

Prepared By

Handaru Jati, Ph.D

Universitas Negeri Yogyakarta

handaru@uny.ac.id

Chapter 8



Using The Excel Solver To Solve Mathematical Programs



Chapter Overview

- 8.1** Introduction
- 8.2** Formulating Mathematical Programs
 - 8.2.1** Parts of the Mathematical Program
 - 8.2.2** Linear, Integer, and Nonlinear Programming
- 8.3** The Excel Solver
 - 8.3.1** The Solver Steps
 - 8.3.1.1** Standard Solver
 - 8.3.1.2** Premium Solver
 - 8.3.2** A Solver Example
 - 8.3.2.1** Product Mix
 - 8.3.2.2** Infeasibility
 - 8.3.2.3** Unboundedness
 - 8.3.3** Understanding Solver Reports
- 8.4** Applications of the Solver
 - 8.4.1** Transportation Problem
 - 8.4.2** Workforce Scheduling
 - 8.4.3** Capital Budgeting
 - 8.4.4** Warehouse Location
- 8.5** Limitations and Manipulations of the Solver
- 8.6** Summary
- 8.7** Exercises

8.1 Introduction

This chapter illustrates how to use the Excel Solver as a tool to solve mathematical programs. We review the basic parts of formulating a mathematical program and present several examples of how the Solver interprets these parts of the program from the spreadsheet. We give examples of linear, integer, and non-linear programming problems to show how the Solver can be used to solve a variety of mathematical programs. We also give an overview of the Premium Solver and its benefits. This chapter is important for the reader to understand as many DSS applications involve solving optimization problems, which are mathematical programs. The reader should be comfortable with preparing the spreadsheet for use with the Solver. In Chapter 19, we revisit the Solver using VBA commands. We have several examples of DSS applications which use the Solver to solve optimization problems, such as Portfolio Management and Optimization.

8.2 Formulating Mathematical Programs

The Excel spreadsheet is unique because it is capable of working with complex mathematical models. Mathematical models transform a word problem into a set of equations that clearly define the values that we are seeking, given the limitations of the problem. Mathematical models are employed in many fields, including all disciplines of engineering. In order to solve a mathematical model, we develop a mathematical program which can numerically be solved and retranslated into a qualitative solution to the mathematical model.

8.2.1 Parts of the Mathematical Program

A mathematical program consists of three main parts. The first is the **decision variables**. *Decision variables* are assigned to a quantity or response that we must determine in a problem. For example, if a toy manufacturer wants to determine how many toy boats and toy cars to produce, we assign a variable to represent the quantity of toy boats produced, x_1 , and the quantity of toy cars produced, x_2 . Decision variables are defined as *negative*, *non-negative*, or *unrestricted*. An *unrestricted* variable can be either *negative* or *non-negative*. These variables represent all other relationships in a mathematical program, including the objective, the limitations, and the requirements.

The second part of the math program, called the **objective function**, is an equation that states the goal, or objective, of the model. In the same example of the toy manufacturer, we want to know the quantities of toy boats and toy cars to produce. However, the goal of the manufacturing plant's production may be to increase profit. If we know that we can profit \$5 for every toy boat and \$4 for every toy car, then our objective function is:

$$\text{Maximize } 5x_1 + 4x_2$$

In other words, we want profit to drive us in determining the quantity of boats and cars to produce. Objective functions are either **maximized** or **minimized**; most applications involve maximizing profit or minimizing cost.

The third part of the math program, the **constraints**, are the limitations of the problem. That is, if we want to maximize our profit, as in the toy manufacturer example, we could produce as many toys as possible if we did not have any limits. However, in most

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

realistic situations, there are certain limitations, or constraints, that we must consider. Constraints can be a limited amount of resources, labor, or requirements for a particular demand. These constraints are also written as equations, or inequalities in terms of the decision variables. That is, if we can only use 20 hours of labor in a week and we need 0.5 hours to produce each toy boat and 0.3 hours to produce each toy car, then we write our constraint as follows:

$$0.5x_1 + 0.3x_2 \leq 20$$



Summary

Decision Variables: Variables assigned to quantities to be determined.
Objective Function: An equation that states the objective of a model.
Constraints: Equations or inequalities that state limits or requirements of a problem.

8.2.2 Linear, Integer, and Nonlinear Programming

There are three main categories of problems for which we can use the above mathematical program parts: **Linear Programming (LP)**, **Integer Programming (IP)**, and **Nonlinear Programming (NLP)**.

Linear programming problems have a linear objective function and linear constraints. That is, there are no variables of multiple powers such as x^2 and x^3 , and no terms involving two variables such as x_1x_2 . In addition, LP problems consist of decision variables with any range or interval of values, $x \geq 0$ or $x \leq 0$. An example of an LP would be a production problem in which we want to maximize profit by determining how many of several different product types we want to produce. The objective function could therefore be expressed as:

$$z = \sum_{i=1}^n p_i x_i$$

Where i = product number for n products, p_i = profit per product i , x_i = amount produced of product i . This is therefore a linear objective function. If we assume that the constraints are also linear, then this is a linear programming problem. We will revisit this example in more detail in section 8.2.2.

Integer Programming is related to linear programming in that both the objective function and constraints are linear; however, the decision variables can have only integer values in a given range. Integer programming is also applied when *decision variables* are **Boolean**, which means that they only take the values *true* or *false*, *yes* or *no*, *go* or *no go* – all of which are mathematically represented as 0 or 1, respectively. An example of an IP would be a capital budgeting problem in which we want to decide which projects to invest in and which not to invest in. This decision is a yes/no decision that can be represented by the following objective function:

$$z = \sum_{i=1}^n y_i x_i$$

Chapter 8: Using The Excel Solver To Solve Mathematical Programs


Where i = project number for n projects, y_i = NPV per project i , x_i = decision to invest per project i . This is therefore a linear objective function. What makes it an integer programming problem is that we limit the values of x_i to 1 or 0 to reflect whether or not we have or have not invested in a project, respectively. We will revisit this example as well in more detail in section 8.3.2.

Nonlinear programming problems do not have a linear objective function and/or constraints. NLP problems use more challenging methods to solve these complex equations. An example of a NLP would be a warehouse location problem in which we are trying to determine a warehouse location that minimizes the distance traveled in shipments to/from several facilities. The sum of the distances from multiple facilities to this warehouse would be calculated as follows:

$$z = \sum_{i=1}^n \sqrt{(x_i - x_w)^2 + (y_i - y_w)^2}$$

Where i = facility number for n facilities, x_i and y_i = coordinates of each facility i , x_w and y_w = coordinates of the warehouse. This is therefore a nonlinear objective function. Even if the constraints are all linear, it is still a nonlinear programming problem. We will also revisit this example in more detail in section 8.3.3.

Several **algorithms**, or methods of solving a mathematical program, are specific to Linear, Integer, and Nonlinear programming problems. They must simultaneously consider each constraint in conjunction with the objective function. We will use the **Excel Solver**, or the **Standard Solver**, to solve these problems. The Solver uses an algorithm called the *Simplex Method* to solve both the LP and IP programs. We will use an advanced version of the Solver, called the **Premium Solver**, to solve NLP programs. (Note: We use the Educational version of the Premium Solver.) The Premium Solver can be used to solve NLP problems with an algorithm called the Genetic Algorithm. (Note that the Standard Solver can solve NLP programs, but it is less efficient than the Premium Solver. The Standard Solver may even cause errors for some IP or LP problems; we will discuss these in the “Limitations and Manipulations” section. Also note that the Premium Solver can solve LP and IP programs, but we will use the Standard Solver to solve these problems in this chapter.) We will compare the algorithms used in the two versions of the Solver in more detail later in the chapter.

 Summary	Linear Programming	Both the objective function and the constraints are linear. Decision variables can have any range or interval of values.
	Integer Programming	An LP in which decision variables can only take integer values in a given range or Boolean values.
	Boolean	Decision variables that only take the values 0 or 1.
	Nonlinear Programming	Either the objective function or constraints or both are not linear.
	Algorithm	A method of solving a mathematical model.
	Standard Solver	Solves LP and IP models using Simplex Method.
	Premium Solver	Can also solve NLP models using Genetic Algorithm

8.3 The Excel Solver

We will now discuss how to operate these two versions of the Solver. In general, the Solver must understand the problem's mathematical program parts, which we take care of by preparing our spreadsheet to contain distinct cells for the decision variables, constraints, and objective function. We must then tell the Solver if we want to minimize or maximize the problem, or if we want to solve it for a particular value of the objective function. There are also several options that we can apply to give more specific instructions to the Solver for solving the problem.

(Note: To find the Solver, go to *Tools > Solver* from the menu options. If you do not see *Solver* in the *Tools* menu, you must first choose the *Solver Add-In*. To do so, select the *Add-In* option from the *Tools* menu. A small dialog box will appear; from there, select *Solver Add-In* from the list. If you do not see *Solver Add-In* in the *Add-In* list, click *Browse* and look for the *Solver.xla* file from the following directory: *C Drive > Program Files > Microsoft Office > Office (or Office10) > Library > Solver*. Double-click this file. Now you should find *Solver Add-In* in the list; check the box next to it. Restart Excel. If you do not find the *Solver.xla* file, go to the *Add-Ins* window as explained above; select *Solver Add-in* and press *OK*. Insert the MS Office CD in CD-ROM drive when asked.)

8.3.1 The Solver Steps

To operate the *Solver*, we must follow a short sequence of steps: 1) read and interpret the problem; 2) prepare the spreadsheet; and 3) solve the model and review the results. We will now describe these steps in detail for both the Standard Solver and the Premium Solver.

The Standard Solver

STEP 1: Read and Interpret the Problem

We must first determine the type of problem that we are dealing with (linear programming, Integer Programming, or nonlinear programming) and outline the model parts. Whether the problem is an LP, IP, or NLP model does not affect the model parts but does affect the Options that we specify for the Solver. They may also require some additional constraint specifications. In each case, we still need to determine the decision variables, the objective function, and the constraints. We need to write these mathematically, with the objective function and constraints in terms of the decision variables.

STEP 2: Prepare the spreadsheet

Next, we transfer these parts of the model into our Excel spreadsheet, clearly defining each part of our model in the spreadsheet. The *Solver* interprets our model according to the location of these model parts on the spreadsheet.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

STEP 2.1: Place the Input Table

Usually the input for the problem is provided for us. We just need to place it on the spreadsheet in the form of a table. We reference this input when forming our constraint and objective function formulas.

