0001 * C;
A <= 60;
C <= 50;
A + 2 * C <= 120;
```


If you solve it, you will see that the profit is still about $1800. However, the production of Cosmos has been increased to 50 from 30, whereas there has been an decrease in the production of Astros to 20 from 60.

1.8.1 The “Snake Eyes” Condition

Alternate optima may exist only if at the reported optimum: a) some constraint has both a slack of 0 and a dual price of 0, or b) some variable has both a value of 0 and has a reduced cost of 0. Notice that in the above solution report, the A-Line constraint has both a slack of 0 and a dual price of 0. This “double 0” configuration is called “snake eyes” by some applied statisticians. Mathematicians, with no intent of moral judgment, refer to such solutions as degenerate.

If there are alternate optima, you may find your computer gives a different solution from that in the text. However, you should always get the same objective function value.

There are, in fact, two ways in which multiple optimal solutions can occur. For the example in Figure 1.12, the two optimal solution reports differ only in the values of the so-called primal variables (i.e., our original decision variables *A*, *C)* and the slack variables in the constraint. There can also be situations where there are multiple optimal solutions in which only the dual variables differ. Consider this variation of the Enginola problem in which the capacity of the Cosmo line has been reduced to 30.

The formulation is:

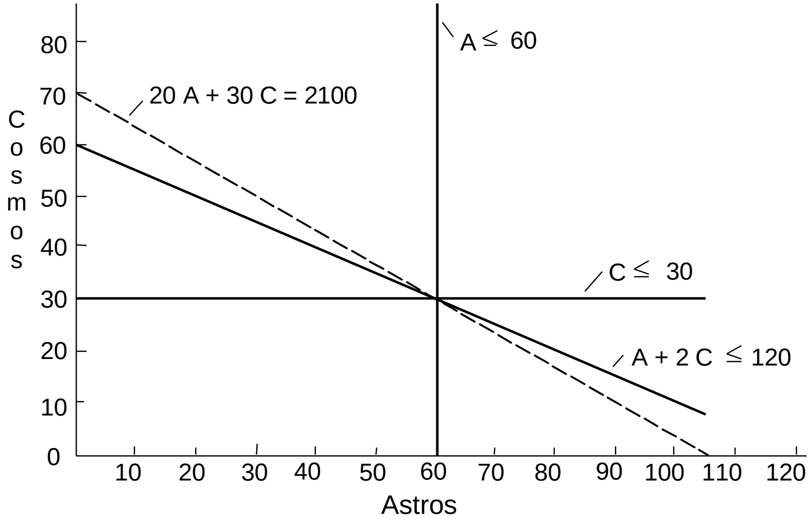
```
MAX = 20 * A + 30 * C;
A <= 60;
C <= 30;
 A + 2 * C <= 120;
```

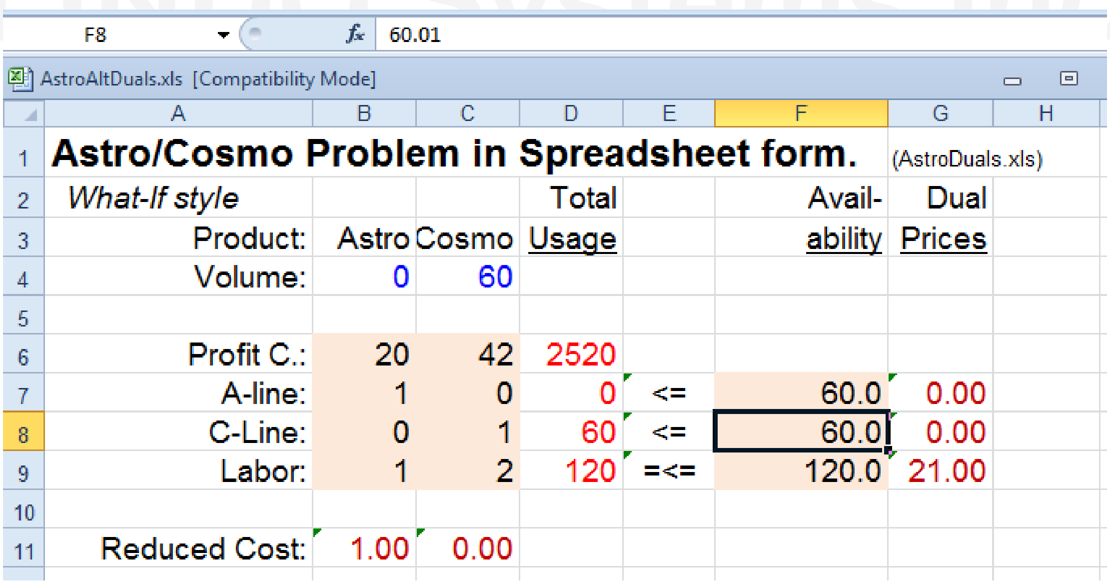
The corresponding graph of this problem appears in Figure 1.10.

	A	B	C	D	E	F	G	H
1	**Astro/Cosmo Problem in Spreadsheet form.**							
2				Total		Avail-	Dual	
3	Product:	Astro	Cosmo	Usage		ability	Prices	
4	Volume:	60	30					
5								
6	Profit C.:	20	30	2100				
7	A-line:	1	0	60	=<=	60.0	20.00	
8	C-Line:	0	1	30	=<=	30.0	30.00	
9	Labor:	1	2	120	=<=	120.0	0.00	
10								
11	Reduced Cost:	0.00	0.00					
12								

Again, notice the "snake eyes" in the solution (i.e., the pair of zeroes in a row of the solution report). This suggests the capacity of the Cosmo line (the RHS of row 3) could be changed without changing the objective value. Figure 1.13 illustrates the situation. Three constraints pass through the point $A = 60$, $C = 30$. Any two of the constraints determine the point. In fact, the constraint $A + 2C \le 120$ is mathematically redundant (i.e., it could be dropped without changing the feasible region).

Figure 1.13 Alternate Solutions in Dual Variables





	A	B	C	D	E	F	G	H
1	**Astro/Cosmo Problem in Spreadsheet form.**						(AstroDuals.xls)	
2	*What-If style*			Total		Avail-	Dual	
3	Product:	Astro	Cosmo	Usage		ability	Prices	
4	Volume:	0	60					
5								
6	Profit C.:	20	42	2520				
7	A-line:	1	0	0	<=	60.0	0.00	
8	C-Line:	0	1	60	<=	60.0	0.00	
9	Labor:	1	2	120	=<=	120.0	21.00	
10								
11	Reduced Cost:	1.00	0.00					

F9 f_x 120.01

AstroAltDuals.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	**Astro/Cosmo Problem in Spreadsheet form.**						(AstroDuals.xls)	
2	*What-If style*			Total		Avail-	Dual	
3	Product:	Astro	Cosmo	Usage		ability	Prices	
4	Volume:	0	60					
5								
6	Profit C.:	20	42	2520				
7	A-line:	1	0	0.01	<=	60.0	0.00	
8	C-Line:	0	1	60	=<=	60.0	2.00	
9	Labor:	1	2	120	=<=	120.0	20.00	
10								
11	Reduced Cost:	0.00	0.00					

If you decrease the RHS of row 3 very slightly, you will get essentially the following solution:

```
Optimal solution found at step:        0
Objective value:                2100.000
Variable            Value        Reduced Cost
       A         60.00000           0.0000000
       C         30.00000           0.0000000
     Row     Slack or Surplus      Dual Price
       1         2100.000            1.000000
       2        0.0000000            5.000000
       3        0.0000000           0.0000000
       4        0.0000000            15.00000
```

Notice this solution differs from the previous one only in the dual values.

We can now state the following rule: If a solution report has the “snake eyes” feature (i.e., a pair of zeroes in any row of the report), then there may be an alternate optimal solution that differs either in the primal variables, the dual variables, or in both.

If a solution report exhibits the “snake eyes” configuration, a natural question to ask is: can we determine from the solution report alone whether the alternate optima are in the primal variables or the dual variables? The answer is “no”, as the following two problems illustrate.

Problem D **Problem P**

```
MAX = X +       Y;              MAX = X + Y;
      X +       Y + Z <= 1;           X + Y +       Z <= 1;
      X + 2 * Y     <= 1;             X +     2 * Z <= 1;
```

Both problems possess multiple optimal solutions. The ones that can be identified by the standard simplex solution methods are:

Solution 1

Problem D

```
      OBJECTIVE VALUE
 1)       1.00000000
Variable    Value       Reduced Cost
 X           1.000000       0 000000
 Y           0.000000       0.000000
 Z           0.000000       1.000000

Row         Slack or     Dual Prices
            Surplus
 2)          0.000000       1.000000
 3)          0.000000       0.000000
```

Problem P

```
      OBJECTIVE VALUE
 1)       1.00000000
Variable    Value       Reduced Cost
 X           1.000000       0.000000
 Y           0.000000       0.000000
 Z           0.000000       1.000000

Row         Slack or     Dual Prices
            Surplus
 2)          0.000000       1.000000
 3)          0.000000       0.000000
```

Solution 2

Problem D

```
      OBJECTIVE VALUE
 1)       1.00000000
Variable    Value       Reduced Cost
 X           1.000000       0.000000
 Y           0.000000       1.000000
 Z           0.000000       0.000000

            Slack or
Row         Surplus     Dual Prices
 2)          0.000000       0.000000
 3)          0.000000       1.000000
```

Problem P

```
      OBJECTIVE VALUE
 1)       1.00000000
Variable    Value       Reduced Cost
 X           0.000000       0.000000
 Y           1.000000       0.000000
 Z           0.000000       1.000000

            Slack or
Row         Surplus     Dual Prices
 2)          0.000000       1.000000
 3)          1.000000       0.000000
```


Notice that:

- *Solution* 1 is exactly the same for both problems;
- *Problem D* has multiple optimal solutions in the dual variables (only); while
- *Problem P* has multiple optimal solutions in the primal variables (only).

Thus, one cannot determine from the solution report alone the kind of alternate optima that might exist. You can generate *Solution 1* by setting the RHS of row 3 and the coefficient of *X* in the objective to slightly larger than 1 (e.g., 1.001). Likewise, *Solution 2* is generated by setting the RHS of row 3 and the coefficient of *X* in the objective to slightly less than 1 (e.g., 0.9999).

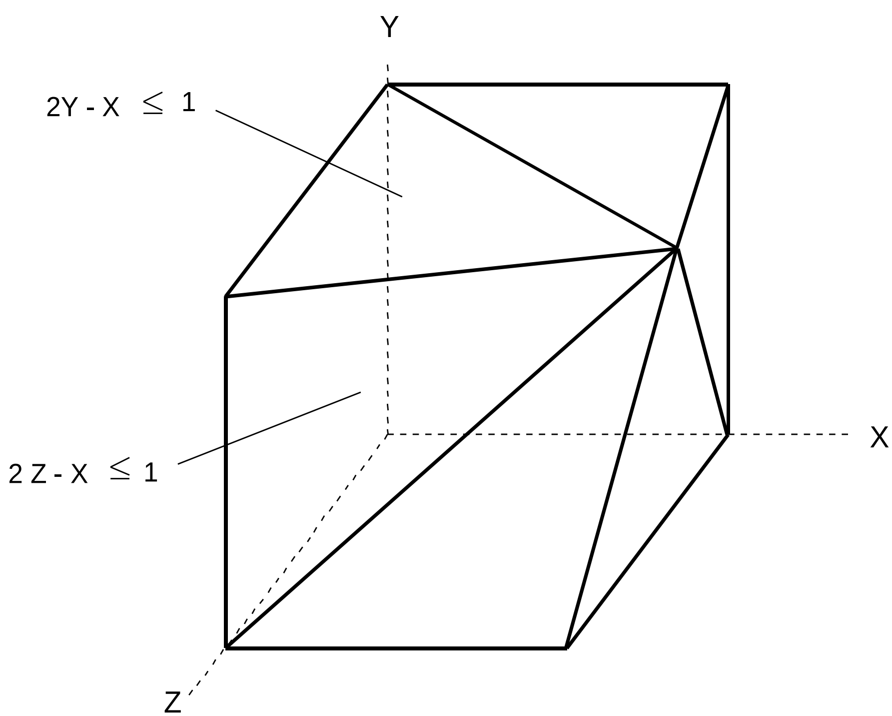
Some authors refer to a problem that has multiple solutions to the primal variables as *dual degenerate* and a problem with multiple solutions in the dual variables as *primal degenerate*. Other authors say a problem has multiple optima only if there are multiple optimal solutions for the primal variables.

1.8.2 Degeneracy and Redundant Constraints

In small examples, degeneracy usually means there are redundant constraints. In general, however, especially in large problems, degeneracy does not imply there are redundant constraints. The constraint set below and the corresponding Figure 1.11 illustrate:

$$2x - y \le 1$$
$$2x - z \le 1$$
$$2y - x \le 1$$
$$2y - z \le 1$$
$$2z - x \le 1$$
$$2z - y \le 1$$

Figure 1.14 Degeneracy but No Redundancy



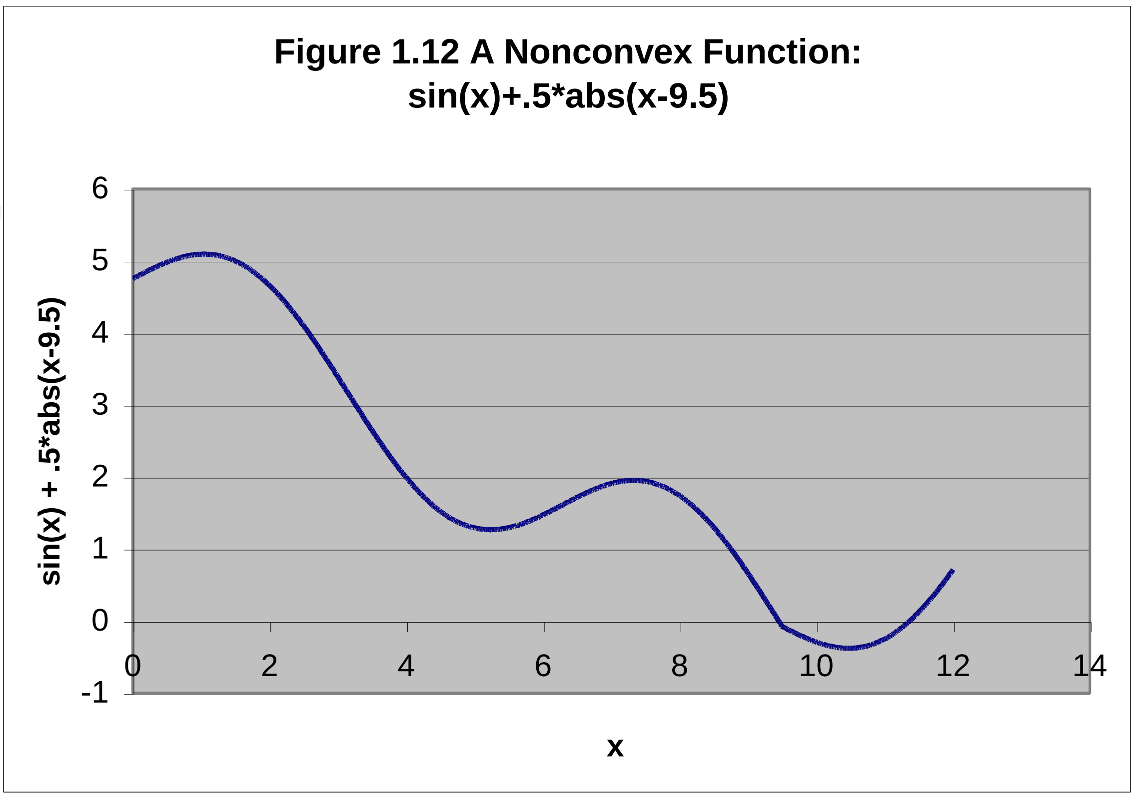
These constraints define a cone with apex or point at $x = y = z = 1$, having six sides. The point $x = y = z = 1$ is degenerate because it has more than three constraints passing through it. Nevertheless, none of the constraints are redundant. Notice the point $x = 0.6$, $y = 0$, $z = 0.5$ violates the first constraint, but satisfies all the others. Therefore, the first constraint is nonredundant. By trying all six permutations of 0.6, 0, 0.5, you can verify each of the six constraints are nonredundant.

1.9 Nonlinear Models and Global Optimization

Throughout this text the emphasis is on formulating linear programs. Historically nonlinear models were to be avoided, if possible, for two reasons: a) they take much longer to solve, and b) once "solved" traditional solvers could only guarantee that you had a locally optimal solution. A solution is a local optimum if there is no better solution nearby, although there might be a much better solution some distance away. Traditional nonlinear solvers are like myopic mountain climbers, they can get you to the top of the nearest peak, but they may not see and get you to the highest peak in the mountain range. For nonlinear models, What'sBest! has a global solver option, click on What'sBest! | Options | Global Solver… If you check the global solver option, then you are guaranteed to get a global optimum, if you let the solver run long enough. To illustrate, suppose our problem is:

```
Min = sin(x)+.5*abs(x-9.5);
          x<=12;
```

The graph of the function appears in Figure 1.12.



If you apply a traditional nonlinear solver to this model you might get one of three solutions, corresponding to the three local minima, either $x = 0$, or $x = 5.235987$, or $x = 10.47197$. If you turn on the Global solver option in What'sBest!, it will report the solution $x = 10.47197$ and label it as a global optimum. Be forewarned that the global solver does not eliminate drawback (a), namely, nonlinear models may take a long time to solve to guaranteed optimality. Nevertheless, the global solver may give a very good, even optimal, solution very quickly but then take a long time to prove that there is no other better solution.

1.9.1 Other Software for Optimization

There are alternatives to What's*Best*! for doing optimization. The most different approach is via modeling languages such as LINGO, see www.lindo.com. A modeling language allows you to describe an optimization model in notation very close to standard mathematical notation. The major advantages of a modeling language such as LINGO are:

1) Scalability and flexibility. It is very easy in LINGO to solve a
   3 supplier, 5 customer, 2 period problem today, and a
   10 supplier, 50 customer, 4 period problem tomorrow.
   No tedious copying of formulae is needed. Only new data need be entered.
2) Auditability: It is very easy to see all the formulae in one place, typically one page.
3) More than two dimensions are not a problem,
   e.g. 10 suppliers, 50 customers, as well as 12 periods, 60 products, and more dimensions.
4) Sparse sets are easily handled,
   e.g., not all suppliers carry all products, do not serve all customers, etc.

In contrast the advantages of modeling in a spreadsheet are:

1) Huge audience of users familiar with spreadsheets.
2) Excellent report formatting, graphing, etc.
3) Excellent for dense two dimensional problems,
   e.g., suppliers and customers, where every supplier can supply every customer.

There are alternative approaches to doing optimization in spreadsheets. Non-What's*Best*! format spreadsheet optimization models can be converted to What's*Best*! format by clicking on:

What's*Best*! | Advanced | Convert Model Format

1.10 Problems

1. Your firm produces two products, Thyristors (T) and Lozenges (L), that compete for the scarce resources of your distribution system. For the next planning period, your distribution system has available 6,000 person-hours. Proper distribution of each T requires 3 hours and each L requires 2 hours. The profit contributions per unit are 40 and 30 for T and L, respectively. Product line considerations dictate that at least 1 T must be sold for each 2 L's.

   (a) Draw the feasible region and draw the profit line that passes through the optimum point.
   (b) By simple common sense arguments, what is the optimal solution?

2. Graph the following LP problem:

   Minimize $4X + 6Y$
   subject to $5X + 2Y \geq 12$
   $3X + 7Y \geq 13$
   $X \geq 0,\ Y \geq 0.$

   In addition, plot the line $4X + 6Y = 18$ and indicate the optimum point.

3. The Volkswagen Company produces two products, the Bug and the SuperBug, which share production facilities. Raw materials costs are $600 per car for the Bug and $750 per car for the SuperBug. The Bug requires 4 hours in the foundry/forge area per car; whereas, the SuperBug, because it uses newer more advanced dies, requires only 2 hours in the foundry/forge. The Bug requires 2 hours per car in the assembly plant; whereas, the SuperBug, because it is a more complicated car, requires 3 hours per car in the assembly plant. The available daily capacities in the two areas are 160 hours in the foundry/forge and 180 hours in the assembly plant. Note, if there are multiple machines, the total hours available per day may be greater than 24. The selling price of the Bug at the factory door is $4800. It is $5250 for the SuperBug. It is safe to assume whatever number of cars are produced by this factory can be sold.

   (a) Write the linear program formulation of this problem.

   (b) The above description implies the capacities of the two departments (foundry/forge and assembly) are sunk costs. Reformulate the LP under the conditions that each hour of foundry/forge time cost $90; whereas, each hour of assembly time cost $60. The capacities remain as before. Unused capacity has no charge.

4. The Keyesport Quarry has two different pits from which it obtains rock. The rock is run through a crusher to produce two products: concrete grade stone and road surface chat. Each ton of rock from the South pit converts into 0.75 tons of stone and 0.25 tons of chat when crushed. Rock from the North pit is of different quality. When it is crushed, it produces a “50-50” split of stone and chat. The Quarry has contracts for 60 tons of stone and 40 tons of chat this planning period. The cost per ton of extracting and crushing rock from the South pit is 1.6 times as costly as from the North pit.

   (a) What are the decision variables in the problem?

   (b) There are two constraints for this problem. State them in words.

   (c) Graph the feasible region for this problem.

   (d) Draw an appropriate objective function line on the graph and indicate graphically and numerically the optimal solution.

   (e) Suppose all the information given in the problem description is accurate. What additional information might you wish to know before having confidence in this model?

5. A problem faced by railroads is of assembling engine sets for particular trains. There are three important characteristics associated with each engine type, namely, operating cost per hour, horsepower, and tractive power. Associated with each train (e.g., the Super Chief run from Chicago to Los Angeles) is a required horsepower and a required tractive power. The horsepower required depends largely upon the speed required by the run; whereas, the tractive power required depends largely upon the weight of the train and the steepness of the grades encountered on the run. For a particular train, the problem is to find that combination of engines that satisfies the horsepower and tractive power requirements at lowest cost.

   In particular, consider the Cimarron Special, the train that runs from Omaha to Santa Fe. This train requires 12,000 horsepower and 50,000 tractive power units. Two engine types, the GM-I and the GM-II, are available for pulling this train. The GM-I has 2,000 horsepower, 10,000 tractive power units, and its variable operating costs are $150 per hour. The GM-II has 3,000 horsepower, 10,000 tractive power units, and its variable operating costs are $180 per hour. The engine set may be mixed (e.g., use two GM-I's and three GM-II's).

   Write the linear program formulation of this problem.

6. Graph the constraint lines and the objective function line passing through the optimum point and indicate the feasible region for the Enginola problem when:

   (a) All parameters are as given except labor supply is 70 rather than 120.

   (b) All parameters are as given originally except the variable profit contribution of a Cosmo is $40 instead of $30.

7. Consider the problem:

$$
\begin{array}{ll}
\text{Minimize} & 4x_1 + 3x_2 \\
\text{Subject to} & 2x_1 + x_2 \ge 10 \\
& -3x_1 + 2x_2 \le 6 \\
& x_1 + x_2 \ge 6 \\
& x_1 \ge 0,\ x_2 \ge 0
\end{array}
$$

   Solve the problem graphically.

8. The surgical unit of a small hospital is becoming more concerned about finances. The hospital cannot control or set many of the important factors that determine its financial health. For example, the length of stay in the hospital for a given type of surgery is determined in large part by government regulation. The amount that can be charged for a given type of surgical procedure is controlled largely by the combination of the market and government regulation. Most of the hospital's surgical procedures are elective, so the hospital has considerable control over which patients and associated procedures are attracted and admitted to the hospital. The surgical unit has effectively two scarce resources, the hospital beds available to it (70 in a typical week), and the surgical suite hours available (165 hours in a typical week). Patients admitted to this surgical unit can be classified into the following three categories:

Patient Type	Days of Stay	Surgical Suite Hours Needed	Financial Contribution
A	3	2	$240
B	5	1.5	$225
C	6	3	$425

For example, each type *B* patient admitted will use (i) 5 days of the $7 \times 70 = 490$ bed-days available each week, and (ii) 1.5 hours of the 165 surgical suite hours available each week. One doctor has argued that the surgical unit should try to admit more type *A* patients. Her argument is that, "in terms of $/days of stay, type *A* is clearly the best, while in terms of $/(surgical suite hour), it is not much worse than *B* and *C*."

Suppose the surgical unit can in fact control the number of each type of patient admitted each week (i.e., they are decision variables). How many of each type should be admitted each week?

Can you formulate it as an LP?

2

Network Applications

2.1 What's Special About Networks

A subclass of models called network LPs warrants special attention for three reasons:

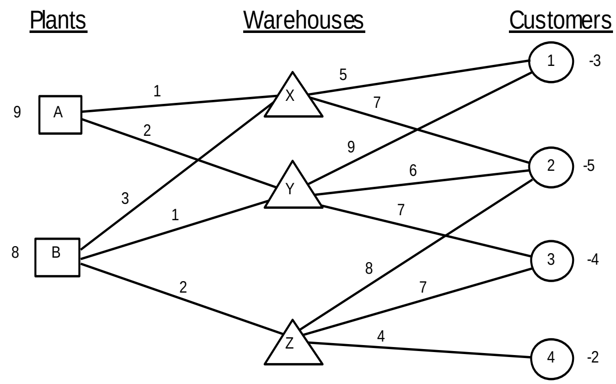
1. They can be completely described by simple, easily understood graphical figures.
2. Under typical conditions, solutions to network LP's have naturally integer answers.
3. They are frequently easier to solve than general LPs with the same number of constraints and variables.

Physical examples that come to mind are pipeline or electrical transmission line networks. Any enterprise producing a product at several locations and distributing it to many warehouses and/or customers may find a network LP a useful device for describing and analyzing shipment strategies.

Figure 2.1 illustrates the standard three level network representing the distribution system of a firm. Each plant produces just one or two products but in large amounts. Each customer needs small amounts of lots of different products. Intermediate warehouses or distribution centers (DC) are used to distribute a product. The firm in Figure 2.1 has two plants (denoted by *A* and *B*), three warehouses (denoted by *X*, *Y*, and *Z*), and four customer areas (denoted by 1, 2, 3, 4). The numbers adjacent to each node denote the availability of material at that node. Plant *A*, for example, has nine units available to be shipped. Customer 3, on the other hand, has −4 units meaning it needs to receive a shipment of four units.

The number above each arc is the cost per unit shipped along that arc. For example, if five of plant *A*'s nine units are shipped to warehouse *Y*, then a cost of $5 \times 2 = 10$ will be incurred as a direct result. The problem is to determine the amount shipped along each arc, so total costs are minimized and every customer has his requirements satisfied.

Figure 2.1 Three-Level Distribution Network



The essential condition on an LP for it to be a network problem is that it be representable as a network. There can be more than three levels of nodes, any number of arcs between any two nodes, and upper and lower limits on the amount shipped along a given arc.

With variables defined in an obvious way, the general LP describing this problem in algebraic (LINGO) form is:

```
[COST] MIN = AX + 2 * AY + 3 * BX + BY + 2 * BZ + 5 * X1
   + 7 * X2 + 9 * Y1 + 6 * Y2 + 7 * Y3 + 8 * Z2 + 7 * Z3
   + 4 * Z4;
[A] AX + AY <= 9;
[B] BX + BY + BZ <= 8;
[X] - AX - BX + X1 + X2 = 0;
[Y] - AY - BY + Y1 + Y2 + Y3 = 0;
[Z] - BZ + Z2 + Z3 + Z4 = 0;
[C1] - X1 - Y1 = -3;
[C2] - X2 - Y2 - Z2 = -5;
[C3] - Y3 - Z3 = -4;
[C4] - Z4 = -2;
```

There is one constraint for each node that is of a “sources = uses” form. Constraint [Y], for example, is associated with warehouse *Y* and states that the amount shipped out minus the amount shipped in must equal 0.

A different view of the structure of a network problem is possible by displaying just the coefficients of the above constraints arranged by column and row. In the picture below, note that the apostrophes are placed every third row and column just to help see the regular patterns:

	A	A	B	B	B	X	X	Y	Y	Y	Z	Z	Z	
	X	Y	X	Y	Z	1	2	1	2	3	2	3	4	
COST:	1	2	3	1	2	5	7	9	6	7	8	7	4	MIN
A:	1	1	'			'			'			'		= 9
B:	'	'	1	1	1	'	'	'	'	'	'	'	'	= 8
X:	−1		−1			1	1		'			'		=
Y:		−1		−1		'		1	1	1		'		=
Z:	'	'	'	'	−1	'	'	'	'	'	1	1	1	=
C1:			'			−1		−1	'			'		= −3
C2:			'			'	−1		−1		−1	'		= -5
C3:	'	'	'	'	'	'	'	'	'	−1	'	−1	'	= -4
C4:			'			'			'			'	−1	= −2

Notice a key feature of the constraint matrix of a network problem: disregarding any simple bound constraints on individual variables, each column has exactly two nonzeroes in the constraint matrix. One of these nonzeroes is a +1, whereas the other is a −1. According to the convention we have adopted, the +1 appears in the row of the node from which the arc takes material, whereas the row of the node to which the arc delivers material is a −1. On a problem of this size, you should be able to deduce the optimal solution manually simply from examining Figure 2.1. You may check it with the solution below:

Variable	Value	Reduced Cost
AX	3.000000	0.000000
AY	3.000000	0.000000
BX	0.000000	3.000000
BY	6.000000	0.000000
BZ	2.000000	0.000000
X1	3.000000	0.000000
X2	0.000000	0.000000
Y1	0.000000	5.000000
Y2	5.000000	0.000000
Y3	4.000000	0.000000
Z2	0.000000	3.000000
Z3	0.000000	1.000000
Z4	2.000000	0.000000

Row	Slack or Surplus	Dual Price
COST	100.000000	-1.000000
A	3.000000	0.000000
B	0.000000	1.000000
X	0.000000	1.000000
Y	0.000000	2.000000
Z	0.000000	3.000000
C1	0.000000	6.000000
C2	0.000000	8.000000
C3	0.000000	9.000000
C4	0.000000	7.000000

This solution exhibits two pleasing features found in the solution to any network problem:

1. If the right-hand side coefficients (the capacities and requirements) are integer, then the variables will also be integer.
2. If the objective coefficients are integer, then the dual prices will also be integer.

We can summarize network LPs as follows:

1. Associated with each node is a number that specifies the amount of commodity available at that node (negative implies that commodity is required.)
2. Associated with each arc are:
   a) a cost per unit shipped (which may be negative) over the arc,
   b) a lower bound on the amount shipped over the arc (typically zero), and
   c) an upper bound on the amount shipped over the arc (infinity in our example).

The problem is to determine the flows that minimize total cost subject to satisfying all the supply, demand, and flow constraints.

2.1.1 Special Cases

There are a number of common applications of LP models that are special cases of the standard network LP. The ones worthy of mention are:

1. *Transportation or distribution problems*. A two-level network problem, where all the nodes at the first level are suppliers, all the nodes at the second level are users, and the only arcs are from suppliers to users, is called a transportation, or distribution model.
2. *Shortest and longest path problems*. Suppose one is given the road network of the United States and wishes to find the shortest route from Bangor to San Diego. This is equivalent to a special case of a network or transshipment problem in which one unit of material is available at Bangor and one unit is required at San Diego. The cost of shipping over an arc is the length of the arc. Simple, fast procedures exist for solving this problem. An important first cousin of this problem, the longest route problem, arises in the analysis of PERT/CPM projects.
3. *The assignment problem*. A transportation problem in which the number of suppliers equals the number of customers, each supplier has one unit available, and each customer requires one unit, is called an assignment problem. An efficient, specialized procedure exists for its solution.
4. *Maximal flow*. Given a directed network with an upper bound on the flow on each arc, one wants to find the maximum that can be shipped through the network from some specified origin, or source node, to some other destination, or sink node. Applications might be to determine the rate at which a building can be evacuated or military material can be shipped to a distant trouble spot.

2.2 Assignment Problems

The assignment problem is a simple LP problem frequently encountered as a major component in more complicated practical problems. There are a number of problems in routing and sequencing that are essentially assignment problems with complications.

The assignment problem is:

Given a matrix of costs:

c_{ij} = cost of assigning task or object i to person or facility j,

and variables:

x_{ij} = 1 if task or object i is assigned to person or facility j.

Then, we want to:

Minimize $\sum_i \sum_j c_{ij}x_{ij}$

subject to

$\sum_i x_{ij} = 1$ for each object i, (each object is assigned to exactly one person)

$\sum_j x_{ij} = 1$ for each person i, (each person is assigned exactly one object)

$x_{ij} \geq 0$.

This problem is easy to solve as an LP and the x_{ij} will be naturally integer. Our description used a "minimize" objective. Alternatively, one might have situations where one wants a "maximize" objective with the same constraints. It is still called an assignment problem.

2.2.1 Example: Assigning In-bound to Out-bound Flights

Some large airlines base their route structure around the hub concept. An airline will try to have a large number of flights arrive at the hub airport during a certain short interval of time (e.g., 9 A.M. to 10 A.M.) and then have a large number of flights depart the hub shortly thereafter (e.g., 10 A.M. to 11 A.M.). This allows customers of that airline to travel between a large combination of origin/destination cities with one stop and at most one change of planes. For example, United Airlines uses Chicago as a hub, Delta Airlines uses Atlanta, and American uses Dallas/Fort Worth.

A desirable goal in using a hub structure is to minimize the amount of changing of planes (and the resulting moving of baggage) at the hub. The following little example illustrates how the assignment model applies to this problem.

A certain airline has six flights arriving at O'Hare airport between 9:00 and 9:30 A.M. The same six airplanes depart on different flights between 9:40 and 10:20 A.M. The average numbers of people transferring between incoming and leaving flights appear below:

	L01	L02	L03	L04	L05	L06
I01	20	15	16	5	4	7
I02	17	15	33	12	8	6
I03	9	12	18	16	30	13
I04	12	8	11	27	19	14
I05	0	7	10	21	10	32
I06	0	0	0	6	11	13

Flight I05 arrives too late to connect with L01. Similarly I06 is too late for flights L01, L02, and L03.

All the planes are identical. A decision problem is assigning planes from incoming flights to which outgoing flights. For example, if incoming flight *I02* is assigned to leaving flight *L03*, then 33 people (and their baggage) will be able to remain on their plane at the stop at O'Hare. How should incoming flights be assigned to leaving flights, so a minimum number of people need to change planes at the O'Hare stop? This problem can be formulated as an assignment problem if we define:

x_{ij} = 1 if incoming flight(task) *i* is assigned to ougoing flight *j*, 0 if not.

The objective is to maximize the number of people not having to change planes (alternatively, minimize the number having to change planes.) A formulation and solution is displayed in Figure 2.2.

Figure 2.2 Assigning In-bound Flights to Out-bound Flights

Plane_Assignment.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I
1	Task to Facility Assignment								
2	**Given the value of assigning task *i* to facility *j* for every combination of *i* and *j*,**								
3	**which assignments should be made, taking account that double assignments are not allowed.**								
4			**Facilities**						
5		**OutBnd_1**	**OutBnd_2**	**OutBnd_3**	**OutBnd_4**	**OutBnd_5**	**OutBnd_6**		
6	**Tasks**	**Value of doing Task i in Facility j**							
7	**InBnd_1**	20	15	16	5	4	7		
8	**InBnd_2**	17	15	33	12	8	6		
9	**InBnd_3**	9	12	18	16	30	13		
10	**InBnd_4**	12	8	11	27	19	14		
11	**InBnd_5**	-999	7	10	21	10	32		
12	**InBnd_6**	-999	-999	-999	6	11	13		
13									
14		135	**: Objective**						
15								**No. times**	**Do**
16		**The task assigments**						**task done.**	**once**
17	**InBnd_1**	1	0	0	0	0	0	1	=
18	**InBnd_2**	0	0	1	0	0	0	1	=
19	**InBnd_3**	0	1	0	0	0	0	1	=
20	**InBnd_4**	0	0	0	1	0	0	1	=
21	**InBnd_5**	0	0	0	0	0	1	1	=
22	**InBnd_6**	0	0	0	0	1	0	1	=
23	**# Assignments:**	1	1	1	1	1	1		
24	**Do once:**	=	=	=	=	=	=		
25									

Notice, we have used a -999 to make the connections that are impossible or prohibitively unattractive.

The key formulae of the model are:

The objective function: B14=SUMPRODUCT(B7:G12,B17:G22)

Each in-bound flight (task) must be assigned: I17=WB(H17,"=",1)

Each out-bound flight (facility) must be assigned: B24=WB(B23,"=",1)

The solution displayed is an optimal one. Notice that not every incoming flight is assigned to its most attractive outgoing flight, and not every outbound flight is assigned its most attractive inbound flight. The solution is naturally integer even though we did not declare any of the variables to be integer. Nevertheless, the number of people who must change planes is minimized.

2.3 Representing Arbitrary Networks in What's*Best*!

A spreadsheet is fine for representing two dimensional problems, such as small assignment and transportation problems, that have just the two dimensions: sources and destinations, but what if there are more than two dimensions, e.g., not just plants and customers, but also multiple DC’s, multiple time periods, multiple products, and more? Arranging such multi-dimensional problems with thousands of nodes and arcs on a two dimensional spreadsheet appears challenging. What can we do? The next section describes an approach that can describe an arbitrarily large, sparse network in systematic form in a spreadsheet. For a practical problem we might have a dozen plants, two dozen DC’s, 2000 customers, and perhaps 500 products. A further complication is “sparsity”. Each customer regularly buys only about 6 of our products, and each plant produces only a modest fraction of all the products.

We will describe a “flat table” or list approach. Essentially, we will describe the network by way of two lists: 1) a list of all nodes, including the attributes of each, and 2) a list of all arcs, including the attributes of each. This is so-called Normal form in database terminology. The Node list fairly simple. The What’s*Best*! version of it is displayed in Figure 2.3. Each node has Name and a Supply amount. A demand is entered as a negative supply. Node names must be unique. For the now, look at only columns C and D. We will shortly explain what is going on in columns E, F, and G.

Figure 2.3 Representing a Network: Node List

E6 =SUMIF(Arcs!B$10:B$23,Nodes!$C6,Arcs!F$10:F$23)+D6

netflofbig1.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H
1								
2			The Node list.				(netflobig1.xls)	
3			Note, a demand is entered as a negative supply.					
4						Force		
5			Node	Supply	Flow in	In >= Out	Flow out	
6			PlntA	9	9	>=	6	
7			PlntB	8	8	=>=	8	
8			WhsX	0	6	=>=	6	
9			WhsY	0	6	=>=	6	
10			WhsZ	0	2	=>=	2	
11			Cust1	-3	0	=>=	0	
12			Cust2	-5	0	=>=	0	
13			Cust3	-4	0	=>=	0	
14			Cust4	-2	0	=>=	0	
15			Guard	0	0	=>=	0	

The Arc list is also relatively simple and is displayed Figure 2.4 for our little three-level example. Each arc has a From node, a To node, Cost/unit flow, and Capacity. We have added two additional features for generality: 1) A capacity for each arc, labeled “Cap”, and 2) a “Guard” row at the end of the list. The purpose of the guard row is to avoid ambiguity when adding items to list. This ensures that any SUM’s that refer to a list are automatically expanded by Excel when an additional arc is inserted. There are constraints in column E that enforce the condition that the flow on an arc cannot exceed its capacity. We set the capacities equal to a large nonbinding number for this particular example. Column F, the flow, is to be determined by the optimization.

Figure 2.4 Representing a Network: Arc List

E10 =WB(D10,">=",F10)

netflofbig1.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I
1	Network Flow Optimization Model in What'sBest								(netflobig1.xls)
3									
4	The example below is based on section 8.1 of Optimization with LINGO.								
5					Flow cost=	100			
6									
7					Flo UB				
8	The arcs				Con-	Arc			
9	From	To	Cost	Cap	straint	Flow			
10	PlntA	WhsX	1	999	>=	6			
11	PlntA	WhsY	2	999	>=	0			
12	PlntB	WhsX	3	999	>=	0			
13	PlntB	WhsY	1	999	>=	6			
14	PlntB	WhsZ	2	999	>=	2			
15	WhsX	Cust1	5	999	>=	3			
16	WhsX	Cust2	7	999	>=	3			
17	WhsY	Cust1	9	999	>=	0			
18	WhsY	Cust2	6	999	>=	2			
19	WhsY	Cust3	7	999	>=	4			
20	WhsZ	Cust2	8	999	>=	0			
21	WhsZ	Cust3	7	999	>=	0			
22	WhsZ	Cust4	4	999	>=	2			
23	Guard	Guard	0	0	=>=	0			

The problem is to determine the flow over each arc, so as to

Minimize total cost of the flow;
subject to
Flow on each arc <= capacity of the arc,
Flow into each node >= flow out of the arc,

2.3.1 Representing a Network: Exploiting the SUMIF( ) Function

The challenge is how do we compute columns E (the flow into a node) and column G (flow out of a node). The important function that is used is Excel's SUMIF function. The general form of the SUMIF function is:

SUMIF(Look_into_range, value_to_lookup, range_to_sum_over).

On the Nodes tab, the formula in cell E9 is:

=SUMIF(Arcs!B$10:B$23,Nodes!$C9,Arcs!F$10:F$23)+D9

This means that Excel looks into the range B10:B23 (the "To" column of the arc list) on the Arcs tab, searching for a match for the node associated with Nodes!C9, (the node WhsY). Where it finds a match, it adds in the corresponding contents in the range Arc!F10:F23. Thus, this function sums up all flows into the node WhsY. A similar formula for flows out appears in column G of the Nodes tab.

When the model is solved, the solution found in the Arcs tab is found. The total cost is 100.

2.3.2 Model Flexibility

When developing a model, one should keep in mind flexibility and generality. How might someone wish to change or extend the model? The approach just described for modeling a network is quite flexible in terms of adding nodes and arcs to the network. Expanding the size of this formulation is quite straightforward, specifically:

To add a Link/Arc:
1) Insert an additional row in the "Arcs" tab.
2) Copy an existing row into it, to get formulae into it.
3) Enter data in the From, To, Cost, and Cap columns

To add a Node:
1) Insert an additional row in the Node tab.
2) Copy an existing row into it, to get formulae into it.
3) Enter data into: Node-name and supply.

2.4 PERT/CPM Networks and LP

PERT and Critical Path Method (CPM) are two closely related techniques for monitoring the progress of a large project. A key part of PERT/CPM is calculating the critical path. That is, identifying the subset of the activities that must be performed exactly as planned in order for the project to finish on time. Officially, PERT stands for Program Evaluation and Review Technique. PERT was given its name by Admiral William F. Raborn, who played a key role in managing the Polaris Fleet Ballistic Missile project around 1956-1958 on which PERT was first used. Craven(2001) makes the interesting observation that at the time, Raborn had a new bride whose nickname was Pert.

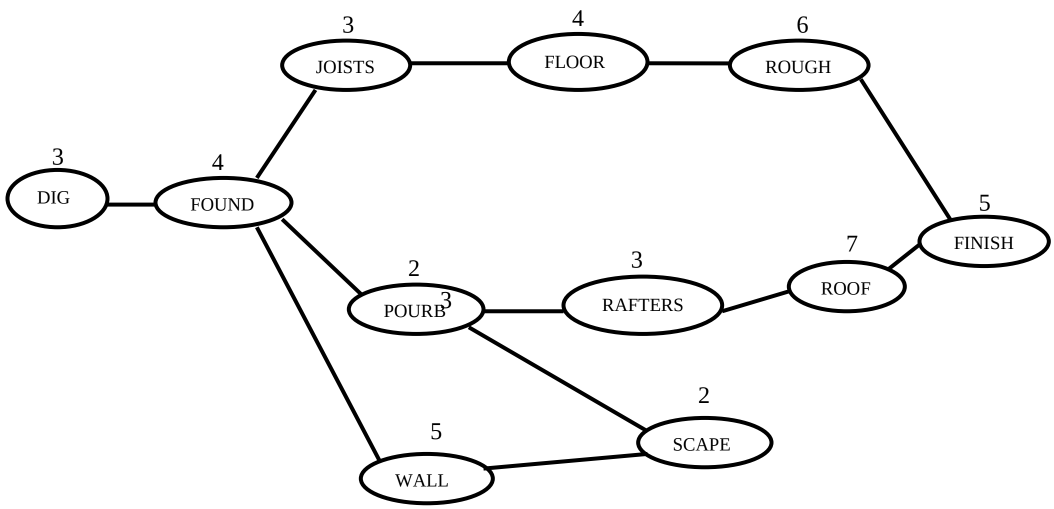
We will show that the calculation of the critical path is a very simple network LP problem, specifically, a longest path problem. You do not need this fact to efficiently calculate the critical path, but it is an interesting observation that becomes useful if you wish to examine a multitude of "crashing" options for accelerating a tardy project.

In the table below, we list the activities involved in the simple, but nontrivial, project of building a house. An activity cannot be started until all of its predecessors are finished.

Activity	Mnemonic	Activity Time	Predecessors (Mnemonic)
Dig Basement	DIG	3	—
Pour Foundation	FOUND	4	DIG
Pour Basement Floor	POURB	2	FOUND
Install Floor Joists	JOISTS	3	FOUND
Install Walls	WALLS	5	FOUND
Install Rafters	RAFTERS	3	WALLS, POURB
Install Flooring	FLOOR	4	JOISTS
Rough Interior	ROUGH	6	FLOOR
Install Roof	ROOF	7	RAFTERS
Finish Interior	FINISH	5	ROUGH, ROOF
Landscape	SCAPE	2	POURB, WALLS

In Figure 2.5, we show the so-called CPM (or activity-on-node) network for this project. We would like to calculate the minimum elapsed time to complete this project. Relative to this figure, the number of interest is simply the longest path from left to right in this figure. The project can be completed no sooner than the sum of the times of the successive activities on this path. Verify for yourself that the critical path consists of activities DIG, FOUND, WALLS, RAFTERS, ROOF, and FINISH and has length 27.

Figure 2.5 PERT/CPM Activity-on-Node Network for House Project



It is again convenient to use the node-list + arc-list style to represent this network in a spreadsheet. Figure 2.6 displays the spreadsheet tab containing the node list. Columns B and C contain the input data. Column D consists of a list of adjustable cells that will contain the finish time of each activity or task. Column E contains constraints that force the finish time of any task to be at least the task duration. For example, cell E13 contains the formula, =WB(C13,”<=”,E13). Column F contains constraints that force the Project time, in cell C25, to be greater than every finish time.

Figure 2.6 Node/Task List for PERT/CPM House Project

F12 =WB(C$25,">=",D12)

PertCPM.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	PERT/CPM Project Length Calculation						(PertCPM.xls)	
2		Given:						
3		A set of tasks, each with a task duration,						
4		Precedence constraints among some of the tasks,						
5		Problem:						
6		Minimize the maximum finish time of any task,						
7		No task can start earlier than the finish time of any of its predeccessors						
8								
9		A house building project.						
10			Duration	Finish		Project time		
11		Task	time, D(i)	time, F(i)	D(i) <= F(i)	>= F(i)		
12		DIG	3	3	=<=	>=		
13		FOUND	4	7	<=	>=		
14		POURB	2	9	<=	>=		
15		JOISTS	3	10	<=	>=		
16		WALLS	5	12	<=	>=		
17		RAFTERS	3	15	<=	>=		
18		FLOOR	4	14	<=	>=		
19		ROUGH	6	20	<=	>=		
20		ROOF	7	22	<=	>=		
21		FINISH	5	27	<=	=>=		
22		SCAPE	2	14	<=	>=		
23		Guard	0	0	=<=	>=		
24								
25	Project time:		27	To be minimized.				

For example, cell F21 contains the formula =WB(D21,”>=”,C425). The objective function is to minimize the project completion time in cell C25.

Figure 2.7 Arc/Precedence List for PERT/CPM House Project

D6 =SUMIF(Nodes!B$12:B$23,Arcs!B6,Nodes!$D$12:$D$23)

PertCPM.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J
1	PERT/CPM Project Length Calculation						(PertCPM.xls)			
2		The precedence constraints.								
3										
4				Predecessor		Successor				
5		Pred	Succ	finish time		start time				
6		DIG	FOUND	3	=<=	3				
7		FOUND	POURB	7	=<=	7				
8		FOUND	JOISTS	7	=<=	7				
9		FOUND	WALLS	7	=<=	7				
10		WALLS	RAFTERS	12	=<=	12				
11		POURB	RAFTERS	9	<=	12				
12		JOISTS	FLOOR	10	=<=	10				
13		FLOOR	ROUGH	14	=<=	14				
14		RAFTERS	ROOF	15	=<=	15				
15		ROUGH	FINISH	20	<=	22				
16		ROOF	FINISH	22	=<=	22				
17		POURB	SCAPE	9	<=	12				
18		WALLS	SCAPE	12	=<=	12				
19		Guard	Guard	0	=<=	0				
20										
21		To add a Precedence/Arc:								
22		1) Insert an additional row in the "Arcs" tab.								
23		2) Copy an existing row into it, to get formulae into it.								
24		3) Enter the Pred(ecessor) and Succ(essor) names in the Pred and Succ columns.								
25		To add a Task/Node:								
26		1) Insert an additional row in the "Nodes" tab.								
27		2) Copy an existing row into it, to get formulae into it.								
28		3) Enter data into the Task-name, and Time columns.								
29			The Node-names must be unique.							

Figure 2.7 shows the Arcs or Precedence tab of the spreadsheet. The Predecessor/Successor pairs are entered as data in columns B and C. Columns D, E, and F find the finish times of the Predecessor/Successor pairs and enforce the precedences. For example, cell D7 contains the formula: =SUMIF(Nodes!B$12:B$23,Arcs!B7,Nodes!$D$12:$D$23).

Cell F8 contains the formula:

=SUMIF(Nodes!B$12:B$23,Arcs!C8,Nodes!$D$12:$D$23)
-SUMIF(Nodes!B$12:B$23,Arcs!C8,Nodes!$C$12:$C$23).

The first SUMIF retrieves the finish time of the successor task in the precedence. The second SUMIF subtracts off the duration of the successor task.

When solved, we see from the Nodes tab that the minimum amount of time in which the project can be done is 27. From columns D,E, and F of the Arcs tab, we can see that the precedences that are not binding are: (POURB, RAFTERS), (ROUGH, FINISH), and (POURB, SCAPE)

2.4.1 Activity-on-Arc vs. Activity-on-Node Network Diagrams

Two conventions are used in practice for displaying project networks: (1) Activity-on-Arc (AOA) and (2) Activity-on-Node (AON). The characteristics of the two are:

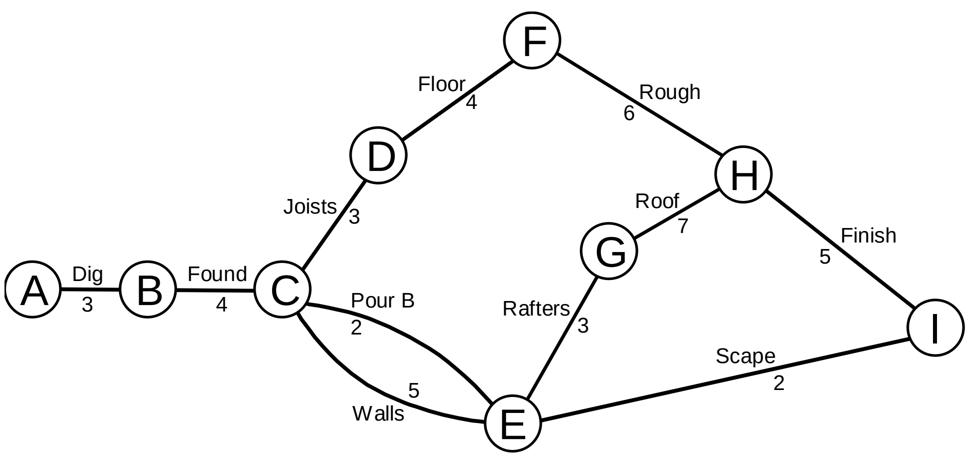
AON

- Each activity is represented by a node in the network.
- A precedence relationship between two activities is represented by an arc or link between the two.
- AON may be less error prone because it does not need dummy activities or arcs.

AOA

- Each activity is represented by an arc in the network.
- If activity *X* must precede activity *Y*, there are *X* leads into arc *Y*. The nodes thus represent events or “milestones” (e.g., “finished activity *X*”). Dummy activities of zero length may be required to properly represent some precedence relationships.
- AOA historically has been more popular, perhaps because of its similarity to Gantt charts used in scheduling.

Figure 2.8 Activity-on-Arc PERT/CPM Network



3

Formulating and Solving Integer Programs

"To be or not to be" is true.
-G. Boole

3.1 Introduction

In many applications of optimization, one would really like the decision variables to be restricted to integer values. One is likely to tolerate a solution recommending GM produce 1,524,328.37 Chevrolets. No one will mind if this recommendation is rounded up or down. If, however, a different study recommends the optimum number of aircraft carriers to build is 1.37, then a lot of people around the world will be very interested in how fraction 0.37 is rounded. It is clear the validity and value of many optimization models could be improved markedly if one could restrict selected decision variables to integer values.

Essentially all optimization modeling systems are augmented with a capability that allows the user to restrict certain decision variables to integer values. Many times, perhaps most of the time, one wants the possible values to be either 0 or 1. Such a cell or variable is said to be a binary variable. In What's*Best*! one can specify that a cell, or range of cells is to be restricted to integer values by: a) highlighting the range of cells with the cursor, and then b) click on either:

WB! | Integers | Integer-Binary | Binary or
WB! | Integers | Integer-Binary | General,

depending whether a binary (0 or 1) or general integer (0, 1, 2, . . .) variable is desired. You will also be prompted to give a name to the range of cells that are required to be integer.

We shall see later that, even though it is easy to specify the integer requirement, sometimes it may be difficult to solve problems with this restriction. The methods for formulating and solving problems with integrality requirements are called *integer programming*. The integrality enforcing capability is perhaps more powerful than the reader at first realizes. A frequent use of integer variables in a model is as a 0/1 variable to represent a go/no-go decision. It is probably true that the majority of real world integer programs are of the 0/1 variety, where the binary variables represent decisions to take or not take specific actions. You may think of them as "Hamlet" variables as in: "To buy or not to buy, that is the question".

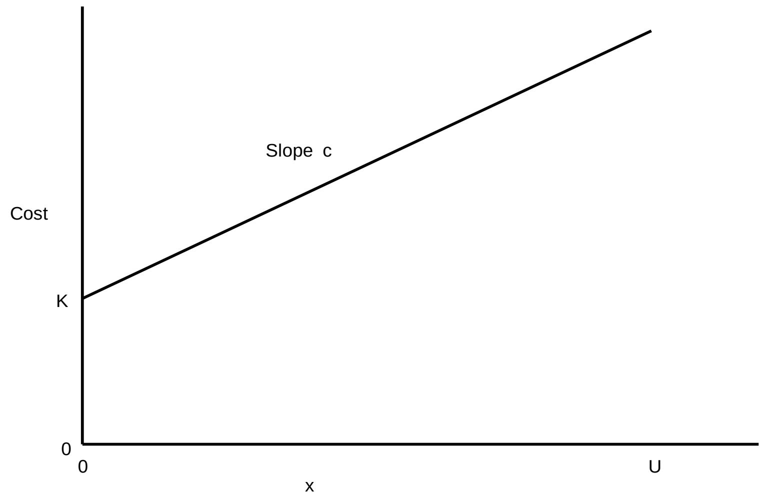
3.2 Exploiting the IP Capability: Standard Applications

You will frequently encounter problems that can be formulated as a linear program (LP) with the exception of just a few combinatorial complications. Many of these complications are fairly standard. The next sections describe many of the standard complications along with the methods for incorporating them into an integer programming (IP) formulation. Most of these complications require only the 0/1 capability rather than the general integer capability.

3.2.1 Fixed Charge Problems

A commonly encountered type of cost function is the fixed plus linear cost illustrated in Figure 3.1:

Figure 3.1 A Fixed Plus Linear Cost Curve



Let x be the volume of some activity, y be a binary (0 or 1) variable, U a given upper bound on x, c a given cost per unit, and K be the fixed cost incurred if $x > 0$. Then, the following components should appear in the formulation:

Minimize $K*y + c*x + \ldots$
subject to
$x \le U*y$
.
.
.

The constraint and the term Ky in the objective imply x cannot be greater than 0 unless a cost K is incurred. For computational efficiency, U should be as small as validly possible.

In Figure 3.2 is an example in What's*Best*! based on the Astro-Cosmo product mix problem in which a fixed charge is incurred if any positive amount of a product is produced. The first 10 rows describe the simple Astro-Cosmo problem without fixed charges. Rows 12:16 add the fixed charge features. Specifically, if you produce any Astros, a fixed charge of 800 must be incurred, regardless of how much is produced. The analogous charge for Cosmos is 900. The solution displayed is the optimal one, namely, produce 50 Cosmos, and no Astros for a net profit contribution of 600.

Figure 3.2 Representing a Decision having a Fixed Charge

AstroCos_1 [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	**The Astro Cosmo Problem**							
2		*We can produce Astros and Cosmos. How many, is constrained by*						
3		*Astro line capacity, Cosmo line capacity, and Labor.*						
4		Astros	Cosmos					
5	Production:	0	50					
6				Totals				
7	Profit contribution:	20	30	1500		Available		
8	A-line capacity:	1	0	0	<=	60		
9	C-line capacity:	0	1	50	=<=	50		
10	Labor:	1	2	100	<=	120		
11								
12	**Fixed charge section**			Fixed cost				
13	Fixed Charge (0/1) Vars:	0	1	incurred				
14	Fixed costs:	800	900	900				
15	Forcing constraints:	=<=	=<=					
16			Net profit:	600				

The formulae underlying the model are displayed in Figure 3.3. Observe that the equivalents of the constraint, $x \leq U*y$, appear in row 15. An upper bound on Astro production is the 60 appearing in cell F8. An upper bound on Cosmo production is the 50 in cell F9.

Figure 3.3 Fixed Charge Formulation Formulae

AstroCos_1 [Compatibility Mode]

	A	B	C	D	E	F
1	**The Astro Cosmo**	**Problem**				
2		*We can produce Astr*				
3		*Astro line capacity, (*				
4		Astros	Cosmos			
5	Production:	0	50			
6				Totals		
7	Profit contribution:	20	30	=SUMPRODUCT(B$5:C$5,B7:C7)		Available
8	A-line capacity:	1	0	=SUMPRODUCT(B$5:C$5,B8:C8)	=WB(D8,"<=",F8)	60
9	C-line capacity:	0	1	=SUMPRODUCT(B$5:C$5,B9:C9)	=WB(D9,"<=",F9)	50
10	Labor:	1	2	=SUMPRODUCT(B$5:C$5,B10:C10)	=WB(D10,"<=",F10)	120
11						
12	**Fixed charge section**			Fixed cost		
13	Fixed Charge (0/1) Vars:	0	1	incurred		
14	Fixed costs:	800	900	=SUMPRODUCT(B13:C13,B14:C14)		
15	Forcing constraints:	=WB(B5,"<=",F8*B13)	=WB(C5,"<=",F9*C13)			
16			Net profit:	=D7-D14		

3.2.2 Minimum Batch Size Constraints

When there are substantial economies of scale in undertaking an activity, many decision makers will specify a minimum "batch" size for the activity. For example, a financial firm may require that if you buy any bonds from the firm, you must buy at least 100. A zero/one variable can enforce this restriction as follows. Let:

x = activity level to be determined (e.g., number of bonds purchased),
y = a zero/one variable = 1, if and only if $x > 0$,
B = minimum batch size for x (e.g., 100), and
U = known upper limit on the value of x.

The following two constraints enforce the minimum batch size condition:

$$x \leq U^*y$$
$$B^*y \leq x.$$

If $y = 0$, then the first constraint forces $x = 0$. While, if $y = 1$, the second constraint forces x to be at least B. Thus, y acts as a switch, which forces x to be either 0 or greater than B. The constant U should be chosen with care. For reasons of computational efficiency, it should be as small as validly possible.

In Figure 3.4 is a version of the Astro-Cosmo problem in which minimum batch size, or production quantity, requirements are placed on the two products. The total profit contribution, to be maximized, is cell D7.

Figure 3.4 Representing a Decision having a Minimum Batch Size

AstroCos_MB [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	**The Astro Cosmo**	**Problem**						
2		*We can produce Astros and Cosmos. How many, is constrained by*						
3		*Astro line capacity, Cosmo line capacity, and Labor.*						
4		Astros	Cosmos					
5	Production:	40	40					
6				Totals				
7	Profit contribution:	20	30	2000		Available		
8	A-line capacity:	1	0	40	<=	60		
9	C-line capacity:	0	1	40	<=	50		
10	Labor:	1	2	120	=<=	120		
11								
12	**Min batch size section**							
13	Min batch (0/1) Vars:	1	1					
14	Min batch sizes:	35	40					
15	Forcing constraints, x ≤ Uy:	<=	<=					
16	Forcing constraints, x ≥ By:	>=	=>=					

Notice the min-batch size constraints appearing in rows 15 and 16 in Figure 3.5:

Figure 3.5 Minimum Batch Size Formulation Formulae

AstroCos_MB [Compatibility Mode]

	A	B	C	D	E	F
1	**The Astro Cosmo**	**Problem**				
2		*We can produce Astros*				
3		*Astro line capacity, Co*				
4		Astros	Cosmos			
5	Production:	40	40			
6				Totals		
7	Profit contribution:	20	30	=SUMPRODUCT(B$5:C$5,B7:C7)		Available
8	A-line capacity:	1	0	=SUMPRODUCT(B$5:C$5,B8:C8)	=WB(D8,"<=",F8)	60
9	C-line capacity:	0	1	=SUMPRODUCT(B$5:C$5,B9:C9)	=WB(D9,"<=",F9)	50
10	Labor:	1	2	=SUMPRODUCT(B$5:C$5,B10:C10)	=WB(D10,"<=",F10)	120
11						
12	**Min batch size section**					
13	Min batch (0/1) Vars:	1	1			
14	Min batch sizes:	35	40			
15	Forcing constraints, x ≤ Uy:	=WB(B5,"<=",F8*B13)	=WB(C5,"<=",F9*C13)			
16	Forcing constraints, x ≥ By:	=WB(B5,">=",B14*B13)	=WB(C5,">=",C14*C13)			

3.2.2 Using Semi-Continuous Variables for Min Batch Size Constraints

Minimum batch size constraints can be represented directly in What'sBest! by means of semi-continuous variables. A variable x is semi-continuous if it is required to be either 0 or in the range $B \leq x \leq U$, for given parameters B and U. No binary variable need be explicitly introduced. In What's*Best*! you can identify a semi-continuous variable by clicking on:

WB! | Integers | Integer-Binary | Semi-Continuous

and then supplying: 1) the lower bound, 2) the upper bound, and 3) the cell to store the condition that the cell is semi-continuous. Figure 3.6 gives the previous Astro-Cosmo problem, but using the Semi-Continuous feature, in default presentation form:

Figure 3.6 Using a Semi-Continuous Variable to Model Minimum Batch Size

B14 f_x =WBSEMIC(B13,F8,B$5)

AstroCos_SC.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	**The Astro Cosmo**	**Problem**						
2		*We can produce Astros and Cosmos. How many, is constrained by*						
3		*Astro line capacity, Cosmo line capacity, and Labor.*						
4		Astros	Cosmos					
5	Production:	40	40					
6				Totals				
7	Profit contribution:	20	30	2000		Available		
8	A-line capacity:	1	0	40	<=	60		
9	C-line capacity:	0	1	40	<=	50		
10	Labor:	1	2	120	=<=	120		
11								
12	**Min batch size section**	If we produce any, we must produce at least the min.						
13	Min batch sizes:	35	40					
14	SemiContin. constraint:	WBSEMIC	WBSEMIC					

The form of the WBSEMIC function can be seen at the top of the screenshot in the formula bar. The statement =WBSEMIC(B13,F8,B5) enforces the condition that either the value of B5 is 0 or it falls in the range of the values stored in B13 and F8, namely the range [35, 60].
The solution displayed is in fact the optimal solution.

3.2.3 Representing Logical Conditions

Binary variables are sometimes also called Boolean variables in honor of the logician George Boole. He developed the rules of the special algebra, now known as Boolean algebra, for manipulating variables that can take on only two values. In Boole's case, the values were "True" and "False".

However, it is a minor conceptual leap to represent "True" by the value 1 and "False" by the value 0. The power of these methods developed by Boole is undoubtedly the genesis of the modern compliment: "Strong, like Boole." For some applications, it may be convenient, perhaps even logical, to state requirements using logical expressions. A logical variable can take on only the values TRUE or FALSE. Likewise, a logical expression involving logical variables can take on only the values TRUE or FALSE. There are two major logical operators, #AND# and #OR#, that are useful in logical expressions.

The logical expression:

A #AND# *B*

is TRUE if and only if both *A* and *B* are true.

The logical expression:

A #OR# *B*

is TRUE if and only if at least one of *A* and *B* is true.

It is sometimes useful also to have the logical operator implication (⇒) written as follows:

$A \Rightarrow B$

with the meaning that if *A* is true, then *B* must be true.

Logical variables are trivially representable by binary variables with:

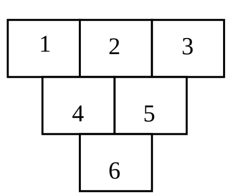
TRUE being represented by 1, and
FALSE being represented by 0.

If *A*, *B*, and *C* are 0/1 variables, then the following constraint combinations can be used to represent the various fundamental logical expressions:

Logical Expression	**Equivalent Mathematical Constraints**
C = *A* #AND# *B*	$C \le A$
	$C \le B$
	$C \ge A + B - 1$
C = *A* #OR# *B*	$C \ge A$
	$C \ge B$
	$C \le A + B$
$A \Rightarrow C$	$A \le C$

Example (A implies B and C):

In doing the long range planning for an open pit mine, the vertical region that is a candidate for mining is typically partitioned into blocks. Consider the following two dimensional simplification of the problem.



It should be clear that we can mine block 4 only if we have also mined blocks 1 AND 2. More generally, define:

y_i = 1 if we mine block i, else 0.

We can represent these logical conditions for our little mine with the following constraints:

$y_4 \leq y_1$; $y_4 \leq y_2$; (Block 4 can be removed only if blocks 1 and 2 are also removed.)
$y_5 \leq y_2$; $y_5 \leq y_3$; (Block 5 can be removed only if blocks 2 and 3 are also removed.)
$y_6 \leq y_4$; $y_6 \leq y_5$; (Block 6 can be removed only if blocks 4 and 5 are also removed.)

3.2.4 Start Up and Shut Down Costs

In the scheduling of generators in electric power industry, it is very important to take into account the cost of starting up and shutting down a generator. On a hot summer day, the demand for electricity varies dramatically over the course of the day. It is extremely expensive to shut down or start up a nuclear powered generator. It is not quite so expensive to start up and shut down a coal fired generator. It is relatively cheap to start up and shut down a natural gas fired generator. Not surprisingly, once running, a nuclear powered generator generates electricity most cheaply per kilo-watt-hour, whereas electricity from a gas fired generator is relatively more expensive per kilo-watt-hour. Schedule planning is usually done for anywhere from a day in advance to a week or more, with time partitioned into one hour periods. There is also a cost of keeping a generator running even though it is generating essentially no output. Define the 0/1 variables for modeling a single generator:

x_t = 1 if the generator is to be running in period t, else 0.
u_t = 1 if the generator is to start running at the beginning of period t, else 0.
v_t = 1 if the generator is to stop running at the beginning of period t, else 0.

In terms of logic, we want $u_t = 1$ if and only if $x_t = 1$ and $x_{t-1} = 0$. We want $v_t = 1$ if and only if $x_t = 0$ and $x_{t-1} = 1$. The start-up and shut-down relationship will be enforced by the following constraint for each period:

$$u_t - v_t = x_t - x_{t-1} ;$$

For unusual cases you may also need the constraint: $u_t + v_t \leq 1$.

We illustrate the startup/shutdown feature in What's*Best*! with a simple production planning problem where we have to pay a setup cost each time we start producing. We also have to pay inventory costs, so we do not want to long production runs that build up large inventories. We want to strike a happy compromise between starting up and shutting down a lot so as to closely track varying demand and keep inventory costs low, vs. having long production runs that keep setup costs low. The spreadsheet in standard form appears Figure 3.7.

Figure 3.7 A Model with Start-up and Shut-down Costs

SimpleProd

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Simple Production Problem with Startup, Shutdown, and Inventory costs.**												
2	Each period of production produces "batch size" units of goods.												
3													
4		Batch size		Inv. Cost		Startup cost		Shutdown cost					
5		5		2		16		5					
6	Period:	1	2	3	4	5	6	7	8	9	10	11	12
7	Produce:	0	1	1	1	0	0	0	0	1	1	0	0
8	Inventory:	0	3	6	6	5	4	4	3	4	7	4	2
9	Demand:	0	2	2	5	1	1	0	1	4	2	3	2
10	Inv >= 0:	=>=	>=	>=	>=	>=	>=	>=	>=	>=	>=	>=	>=
11	Startup:	0	1	0	0	0	0	0	0	1	0	0	0
12	Shutdown:	0	0	0	0	1	0	0	0	0	0	1	0
13	Startup >= 0:	=>=	>=	=>=	=>=	=>=	=>=	=>=	=>=	>=	=>=	=>=	=>=
14													
15	Inv costs:	96											
16	Startup costs:	32											
17	Shutdown costs:	10											
18	Total costs:	138											
19													

The same spreadsheet with formulae displayed appears in Figure 3.8. Notice the formulae in rows 11:13 that force the startup and shutdown variables to take on the proper value. The production variables in row 7 are declared binary (0/1).

Figure 3.8 Start-up and Shut-down Formulae

SimpleProd

	A	B	C	D	E	F	G	H
1	**Simple Production**							
2	Each period of productic							
3								
4		Batch size		Inv. Cost		Startup cost		Shutdown cost
5		5		2		16		5
6	Period:	1	=1+B6	=1+C6	=1+D6	=1+E6	=1+F6	=1+G6
7	Produce:	0	1	1	1	0	0	0
8	Inventory:	=$B5*B7-B9	=B8+$B5*C7-C9	=C8+$B5*D7-D9	=D8+$B5*E7-E9	=E8+$B5*F7-F9	=F8+$B5*G7-G9	=G8+$B5*H7-H9
9	Demand:	0	2	2	5	1	1	0
10	Inv >= 0:	=WB(B8,">=",0)	=WB(C8,">=",0)	=WB(D8,">=",0)	=WB(E8,">=",0)	=WB(F8,">=",0)	=WB(G8,">=",0)	=WB(H8,">=",0)
11	Startup:	=B7	=C7+C12-B7	=D7+D12-C7	=E7+E12-D7	=F7+F12-E7	=G7+G12-F7	=H7+H12-G7
12	Shutdown:	0	0	0	0	1	0	0
13	Startup >= 0:	=WB(B11,">=",0)	=WB(C11,">=",0)	=WB(D11,">=",0)	=WB(E11,">=",0)	=WB(F11,">=",0)	=WB(G11,">=",0)	=WB(H11,">=",0)
14								
15	Inv costs:	=D5*SUM(B8:M8)						
16	Startup costs:	=F5*SUM(B11:M11)						
17	Shutdown costs:	=H5*SUM(B12:M12)						
18	Total costs:	=SUM(B15:B17)						
19								

3.2.5 Knapsack Problems

A simple but common type of constraint that appears in lots of situations is the knapsack constraint. A binary knapsack constraint is a constraint of the form:

$w_1*y_1+ w_2*y_2+. . .+ w_n*y_n \leq b$;

where the w_j and b are given constants, and the y_j are 0/1 variables. Some example situations are:

The w_j represent	b represents	Situation
Pallet weights	Truck capacity	Deciding which pallets to load on a truck.
Material widths	Raw material width	Choosing a cutting pattern
Cost of a project	Annual budget	Deciding which projects get funded.

An example of a knapsack problem is illustrated below. The decision variables are in column D. The objective cell, to be maximized, is B16. An optimal solution is displayed. A simple heuristic for loading might be to start with the items with the higher value/weight and load the truck until it is full. Just for reference, in column G we calculated the value/weight of each item. Notice that this heuristic would be able to load only item 2 for a total value loaded of $24,000, vs. a value of $27,250 for an optimal solution. The optimal solution displayed in fact chooses the three items that have the least value/weight. It happens, however, that these three items fit together well within the very limited capacity of the truck.

Figure 3.9 A Knapsack/Truck Loading Problem

C16 =SUMPRODUCT(C7:C12,$D7:$D12)

TRUCK.XLS [Compatibility Mode]

Truck Loading/Knapsack Problem

Given a set of items, each with a weight and a value for delivering, which items should be loaded on a delivery truck that has a maximum weight limit?

Keywords: Knapsack, Loading, Truck Loading;

Item	Value	Weight	Loaded?(1)			Value/ weight
1	$22,500	7500	0			3
2	$24,000	7500	0			3.2
3	$8,000	3000	1			2.667
4	$9,500	3500	1			2.714
5	$11,500	4000	0			2.875
6	$9,750	3500	1			2.786
	Total Value of Load:	Total Weight of Load:		Maximum Load Weight		
	$27,250	10000	=<=	10000		

3.2.6 Bin Packing and Line Balancing Problems

A close cousin of the knapsack problem is the bin packing problem, a problem in which one has an unlimited number of knapsacks, or bins, available, each of a specified size, and one wants to find the minimum number of bins required to contain a collection of items, each of a specified size. A generalization of the bin packing problem is the line balancing problem. In a line balancing problem we want to set up a production line for high volume production of some item. A key feature of the line is that it must be partitioned into stations. A station is analogous to a knapsack or a bin. In the simplest form, one person works in each station and performs a specified set of tasks on each item that proceeds down the line. Only one task can be done at a time in each station and each task has a specified required time. It should be obvious that the production rate for the line is determined by the slowest station, that is, the station that has the most work assigned to it. A further complication is that there are precedence constraints among the tasks. The standard example of a precedence constraint is that you cannot put on your right shoe before you put on your right sock, although you could put on your right shoe (and sock) before you put on your left sock. As an example of precedence in assembling a mobile phone, you typically must insert the SIM card before inserting the battery, and the battery must be inserted before the cover is put in place. In some industries a mechanized production line is also known as a transfer line.

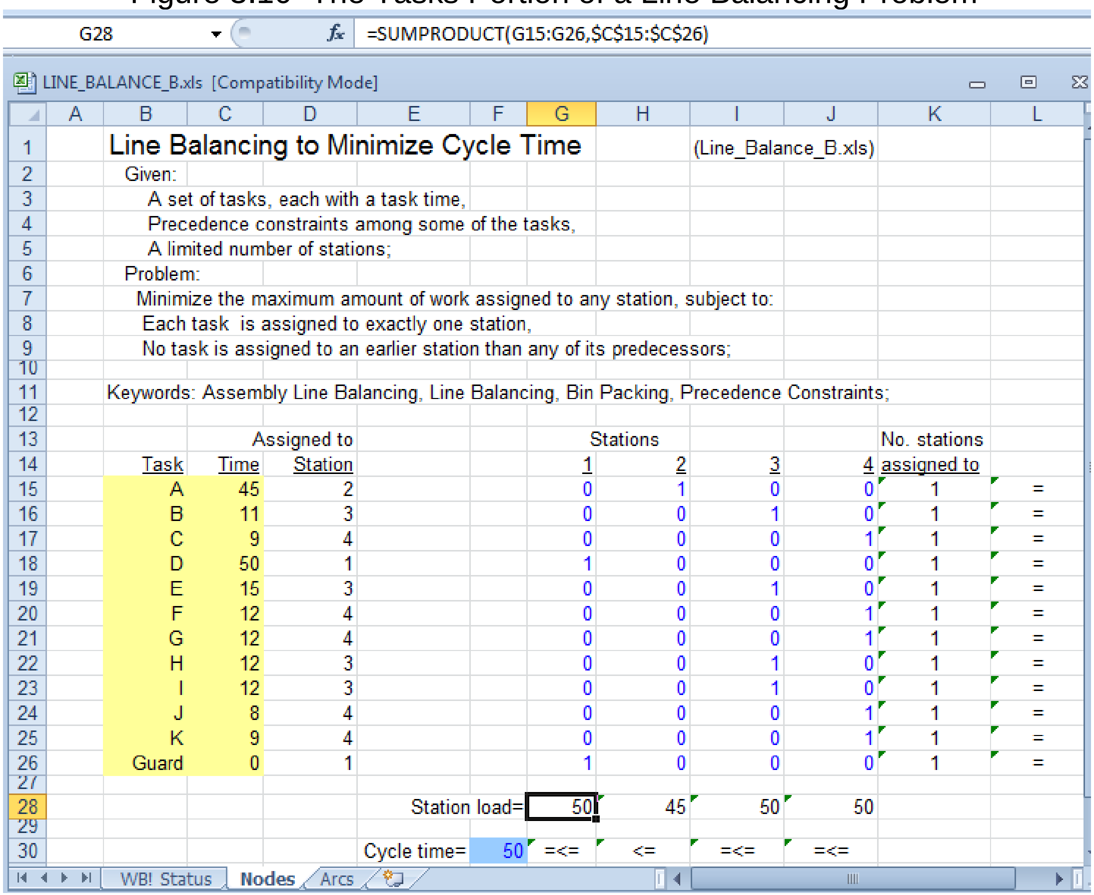
Perhaps the most well-known example of the production line approach to manufacturing is an automobile assembly line. Other products frequently produced on production lines are various kinds of appliances such as display monitors, printers, stoves, refrigerators, mobile phones, and lawn mowers. So we can summarize the simplest version of the line balancing problem in words as, we are given:

A set of tasks, each with a task time,
Precedence constraints among some of the tasks in the form of (predecessor, successor) pairs,
A limited number of stations (or bins), numbered 1, 2, …

Find:

An assignment of each task to a station so:
No task is assigned to an earlier station than any of its predecessors,
The maximum amount of work assigned to any station is minimized.

Figure 3.10 The Tasks Portion of a Line Balancing Problem


G28 =SUMPRODUCT(G15:G26,$C$15:$C$26)

LINE_BALANCE_B.xls [Compatibility Mode]

Line Balancing to Minimize Cycle Time (Line_Balance_B.xls)
Given:
A set of tasks, each with a task time,
Precedence constraints among some of the tasks,
A limited number of stations;
Problem:
Minimize the maximum amount of work assigned to any station, subject to:
Each task is assigned to exactly one station,
No task is assigned to an earlier station than any of its predecessors;

Keywords: Assembly Line Balancing, Line Balancing, Bin Packing, Precedence Constraints;

		Assigned to		Stations			No. stations	
Task	Time	Station	1	2	3	4	assigned to	
A	45	2	0	1	0	0	1	=
B	11	3	0	0	1	0	1	=
C	9	4	0	0	0	1	1	=
D	50	1	1	0	0	0	1	=
E	15	3	0	0	1	0	1	=
F	12	4	0	0	0	1	1	=
G	12	4	0	0	0	1	1	=
H	12	3	0	0	1	0	1	=
I	12	3	0	0	1	0	1	=
J	8	4	0	0	0	1	1	=
K	9	4	0	0	0	1	1	=
Guard	0	1	1	0	0	0	1	=
		Station load=	50	45	50	50		
		Cycle time= 50	=<=	<=	=<=	=<=		

WB! Status | Nodes | Arcs

For example, if the maximum amount of work in any station is 3 minutes, then the production rate is 1/3 units per minute, or 60/3 = 20 units per hour. A formulation of a line balancing problem appears in Figures 3.10 and 3.11. Precedences are naturally represented as a network, so we represent the precedences in this problem in the same way as earlier when we introduced network problems.
The features of the tasks, or nodes, are described in a Nodes tab in the spreadsheet in Figure 3.10. The precedence constraints are described in the Arcs tab shown in Figure 3.11. We can summarize the ABC's of the formulation as follows:

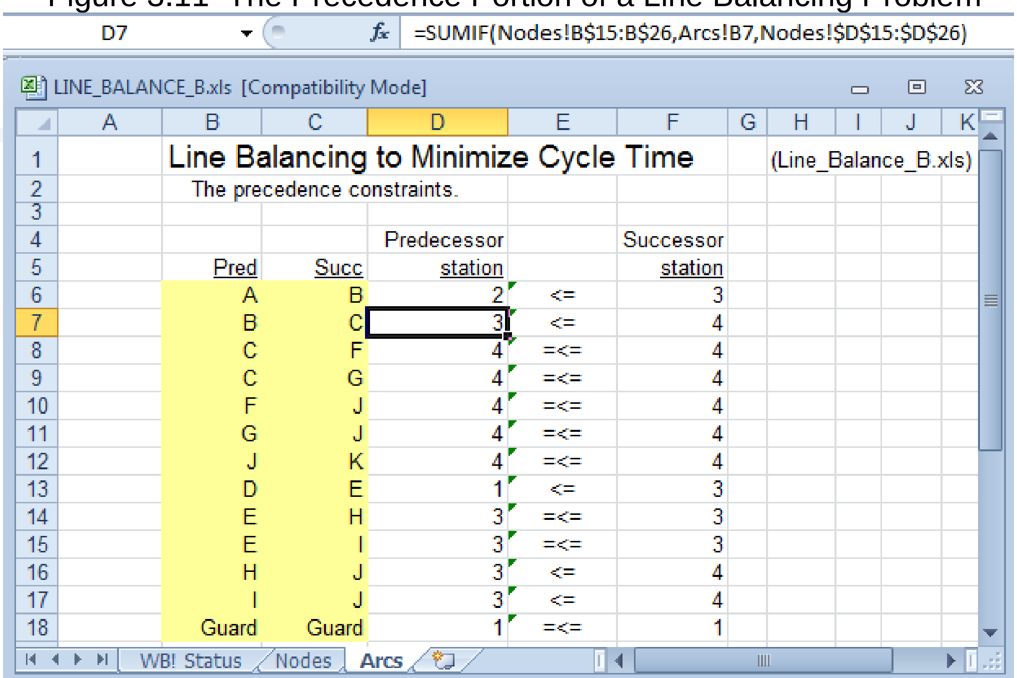
A) The Adjustable cells, or decision variables, are the 0/1 variables appearing in range G15:J26 of the Nodes tab. A "1" means that the task in column B is assigned to the station in row 14.

B) The Best or Objective cell is F30 in the Nodes tab. It is the total amount of work assigned to the busiest station and it is to be minimized.

C) The constraints on the Nodes tab are: 1) Column L has constraints that force each task to be assigned to exactly one station. 2) The constraints in row 30 force cell F30 to be at least as large as the amount of work assigned to any station in row 28 of the Nodes tab.

Column K of the Nodes tab sums up the number stations to which each task is assigned. For example, K17=SUM(G17:J17). The fact that this sum must be 1 is enforced in column L. For example, L17 =WB(K17,"=",1). Further, G30 =WB(G28,"<=",$F$30).

Figure 3.11 The Precedence Portion of a Line Balancing Problem



D7 =SUMIF(Nodes!B$15:B$26,Arcs!B7,Nodes!$D$15:$D$26)

LINE_BALANCE_B.xls [Compatibility Mode]

	A	B	C	D	E	F
1		Line Balancing to Minimize Cycle Time				(Line_Balance_B.xls)
2		The precedence constraints.				
3						
4				Predecessor		Successor
5		Pred	Succ	station		station
6		A	B	2	<=	3
7		B	C	3	<=	4
8		C	F	4	=<=	4
9		C	G	4	=<=	4
10		F	J	4	=<=	4
11		G	J	4	=<=	4
12		J	K	4	=<=	4
13		D	E	1	<=	3
14		E	H	3	=<=	3
15		E	I	3	=<=	3
16		H	J	3	<=	4
17		I	J	3	<=	4
18		Guard	Guard	1	=<=	1

WB! Status | Nodes | Arcs

The precedence constraints are enforced on the Arcs tab. Examples of the key cell formulae are:
D15=SUMIF(Nodes!B$15:B$26,Arcs!B7,Nodes!$D$15:$D$26);
F15=SUMIF(Nodes!B$15:B$26,Arcs!C15,Nodes!$D$15:$D$26);

E15 =WB(D15,"<=",F15).

We see from Figure 3.10, an optimal solution is:

Station:	1	2	3	4
Tasks assigned:	D	A	B, E, H, I	C, F, G, J, K
Work:	50	45	50	50

The maximum time in any station is 50. If time is in seconds, this means the line can produce 1/50 units per second, or 60/50 = 1.2 units per minute.

3.2.7 Binary Representation of General Integer Variables

A curious observation is that any general integer variable with a finite range can be represented by a small set of 0/1 variables. For example, suppose X is restricted to the set [0, 1, 2,...,15]. Introduce the four 0/1 variables: y_1, y_2, y_3, and y_4. Add the constraint: $X = y_1 + 2 \square y_2 + 4 \square y_3 + 8 * y_4$, and declare the y_j to be binary variables. Notice that every possible integer in [0, 1, 2, ..., 15] can be represented by some setting of the values of y_1, y_2, y_3, and y_4. Verify that, if the maximum value X can take on is 31, you will need five 0/1 variables. If the maximum value is 63, you will need 6 0/1 variables. In fact, if you use k 0/1 variables, the maximum value that can be represented is 2^k-1. Taking logarithms, you can observe that the number of 0/1 variables required in this so-called binary expansion is approximately proportional to the log of the maximum value X can take on.

Although this substitution is valid, it should be avoided if possible. Most integer programming algorithms are less efficient when applied to models containing this substitution. There are certain situations, however, where the binary expansion is convenient. Suppose that X above represents the decision of how many floors to have in a certain building. You want to consider all possible values for X in [0, 1, 2, ..., 15], except to avoid bad luck you want to prohibit X = 13. Notice that X = 13 corresponds to $y_1 = y_3 = y_4 = 1$ and $y_2 = 0$. Verify that adding the constraint $(1- y_1) + y_2 + (1-y_3) + (1- y_4) \geq 1$, or, $- y_1 + y_2 - y_3 - y_4) \geq -2$, will prohibit X = 13.

3.2.8 Plant Location Problems

The so-called capacitated plant location problem assumes that we have a number of customers, each with a known demand, a number of potential plant sites, each with an available capacity and a fixed cost of being open, and a shipment cost matrix that specifies the cost per unit of shipping from a given supply point to a customer demand point. The problems to a) decide which plants to open, and b) how much to ship from each open supply point to each demand point so as to minimize the total cost and not ship any more from a plant than its available capacity and shipping enough to each demand point to satisfy its demand. The problem formulation is:

Parameters:

D_j = volume or demand associated with customer j,
K_i = capacity of a plant located at i,
f_i = *fixed cost of having a plant at i,*
c_{ij} = cost per unit of shipping from i to j,

Variables:

x_{ij} = amount shipped from plant i to customer j,

y_i = 1 if plant i is open, else 0.

The IP formulation is:

Minimize $\sum_i f_i \, y_i + \sum_i \sum_j c_{ij} \, x_{ij}$ (Minimize fixed costs + shipping costs),
subject to

$\sum_j x_{ij} \leq K_i \, y_i$ for i = 1 to n, (Capacity constraints)
$\sum_i x_{ij} = D_j$ for j = 1 to m, (Demand constraints)
y_i = 0 or 1 for i = 1 to n. (Plant open or closed)

Example: Capacitated Plant Location

The Zzyzx Company of Zzyzx, California currently has a warehouse in each of the following cities: Baltimore, Cheyenne, Salt Lake City, Memphis, and Wichita. These warehouses supply customer regions throughout the U.S. It is convenient to aggregate customer areas and consider the customers to be located in the following cities: Atlanta, Boston, Chicago, Denver, Omaha, and Portland, Oregon. There is some feeling that Zzyzx is "overwarehoused". That is, it may be able to save substantial fixed costs by closing some warehouses without unduly increasing transportation and service costs. Relevant data have been collected and assembled on a "per month" basis and are displayed in a spreadsheet shown in Figure 3.12.

Figure 3.12 Capacitated Plant Location Data

PLANTLOCN.xls [Compatibility Mode]

PLANT LOCATION MODEL

Given transportation rates between cities and fixed costs of having a plant open,
1) Which supply plants should be opened, and
2) how much should be shipped from each plant to each demand city?

	Demand City							
	Atlanta	Boston	Chicago	Denver	Omaha	Portland	Monthly	
Demand:	10.0	8.0	12.0	6.0	8.0	11.0		Monthly
Candidate Supply Cities	Shipping Costs Per Ton / Month						Fixed Cost	Supply Capacity in Tons
Baltimore	1675	400	685	1630	1160	2800	$7,650	18.0
Cheyenne	1460	1940	970	100	495	1200	$3,500	24.0
Salt Lake City	1925	2400	1425	500	950	800	$5,000	27.0
Memphis	380	1355	543	1045	665	2321	$4,100	22.0
Wichita	922	1646	700	508	311	1797	$2,500	31.0

WB! Status | Data | Model_Decisions

For example, closing the warehouse at Baltimore would result in a monthly fixed cost saving of \$7,650. If Omaha gets all of its monthly demand from Wichita, then the associated transportation cost for supplying Omaha is 7 × 311 = \$2,177 per month. A customer need not get all of its supply from a single source. Such "multiple sourcing" may result from the limited capacity of each warehouse (e.g., Cheyenne can only process 24 tons per month. Should Zzyzx close any warehouses and, if so, which ones?)

To construct an optimization model, we put the decision variables and constraints on a separate tab, "Models_Decisions" and displayed in Figure 3.13.

Figure 3.13 Capacitated Plant Location Variables and Constraints

B16 =SUMPRODUCT(Data!B14:B18,Model_Decisions!B8:B12)

PLANTLOCN.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J
1										
2				Amount Shipped						
3										
4	Proposed			Demand City						
5	Supply							Open(1)	Supply	Incurred
6	City:	Atlanta	Boston	Chicago	Denver	Omaha	Portland	Plant	Limit	Cost
7										
8	Baltimore	0.0	8.0	1.0	0.0	0.0	0.0	1	<=	\$7,650
9	Cheyenne	0.0	0.0	0.0	6.0	7.0	11.0	1	=<=	\$3,500
10	Salt Lake City	0.0	0.0	0.0	0.0	0.0	0.0	0	=<=	\$0
11	Memphis	10.0	0.0	11.0	0.0	1.0	0.0	1	=<=	\$4,100
12	Wichita	0.0	0.0	0.0	0.0	0.0	0.0	0	=<=	\$0
13										
14	Meet Demand	=>=	=>=	=>=	=>=	=>=	=>=			
15								\$15,250	: Operating Cost	
16	Trans. Cost	\$3,800	\$3,200	\$6,658	\$600	\$4,130	\$13,200	\$31,588	: Trans. Costs	
17										
18						TOTAL OPERATING COST:		\$46,838		
19										

WB! Status | Data | Model_Decisions

The fixed costs incurred are computed in Cells J8:J12, e.g., J18 =H8*Data!H14.

Cell H15 sums them up, H15=SUM(J8:J12)

The shipping costs of supplying each demand city are computed in Cells B16:G16, e.g., B16=SUMPRODUCT(Data!B14:B18,B8:B12)

Cell H16 sums them up, H16=SUM(B16:G16)

The objective function, the total cost is in Cell H18, i.e., H18=H15+H16.

The demand constraints are enforced in Cells B14:G14, e.g., B14=WB(SUM(B8:B12),"=",Data!B9)

The capacity constraints are enforced in Cells I8:I12, e.g.,

I8=WB(SUM(B8:G8),"<=",Data!I14*H8).

3.3 Lotsizing Problems

It is interesting that multiperiod production planning problems can be formulated in a fashion very similar to plant location problems. The single product dynamic lotsizing problem is described by the following parameters:

n = number of periods for which production is to be planned for a product;
D_j = predicted demand in period j, for $j = 1, 2, \ldots, n$;
f_i = fixed cost of making a production run in period i;
h_i = cost per unit of product carried from period i to i + 1.

This problem can be cast as a simple plant location problem if we define:

$$c_{ij} = D_j \sum_{t=i}^{j-1} h_t.$$

That is, c_{ij} is the cost of supplying period j's demand from period i production. Each period can be thought of as both a potential plant site (period for a production run) and a customer.

If, further, there is a finite production capacity, K_i, in period i, then this capacitated dynamic lotsizing problem is a special case of the capacitated plant location problem.

3.2.8 Dual Prices and Reduced Costs in Integer Programs

Dual prices and reduced costs in solution reports for integer programs have a restricted interpretation. For first time users of IP, it is best to simply disregard the reduced cost and dual price column in the solution report. For the more curious, the dual prices and reduced costs in a solution report are obtained from the linear program that remains after all integer variables have been fixed at their optimal values and removed from the model. Thus, for a pure integer program (i.e., all variables are required to be integer), you will generally find:

- all dual prices are zero, and
- the reduced cost of a variable is simply its objective function coefficient (with sign reversed if the objective is MAX).

For mixed integer programs, the dual prices may be of interest. For example, for a plant location problem where the location variables are required to be integer, but the quantity-shipped variables are continuous, the dual prices reported are those from the continuous problem where the locations of plants have been specified beforehand (at the optimal locations).

3.4 Sequencing, Routing and the Assignment Problem

Recall that the assignment problem is a simple LP problem of the form:

Minimize $\sum_i \sum_j c_{ij}x_{ij}$

subject to

$\sum_i x_{ij} = 1$ for each object i, (each object is assigned to exactly one person)

$\sum_j x_{ij} = 1$ for each person i, (each person is assigned exactly one object)

$x_{ij} \geq 0$.

Many problems related to sequencing are generalizations of the assignment problem.

3.4.1 Sequencing Problems and the Traveling Salesperson Problem

One of the more famous optimization problems is the traveling salesperson problem (TSP). In a TSP, one wants to visit each of a given set of cities exactly once, covering a minimum distance. Lawler et al. (1985) presents a tour-de-force on this fascinating problem. One example of a TSP occurs in the manufacture of electronic circuit boards. Danusaputro, Lee, and Martin-Vega (1990) discuss the problem of how to optimally sequence the drilling of holes in a circuit board, so the total time spent moving the drill head between holes is minimized. A similar TSP occurs in circuit board manufacturing in determining the sequence in which components should be inserted onto the board by an automatic insertion machine. Another example is the sequencing of cars on a production line for painting: each time there is a change in color, a setup cost and time is incurred.

The TSP is a variation of the assignment problem, but with some additional conditions that happen to make TSP much more difficult than the assignment problem. A TSP is described by the data:

c_{ij} = cost of traveling directly from city i to city j, e.g., the distance.

A solution is described by the variables:

y_{ij} = 1 if we travel directly from i to j, else 0.

There must be exactly one link going into each city and exactly one link out of each city. These constraints correspond exactly to the Assignment problem. Not so obvious is that the links chosen must constitute a complete, connected tour of the cities. Let us first consider the Assignment formulation to see why it is not quite complete.

The Assignment Relaxation:

The assignment problem is a starting point for all formulations of the TSP. It is:

Minimize $\sum_i \sum_j c_{ij}y_{ij}$

subject to

(1) We must enter each city j exactly once:

$\sum_{i \neq j}^{n} y_{ij} = 1$ for $j = 1$ to n,

(2) We must exit each city i exactly once:

$\sum_{j \neq i}^{n} y_{ij} = 1$ for $i = 1$ to n,

(3) y_{ij} = 0 or 1, for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$, $i \neq j$:

An example in a spreadsheet appears in Figure 3.14.

Figure 3.14 Traveling Salesman Assignment Relaxation

WBMIN =SUMPRODUCT(C16:J23,C5:J12)

TSPAssgn.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Traveling Salesman Problem					**Assignment Relaxation**				(TSPAssgn.xls)			
2	We want to find a minimum distance route that visits each city exactly once.												
3					Distance matrix								
4		From - To:	Chicago	Denver;	Fresno	Houston	KC	LA	Oakland	Phoenix			
5		Chicago	0	996	2162	1067	499	2054	2134	1713			
6		Denver;	996	0	1167	1019	596	1059	1227	792			
7		Fresno	2162	1167	0	1747	1723	214	168	598			
8		Houston	1067	1019	1747	0	710	1538	1904	1149			
9		KC	499	596	1723	710	0	1589	1827	1214			
10		LA	2054	1059	214	1538	1589	0	371	389			
11		Oakland	2134	1227	168	1904	1827	371	0	755			
12		Phoenix	1713	792	598	1149	1214	389	755	0			
13													
14					"Links used" matrix. A "1" means go from row city to colum city.								Must exit
15			Chicago	Denver;	Fresno	Houston	KC	LA	Oakland	Phoenix		Sum	constraint
16		Chicago	0	0	0	0	1	0	0	0		1	=
17		Denver;	0	0	0	1	0	0	0	0		1	=
18		Fresno	0	0	0	0	0	0	1	0		1	=
19		Houston	0	1	0	0	0	0	0	0		1	=
20		KC	1	0	0	0	0	0	0	0		1	=
21		LA	0	0	0	0	0	0	0	1		1	=
22		Oakland	0	0	1	0	0	0	0	0		1	=
23		Phoenix	0	0	0	0	0	1	0	0		1	=
24													
25		Sum:	1	1	1	1	1	1	1	1		Total dist.	
26		Must enter:	=	=	=	=	=	=	=	=		4150	(Minimize)

The Adjustable cells are C16:J23.
The objective or Best cell is L26, containing the formula: =SUMPRODUCT(C16:J23,C5:J12).
The crucial formulae for the constraints are illustrated as follows:
1) We are forced to enter Denver by the formulae
In cell D25: =SUM(D16:D23) and in cell D26: =WB(D25,"=",1).
2) We are forced to depart Fresno by the formulae:
In cell L18: =SUM(C18:J18) and in cell M18: =WB(L18,"=",1).

The spreadsheet displays an optimal solution to the assignment problem. Unfortunately for the traveling salesman, notice that it contains subtours. Notice that from Chicago one goes to KC, and then from KC directly back to Chicago. We do not have a connected tour of all the cities. There are three other subtours, all with two cities: (Denver – Houston), (Fresno – Oakland) , and (Phoenix – LA).

We will describe following four formulations of the TSP that ensure connectedness/prevent subtours:

1) Single commodity formulation of Miller-Tucker-Zemlin (MTZ),
2) Subtour elimination cuts,
3) Multi-commodity formulation.
4) Time-space diagram,

The single commodity formulation is the most compact and most convenient for problems of a dozen or so cities, however, it may be difficult to solve problems with a large number of stops in a trip. The subtour elimination method, although complicated, is the most effective for large problems. The Time-space diagram approach is a useful way of looking at “Full Truck Load (FTL)” routing problems that are similar to TSP. The Multi-Commodity formulation is interesting because, although it is a very big, with about n^3 variables, it is simpler than the Subtour elimination approach, and in a certain theoretical sense, it is comparable to the Subtour elimination formulation.

3.4.2 Single Commodity Formulation

In the single commodity formulation of the n city TSP, think of a vehicle starting at city 1, picking up one unit of a certain commodity at each of the other cities. Define: u_j = the sequence number of city j on the trip, with city 1 having sequence number 0. Equivalently, u_j = cumulative number of units picked up after the stop at j. In order to force each city to have exactly one unit of commodity picked up as part of a single connected tour, we add the following $(n\text{-}1)^2$ constraints:

For $j = 2, 3, 4, \; . \; . \; ., n$:

For $i = 2, . . ., n, \quad j \neq i$:

$u_j \geq u_i + (n\text{-}1)y_{ij} \; - n + 2,$

$u_j \geq 2 - y_{1j} + (n\text{-}3)\, y_{j1},$

Notice that these constraints are consistent with the following observations:

if $y_{ij} = 1$, then $u_j \geq u_i + 1$, (it in fact holds as an =)
if $y_{ij} = 0$, then $u_j \geq u_i - (n - 2)$,
if $y_{1j} = 1$, then $u_j \geq 1$, (it in fact holds as an =)
if $y_{j1} = 1$, then $u_j \geq n\text{-}1$, (it in fact holds as an =)

This method of prohibiting subtours is attributed to Miller, Tucker, and Zemlin(1960). Problems with more than a dozen or so cities may take a long time to solve using the MTZ formulation. A spreadsheet illustrating the MTZ formulation appears in Figure 3.15.

Figure 3.15 Formulation of the Traveling Salesman Problem

G36 =WB($A$37,">=",$A36+($E$30-1)*G19+($E$30-3)*$F$20-$E$30+2)

TSP_MTZ.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Traveling Salesman Problem								(TSP_MTZ.xls)				
2	We want to find a route visiting each city exactly once so as to minimize the distance traveled.												
3					Distance matrix								
4		From;	Chicago	Denver	Fresno	Houston	KC	LA	Oakland	Phoenix			
5	1	Chicago	0	996	2162	1067	499	2054	2134	1713			
6	2	Denver	996	0	1167	1019	596	1059	1227	792			
7	3	Fresno	2162	1167	0	1747	1723	214	168	598			
8	4	Houston	1067	1019	1747	0	710	1538	1904	1149			
9	5	KC	499	596	1723	710	0	1589	1827	1214			
10	6	LA	2054	1059	214	1538	1589	0	371	389			
11	7	Oakland	2134	1227	168	1904	1827	371	0	755			
12	8	Phoenix	1713	792	598	1149	1214	389	755	0			
13			1	2	3	4	5	6	7	8			
14													Must exit
15					"Links used" matrix. A "1" means go from row city to colum city							Sum	constraint
16		Chicago	0	0	0	1	0	0	0	0		1	=
17		Denver	0	0	0	0	1	0	0	0		1	=
18		Fresno	0	0	0	0	0	0	1	0		1	=
19		Houston	0	0	0	0	0	0	0	1		1	=
20		KC	1	0	0	0	0	0	0	0		1	=
21		LA	0	0	1	0	0	0	0	0		1	=
22		Oakland	0	1	0	0	0	0	0	0		1	=
23		Phoenix	0	0	0	0	0	1	0	0		1	=
24													
25		Sum:	1	1	1	1	1	1	1	1		Total dist.	
26		Must enter:	=	=	=	=	=	=	=	=		5309	<<Minimize
27													
28	The single commodity/Miller/Tucker/Zemlin constraints for preventing subtours.												
29	Essentially: Position number of city j >= 1+ Position number of city i if the the link from i to j is used.												
30	Posn			Number cities:	8								
31	no.	Compute units left on vehicle upon exit of city:											
32			Chicago	Denver	Fresno	Houston	KC	LA	Oakland	Phoenix			
33	0	Chicago		>=	>=	=>=	=>=	>=	>=	=>=			
34	6	Denver			>=	>=	=>=	>=	=>=	>=			
35	4	Fresno		>=		>=	>=	=>=	=>=	>=			
36	1	Houston		>=	>=		>=	>=	>=	=>=			
37	7	KC		=>=	>=	=>=		>=	>=	>=			
38	3	LA		>=	=>=	>=	>=		>=	=>=			
39	5	Oakland		=>=	=>=	>=	>=	>=		>=			
40	2	Phoenix		>=	>=	=>=	>=	=>=	>=				
41	Optional Tightening constraints:												
42	28	=	(Sum of the uj = n*(n-1)/2)										
43	4	=<=	4	(Kill symmetry if distance matrix symmetric)									

The constraints in D33:J40 implement the single commodity/MTZ subtour elimination constraints.

Additional Tightening Constraints

Even though a formulation for an integer program is logically correct, solve time can sometimes be reduced by including additional tightening constraints. Here we mention three such optional constraints for the MTZ formulation of the TSP.

1) In an integer feasible solution there will be one $u_j = 1$, one $u_j = 2$, etc. Thus, the sum of the u_j will equal $(n\text{-}1)n/2$ and we are justified to add the constraint:

$$\sum_j u_j = (n\text{-}1)n/2;$$

This constraint might be otherwise violated in a continuous relaxation of the model.

2) Notice that if $y_{ji} = 1$, then $u_j \geq u_i - 1$, (it in fact holds as an =) so we can replace the constraint:

$$u_j \geq u_i + (n\text{-}1)y_{ij} - n + 2,$$

by the slightly tighter constraint:

$$u_j \geq u_i + (n\text{-}1)y_{ij} + (n\text{-}3)\, y_{ji} - n + 2;$$

3) If the distance matrix is symmetric, then a route can be traveled in reverse and still travel the same distance. Thus, we can restrict ourselves to a route in which the index of the first city visited after city 1 is less than the last city visited just before returning to 1. The following constraint is justified:

$$\sum_{j>1} j y_{1j} \leq \sum_{i>1} i\, y_{i1} - 1;$$

The additional optional tightening constraints are at the bottom of the spreadsheet formulation.

3.4.3 Subtour Elimination Cut Approach to TSP:

A large TSP may take a long time to solve with the MTZ formulation. The subtour cut approach has been successfully applied to rather larger TSP's. It is an iterative approach that adds constraints or "cuts" as needed to prevent subtours. Padberg and Rinaldi (1987) used essentially this iterative approach and were able to solve to optimality problems with over 2000 cities. The solution time was several hours on a large computer. The key idea of this approach is as follows. Given a solution to the assignment relaxation of the TSP, let S be a set of cities that constitute a subtour with $|S|$ being the number of cities in the subtour. Thus, there are $|S|$ arcs used in the subtour. We add the constraint:

$$\sum_{i,j \in S} y_{ij} \leq |S| - 1,$$

The above formulation is usually attributed to Dantzig, Fulkerson, and Johnson(1954). A point of concern about the subtour elimination cut formulation is that if there are n cities, then there are approximately 2^n different subsets for which a cut could be added. It practice it appears that only a very small fraction of these possible cuts need be added. We will illustrate with our previous example. Recall that with just the assignment relaxation of the TSP, we got a solution with length 4150 but with a subtour involving Chicago and LA. The spreadsheet Figure 3.16 shows how to add a cut that prevents a subtour involving Chicago and LA.

Figure 3.16 TSP with Subtour Elimination Cuts

I39 =WB(SUMPRODUCT(C31:J38,C$16:J$23),"<=",SUM(A31:A38)-1)

TSPSubTour1.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Traveling Salesman Problem					With subtour elimination cuts.				(TSPSubTour1.xls)			
2	We want to find a route visiting each city exactly once so as to minimize the distance traveled.												
3					Distance matrix								
4		From;	Chicago	Denver	Fresno	Houston	KC	LA	Oakland	Phoenix			
5		Chicago	0	996	2162	1067	499	2054	2134	1713			
6		Denver	996	0	1167	1019	596	1059	1227	792			
7		Fresno	2162	1167	0	1747	1723	214	168	598			
8		Houston	1067	1019	1747	0	710	1538	1904	1149			
9		KC	499	596	1723	710	0	1589	1827	1214			
10		LA	2054	1059	214	1538	1589	0	371	389			
11		Oakland	2134	1227	168	1904	1827	371	0	755			
12		Phoenix	1713	792	598	1149	1214	389	755	0			
13													
14					"Links used" matrix. A "1" means go from row city to colum city.								Must exit
15			Chicago	Denver	Fresno	Houston	KC	LA	Oakland	Phoenix		Sum	constraint
16		Chicago	0	0	0	0	1	0	0	0		1	=
17		Denver	0	0	0	1	0	0	0	0		1	=
18		Fresno	0	0	0	0	0	0	1	0		1	=
19		Houston	1	0	0	0	0	0	0	0		1	=
20		KC	0	1	0	0	0	0	0	0		1	=
21		LA	0	0	0	0	0	0	0	1		1	=
22		Oakland	0	0	1	0	0	0	0	0		1	=
23		Phoenix	0	0	0	0	0	1	0	0		1	=
24													
25		Sum:	1	1	1	1	1	1	1	1		Total dist.	
26		Must enter:	=	=	=	=	=	=	=	=		4295	(Minimize)
27													
28		Subtour elimination cut:			1								
29		Subtour:	1				1						
30			Chicago	Denver	Fresno	Houston	KC	LA	Oakland	Phoenix			
31	1	Chicago	1	0	0	0	1	0	0	0			
32		Denver	0	0	0	0	0	0	0	0			
33		Fresno	0	0	0	0	0	0	0	0			
34		Houston	0	0	0	0	0	0	0	0			
35	1	KC	1	0	0	0	1	0	0	0			
36		LA	0	0	0	0	0	0	0	0			
37		Oakland	0	0	0	0	0	0	0	0			
38		Phoenix	0	0	0	0	0	0	0	0			
39						The subtour eliminatio[illegible]t:			=<=				

Properly arranged, it is quite easy to add subtour cuts in a spreadsheet. The cities involved in the cut are marked with a 1 in C29:J29 an A31:A38. The matrix in C31:J38 contains a 1 for every link between two cities in the subtour. The constraint in I39 cuts of the subtour. The key formulae are:

```
G35=$A35*G$29
I39=WB(SUMPRODUCT(C31:J38,C$16:J$23),"<=",SUM(A31:A38)-1)
```

Notice that when this cut is added, the objective value increases to 4295 from 4150. Subtours still remain, however. Additional cuts can be added by simply copying down the range A28:J39 and marking the new subtour cities in the copied version of row 29. The half dozen additional cuts required are summarized below.

Iteration	Objective	Subtour found
0	4150	Chicago, KC
1	4295	Fresno, Oakland
2	4613	Denver, Phoenix
3	4707	Fresno, LA, Oakland, Phoenix
4	4856	Fresno, LA, Oakland
5	5066	Chicago, Houston, KC
6	5309	-no subtours remaining-

3.4.5 Multi-commodity Flow Formulation:

Similar to the MTZ formulation, imagine that each city needs one unit of some commodity, but in this case the commodity is distinct to the destination city. Define:

x_{ijk} = units of commodity carried from i to j, destined for ultimate delivery to k.

If we assume that we start at city 1 and there are n cities, then we add the following constraints to the assignment formulation:

For $k = 1, 2, 3, \ldots, n$:

$\Sigma_{j>1}\, x_{1jk} = 1$; (Each unit must be shipped out of the origin.)

$\Sigma_{i\neq k}\, x_{ikk} = 1$; (Each city k must get its unit.)

For $j = 2, 3, \ldots, n,\ k = 1, 2, 3, \ldots, n,\ j \neq k$:

$\Sigma_i\, x_{ijk} = \Sigma_{t\neq j}\, x_{jtk}$ (Units entering j, but not destined for j, must depart j to some city t.)

A unit cannot return to 1, except if its final destination is 1:

$\Sigma_i \Sigma_{k>1}\, x_{i1k} = 0$,

For $i = 1, 2, \ldots, n,\ \ j = 1, 2, \ldots, n,\ \ k = 1, 2, \ldots, n,\ \ i \neq j$:

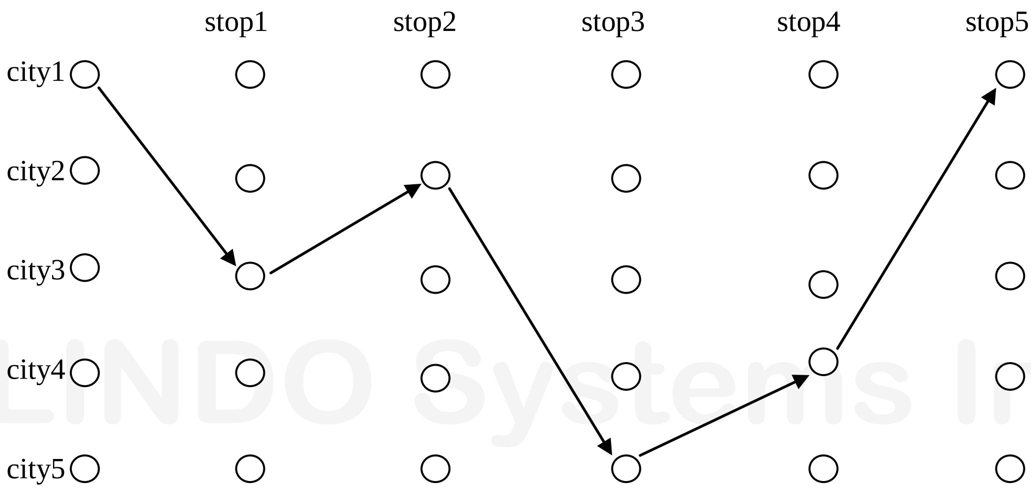
$x_{ijk} \leq y_{ij}$ (If anything shipped from i to j, then turn on y_{ij}.)

The drawback of this formulation is that it has approximately n^3 constraints and variables. A remarkable feature of the multicommodity flow formulation is that it is just as tight as the Subtour Elimination formulation. The multi-commodity formulation is due to Claus(1984).

3.4.6 Time-Space Formulation of the TSP

For some routing problems where time is an important consideration a “space-time” diagram like that in Figure 3.17 may be helpful for visualizing the problem.

Figure 3.17 Space-Time Diagram for a TSP



A path through this network is a traveling salesman tour if it makes a visit to every row of the network exactly once, except for the first row, where the path starts and ends.

Define:

w_{ijk} = 1 if we leave city i at stop k-1 and arrive at city j at stop k, else 0.

The formulation corresponding to the above graph is:

Minimize $\sum_i \sum_j \sum_k c_{ij} w_{ijk}$
subject to

We must enter each city j exactly once:
$\sum_{i \neq j} \sum_k w_{ijk} = 1$ for j = 1 to n,

We must exit each city i exactly once:
$\sum_{j \neq i} \sum_k w_{ijk} = 1$ for i = 1 to n,

We must enter exactly one city at each step k:

$\sum_{i} \sum_{j \neq i} \sum_{k} w_{ijk} = 1$ for $k = 1$ to n,

$w_{ijk} = 0$ or 1, for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, n$, $i \neq j$;

It is useful to kill some symmetry by requiring the tour start and end in city 1, so one can add the constraints:

$$\sum_{j \neq 1} w_{1j1} = 1,$$

$$\sum_{i \neq 1} w_{i1n} = 1.$$

The space/time formulation is tighter than the MTZ formulation, but not as tight as the Multi-commodity formulation.

Heuristics

For practical problems, it may be important to get good, but not necessarily optimal, answers in just a few seconds or minutes rather than hours. The most commonly used heuristic for the TSP is due to Lin and Kernighan (1973). This heuristic tries to improve a given solution by clever re-orderings of cities in the tour. For practical problems (e.g., in operation sequencing on computer controlled machines), the heuristic seems always to find solutions no more than 2% more costly than the optimum. Bland and Shallcross (1989) describe problems with up to 14,464 "cities" arising from the sequencing of operations on a computer controlled machine. In no case was the Lin-Kernighan heuristic more than 1.7% from the optimal for these problems.

3.5 Capacitated Multiple TSP/Vehicle Routing Problems

An important practical problem is the routing of vehicles from a central depot. An example is the routing of delivery trucks for the home delivery portion of an overnight package delivery service such as UPS, or for a metropolitan newspaper. You can think of this as a multiple traveling salesperson problem with finite capacity for each salesperson. This problem is sometimes called the LTL(Less than TruckLoad) routing problem because a typical recipient receives less than a truck load of goods. A formulation is:

Given:

V = capacity of a vehicle
d_j = demand of city or stop j

Define the variables:

$y_{ij} = 1$ if a vehicle travels from city i to city j, else 0.

A solution must satisfy not only the assignment-like constraints:

Each city, j, must be visited once for $j > 1$:

$$\sum_{j} x_{ij} = 1$$

Each city $i > 1$, must be exited once:

$$\sum_i x_{ij} = 1$$

but additionally, a) there must be no subtours excluding city 1 (the depot or hub), and b) the demand of the cities on any trip (or valid subtour) cannot exceed the vehicle capacity V. We give a compact formulation of this problem by generalizing the Miller-Tucker-Zemlin TSP formulation. Define:

U_j = cumulative deliveries made by the vehicle after stopping at city j.

We have a complete formulation if we add the constraints:

Each city, $j > 1$:

$$d_j \leq U_j \leq V;$$

For every combination $i \neq j, j > 1$:

$$U_j \geq U_i + d_j - V(1 - y_{ij}),$$

or equivalently:

$$U_j - U_i - V^*y_{ij} - d_j + V \geq 0,$$

If $y_{ij} =1$, this constraint implies $U_j \geq U_i + d_j$.

If $y_{ij} = 0$, it implies the redundant constraint: $U_j \geq U_i - V + d_j$.

These constraints prohibit subtours that do not include the hub city 1 by the following reasoning. Suppose there is a subtour excluding 1. The constraint set implies that as one traces around the subtour, the U_j must be strictly increasing. (We assume $d_j > 0$). This, however, leads to a contradiction.

This formulation can solve to optimality modest-sized problems of say, 25 cities. For larger or more complicated practical problems, the heuristic method of Clarke and Wright (1964) is a standard starting point for quickly finding good, but not necessarily optimal, solutions. In Figure 3.16 is a generic What's*Best*! model for vehicle routing problems. The constraints that enforce truck capacity appear in Figure 3.18. One can make the constraint a little tighter by observing that if $y_{ji} = 1$, then $U_j - U_i = - d_i$. Thus, one can extend the constraint to:

$$U_j - U_i - V^*y_{ij} - d_j + V - (V+d_i - d_j)^*y_{ji} \geq 0;$$

Figure 3.18 Capacitated Vehicle Routing

N38 =SUMPRODUCT(C26:L35,C11:L20)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Vehicle Routing, or LTL Delivery Model									(VehicleRouting.xls)						
2	The first city is the Depot city. Vehicles, each of the same capacity, make deliveries to the other cities.															
3	Each city other than the first has a demand, e.g., number of pallets, to be delivered.															
4	We want to find routes for the vehicles so as to minimize the distance traveled.															
5	Warning: Solve time may be large if number of cities > 12.															
6	Keywords: vehicle routing, routing, LTL delivery, traveling salesman;															
7																
8	Vehicle		Distance matrix need not be symmetric.													
9	capacity				Distance matrix											
10	18	From;	Chicago	Denver	Fresno	Houstn	KCity	LA	Oaklnd	Anahm	Peoria	Phoenx				
11	Demand	Chicago	0	996	2162	1067	499	2054	2134	2050	151	1713				
12	6	Denver	996	0	1167	1019	596	1059	1227	1055	904	792				
13	3	Fresno	2162	1167	0	1747	1723	214	168	250	2070	598				
14	7	Houstn	1067	1019	1747	0	710	1538	1904	1528	948	1149				
15	7	KCity	499	596	1723	710	0	1589	1827	1579	354	1214				
16	18	LA	2054	1059	214	1538	1589	0	371	36	1943	389				
17	4	Oaklnd	2134	1227	168	1904	1827	371	0	407	2043	755				
18	5	Anahm	2050	1055	250	1528	1579	36	407	0	1933	379				
19	2	Peoria	151	904	2070	948	354	1943	2043	1933	0	1568				
20	6	Phoenx	1713	792	598	1149	1214	389	755	379	1568	0				
21		Transposed demand:		6	3	7	7	18	4	5	2	6				
22	(Upper bnd)															
23	Cumulative													Number	Must exit	
24	load at city				Links used matrix. A "1" means go from row city to column city.									departing	constraint	
25			Chicago	Denver	Fresno	Houstn	KCity	LA	Oaklnd	Anahm	Peoria	Phoenx				
26	0	Chicago	0	1	0	0	0	1	1	0	1	0		4		Notice
27	6	Denver	0	0	0	1	0	0	0	0	0	0		1	=	diagonals
28	7	Fresno	0	0	0	0	0	0	0	1	0	0		1	=	are
29	13	Houstn	1	0	0	0	0	0	0	0	0	0		1	=	fixed = 0.
30	9	KCity	1	0	0	0	0	0	0	0	0	0		1	=	
31	18	LA	1	0	0	0	0	0	0	0	0	0		1	=	
32	4	Oaklnd	0	0	1	0	0	0	0	0	0	0		1	=	
33	12	Anahm	0	0	0	0	0	0	0	0	0	1		1	=	
34	2	Peoria	0	0	0	0	1	0	0	0	0	0		1	=	
35	18	Phoenx	1	0	0	0	0	0	0	0	0	0		1	=	
36																
37		Number entering:	4	1	1	1	1	1	1	1	1	1		Total dist.		
38		Must enter:		=	=	=	=	=	=	=	=	=		12838		

An optimal solution of distance 12838 is displayed in Figure 3.18. Starting with the row corresponding to Chicago, we can trace out the trips as:

1) Chicago, Denver, Houston, Chicago.

2) Chicago, LA, Chicago.

3) Chicago, Oakland, Fresno, Anaheim, Phoenix, Chicago.

4) Chicago, Peoria, K City, Chicago.

Figure 3.19 Capacitated Vehicle Routing Constraints

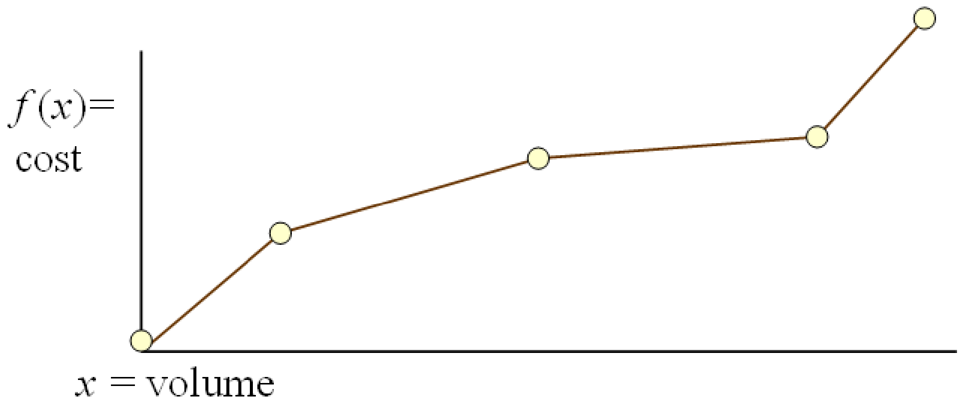
E41 =SUMIF($B26:$B35,E10,$A26:$A35)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Vehicle	Routing, or LTL Delivery Model								(VehicleRouting.xls)				
39														
40		**Computations related to computing cumulative deliveries at each city.**												12838.09
41		Transposed load:		6	7	13	9	18	4	12	2	18		
42		Capacity constraint:		<=	<=	<=	<=	=<=	<=	<=	<=	=<=		
43		Compute: load(j) - load(i) + VehCap*(1-X(i,j)) - Demand(j);												
44			Chicago	Denver	Fresno	Houstn	KCity	LA	Oaklnd	Anahm	Peoria	Phoenx		
45		Chicago		0	22	24	20	0	0	25	0	30		
46		Denver		12	16	0	14	12	12	19	12	24		
47		Fresno		11	15	17	13	11	11	0	11	23		
48		Houstn		5	9	11	7	5	5	12	5	17		
49		KCity		9	13	15	11	9	9	16	9	21		
50		LA		0	4	6	2	0	0	7	0	12		
51		Oaklnd		14	0	20	16	14	14	21	14	26		
52		Anahm		6	10	12	8	6	6	13	6	0		
53		Peoria		16	20	22	0	16	16	23	16	28		
54		Phoenx		0	4	6	2	0	0	7	0	12		
55														
56		**Constraints that force the cumulative load at j to be large enough.**												
57		**Above expression must be >= 0**												
58			Chicago	Denver	Fresno	Houstn	KCity	LA	Oaklnd	Anahm	Peoria	Phoenx		
59		Chicago		=>=	>=	>=	>=	=>=	=>=	>=	=>=	>=		
60		Denver		>=	>=	=>=	>=	>=	>=	>=	>=	>=		
61		Fresno		>=	>=	>=	>=	>=	>=	=>=	>=	>=		
62		Houstn		>=	>=	>=	>=	>=	>=	>=	>=	>=		
63		KCity		>=	>=	>=	>=	>=	>=	>=	>=	>=		
64		LA		=>=	>=	>=	>=	=>=	=>=	>=	=>=	>=		
65		Oaklnd		>=	=>=	>=	>=	>=	>=	>=	>=	>=		
66		Anahm		>=	>=	>=	>=	>=	>=	>=	>=	=>=		
67		Peoria		>=	>=	>=	=>=	>=	>=	>=	>=	>=		
68		Phoenx		=>=	>=	>=	>=	=>=	=>=	>=	=>=	>=		
69														

3.6 Representing Piecewise Linear Functions

There are many applications where the cost function is nonlinear but does not have a standard simple mathematical description. Piecewise linear functions may be useful in such cases. For example, if you ask a vendor to provide a quote for selling you some quantity of material, the vendor will typically give you a discount if you buy a large quantity. Piecewise linear functions are found not only in purchasing but also are frequently used in the modeling of energy conversion processes such as the generation of electricity. The amount of electrical energy produced by a hydro-electric or fossil fuel burning generator may be a nonlinear function of the input volume of water or fuel. Flow through a pipe, in pipeline network, as a function of pressure, is sometimes represented by a piecewise linear function. Several different possible cost curves are shown below.

Figure 3.20 An Arbitrary Continuous Piecewise Linear Cost



Vendors typically offer two general forms: a) Incremental discounts, or b) All units discount. In "all units" the discount given if you order more than a specified threshold, say 1000, applies to all units purchased, whereas, with incremental units discount, the discount applies only to the units in excess of 1000. This leads to two different cost curves.

Figure 3.21 An All-Units Discount Cost Curve

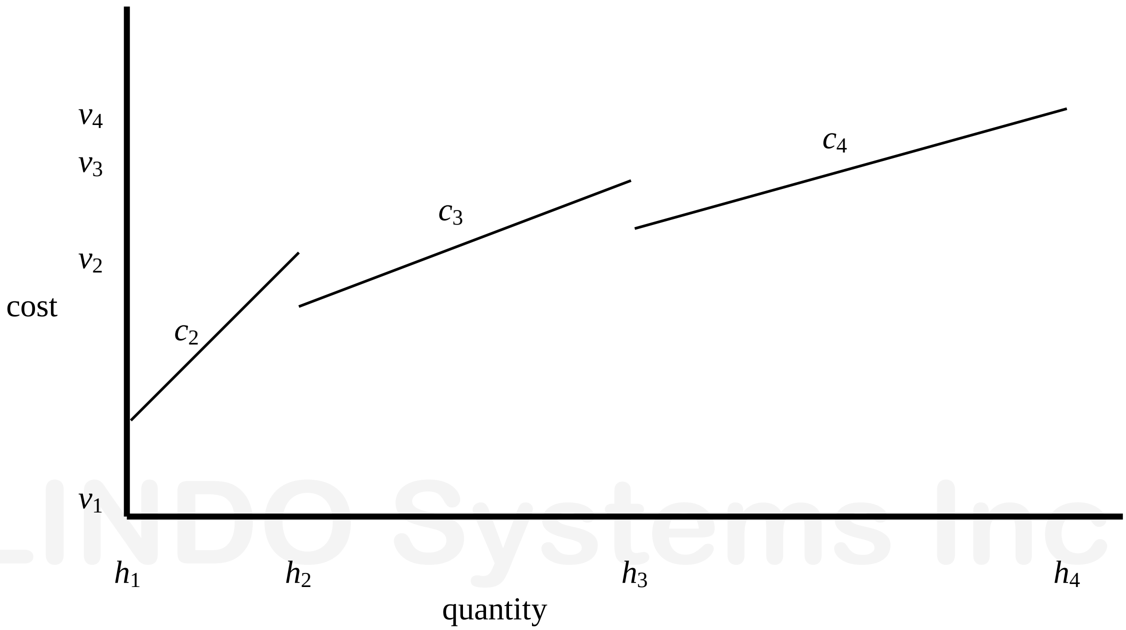
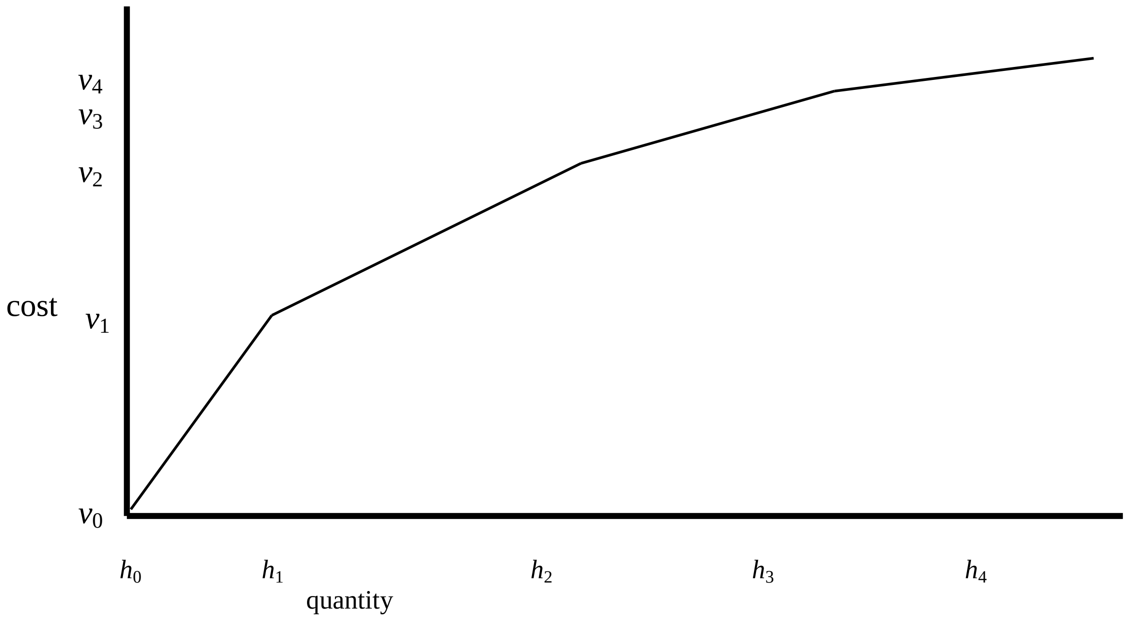


Figure 3.22 An Incremental Units Discount Cost Curve



There are several ways of representing a piecewise linear cost curve. In the general case, some form of linear integer programming is required. The approach we will describe is based on the observation that any point on the curve can be represented as a weighted combination of the two breakpoints that bracket the point.

Define the variables:

w_i = nonnegative weight to be applied to point i, for i = 0, 1, 2, 3, 4.
x = amount purchased,
cost = total cost of the purchase.

We can cause x and *cost* to almost be calculated correctly by writing the constraints:

$$x = w_0h_0 + w_1h_1 + w_2h_2 + w_3h_3 + w_4h_4;$$
$$\text{cost} = w_0v_0 + w_1v_1 + w_2v_2 + w_3v_3 + w_4v_4;$$
$$1 = w_0 + w_1 + w_2 + w_3 + w_4v_4;$$

This method is sometimes called the lambda method because the Greek symbol lambda was used sometimes to represent the weights.

Any point on the line segment connecting the two points (h_i, v_i) and (h_{i+1}, v_{i+1}) can be represented by choosing appropriate values for w_i and w_{i+1} so that $w_i + w_{i+1} = 1$, and $w_i, w_{i+1} \geq 0$. To ensure that the point corresponding to a particular set of values for the w_i lies on the curve, we need to require that if two or more of the w_i are > 0, they must be adjacent. We said "almost" in the earlier sentence because there is nothing in the three constraints above that enforce this adjacency condition. There are two ways of enforcing this adjacency condition: a) declare the w_i to be members of an SOS2 set in What's*Best*!, or b) add a number of binary variables to enforce the condition. If a set of variables have a constraint that at most two can be nonzero, and they must be adjacent, then the set is called a "Special Ordered Set" of type 2, or SOS2 for short.

We will illustrate the SOS2 approach in What's*Best*!. A vendor price schedule states that the first 1,000 liters of the product can be purchased for $2 per liter. The price drops to $1.90 per liter for units beyond 1000, $1.80 for units above 3500, and $1.75 for units beyond 5000. We would never purchase more than 7000 units. The vendor gives incremental, rather than all units discouts. The total cost schedule is easily seen to be:

```
Quantity    Total cost
   0            0
 1000         2000      (2.00*1000)
 3500         6750      (2000 + 1.9*(3500-1000))
 5000         9450      (6750 + 1.8*(5000-3500))
 7000        12950      (9450 + 1.75*(7000-5000)
```


The associated equations are:

```
   x = w0*0 + w1*1000 + w2*3500 + w3*5000 + w4* 7000;
cost = w0*0 + w1*2000 + w2*6750 + w3*9450 + w4*12950;
   1 = w0   + w1      + w2      + w3      + w4;
```

Suppose we have only the above constraints, arbitrarily add the constraint, x = 4300, and minimize the cost. We get the solution:

```
Variable        Value
       X      4300.00
    COST      7955.00
      W0     0.428571
      W1     0.000000
      W2     0.000000
      W3     0.000000
      W4     0.571429
```

The cost is wrong. It should be 6750 + 1.8*(4300-3500) = 8190. The problem is that the two nonzero weights, w0 and w4, are not adjacent. If the SOS2 feature is turned on, we get the desired result:

```
Variable        Value
       X      4300.00
    COST      8190.00
      W0     0.000000
      W1     0.000000
      W2     0.466667
      W3     0.533333
      W4     0.000000
```

If for some reason you do not want to use the SOS2 feature, you can introduce 4 binary variables:

y_i = 1 if x is in the interval with endpoints h_{i-1} and h_i, for i = 1, 2, 3, 4. We would replace the SOS2 declarations by the constraints: that the y_i must be 0 or 1 and:

```
w0 <= y1;
w1 <= y1 + y2;
w2 <= y2 + y3;
w3 <= y3 + y4;
w4 <= y4;
```

Piecewise Linear Cost Curve, Two Vendor Example

Now suppose a second vendor appears at our door and offers the following price schedule for the same product. Any quantity < 600 costs $1.96 per liter. Any quantity of 600 or more costs $1.79. Further, this is an "all units" discount, applying to all units purchased. One complication with the second vendor is that a $500 shipping and handling charge is applied to any order. How much should be purchased from each vendor? A spreadsheet answering this question is displayed in Figure 3.23.

Figure 3.23 Purchasing from Two Suppliers with Nonlinear Price Schedules

D21 =WBSOS2(D15:D19)

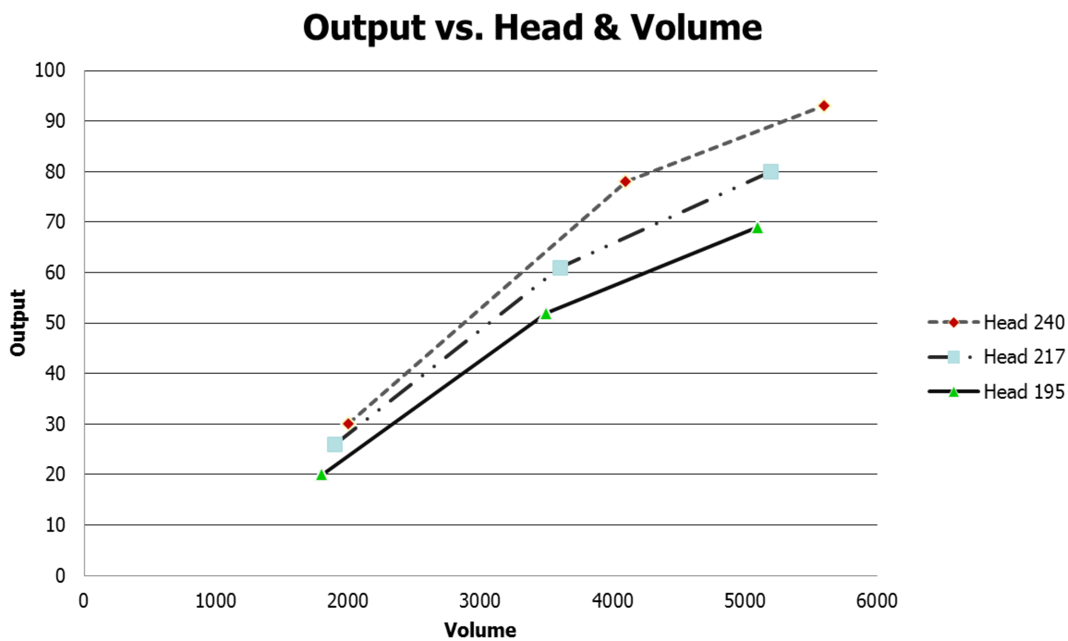
PieceWiseBuy2.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H
1	**Purchasing with a Nonlinear Price Schedule.**						(PieceWiseBuy2.xls)	
2	We need to buy a specified amount of commodity.							
3	Two vendors each offer their own price curve (piecewise linear).							
4	How much should we buy from each to minimize total cost?							
5	We use the SOS2 capability of What's*Best* ! to represent piecewise linear cost curves.							
6								
7	Units needed:	4300						
8		=	<<== Constraint, bought = needed.					
9	Total bought:	4300	8190	: Total cost (minimize)				
10								
11			Vendor 1				Vendor 2	
12		Amt to buy	Cost	Sum to 1		Amt to buy	Cost	Sum to 1
13		4300	8190	=		0	0	=
14		Vol Pt	Tcost Pt	Pt wgt		Vol Pt	Tcost Pt	Pt wgt
15		0	0	0		0	0	1
16		1000	2000	0		1	501.96	0
17		3500	6750	0.46667		599	1674.04	0
18		5000	9450	0.53333		600	1574	0
19		7000	12950	0		7000	13030	0
20								
21	Data			WBSOS2	<<== Two weights adjacent ==>>			WBSOS2
22		Click on What's*Best* ! | Integers | Special Ordered Set | Type 2						

3.6.1 Piecewise Linear Approximations to Multivariate Functions

Can we extend the piecewise linear interpolation method to functions of 2 variables? An example might be the power output from a hydro generator as a function of the two variables: 1) head or pressure, and 2) flow volume.

Figure 3.24 Piecewise Linear Function of Two Variables



It is an interesting challenge to figure out how to use the SOS2 constraint type to represent such a function.

3.7 Assignment Constraints & AllDifferent Constraints

Many problems can be thought of as requiring the assignment of a unique integer to each of a set of tasks or objects, e.g.,

Task	Label/Sequence/Position
1	**4**
2	**1**
3	**3**
4	**5**
5	**2**

It is useful to think of this as an Assignment problem, where
y_{ij} = 1 if task i is assigned label/position j, else 0.
$p_i = \sum_j j^* y_{ij}$ = position number assigned to i.

Constraint programming languages allow you to directly specify an **"AllDiff"** constraint on label numbers.

3.7.1 Assignment Constraints, *AllDifferent*, Example

A parent of a college student tells the student he will answer his latest request if the student can solve the following puzzle:

`SEND`	
`+ MORE`	Find values in [0, 9] for S, E, N, D, M, O, R, Y
`MONEY`	so the addition on the left makes sense.

Mathematically we want to satisfy the constraint:

$$1000^*S + 100^*E + 10^*N + D$$
$$+ 1000^*M + 100^*O + 10^* R + E$$
$$= 10000^*M + 1000^*O + 100^*N + 10^*E + Y;$$

In words, we want the variables S, E, N, D, M, O, R, Y to be integers in [0, 9]. A not so easy constraint is that the values must all be different, e.g., $S \neq E$, etc. Also, the leading digits, S and $M \neq 0$.)

Remembering our old friend, or at least acquaintance, the Assignment problem, helps.

3.7.2 *AllDifferent* Formulated as Assignment Constraints

Assignment formulation:
y_{ij} = 1 if letter i is assigned value j.

The constraints are:
Each letter gets a value:

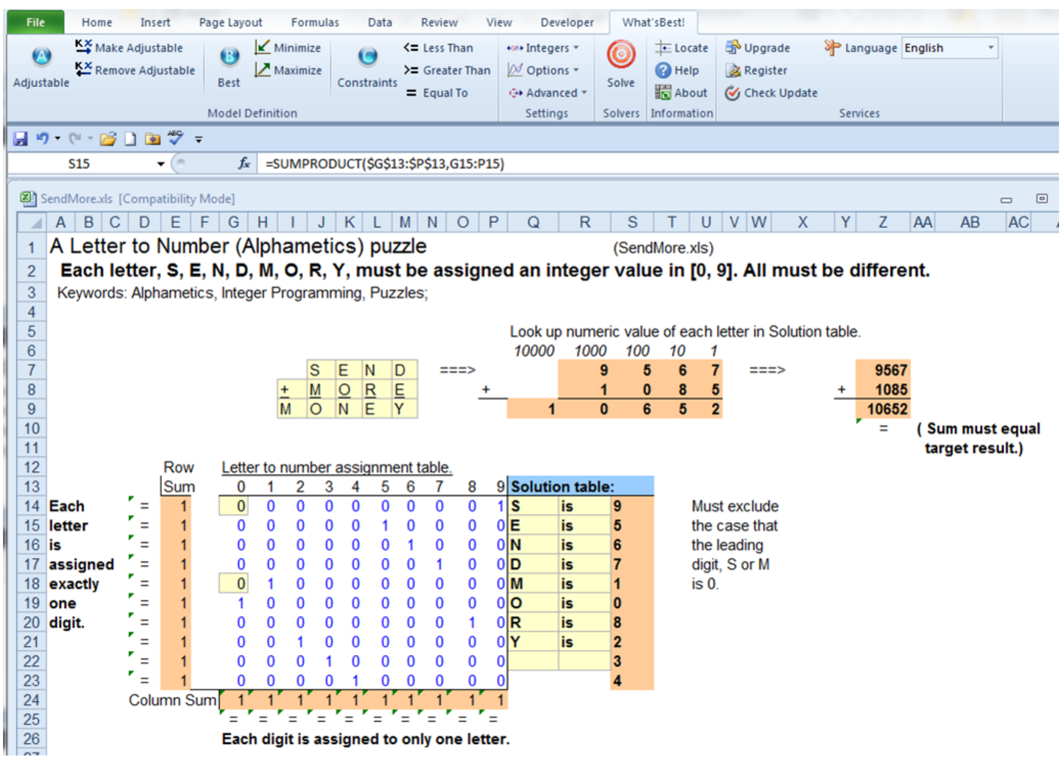
$yS0 + yS1 + yS2 + yS3 + yS4 + yS5 + yS6 + yS7 + yS8 + yS9 = 1;$
$yE0 + yE1 + yE2 + yE3 + yE4 + yE5 + yE6 + yE7 + yE8 + yE9 = 1;$ etc.

Each value can be used at most once (the Alldifferent part):
$yS0 + yE0 + yN0 + yD0 + yM0 + yO0 + yR0 + yY0 \le 1;$
$yS1 + yE1 + yN1 + yD1 + yM1 + yO1 + yR1 + yY1 \le 1;$ etc.

Connect the Assignment view to the Position value view:
$S = yS1 + 2yS2 + 3yS3 + 4yS4 + 5yS5 + 6yS6 + 7yS7 + 8yS8 + 9yS9;$
$E = yE1 + 2yE2 + 3yE3 + 4yE4 + 5yE5 + 6yE6 + 7yE7 + 8yE8 + 9yE9;$
etc.

The spreadsheet incarnation of this formulation appears in Figure 3.23.

Figure 3.23 Puzzle Example of Assignment Form of AllDiff Constraints.



Test your understanding and skill with a slightly bigger puzzle:

```
    PLUTO
   SATURN
   URANUS
+ NEPTUNE
  -------
  PLANETS
```

Find values in [0, 9] for
P, L, U, T, O, S, A, R, N, E
so the addition on the left makes sense.

3.8 Outline of Integer Programming Methods

The time a computer requires to solve an IP may depend dramatically on how you formulated it. It is, therefore, worthwhile to know a little about how IPs are solved. There are two general approaches for solving IPs: "cutting plane" methods and "branch-and-bound" (B & B) method. For a comprehensive introduction to integer programming solution methods, see Nemhauser and Wolsey (1988), and Wolsey (1998). Most commercial IP programs use the B & B method, but aided by some cutting plane features. Fortunately for the reader, the B & B method is the easier to describe. In most general terms, B & B is a form of intelligent enumeration.

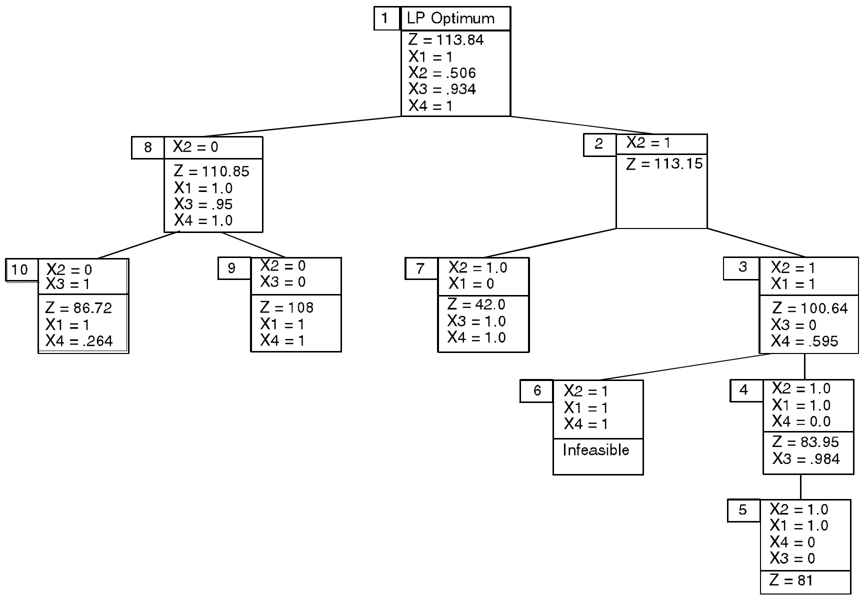
More specifically, B & B first solves the problem as an LP. If the LP solution is integer valued in the integer variables, then no more work is required. Otherwise, B & B resorts to an intelligent search of all possible ways of rounding the fractional variables.

We shall illustrate the application of the branch-and-bound method with the following problem:

```
MAX= 75 * X1  + 6 * X2   + 3 * X3 + 33 * X4;
    774 * X1 + 76 * X2  + 22 * X3 + 42 * X4 <= 875;
     67 * X1 + 27 * X2 + 794 * X3 + 53 * X4 <= 875;
    @BIN( X1); @BIN( X2); @BIN( X3); @BIN( X4);
```


The search process a computer might follow in finding an integer optimum is illustrated in Figure 3.5. First, the problem is solved as an LP with the constraints Xj = 0 or 1 replaced by the relaxations $0 \leq Xj \leq 1$. This solution is summarized in the box labeled 1. The solution has fractional values for *X2* and *X3* and is, therefore, unacceptable. At this point, *X2* is arbitrarily selected and the following reasoning is applied. At the integer optimum, *X2* must equal either 0 or 1.

Figure 3.5 Branch-and-Bound Search Tree



Therefore, replace the original problem by two new subproblems. One with *X2* constrained to equal 1 (box or node 2) and the other with *X2* constrained to equal 0 (node 8). If we solve both of these new IPs, then the better solution must be the best solution to the original problem. This reasoning is the motivation for using the term “branch”. Each subproblem created corresponds to a branch in an enumeration tree.

The numbers to the upper left of each node indicate the order in which the nodes (or equivalently, subproblems) are examined. The variable *Z* is the objective function value. When the subproblem with *X2* constrained to 1 (node 2) is solved as an LP, we find *X1* and *X3* take fractional values. If we argue as before, but now with variable *X1*, two new subproblems are created:

Node 7) one with *X1* constrained to 0 , and
Node 3) one with *X1* constrained to 1.

This process is repeated with *X4* and *X3* until node 5. At this point, an integer solution with *Z* = 81 is found. We do not know this is the optimum integer solution, however, because we must still look at subproblems 6 through 10. Subproblem 6 need not be pursued further because there are no feasible solutions having all of *X2*, *X1*, and *X4* equal to 1. Subproblem 7 need not be pursued further because it has a *Z* of 42, which is worse than an integer solution already in hand.

At node 9, a new and better integer solution with $Z = 108$ is found when *X3* is set to 0. Node 10 illustrates the source for the "bound" part of "branch-and-bound". The solution is fractional. However, it is not examined further because the *Z*-value of 86.72 is less than the 108 associated with an integer solution already in hand. The *Z*-value at any node is a bound on the *Z*-value at any offspring node. This is true because an offspring node or subproblem is obtained by appending a constraint to the parent problem. Appending a constraint can only hurt. Interpreted in another light, this means the *Z*-values cannot improve as one moves down the tree. The tree presented in the preceding figure was only one illustration of how the tree might be searched. Other trees could be developed for the same problem by playing with the following two degrees of freedom:

(a) Choice of next node to examine, and
(b) Choice of how the chosen node is split into two or more subnodes.

For example, if nodes 8 and then 9 were examined immediately after node 1, then the solution with $Z = 108$ would have been found quickly. Further, nodes 4, 5, and 6 could then have been skipped because the *Z*-value at node 3 (100.64) is worse than a known integer solution (108), and, therefore, no offspring of node 3 would need examination.

In the example tree, the first node is split by branching on the possible values for *X2*. One could have just as well chosen *X3* or even *X1* as the first branching variable.

The efficiency of the search is closely related to how wisely the choices are made in (a) and (b) above. Typically, in (b) the split is made by branching on a single variable. For example, if, in the continuous solution, $x = 1.6$, then the obvious split is to create two subproblems. One with the constraint $x \leq 1$, and the other with the constraint $x \geq 2$. The split need not be made on a single variable. It could be based on an arbitrary constraint. For example, the first subproblem might be based on the constraint $x_1 + x_2 + x_3 \leq 0$, while the second is obtained by appending the constraint $x_1 + x_2 + x_3 \geq 1$. Also, the split need not be binary. For example, if the model contains the constraint $y_1 + y_2 + y_3 = 1$, then one could create three subproblems corresponding to either $y_1 = 1$, or $y_2 = 1$, or $y_3 = 1$.

If the split is based on a single variable, then one wants to choose variables that are "decisive." In general, the computer will make intelligent choices and the user need not be aware of the details of the search process. The user should, however, keep the general B & B process in mind when formulating a model. If the user has *a priori* knowledge that an integer variable *x* is decisive, then for the What'sBest! program it is useful to place *x* early in the formulation to indicate its importance. This general understanding should drive home the importance of a "tight" LP formulation. A tight LP formulation is one which, when solved, has an objective function value close to the IP optimum. The LP solutions at the subproblems are used as bounds to curtail the search. If the bounds are poor, many early nodes in the tree may be explicitly examined because their bounds look good even though, in fact, these nodes have no good offspring.

3.8.1 Computational Difficulty of Integer Programs

Integer programs can be very difficult to solve. This is in marked contrast to LP problems. The solution time for an LP is fairly predictable. For an LP, the time increases approximately proportionally with the number of variables and approximately with the square of the number of constraints. For a given IP problem, the time may in fact decrease as the number of constraints is increased. As the number of integer variables is increased, the solution time may increase dramatically. Some small IPs (e.g., 6 constraints, 60 variables) are extremely difficult to solve.

Just as with LPs, there may be alternate IP formulations of a given problem. With IPs, however, the solution time generally depends critically upon the formulation. Producing a good IP formulation requires skill. For many of the problems in the remainder of this chapter, the difference between a good formulation and a poor formulation may be the difference between whether the problem is solvable or not.

3.8.2 NP-Complete Problems

Integer programs belong to a class of problems known as *NP*-hard. We may somewhat loosely think of *NP* as meaning "not polynomial". This means that there is no known algorithm of solving these problems such that the computational effort at worst increases as a polynomial in the problem size. For our purposes, we will say that the computational complexity of an algorithm is polynomial if there is a positive constant k, such that the time to solve a problem of size n is proportional to n^k. For example, sorting a set of n numbers can easily be done in (polynomial) time proportional to n^2,($n\ log(n)$ if one is careful), whereas solving an integer program in n zero/one variables may, in the worst case, take (exponential) time proportional to 2^n. There may be a faster way, but no one has published an algorithm for integer programs that is guaranteed to take polynomial time on every problem presented to it. The terms *NP*-complete and *P*-complete apply to problems that can be stated as "yes/no" or feasibility problems. The yes/no variation of an optimization problem would be a problem of the form: Is there a feasible solution to this problem with cost less-than-or-equal-to 1250. In an optimization problem, we want a feasible solution with minimum cost. Khachian (1979) showed that the feasibility version of LP is solvable in polynomial time. So, we say LP is in *P*. Integer programming stated in feasibility form, and a wide range of similar problems, belong to a class of problems called *NP*-complete. These problems have the feature that it is possible to convert any one of these problems into any other *NP*-complete problem in time that is polynomial in the problem size. Thus, if we can convert problem *A* into problem *B* in polynomial time, then solve *B* in polynomial time, and then convert the solution to *B* to a valid solution to *A* in polynomial time, we then have a way of solving *A* in polynomial time.

The notable thing about *NP*-complete problems is that, if someone develops a guaranteed fast (e.g., polynomial worst case) time method for solving one of these problems, then that someone also has a polynomial time algorithm for every other *NP*-complete problem. An important point to remember is that the *NP*-completeness classification is defined in terms of worst case behavior, not average case behavior. For practical purposes, one is interested mainly in average case behavior. The current situation is that the average time to solve many important practical integer programming problems is quite short. The fact that someone may occasionally present us with an extremely difficult integer programming problem does not prevent us from profiting from the fact that a large number of practical integer programs can be solved rapidly. Perhaps the biggest open problem in modern mathematics is whether the problems in the *NP*-complete class are inherently difficult. This question is cryptically phrased as is: $P = NP$? Are these problems really difficult, or is it that we are just not smart enough to discover the universally fast algorithm? In fact, a “Millenium prize” of $1,000,000 is offered by the Clay Mathematics Institute, www.claymath.org, for an answer to this question. For a more comprehensive discussion of the *NP*-complete classification, see Martin (1999).

3.9 Problems

1. The following problem is known as a segregated storage problem. A feed processor has various amounts of four different commodities, which must be stored in seven different silos. Each silo can contain at most one commodity. Associated with each commodity and silo combination is a loading cost. Each silo has a finite capacity, so some commodities may have to be split over several silos. For a similar problem arising in the loading of fuel tank trucks at Mobil Oil Company, see Brown, Ellis, Graves, and Ronen (1987). The following table contains the data for this problem.

Loading Cost per Ton

	Silo							**Amount of Commodity To Be Stored**
Commodity	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
A	\$1	\$2	\$2	\$3	\$4	\$5	\$5	75 tons
B	2	3	3	3	1	5	5	50 tons
C	4	4	3	2	1	5	5	25 tons
D	1	1	2	2	3	5	5	80 tons
Silo Capacity in Tons	25	25	40	60	80	100	100	

a) Present a formulation for solving this class of problems.

b) Find the minimum cost solution for this particular example.

c) How would your formulation change if additionally there was a fixed cost associated with each silo that is incurred if anything is stored in the silo?

2. You are the scheduling coordinator for a small, growing airline. You must schedule exactly one flight out of Chicago to each of the following cities: Atlanta, Los Angeles, New York, and Peoria. The available departure slots are 8 A.M., 10 A.M., and 12 noon. Your airline has only two departure lounges, so at most two flights can be scheduled per slot. Demand data suggest the following expected profit contribution per flight as a function of departure time:

Expected Profit Contribution in \$1000's

	Time		
Destination	**8**	**10**	**12**
Atlanta	10	9	8.5
Los Angeles	11	10.5	9.5
New York	17	16	15
Peoria	6.4	2.5	–1

Formulate a model for solving this problem.

3. A problem faced by an electrical utility each day is that of deciding which generators to start up at which hour based on the forecast demand for electricity each hour. This problem is also known as the unit commitment problem. The utility in question has three generators with the following characteristics:

Generator	Fixed Startup Cost	Fixed Cost per Period of Operation	Cost per Period per Megawatt Used	Maximum Capacity in Megawatts Each Period
A	3000	700	5	2100
B	2000	800	4	1800
C	1000	900	7	3000

There are two periods in a day and the number of megawatts needed in the first period is 2900. The second period requires 3900 megawatts. A generator started in the first period may be used in the second period without incurring an additional startup cost. All major generators (e.g., *A*, *B*, and *C* above) are turned off at the end of each day.

a) First, assume fixed costs are zero and thus can be disregarded. What are the decision variables?

b) Give the LP formulation for the case where fixed costs are zero.

c) Now, take into account the fixed costs. What are the additional (zero/one) variables to define?

d) What additional terms should be added to the objective function? What additional constraints should be added?

4. *Crude Integer Programming*. Recently, the U.S. Government began to sell crude oil from its Naval Petroleum Reserve in sealed bid auctions. There are typically six commodities or products to be sold in the auction, corresponding to the crude oil at the six major production and shipping points. A "bid package" from a potential buyer consists of (a) a number indicating an upper limit on how many barrels (bbl.) the buyer is willing to buy overall in this auction and (b) any number of "product bids". Each product bid consists of a product name and three numbers representing, respectively, the bid price per barrel of this product, the minimum acceptable quantity of this product at this price, and the maximum acceptable quantity of this product at this price. Not all product bids of a buyer need be successful. The government usually places an arbitrary upper limit (e.g., 20%) on the percentage of the total number of barrels over all six products one firm is allowed to purchase.

To illustrate the principal ideas, let us simplify slightly and suppose there are only two supply sources/products, which are denoted by *A* and *B*. There are 17,000 bbls. available at *A* while *B* has 13,000. Also, there are only two bidders, the Mobon and the Exxil companies. The government arbitrarily decides either one can purchase at most 65% of the total available crude. The two bid packages are as follows:

Mobon:

Maximum desired = 16,000 bbls. total.

Product	Bid per Barrel	Minimum Barrels Accepted	Maximum Barrels Wanted
A	43	9000	16,000
B	51	6000	12,000

Exxil:

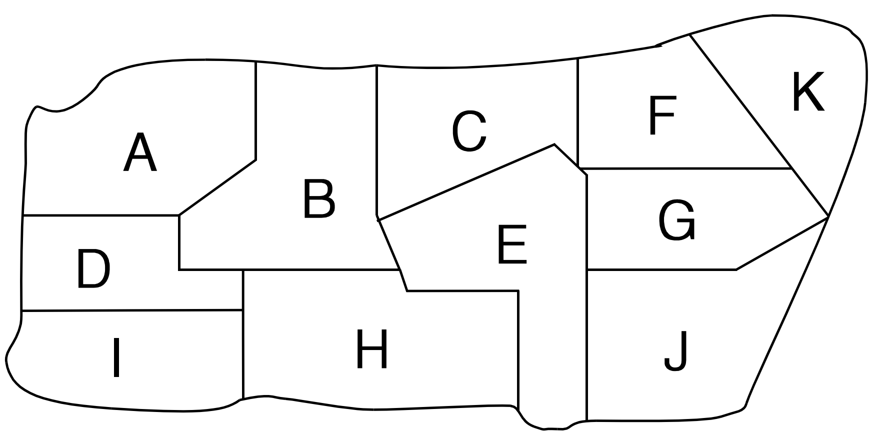
Maximum desired = No limit.

Product	Bid per Barrel	Minimum Barrels Accepted	Maximum Barrels Wanted
A	47	5000	10,000
B	50	5000	10,000

Formulate and solve an appropriate IP for the seller.

5. A certain state allows a restricted form of branch banking. Specifically, a bank can do business in county *i* if the bank has a "principal place of business" in county *i* or in a county sharing a nonzero-length border with county *i*. Figure 11.10 is a map of the state in question:

Figure 11.10 Districts in a State



Formulate the problem of locating a minimum number of principal places of business in the state, so a bank can do business in every county in the state. If the problem is formulated as a covering problem, how many rows and columns will it have? What is an optimal solution? Which formulation is tighter: set covering or simple plant location?

6. *Data Set Allocation Problem*. There are 10 datasets or files, each of which is to be allocated to 1 of 3 identical disk storage devices. A disk storage device has 885 cylinders of capacity. Within a storage device, a dataset will be assigned to a contiguous set of cylinders. Dataset sizes and interactions between datasets are shown in the table below. Two datasets with high interaction rates should not be assigned to the same device. For example, if datasets *C* and *E* are assigned to the same disk, then an interaction cost of 46 is incurred. If they are assigned to different disks, there is no interaction cost between *C* and *E*.

Dataset for Interaction (Seek Transition) Rates

	A	B	C	D	E	F	G	H	I	J	Dataset Size in Cylinders
A											110
B	43										238
C	120	10									425
D	57	111	188								338
E	96	78	46	88							55
F	83	58	421	60	63						391
G	77	198	207	109	73	74					267
H	31	50	43	47	51	21	88				105
I	38	69	55	21	36	391	47	96			256
J	212	91	84	53	71	40	37	35	221		64
											2249

Find an assignment of datasets to disks, so total interaction cost is minimized and no disk capacity is exceeded.

7. The game or puzzle of mastermind pits two players, a “coder” and a “decoder”, against each other. The game is played with a pegboard and a large number of colored pegs. The pegboard has an array of 4 × 12 holes. For our purposes, we assume there are only six colors: red, blue, clear, purple, gold, and green. Each peg has only one color. The coder starts the game by selecting four pegs and arranging them in a fixed order, all out of sight of the decoder. This ordering remains fixed throughout the game and is appropriately called the code. At each play of the game, the decoder tries to match the coder’s ordering by placing four pegs in a row on the board. The coder then provides two pieces of information about how close the decoder’s latest guess is to the coder’s order:

    1) The number of pegs in the correct position (i.e., color matching the coder’s peg in that position), and
    2) The maximum number of pegs that would be in correct position if the decoder were allowed to permute the ordering of the decoder’s latest guess.

    Call these two numbers *m* and *n*. The object of the decoder is to discover the code in a minimum number of plays.

    The decoder may find the following IP of interest.

```
MAX  = XRED1;
  XRED1 + XBLUE1 + XCLEAR1 + XPURP1 + XGOLD1
      + XGREEN1 = 1;
  XRED2 + XBLUE2 + XCLEAR2 + XPURP2 + XGOLD2
      + XGREEN2 = 1;
  XRED3 + XBLUE3 + XCLEAR3 + XPURP3 + XGOLD3
      + XGREEN3 = 1;
  XRED4 + XBLUE4 + XCLEAR4 + XPURP4 + XGOLD4
      + XGREEN4 = 1;
 XRED1 + XRED2  + XRED3 + XRED4 - RED = 0;
 XBLUE1 + XBLUE2  + XBLUE3 + XBLUE4 - BLUE = 0;
 XCLEAR1 + XCLEAR2 + XCLEAR3 + XCLEAR4 - CLEAR = 0;
 XPURP1 + XPURP2 + XPURP3 + XPURP4 - PURP = 0;
 XGOLD1 + XGOLD2 + XGOLD3 + XGOLD4 - GOLD = 0;
 XGREEN1 + XGREEN2 + XGREEN3 + XGREEN4 - GREEN = 0;
END
```

    All variables are required to be integer. The interpretation of the variables is as follows. *XRED1* = 1 if a red peg is in position 1, otherwise 0, etc.; *XGREEN4* = 1 if a green peg is in position 4, otherwise 0. Rows 2 through 5 enforce the requirement that exactly one peg be placed in each position. Rows 6 through 11 are simply accounting constraints, which count the number of pegs of each color. For example, *RED* = the number of red pegs in any position 1 through 4. The objective is unimportant. All variables are (implicitly) required to be nonnegative.

    At each play of the game, the decoder can add new constraints to this IP to record the information gained. Any feasible solution to the current formulation is a reasonable guess for the next play. An interesting question is what constraints can be added at each play.

To illustrate, suppose the decoder guesses the solution *XBLUE1* = *XBLUE2* = *XBLUE3* = *XRED4* = 1, and the coder responds with the information that $m = 1$ and $m - n = 1$. That is, one peg is in the correct position and, if permutations were allowed, at most two pegs would be in the correct position. What constraints can be added to the IP to incorporate the new information?

8. The Mathematical Football League (MFL) is composed of M teams (M is even). In a season of $2(M - 1)$ consecutive Sundays, each team will play $(2M - 1)$ games. Each team must play each other team twice, once at home and once at the other team's home stadium. Each Sunday, k games from the MFL are televised. We are given a matrix $\{v_{ij}\}$ where v_{ij} is the viewing audience on a given Sunday if a game between teams i and j playing at team j's stadium is televised.

   a) Formulate a model for generating a schedule for the MFL that maximizes the viewing audience over the entire season. Assume viewing audiences are additive.

   b) Are some values of k easier to accommodate than others? How?

9. The typical automobile has close to two dozen electric motors. However, if you examine these motors, you will see that only about a half dozen distinct motor types are used. For inventory and maintenance reasons, the automobile manufacturer would like to use as few distinct types as possible. For cost, quality, and weight reasons, one would like to use as many distinct motor types as possible, so the most appropriate motor can be applied to each application. The table below describes the design possibilities for a certain automobile:

24-Month Failure Probability

	Number Required	**Motor type**				
Application		**A**	**B**	**C**	**D**	**E**
Head lamps	2	0.002	0.01		0.01	0.007
Radiator fan	2		0.01	0.002		0.004
Wipers	2				0.007	
Seat	4	0.003			0.006	0.008
Mirrors	2			0.004	0.001	
Heater fan	1		0.006	0.001		
Sun roof	1	0.002			0.003	0.009
Windows	4	0.004	0.008	0.005		
Antenna	1	0.003		0.003	0.002	
	Weight	2	3	1.5	1	4
	Cost per Motor	24	20	36	28	39

For example, two motors are required to operate the headlamps. If type *D* motors are used for headlamps, then the estimated probability of a headlamp motor failure in two years is about 0.01. If no entry appears for a particular combination of motor type and application, it means the motor type is inappropriate for that application (e.g., because of size).

Formulate a solvable linear integer program for deciding which motor type to use for each application, so at most 3 motor types are used, the total weight of the motors used is at most 36, total cost of motors used is at most 585, and probability of any failure in two years is approximately minimized.

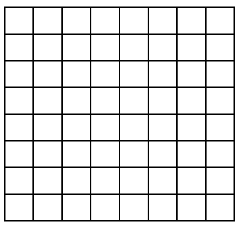
10. We have a rectangular three-dimensional container that is $30 \times 50 \times 50$. We want to pack in it rectangular three-dimensional boxes of the three different sizes: (a) $5 \times 5 \times 10$, (b) $5 \times 10 \times 10$, and (c) $5 \times 15 \times 25$.

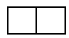
    A particular packing of boxes into the container is undominated if there is no other packing that contains at least as many of each of the three box types and strictly more of one of the box types.

    Show there are no more than 3101 undominated packings.

11. Given the following:

    Checkerboard and *domino*

    If two opposite corners of the checkerboard are made unavailable, prove there is no way of exactly covering the remaining grid with 31 dominoes.

12. Which of the following requirements could be represented exactly with linear constraints? (You are allowed to use transformations if you wish.)

    (a) $(3 \times x + 4 \times y)/(2 \times x + 3 \times y) \leq 12$;
    (b) MAX $(x, y) < 8$;
    (c) $3 \times x + 4 \times y \times y \geq 11$; where y is 0 or 1;
    (d) ABS $(10 - x) \leq 7$ (Note ABS means absolute value);
    (e) MIN $(x, y) < 12$.

13. A common way of controlling access in many systems, such as information systems or the military, is with priority levels. Each user i is assigned a clearance level U_i. Each object j is assigned a security level L_j. A user i does not have access to object j if the security level of j is higher than the clearance level of i. Given a set of users; and, for each user, a list of objects to which that user does not to have access; and a list of objects to which the user should have access, can we assign U_i's and L_j's, so these access rights and denials are satisfied? Formulate as an integer program.

14. One of the big consumption items in the U.S. is automotive fuel. Any petroleum distributor who can deliver this fuel reliably and efficiently to the hundreds of filling stations in a typical distribution region has a competitive advantage. This distribution problem is complicated by the fact that a typical customer (i.e., filling station) requires three major products: premium gasoline, an intermediate octane grade (e.g., "Silver"), and regular gasoline. A typical situation is described below. A delivery tank truck has four compartments with capacities in liters of 13,600, 11,200, 10,800, and 4400. We would like to load the truck according to the following limits:

	Liters of		
	Premium	**Intermediate**	**Regular**
At least:	8,800	12,000	12,800
At most:	13,200	17,200	16,400

Only one gasoline type can be stored per compartment in the delivery vehicle. Subject to the previous considerations, we would like to maximize the amount of fuel loaded on the truck.

(a) Define the decision variables you would use in formulating this problem as an IP.
(b) Give a formulation of this problem.
(c) What allocation do you recommend?

15. Most lotteries are of the form:

Choose n numbers (e.g., n = 6) from the set of numbers {1, 2, ..., m} (e.g., m = 54).

You win the grand prize if you buy a ticket and choose a set of n numbers identical to the n numbers eventually chosen by lottery management. Smaller prizes are awarded to people who match k of the n numbers. For n = 6, typical values for k are 4 and 5. Consider a modest little lottery with m = 7, n = 3, and k = 2. How many tickets would you have to buy to guarantee winning a prize? Can you set this up as a grouping/covering problem?

16. A recent marketing phenomenon is the apparent oxymoron, "mass customization". The basic idea is to allow each customer to design his/her own product, and yet do it on an efficient, high-volume scale. A crucial component of the process is to automate the final design step involving the customer. As an example, IBM and Blockbuster recently announced a plan to offer "on-demand" production of customized music products at retail stores. Each store would carry an electronic "master" for every music piece a customer might want. The physical copy for the customer would then be produced for the customer while they wait. This opens up all manners of opportunities for highly customized musical products. Each customer might provide a list of songs to be placed on an audiocassette. A design issue when placing songs on a two-sided medium such as a cassette is how to allocate songs to sides. A reasonable rule is to distribute the songs, so the playing times on the two sides are as close to equal as possible. For an automatic tape player, this will minimize the "dead time" when switching from one side to another. As an example, we mention that Willie Nelson has recorded the following ten songs in duets with other performers:

Song	**Time (min:secs)**	**Other Performer**
1) Pancho and Lefty	4:45	Merle Haggard
2) Slow Movin Outlaw	3:35	Lacy J. Dalton
3) Are There any More Real Cowboys	3:03	Neil Young
4) I Told a Lie to My Heart	2:52	Hank Williams
5) Texas on a Saturday Night	2:42	Mel Tillis
6) Seven Spanish Angels	3:50	Ray Charles
7) To All the Girls I've Loved Before	3:30	Julio Iglesias
8) They All Went to Mexico	4:45	Carlos Santana
9) Honky Tonk Women	3:30	Leon Russell
10) Half a Man	3:02	George Jones

You want to collect these songs on a two-sided tape cassette album to be called "Half Nelson."

(a) Formulate and solve an integer program for minimizing the dead time on the shorter side.

(b) What are some of the marketing issues of allowing the customer to decide which song goes on which side?

17. Bill Bolt is hosting a party for his daughter Lydia on the occasion of her becoming of college age. He has reserved a banquet room with 18 tables at the Racquet Club on Saturday night. Each table can accommodate at most 8 people. A total of 140 young people are coming, 76 young men and 64 young ladies. Lydia and her mother, Jane, would like to have the sexes as evenly distributed as possible at the tables. They want to have at least 4 men and at least 3 women at each table.

    (a) Is it possible to have an allocation satisfying the above as well as the restriction there be at most 4 men at each table?

    (b) Provide a good allocation of the sexes to the tables.

18. The game or puzzle of Clue is played with a deck of 21 cards. At the beginning of a game, three of the cards are randomly selected and placed face down in the center of the table. The remaining cards are distributed face down as evenly as possible to the players. Each player may look at his or her own cards. The object of the game is to correctly guess the three cards in the center. At each player's turn, the player is allowed to either guess the identity of the three cards in the center or ask any other player a question of the form: "Do you have any of the following three cards?" (The asking player then publicly lists the three cards.) If the asked player has one of the three identified cards, then the asked player must show one of the cards to the asking player (and only to the asking player). Otherwise, the asked player simply responds "No". If a player correctly guesses the three cards in the center, then that player wins. If a player incorrectly guesses the three cards in the center, the player is out of the game.

    Deductions about the identity of various cards can be made if we define:

    $X(i, j) = 1$ if player i has card j, else 0.

    Arbitrarily define the three cards in the center as player 1. Thus, we can initially start with the constraints:

$$\sum_{j=1}^{21} X(1, j) = 3.$$

    For each card, $j = 1, 2, \ldots, 21$:

$$\sum_{i} X(i, j) = 1.$$

    (a) Suppose player 3 is asked: "Do you have either card 4, 8, or 17?" and player 3 responds "No." What constraint can be added?

    (b) Suppose in response to your question in (a), player 3 shows you card 17. What constraint can be added?

    (c) What LP would you solve in order to determine whether card 4 must be one of the cards in the center?

    Note, in the "implementation" of the game marketed in North America, the 21 cards are actually divided into three types: (i) six suspect cards with names like "Miss Scarlet," (ii) six weapons cards with names like "Revolver," and (c) nine room cards with names like "Kitchen." This has essentially no effect on our analysis above.

4

Portfolio Optimization

4.1 Introduction

Financial portfolio models are concerned with investments where there are typically two criteria: *expected return* and *risk*. The investor wants the former to be high and the latter to be low. There are a variety of measures of risk. The most popular measure of risk has been variance in return. Even though there are some problems with it, we will first look at it very closely.

4.2 The Markowitz Mean/Variance Portfolio Model

The portfolio model introduced by Markowitz (1959) (see also Roy (1952)), assumes an investor has two considerations when constructing an investment portfolio: *expected return* and *variance in return* (i.e., risk). Variance measures the variability in realized return around the expected return, giving equal weight to realizations below the expected and above the expected return. The Markowitz model might be mildly criticized in this regard because the typical investor is probably concerned only with variability below the expected return, so-called *downside risk*.

The Markowitz model requires two major kinds of information: (1) the estimated expected return for each candidate investment and (2) the covariance matrix of returns. The covariance matrix characterizes not only the individual variability of the return on each investment, but also how each investment's return tends to move with other investments. We assume the reader is somewhat familiar with the concepts of *variance* and *covariance* as described in most intermediate statistics texts.

4.2.1 Example

We will use some publicly available data from Markowitz (1959). The following table shows the increase in price, including dividends, for three stocks over a twelve-year period:

		Growth in		
Year	**S&P500**	**ATT**	**GMC**	**USX**
43	1.259	1.300	1.225	1.149
44	1.198	1.103	1.290	1.260
45	1.364	1.216	1.216	1.419
46	0.919	0.954	0.728	0.922
47	1.057	0.929	1.144	1.169
48	1.055	1.056	1.107	0.965
49	1.188	1.038	1.321	1.133
50	1.317	1.089	1.305	1.732
51	1.240	1.090	1.195	1.021
52	1.184	1.083	1.390	1.131
53	0.990	1.035	0.928	1.006
54	1.526	1.176	1.715	1.908

For reference later, we have also included the change each year in the Standard and Poor’s/S&P 500 stock index. To illustrate, in the first year, *ATT* appreciated in value by 30%. In the second year, *GMC* appreciated in value by 29%.

Computing Covariances, Variances and Standard Deviations in Excel

Excel has the function COVAR() for computing covariances, VAR() for computing variances, *STDEV()* for computing standard deviations, and CORREL() for computing correlations. For reasons of numerical accuracy, we suggest that VAR() and STDEV() not be used. Below we discuss the usage of these functions. Given n observations on two random variables $\{X_i\}$ and $\{Y_i\}$, the population means are defined as the expectations:

$$\mu_X = \mathrm{E}[X_i], \quad \text{and} \quad \mu_Y = \mathrm{E}[Y_i].$$

The sample means are defined as the averages:

$$xbar = \textstyle\sum_i X_i / n, \quad \text{and} \quad ybar = \textstyle\sum_i Y_i / n.$$

The population covariance between X and Y is defined as the expectation:

$$\sigma^2_{XY} = \mathrm{E}[(X_i - \mu_X)(Y_i - \mu_Y)].$$

With some effort it can be shown that this is algebraically equivalent to:

$$\sigma^2_{XY} = \mathrm{E}[X_i\, Y_i\,] - \mu_X\, \mu_Y.$$

When X and Y are the same random variable, σ^2_{XX} is called the population variance.

The population standard deviation of X is defined as the square root of σ^2_{XX}, i.e.:

$$\sigma_X = (\sigma^2_{XX})^{0.5}.$$

The population correlation between X and Y is defined as

$$\rho_{XY} = (\sigma^2_{XY})/(\sigma_X \sigma_Y).$$

The sample covariance between X and Y is defined as the average:

$$s^2_{XY} = \sum_i [(X_i - xbar)(Y_i - ybar)]/n. \qquad (1)$$

With some effort it can be shown that this is algebraically equivalent to:

$$s^2_{XY} = \sum_i [X_i\, Y_i]/n - xbar * ybar. \qquad (2)$$

If in fact X and Y are the same random variable, s^2_{XX} is called the sample variance.

The sample standard deviation of X is defined as the square root of s^2_{XX}, i.e.:

$$s_X = (s^2_{XX})^{0.5}.$$

The sample correlation between X and Y is defined as

$$r_{XY} = (s^2_{XY})/(s_X\, s_Y).$$

Given weights w_X and w_Y, and the definition that $Z = w_X X + w_Y Y$, it can be shown that the variance of Z is:

$$\begin{aligned}\sigma^2_{ZZ} &= w_X^2 \sigma^2_{XX} + w_X w_Y \sigma^2_{XY} + w_Y w_X \sigma^2_{YX} + w_Y^2 \sigma^2_{YY} \\ &= w_X^2 \sigma^2_{XX} + 2 w_X w_Y \sigma^2_{XY} + w_Y^2 \sigma^2_{YY}.\end{aligned}$$

Although formulae (1) and (2) are algebraically equivalent, they are not numerically equivalent on a computer because of round-off error. Formula (1) is more accurate. In Excel, the functions VAR() and STDEV() are based on formula (2), whereas COVAR() and CORREL() are based on (1). In Excel, you can expect COVAR() and CORREL() to be accurate to at least six decimal places, whereas VAR() and STDEV() may have essentially no accuracy if *xbar* is large relative to s_X. To illustrate, suppose $n = 2$ with {X1, X2} = {123456789, 123456787}. If you use VAR()to compute the sample variance, or STDEV() to compute the sample standard deviation, they will both give an answer of 0.0, whereas it is easy to see that the sample variance should be $[(1)^2+(-1)^2]/2 = 1$.

One should also be interested in whether s^2_{XY} is a good estimator of the unknown parameter σ^2_{XY}. With a bit more algebra it can be shown that the expected value, $E(s^2_{XY}) = \sigma^2_{XY}\,(n-1)/n$. That is, s^2_{XY} underestimates σ^2_{XY}, especially when n is small. Thus, one typically applies a $n/(n-1)$ adjustment factor to s^2_{XY}. VAR() and STDEV() include the adjustment factor, but COVAR() does not. Because CORREL() is ratio of two estimators, the adjustment does not matter.

So, based on the twelve years of data, we use the COVAR() function in Excel to calculate the sample covariances for three stocks: ATT, GMC, and USX. Multiplying the results by 12/11 gives the following covariance matrix.

	ATT	**GMC**	**USX**
ATT	0.01080754	0.01240721	0.01307513
GMC	0.01240721	0.05839170	0.05542639
USX	0.01307513	0.05542639	0.09422681

From the same data, we estimate the expected return per year, including dividends, for *ATT*, *GMC*, and *USX* as 0.0890833, 0.213667, and 0.234583, respectively.

The correlation matrix makes it more obvious how two random variables move together. The correlation between two random variables equals the covariance between the two variables, divided by the product of the standard deviations of the two random variables. For our three investments, the sample correlation matrix is:

	ATT	GMC	USX
ATT	1.0		
GMC	0.493895589	1.0	
USX	0.409727718	0.747229121	1.0

The correlation can be between –1 and +1 with +1 being a high correlation between the two. Notice GMC and USX are highly correlated. ATT tends to move with GMC and USX, but not nearly so much as GMC moves with *USX*.

Let the symbols ATT, GMC, and USX represent the fraction of the portfolio devoted to each of the three stocks. Suppose, we desire a 15% yearly return. For the objective, we want to minimize the variance in the portfolio value after one year. In algebraic notation, what we want to do is:

```
Minimize
   0.01080754*ATT*ATT + 0.01240721*ATT*GMC +
   0.01307513*ATT*USX + 0.01240721*GMC*ATT +
   0.05839170*GMC*GMC + 0.05542639*GMC*USX +
   0.01307513*USX*ATT + 0.05542639*USX*GMC +
   0.09422681*USX*USX;
```

Use exactly 100% of the starting budget:

```
ATT + GMC + USX = 1;
```

Required wealth at end of period:

```
1.089083 * ATT + 1.213667 * GMC + 1.234583 * USX ≥ 1.15;
```

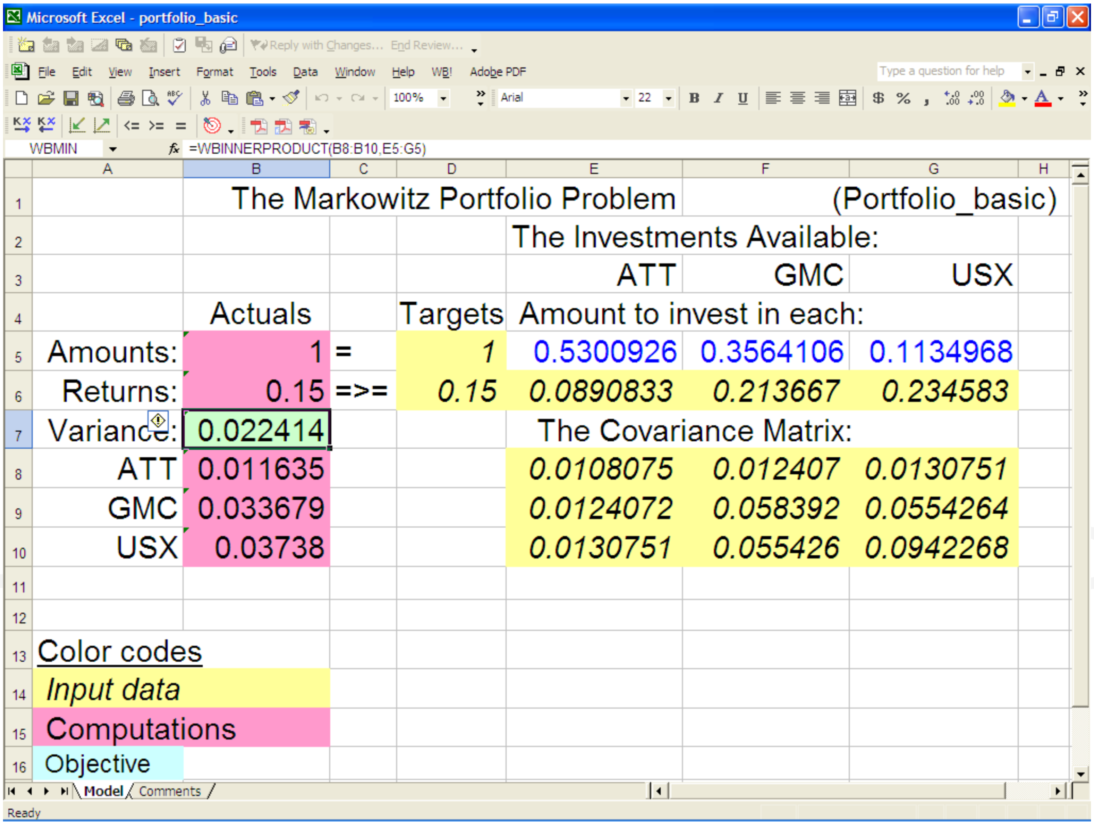
Note the two constraints are effectively in the same units. The first constraint is effectively a “beginning inventory” constraint, while the second constraint is an “ending inventory” constraint. Alternatively, we could have stated the expected return constraint just as easily as:

```
.0890833 * ATT + .213667 * GMC + .234583 * USX ≥ .15
```

Although perfectly correct, this latter style does not measure end-of-period state in quite the same way as start-of-period state. Fans of consistency may prefer the former style.

In preparation for writing the model in a spreadsheet, note that we can also write the objective as:

```
  ATT*(.01080754 * ATT +.01240721 * GMC +.01307513 * USX)
+ GMC*(.01240721 * ATT +.05839170 * GMC +.05542639 * USX)
+ USX*(.01307513 * ATT +.05542639 * GMC +.09422681 * USX);
```



In the spreadsheet, portfolio_basic, we calculate the expressions in parentheses in column B using the SUMPRODUCT() function, e.g., B8=SUMPRODUCT(E$5:G$5,E8:G8) in. We calculate the variance with cell B7=WBINNERPRODUCT(B8:B11,E5:G5). The WBINNERPRODUCT() function is similar to SUMPRODUCT(), except that it allows you to multiply a row vector by column vector. WBINNERPRODUCT expects one range to be a row range and the other a column range.

The “ABC’s of Optimization” for this spreadsheet are:

A) Adjustable Cells or Decision Variables, specifying how much to invest in each asset appear in row 5, cells E5:G5;

B) The Best or objective cell, the portfolio variance to be minimized is cell B7. The most complicated computation for this model is the computation of the variance of the portfolio. If x_i is the amount invested in asset i, and σ^2_{ij} is the covariance between one unit of i and one unit of j, then the portfolio variance = $\Sigma_i \Sigma_j\, x_i * x_j * \sigma^2_{ij}$. This can be rewritten:

$$variance = \Sigma_i\, x_i\, \Sigma_j\, x_j * \sigma^2_{ij}.$$

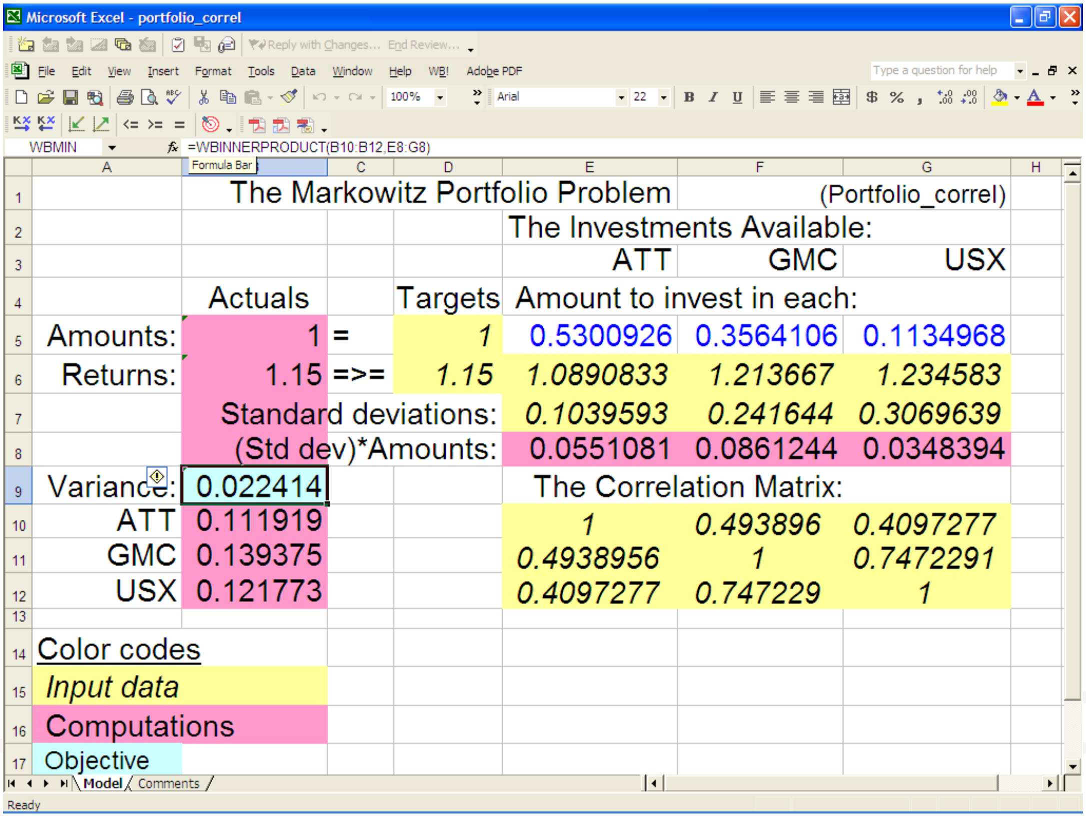
In the spreadsheet, Column B computes the inner summation, $\Sigma_j\, x_j * \sigma^2_{ij}$. For example, cell B8 contains the formula =SUMPRODUCT(E8:G8,E$5:G$5). The "$5" holds row 5 constant when the formula is copied down to cells B9:B10. The final summation, $\Sigma_i\, x_i\, \Sigma_j$: $x_j * \sigma^2_{ij}$, is done in cell B7.

C) Constraints: There are two constraints in this model. Cell C5, which contains =WB(B5,"=",D5), says the amount invested(computed in B5) must equal the target amount to invest given as input in D5. Cell C6, which contains =WB(B6,">=",D6), says the expected return(computed in B6) must be greater than or equal to the target return specified in D6.

The solution recommends about 53% of the portfolio be put in *ATT*, about 36% in *GMC* and just over 11% in *USX*. The expected return is 15%, with a variance of 0.02241381 or, equivalently, a standard deviation of about 0.1497123.

Using a Correlation Matrix

We based the previous model simply on straightforward statistical data based on yearly returns. In practice, it may be more typical to use monthly rather than yearly data as a basis for calculating covariances. Also, rather than use historical data for estimating the expected return of an asset, a decision maker might base the expected return estimate on more current, proprietary information about expected future performance of the asset. One may also wish to use considerable care in estimating the covariances and the expected returns. For example, one could use quite recent data to estimate the standard deviations. A larger set of data extending further back in time might be used to estimate the correlation matrix. Then, using the relationship between the correlation matrix and the covariance matrix, one could derive a covariance matrix. The version portfolio_correl, illustrates two alternative approaches to this problem: a) using the correlation matrix instead of the covariance matrix to describe how investments tend to move together, and b) and stating the desired return as a growth factor, 1.15, rather than a fraction return, 0.15.



	A	B	C	D	E	F	G	H
1		The Markowitz Portfolio Problem					(Portfolio_correl)	
2					The Investments Available:			
3					ATT	GMC	USX	
4		Actuals		Targets	Amount to invest in each:			
5	Amounts:	1	=	1	0.5300926	0.3564106	0.1134968	
6	Returns:	1.15	=>=	1.15	1.0890833	1.213667	1.234583	
7		Standard deviations:			0.1039593	0.241644	0.3069639	
8		(Std dev)*Amounts:			0.0551081	0.0861244	0.0348394	
9	Variance:	0.022414			The Correlation Matrix:			
10	ATT	0.111919			1	0.493896	0.4097277	
11	GMC	0.139375			0.4938956	1	0.7472291	
12	USX	0.121773			0.4097277	0.747229	1	
13								
14	Color codes							
15	Input data							
16	Computations							
17	Objective							

The most significant difference between this formulation and the previous one is in the computation of the portfolio variance. Here we exploit the fact that the variance can be written in terms of the correlations and the standard deviations as:

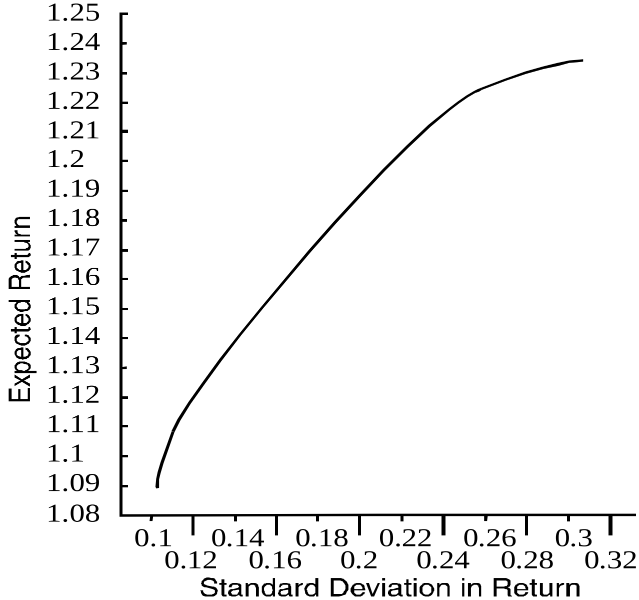
$$\text{variance} = \Sigma_i \Sigma_j \, x_i * x_j * \sigma_i * \sigma_j * \rho_{ij}. = \Sigma_i \, x_i * \sigma_i \; \Sigma_j \, x_j * \sigma_j * \rho_{ij}.$$

In row 8 we compute the term, $x_j * \sigma_j$, e.g., with formulae such as: E8=E5*E7. In column B we compute the inner sum, $\Sigma_j \, x_j * \sigma_j * \rho_{ij}$, with formulae such as B10=SUMPRODUCT(E10:G10,E$8:G$8). The outer summation is computed in cell B9 with the formula: B9=WBINNERPRODUCT(B10:B13,E8:H18). Observe that the same solution is obtained.

4.3 Dualing Objectives: Efficient Frontier and Parametric Analysis

There is no obvious way for an investor to determine the "correct" tradeoff between risk and return. Thus, one is frequently interested in looking at the tradeoff between the two. If an investor wants a higher expected return, she generally has to "pay for it" with higher risk. In finance terminology, we would like to trace out the efficient frontier of return and risk. If we solve for the minimum variance portfolio over a range of values for the expected return, ranging from 0.0890833 to 0.234583, we get the following plot or tradeoff curve for our little three-asset example:

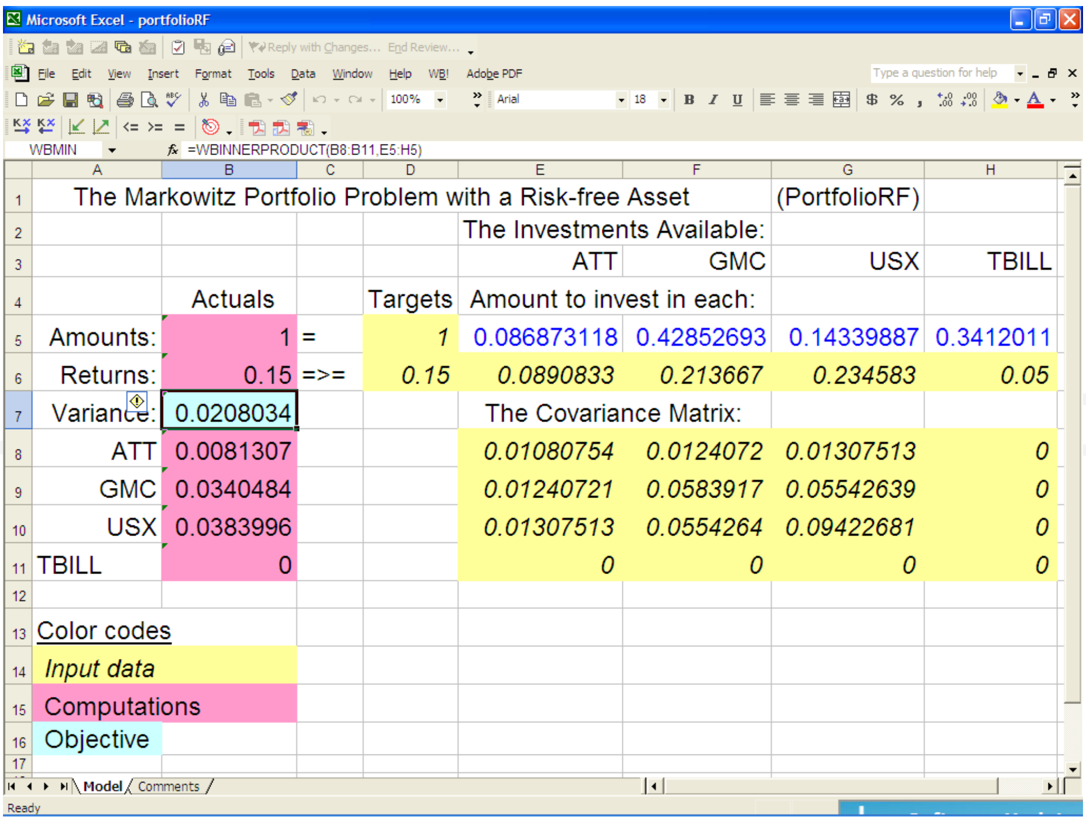
Figure 2.1 Efficient Frontier



Notice the “knee” in the curve as the required expected return increases past 1.21894. This is the point where ATT drops out of the portfolio.

4.3.1 Portfolios with a Risk-Free Asset

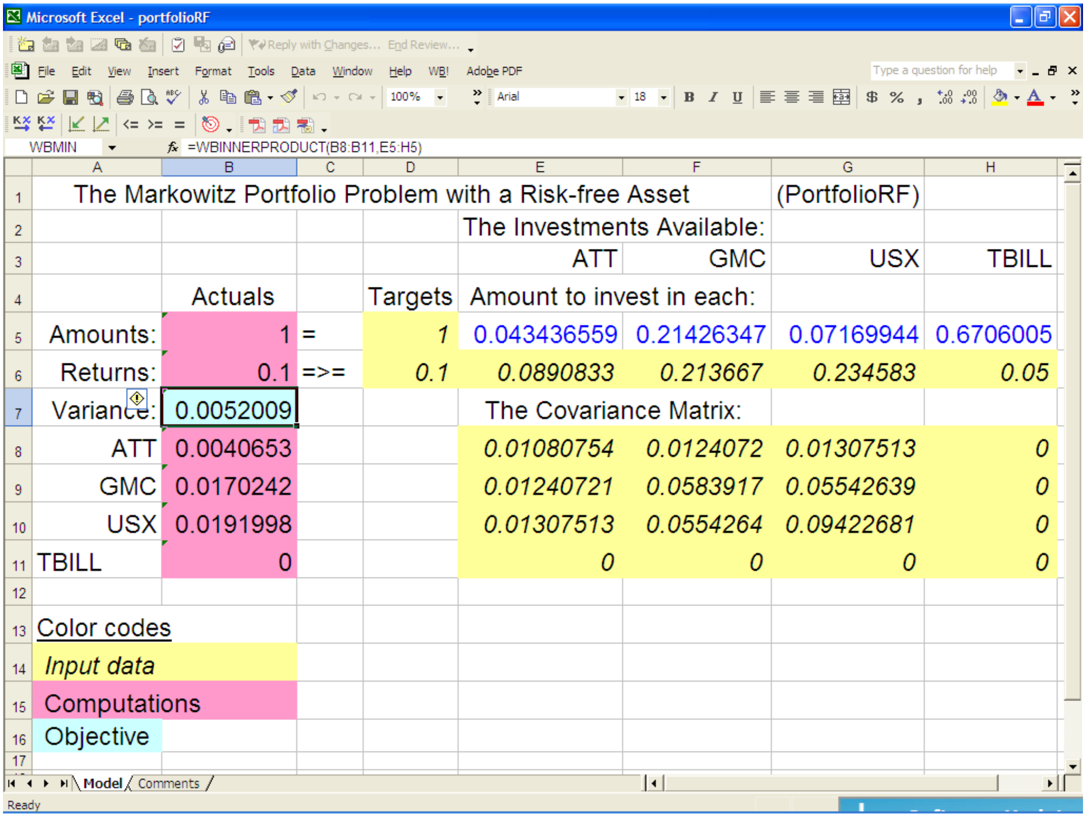
When one of the investments available is risk free, then the optimal portfolio composition has a particularly simple form. Suppose the opportunity to invest money risk free (e.g., in government treasury bills) at 5% per year has just become available. Working with our previous example, we now have a fourth investment instrument that has zero variance and zero covariance. There is no limit on how much can be invested at 5%. We ask the question: How does the portfolio composition change as the desired rate of return changes from 15% to 5%?



	A	B	C	D	E	F	G	H
1	The Markowitz Portfolio Problem with a Risk-free Asset						(PortfolioRF)	
2					The Investments Available:			
3					ATT	GMC	USX	TBILL
4		Actuals		Targets	Amount to invest in each:			
5	Amounts:	1	=	1	0.086873118	0.42852693	0.14339887	0.3412011
6	Returns:	0.15	=>=	0.15	0.0890833	0.213667	0.234583	0.05
7	Variance:	0.0208034			The Covariance Matrix:			
8	ATT	0.0081307			0.01080754	0.0124072	0.01307513	0
9	GMC	0.0340484			0.01240721	0.0583917	0.05542639	0
10	USX	0.0383996			0.01307513	0.0554264	0.09422681	0
11	TBILL	0			0	0	0	0
12								
13	Color codes							
14	Input data							
15	Computations							
16	Objective							
17								

Notice that more than 34% of the portfolio was invested in the risk-free investment, the T-bill, even though its return rate, 5%, is less than the target of 15%. Further, the variance has dropped to about 0.0208 from about 0.0224.

What happens as we decrease the target return towards 5%? Clearly, at 5%, we would put zero in *ATT*, *GMC*, and *USX*. A simple form of solution would be to keep the same proportions in *ATT*, *GMC*, and *USX*, but just change the allocation between the risk-free asset and the risky ones. Let us check an intermediate point. When we decrease the required return to 10%, we get the following solution:



	A	B	C	D	E	F	G	H
1	The Markowitz Portfolio Problem with a Risk-free Asset						(PortfolioRF)	
2					The Investments Available:			
3					ATT	GMC	USX	TBILL
4		Actuals		Targets	Amount to invest in each:			
5	Amounts:	1	=	1	0.043436559	0.21426347	0.07169944	0.6706005
6	Returns:	0.1	=>=	0.1	0.0890833	0.213667	0.234583	0.05
7	Variance:	0.0052009			The Covariance Matrix:			
8	ATT	0.0040653			0.01080754	0.0124072	0.01307513	0
9	GMC	0.0170242			0.01240721	0.0583917	0.05542639	0
10	USX	0.0191998			0.01307513	0.0554264	0.09422681	0
11	TBILL	0			0	0	0	0
12								
13	Color codes							
14	Input data							
15	Computations							
16	Objective							
17								

This solution supports our conjecture:

> *As we change our required return, the relative proportions devoted to risky investments do not change. Only the allocation between the risk-free asset and the risky asset change.*

From the above solution, we observe that, except for round-off error, the amount invested in *ATT*, *GMC*, and *USX* is allocated in the same way for both solutions. Thus, two investors with different risk preferences would nevertheless both carry the same mix of risky stocks in their portfolio. Their portfolios would differ only in the proportion devoted to the risk-free asset. Our observation from the above example in fact holds in general. Thus, the decision of how to allocate funds among stocks, given the amount to be invested, can be separated from the questions of risk preference. Tobin received the Nobel Prize in 1981, largely for noticing the above feature, the so-called Separation Theorem. So, if you noticed it, you must be Nobel Prize caliber.

4.3.2 The Sharpe Ratio

For some portfolio p, of risky assets, excluding the risk-free asset, let:

R_p = its expected return,

s_p = its standard deviation in return, and
r_0 = the return of the risk-free asset.

A plausible single measure (as opposed to the two measures, risk and return) of attractiveness of portfolio p is the Sharpe ratio:

$(R_p - r_0) / s_p$.

In words, it measures how much additional return we achieved for the additional risk we took on, relative to putting all our money in the risk-free asset.

It happens the portfolio that maximizes this ratio has a certain well-defined appeal. Suppose:

t = our desired target return,
w_p = fraction of our wealth we place in portfolio p
(the rest placed in the risk-free asset).

To meet our return target, we must have:

$(1 - w_p) * r_0 + w_p * R_p = t.$

The standard deviation of our total investment is:

$w_p * s_p.$

Solving for wp in the return constraint, we get:

$w_p = (t - r_0) /(R_p - r_0).$

Thus, the standard deviation of the portfolio is:

$w_p * s_p = [(t - r_0) /(R_p - r_0)] * s_p.$

Minimizing the portfolio standard deviation means:

$\text{Min } [(t - r_0) /(R_p - r_0)] * s_p$
or
$\text{Min } [(t - r_0) * s_p /(R_p - r_0)].$

This is equivalent to:

$\text{Max } (R_p - r_0) /s_p.$

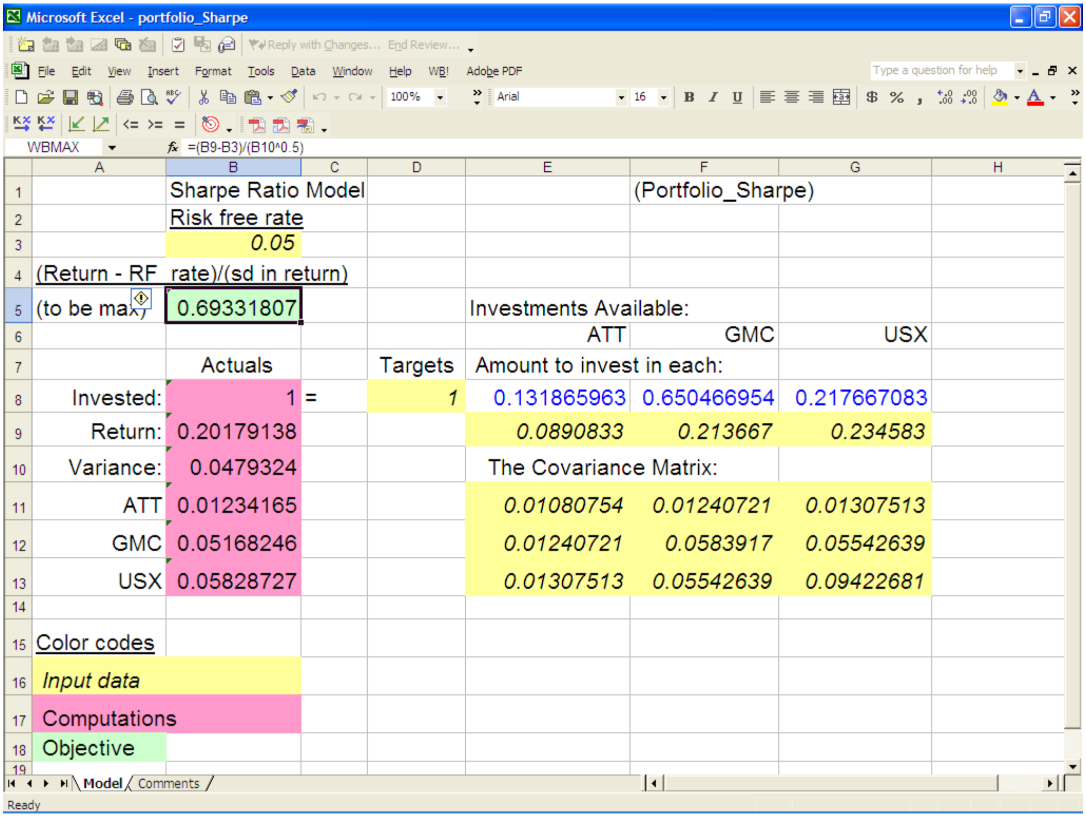
So, regardless of our risk/return preference, the money we invest in risky assets should be invested in the risky portfolio that maximizes the Sharpe ratio.

Algebraically, if the risk free rate is 5%, then what we would like to do is:

```
! Maximize the Sharpe ratio;
 MAX =
(1.089083*ATT + 1.213667*GMC + 1.234583*USX - 1.05)/
 ((.01080754*ATT*ATT + .01240721*ATT*GMC + .01307513*ATT*USX
 + .01240721*GMC*ATT + .05839170*GMC*GMC + .05542639*GMC*USX
 + .01307513*USX*ATT + .05542639*USX*GMC + .09422681*USX*USX)^.5);

! Use exactly 100% of the starting budget;
  ATT + GMC + USX = 1;
```

The spreadsheet *portfolio_sharpe* illustrates. The crucial differences from the previous models are: a) There is no target return constraint, and b) the Sharpe ratio is computed with: B5=(B9-B3)/(B10^0.5).



	A	B	C	D	E	F	G	H
1		Sharpe Ratio Model				(Portfolio_Sharpe)		
2		Risk free rate						
3		0.05						
4	(Return - RF rate)/(sd in return)							
5	(to be max)	0.69331807			Investments Available:			
6					ATT	GMC	USX	
7		Actuals		Targets	Amount to invest in each:			
8	Invested:	1	=	1	0.131865963	0.650466954	0.217667083	
9	Return:	0.20179138			0.0890833	0.213667	0.234583	
10	Variance:	0.0479324			The Covariance Matrix:			
11	ATT	0.01234165			0.01080754	0.01240721	0.01307513	
12	GMC	0.05168246			0.01240721	0.0583917	0.05542639	
13	USX	0.05828727			0.01307513	0.05542639	0.09422681	
14								
15	Color codes							
16	Input data							
17	Computations							
18	Objective							
19								

Notice the relative proportions of ATT, GMC, and USX are the same as in the previous model where we explicitly included a risk free asset with a return of 5%. E.g., except for round-off error:

```
0.131865963/0.6504669543 = 0.086873118/0.42852693.
```

The formulae in the spreadsheet Portfolio_Sharpe are essentially the same as in the previous except for the objective function in cell B5. It is B5=(B9-B3)/(B10^0.5), that is,

(expected_return – risk_free_rate)/(square_root_of_portfolio_variance).

4.4 Important Variations of the Portfolio Model

There are several issues that may concern you when you think about applying the Markowitz model in its simple form:

a) As we increase the number of assets to consider, the size of the covariance matrix becomes overwhelming. For example, 1000 assets implies 1,000,000 covariance terms, or at least 500,000 if symmetry is exploited.

b) If the model were applied every time new data become available (e.g., weekly), we would "rebalance" the portfolio frequently, making small, possibly unimportant adjustments in the portfolio.

c) There are no upper bounds on how much can be held of each asset. In practice, there might be legal or regulatory reasons for restricting the amount of any one asset to no more than, say, 5% of the total portfolio. Some portfolio managers may set the upper limit on a stock to one day's trading volume for the stock. The reasoning being, if the manager wants to "unload" the stock quickly, the market price would be affected significantly by selling so much.

Two approaches for simplifying the covariance structure have been proposed: the scenario approach and the factor approach. For the issue of portfolio "nervousness", the incorporation of transaction costs is useful.

4.4.1 Portfolios with Transaction Costs

The models above do not tell us much about how frequently to adjust our portfolio as new information becomes available, e.g., new estimates of expected return and new estimates of variance. If we applied the above models every time new information became available, we would be constantly adjusting our portfolio. This might make our broker happy because of all the commission fees, but that should be a secondary objective at best. The important observation is that there are costs associated with buying and selling. There are the obvious commission costs, and the not so obvious bid-ask spread. The bid-ask spread is effectively a transaction cost for buying and selling.

The method we will describe assumes transaction costs are paid at the beginning of the period. It is a straightforward exercise to modify the model to handle the case of transaction costs paid at the end of the period. The major modifications to the basic portfolio model are:

a) We must introduce two additional variables for each asset, an "amount bought" variable and an "amount sold" variable.

b) The budget constraint must be modified to include money spent on commissions.

c) An additional constraint must be included for each asset to enforce the requirement:
amount invested in asset *i* = (initial holding of *i*) +
(amount bought of *i*) – (amount sold of *i*).

4.4.2 Example

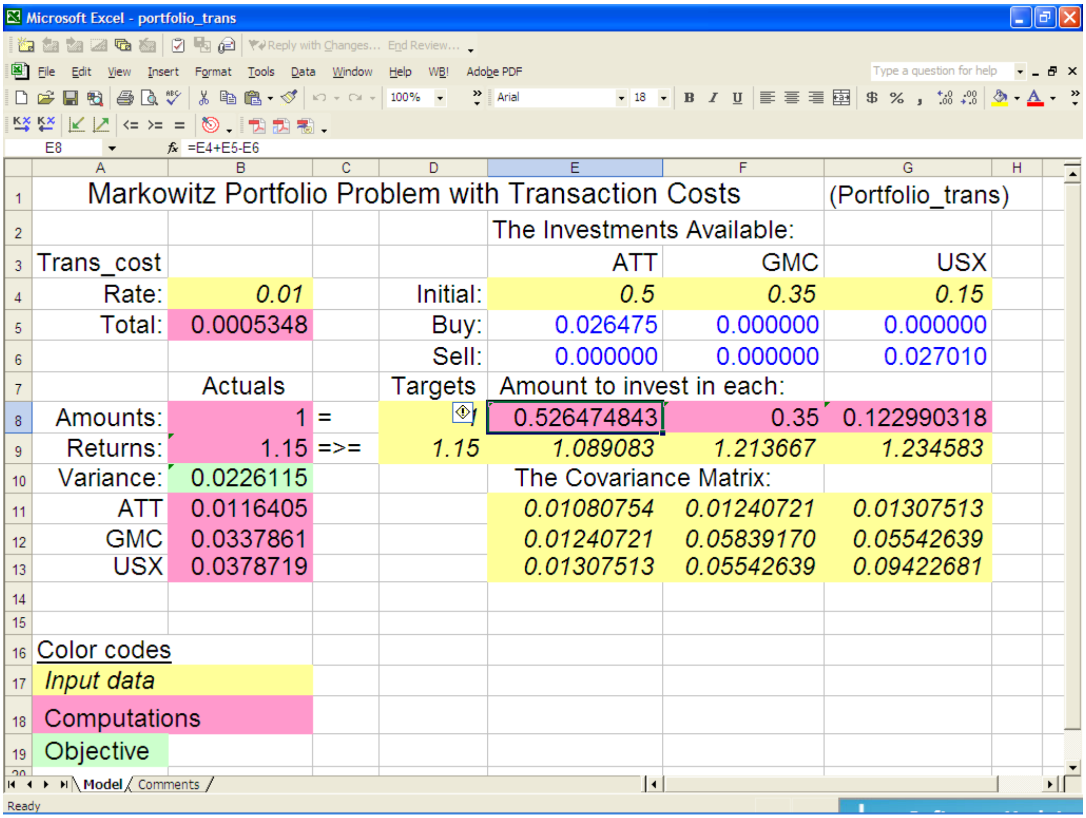
Suppose we have to pay a 1% transaction fee on the amount bought or sold of any stock and our current portfolio is 50% *ATT*, 35% *GMC*, and 15% *USX*. This is pretty close to the optimal mix. Should we incur the cost of adjusting? The following is the relevant model:

```
MIN = .01080754 * ATT * ATT +.01240721 * ATT * GMC +.01307513 * ATT
 * USX +.01240721 * GMC * ATT +.05839170 * GMC * GMC +.05542639 *
 GMC * USX +.01307513 * USX * ATT +.05542639 * USX * GMC +.09422681
 * USX * USX;
ATT + GMC + USX + .01 * ( BA + BG + BU + SA + SG + SU) = 1;
1.089083 * ATT + 1.213667 * GMC + 1.234583 * USX >= 1.15;
ATT = .50 + BA - SA;
GMC = .35 + BG - SG;
USX = .15 + BU - SU;
```

The first constraint says the total uses of funds must equal 1. Another way of interpreting this constraint is to subtract each of the next three constraints from it. We then get:

```
.01 * (BA + BG + BU + SA + SG + SU) + BA + BG + BU=SA + SG + SU;
```

It says any purchases plus transaction fees must be funded by selling. The spreadsheet model is:



	A	B	C	D	E	F	G
1	Markowitz Portfolio Problem with Transaction Costs						(Portfolio_trans)
2					The Investments Available:		
3	Trans_cost				ATT	GMC	USX
4	Rate:	0.01		Initial:	0.5	0.35	0.15
5	Total:	0.0005348		Buy:	0.026475	0.000000	0.000000
6				Sell:	0.000000	0.000000	0.027010
7		Actuals		Targets	Amount to invest in each:		
8	Amounts:	1	=		0.526474843	0.35	0.122990318
9	Returns:	1.15	=>=	1.15	1.089083	1.213667	1.234583
10	Variance:	0.0226115			The Covariance Matrix:		
11	ATT	0.0116405			0.01080754	0.01240721	0.01307513
12	GMC	0.0337861			0.01240721	0.05839170	0.05542639
13	USX	0.0378719			0.01307513	0.05542639	0.09422681
14							
15							
16	Color codes						
17	Input data						
18	Computations						
19	Objective						

The solution recommends buying a little bit more *ATT*, neither buy nor sell any *GMC*, and sell a little *USX*.

The ABC's of this spreadsheet are:

A) The Adjustable cells are the Buy variables in row 5, and the Sell variables in row 6.

B) The "Best" or objective cell is cell B10=WBINNERPRODUCT(B11:B13,E8:G8),

i.e., the variance in the end of period portfolio value.

C) There are two constraints:

C8 contains =WB(B8,"=",D9), and C9 contains =WB(B8,">=",D9).

The crucial formulae are:

Row 8 computes the amount held of each asset after transactions, e.g., E8=E4+E5-E6.

Column B computes the first half of the variance calculation, e.g.,

B11=SUMPRODUCT(E11:G11,E$8:G$8).

Cell B10 completes the variance calculation with

B10=WBINNERPRODUCT(B11:B13,E8:G8),

Cell B5 computes total transaction expenses from both buying and selling:

B5=B4*SUM(E5:G6);

Cell B8 computes the total uses of funds, i.e., transactions expense + amount in assets after transactions:

B8=B5+SUM(E8:G8);

Cell B9 computes the expected portfolio value at the end of the period:

B9=SUMPRODUCT(E9:G9,E$8:G$8);

4.4.3 Portfolios with Taxes

Taxes are an unpleasant complication of investment analysis that should be considered. The effect of taxes on a portfolio is illustrated by the following results during one year for two similar "growth-and-income" portfolios from the Vanguard company. Portfolio S was managed without (Sans) regard to taxes. Portfolio T was managed with after-tax performance in mind:

	Distributions		Initial	
Portfolio	**Income**	**Gain-from-sales**	**Share-price**	**Return**
S	$0.41	$2.31	$19.85	33.65%
T	$0.28	$0.00	$13.44	34.68%

The tax managed portfolio, probably just by chance, in fact had a higher before tax return. It looks even more attractive after taxes. If the tax rate for both dividend income and capital gains is 30%, then the tax paid at year end per dollar invested in portfolio *S* is .3 × (.41 + 2.31) /19.85 = 4.1 cents; whereas, the tax per dollar invested in portfolio *S* is .3 × .28/13.44 = 0.6 of a cent.

Below is a generalization of the Markowitz model to take into account taxes. As input, it requires in particular:

a) number of shares held of each kind of asset,

b) price per share paid for each asset held, and

c) estimated dividends per share for each kind of asset.

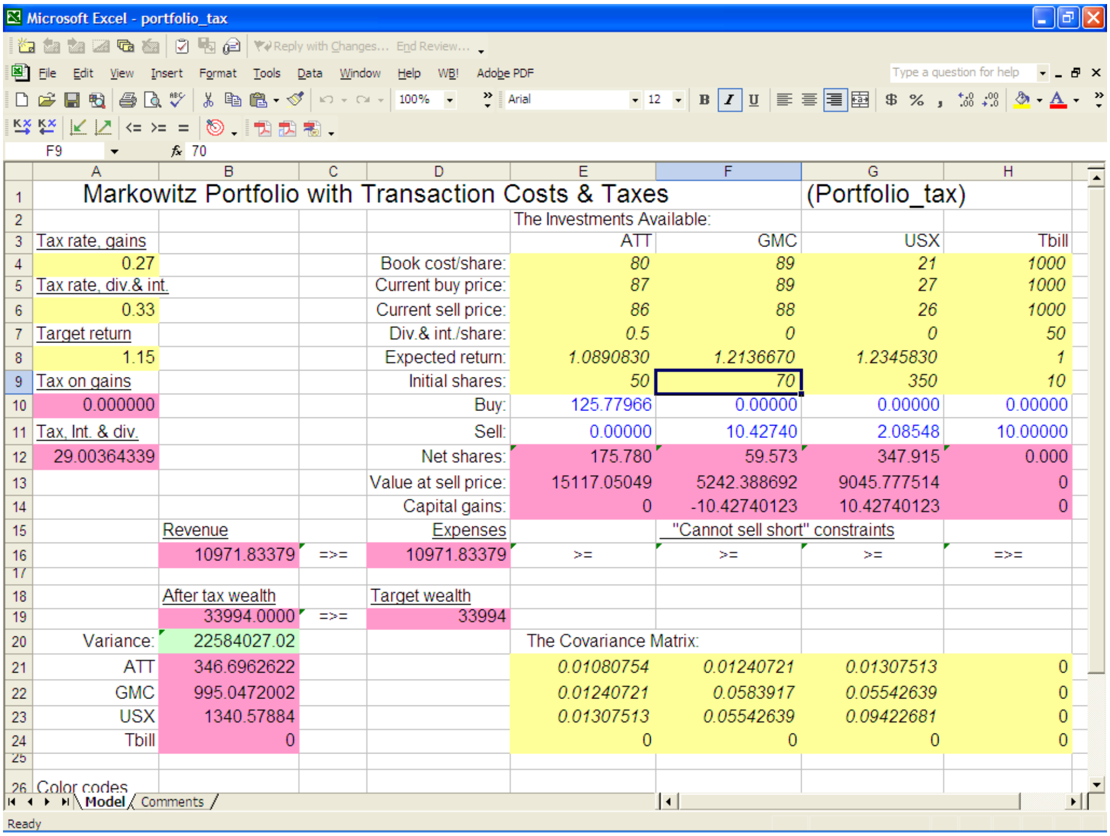
The results from this model will differ from a model that does not consider taxes in that this model, when considering equally attractive assets, will tend to:

i. purchase the asset that does not pay dividends, so as to avoid the immediate tax on dividends,

ii. sell the asset that pays dividends, and

iii. sell the asset whose purchase cost was higher, so as to avoid more tax on capital gains.

This is all given that two assets are otherwise identical (presuming rates of return are computed including dividends). For completeness, this model also includes transaction costs.



Microsoft Excel - portfolio_tax

F9 = 70

	A	B	C	D	E	F	G	H
1	Markowitz Portfolio with Transaction Costs & Taxes						(Portfolio_tax)	
2					The Investments Available:			
3	Tax rate, gains				ATT	GMC	USX	Tbill
4	0.27			Book cost/share:	80	89	21	1000
5	Tax rate, div.& int.			Current buy price:	87	89	27	1000
6	0.33			Current sell price:	86	88	26	1000
7	Target return			Div.& int./share:	0.5	0	0	50
8	1.15			Expected return:	1.0890830	1.2136670	1.2345830	1
9	Tax on gains			Initial shares:	50	70	350	10
10	0.000000			Buy:	125.77966	0.00000	0.00000	0.00000
11	Tax, Int. & div.			Sell:	0.00000	10.42740	2.08548	10.00000
12	29.00364339			Net shares:	175.780	59.573	347.915	0.000
13				Value at sell price:	15117.05049	5242.388692	9045.777514	0
14				Capital gains:	0	-10.42740123	10.42740123	0
15		Revenue		Expenses		"Cannot sell short" constraints		
16		10971.83379	=>=	10971.83379	>=	>=	>=	=>=
17								
18		After tax wealth		Target wealth				
19		33994.0000	=>=	33994				
20	Variance:	22584027.02			The Covariance Matrix:			
21	ATT	346.6962622			0.01080754	0.01240721	0.01307513	0
22	GMC	995.0472002			0.01240721	0.0583917	0.05542639	0
23	USX	1340.57884			0.01307513	0.05542639	0.09422681	0
24	Tbill	0			0	0	0	0
25								
26	Color codes							

Model / Comments

Ready

Notice that the solution recommends selling 2.08548 shares of USX at \$26/share. Because these shares were bought at 21, this generates a capital gain of 10.4274. This gain, however, is exactly cancelled out by selling 10.4274 shares of GMC at \$88/share. These shares were bought at \$87, so this generates a capital loss of 10.4274, so the portfolio does not have to pay any capital gains tax.

There are no constraints in the model to prevent both selling and buying a given stock or instrument. In fact, in some instances the model may recommend doing this so as to recognize or claim a capital loss. This is called a “wash sale” and U.S. tax rules prevent you from claiming the capital loss. The general rule is that if you sell a security and also buy the same security within the 30 days before, the same day, or the 30 days after the sale, then you cannot claim a capital loss from the sale. To the extent that wash sales are recommended by the model, it does not accurately model U.S. tax rules.

The ABC’s of this spreadsheet are:

A) The Adjustable cells are the Buy variables E11:H11, and the Sell variables in row E12:H12.

B) The “Best” or objective cell is B20=WBINNERPRODUCT(B21:B24,E$13:H$13),

i.e., the variance in the end of period portfolio value.

C) The constraints are:

C16=WB(B16,”>=”,D16)
C19=WB(B19,”>=”,D19),

Cannot sell short, i.e., hold negative quantities of an asset, cells E16:H16.

E16=WB(12,”>=”,0),

The crucial formulae are:

A10=A4*MAX(0,SUM(E14:H14),
A12=A6*SUMPRODUCT(E12:H14)
B16=SUMPRODUCT(E11:H11,E6:H6),

B19 computes the expected portfolio value at the end of the period:

B19=SUMPRODUCT(E8:H8,E13:H13),

Column B computes the first half of the variance calculation, e.g.,

B21=SUMPRODUCT(E21:H21,E$13:H$13),

Cell B20 completes the variance calculation with

B20=WBINNERPRODUCT(B21:B24,E13:G13),
D16=SUMPRODUCT(E10:H10,E5:H5)+A1,
D19=A8*SUMPRODUCT(E6:H6,E9:H9)

Row 12 computes the amount held of each asset after transactions, e.g.,

E12=E9+E10-E11,
E13=E12*E6,
E14=(E6-E4)*E11,

4.4.4 Factors Model for Simplifying the Covariance Structure

Sharpe (1963) introduced a substantial simplification to the modeling of the random behavior of stock market prices. He proposed that there is a “market factor” that has a significant effect on the movement of a stock. The market factor might be the Dow-Jones Industrial average, the S&P 500 average, or the Nikkei index. If we define:

M = the market factor,
m_0 = E(M),
s_0^2 = var(M),
e_i = random movement specific to stock i,
s_i^2 = var(e_i).

Sharpe’s approximation assumes (where E( ) denotes expected value):

$E(e_i) = 0$
$E(e_i\, e_j) = 0 \quad \text{for } i \neq j,$

$E(e_i M) = 0.$

Then, according to the Sharpe single factor model, the return of one dollar invested in stock or asset i is:

$u_i + b_i M + e_i.$

The parameters u_i and b_i are obtained by regression (e.g., least squares, of the return of asset i on the market factor). The parameter b_i is known as the "beta" of the asset. Let:

X_i = amount invested in asset i and

define the variance in return of the portfolio as:

$$\begin{aligned}&\operatorname{var}[\sum X_i(u_i + b_i M + e_i)]\\ &= \operatorname{var}(\sum X_i b_i M) + \operatorname{var}(\sum X_i e_i)\\ &= (\sum X_i b_i)^2 s_o^2 + \sum X_i^2 s_i^2.\end{aligned}$$

Thus, our problem can be written:

$$\begin{aligned}&\text{Minimize} \quad Z^2 s_o^2 + \sum X_i^2 s_i^2\\ &\text{subject to}\\ &\quad Z - \sum X_i b_\mathrm{i} = 0\\ &\quad \sum X_i = 1\\ &\quad \sum X_i (u_i + b_i m_o) \geq r.\end{aligned}$$

So, at the expense of adding one constraint and one variable, we have reduced a dense covariance matrix to a diagonal covariance matrix.

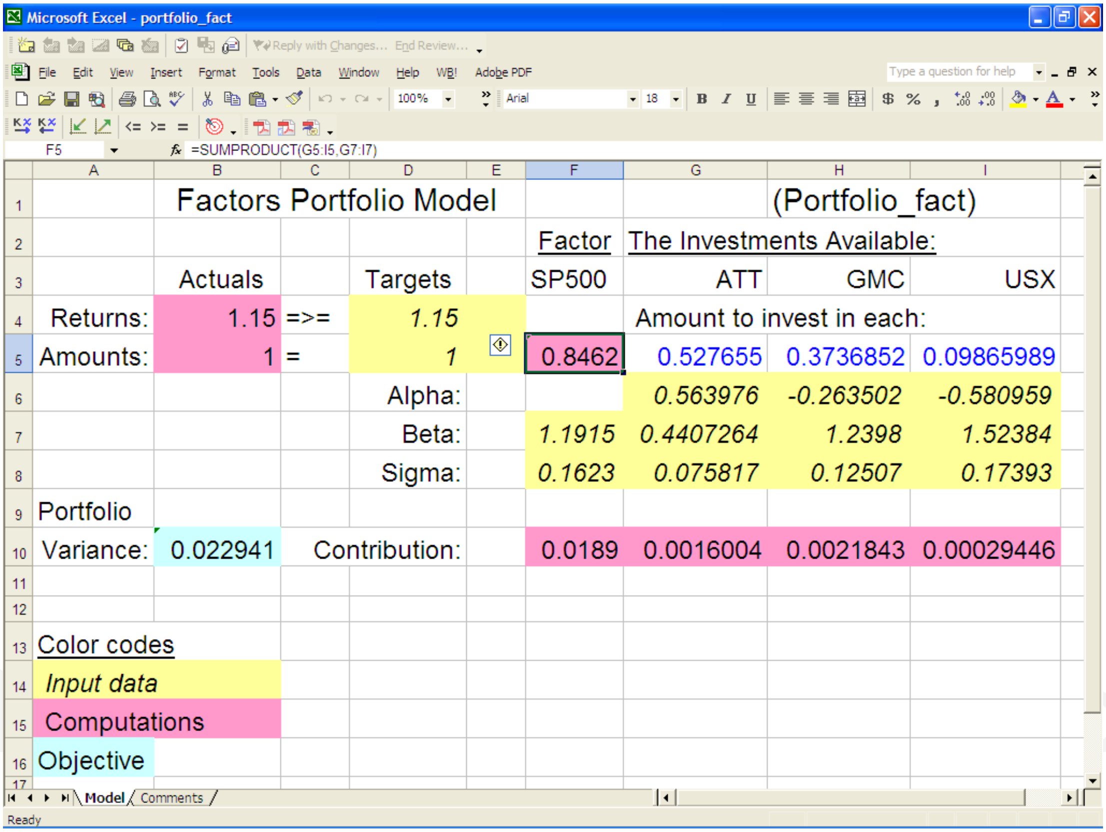
In practice, perhaps a half dozen factors might be used to represent the "systematic risk". That is, the return of an asset is assumed to be correlated with a number of indices or factors. Typical factors might be a market index such as the S&P 500, interest rates, inflation, defense spending, energy prices, gross national product, correlation with the business cycle, various industry indices, etc. For example, bond prices are very affected by interest rate movements.

4.4.5 Example of the Factor Model

The Factor Model represents the variance in return of an asset as the sum of the variances due to the asset's movement with one or more factors, plus a factor-independent variance.

To illustrate the factor model, we used multiple regression to regress the returns of *ATT*, *GMC*, and *USX* on the S&P 500 index for the same period. The stocks were regressed on the factor, SP500, based on the formula: Return(i) = Alpha(i) + Beta(i) * SP500 + error(i). The results were:

```
ASSET =    ATT          GMC       USX;
ALPHA = .563976   -.263502  -.580959;
BETA  = .4407264  1.23980   1.52384;
SIGMA = .075817    .125070   .173930;
```



Microsoft Excel - portfolio_fact

F5 =SUMPRODUCT(G5:I5,G7:I7)

	A	B	C	D	E	F	G	H	I
1		Factors Portfolio Model						(Portfolio_fact)	
2						Factor	The Investments Available:		
3		Actuals		Targets		SP500	ATT	GMC	USX
4	Returns:	1.15	=>=	1.15			Amount to invest in each:		
5	Amounts:	1	=	1		0.8462	0.527655	0.3736852	0.09865989
6				Alpha:			0.563976	-0.263502	-0.580959
7				Beta:		1.1915	0.4407264	1.2398	1.52384
8				Sigma:		0.1623	0.075817	0.12507	0.17393
9	Portfolio								
10	Variance:	0.022941	Contribution:			0.0189	0.0016004	0.0021843	0.00029446
11									
12									
13	Color codes								
14	*Input data*								
15	Computations								
16	Objective								

Model / Comments

Notice the portfolio makeup is slightly different. However, the estimated variance of the portfolio is very close to our original portfolio.

The important formulae are:

B4=SUMPRODUCT(G5:I5,G6:I6)+F5*F7,
B5=SUM(G5:I5),
C4=WB(B4,">=",D4),
C5=(B5,"=",D5),
F5=SUMPRODUCT(G5:I5,G7:I7),
F10=(F8*F5)^2,
B10=SUM(F10:I10).

4.4.6 Scenario Model for Representing Uncertainty

The scenario approach to modeling uncertainty assumes the possible future situations can be represented by a small number of "scenarios". The smallest number used is typically three (e.g., "optimistic," "most likely," and "pessimistic"). Some of the original ideas underlying the scenario approach come from the approach known as stochastic programming; see Madansky (1962), for example. For a discussion of the scenario approach for large portfolios, see Markowitz and Perold (1981) and Perold (1984). For a thorough discussion of the general approach of stochastic programming, see Infanger (1992). Eppen, Martin, and Schrage (1988) use the scenario approach for capacity planning in the automobile industry.

Let:

P_s = Probability scenario s occurs,
u_{is} = return of asset i if the scenario is s,
X_i = investment in asset i,
Y_s = deviation of actual return from the mean if the scenario is s;
$= \sum_i X_i(u_{is} - \sum_q P_q u_{iq})$.

Our problem in algebraic form is:

Minimize $\sum_s P_s Y_s^2$
subject to

$Y_s - \sum_i X_i(u_{i\,s} - \sum_q P_q u_{iq}) = 0$ (deviation from mean of each scenario, s)
$\sum_i X_i = 1$ (budget constraint)
$\sum_i X_i \sum_s P_s u_{is} \geq r$ (desired return).

If asset i has an inherent variability v_i^2, the objective generalizes to:

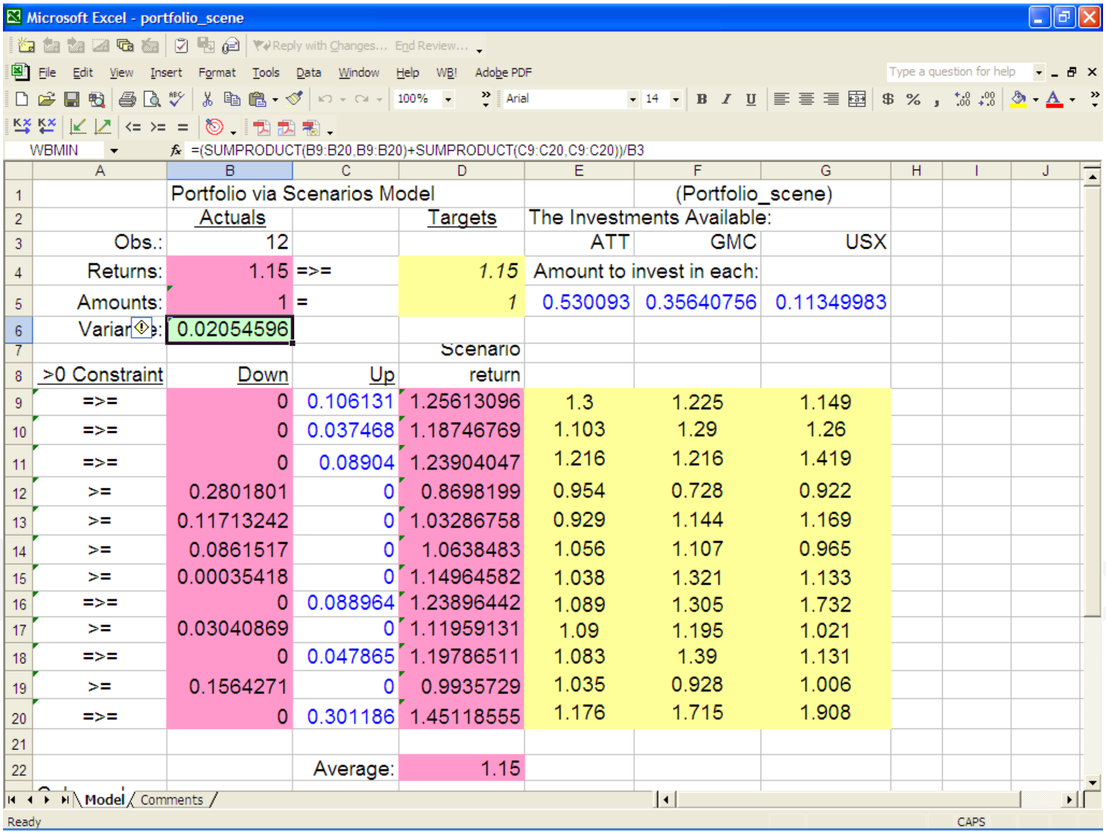
$$\text{Min } \sum_i X_i^2 v_i^2 + \sum_s P_s Y_s^2$$

The key feature is that, even though this formulation has a few more constraints, the covariance matrix is diagonal and, thus, very sparse.

You will generally also want to put upper limits on what fraction of the portfolio is invested in each asset. Otherwise, if there are no upper bounds or inherent variabilities specified, the optimization will tend to invest in only as many assets as there are scenarios.

4.4.7 Example: Scenario Model for Representing Uncertainty

We will use the original data from Markowitz once again. We simply treat each of the 12 years as being a separate scenario, independent of the other 11 years.


Microsoft Excel - portfolio_scene

WBMIN =(SUMPRODUCT(B9:B20,B9:B20)+SUMPRODUCT(C9:C20,C9:C20))/B3

	A	B	C	D	E	F	G
1		Portfolio via Scenarios Model				(Portfolio_scene)	
2		Actuals		Targets	The Investments Available:		
3	Obs.:	12			ATT	GMC	USX
4	Returns:	1.15	=>=	1.15	Amount to invest in each:		
5	Amounts:	1	=	1	0.530093	0.35640756	0.11349983
6	Variance:	0.02054596					
7				Scenario			
8	>0 Constraint	Down	Up	return			
9	=>=	0	0.106131	1.25613096	1.3	1.225	1.149
10	=>=	0	0.037468	1.18746769	1.103	1.29	1.26
11	=>=	0	0.08904	1.23904047	1.216	1.216	1.419
12	>=	0.2801801	0	0.8698199	0.954	0.728	0.922
13	>=	0.11713242	0	1.03286758	0.929	1.144	1.169
14	>=	0.0861517	0	1.0638483	1.056	1.107	0.965
15	>=	0.00035418	0	1.14964582	1.038	1.321	1.133
16	=>=	0	0.088964	1.23896442	1.089	1.305	1.732
17	>=	0.03040869	0	1.11959131	1.09	1.195	1.021
18	=>=	0	0.047865	1.19786511	1.083	1.39	1.131
19	>=	0.1564271	0	0.9935729	1.035	0.928	1.006
20	=>=	0	0.301186	1.45118555	1.176	1.715	1.908
21							
22			Average:	1.15			

The solution should be familiar. The alert reader may have noticed the solution suggests the same portfolio (except for round-off error) as our original model based on the covariance matrix (based on the same 12 years of data as in the above scenario model). This, in fact, is a general result. In other words, if the covariance matrix and expected returns are calculated directly from the original data by the traditional statistical formulae, then the covariance model and the scenario model, based on the same data, will recommend exactly the same portfolio.

The careful reader will have noticed the objective function from the scenario model (0.02056) is slightly less than that of the covariance model (.02241). The exceptionally perceptive reader may have noticed $12 \times 0.02054597/11$ is, except for round-off error, equal to 0.002241. The difference in objective value is a result simply of the fact that standard statistics packages tend to divide by $N - 1$ rather than N when computing variances and covariances, where N is the number of observations. Thus, a slightly more general statement is, if the covariance matrix is computed using a divisor of N rather than $N - 1$, then the covariance model and the scenario model will give the same solution, including objective value.

The crucial formulae are:

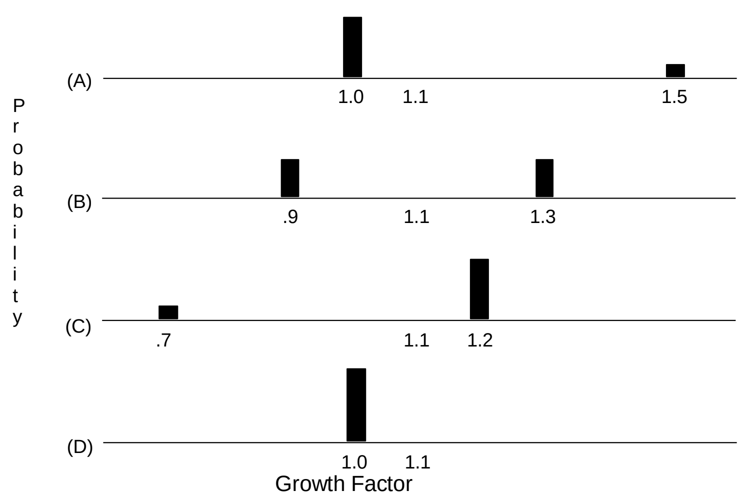
B4=D22,
B5=SUM(E5:G5),
B6=(SUMPRODUCT(B9:B20,B9:B20)+SUMPRODUCT(C9:C20,C9:C20))/B3,
B9=C9-D9+$D$22,
D9=SUMPRODUCT(E9:G9,E$5:G$5),
D22=AVERAGE(D9:D20).

4.5 Measures of Risk other than Variance

The most common measure of risk is variance (or its square root, the standard deviation). This is a reasonable measure of risk for assets that have a symmetric distribution and are traded in a so-called "efficient" market. If these two features do not hold, however, variance has some drawbacks. Consider the four possible growth distributions in Figure 4.2.

Investments *A*, *B*, and *C* are equivalent according to the variance measure because each has an expected growth of 1.10 (an expected return of 10%) and a variance of 0.04 (standard deviation around the mean of 0.20). Risk-averse investors would, however, probably not be indifferent among the three. Under distribution (*A*), you would never lose any of your original investment, and there is a 0.2 probability of the investment growing by a factor of 1.5 (i.e., a 50% return). Distribution (*C*), on the other hand, has a 0.2 probability of an investment decreasing to 0.7 of its original value (i.e., a negative 30% return). Risk-averse investors would tend to prefer (*A*) most and to prefer (*C*) least. This illustrates variance need not be a good measure of risk if the distribution of returns is not symmetric:

Figure 4.2 Possible Growth Factor Distributions



Investment (*D*) is an inefficient investment. It is dominated by (*A*). Suppose the only investments available are (*A*) and (*D*) and our goal is to have an expected return of at least 5% (i.e., a growth factor of 1.05) and the lowest possible variance. The solution is to put 50% of our investment in each of (*A*) and (*D*). The resulting variance is 0.01 (standard deviation = 0.1). If we invested 100% in (*A*), the standard deviation would be 0.20. Nevertheless, we would prefer to invest 100% in (*A*). It is true the return is more random. However, our profits are always at least as high under every outcome. (If the randomness in profits is an issue, we can always give profits to a worthy educational institution when our profits are high to reduce the variance.) Thus, the variance objective may cause us to choose inefficient investments.

In active and efficient markets such as major stock markets, you will tend not to find investments such as (*D*) because investors will realize (*A*) dominates (*D*). Thus, the market price of (*D*) will drop until its return approaches competing investments. In investment decisions regarding new physical facilities, however, there are no strong market forces making all investment candidates "efficient", so the variance risk measure may be less appropriate in such situations.

4.5.1 Utility Functions

A variety of utility functions have been proposed for measuring expected risk. If w is our wealth at the end of the period then the utility function $U(w)$ measures the utility of that wealth. Sensible utility functions have two features: a) they are increasing in w, or at least non-decreasing(more wealth cannot hurt), and b) they are concave(each additional $ of wealth is no more valuable than the previous one, maybe less). Some commonly proposed utility functions are:

1) Downside risk: $U(w) = w - \max(w\text{-}t, 0)$, where t is the threshold,
2) Log: $U(w) = \text{Log}(w)$, sometimes called the Kelly criterion,
3) Quadratic: $U(w) = a*w - b*w^2$,
4) Exponential: $U(w) = -\exp(-a*w)$,
5) Power: $U(w) = w^{(1-r)}/(1-r)$,
6) Hyperbolic: $U(w) = [(1-\gamma)/\gamma]*[a*w/(1-\gamma)+b)]^{\gamma}$.
   The Hyperbolic includes the quadratic, exponential, and power utilities as special cases.

In the next section we set what kind of anomalous situations can arise if we do not use a "sensible" utility function in the above sense.

4.5.2 Maximizing the Minimum Return

A very conservative investor might react to risk by maximizing the minimum return over scenarios. There are some curious implications from this. Suppose the only investments available are A and C above and the two scenarios are:

Scenario	Probability	Payoff from A	Payoff from C
1	0.8	1.0	1.2
2	0.2	1.5	0.7

If we wish to maximize the minimum possible wealth, the probability of a scenario does not matter, as long as the probability is positive. Thus, the following LP is appropriate:

```
    MAX = WMIN;
!  Initial budget constraint;
```

```
                A +       C = 1;
!  Wealth under scenario 1;
       WMIN   <=       A + 1.2 * C > 0;
!  Wealth under scenario 2;
       WMIN  <= 1.5 * A + 0.7 * C > 0;
```

It is not difficult to deduce that the solution is:

```
Variable           Value
    WMIN        1.100000
       A       0.5000000
       C       0.5000000
```

Given that both investments have an expected return of 10%, it is not surprising the expected growth factor is 1.10. That is, a return of 10%. The possibly surprising thing is there is no risk. Regardless of which scenario occurs, the $1 initial investment will grow to $1.10 if 50 cents is placed in each of *A* and *C*.

Now, suppose an extremely reliable friend provides us with the interesting news that, “if scenario 1 occurs, then investment *C* will payoff 1.3 rather than 1.2”. This is certainly good news. The expected return for *C* has just gone up, and its downside risk has certainly not gotten worse. How should we react to it? We make the obvious modification in our model:

```
  MAX = WMIN;
!  Initial budget constraint;
                        A       + C = 1;
!  Wealth under scenario 1;
       WMIN  <=        A + 1.3 * C ;
!  Wealth under scenario 2;
       WMIN  <= 1.5 * A + 0.7 * C ;

and re-solve it to find:
Variable           Value
    WMIN        1.136364
       A       0.5454545
       C       0.4545455
```

This is a bit curious. We have decreased our investment in *C*. This is as if our friend had continued on: “I have this very favorable news regarding stock *C*. Let’s sell it before the market has a chance to react”. Why the anomaly? The problem is we are basing our measure of goodness on a single point among the possible payoffs. In this case, it is the worst possible. For a further discussion of these issues, see Clyman (1995).

4.5.2 Value at Risk

In 1994, J.P. Morgan popularized the "Value at Risk" (VaR) concept with the introduction of their RiskMetrics™ system. To use VaR, you must specify two numbers: 1) an interval of time (e.g., one day) over which you are concerned about losing money, and 2) a probability threshold (e.g., 5%) beyond which you care about harmful outcomes. VaR is then defined as that amount of loss in one day that has at most a 5% probability of being exceeded. A comprehensive survey of VaR is Jorion (2001).

Example

Suppose that one day from now we think that our portfolio will have appreciated in value by $12,000. The actual value, however, has a Normal distribution with a standard deviation of $10,000. From a Normal table, we can determine that a left tail probability of 5% corresponds to an outcome that is 1.644853 standard deviations below the mean. Now:

$$12000 - 1.644853 * 10000 = -4448.50.$$

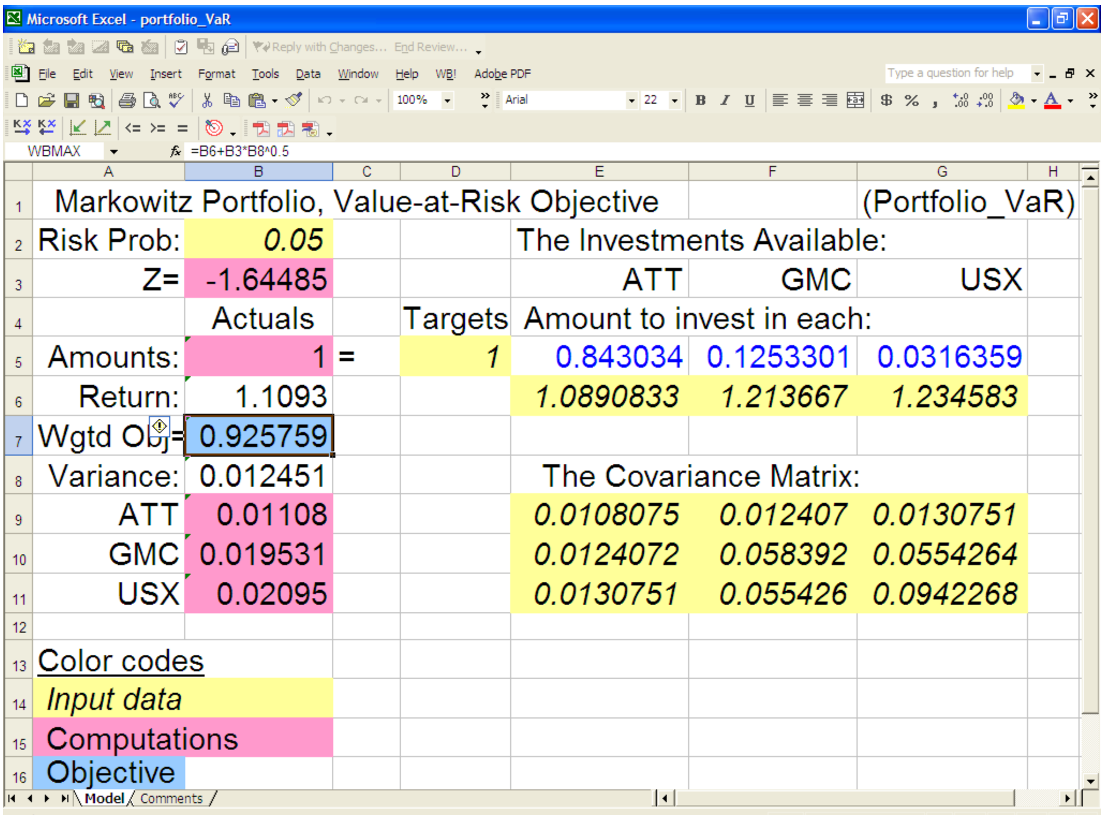
So, we would say that the value at risk is $4448.50.

4.5.3 Example of VaR

Let us apply the VAR approach to our standard example, the ATT/GMC/USC model. Suppose that our time interval of interest is one year and our risk tolerance is 5% and we want to minimize the value at risk of our portfolio. This is equivalent to maximizing that threshold, so the probability our wealth is below this threshold is at most .05.

Analysis:

A left tail probability of 5% corresponds to the probability threshold. We want to consider the point that is 1.64485 standard deviations below the mean. Minimizing the value at risk corresponds to choosing the mean and standard deviation of the portfolio, so the (mean – 1.64485 * (standard deviation)) is maximized. The following model will do this:



Microsoft Excel - portfolio_VaR

WBMAX =B6+B3*B8^0.5

	A	B	C	D	E	F	G
1	Markowitz Portfolio, Value-at-Risk Objective						(Portfolio_VaR)
2	Risk Prob:	0.05			The Investments Available:		
3	Z=	-1.64485			ATT	GMC	USX
4		Actuals		Targets	Amount to invest in each:		
5	Amounts:	1	=	1	0.843034	0.1253301	0.0316359
6	Return:	1.1093			1.0890833	1.213667	1.234583
7	Wgtd Obj=	0.925759					
8	Variance:	0.012451			The Covariance Matrix:		
9	ATT	0.01108			0.0108075	0.012407	0.0130751
10	GMC	0.019531			0.0124072	0.058392	0.0554264
11	USX	0.02095			0.0130751	0.055426	0.0942268
12							
13	Color codes						
14	Input data						
15	Computations						
16	Objective						

Model / Comments

Ready

Note that, if we invested solely in ATT, the portfolio variance would be .01080754. So, the standard deviation would be .103959, and the VAR would be 1 - (1.089083 - 1.644853 * .103959) = .0818.

The portfolio is efficient because it is maximizing a weighted combination of the expected return and (a negatively weighted) standard deviation. Thus, if there is a portfolio that has both higher expected return and lower standard deviation, then the above solution would not maximize the objective function above.

Note, if you use: *PROB* = .1988, you get essentially the original portfolio considered for the ATT/GMC/USX problem.

The crucial formulae are:

B3==NORMSINV(B2),
B5=SUM(E5:G5),
B7=B6+B3*B8^0.5,
B9=SUMPRODUCT(E9:G9,E$5:G$5)
C5=WB(B5,"=",D5).

4.6 Scenario Model and Minimizing Downside Risk

Minimizing the variance in return is appropriate if either:

1) the actual return is Normal-distributed or

2) the portfolio owner has a quadratic utility function.

In practice, it is difficult to show either condition holds. Thus, it may be of interest to use a more intuitive measure of risk. One such measure is the downside risk, which intuitively is the expected amount by which the return is less than a specified target return. The approach can be described if we define:

T = user specified target threshold. When risk is disregarded, this is typically less than the maximum expected return and greater than the return under the worst scenario.

Y_s = amount by which the return under scenario s falls short of target.

$= \max\{0, T - \sum X_i u_{is}\}$

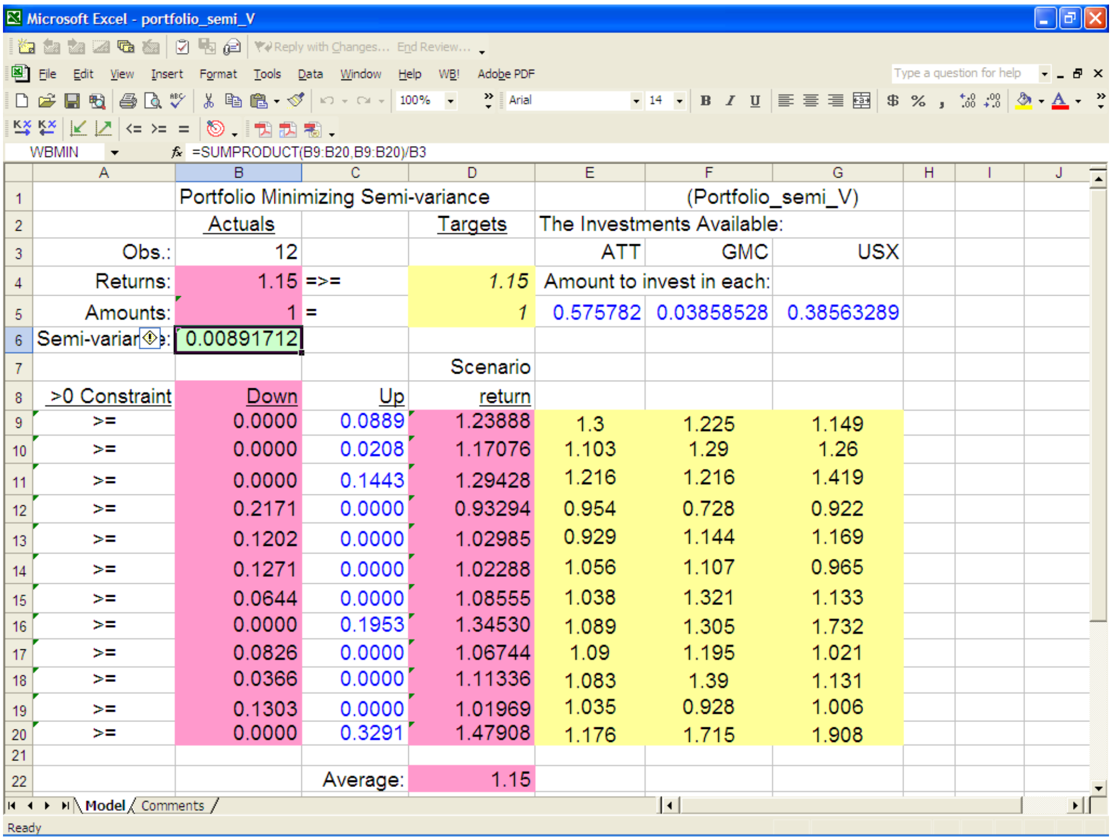
The model in algebraic form is then:

Min $\sum P_s Y_s$! Minimize expected downside risk
subject to
(compute deviation below target of each scenario, s):
$Y_s - T + \sum X_i u_{is} \geq 0$
$\sum X_i = 1$! (budget constraint)
$\sum X_i \sum P_s u_{is} \geq r$! (desired return).

Notice this is just a linear program.

4.6.1 Semi-variance and Downside Risk

The most common alternative suggested to variance as a measure of risk is some form of downside risk. One such measure is semi-variance. It is essentially variance, except only deviations below the mean are counted as risk. The scenario model is well suited to such measures. The previous scenario model needs only a slight modification to convert it to a semi-variance model.


Microsoft Excel - portfolio_semi_V

WBMIN =SUMPRODUCT(B9:B20,B9:B20)/B3

	A	B	C	D	E	F	G
1		Portfolio Minimizing Semi-variance				(Portfolio_semi_V)	
2		Actuals		Targets	The Investments Available:		
3	Obs.:	12			ATT	GMC	USX
4	Returns:	1.15	=>=	1.15	Amount to invest in each:		
5	Amounts:	1	=	1	0.575782	0.03858528	0.38563289
6	Semi-variance:	0.00891712					
7				Scenario			
8	>0 Constraint	Down	Up	return			
9	>=	0.0000	0.0889	1.23888	1.3	1.225	1.149
10	>=	0.0000	0.0208	1.17076	1.103	1.29	1.26
11	>=	0.0000	0.1443	1.29428	1.216	1.216	1.419
12	>=	0.2171	0.0000	0.93294	0.954	0.728	0.922
13	>=	0.1202	0.0000	1.02985	0.929	1.144	1.169
14	>=	0.1271	0.0000	1.02288	1.056	1.107	0.965
15	>=	0.0644	0.0000	1.08555	1.038	1.321	1.133
16	>=	0.0000	0.1953	1.34530	1.089	1.305	1.732
17	>=	0.0826	0.0000	1.06744	1.09	1.195	1.021
18	>=	0.0366	0.0000	1.11336	1.083	1.39	1.131
19	>=	0.1303	0.0000	1.01969	1.035	0.928	1.006
20	>=	0.0000	0.3291	1.47908	1.176	1.715	1.908
21							
22			Average:	1.15			

Notice the objective value is less than half that of the variance model. We would expect it to be at most half, because it considers only the down (not the up) deviations. The most noticeable change in the portfolio is substantial funds have been moved to *USX* from *GMC*. This is not surprising if you look at the original data. In the years in which *ATT* performs poorly, *USX* tends to perform better than *GMC*.

The formulae and constraints are essentially as with the model Portfolio_scene, except for the objective cell.

The crucial formulae are:

B4=D22,
B5=SUM(E5:G5),

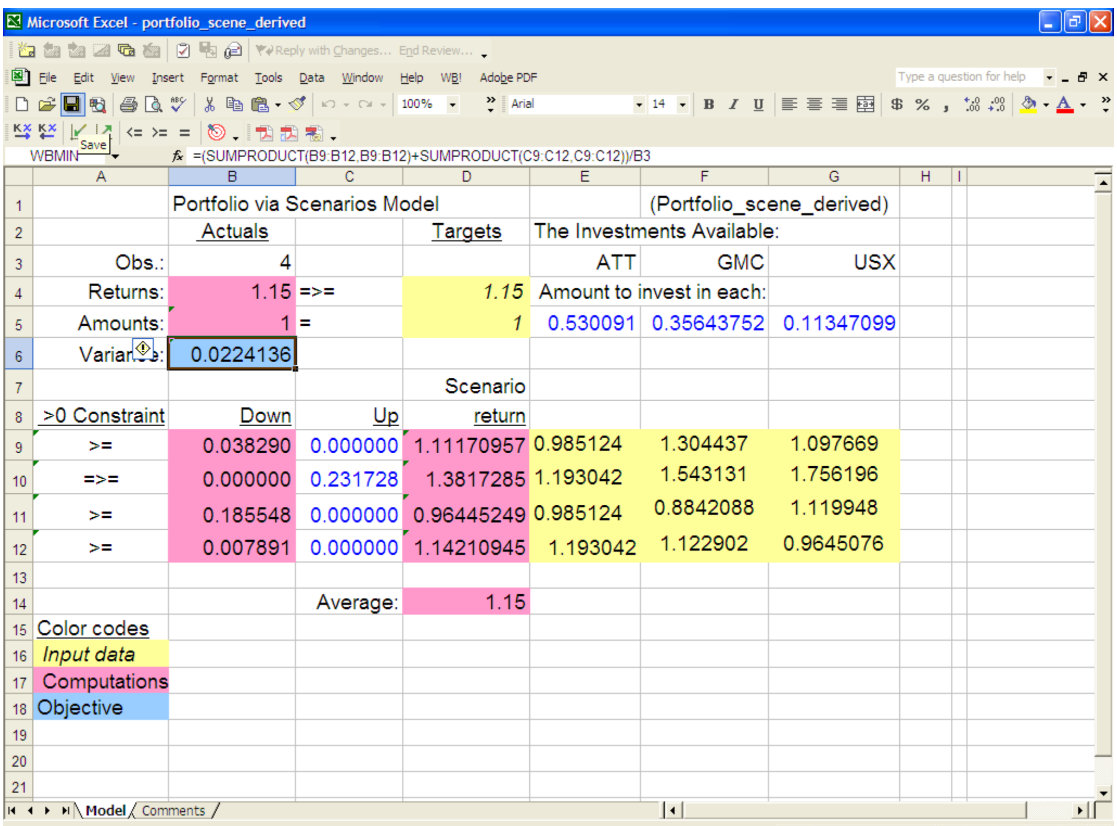
B6=SUMPRODUCT(B9:B20,B9:B20)/B3,
B9=C9-D9+$D$22,
D9=SUMPRODUCT(E9:G9,E$5:G$5),
D22=AVERAGE(D9:D20).

4.6.2 Downside Risk and MAD

If the threshold for determining downside risk is the mean return, then minimizing the downside risk is equivalent to minimizing the mean absolute deviation (MAD) about the mean. This follows easily because the sum of deviations (not absolute) about the mean must be zero. Thus, the sum of deviations above the mean equals the sum of deviations below the mean. Therefore, the sum of absolute deviations is always twice the sum of the deviations below the mean. Thus, minimizing the downside risk below the mean gives exactly the same recommendation as minimizing the sum of absolute deviations below the mean. Konno and Yamazaki (1991) use the MAD measure to construct portfolios from stocks on the Tokyo stock exchange.

4.6.3 Scenarios Based Directly Upon a Covariance Matrix

If only a covariance matrix is available, rather than original data, then, not surprisingly, it is nevertheless possible to construct scenarios that match the covariance matrix. The following example uses just four scenarios to represent the possible returns from the three assets: ATT, GMC, and USX. These scenarios have been constructed, using the methods of section 2.8.2, so they mimic behavior consistent with the original covariance matrix:



	A	B	C	D	E	F	G
1		Portfolio via Scenarios Model				(Portfolio_scene_derived)	
2		Actuals		Targets	The Investments Available:		
3	Obs.:	4			ATT	GMC	USX
4	Returns:	1.15	=>=	1.15	Amount to invest in each:		
5	Amounts:	1	=	1	0.530091	0.35643752	0.11347099
6	Variance:	0.0224136					
7				Scenario			
8	>0 Constraint	Down	Up	return			
9	>=	0.038290	0.000000	1.11170957	0.985124	1.304437	1.097669
10	=>=	0.000000	0.231728	1.3817285	1.193042	1.543131	1.756196
11	>=	0.185548	0.000000	0.96445249	0.985124	0.8842088	1.119948
12	>=	0.007891	0.000000	1.14210945	1.193042	1.122902	0.9645076
13							
14			Average:	1.15			
15	Color codes						
16	Input data						
17	Computations						
18	Objective						

Notice the objective function value and the allocation of funds over *ATT*, *GMC*, and *USX* are essentially identical to our original portfolio example.

The crucial formulae are:

```
B4=D14
C4=WB(B4,">=",D4)
B5=SUM(E5:G5)
C5=WB(B5,"=",D5)
B6=(SUMPRODUCT(B9:B12,B9:B12)+SUMPRODUCT(C9:C12,C9:C12))/B3
B9=C9-D9+$D$14
D9=SUMPRODUCT(E9:G9,E$5:G$5)
D14=AVERAGE(D9:D12)
```

4.7 Hedging, Matching and Program Trading

4.7.1 Portfolio Hedging

Given a “benchmark” portfolio *B*, we say we hedge *B* if we construct another portfolio *C* such that, taken together, *B* and *C* have essentially the same return as *B*, but lower risk than *B*. Typically, our portfolio *B* contains certain components that cannot be removed. Thus, we want to buy some components negatively correlated with the existing ones. Examples are:

a) An airline knows it will have to purchase a lot of fuel in the next three months. It would like to be insulated from unexpected fuel price increases.
b) A farmer is confident his fields will yield $200,000 worth of corn in the next two months. He is happy with the current price for corn. Thus, would like to “lock in” the current price.

4.7.2 Portfolio Matching, Tracking, and Program Trading

Given a benchmark portfolio *B*, we say we construct a matching or tracking portfolio if we construct a new portfolio *C* that has stochastic behavior very similar to ***B*,** but excludes certain instruments in *B*. Example situations are:

a) A portfolio manager does not wish to look bad relative to some well-known index of performance such as the S&P 500, but for various reasons cannot purchase certain instruments in the index.
b) An arbitrageur with the ability to make fast, low-cost trades wants to exploit market inefficiencies (i.e., instruments mispriced by the market). If he can construct a portfolio that perfectly matches the future behavior of the well-defined portfolio, but costs less today, then he has an arbitrage profit opportunity (if he can act before this “mispricing” disappears).
c) A retired person is concerned mainly about inflation risk. In this case, a portfolio that tracks inflation is desired.

As an example of (a), a certain so-called “green” mutual fund will not include in its portfolio companies that derive more than 2% of their gross revenues from the sale of military weapons, own directly or operate nuclear power plants, or participate in business related to the nuclear fuel cycle.

The following table, for example, compares the performance of six Vanguard portfolios with the indices the portfolios were designed to track; see Vanguard (1995):

Total Return Six Months Ended June 30, 1995

Vanguard Portfolio Name	Portfolio Growth	Comparative Growth Index	Index Name
500 Portfolio	+20.1%	+20.2%	S&P500
Growth Portfolio	+21.1	+21.2	S&P500/BARRA Growth
Value Portfolio	+19.1	+19.2	S&P500/BARRA Value
Extended Market Portfolio	+17.1%	+16.8%	Wilshire 4500 Index
SmallCap Portfolio	+14.5	+14.4	Russell 2000 Index
Total Stock Market Portfolio	+19.2%	+19.2%	Wilshire 5000 Index

Notice, even though there is substantial difference in the performance of the portfolios, each matches its benchmark index quite well.

4.8 Methods for Constructing Benchmark Portfolios

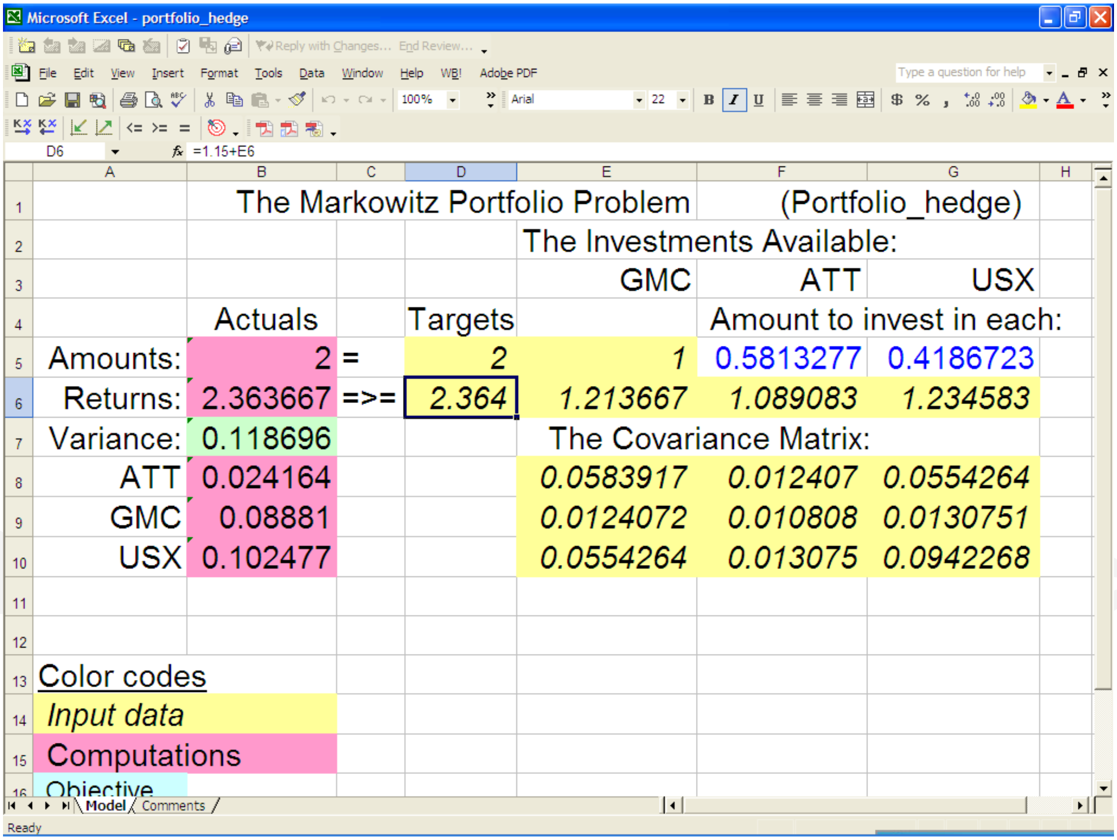
A variety of approaches has been used for constructing hedging and matching portfolios. For matching portfolios, an intuitive approach has been to generalize the Markowitz model, so the objective is to minimize the variance in the difference in return between the target portfolio and the tracking portfolio.

A useful way to think about hedging or matching of a benchmark is to think of it as our being forced to include the benchmark or its negative in our portfolio. Suppose the benchmark is a simple index such as the S&P500. If our measure of risk is variance, then proceed as follows:

1. Include the benchmark in the covariance matrix just like any other instrument, except do not include it in the budget constraint. We presume we have a budget of $1 to invest in the controllable, non-benchmark portion of our portfolio.
2. To get a “matching” portfolio (e.g., one that mimics the S&P 500), set the value of the benchmark factor to −1. The essential effect is the off diagonal covariance terms are negated in the row/column of the benchmark factor. Effectively, we have shorted the factor. If we can get the total variance to zero, we have perfectly matched the randomness of the benchmark.
3. To get a “hedging” portfolio (e.g., one as negatively correlated with the S&P 500 as possible), set the value of the benchmark factor to +1. Thus, we will compose the rest of the portfolio to counteract the effect of the factor we are stuck with having in the portfolio.

One might even want to drop the budget constraint. The solution will then tell you how much to invest in the controllable portfolio to get the best possible hedge or match per $ of the benchmark.

The following model illustrates the extension of the Markowitz approach to the hedging case where we want to “cancel out” some benchmark. In the case of *GMC*, it could be that our decision maker works for *GMC* and thus has his fortunes unavoidably tied to those of *GMC*. He might wish to find a portfolio negatively correlated with *GMC*:



Microsoft Excel - portfolio_hedge

D6 =1.15+E6

	A	B	C	D	E	F	G
1		The Markowitz Portfolio Problem				(Portfolio_hedge)	
2					The Investments Available:		
3					GMC	ATT	USX
4		Actuals		Targets		Amount to invest in each:	
5	Amounts:	2	=	2	1	0.5813277	0.4186723
6	Returns:	2.363667	=>=	2.364	1.213667	1.089083	1.234583
7	Variance:	0.118696			The Covariance Matrix:		
8	ATT	0.024164			0.0583917	0.012407	0.0554264
9	GMC	0.08881			0.0124072	0.010808	0.0130751
10	USX	0.102477			0.0554264	0.013075	0.0942268
11							
12							
13	Color codes						
14	Input data						
15	Computations						
16	Objective						

Model / Comments

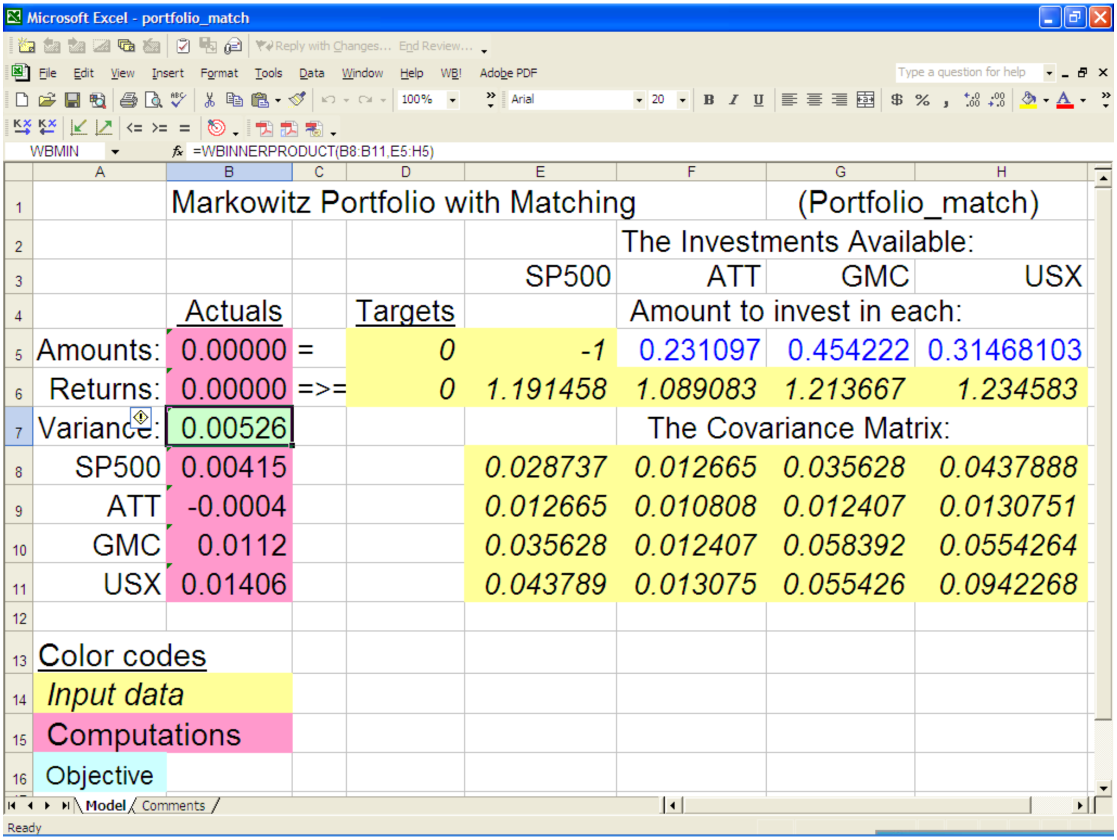
Ready

Thus, our investor puts more of the portfolio in *ATT* than in *USX* (whose fortunes are more closely tied to those of *GMC*).

The crucial formulae are:

B5=SUM(E5:G5),
C5=WB(B5,"=",D5),
B6=SUMPRODUCT(E6:G6,E5:G5),
B7=WBINNERPRODUCT(B8:B10,E5:G5),
B8=SUMPRODUCT(E9:G9,E$5:G$5)

The following model illustrates the extension of the Markowitz approach to the matching case where we want to construct a portfolio that *mimics or matches* a benchmark portfolio. In this case, we want to match the S&P500, but limit ourselves to investing in only *ATT*, *GMC*, and *USX*.

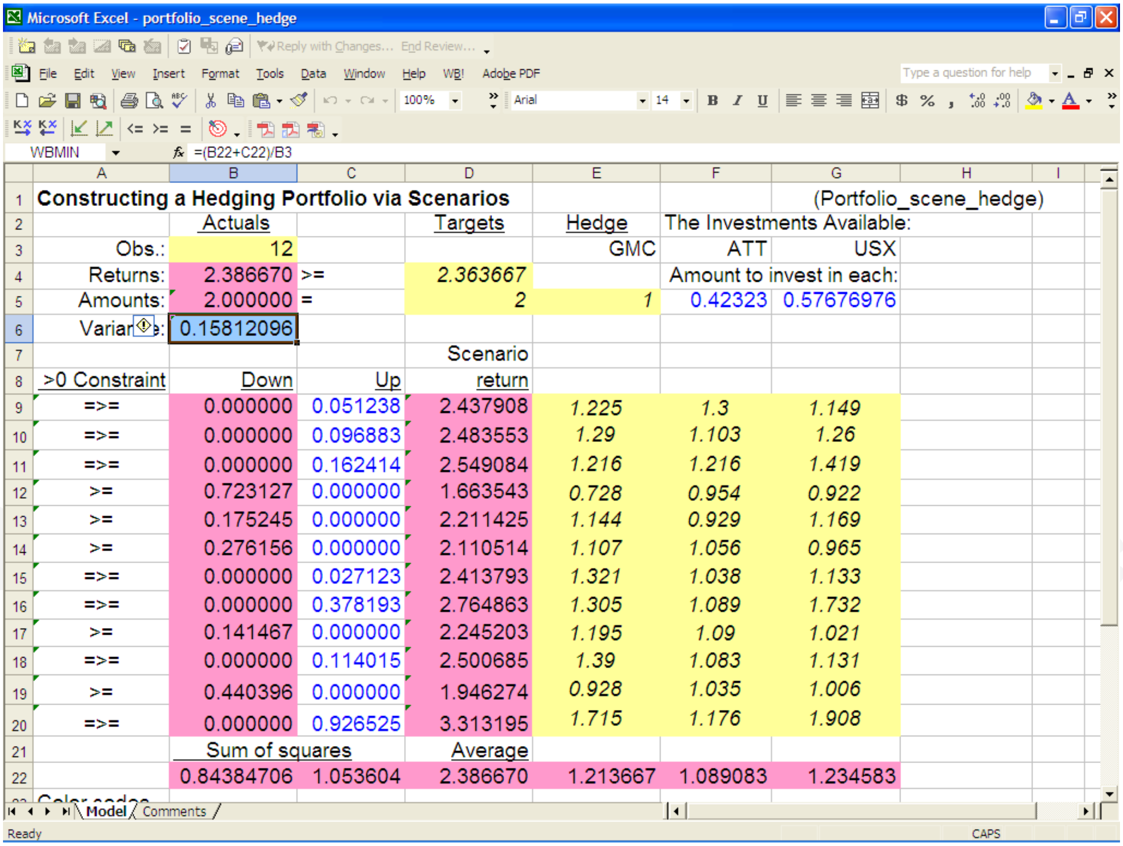


	A	B	C	D	E	F	G	H
1		Markowitz Portfolio with Matching					(Portfolio_match)	
2						The Investments Available:		
3					SP500	ATT	GMC	USX
4		Actuals		Targets		Amount to invest in each:		
5	Amounts:	0.00000	=	0	-1	0.231097	0.454222	0.31468103
6	Returns:	0.00000	=>=	0	1.191458	1.089083	1.213667	1.234583
7	Variance:	0.00526				The Covariance Matrix:		
8	SP500	0.00415			0.028737	0.012665	0.035628	0.0437888
9	ATT	-0.0004			0.012665	0.010808	0.012407	0.0130751
10	GMC	0.0112			0.035628	0.012407	0.058392	0.0554264
11	USX	0.01406			0.043789	0.013075	0.055426	0.0942268
12								
13	Color codes							
14	Input data							
15	Computations							
16	Objective							

The formulae in the matching model are the same as in the hedging model. The only difference is in the data entered.

4.8.1 Scenario Approach to Benchmark Portfolios

The scenario approach can be used for constructing hedging and matching portfolios in much the same way as the classical Markowitz model was used. The following model tries to construct a hedge relative to GMC from ATT and USX.

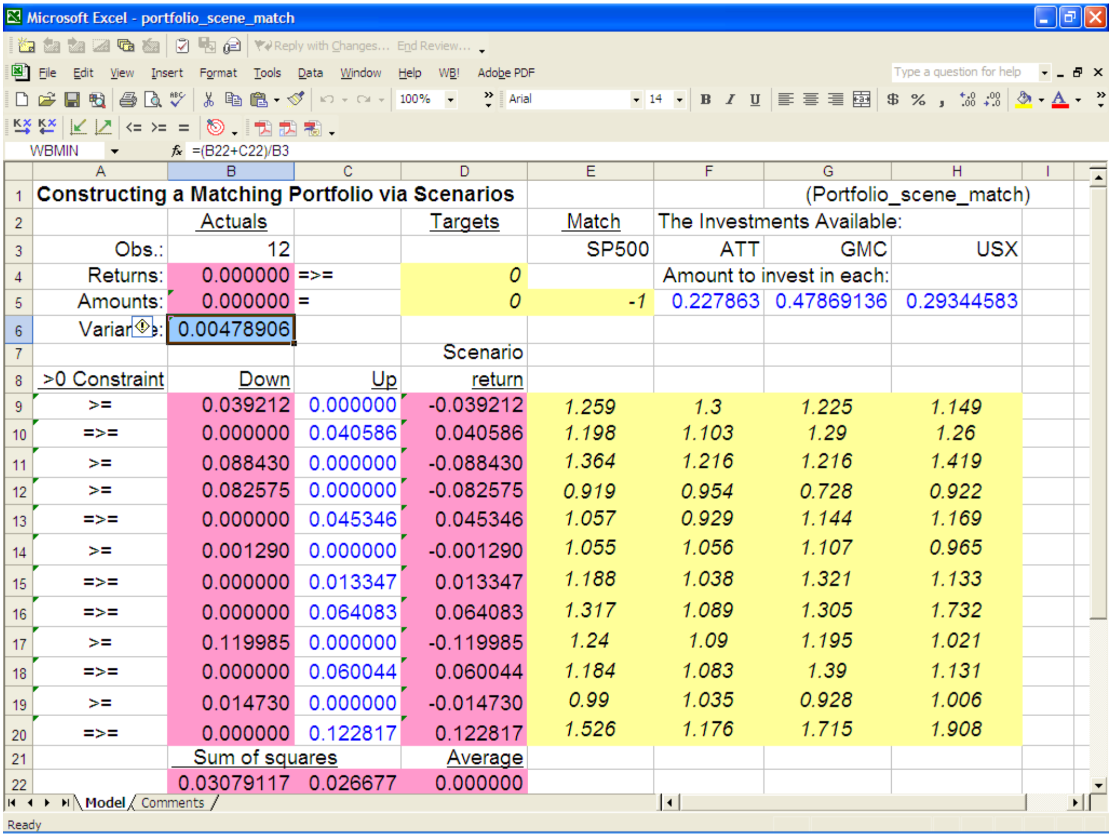


	A	B	C	D	E	F	G
1	Constructing a Hedging Portfolio via Scenarios						(Portfolio_scene_hedge)
2		Actuals		Targets	Hedge	The Investments Available:	
3	Obs.:	12			GMC	ATT	USX
4	Returns:	2.386670	>=	2.363667		Amount to invest in each:	
5	Amounts:	2.000000	=	2	1	0.42323	0.57676976
6	Variance:	0.15812096					
7				Scenario			
8	>0 Constraint	Down	Up	return			
9	=>=	0.000000	0.051238	2.437908	1.225	1.3	1.149
10	=>=	0.000000	0.096883	2.483553	1.29	1.103	1.26
11	=>=	0.000000	0.162414	2.549084	1.216	1.216	1.419
12	>=	0.723127	0.000000	1.663543	0.728	0.954	0.922
13	>=	0.175245	0.000000	2.211425	1.144	0.929	1.169
14	>=	0.276156	0.000000	2.110514	1.107	1.056	0.965
15	=>=	0.000000	0.027123	2.413793	1.321	1.038	1.133
16	=>=	0.000000	0.378193	2.764863	1.305	1.089	1.732
17	>=	0.141467	0.000000	2.245203	1.195	1.09	1.021
18	=>=	0.000000	0.114015	2.500685	1.39	1.083	1.131
19	>=	0.440396	0.000000	1.946274	0.928	1.035	1.006
20	=>=	0.000000	0.926525	3.313195	1.715	1.176	1.908
21		Sum of squares		Average			
22		0.84384706	1.053604	2.386670	1.213667	1.089083	1.234583

The crucial formulae are:

B4=D22,
C4=wb(B4,">=",D4),
B5=SUM(E5:G5),
C5=wb(B5,"=",D5),
B6=(B22+C22)/B3,
B9=C9-D9+$D$22,
D9=SUMPRODUCT(E9:G9,E$5:G$5),
B22=SUMPRODUCT(B9:B20,B9:B20),
C22=SUMPRODUCT(C9:C20,C9:C20),
D22=AVERAGE(D9:D20),
E22=AVERAGE(E9:E20).

The following is a scenario model for constructing a portfolio matching the S&P500:



	A	B	C	D	E	F	G	H
1	Constructing a Matching Portfolio via Scenarios						(Portfolio_scene_match)	
2		Actuals		Targets	Match	The Investments Available:		
3	Obs.:	12			SP500	ATT	GMC	USX
4	Returns:	0.000000	=>=	0		Amount to invest in each:		
5	Amounts:	0.000000	=	0	-1	0.227863	0.47869136	0.29344583
6	Variance:	0.00478906						
7				Scenario				
8	>0 Constraint	Down	Up	return				
9	>=	0.039212	0.000000	-0.039212	1.259	1.3	1.225	1.149
10	=>=	0.000000	0.040586	0.040586	1.198	1.103	1.29	1.26
11	>=	0.088430	0.000000	-0.088430	1.364	1.216	1.216	1.419
12	>=	0.082575	0.000000	-0.082575	0.919	0.954	0.728	0.922
13	=>=	0.000000	0.045346	0.045346	1.057	0.929	1.144	1.169
14	>=	0.001290	0.000000	-0.001290	1.055	1.056	1.107	0.965
15	=>=	0.000000	0.013347	0.013347	1.188	1.038	1.321	1.133
16	=>=	0.000000	0.064083	0.064083	1.317	1.089	1.305	1.732
17	>=	0.119985	0.000000	-0.119985	1.24	1.09	1.195	1.021
18	=>=	0.000000	0.060044	0.060044	1.184	1.083	1.39	1.131
19	>=	0.014730	0.000000	-0.014730	0.99	1.035	0.928	1.006
20	=>=	0.000000	0.122817	0.122817	1.526	1.176	1.715	1.908
21		Sum of squares		Average				
22		0.03079117	0.026677	0.000000				

Notice that we get the same portfolio as with the Markowitz model.

The two scenario models both used variance for the measure of risk relative to the benchmark. It is easy to modify them, so more asymmetric risk measures, such as downside risk, could be used.

The formulae in this model are the same as in the previous.

4.8.2 Efficient Benchmark Portfolios

We say a portfolio is on the efficient frontier if there is no other portfolio that has both higher expected return and lower risk.

Let:

r_i = expected return on asset i,
t = an arbitrary target return for the portfolio.

A portfolio, with weight m_i on asset i, is efficient if there exists some target t for which the portfolio is a solution to the problem:

Minimize risk
subject to

$$\sum_{i=0}^{n} m_i = 1 \qquad \text{(budget constraint)}$$

$$\sum_{i=0}^{m} r_i\, m_i \geq t \qquad \text{(return target constraint).}$$

Portfolio managers are frequently evaluated on their performance relative to some benchmark portfolio. Let b_i = the weight on asset i in the benchmark portfolio. If the benchmark portfolio is not on the efficient frontier, then an interesting question is: What are the weights of the portfolio on the efficient frontier that is closest to the benchmark portfolio in the sense that the risk of the efficient portfolio relative to the benchmark is minimized?

There is a particularly simple answer when the measure of risk is portfolio variance, there is a risk-free asset, borrowing is allowed at the risk-free rate, and short sales are permitted. Let m_0 = the weight on the risk-free asset. An elegant result, in this case, is that there is a so-called “market” portfolio with weights mi on asset i, such that effectively only m_0 varies as the return target varies. Specifically, there are constants m_i, for i = 1, 2, . . . , n, such that the weight on asset i is simply $(1 - m_0) \times m_i$. Define:

$q = 1 - m_0$ = weight to put on the market portfolio,
R_i = random return on asset i.

Then the variance of any efficient portfolio relative to the benchmark portfolio can be written as:

$$\text{var}\Big(\sum_{i=1}^{n} R_i[q^*m_i - b_i]\Big)$$

$$= \sum_{i=1}^{n} (q^*m_i - b_i)^2 \,\text{var}\,(R_i) + 2 \sum_{j>i} (q^*m_i - b_i)(q^*m_j - b_j)\,\text{Cov}(R_i,R_j).$$

Setting the derivative of this expression with respect to q equal to zero gives the result:

$$q = \frac{\sum_{i=1}^{n} m_i\, {}^*b_i \,\text{var}\,(R_i) + \sum_{j>i} (m_i{}^*b_j{}^*m_j{}^*b_i)\,\text{Cov}\,(R_i, R_j)}{\sum_{i=1}^{n} m_i^2 \,\text{var}\,(R_i) + 2 \sum_{j>i} m_i\, m_j \,\text{Cov}\,(R_i, R_j)}$$

For example, if the benchmark portfolio is on the efficient frontier with weight b_0 on the risk-free asset, then $b_i = (1 - b_0)m_i$ and therefore $q = 1 - b_0$. Thus, a manager who is told to outperform the benchmark portfolio $\{b_0, b_1, . . ., b_n\}$ should perhaps, in fact, be compensated according to his performance relative to the efficient portfolio given by q above.

4.9 Project Portfolios

Some organizations use a yearly budgeting process to select which projects to pursue in the coming year. Examples of projects might be: which crude oil fields to develop for a petroleum exploration firm, which drugs to develop for a pharmaceutical firm, and which types of markets and technologies to pursue for a telecommunications firm. Many of the ideas underlying the portfolio models considered thus far also apply to the project selection portfolio problem. For example, an overall budget may be set at the beginning of the planning exercise for how much can be invested in new projects this year. The major differences distinguishing the project portfolio problem are: a) the investment variables are 0/1, "go/no go" decision variables, b) it is much less obvious how one develops the covariance or correlation matrix describing the project and interproject risks, and c) there may be logical constraints among the projects, typically of an "either-or" nature or an "if we do project A we must do project B" flavor. Consider the following.

Example

The BTT communications company has six projects it is considering for the coming year.

Project Tech1 is a technology development project that requires an initial investment of \$1.9M and has an expected value of \$2.36M after one year. The standard deviation in the value after one year is \$.37M

Project Tech2 is an alternative to Tech1. It requires an initial investment of \$2.5M and has an expected value of \$3.1M after one year. The standard deviation in the value after one year is \$.39M.

Project Ads is an advertising campaign for a certain metropolitan area for a new kind of call handling service. This service has already been introduced on a trial basis in some regions of the city. It requires an initial investment of \$1.7M and has an expected value of \$1.5M after one year. The standard deviation in the value after one year is \$.3M. Note that its incremental return is negative, so that it does not appear worthwhile until we consider projects Regn1, Regn2, and Regn3.

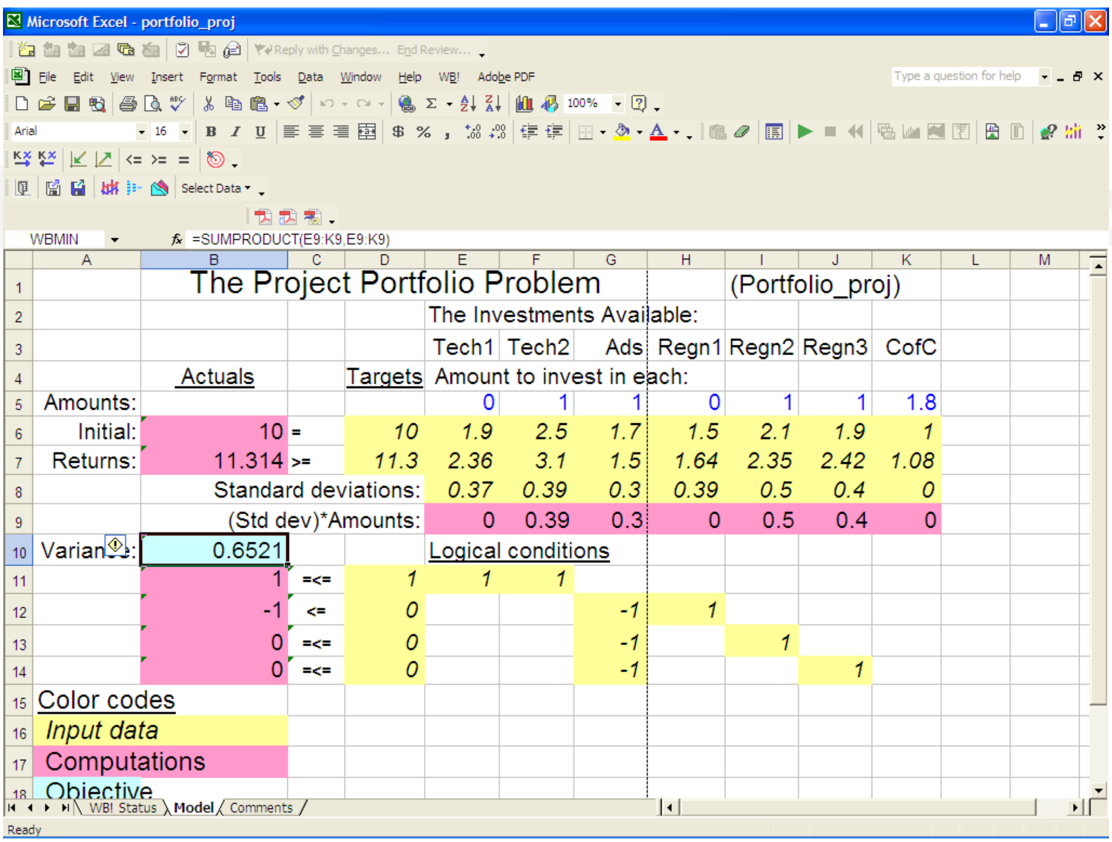
Project Regn1 is the project to install the new call handling capability into Region 1. It requires an initial investment of \$1.5M and has an expected value of \$1.64M after one year. The standard deviation in the value after one year is \$.39M. Note, this expected return for Regn1 is based on the assumption that the major metropolitan advertising campaign, project Ads above, for the call handling service will be undertaken, else project Regn1 will not be worthwhile.

Project Regn2 is similar to Regn1, except it applies to region 2. Regn2 requires an initial investment of \$2.1M and has an expected value of \$2.35M after one year. The standard deviation in the value after one year is \$.5M. This expected return for Regn2 is based on the assumption that the major metropolitan advertising campaign for the call handling service will be undertaken, else project Regn2 will not be worthwhile.

Project Regn3 is similar to Regn1, except it applies to region 3. Regn3 requires an initial investment of \$1.9M and has an expected value of \$2.42M after one year. The standard deviation in the value after one year is \$.4M. This expected return for Regn3 is based on the assumption that the major metropolitan advertising campaign for the call handling service will be undertaken, else project Regn3 will not be worthwhile.

BTT has available a budget of \$10M to invest in these projects. Either because of the lumpiness of the project, or perhaps for other reasons, we may not wish to use exactly \$10M. How should we treat any left over funds? If we are borrowing the money, then we should simply apply the borrowing rate to these left over funds because we avoid the interest payment. Alternatively, we may have other standard investments with fairly reliable returns in which left over funds are invested. For BTT, this "Cost of Capital" rate is 8%. It is represented in the model as the investment "CofC". Suppose that after one year, BTT would like its investment to have an expected return of 13%. This means BTT would like the \$10M budget to grow to a value \$11.3 after one year.

Which projects should be undertaken? The following spreadsheet illustrates the model and the suggested solution.



Microsoft Excel - portfolio_proj

WBMIN =SUMPRODUCT(E9:K9,E9:K9)

	A	B	C	D	E	F	G	H	I	J	K
1		The Project Portfolio Problem							(Portfolio_proj)		
2					The Investments Available:						
3					Tech1	Tech2	Ads	Regn1	Regn2	Regn3	CofC
4		Actuals		Targets	Amount to invest in each:						
5	Amounts:				0	1	1	0	1	1	1.8
6	Initial:	10	=	10	1.9	2.5	1.7	1.5	2.1	1.9	1
7	Returns:	11.314	>=	11.3	2.36	3.1	1.5	1.64	2.35	2.42	1.08
8		Standard deviations:			0.37	0.39	0.3	0.39	0.5	0.4	0
9		(Std dev)*Amounts:			0	0.39	0.3	0	0.5	0.4	0
10	Variance:	0.6521			Logical conditions						
11		1	=<=	1	1	1					
12		-1	<=	0			-1	1			
13		0	=<=	0			-1		1		
14		0	=<=	0			-1			1	
15	Color codes										
16	Input data										
17	Computations										
18	Objective										

WB! Status / Model / Comments

Ready

The solution suggests that we should invest in projects Tech2, Ads, Regn2, and Regn3 and leave 1.8 million in the Cost of Capital fund. This solution has some interesting features. For example, the rate of return for Tech1 is (2.36 – 1.9)/ 1.9 = .2421, whereas the return on Tech2 is (3.1 – 2.5)/2.5 = 1.24. So Tech1 has a slightly higher return, and Tech1 has lower risk, .37, than Tech2, .39. Nevertheless, Tech2 was chosen over the alternative Tech1. Why? The key is that Tech2 allows us to invest more money at a very good rate. If we invested in Tech1 rather than Tech2, where would we invest the 2.5 – 1.9 million dollars that would become available? The obvious place would be in the CofC fund. But there it only earns an incremental return of .08, vs. the .24 return it would earn in Tech2.

The "ABC's of Optimization" for this model are:

A) The adjustable cells in this model are E5:K5. Cells E5:J5 are declared to be 0/1 or binary variables, whereas the investment of surplus funds in CofC is left as a continuous variable.

B) The "Best" or objective cell, to be minimized, is the variance computed in cell B10 by the formula: =SUMPRODUCT(E9:K9,E9:K9).

C) The constraints are computed essentially by the formulae in column B, e.g.
B6= SUMPRODUCT(E6:K6,$E$5:$K$5),
B7= SUMPRODUCT(E7:K7,$E$5:$K$5),
B11= SUMPRODUCT(E11:K11,$E$5:$K$5),

4.9.1 Implementation Issues

The above simple model requires the estimation of three data for each project: a) initial investment, b) expected value after one period, and c) standard deviation in value after one period. Typically, each project in an organization will have a "champion" or supporter. This person may be the best informed person for estimating the above data. The "champion" of a project, however, has an incentive to try to get his project funded this year and worry later about justifying the project if things do not turn out well. Thus, the "champion" will tend to underestimate the initial investment required, overestimate the expected return, and underestimate the expected risk. Thus, you also need an auditor, referee, or arbitrator who can examine the submitted data and try to keep it as unbiased as possible.

The above model approximates the risk only by a standard deviation for each project. It does not include any covariance risk among projects. Our reasoning in this regard is that it is difficult enough to provide an estimate of the standard deviation of a random variable for which we have no historical data. One way of trying to elicit the an estimate of the standard deviation is to assume returns are Normal distributed, in which case, the probability that a return is one standard deviation below the expected value is about one chance in six. Thus, one could ask someone who is knowledgeable about a project: "How much worse the could the value of the project be, so that there is one chance in six of the project doing this poorly?". Treat this difference as one standard deviation.

4.10 Problems

1. You are considering three stocks, IBM, GM, and Georgia-Pacific (GP), for your stock portfolio. The covariance matrix of the yearly percentage returns on these stocks is estimated to be:

	IBM	**GM**	**GP**
IBM	10	3.5	1
GM	3.5	4	1.5
GP	1	1.5	9

Thus, if equal amounts were invested in each, the variance would be proportional to 10 + 4 + 9 + 2 (2.5 + 1 + 1.5). The predicted yearly percentage returns for *IBM*, *GM*, and *GP* are 9, 6 and 5, respectively. Find a minimum variance portfolio of these three stocks for which the yearly return is at least 7, at most 80% of the portfolio is invested in *IBM*, and at least 10% is invested in *GP*.

2. Modify your formulation of problem 1 to incorporate the fact that your current portfolio is 50% IBM and 50% GP. Further, transaction costs on a buy/sell transaction are 1% of the amount traded.

3. The manager of an investment fund hypothesizes that three different scenarios might characterize the economy one year hence. These scenarios are denoted Green, Yellow and Red and subjective probabilities 0.7, 0.1, and 0.2 are associated with them. The manager wishes to decide how a model portfolio should be allocated among stocks, bonds, real estate and gold in the face of these possible scenarios. His estimated returns in percent per year as a function of asset and scenario are given in the table below:

	Stocks	**Bonds**	**Real Estate**	**Gold**
Green	9	7	8	-2
Yellow	–1	5	10	12
Red	10	4	-1	15

Formulate and solve the asset allocation problem of minimizing the variance in return subject to having an expected return of at least 6.5.

4. Consider the ATT/GMC/USX portfolio problem discussed earlier. The desired or target rate of return in the solved model was 15%.

   a) Suppose we desire a 16% rate of return. Using just the solution report, what can you predict about the standard deviation in portfolio return of the new portfolio?

   b) We illustrated the situation where the opportunity to invest money risk-free at 5% per year becomes available. That is, this fourth option has zero variance and zero covariance. Now, suppose the risk-free rate is 4% per year rather than 5%. As before, there is no limit on how much can be invested at 4%. Based on only the solution report available for the original version of the problem (where the desired rate of return is 15% per year), discuss whether this new option is attractive when the desired return for the portfolio is 15%.

   c) You have $100,000 to invest. What modifications would need to be made to the original ATT/GMC/USX model, so the answers in the solution report would come in the appropriate units (e.g., no multiplying of the numbers in the solution by 100,000)?

d) What is the estimated standard deviation in the value of your end-of-period portfolio in (c) if invested as the solution recommends

5

Optimization Under Uncertainty, Stochastic Programming

5.1 Introduction to Decision Making Under Uncertainty

We apply the term stochastic program or scenario planning (SP) to any optimization problems (linear, nonlinear or mixed-integer) in which some of the model parameters are not known with certainty, and the uncertainty can be expressed with known probability distributions. Applications arise in a variety of industries:

- Financial portfolio planning over multiple periods for insurance and other financial companies, in face of uncertain prices, interest rates, and exchange rates
- Exploration planning for petroleum companies,
- Fuel purchasing when facing uncertain future fuel demand,
- Fleet assignment: vehicle type to route assignment in face of uncertain route demand,
- Electricity generator unit commitment in face of uncertain demand,
- Hydro management and flood control in face of uncertain rainfall,
- Optimal time to exercise for options in face of uncertain prices,
- Capacity and Production planning in face of uncertain future demands and prices,
- Foundry metal blending in face of uncertain input scrap qualities,
- Product planning in face of future technology uncertainty,
- Revenue management in the hospitality and transport industries.

5.2 Formulation and Structure of an SP Problem

In decisionmaking under uncertainty, the sequence in which information becomes available and we make decisions is important. We use the term *stage* to described the sequence pair [1)information becomes available, 2) we make a decision]. Usually, one can think of a stage as a 'time period', however there are situations where a stage may consist of several time periods. A stage: a) begins with one or more random events, e.g., some demands occur, and b) ends with our making one or more decisions, e.g., sell some excess product or order some more product.

Multistage decision making under uncertainty involves making optimal decisions for a *T*-stage horizon before uncertain events (random parameters) are revealed while trying to protect against unfavorable outcomes that could be observed in the future.

In its most general form, a multistage decision process with T+1 stages follows an alternating sequence of random events and decisions. Slightly more explicitly:

0.1) in stage 0, we make some initial decision, e.g., how much to order, taking into account that…

1.0) at the beginning of stage 1, "Nature" takes a set of random decisions, e.g., how much customers want to buy, leading to realizations of all random events in stage 1, and…

1.1) at the end of stage 1, having seen nature's decision, as well as our previous decision, we make a recourse decision, e.g., sell off excess product or order even more, taking into account that …

2.0) at the beginning of stage 2, "Nature" takes a set of random decisions, leading to realizations of all random events in stage-2, and…

2.1) at the end of stage 2, having seen nature's decision, as well as our previous decisions, we make another recourse decision taking into account that …

.
.
.

T.0) At the beginning of stage T, "Nature" takes a random decision, leading to realizations of all random events in stage T, and…

T.1) at the end of stage T, having seen all of nature's T previous decisions, as well as all our previous decisions, we make the final recourse decision.

The decision taken in stage 0 is called the *initial decision*, whereas decisions taken in succeeding stages are sometimes called *recourse decisions*. Recourse decisions are interpreted as corrective actions that are based on the actual values the random parameters realized so far, as well as the past decisions taken thus far.

All information about an SP model is stored explicitly/openly on the spreadsheet in What'sBest!. There are no hidden menus that need be accessed to see the details of the SP model. The essential steps in formulating an SP in What's*Best*! are:

1) Write a standard deterministic model (the core model) as if

the random variables are variables or parameters. You can plug in specific numbers

in a random cell to check results.

2) Identify the random variables, and decision variables,

and their staging. This is done using the:

WBSP_VAR(stage, cell_list) function for decisions variables, and

WBSP_RAND(stage, cell_list) function for random variables.

3) Provide the distributions describing the random variables. Distribution specification is stored in

WBSP_DIST_distn(table, cell_list) function, where distn specifies the distribution,

e.g., NORMAL.

4) Specify manner of sampling from the distributions, (mainly the sample size).

This information is provided via the

WBSP_STSC(table);

5) List the variables for which we want a scenario by scenario report or a histogram:

WBSP_REP(cell_list) for scenario list of values, or

WBSP_HIST(bins, cell) for histograms.

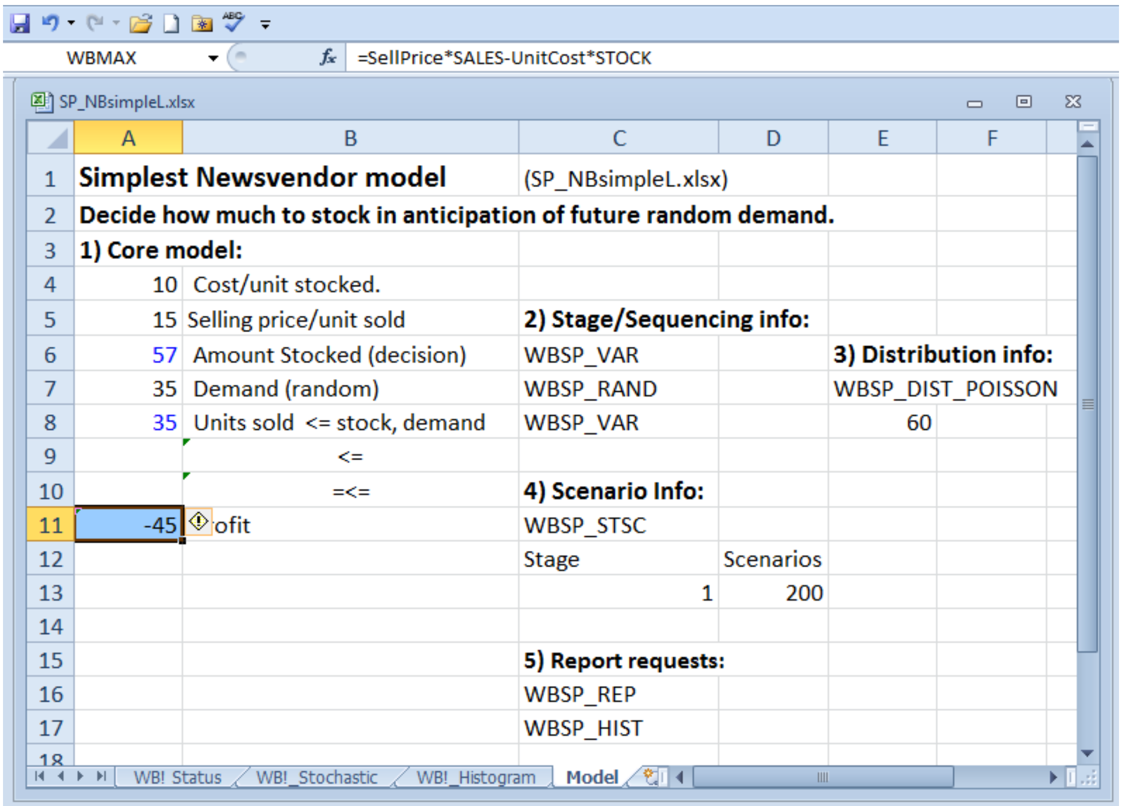
5.3 Single Stage Decisions Under Uncertainty

The simplest problems of decision making under uncertainty involve the case where there is but a single stage with randomness.

5.3.1 The News Vendor Problem

The simplest problem of decision making under uncertainty is the News Vendor problem, i.e., we must decide how much to stock in anticipation of demand, before knowing exactly what the demand will be. Figure 5.1 illustrates how to set this up in What'sBest!.

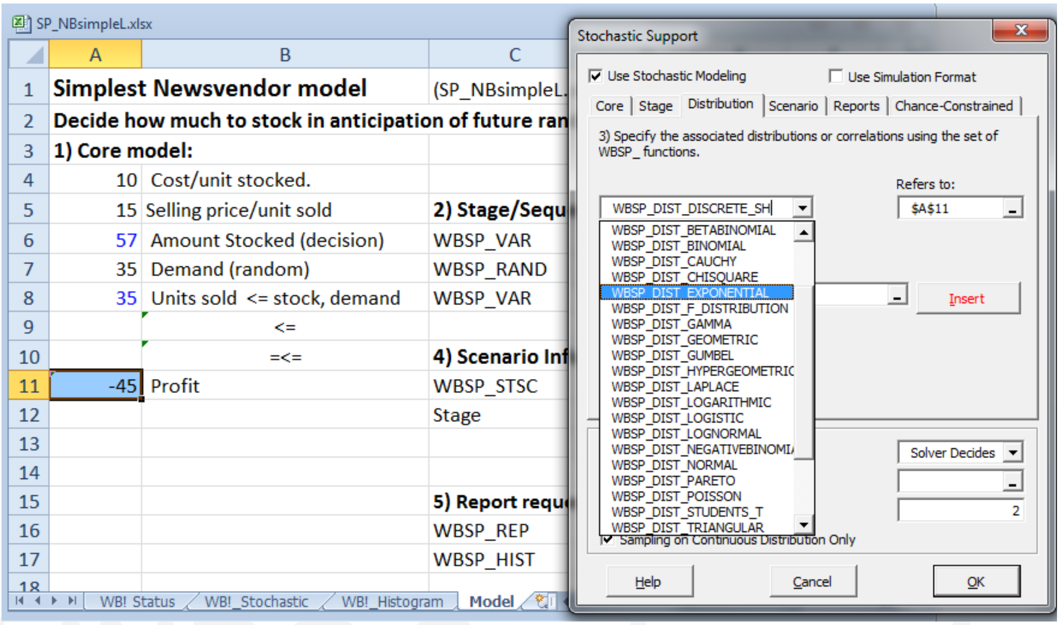
Figure 5.1 The Newsvendor Inventory Problem



If you click on What'sBest! | Options… | Stochastic

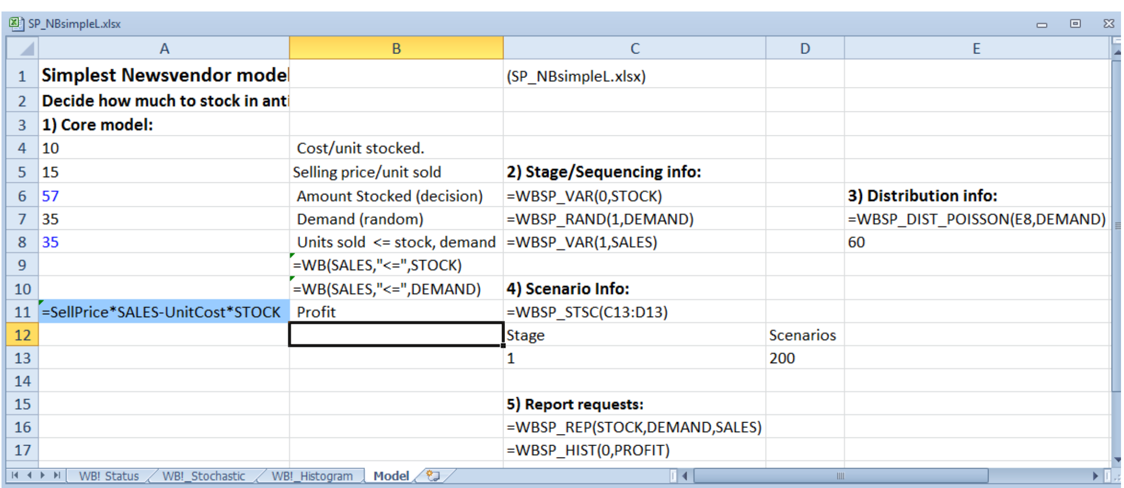
then What'sBest! will guide you through the five steps listed above for adding the stochastic features to the model via a dialog box such as that below:

Figure 5.2 Menu Steps for Setting Up an SP Model



By typing Ctrl ~ you can see in Figure 5.3 the exact nature of the formulae added to represent SP features:

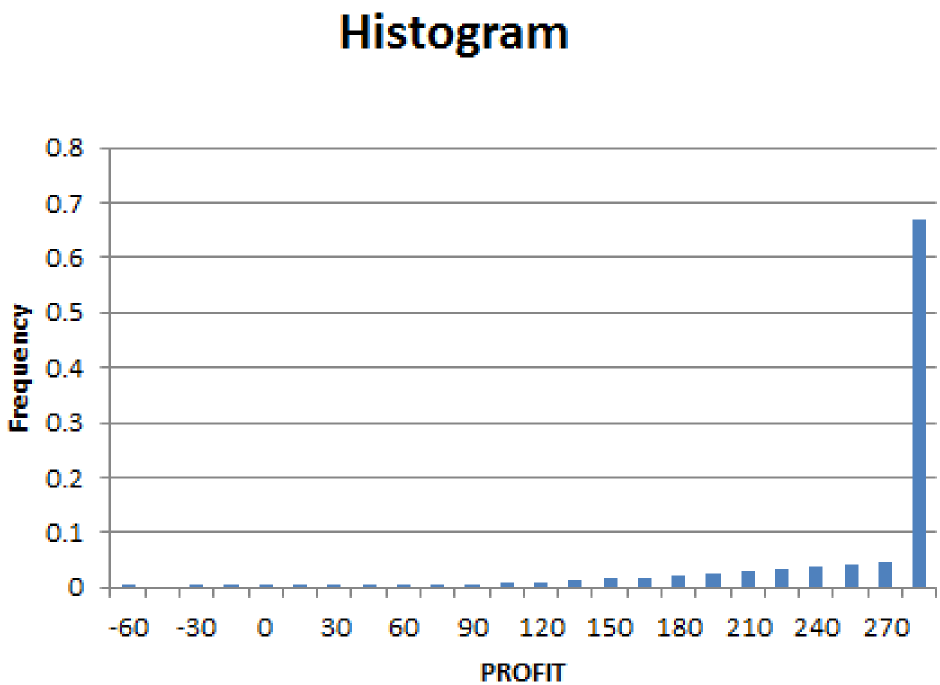
Figure 5.3 The Formulae Setting Up an SP Model



When solved, we see we should stock 57 units, slightly less than the expected demand of 60, with an expected profit of about 258. This is a little less than what you might have thought, e.g., (15 – 10)*60 = 300, which you would get if demand were always exactly 60 and we stocked 60 units. Uncertainty is is fact costing us about $42. What's*Best*! can in fact automatically compute this number for you. It appears in the WB!_Stochastic tab where it is labelled as the "Expected Value of Perfect Information" (EVPI).

An interesting exercise is to think about what the distribution of profit might look like. You might be surprised by the distribution in Figure 5.4.

Figure 5.4 Histogram of Profit for Newsvendor Inventory Problem



:

We see that close to 70% of the time, our profit is in fact $285 [=57*(15-10)], corresponding to demand being 57 or more, but we sell only what we have on hand, 57.

5.3.2 Facility Location Under Uncertainty

The next example analyzes which facilities (e.g., plants or distribution center) we should open or keep in anticipation of random demands at several different demand points. The staging of events are:

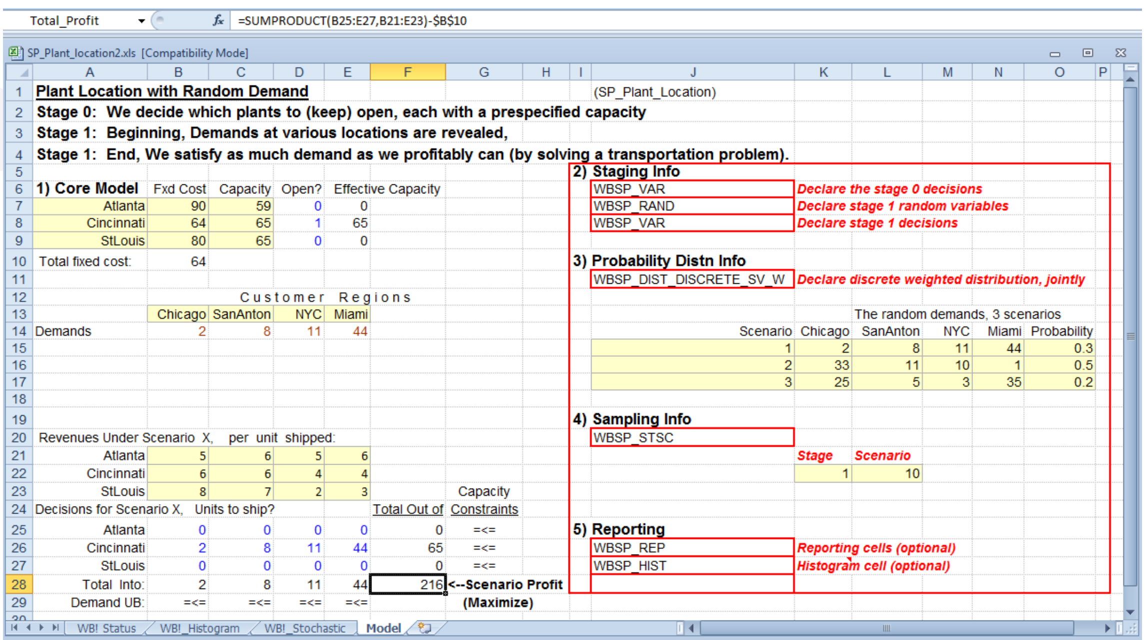
Stage 0, we decide which of three locations, Atlanta, St. Louis, or Cincinnati should have a supply facility. There is a fixed cost associated with each facility. Each facility has a prescribed capacity. In addition to the fixed cost, there is a given profit contribution per unit shipped for each combination of facility and demand point.

Stage 1, beginning: we observe the demands at Chicago, San Antonio, NYC, and Miami.

Stage 1, end: We solve a transportation problem to determine how much should be shipped from which open facilities to satisfy demand in the most profitable fashion.

Figure 5.5 shows the What'sBest! Formulation.The stage 0 decision variables are the 0/1 variables in the cells D7:D9. The stage 1 random demands occur in cells B14:E14. There are three possible demand scenarios, described in cells K15:O17. The optimal stage 0 decision is displayed in the spreadsheet, namely, open only the facility in Cincinnati.

Figure 5.5 Plant Location with Uncertain Demand



If you experiment with this example you will discover the follow perhaps surprising feature: The optimal initial decision may not be optimal for any specific scenario. More specifically: 1) If we know that scenario 1 will occur, then the best thing to do is open only the facility in Atlanta. 2) If we know that scenario 2 will occur, then the best thing to do is open only the facility in St. Louis. 3) If we know that scenario 3 will occur, then the best thing to do is open the two facilities, one in Atlanta and one in St. Louis. You can check the optimal decision for a particular scenario by setting the probability of that scenario to 1 in column O, and setting the other probabilities to 0. We have just seen, however, if we do not know for sure which demand scenario will occur, then the best thing to do is open neither Atlanta nor St. Louis, but rather, open the facility in Cincinnati. Loosely speaking, Cincinnati is the best hedge against uncertainty. Even though it is not best for any specific scenario, it is a pretty good second best for every scenario, and stochastic programming figures out that it is in fact best in terms of maximizing expected profit in the face of uncertainty.

5.4 Multi-Stage Decisions Under Uncertainty

Our examples thus far have been at most two stages. In stage 0, we make a decision, and then in stage 1 at the beginning there is one occurrence of a random event, and then finally we make one recourse decision. A slightly more complicated class of problems is the set of problems in which there are two or more separate random stages, with an intervening set of decisions. Perhaps the simplest multi-stage problems of decision making under risk are "stopping " problems, examined next.

5.4.1 Stopping Rule and Option to Exercise Problems

Some sequential decision problems are of the form: a) Each period we have to make an accept or reject decision; b) once we accept, the "game is over". We then have to live with that decision. Our next example is the simplest example of a problem known variously as a stopping problem, the college acceptance problem, the secretary problem, or the dating game. The general situation is as follows. Each period we are offered an object of known quality. We have a choice of either a) accept the object and end the game, or b) reject the object and continue in the hope that a better object will become available in a future period. The following illustrates. Each period we will see either a 2, a 7, or a 10, where 10 is the best possible, and 2 is the worst. It is clear that once we see a 10, we might as well accept. We can never do better. If we see a 2, we should never accept unless it is the last period. Whether we should accept or reject a 7 in intermediate periods is at the moment a puzzle, depending upon the probabilities of the various outcomes. There are four periods, i.e., we have 4 chances. The completely deterministic "core" model is quite simple, namely:

Maximize $v_1y_1 + v_2y_2 + v_3y_3 + v_4y_4$;

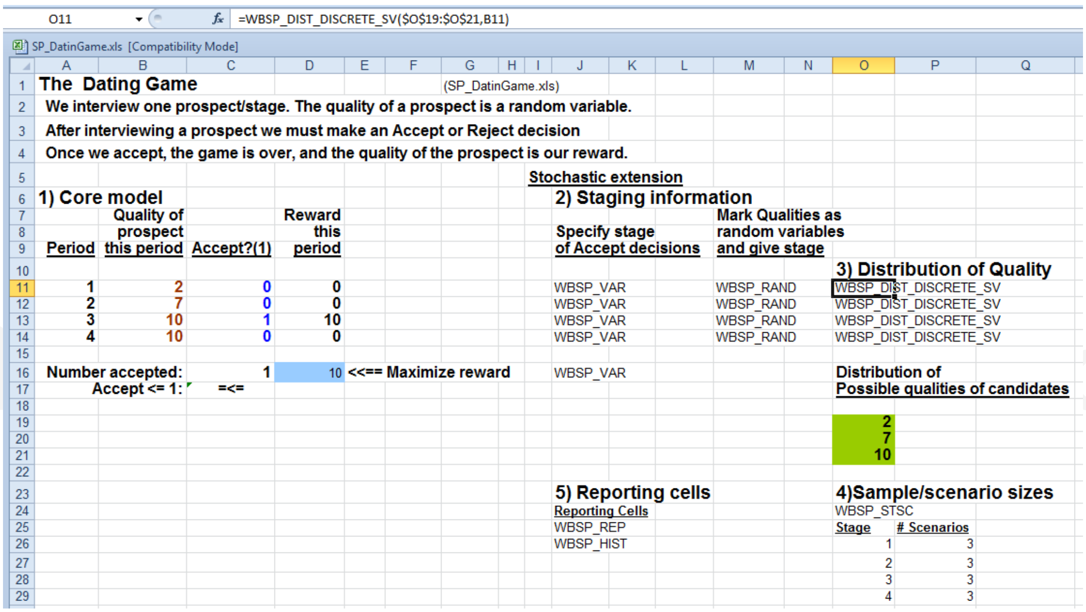
subject to:

$$y_1 + y_2 + y_3 + y_4 \leq 1;$$

$$y_j = 0 \text{ or } 1, \text{ for } j = 1, 2, 3, 4;$$

The complication is that we do not know the y_j in advance. In particular, we must choose the value for y_j immediately after seeing v_j, without knowing the future v_j 's. If we follow the simple rule of accepting the first candidate, i.e., setting = 1, then the expected value of the objective function is (2+7+10)/3 = 6.3333. To check our understanding, we might ask ourselves several questions. How much better than 6.3333 can we do by being more thoughtful? What will the optimal policy look like? We can deduce certain features of it, such as: 1) If we see a 10, then accept it immediately. We can do no better; 2) If we see a 2, reject it, except if it is the last period, then accept. The big question is what to do when we see a 7 in any period before the last. The model formulated in What'sBest! appears be Figure 5.6.

Figure 5.6 A Simple "Choose When to Stop" Problem



When solved, if we look on the WB! Status tab, we see that the expected objective value is 9.012346, quit a bit better than the 5.333333 we would get by taking the first offer. With regard to the policy, in particular, what to do when we are offered a "7", we can look at the WB!_Stochastic tab below.

Figure 5.7 Deducing Optimal Policy for Deciding When to Stop

	A	B	C	D	E	F	G	H	I	J
17			REPORTING CELLS							
18	SCENARIO	PROBABILITY								
19			Core_Model!B11	Core_Model!C11	Core_Model!B12	Core_Model!C12	Core_Model!B13	Core_Model!C13	Core_Model!B14	Core_Model!C14
20			QUALITY1	WBBINACCEPT	QUALITY2	WBBINACCEPT	QUALITY3	WBBINACCEPT	QUALITY4	WBBINACCEPT
21			STAGE 1	STAGE 1	STAGE 2	STAGE 2	STAGE 3	STAGE 3	STAGE 4	STAGE 4
47	– 23–	0.01235	10	1	2	0	2	0	7	0
48	– 24–	0.01235	10	1	2	0	2	0	2	0
49	– 70–	0.01235	7	0	10	1	10	0	10	0
50	– 71–	0.01235	7	0	10	1	10	0	7	0
51	– 72–	0.01235	7	0	10	1	10	0	2	0
52	– 64–	0.01235	7	0	10	1	7	0	10	0
53	– 65–	0.01235	7	0	10	1	7	0	7	0
54	– 66–	0.01235	7	0	10	1	7	0	2	0
55	– 67–	0.01235	7	0	10	1	2	0	10	0
56	– 68–	0.01235	7	0	10	1	2	0	7	0
57	– 69–	0.01235	7	0	10	1	2	0	2	0
58	– 61–	0.01235	7	0	7	0	10	1	10	0
59	– 62–	0.01235	7	0	7	0	10	1	7	0
60	– 63–	0.01235	7	0	7	0	10	1	2	0
61	– 55–	0.01235	7	0	7	0	7	1	10	0
62	– 56–	0.01235	7	0	7	0	7	1	7	0
63	– 57–	0.01235	7	0	7	0	7	1	2	0
64	– 58–	0.01235	7	0	7	0	2	0	10	1

WB! Status | WB!_Histogram | WB!_Stochastic | Core_Model

Notice from the highlighted row, for the given probabilities, if we see a 7 in stage 1 or 2, we do not accept (0) it, however, when we see a 7 in stage 3, whe accept (1).

5.4.2. An Option Exercise Stopping Problem

In financial markets it is frequently possible to buy options to buy or sell some financial instrument at an agreed upon "strike" price. This is a type of stopping problem. Once we have exercised the option, the game is over. The option exercise problem differs from our previous stopping problem example only in the manner in which the random variables, in this case the price of the financial instrument, is determined. In this particular example we will have five periods/stages/decision points, so the core model is similar to before:

Maximize $v_1y_1 + v_2y_2 + v_3y_3 + v_3y_3 + v_5y_5$;

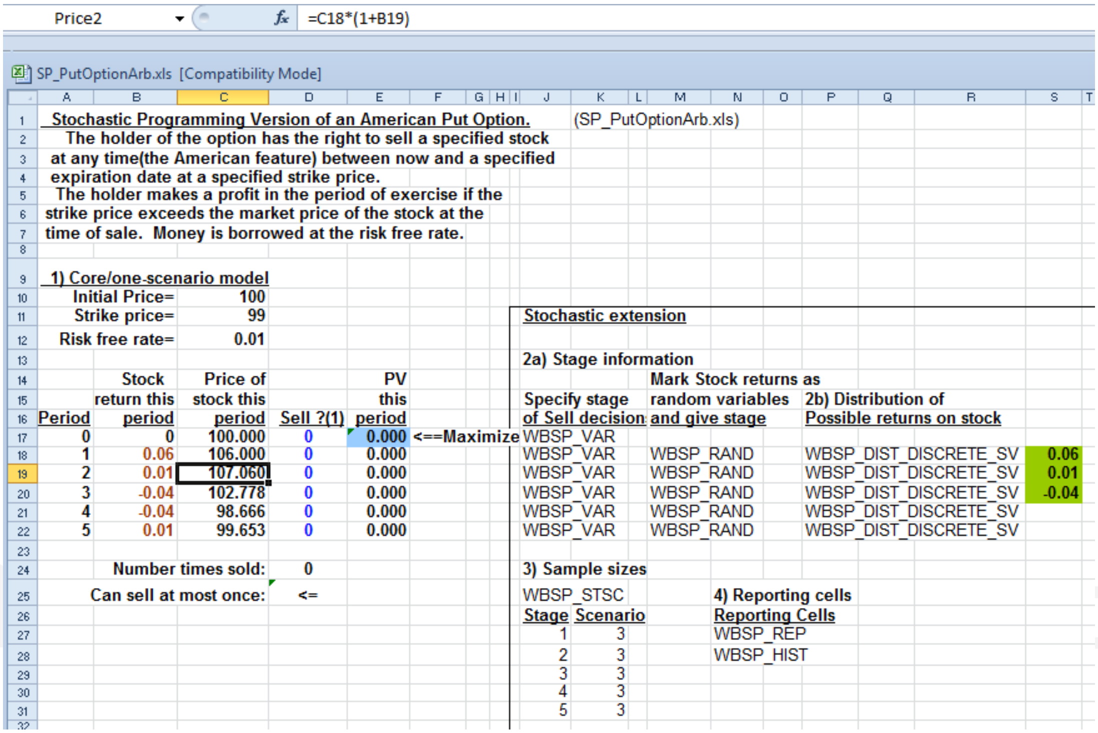
subject to:

$$y_1 + y_2 + y_3 + y_4 + y_5 \leq 1;$$

$$y_j = 0 \text{ or } 1, \text{ for } j = 1, 2, 3, 4, 5;$$

The difference is the manner in which the v_j are determined. In this particular example, we assume that with equal probability the financial instrument, say a stock, changes each period by either 1) increasing by 6%, or 2) increases, by 1%, or 3) decreases by 4%. Further, we have to pay for the option up front, however, if and when we exercise the option, we get paid (difference between the strike price minus the then current price) only later at the point of exercise. Therefore, we want to discount the future cash inflow back to the point in time that we purchase the option. Figure 5.8 shows the setup in What's*Best*!.

Figure 5.8 Deciding When to Exercise a Put Option



When solved, from the WB! Status tab, we see that the expected value of the objective is 1.669324. This means, that we would be we be willing to pay up to about 1.67 for this option. One of the attractive features of using stochastic programming is that you get to see the distribution of the profit. If we look on the WB!_Histogram tab, we see the histogram in Figure 5.9. The interesting message from this histogram is that even though the expected profit contribution from exercising the option is about 1.67, we should expect a profit contribution of zero about 70% of the time.

With regard the policy of when to sell, recall that the strike price was 99, so we would never sell if the price > 99. From looking at the WB_Stochastic tab in Figure 5.10, we see that the policy is:

Stage	Sell at Strike Price if Market Price ≤
1	never
2	92.16
3	93.08
4	94.01
5	99.

Figure 5.9 Histogram of Value of the Put Option

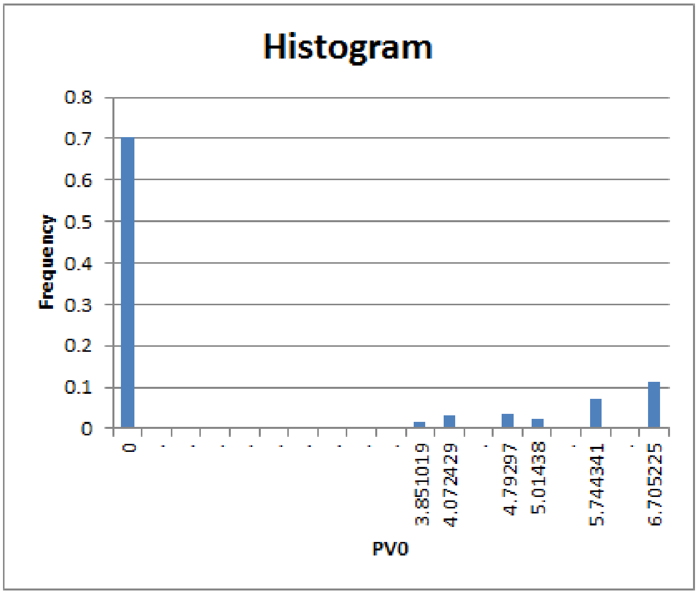


Figure 5.10 Observing When to Exercise the Option

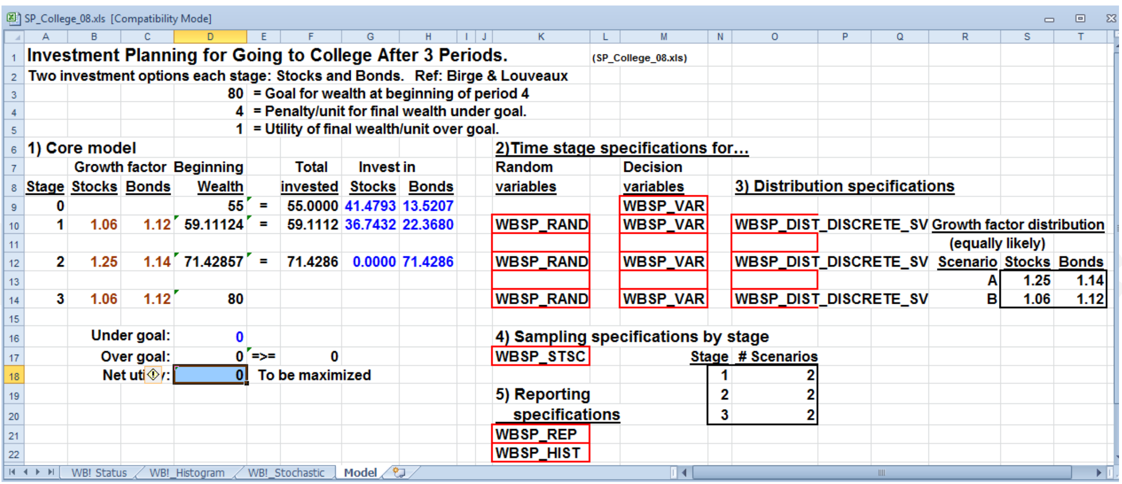
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
19	Expected Value (EV)				1.67E+00									
20	Expected Value of Perfect Information				1.05E+00									
21														
22			REPORTING CELLS											
23	SCENARIO	PROBABILITY												
24			Model!D17	Model!C18	Model!D18	Model!C19	Model!D19	Model!C20	Model!D20	Model!C21	Model!D21	Model!C22	Model!D22	Model!E17
25			SELL0	PRICE1	SELL1	PRICE2	SELL2	PRICE3	SELL3	PRICE4	SELL4	PRICE5	SELL5	PV0
26			STAGE 0	STAGE 1	STAGE 1	STAGE 2	STAGE 2	STAGE 3	STAGE 3	STAGE 4	STAGE 4	STAGE 5	STAGE 5	STAGE 5
27	------------------------------------													
48	– 146–	0.004115	0	96	0	92.16	1	97.6896	0	93.782016	0	99.408937	0	6.705225
49	– 153–	0.004115	0	96	0	92.16	1	97.6896	0	98.666496	0	94.719836	0	6.705225
50	– 151–	0.004115	0	96	0	92.16	1	97.6896	0	98.666496	0	99.653161	0	6.705225
51	– 152–	0.004115	0	96	0	92.16	1	97.6896	0	98.666496	0	104.586486	0	6.705225
52	– 150–	0.004115	0	96	0	92.16	1	97.6896	0	103.550976	0	99.408937	0	6.705225
53	– 148–	0.004115	0	96	0	92.16	1	97.6896	0	103.550976	0	104.586486	0	6.705225
54	– 149–	0.004115	0	96	0	92.16	1	97.6896	0	103.550976	0	109.764035	0	6.705225
55	– 84–	0.004115	0	96	0	96.96	0	93.0816	1	89.358336	0	85.784003	0	5.744341
56	– 82–	0.004115	0	96	0	96.96	0	93.0816	1	89.358336	0	90.251919	0	5.744341
57	– 83–	0.004115	0	96	0	96.96	0	93.0816	1	89.358336	0	94.719836	0	5.744341
58	– 90–	0.004115	0	96	0	96.96	0	93.0816	1	94.012416	0	90.251919	0	5.744341
59	– 88–	0.004115	0	96	0	96.96	0	93.0816	1	94.012416	0	94.95254	0	5.744341
60	– 89–	0.004115	0	96	0	96.96	0	93.0816	1	94.012416	0	99.653161	0	5.744341
61	– 87–	0.004115	0	96	0	96.96	0	93.0816	1	98.666496	0	94.719836	0	5.744341
62	– 85–	0.004115	0	96	0	96.96	0	93.0816	1	98.666496	0	99.653161	0	5.744341
63	– 86–	0.004115	0	96	0	96.96	0	93.0816	1	98.666496	0	104.586486	0	5.744341
64	– 102–	0.004115	0	96	0	96.96	0	97.9296	0	94.012416	1	90.251919	0	4.79297
65	– 100–	0.004115	0	96	0	96.96	0	97.9296	0	94.012416	1	94.95254	0	4.79297
66	– 101–	0.004115	0	96	0	96.96	0	97.9296	0	94.012416	1	99.653161	0	4.79297
67	– 108–	0.004115	0	96	0	96.96	0	97.9296	0	98.908896	0	94.95254	1	3.851019

WB! Status | WB!_Histogram | WB!_Stochastic | Model

5.5 Multi-stage Portfolio Choice, Meeting a Future Wealth Target

Our next example is different from the previous in at least three ways: 1) It is a multistage problem, and 2) the distribution involves two jointly distribute variables, rather than one, and 3) the distribution is an empirical discrete distribution rather than a standard distribution such as the Normal or Poisson. The decision problem is as follows. We have an initial wealth of 55,000. Initially we can invest this wealth into some combination of stocks and bonds. There will be some random return on these investments and we will have two more opportunities to reallocate our investment. At the end, we would like to have 80,000 available to provide for the college education of our child, who will be ready for college at that time. If we have more than 80,000 at the end, that will be fine. If we have less than 80,000, we will feel really bad, in fact we can quantify our disappointment by assessing a utility penalty of 4 for every unit by which we fall short of our target of 80,000. The details are specified in Figure 5.11 in What'sBest!.

Figure 5.11 A Multi-period Portfolio Allocation Problem

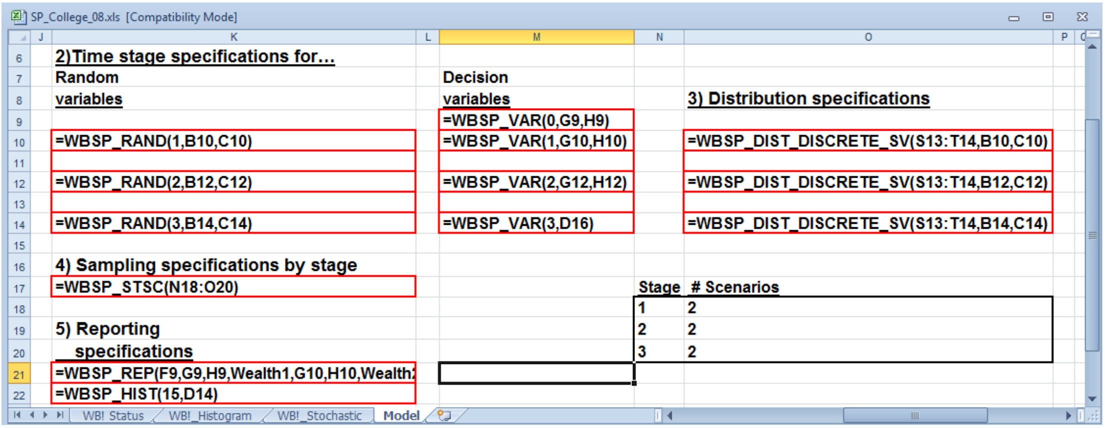

SP_College_08.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Investment Planning for Going to College After 3 Periods.											(SP_College_08.xls)								
2	Two investment options each stage: Stocks and Bonds. Ref: Birge & Louveaux																			
3				80	= Goal for wealth at beginning of period 4															
4				4	= Penalty/unit for final wealth under goal.															
5				1	= Utility of final wealth/unit over goal.															
6	1) Core model										2)Time stage specifications for…									
7		Growth factor		Beginning		Total	Invest in				Random		Decision							
8	Stage	Stocks	Bonds	Wealth		invested	Stocks	Bonds			variables		variables		3) Distribution specifications					
9	0			55	=	55.0000	41.4793	13.5207					WBSP_VAR							
10	1	1.06	1.12	59.11124	=	59.1112	36.7432	22.3680			WBSP_RAND		WBSP_VAR		WBSP_DIST_DISCRETE_SV			Growth factor distribution		
11																		(equally likely)		
12	2	1.25	1.14	71.42857	=	71.4286	0.0000	71.4286			WBSP_RAND		WBSP_VAR		WBSP_DIST_DISCRETE_SV			Scenario	Stocks	Bonds
13																		A	1.25	1.14
14	3	1.06	1.12	80							WBSP_RAND		WBSP_VAR		WBSP_DIST_DISCRETE_SV			B	1.06	1.12
15																				
16			Under goal:	0							4) Sampling specifications by stage									
17			Over goal:	0	=>=	0					WBSP_STSC		Stage		# Scenarios					
18			Net uti[illegible]y:	0	To be maximized									1	2					
19											5) Reporting			2	2					
20											specifications			3	2					
21											WBSP_REP									
22											WBSP_HIST									

WB! Status | WB!_Histogram | WB!_Stochastic | Model

Notice in particular how we specify the staging or sequencing of random variables in column K and the decision variables in column M. The details of the formulae appear in Figure 5.12.

Figure 5.12 Multiperiod Portfolio Allocation Formulae



Notice that the formulae in column K specify that the three scenarios on investment returns (columns B and C) are observed at the beginning of stages 1, 2, and 3. In contrast, column M specifies that our three decisions regarding allocation between stocks and bonds occur at the ends of periods 0, 1, and 2. Cell K21 asks for a scenario by scenario on report investent outcomes and our investment policy. This information is placed on the WB!_Stochastic tab. It is interesting to observe the possibly surprising recommended investment policy below:

Model!D12 WEALTH2 STAGE 2	Model!G12 STOCKINVEST2 STAGE 2	Model!H12 BONDINVEST2 STAGE 2	Model!D14 WEALTH3 STAGE 3	Model!D16 UNDERGOAL STAGE 3	Model!D17 OVERGOAL STAGE 3	Model!D18 NETUTILITY STAGE 3
71.428571	0	71.428571	81.428571	0	1.428571	1.428571
71.428571	0	71.428571	80	0	0	0
83.839905	83.839905	0	104.799881	0	24.799881	24.799881
83.839905	83.839905	0	88.870299	0	8.870299	8.870299
64	64	0	80	0	0	0
64	64	0	67.84	12.16	0	-48.64
71.428571	0	71.428571	81.428571	0	1.428571	1.428571
71.428571	0	71.428571	80	0	0	0

Notice the recommended policy, specifically:

if our wealth at the end of stage 2 is either quite low, i.e., 64, or quite high, i.e., 83.839905

then we should invest everything in stocks, whereas,

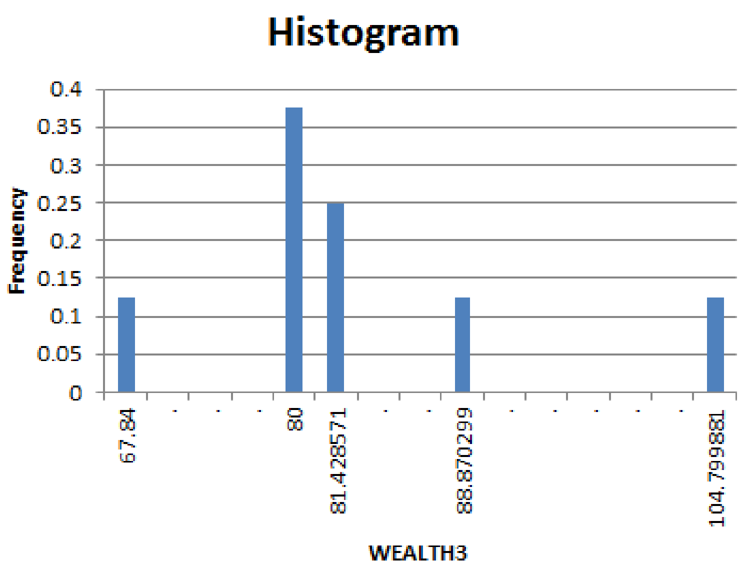
if our wealth at the end of stage 2 is intermediate, i.e. 71.428571

then we should invest everything in bonds.

Why this schizophrenic behaviour? It is because of our utility function. If you look closely at the possible returns from stocks and bonds, we see that if our wealth is 64 at the end of stage 2, then regardless of which scenario occurs, we will not reach our target, so we might as well maximize expected returns by putting everything in stocks. Similarly, if our wealth is already 83.839905, then we have already achieved our goal, so we might as well put everything in the investment with higher expected return. Now if our wealth at the end of stage 2 is 71.428571, then we can be sure of reaching our goal only if we put everything in the less variable investment, bonds.

Looking at the histogram tab in Figure 5.13, we see that 87% of the time we reach our goal of 80,000.

Figure 5.13 Histogram of Terminal Wealth



5.6 Correlated and Dependent Random Variables

In the real world, if we have several random variables, they tend to be correlated, or more generally, dependent. If we have good information about this correlation, then we would like to use this correlation information in our analysis. What'sBest! allows several ways of describing dependence among random variables. One of the easiest ways is with a correlation matrix. Figure 5.14 gives an example of a inventory/capacity planning problem in which the demands for products are correlated. We want to stock inventory for three styles of ski parkas: Anita, Daphne, and Electra. An interesting and useful complicating feature is that we can set aside a limited amount of "quick response" generic capacity that can be used to satisfy any of the three products. The example here is a variation on one we studied earlier. The correlation matrix appears in the range J15:L17 of the Figure 5.14. The random demands appear in cells B23:D23. We can connect a correlation matrix to a set of random variables by clicking on

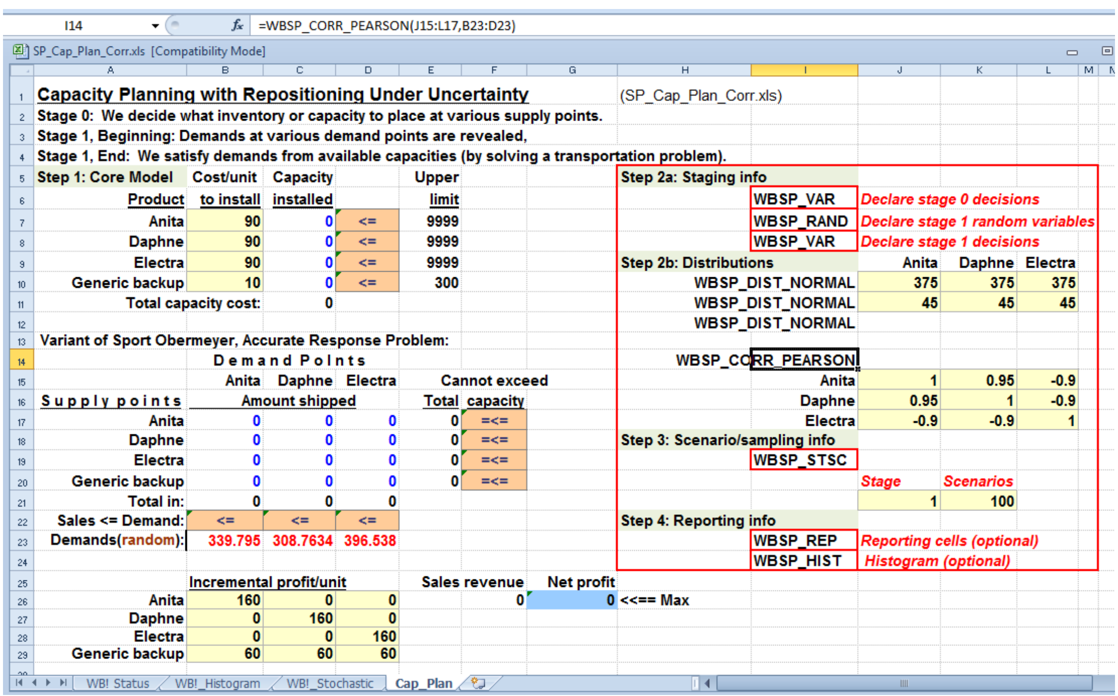
What'sBest! | Options | Stochastic Solver | Distribution | WBSP_CORR_PEARSON.

Notice that cell I14 contains the formula WBSP_CORR_PEARSON(J15:L17, B23:D23). This says that the correlation matrix is in J15:L17 and it applies to the random variables in cells B23:D23.

Before seeing the solution, it is interesting to conjecture how the correlation in demands will affect the optimal stocking levels for the three products. Some features to note:

The products are identical with respect to cost, profit contribution, and mean and standard deviation in demand. The only differences are in the correlation.

Figure 5.14 Capacity Planning with Correlated Demands


I14 =WBSP_CORR_PEARSON(J15:L17,B23:D23)

SP_Cap_Plan_Corr.xls [Compatibility Mode]

	A	B	C	D	E	F	G	H	I	J	K	L
1	Capacity Planning with Repositioning Under Uncertainty							(SP_Cap_Plan_Corr.xls)				
2	Stage 0: We decide what inventory or capacity to place at various supply points.											
3	Stage 1, Beginning: Demands at various demand points are revealed,											
4	Stage 1, End: We satisfy demands from available capacities (by solving a transportation problem).											
5	Step 1: Core Model	Cost/unit	Capacity		Upper			Step 2a: Staging info				
6	Product	to install	installed		limit				WBSP_VAR	Declare stage 0 decisions		
7	Anita	90	0	<=	9999				WBSP_RAND	Declare stage 1 random variables		
8	Daphne	90	0	<=	9999				WBSP_VAR	Declare stage 1 decisions		
9	Electra	90	0	<=	9999			Step 2b: Distributions		Anita	Daphne	Electra
10	Generic backup	10	0	<=	300				WBSP_DIST_NORMAL	375	375	375
11	Total capacity cost:		0						WBSP_DIST_NORMAL	45	45	45
12									WBSP_DIST_NORMAL			
13	Variant of Sport Obermeyer, Accurate Response Problem:											
14		Demand Points							WBSP_CORR_PEARSON			
15		Anita	Daphne	Electra	Cannot exceed				Anita	1	0.95	-0.9
16	Supply points	Amount shipped			Total	capacity			Daphne	0.95	1	-0.9
17	Anita	0	0	0	0	=<=			Electra	-0.9	-0.9	1
18	Daphne	0	0	0	0	=<=		Step 3: Scenario/sampling info				
19	Electra	0	0	0	0	=<=			WBSP_STSC			
20	Generic backup	0	0	0	0	=<=				Stage	Scenarios	
21	Total in:	0	0	0						1	100	
22	Sales <= Demand:	<=	<=	<=				Step 4: Reporting info				
23	Demands(random):	339.795	308.7634	396.538					WBSP_REP	Reporting cells (optional)		
24									WBSP_HIST	Histogram (optional)		
25		Incremental profit/unit			Sales revenue		Net profit					
26	Anita	160	0	0		0	0	<<== Max				
27	Daphne	0	160	0								
28	Electra	0	0	160								
29	Generic backup	60	60	60								

WB! Status / WB!_Histogram / WB!_Stochastic / Cap_Plan

Anita and Daphne are highly positively correlated, while Electra is highly negatively correlated with both Anita and Daphne, i.e., when Anita and Daphne have high demand, then Electra will tend to have low demand. Summarizing the sequence of events:

In stage 0, we decide how much dedicated production/inventory to build for each product and how much quick response generic backup production to contract.

In stage 1, we see the demands, and then satisfy as much of each product demand from its dedicated inventory. Any remaining unsatisfied demand is satisfied as much as possible from generic backup available.

For each product, it is more profitable, 160-90 = 70, to serve the product from dedicated production rather than from generic production, 60-10 = 50. So if demand were deterministic, Generic backup would not be used. Some interesting questions are:

a) How much of the less profitable but more flexible generic backup capacity should we commit to in order to compensate for uncertainty?
b) How will the correlation in demands affect our stocking policies? Will the positive correlation in demands for Anita and Daphne cause us to stock more?

When we optimize the model, we get the solution:

Production option	Amount to stock or commit
Anita	338
Daphne	338
Electra	327
Generic backup	169

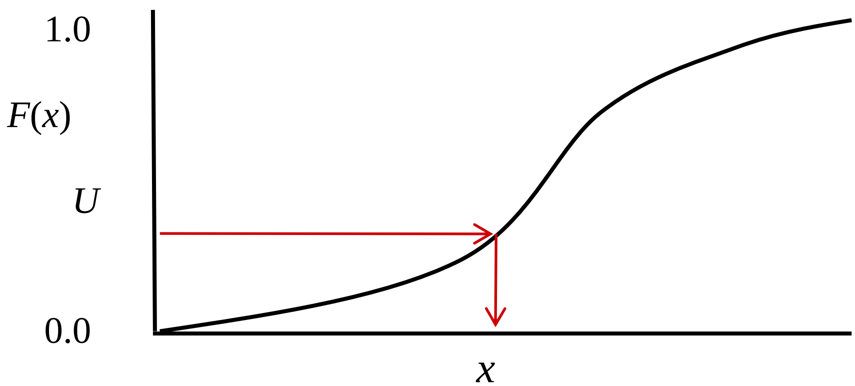
The total expected profit is about 74,062. Intuitively, we do not stock as much of Electra because when Electra demand is high, demand for Anita and Daphne will be low, so the Generic backup capability that we purchased can be used to satisfy Electra demand. We have to protect ourselves with more initial inventory for Anita and Daphne because when demand of one of them is high, the demand for the other will also be high. An interesting exercise is to look at the value of the generic backup, e.g., set the generic backup upper limit to closer to zero, from its given value of 300.

What'sBest! provides three different types of correlation, Pearson, Spearman rank, and Kendall rank. The most commonly used type of correlation, e.g., as taught in an introductory statistics course, is Pearson correlation. It is appropriate if the random variables of interest have a Normal distribution. For non-Normal random variables, the Spearman or Kendall rank correlation measures tend to be more appropriate.

5.7 Accuracy of Results and Random Number Generation

In many of the applications of SP, we <u>sample</u> from a distribution. When sampling is involved, a natural question is then how much confidence should we have in the results? We discuss two things in this regard: 1) what can be done to reduce the variability of the results, and 2) How can we compute confidence intervals on the results? We will first give some details about how random numbers are generated in What'sBest! when we are sampling from a univariate distribution. What'sBest! uses the inverse transform method for generating random variables from a specific distribution such as the normal. This method is best explained graphically. For every distribution, What'sBest! "knows" the formula for the cumulative distribution function (cdf), or at least has a highly accurate approximation, e.g., 8 to 14 decimal digits. Graphically the distribution looks qualitatively as in Figure 5.15.

Figure 5.15 CDF of a Distribution and Inverse Transform Method



For any value of x, the cdf, F(x), gives the probability that the random variable is less than or equal to x for the associated distribution. Remember that probability is a number between 0 and 1. The inverse transform method "inverts" the function F(x), call it $F^{-1}(u)$, so as to "run it backwards". In the graph, we input a number between 0 and 1 on the vertical axis and get a corresponding horizontal axis. We do not prove it here, but it is somewhat intuitive that if the number U is chosen randomly from a uniform distribution, then the corresponding $x = F^{-1}(u)$, follows the distribution described by F( ).

5.7.1 Latin Hypercube Sampling for Variance Reduction

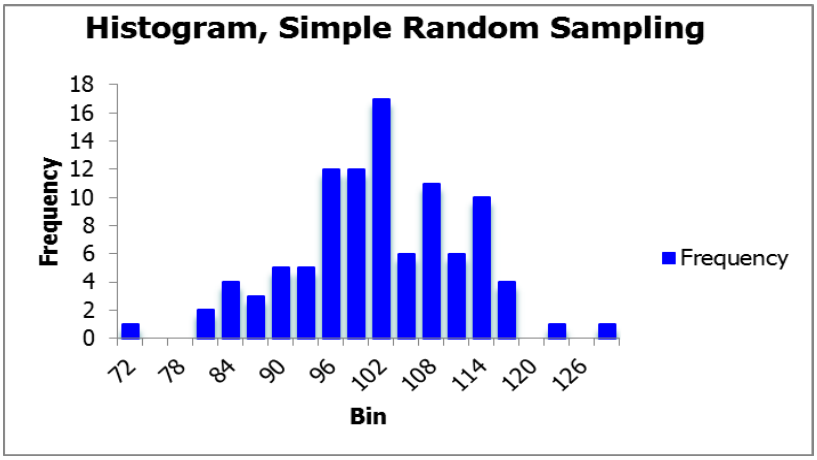
By default, What'sBest! uses a form of sampling called Latin Hypercube Sampling (LHS). The basic idea is that if we need *N* samples from a distribution, we partition the distribution into *N* equiprobable intervals and choose one sample randomly from each interval. This is easy to do if Inverse Transform Method is used. To illustrate, we asked What's*Best*! to generate 10 random demands from a uniform distribution over the continuous interval (0, 10). The ten numbers generated were:

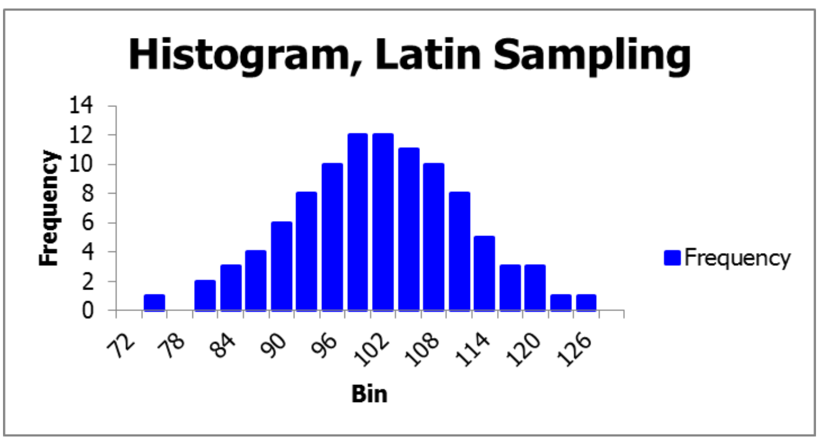
```
 0.02773
7.89123
5.54321
3.76877
4.30992
2.79945
9.41034
8.37275
1.33699
6.72890
```

Notice there is one lead digit = 0, one lead digit = 1, . . ., and one lead digit = 9. A key features are that every interval is represented, but also every number in the interval (0, 10) has the same probability of being chosen/generated. This is sufficient to argue that using LHS does not introduce any bias.

The effect of using LHS can be illustrated graphically. We generated 100 draws from a normal distribution with mean 100 and standard deviation 10. When we used simple random sampling we got the top histogram in Figure 5.16.

Figure 5.16 Random Sampling vs. Latin Hypercube Sampling





When we used LHS, in the bottom of Figure 5.16, we get a histogram that looks closer to a normal distribution.

There is an optimistic "optimization" bias of the order of $(n\text{-}1)/n$ in the objective function value from simple SP. The bias arises when one uses sampling because the optimization will choose the policy that does best for the chosen finite sample. If one chose the policy that is optimal for an infinite sample, that globally optimal policy cannot do better for the finite sample than the policy that is optimal for the finite sample. Using LHS tends to reduce this bias, as well as the variance of the estimate. We give two examples taken from Yang(2004).

Problem	Simple random sampling Mean	S. Error	LHS Mean	S. Error
1) One product Newsvendor. Scenarios = 1000, Replications = 100. Minimize cost,	5546.7	28.83	5547.2	9.86
2) Multi-product inventory and with substitution. Scenarios = 256, Replications = 100. Maximize profit,	189902	3162	189173	1275

Notice that in both cases, using LHS resulted in substantially lower standard error, that is, the variance among the 100 replications was substantially lower. Also, the bias is apparently less with LHS. In case 1, where we are minimizing cost, simple random sampling has a lower average cost over the 100 replications, probably because of this bias. In case 2, where we are maximizing profit, simple random sampling has a higher average profit over the 100 replications, probably because of this bias.

5.7.1 Computing Approximate Confidence Intervals for SP

How confident should we be statistically, of the results of an SP optimization? We will give a simple method for computing an approximate confidence interval on the expected profit from an SP analysis. We will argue that we expect the approximation to be of higher quality if LHS is use. There are (at least) two effects to think about:e 1) There is an optimistic bias of the order of $(n\text{-}1)/n$ in the objective function value from an SP optimization, because the optimization chooses the policy best for the sample chosen, and 2) If we use Latin Hypercube sampling, then the samples or scenarios tend to be negatively correlated, rather than independent. This is because a random outcome far below the median will be compensated by an observation far above. Thus, an estimate of standard deviation among the samples based on the assumption of independence is wrong. We illustrate these effects with a simple analysis of a Newsvendor inventory problem with the following parameters:

1000 = Mean demand for the one period;
300 = Standard deviation in demand;
140 = Revenue/unit sold;
60 = Cost/unit purchased;
0 = Penalty/unit unsatisfied demand;
40 = Holding cost/unit left in inventory;
15 = Number of scenarios sampled in the SP optimization.

The 15 was chosen for illustrative purposes only, not as a recommended sample size;

We repeated or replicated the above 15-sample SP 1000 times. For each replication we computed

a) the observed average profit, *xbar*;

b) the traditional "unbiased" estimate of the population standard deviation by

$[\Sigma_i (x_i - xbar)^2/(n\text{-}1)]^{0.5}$, and,

c) a 90% coverage interval for *xbar*, estimating the standard deviation of *xbar* by

$s = [\Sigma_i (x_i - xbar)^2/(n(n\text{-}1))]^{0.5}$.

For the simple Newsvendor inventory problem, the true expected profit can be computed analytically. For each replication we recorded whether the computed confidence interval in fact covered the true expected profit of $71,601. Results for the 1000 replications are shown below.

Sampling method	Mean profit	Mean sample standard deviation	Actual 90% confidence interval coverage
Random	$72,127	$25,945	.898
LHS	$71,595	$26,761	1.000
True/Analytical	$71,601		

Some things to note:

1) Because of the modest number of scenarios, n = 15, SP with simple random sampling seriously overestimates the expected profit by $526. SP with LHS actually, by chance, slightly underestimates, by $6, the true expected profit.

2) The sample standard deviation under LHS is substantially less of an underestimate of the (unknown) population standard deviation in profit than is that under simple random sampling.

3) The confidence intervals computed under simple random sampling do not quite achieve the desired 90% coverage, perhaps because the intervals are not correctly centered because of the optimistic bias in *xbar*.

4) The confidence intervals from SP with LHS are extremely conservative, and in fact achieve 100% coverage.

5.8 The Cost of Uncertainty

In making the decision to use SP for a particular application, you may want to ask how much uncertainty is costing you with regard to this application. We will describe two simple values for measuring the cost of uncertainty:

1) *EVPI* (Expected Value of Perfect Information)

= Expected increase in profit if we had a perfect forecast, and

2) *EVMU* (Expected Value of Modeling Uncertainty)

= Expected decrease in profit if we act as if each random variable is a constant, typically its mean, instead of taking into account the randomness.

What's*Best*! can provide the above two numbers for any SP problem (Click on:
What's*Best*! | Options… | Stochastic Solver… | Reports | Expected Value of Perfect Information).
In order to make clear what is meant by perfect forecasts, let us stress the distinction between variability and uncertainty. Suppose the appropriate random variable to think about is rain tomorrow. The amount of rain tomorrow is variable. If we do not have perfect forecasts, then the amount of rain tomorrow is a random variable. A perfect forecast does not remove the variability. It removes only the uncertainty about the variability. In the stochastic programming literature, EVMU is sometimes also known as the Value of Stochastic Solution (VSS).

We will illustrate with a capacity planning/multi-product inventory problem, customized to a simplified model of an apparel retailer.

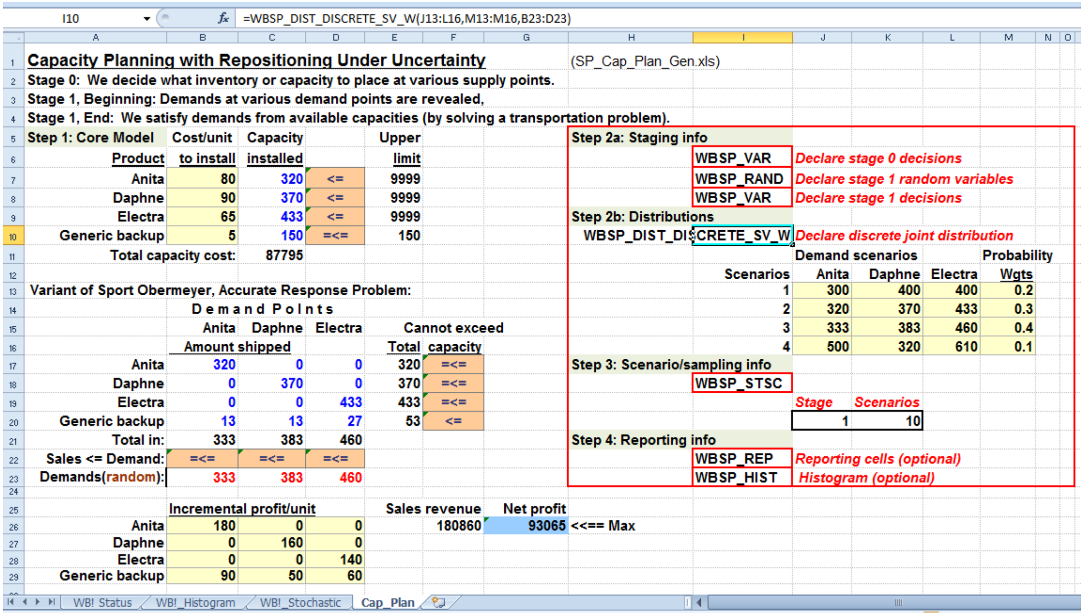
Example: Multi-product Inventory with Repositioning

This is example is a very simplified illustration of an inventory management approach used by some apparel retailers. The general sequence of events is:

Stage 0) Before the selling season starts
the retailer commits inventory to a number of locations and/or products.
Stage 1, beginning) Demands at the various locations or products is observed.
Stage 1, end) Product can be repositioned to some extent, at some additional cost, among the various locations/products, generally moving inventory to the locations/products with higher than expected demand.

This is a very crude simplified representation of an inventory allocation system with reallocation used at the clothing retailers Sport Obermeyer, see Fisher and Raman(1996) and at the Spanish firm Zara, see Caro and Gallien (2010). Our example below is closer to that of Sport Obermeyer, where the secondary reallocation is over products, whereas at Zara, the reallocation is over locations. In the example below, in stage 0 we need to decide what initial quantities should be produced to inventory of three types of parkas, the "Anita", "Daphne", and "Electra". After this initial production run, we observe the demands for the three parkas. In our example, there are four possible scenarios with associated probabilities. Once we see the demands, we have access to a fast backup production facility of limited capacity that can produce any of the three products. Although this backup facility is fast, it is also very expensive per unit produced, and it has limited capacity, so if we had perfectly accurate forecasts, we would not use the backup facility. We would produce just the right amount of each product from the outset. In the real world where perfect forecasts are the exception, the main question is: How much should we produce of each product initially, taking into account that we can use the somewhat expensive backup facility to partially compensate for our forecast errors.

Figure 5.17 An Inventory/Production Repositioning Problem



I10 =WBSP_DIST_DISCRETE_SV_W(J13:L16,M13:M16,B23:D23)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Capacity Planning with Repositioning Under Uncertainty							(SP_Cap_Plan_Gen.xls)					
2	Stage 0: We decide what inventory or capacity to place at various supply points.												
3	Stage 1, Beginning: Demands at various demand points are revealed,												
4	Stage 1, End: We satisfy demands from available capacities (by solving a transportation problem).												
5	Step 1: Core Model	Cost/unit	Capacity		Upper			Step 2a: Staging info					
6	Product	to install	installed		limit				WBSP_VAR	Declare stage 0 decisions			
7	Anita	80	320	<=	9999				WBSP_RAND	Declare stage 1 random variables			
8	Daphne	90	370	<=	9999				WBSP_VAR	Declare stage 1 decisions			
9	Electra	65	433	<=	9999			Step 2b: Distributions					
10	Generic backup	5	150	=<=	150			WBSP_DIST_DISCRETE_SV_W		Declare discrete joint distribution			
11	Total capacity cost:		87795							Demand scenarios			Probability
12									Scenarios	Anita	Daphne	Electra	Wgts
13	Variant of Sport Obermeyer, Accurate Response Problem:								1	300	400	400	0.2
14		D e m a n d P o i n t s							2	320	370	433	0.3
15		Anita	Daphne	Electra	Cannot exceed				3	333	383	460	0.4
16		Amount shipped			Total	capacity			4	500	320	610	0.1
17	Anita	320	0	0	320	=<=		Step 3: Scenario/sampling info					
18	Daphne	0	370	0	370	=<=			WBSP_STSC				
19	Electra	0	0	433	433	=<=				Stage	Scenarios		
20	Generic backup	13	13	27	53	<=				1	10		
21	Total in:	333	383	460				Step 4: Reporting info					
22	Sales <= Demand:	=<=	=<=	=<=					WBSP_REP	Reporting cells (optional)			
23	Demands(random):	333	383	460					WBSP_HIST	Histogram (optional)			
24													
25		Incremental profit/unit			Sales revenue		Net profit						
26	Anita	180	0	0		180860	93065	<<== Max					
27	Daphne	0	160	0									
28	Electra	0	0	140									
29	Generic backup	90	50	60									

WB! Status / WB!_Histogram / WB!_Stochastic / Cap_Plan

The key features of the spreadsheet are:

Cell I6 declares that the capacity/initial inventory decisions in C7:C10 must be made first.

Cell I7 tells What'sBest! that next the demands for the three products are observed in B23:B25,

Cell I8 declares that finally the allocation of inventory/capacity to product is done in B17:D20.

Cell I10 tells What'sBest! that the possible scenarios are described in J13:M16.

Notice in the profit contribution table: B26:D29, that Anita capacity/inventory can be used profitably only to satisfy Anita demand. Similarly for Daphne and Electra. Generic backup capacity can be used to satisy any demand, though not nearly as profitably as specifically committed inventory.

The spreadsheet shows the optimal stage 0 decision in cells C7:C10, namely, our initial production quantities should be Anita = 320, Daphne = 370, and Electra = 433. These happen to be the demands found in scenario 2. Looking at the WB!_Stochastic tab, we see expected values below, in particular, the expected profit is 90207.00.