STEP 2.2: Set the Decision Variables Cells

Next, we list the decision variables in individual cells with an empty cell next to each one. The *Solver* places values in these cells for each decision variable as it solves the model. We recommend naming the range of decision variables for easier reference in constraint and objective function formulas.

STEP 2.3: Enter the Constraint Formulas

Now we place the constraint equations in the spreadsheet; we enter them separately using formulas, with an optional description next to each constraint. As each constraint is in terms of the decision variables, all of these formulas must be in terms of the decision variable cells already defined.

Another important consideration when laying out the constraints in preparation for the *Solver* is that there must be individual cells for the right-hand side (RHS) values as well. We should also place all inequality signs in their own cells. This organization will become clear once we explain how the *Solver* interprets our model.

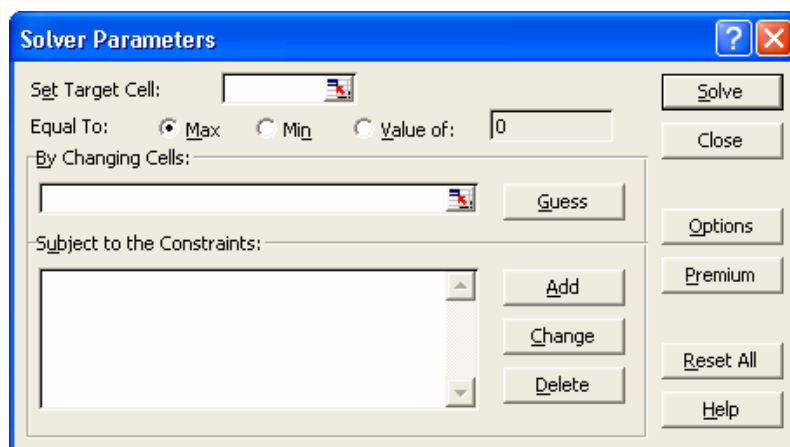
Another advantageous way to keep our constraints organized as we use the *Solver* is to name cells. We can also group constraints that have the same inequality signs. The benefit of this habit will become apparent once we input the model parts for the *Solver*.

STEP 2.4: Enter the Objective Function Formula

We can now place our objective function in a cell by transforming this equation into a formula in terms of the decision variables. The spreadsheet is now prepared for the *Solver* with all three parts of the model clearly displayed.

STEP 3: Solve the Model with the Solver

The *Solver* can now interpret this information and use algorithms to solve the model. The *Solver* receives the decision variables, constraint equations, and objective function equation as input into a hidden programming code that applies the algorithm to the data. We will explain in more detail how this programming works when we discuss VBA. To use *Solver*, we choose *Tools > Solver* from the menu; the window in Figure 8.3 then appears.



Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Figure 8.3 The Solver Parameters window reads the decision variables, constraints, and objective function as parameters of the model.

Step 3.1: Set Target Cell

Three main areas of this window correspond to the three parts of the model. The first area is the *Target Cell*, which refers to the location of the formula for the objective function; it can also be called the Set Cell. To select this cell, we click on the button to the right of the text box and highlight our objective function cell, or we can enter the cell reference directly. In describing this part of the model, we must also determine if we are maximizing or minimizing the objective function. We can specify this by selecting the *Max* or *Min* options below the *Target Cell* textbox. We also have the option of solving the objective function to reach a particular value. In that case, we select *Value of* and enter the value that we want the objective function to achieve.

STEP 3.2: Select Changing Cells

In the next area of the window, we select our decision variables, called *Changing Cells*. We click on the button to the right of the text box and highlight our decision variable cells, or we can enter the cell reference or a range name directly. Note that if we have already named the range on our spreadsheet, that name appears automatically in the Solver dialog box after the range is selected. The *Solver* places different values in these *Changing Cells* and checks the constraints and the objective function value against the formulas that we have provided until all are simultaneously satisfied.

STEP 3.3: Add Constraints

In the third area of the *Solver* window, we specify our constraints. To do so, we press the *Add* button next to the large *Constraint* textbox. The smaller window shown in Figure 8.4 then appears. (Note: The *Comment* and *Report* options in this dialog box may not appear when using the Standard Solver). We must include the following three pieces of information in each added constraint: the cell with the constraint formula; the inequality sign; and the cell with the *RHS* value or a directly entered numerical value. We click *Add* to define the next constraints.

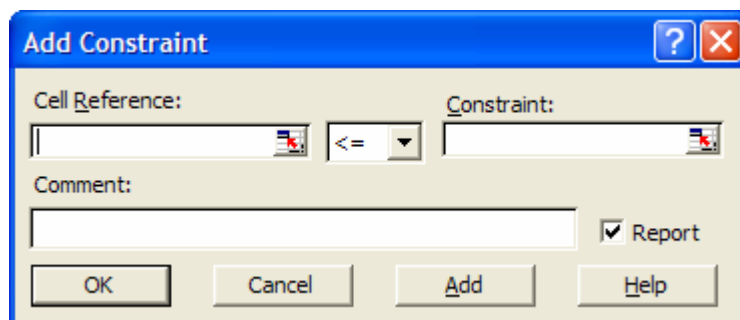


Figure 8.4 Adding Constraints involves selecting the cell with the equation formula, choosing the inequality or equality sign, and selecting the cell with the RHS value. *Comment* and *Report* are optional.

Excel also allows us to define more than one constraint at a time. By grouping constraints that have the same inequality signs, we can select an entire range of constraint formulas and RHS values and choose the common inequality sign. Naming

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

constraints with the same inequality can also clarify what we add to the *Solver* and prevent us from making any mistakes. If multiple ranges are not adjacent, we can select them by holding down the CTRL key or by separating them with commas in the Constraint window.

We have now added all of our constraints, so we press *OK* and return to the *Solver* window. We can observe all of our constraints in the *Constraint* textbox. Note that we can press the *Change* button to edit any of the constraints and the *Delete* button to erase them.

STEP 3.4: Set Solver Options

We should review and modify *Solver Options* before we solve the model. Clicking the *Options* button takes us to a new *Options* window (see Figure 8.5). The first option is the time that the *Solver* should take to find a solution to the model. We can set a *Maximum Time* at a small number if we want a quick answer or at a large number if we want a slower but more precise answer. The *number of Iterations* is also an option; it affects the length of time that the *Solver's* algorithm runs and the accuracy of the *Solver* solution. The *Precision*, *Tolerance*, and *Convergence* options also affect the accuracy and range of the *Solver* solutions. We can change any of these options or leave them at their default values.

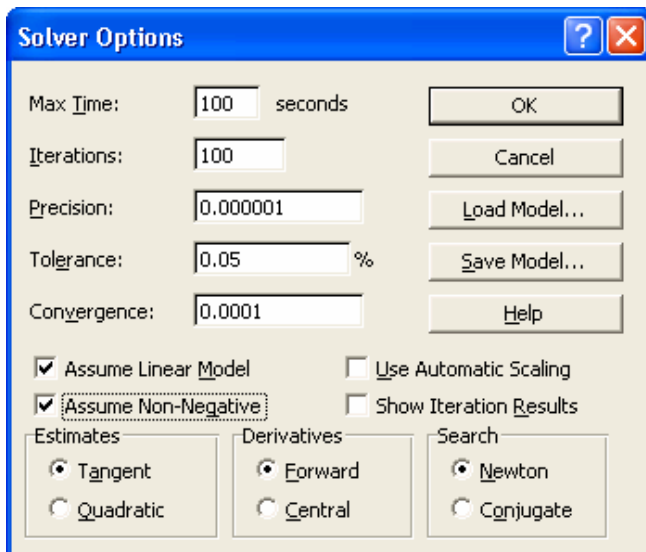


Figure 8.5 The Options window allows a user to specify how Solver will work with a model.

For now, we leave the *Estimates*, *Derivatives*, and *Search* options at their default values. *Assuming a Linear Model* and *Assuming Non-Negative* decision variables are essential options to consider. It is important to always check the *Assume a Linear Model* option when solving a Linear programming problem in order to guarantee an optimal solution.

STEP 3.5: Solve the Model and Review the Results

We now press *Solve* to run the *Solver*. After it has found a solution, the window in Figure 8.6 appears and we are able to observe the *Solver* solution in the background on our spreadsheet. This window reveals one of three results: "Solver found a solution. All constraints and optimality conditions are satisfied" "Solver could not find a feasible

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

solution”; or “Target Cell values (or Set Cell values) could not converge.” The spreadsheet cells we set when formulating the model now have values for the decision variable cells. Therefore, they also have values in the constraint and objective function cells since they contain formulas referencing the decision variable cells. We can confirm that all constraints have been satisfied by noting that the values in the constraint cells with the *Solver* solution are all less than or equal to, or greater than or equal to the *RHS* values, respectively.

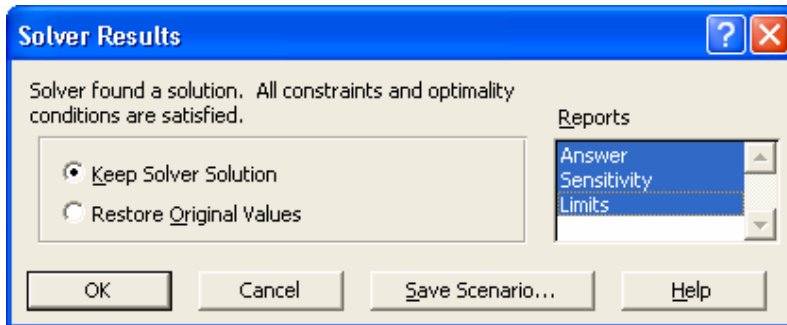



Figure 8.6 Excel displays this window after we run Solver and find a solution.

After reviewing this solution, we can choose to *Keep the Solver Solution* or *Restore our Original Values*. We can also opt to have some extra reports made from the *Solver* solution: the *Answer* report, the *Sensitivity* report, and the *Limits* report. We will discuss these reports in more detail later.

 Summary	<p><u>Steps for using the Standard Solver</u></p> <p>Step 1: Read and Interpret the Problem.</p> <ul style="list-style-type: none">1.1: Decision Variables.1.2: Objective Function.1.3: Constraints. <p>Step 2: Prepare the Spreadsheet.</p> <ul style="list-style-type: none">2.1: Place the Input Table.2.2: Set the Decision Variables Cells.2.3: Enter the Constraint Formulas.2.4: Enter the Objective Function Formula. <p>Step 3: Solve the model with <i>Solver</i>.</p> <ul style="list-style-type: none">3.1: Set the Target Cell.3.2: Select the Changing Cells.3.3: Add the Constraints.3.4: Set the <i>Solver</i> Options.
---	---

Premium Solver

Steps 1 and 2 are exactly the same when using the Premium Solver. Even the sub-steps 3.1, 3.2, and 3.3 are no different. However, depending on which algorithm we choose to use, there is a different set of Options for sub-step 3.4.

To use the Premium Solver, we simply press the *Premium* button on the Solver window. (Note that the *Premium* button now becomes a *Standard* button to allow us to switch back to the standard Solver.) The main Solver window has a drop down list to select

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

which Solver version, or which set of algorithms, we want to use to solve our problem (see Figure 8.7). The **Standard GRG Nonlinear** and the **Standard Simplex LP** versions are basically the same as the *Standard Solver* for LP, IP, and NLP models. We use the **Standard Evolutionary** version to solve our more complex NLP models; we will refer to it as the *Premium Solver*.

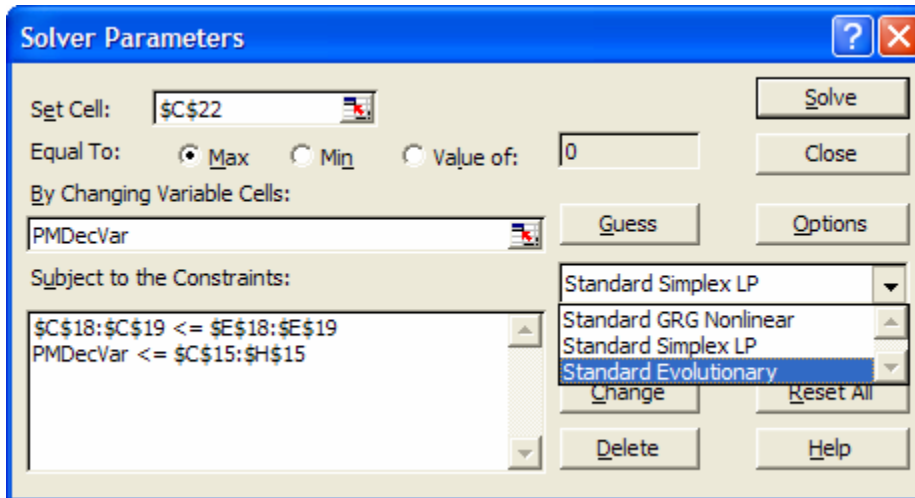


Figure 8.7 The Premium Solver offers three different Solver methods.

The *Evolutionary Solver* employs the *Genetic Algorithm* to solve Nonlinear problems. (We will discuss this in more detail in a later section.) Once we select this version, we click on the *Options* button to see the window shown in Figure 8.8. We can observe several similar options as in the *Standard Solver* such as *Max Time*, *Iterations*, *Precision*, and *Assume Non-Negative*.

The *Population Size* and *Mutation Rate* are parameters associated with the Genetic Algorithm. The *Population Size* basically sets the initial number of feasible solutions. The *Mutation Rate* determines how often the “Mutation” part of the Genetic Algorithm is performed. We recommend setting the *Population Size* to at least 50; if there are several variables and constraints, it may be wiser to have a *Population Size* of 100 or 1,000. The *Mutation Rate* can be small, but it usually performs better at 0.25 than at 0.075. The Local Search options can be specified to further determine the behavior of the Genetic Algorithm. At this point, we will not further discuss this algorithm.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

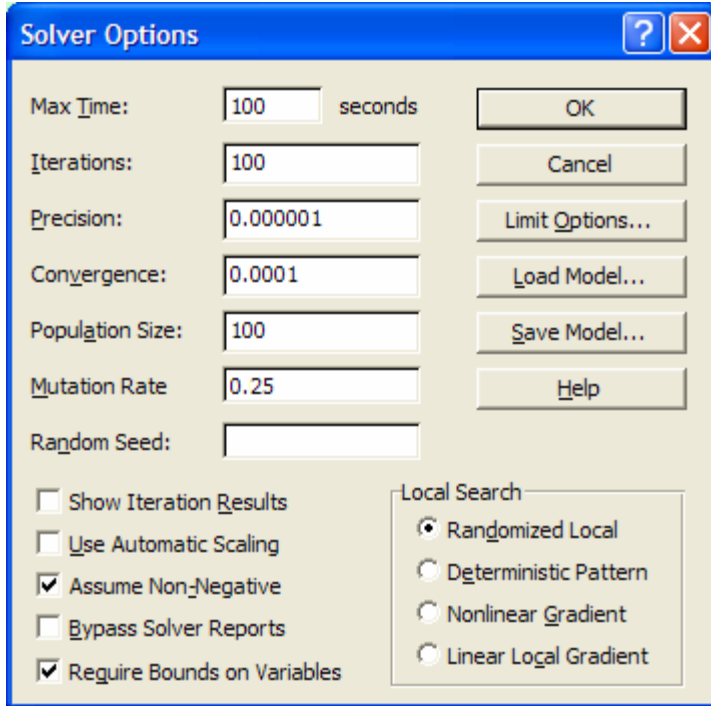


Figure 8.8 The Options for the Standard Evolutionary Solver.

After we have specified these options, we can press *Solve* to run the Solver. The Solver Results window as shown in Figure 8.9 then appears. We may see “Solver found Solution” or “Stop chosen when...” as possible messages in this window. We also have a different list of available Solver Reports, including the Population Report. This report provides the best, mean, maximum, and minimum values along with the standard deviation of the *Changing Cells*, or decision variable, values across all iterations.

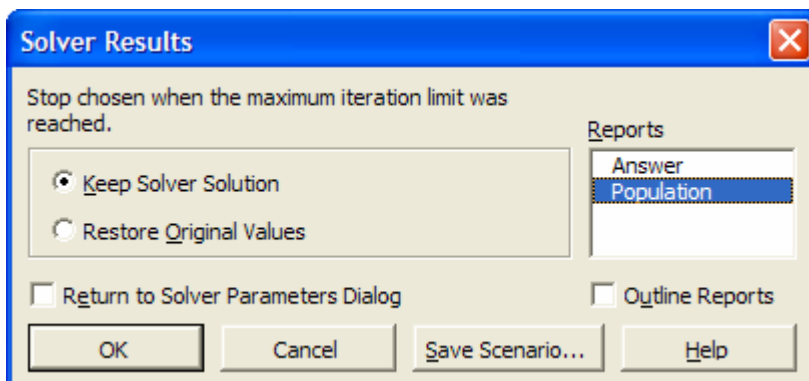


Figure 8.9 The Solver Reports for the Evolutionary Solver include a Population Report.

8.2.2 Solver Example

Now that we are familiar with the three steps of using the *Solver* to determine the solution of a mathematical model, let's work through a more complicated example.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Problem Description:

A company produces six different types of products. They want to schedule their production to determine how much of each product type should be produced in order to maximize their profits. This situation is known as the “Product Mix” problem.

Production of each product type requires labor and raw materials, but the company is limited by the amount of resources available. There is also a limited demand for each product, and no more than this demand per product type should be produced. Input tables for the necessary resources and the demand are provided.

This is a Linear programming problem, as the constraints and objective function are linear with respect to the decision variables, as we will see below. We therefore decide to use the *Standard Solver*. Let’s now outline the model parts.

Decision Variables:

For the amount produced of each product type, we use the following variable representation:

$$x_1, x_2, x_3, x_4, x_5, x_6$$

In other words, x_1 is the amount produced of product 1, x_2 is the amount produced of product 2, etc. Note that all of these decision variables are non-negative; that is, we cannot produce a negative amount of any product type.

Objective Function:

Maximize Profit. Profit is calculated as the sum of the array multiplication of the unit profit, p , and the amount produced of each product type. We write this equation as follows:

$$z = \sum_i p_i x_i \quad \text{for } i = 1 \text{ to } 6$$

Here, p_1 is the amount of profit gained per unit of product 1. Therefore, $p_1 * x_1$ is the amount of profit per unit of product 1 times the number of units produced of product 1, thus yielding the total profit from product 1. The same follows for the other products, 2 through 6.

Constraints:

There are two resource constraints: labor, l , and raw material, r . Available amounts are provided for each resource, and required amounts are provided for the production of each product type. We therefore say that the sum of the array multiplication of the resource requirements and the amount produced of each product type must be less than or equal to the amounts available of each resource. These equations are written:

$$\begin{aligned} & \text{Labor Constraint:} \\ & = \sum_i l_i x_i \leq \text{available labor} = 4500 \end{aligned}$$

$$\begin{aligned} & \text{Raw Material Constraint:} \\ & = \sum_i r_i x_i \leq \text{available raw material} = 1600 \end{aligned}$$

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Here l_1 is the amount of labor required per unit produced of product 1. Similarly, r_1 is the amount of raw material required per unit produced of product 1. Therefore, the equations represent the total labor and raw material needed for all products.

There is also a constraint that all demand, D , should be met (but is not enforced), and no extra amount should be produced. Therefore, the amount produced of each product type must be less than or equal to the given demand quantities. This constraint can be written as follows:

$$\text{Demand Constraint:} \\ x_i \leq D_i \quad \text{for } i = 1 \text{ to } 6$$

Next, we set up the spreadsheet to list these parts of the model in such a way that the Solver can read them clearly.

The Spreadsheet:

In Figure 8.10, we observe the overall spreadsheet layout. We have organized our cells by Input, Decision Variables, Constraints, and Objective Function. The Input table was given. We calculate the Unit Profit row by subtracting the Variable Cost from the Unit Price.