```
Expected Value (EV)                                          90207.00
Expected Value of Wait-and-See (EVWS)                        94371.50
Expected Value using Expected Value Policy (EVEVP)           87823.30
Expected Value of Perfect Information (=|EVWS-EV|)            4164.50
Expected Value of Modeling Uncertainty (=|EV-EVEVP|)          2383.70
```

Now let us explain the other expected values. The Wait-and-See value is based on the situation where we know in advance which scenario will occur. This would occur if we had perfect forecasts. If we knew demand in advance, we would simply set initial inventory of each product to demand, and not use any generic backup. The computations to arrive at the amount, 94371.00 are shown below.

Scenario	Profit if demand known in advance	Probability
1	88,000	0.2
2	90,375	0.3
3	94,610	0.4
4	118,150	0.1

Expected profit under Wait-and-See 94,371.50 (=88000*0.2 + … +118150*0.1).

Thus the Expected Value of Perfect Information (EVPI) = 94,371.50 – 90207 = 4164.50. The EVPI gives an upper bound on the value of better forecasts.

Now let us look at how EVMU is computed. Historically, many firms used single number forecasts, without giving any official comment about the uncertainty in this single number. The most obvious single number to use is expected demand. That is the default method that What's*Best*! uses in computing EVMU. So EVMU measures how much worse you would do by acting as if all random variables are in fact single numbers equal to the expectation of the random variables.

The computation of EVMU are tabulated below. For example the expected demand for Anita parkas is 300*0.2+320*0.3+333*0.4+500*0.1 = 339.2.

Product	Cost/unit to install	Expected demand	Capacity installed
Anita	80	339.2	339.2
Daphne	90	376.2	376.2
Electra	65	454.9	454.9
Generic backup	5	0	0

Contemplated profit under mean demand = 94,371.50

Actual expected profit of mean based policy = 87,823.30

If you act as if all random variables will always take on their mean value, then for this problem we would think we would make a profit of 94,371.50. That number is misleading, however, because under the mean based policy when the demands are uncertain, the expected profit is in fact only 87,823.30. This can be seen by setting the initial inventories/capacities in cells C3:C6 to 339.2, 376.2, 454.9, and 0, fixing them at those values and then solving the restricted SP. Thus, we see that the expected value of using SP, rather than acting as if the demands were constants equal to their means, is 90207 – 87823.30 = 2383.70. It may be useful to think of the EVMU and EVPI in terms of the little line graph below.

```
                    Expected Profit  →
            EVMU                         EVPI
  |---------------------|--------------------------------------|
87823.30             90207.00                                 94371.50
Disregard             Use SP                                  Perfect
uncertainty                                                  forecasts
```

Details and Complications with EVMU and EVPI

The default in What'sBest! is to use the mean when computing EVMU. There are some unusual distributions, such as the Cauchy, for which the mean does not exist. Also, for discrete random variables, such as the binomial, the mean may be a fraction, even though the random variable always takes on an integer value, so there might be some complications in the model as a result.

Another fine point is that for continuous distributions such as the normal, What's*Best*! uses sampling. EVPI and EVMU are computed based on this sample, rather than on the true population, which would correspond to an infinite number of scenarios. In these cases, the values for EVPI and EVMU reported are estimates rather than true values.

References

Arnold, L., and D. Botkin. "Portfolios to Satisfy Damage Judgement: A Linear Programming Approach", *Interfaces*, vol. 8, No. 2 (Feb. 1978).

Birge, J. R. (1997), "Stochastic Programming Computation and Applications", *INFORMS Journal on Computing*, vol. 9, No. 2, pp.111-133.

Birge, J. and F. Louveaux (1997), *Introduction to Stochastic Programming*, Springer-Verlag, New York, NY.

Bracken, J. and G.P. McCormick (1968), *Selected Applications of Nonlinear Programming*, John Wiley & Sons, Inc., New York, NY.

Carino, D.R., T. Kent, D.H. Myers, C. Stacy, M. Sylvanus, A.L. Turner, K. Watanabe, and W.T. Ziemba (1994), "The Russell-Yasuda Kasai Model: An Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming", *Interfaces*, Vol. 24, No. 1, pp. 29-49.

Caro, F. and J. Gallien (2010), "Inventory Management of a Fast-Fashion Retail Network", *Operations Research*, vol. 58, no. 2, pp. 257-273.

Clarke, G. and J. W. Wright (1964), "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol. 12, No. 4 (July-Aug.), pp. 568-581.

Claus, A.(1984), "A New Formulation for the Travelling Salesman Problem", *SIAM Journal on Algebraic and Discrete Methods*, vol. 5, no. 1, pp. 21-25.

Clyman, D.R. (1995), "Unreasonable Rationality?", *Management Science*, vol. 41, No. 9 (Sept.), pp. 1538-1548.

Craven, J. P.(2001), *The Silent War: the Cold War Battle Beneath the Sea*, Simon & Schuster, New York.

Dantzig, G. (1963), *Linear Programming and Extensions*, Princeton University Press, Princeton.

Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson(1954), "Solution of a Large-Scale Traveling-Salesman Problem", *Operations Research*, vol. 2, no. 4, pp. 393-410.

Dantzig, G. and N. N. Thapa (1997), *Linear Programming*, vol. 1, Springer, New York.

Danusaputro, S., C. Lee, and L. Martin-Vega (1990), "An Efficient Algorithm for Drilling Printed Circuit Boards", *Computers and Industrial Engineering*, vol. 18, pp. 145-151.

DeRosa, D. (1992), *Options on Foreign Exchange*, Irwin Professional Publishing, New York.

Dikin, I. I. (1967), "Iterative Solution of Problems of Linear and Quadratic Programming", *Soviet Mathematics Doklady*, vol. 8, pp. 674-675.

Eppen, G., K. Martin, and L. Schrage (1988), “A Scenario Approach to Capacity Planning”, *Operations Research*, vol. 37, No. 4 (July-August), pp. 517-530.

Fisher, M. and A. Raman (1996), “Reducing the Cost of Demand Uncertainty Through Accurate Response to Early Sales”, *Operations Research,* Vol. 44, No. 1 (Jan.-Feb.), pp. 87-99.

Infanger, G. (1994), *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs,* Boyd & Fraser, Danvers, MA.

Jorion, P. (2001), *Value at Risk*, 2nd ed., McGraw-Hill.

Kall, P. and S.W. Wallace (1994), *Stochastic Programming*, John Wiley & Sons, New York, NY.

Kara, I., G. Laporte and T. Bektas (2004), “A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem”, *European Journal of Operational Research*, vol. 158, pp. 793-795.

Konno, H. and H. Yamazaki (1991), “Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market”, *Management Science*, vol. 37, No. 5 (May), pp. 519-531.

Levy, F. K. (1978), “Portfolios to Satisfy Damage Judgements: A Simple Approach”, *Interfaces,* vol. 9, No. 1 (Nov.), pp. 106-107.

Lin, S. and B. Kernighan (1973), “An Effective Heuristic Algorithm for the Traveling Salesman Problem”, *Operations Research*, vol. 21, pp. 498-516.

Madansky, A. (1962), “Methods of Solution of Linear Programs Under Uncertainty”, *Operations Research*, vol. 10, pp. 463-471.

Markowitz, H. M. (1959), *Portfolio Selection, Efficient Diversification of Investments*, John Wiley & Sons, Inc..

Markowitz, H. and A. Perold (1981), “Portfolio Analysis with Scenarios and Factors”, *Journal of Finance,* vol. 36, pp. 871-877.

Martin, R. K.(1999) *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Boston.

Miller, C., A. Tucker and R. Zemlin (1960), “Integer programming formulation of travelling salesman problems”, *J. of ACM*, vol. 7, pp. 326-329.

Nauss, R. M. (1986), “True Interest Cost in Municipal Bond Bidding: An Integer Programming Approach”, *Management Science*, vol. 32, No. 7, pp. 870-877.

Nauss, R. M. and B. R. Keeler (1981), “Minimizing Net Interest Cost in Municipal Bond Bidding”, *Management Science*, vol. 27, No. 4 (April), pp. 365-376.

Nauss, R. M. And R. Markland (1981), “Theory and Application of an Optimization Procedure for Lock Box Location Analysis”, *Management Science*, vol. 27, No. 8 (August), pp. 855-865.

Nemhauser, G. L. and L. A. Wolsey (1988), *Integer and Combinatorial Optimization,* John Wiley & Sons, Inc.

Orman, A.J. and H. P. Williams (2007), "A Survey of Different Integer Programming Formulations of the Travelling Salesman problem", *Optimisation, Econometric and Financial Analysis, Advances in Computational Management Science*, vol. 9, part 1, pp. 91-104.

Padberg, M. and G. Rinaldi (1987), "Optimization of a 532-City Symmetric Traveling Salesman Problem by Branch and Cut", *Operations Research Letters*, vol. 6, No. 1.

Perold, A. F. (1984), "Large Scale Portfolio Optimization", *Management Science*, vol. 30, pp. 1143-1160.

Rardin, R. L. (1998), *Optimization in Operations Research*, Prentice Hall, New Jersey.

Roy, A. D. (1952), "Safety First and the Holding of Assets", *Econometrica*, vol. 20 (July), pp. 431-439.

Sharpe, W. F. (1963), "A Simplified Model for Portfolio Analysis", *Management Science*, vol. 9 (Jan.), pp. 277-293.

Weingartner, H. M. (1972), "Municipal Bond Coupon Schedules with Limitations on the Number of Coupons", *Management Science*, vol. 19, No. 4 (Dec.), pp. 369-378.

What's Best User Manual. LINDO Systems, Chicago, (1998).

Wolsey, L.(1998), "Integer Programming", *Wiley Interscience*, New York.

Yang, H. and L. Schrage (2009). Conditions that Cause Risk Pooling to Increase Inventory. *European Journal of Operational Research*, vol. 192, no. 3, pp. 837-851.

Index