	A	B	C	D	E	F	G	H	I
1	Product Mix								
2									
3	Input		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	Available
4		Labor	6	5	4	3	2.5	1.5	4500
5		Raw Material	3.2	2.6	1.5	0.8	0.7	0.3	1600
6		Unit price	\$12.50	\$11.00	\$9.00	\$7.00	\$6.00	\$3.00	
7		Variable cost	\$6.50	\$5.70	\$3.60	\$2.80	\$2.20	\$1.20	
8									
9		Unit profit cont.	\$6.00	\$5.30	\$5.40	\$4.20	\$3.80	\$1.80	
10									
11									
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	
13		Amount produced							
14			<=	<=	<=	<=	<=	<=	
15		Demand	960	928	1041	977	1084	1055	
16									
17	Constraints				Available				
18		Labor Used	0.00	<=	4500				
19		Raw Material Used	0.00	<=	1600				
20									
21	Objective Function								
22		Profit	\$0.00						

Figure 8.10 The Spreadsheet Layout for the Product Mix.

We have also labeled some ranges on the spreadsheet. We have named the Decision Variable range “PMDecVar,” the Labor resource requirement row “PMLabor,” the Raw Material resource requirement row “PMRawMat,” and the Unit Profit row “PMUnitProfit.” These names will be helpful for writing the constraint and objective function formulas as well as for inserting cell references in the Solver.

To prepare the constraint and objective function formulas, we use the SUMPRODUCT function. Remember from Chapter 4 that this function takes two arrays, or ranges, as

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

parameters for which it will multiply and sum all values. Referring to the equations written above and the range names created, we write the constraint formulas as follows:

Labor Constraint:
 $=SUMPRODUCT(PMDecVar, PMLabor)$

Raw Material Constraint:
 $=SUMPRODUCT(PMDecVar, PMRawMat)$

The right-hand side values are equal to the “Available” amounts from the Input table (see Figure 8.11).

C17		fx =SUMPRODUCT(PMDecVar,PMLabor)							
	A	B	C	D	E	F	G	H	I
3	Input		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	Available
4		Labor	6	5	4	3	2.5	1.5	4500
5		Raw Material	3.2	2.6	1.5	0.8	0.7	0.3	1600
11									
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	
13		Amount produced							
14									
15	Constraints								
16									Available
17		Labor Used	0.00	<=					4500
18		Raw Material Used	0.00	<=					1600
19									

Figure 8.11 The Labor and Raw Material constraint formulas use the SUMPRODUCT function.

For the demand constraint, we simply need to ensure that the values in our decision variable range are less than each of the corresponding values in the “Demand” range. We do not require a formula for this constraint (see Figure 8.12).

11									
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	
13		Amount produced							
14			<=	<=	<=	<=	<=	<=	
15		Demand	960	928	1041	977	1084	1055	
16									

Figure 8.12 The Demand Constraint does not require a formula.

The objective function formula is also written with the SUMPRODUCT function (see Figure 8.13). Referring to the equation and range names above, we type the following formula:

$$=SUMPRODUCT(PMUnitProfit, PMDecVar)$$

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E	F	G	H
9								
10		Unit profit cont.	\$6.00	\$5.30	\$5.40	\$4.20	\$3.80	\$1.80
11								
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6
13		Amount produced						
14								
20	Objective Function							
21		Profit	\$0.00					

Figure 8.13 The objective function formula employs the SUMPRODUCT function.

After appropriately preparing the cells, we can now use the *Solver* to find our solution. In Figure 8.14, the Solver dialog box appears. We have set the Target Cell to the location of the objective function formula, the Changing Cells are set to the empty decision variable cells, which we named “PMDecVar,” and the “Constraints” box shows the left and right sides of the constraint equations with the corresponding inequalities. Here, the labor and raw material constraints are listed first, then the demand constraint is listed with the decision variable cells. We have also selected the two options *Assume Linear Model* and *Assume Non-Negative*.

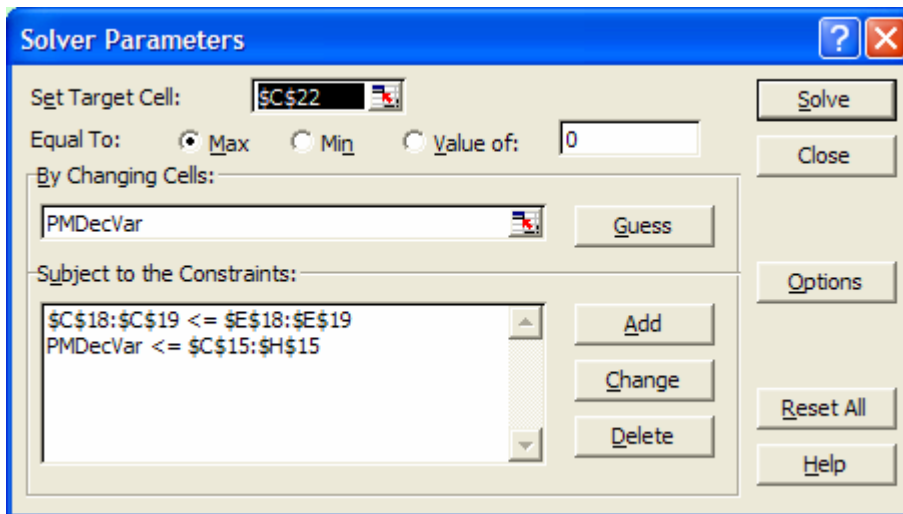


Figure 8.14 The Solver dialog box.

The Solver Results window reveals that a solution was successfully found (see Figure 8.15). We can view the final results in Figure 8.16. Notice that all constraints are met. The company now knows how much to produce of each product type and what their maximum profit will be. (There may be multiple solutions, but we cannot find them or determine if there are more solutions with the Standard Solver.)

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

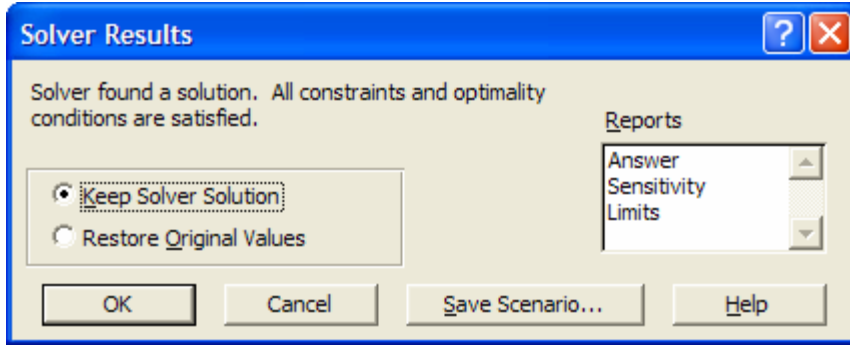


Figure 8.15 The Solver finds a solution.

	A	B	C	D	E	F	G	H	I
1	Product Mix								
2									
3	Input		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	Available
4		Labor	6	5	4	3	2.5	1.5	4500
5		Raw Material	3.2	2.6	1.5	0.8	0.7	0.3	1600
6		Unit price	\$12.50	\$11.00	\$9.00	\$7.00	\$6.00	\$3.00	
7		Variable cost	\$6.50	\$5.70	\$3.60	\$2.80	\$2.20	\$1.20	
8									
9		Unit profit cont.	\$6.00	\$5.30	\$5.40	\$4.20	\$3.80	\$1.80	
10									
11									
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	
13		Amount produced	0.00	0.00	0.00	596.67	1084.00	0.00	
14			<=	<=	<=	<=	<=	<=	
15		Demand	960	928	1041	977	1084	1055	
16									
17	Constraints					Available			
18		Labor Used	4500.00	<=		4500			
19		Raw Material Used	1236.13	<=		1600			
20									
21	Objective Function								
22		Profit	\$6,625.20						

Figure 8.16 The final Solver solution.

Infeasibility

An infeasible problem is one in which at least one of the constraints cannot be met. For this example, we consider infeasibility based on the “Demand” Constraint. Note in the solution above that some of the product types did not meet their demand since the “Demand” Constraint inequalities were \leq in order to avoid the cost of over-production. Now let’s assume that the company insists that demand is always met, even if some waste is produced. We now need to change the “Demand” Constraint inequality from \leq to \geq . To do so, we select the “Demand” Constraint in the Solver window from the list of constraints and use the *Change* button (see Figure 8.17).

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

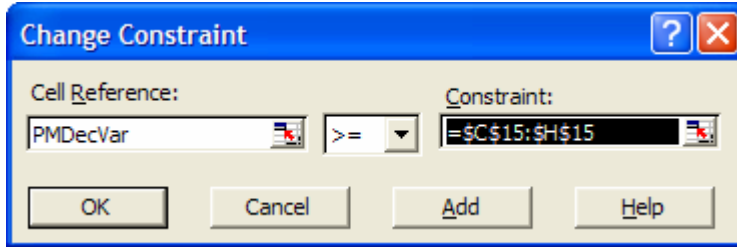


Figure 8.17 Changing the Demand Constraint inequality sign.

However, now when we press *Solve*, the Solver Results window conveys that the Solver could not find a feasible solution with this modified constraint (see Figure 8.18). If there are not enough resources available to meet the demand, then the solution is infeasible. The result of this infeasible solution is shown in Figure 8.19.

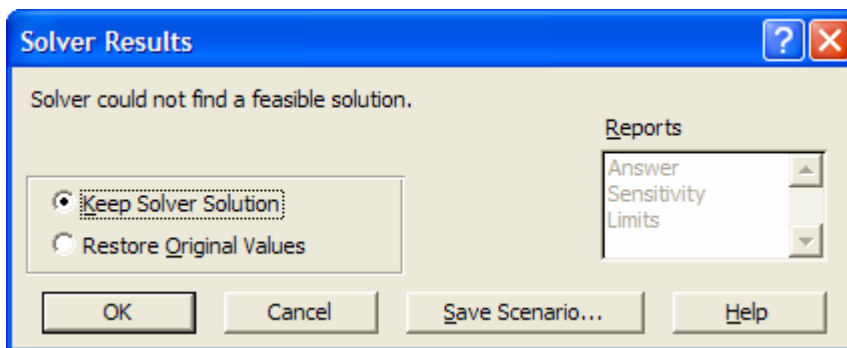


Figure 8.18 No feasible solution is found.

	A	B	C	D	E	F	G	H	I
1	Product Mix								
2									
3	Input		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	Available
4		Labor	6	5	4	3	2.5	1.5	4500
5		Raw Material	3.2	2.6	1.5	0.8	0.7	0.3	1600
6		Unit price	\$12.50	\$11.00	\$9.00	\$7.00	\$6.00	\$3.00	
7		Variable cost	\$6.50	\$5.70	\$3.60	\$2.80	\$2.20	\$1.20	
8									
9		Unit profit cont.	\$6.00	\$5.30	\$5.40	\$4.20	\$3.80	\$1.80	
10									
11									
12	Decision Variables		Product Type 1	Product Type 2	Product Type 3	Product Type 4	Product Type 5	Product Type 6	
13		Amount produced	960.00	928.00	1041.00	977.00	1084.00	1055.00	
14			>=	>=	>=	>=	>=	>=	
15		Demand	960	928	1041	977	1084	1055	
16									
17	Constraints				Available				
18		Labor Used	21787.50	<=	4500				
19		Raw Material Used	8903.20	<=	1600				
20									
21	Objective Function								
22		Profit	\$26,421.40						

Figure 8.19 The infeasible result.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Unboundedness

An unbounded problem is one in which the objective function can reach an unreasonably large number (if we are maximizing) or small number (if we are minimizing). Such a situation implies that the constraints are not inclusive enough. We consider unboundedness for this example in the case where we are able to maximize the profit to an unreasonably huge value. Now, let's suppose that there are no "Demand" Constraints. We select the "Demand" Constraint from the constraint list in the Solver window and press *Delete*. Let's also assume at this point that negative decision variable values are allowed. To do so, we re-visit the *Options* window and uncheck the *Assume Non-Negative* option. The updated model now appears in the Solver window, displayed in Figure 8.20.

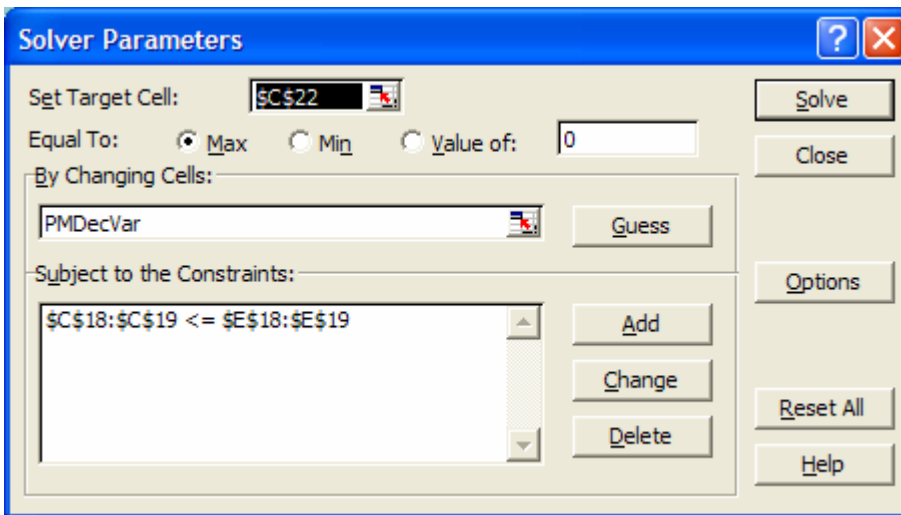


Figure 8.20 The updated Solver model does not have the "Demand" Constraints.

This time, after pressing *Solve*, the Solver Results window tells us that a solution did not converge (see Figure 8.21). That means that without the "Demand" Constraints or non-negativity assumption we are able to achieve extremely large profit values.

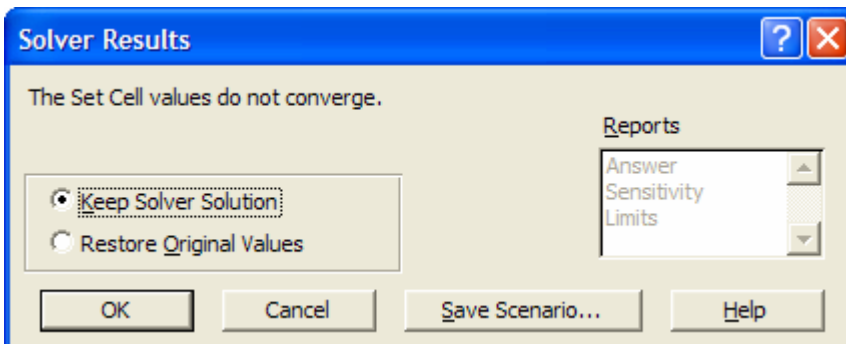


Figure 8.21 The Solver solution did not converge without the "Demand" Constraints and non-negativity assumption.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

8.2.3 Understanding Solver Reports

The three main reports available when using the Standard Solver are the *Answer report*, the *Sensitivity report*, and the *Limits report*. We will now briefly review what information is contained in these reports. (Note that there are some limitations to the Solver Reports that we will discuss in the “Limitations and Manipulations” section.)

	A	B	C	D	E	F	G
1	Microsoft Excel 10.0 Answer Report						
2	Worksheet: [Chapter 8 Examples_Solver.xls]Product Mix						
3	Report Created: 1/8/2004 8:54:16 AM						
4							
5							
6	Target Cell (Max)						
7	Cell		Name	Original Value	Final Value		
8	\$C\$22		Profit <=	\$0.00	\$6,625.20		
9							
10							
11	Adjustable Cells						
12	Cell		Name	Original Value	Final Value		
13	\$C\$13		Amount produced Product Type 1	0.00	0.00		
14	\$D\$13		Amount produced Product Type 2	0.00	0.00		
15	\$E\$13		Amount produced Product Type 3	0.00	0.00		
16	\$F\$13		Amount produced Product Type 4	0.00	596.67		
17	\$G\$13		Amount produced Product Type 5	0.00	1084.00		
18	\$H\$13		Amount produced Product Type 6	0.00	0.00		
19							
20							
21	Constraints						
22	Cell		Name	Cell Value	Formula	Status	Slack
23	\$C\$18		Labor Used <=	4500.00	\$C\$18<=\$E\$18	Binding	0
24	\$C\$19		Raw Material Used <=	1236.13	\$C\$19<=\$E\$19	Not Binding	363.8666667
25	\$C\$13		Amount produced Product Type 1	0.00	\$C\$13<=\$C\$15	Not Binding	960
26	\$D\$13		Amount produced Product Type 2	0.00	\$D\$13<=\$D\$15	Not Binding	928
27	\$E\$13		Amount produced Product Type 3	0.00	\$E\$13<=\$E\$15	Not Binding	1041
28	\$F\$13		Amount produced Product Type 4	596.67	\$F\$13<=\$F\$15	Not Binding	380.3333333
29	\$G\$13		Amount produced Product Type 5	1084.00	\$G\$13<=\$G\$15	Binding	0
30	\$H\$13		Amount produced Product Type 6	0.00	\$H\$13<=\$H\$15	Not Binding	1055
31							
32							

Figure 8.22 The Answer Report.

The Answer Report provides the original and final values of the Target Cell, Changing Cells, or Adjustable Cells, and the Constraints (see Figure 8.22). It also gives the reference of all of these cells on the spreadsheet. The names for each cell are based on the row and column labels next to the tables on our spreadsheet. The formulas for the constraints are provided only as references for where they formulas are held; in other words, any functions used are not reported here. The Status of the Constraint part of the report conveys whether or not a constraint is binding. A constraint is binding when its slack value is zero.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E	F	G	H
1	Microsoft Excel 10.0 Sensitivity Report							
2	Worksheet: [Chapter 8 Examples_Solver.xls]Product Mix							
3	Report Created: 1/8/2004 8:54:16 AM							
4								
5								
6	Adjustable Cells							
7								
8	Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease	
9	\$C\$13	Amount produced Product Type 1	0.00	-2.40	6	2.4	1E+30	
10	\$D\$13	Amount produced Product Type 2	0.00	-1.70	5.3	1.7	1E+30	
11	\$E\$13	Amount produced Product Type 3	0.00	-0.20	5.4	0.2	1E+30	
12	\$F\$13	Amount produced Product Type 4	596.67	0.00	4.2	0.36	0.15	
13	\$G\$13	Amount produced Product Type 5	1084.00	0.30	3.8	1E+30	0.3	
14	\$H\$13	Amount produced Product Type 6	0.00	-0.30	1.8	0.3	1E+30	
15								
16	Constraints							
17								
18	Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease	
19	\$C\$18	Labor Used <=	4500.00	1.40	4500	1141	1790	
20	\$C\$19	Raw Material Used <=	1236.13	0.00	1600	1E+30	363.8666667	
21								
22								

Figure 8.23 The Sensitivity Report.

The Sensitivity Report provides information about the Changing Cells and the Constraints (see Figure 8.23) as well as their final values. The reduced cost or shadow price and the allowable increase and decrease indicate how much flexibility can be allowed with any of these values in order to achieve the desired objective function value.

	A	B	C	D	E	F	G	H	I	J	K
1	Microsoft Excel 10.0 Limits Report										
2	Worksheet: [Chapter 8 Examples_Solver.xls]Limits Report 2										
3	Report Created: 1/8/2004 8:54:16 AM										
4											
5											
6											
7	Cell	Target Name	Value								
8	\$C\$22	Profit <=	\$6,625.20								
9											
10											
11	Cell	Adjustable Name	Value	Lower Limit	Target Result	Upper Limit	Target Result				
12	\$C\$13	Amount produced Product Type 1	0.00	0.00	6625.20	0.00	6625.20				
13	\$D\$13	Amount produced Product Type 2	0.00	0.00	6625.20	0.00	6625.20				
14	\$E\$13	Amount produced Product Type 3	0.00	0.00	6625.20	0.00	6625.20				
15	\$F\$13	Amount produced Product Type 4	596.67	0.00	4119.20	596.67	6625.20				
16	\$G\$13	Amount produced Product Type 5	1084.00	0.00	2506.00	1084.00	6625.20				
17	\$H\$13	Amount produced Product Type 6	0.00	0.00	6625.20	0.00	6625.20				
18											
19											
20											

Figure 8.24 The Limits Report.

The Limits Report provides information about the Target Cell and the Changing Cells (see Figure 8.24); it also includes the value of each cell. The lower and upper limits of the Changing Cells are listed next to the corresponding Target Cell value that would result if the Changing Cell had the limit value.

8.4 Applications of the Solver

Mathematical models are utilized in many fields to formulate a problem into equations that can be solved using algorithms. The *Solver* allows managers and investors to solve these problems without knowing how the algorithms work. However, each problem must still be interpreted so that the *Solver* can read the correct *Target Cells*, *Changing Cells*, and *Constraints*. Below are a few examples of applications with the correct interpretation of these three model parts. These examples are grouped by Linear, Integer, and Nonlinear programming problems. It is important to ensure that constraints and options are specified to reflect what type of problem is being solved.

8.4.1 Transportation Problem

An example of a linear programming problem is a Transportation problem. A company ships their products from three different plants (one in Los Angeles, one in Atlanta, and one in New York City) to four regions of the United States (East, Midwest, South, West). Each plant has a limited capacity on how many products can be sent out, and each region has a demand of products that they must receive. There is a different transportation cost between each plant, or each city, and each region. The company wants to determine how many products each plant should ship to each region in order to minimize the total transportation cost.

The Input for this problem is in the first table in Figure 8.25. It contains the transportation cost between each city and each region. It also displays the capacity per plant and the demand per region.

The decision variables are the amount to ship from each plant to each region. We have created a table with empty cells for these decision variables.

There are two constraints for this problem: "Demand" and "Capacity." We need to ensure that the total number of products shipped from a plant (to each region) is less than or equal to its capacity, and we also need to ensure that the total number of products received by a region (from each plant) is greater than or equal to its demand. We have used the SUM function to create a column and row for these respective constraints. We have then copied the capacity and demand from the Input table as the RHS value.

The objective function is to minimize the total transportation costs. We need to sum the array multiplication between the given costs between each plant and region with the amount shipped between each plant and region. To do so, we use the SUMPRODUCT function. We have actually also named the range of decision variables as "TransShipped" and the range of input costs as "TransCosts," so the formula for the objective function is simply:

$$=SUMPRODUCT(TransShipped, TransCosts)$$

We can identify these parts of the model by referring to Figure 8.25.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E	F	G	H	I	J	
1	Transportation										
2	Where products should be produced and how they should be shipped to customers.										
3											
4	Inputs										
5			EAST	MIDWEST	SOUTH	WEST		CAPACITY			
6		LA	\$5.00	\$3.50	\$4.20	\$2.20		10000			
7		ATLANTA	\$3.20	\$2.60	\$1.80	\$4.80		12000			
8		NEW YORK CITY	\$2.50	\$3.10	\$3.30	\$5.40		14000			
9											
10		DEMAND	9000	6000	6000	13000					
11											
12	Decision Variables and Constraints										
13			EAST	MIDWEST	SOUTH	WEST		Sent		Capacity	
14		LA						0	<=	10000	
15		ATLANTA						0	<=	12000	
16		NEW YORK CITY						0	<=	14000	
17											
18		Received	0	0	0	0					
19			>=	>=	>=	>=					
20		Demand	9000	6000	6000	13000					
21											
22											
23	Objective Function										
24		Total Cost	\$0.00								

Figure 8.25 The spreadsheet preparation for the Transportation problem.

We are now ready to use the Solver (see Figure 8.26). We set the Target Cell and choose Min for our objective function. We then set the Changing Cells (notice that the name of this range appears in the Solver window). We add both the capacity and demand constraints to the constraint list. It is also very important that we specify two Options for this Linear programming problem as well: *Assume Linear Model* and *Assume Non-Negative*.

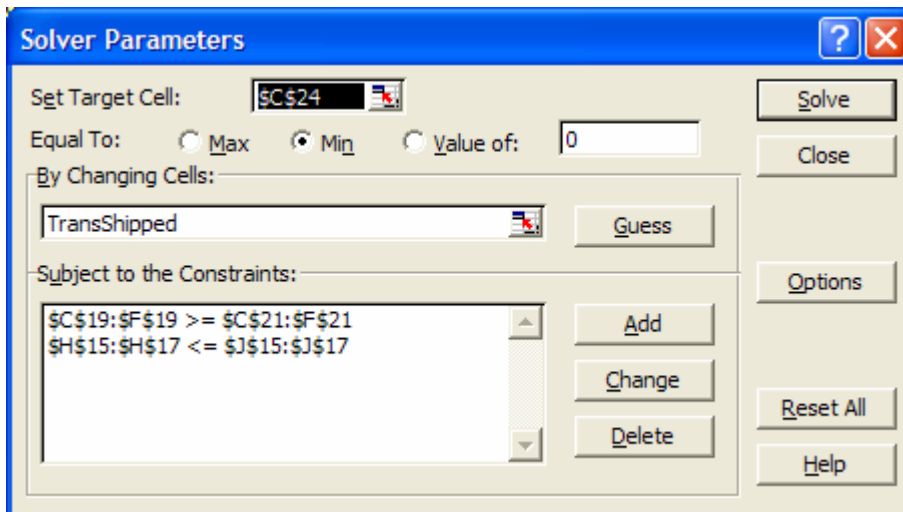


Figure 8.26 Completing the Solver window and specifying the Options.

The Solver solution appears in Figure 8.27. We have found the number of products to be shipped from each plant to each region and the value of the resulting minimal transportation cost. We can also check that all constraints have been met.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E	F	G	H	I	J
1	Transportation									
2	Where products should be produced and how they should be shipped to customers.									
3										
4	Inputs									
5			EAST	MIDWEST	SOUTH	WEST		CAPACITY		
6		LA	\$5.00	\$3.50	\$4.20	\$2.20		10000		
7		ATLANTA	\$3.20	\$2.60	\$1.80	\$4.80		12000		
8		NEW YORK CITY	\$2.50	\$3.10	\$3.30	\$5.40		14000		
9										
10		DEMAND	9000	6000	6000	13000				
11										
12	Decision Variables									
13	and Constraints									
14			EAST	MIDWEST	SOUTH	WEST		Sent		Capacity
15		LA	0.00	0.00	0.00	10000.00		10000.00	<=	10000
16		ATLANTA	0.00	3000.00	6000.00	3000.00		12000.00	<=	12000
17		NEW YORK CITY	9000.00	3000.00	0.00	0.00		12000.00	<=	14000
18										
19		Received	9000.00	6000.00	6000.00	13000.00				
20			>=	>=	>=	>=				
21		Demand	9000	6000	6000	13000				
22										
23	Objective Function									
24		Total Cost	\$86,800.00							

Figure 8.27 The Solver solution to the Transportation problem.

8.4.2 Workforce Scheduling

Another example of a linear programming problem is a Workforce Scheduling problem. A company wants to schedule its employees for every day of the week. Employees work 5 days consecutively, so the company wants to schedule on which day each employee starts working, or, in other words, how many employees start working each day. There is a certain number of employees needed each day of the week. The objective function is to find the schedule that minimizes the total number of employees working for the week.

Shown in the first table of Figure 8.28, the main Input for this problem is the number of workers needed for each day of the week.. We also know that each employee works 5 consecutive days. We have represented this schedule in the second table by recording a sequence of 1's beginning on the day listed in each row. So, the Monday row has a 1 in the Monday, Tuesday, Wednesday, Thursday, and Friday columns. The Tuesday row has a 1 in the Tuesday, Wednesday, Thursday, Friday, and Saturday columns, and so on. This table of consecutive 1's will be used for the constraint formula.

The decision variables for this problem are the number of employees who will begin working (for 5 consecutive days) on each day of the week. The column next to the second table with empty cells is for the decision variables. We have also named this range "SchedDecVar."

There is only one constraint for this problem, which is to ensure that the total number of employees working on a given day (regardless of which day they started working) is greater than or equal to the number of employees needed on that particular day. To achieve this, we again use the SUMPRODUCT function. We sum the array multiplication of the decision variable column with the column of 1's for each day. Since we have named our decision variable range, this formula is:

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

$=SUMPRODUCT(SchedDecVar, DayColumn9:DayColumn15)$

The *DayColumn* letter value would change from D to J for Monday through Sunday respectively. This formula appears in Figure 8.28.

The objective function is to minimize the total number of employees needed. To determine this value, we simply need to sum the total number of employees starting on each day of the week. The formula we use is:

$=SUM(SchedDecVar)$

D17		fx =SUMPRODUCT(SchedDecVar,D9:D15)								
	A	B	C	D	E	F	G	H	I	J
1	Scheduling									
2										
3	Inputs									
4			Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
5		Number needed	17	13	15	17	9	9	12	
6										
7	Decision Variables and Constraints		Employees work 5 consecutive days							
8		Number starting	Day worker starts	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9			Monday	1	1	1	1	1		
10			Tuesday		1	1	1	1	1	
11			Wednesday			1	1	1	1	1
12			Thursday	1			1	1	1	1
13			Friday	1	1			1	1	1
14			Saturday	1	1	1			1	1
15			Sunday	1	1	1	1			1
16										
17			Number working	0	0	0	0	0	0	0
18				>=	>=	>=	>=	>=	>=	>=
19			Number needed	17	13	15	17	9	9	12
20										
21	Objective Function									
22		Total	0							
23										

Figure 8.28 The spreadsheet preparation for the Workforce Scheduling problem.

We are now ready to use the Solver (see Figure 8.29), so we specify the Target Cell and choose Min for the objective function. We then set the Changing Cells. (Notice that the name of this range appears in the Solver window.) We add the one constraint to the constraint list. It is also very important that we specify two Options for this Linear programming problem: *Assume Linear Model* and *Assume Non-Negative*.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

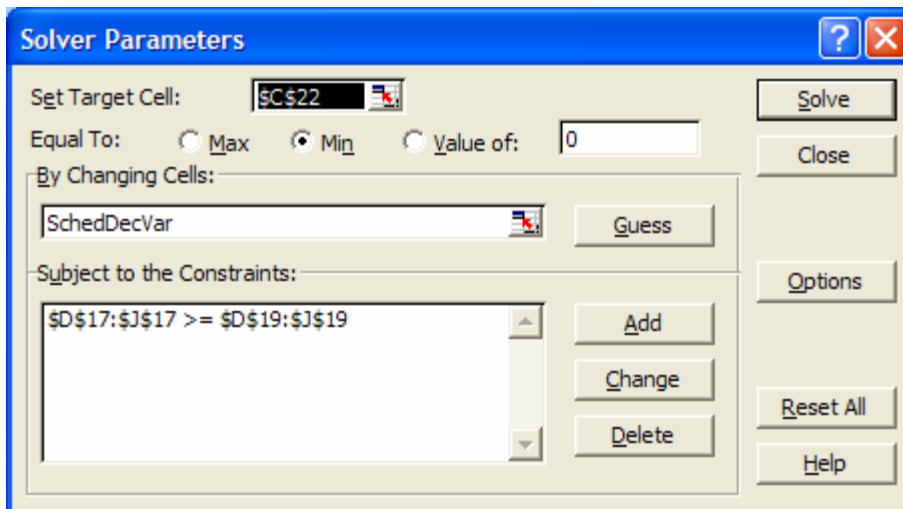


Figure 8.29 Completing the Solver window and specifying the Options.

The Solver's solution, shown in Figure 8.30 reveals the number of employees who will start work on each day of the week. We can also check that the constraint is met. However, we notice that some of the results of the decision variables and objective function are non-integer. Technically, this solution is correct for the way we communicated with the Solver, but it does not make much realistic sense to hire a total of 19.33 employees.

	A	B	C	D	E	F	G	H	I	J
1	Scheduling									
2										
3	Inputs									
4			Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
5		Number needed	17	13	15	17	9	9	12	
6										
7	Decision Variables		Employees work 5 consecutive days							
8	and Constraints	Number starting	Day worker starts	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9		7.00	Monday	1	1	1	1	1		
10		0.33	Tuesday		1	1	1	1	1	
11		2.00	Wednesday			1	1	1	1	1
12		4.33	Thursday	1			1	1	1	1
13		0.00	Friday	1	1			1	1	1
14		2.33	Saturday	1	1	1			1	1
15		3.33	Sunday	1	1	1	1			1
16										
17			Number working	17	13	15	17	13.66667	9	12
18				>=	>=	>=	>=	>=	>=	>=
19			Number needed	17	13	15	17	9	9	12
20										
21	Objective Function									
22		Total		19.33						

Figure 8.30 The Solver solution for the Scheduling problem.

Therefore, we need to enforce integer decision variables, thus making this an integer programming problem. So, we actually need to add a constraint in the Solver window. For this new constraint, we highlight the range of decision variables as the constraint formula and choose *int* from the list of inequalities (see Figure 8.31). The RHS value then automatically becomes “integer.”

Chapter 8: Using The Excel Solver To Solve Mathematical Programs



Figure 8.31 The additional constraint enforces the decision variables to be integers.

Note that an extra constraint has been added to the constraint list in the Solver window (see Figure 8.32). Since we named our decision variable range, this new constraint is displayed in the constraint list simply as:

$$\text{SchedDecVar} = \text{integer}$$

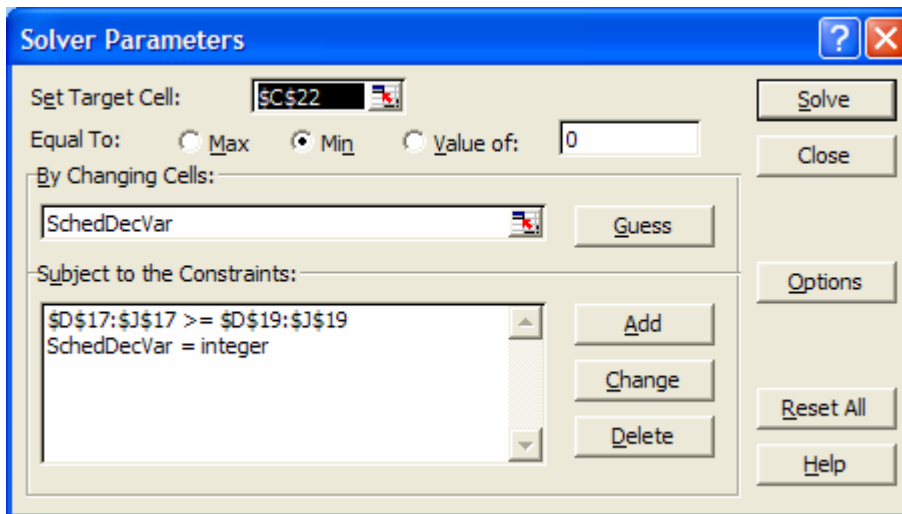


Figure 8.32 The modified Solver window.

The updated solution now has integer values for the decision variables and an objective function that is more realistic (see Figure 8.33).

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E	F	G	H	I	J	
1	Scheduling										
2											
3	Inputs										
4			Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday		
5		Number needed	17	13	15	17	9	9	12		
6											
7	Decision Variables		Employees work 5 consecutive days								
8	and Constraints	Number starting	Day worker starts	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
9		8	Monday	1	1	1	1	1			
10		0	Tuesday		1	1	1	1	1		
11		2	Wednesday			1	1	1	1	1	
12		5	Thursday	1			1	1	1	1	
13		0	Friday	1	1			1	1	1	
14		3	Saturday	1	1	1			1	1	
15		2	Sunday	1	1	1	1			1	
16											
17			Number working	18	13	15	17	15	10	12	
18				>=	>=	>=	>=	>=	>=	>=	
19			Number needed	17	13	15	17	9	9	12	
20											
21	Objective Function										
22		Total	20								
23											

Figure 8.33 The updated Solver solution for the Scheduling problem.

8.4.3 Capital Budgeting

An example of an integer programming problem is the Capital Budgeting problem; it actually has an additional integer constraint because it only allows binary variable values (0 or 1). In this problem, there are 20 projects in which a company, or individual, can invest. Each project's net present value (NPV) and cost per year are provided. The company, or investor, wants to determine how much to invest in each project, given a limited amount of yearly funds available, in order to maximize the total NPV of the investment.

The Input table lists the NPV and yearly costs for each project (see Figure 8.34).

	A	B	C	D	E	F	G	H	I
1	Capital Budgeting								
2									
3	Inputs		NPV	Cost Year 1	Cost Year 2	Cost Year 3	Cost Year 4	Cost Year 5	Cost Year 6
4		Project 1	\$928	\$398	\$180	\$368	\$111	\$108	\$123
5		Project 2	\$908	\$151	\$269	\$248	\$139	\$86	\$83
6		Project 3	\$801	\$129	\$189	\$308	\$56	\$61	\$23
7		Project 4	\$543	\$275	\$218	\$220	\$54	\$70	\$59
8		Project 5	\$944	\$291	\$252	\$228	\$123	\$141	\$70
9		Project 6	\$848	\$80	\$283	\$285	\$119	\$84	\$37
10		Project 7	\$545	\$203	\$220	\$77	\$54	\$44	\$42
11		Project 8	\$808	\$150	\$113	\$143	\$67	\$101	\$43
12		Project 9	\$638	\$282	\$141	\$160	\$37	\$55	\$64
13		Project 10	\$841	\$214	\$254	\$355	\$130	\$72	\$62
14		Project 11	\$664	\$224	\$271	\$130	\$51	\$79	\$58
15		Project 12	\$546	\$225	\$150	\$33	\$35	\$107	\$63
16		Project 13	\$699	\$101	\$218	\$272	\$43	\$90	\$71
17		Project 14	\$599	\$255	\$202	\$70	\$3	\$75	\$83
18		Project 15	\$903	\$228	\$351	\$240	\$60	\$93	\$80
19		Project 16	\$859	\$303	\$173	\$431	\$60	\$90	\$41
20		Project 17	\$748	\$133	\$427	\$220	\$59	\$40	\$39
21		Project 18	\$668	\$197	\$98	\$214	\$95	\$96	\$74
22		Project 19	\$888	\$313	\$278	\$291	\$66	\$75	\$74
23		Project 20	\$655	\$152	\$211	\$134	\$85	\$59	\$70

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Figure 8.34 The Input table for the Capital Budgeting problem.

The decision variables for this problem are the projects that we do and do not invest in. These will have *yes,no* or *go,no go* values. We represent these binary options using 1's and 0's. We have to ensure that the decision variables are only given binary values when we add constraints to the Solver. We name this range "CBDecVar."

There is only one constraint for this problem, which is that no more than the yearly available funds can be spent annually. Since each project has associated yearly costs, we must sum the costs of all of the projects that we have invested in each year to determine if this constraint is met. To do so, we again use the SUMPRODUCT function. The arrays for this function are the decision variables and the column of yearly costs from the input table. Since the decision variable values are binary, only the costs for the projects in which we will invest will be summed. Applying the range name given to the decision variables, the formula is:

$$=SUMPRODUCT(CBDecVar, CostColumn4:CostColumn24)$$

The *CostColumn* value changes for each year from *C* to *I*.

The objective function is to maximize the total NPV. To determine this, we sum the array multiplication of the decision variables and the column of NPV values for each project. We have named this NPV column "CB_NPV." Using this range name and the name of the decision variable range, this formulas is:

$$=SUMPRODUCT(CBDecVar, CB_NPV)$$

See Figure 8.35 for the location and formulation of these model parts.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

C53		=SUMPRODUCT(CBDecVar,CB_NPV)							
	A	B	C	D	E	F	G	H	I
22		Project 19	\$888	\$313	\$278	\$291	\$66	\$75	\$74
23		Project 20	\$655	\$152	\$211	\$134	\$85	\$59	\$70
24									
25	Decision Variables								
26		Project 1							
27		Project 2							
28		Project 3							
29		Project 4							
30		Project 5							
31		Project 6							
32		Project 7							
33		Project 8							
34		Project 9							
35		Project 10							
36		Project 11							
37		Project 12							
38		Project 13							
39		Project 14							
40		Project 15							
41		Project 16							
42		Project 17							
43		Project 18							
44		Project 19							
45		Project 20							
46									
47									
48	Constraints	Used	\$0	\$0	\$0	\$0	\$0	\$0	
49			<=	<=	<=	<=	<=	<=	
50		Available	\$2,500	\$2,800	\$2,900	\$900	\$900	\$900	
51									
52	Objective Function								
53		Total NPV	\$0						
54									

Figure 8.35 Spreadsheet preparation for the Capital Budgeting problem.

Now we are ready to use the Solver. After specifying the Target Cell and Max for the objective function, the Changing Cells, and the one constraint, we must also include the additional binary variable constraint. To do so, we list the decision variables as a constraint formula and choose “bin” from the list of inequalities. The word “binary” automatically appears as the RHS value (see Figure 8.36). Now, when we return to the Solver window (see Figure 8.37), we can see that this additional constraint has been added as:

CBDecVar = binary

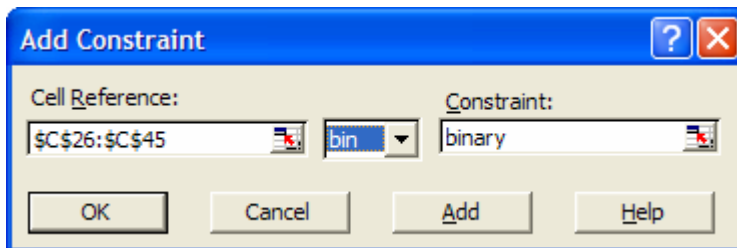


Figure 8.36 Adding an additional constraint to enforce binary decision variable values.

We also still specify the two Options: *Assume Linear Model* and *Assume Non-Negative*.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

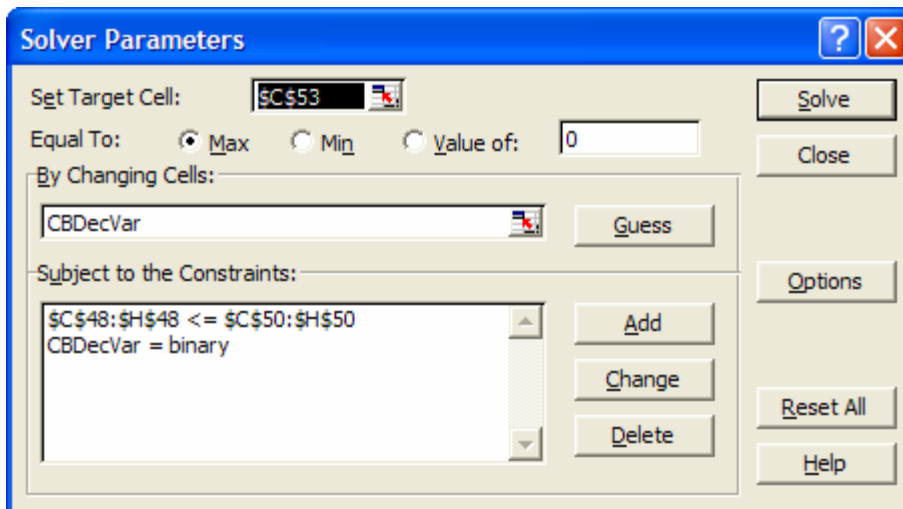


Figure 8.37 The completed Solver window for the Capital Budgeting problem with binary decision variables.

Note that it takes a slightly longer time for the Solver to ascertain a solution, which is provided as 1's and 0's in the column of decision variables (see Figure 8.38). This result can be interpreted as: invest in the projects with 1's; do not invest in the projects with 0's. Therefore, in order to maximize NPV, we should only invest in 12 of the projects.

22		Project 19	\$888	\$313	\$278	\$291	\$66	\$75	\$74
23		Project 20	\$655	\$152	\$211	\$134	\$85	\$59	\$70
24									
25	Decision Variables								
26		Project 1	0						
27		Project 2	1						
28		Project 3	1						
29		Project 4	0						
30		Project 5	0						
31		Project 6	1						
32		Project 7	1						
33		Project 8	1						
34		Project 9	1						
35		Project 10	1						
36		Project 11	0						
37		Project 12	0						
38		Project 13	0						
39		Project 14	1						
40		Project 15	1						
41		Project 16	1						
42		Project 17	0						
43		Project 18	0						
44		Project 19	1						
45		Project 20	1						
46									
47									
48	Constraints	Used	\$2,460	\$2,684	\$2,742	\$876	\$895	\$702	
49			<=	<=	<=	<=	<=	<=	
50		Available	\$2,500	\$2,800	\$2,900	\$900	\$900	\$900	
51									
52	Objective Function								
53		Total NPV	\$9,293						
54									

Figure 8.38 The Solver solution for the Capital Budgeting problem.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

8.4.4 Warehouse Location

The Warehouse Location problem is an example of a nonlinear programming problem. A company stores all of its products in one warehouse. Its customers are in cities around the United States and the management is trying to determine the best location for their warehouse in order to minimize total transportations costs. Each city's location is identified by its latitude and longitude. The number of shipments made to each city is also provided. We are to determine the warehouse location based on its latitude and longitude values.

The Input for this problem is the location of each city identified by its latitude and longitude. We are also provided with the number of shipments made to each city. This input is illustrated in the first table of Figure 8.39. We have named the column of shipments "WHShipments."

The decision variables are the latitude and longitude values of the warehouse location. We have created two empty cells for these and named each one "WHLat" and "WHLong," respectively.

There is only one constraint for this problem, which is that the latitude and longitude for the warehouse location must be between the values of 0 and 120. We only need to add the constraint that they be less than or equal to 120 since non-negativity is a Solver option.

M29		fx			
	A	B	C	D	E
1	Warehouse Location				
2					
3					
4	Inputs		Latitude	Longitude	Shipments
5		Atlanta	33.8	84.4	18
6		Boston	42.3	71.0	6
7		Chicago	41.8	87.7	16
8		Denver	39.8	104.9	15
9		Houston	29.8	95.4	12
10		LA	34.1	118.4	19
11		Miami	25.8	802.0	5
12		New Orleans	30.0	89.9	9
13		New York	40.7	73.9	14
14		Philadelphia	40.0	75.1	7
15		Phoenix	33.5	112.1	8
16		Salt Lake City	40.8	111.9	12
17		San Fransisco	37.8	122.6	16
18		Seattle	41.6	122.4	7
19					
20	Decision Variables		Latitude	Longitude	
21		Warehouse			
22					
23	Constraints				
24		Both Lat and Long must be <= 120 and >= 0			
25					

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Figure 8.39 Spreadsheet preparation for the *Warehouse Location* problem.

We now need to keep track of the distances between each city and the possible warehouse location. These distances are calculated using the following Nonlinear equation:

$$=69*\text{SQRT}((\text{WHLat}-\text{CityLatitude})^2 + (\text{WHLong}-\text{CityLongitude})^2)$$

The *CityLatitude* and *CityLongitude* are calculated from the columns of the input table for each city row. The SQRT function calculates the square root, which is a Nonlinear manipulation of the decision variables. This column of distances appears in Figure 8.40. We have named this column “WHDist.” (Note: The value 69 is based on the earth’s curvature and is only used when computing latitude and longitude distances for US cities.)

C30		=69*SQRT((WHLat-C5)^2+(WHLong-D5)^2)							
	A	B	C	D	E	F	G	H	I
19									
20	Decision Variables		Latitude	Longitude					
21		Warehouse							
22									
23	Constraints								
24		Both Lat and Long must be <= 120 and >= 0							
25									
26	Objective Function								
27		Total Distance is distance from each city to the warehouse multiplied by shipments made.							
28			Distance						
29									
30		Atlanta	6273.2						
31		Boston	5702.5						
32		Chicago	6703.5						
33		Denver	7741.6						
34		Houston	6896.3						
35		LA	8501.7						
36		Miami	55366.6						
37		New Orleans	6539.4						
38		New York	5821.3						
39		Philadelphia	5871.1						
40		Phoenix	8072.9						
41		Salt Lake City	8218.3						
42		San Fransisco	8852.4						
43		Seattle	8920.1						

Figure 8.40 Calculating the distance between each city and the possible warehouse location.

The objective function is to minimize the total distance traveled from the warehouse to each city. It is the sum of the array multiplication between this column of distances and the column of shipments made to each city from the input table. Since we have named both of these ranges, the formula for the objective function is:

$$=\text{SUMPRODUCT}(\text{WHDist}, \text{WHShipments})$$

See Figure 8.41.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

	A	B	C	D	E
25					
26	Objective Function				
27		Total Distance is distance from each city to the v			
28					
29			Distance		
30		Atlanta	6273.2		
31		Boston	5702.5		
32		Chicago	6703.5		
33		Denver	7741.6		
34		Houston	6896.3		
35		LA	8501.7		
36		Miami	55366.6		
37		New Orleans	6539.4		
38		New York	5821.3		
39		Philadelphia	5871.1		
40		Phoenix	8072.9		
41		Salt Lake City	8218.3		
42		San Fransisco	8852.4		
43		Seattle	8920.1		
44					
45		Total Distance	1440364.0		

Figure 8.41 The objective function formula for the Warehouse Location problem.

Let's now use the Premium Solver, or Evolutionary Solver, to solve this nonlinear programming problem. We begin by setting the Target Cell and choosing Min for the objective function. Then, we set the Changing Cells and the Constraints. We next choose *Standard Evolutionary* from the list of Solver versions (see Figure 8.42).

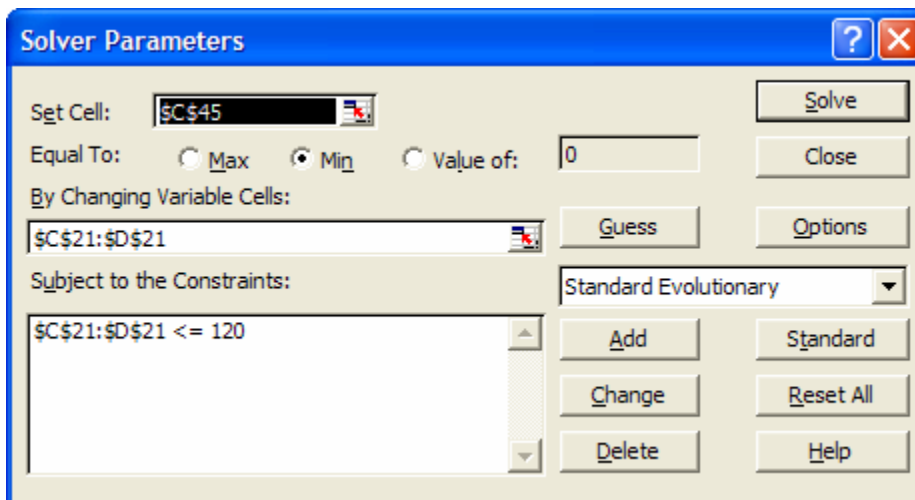


Figure 8.42 The Standard Evolutionary Solver.

We now click on *Options* to view the *Evolutionary Solver Options* shown in Figure 8.43. Here, we want to check *Assume Non-Negative*. (Note that *Assume Linear Model* is not an option here.) We set the *Population Size* and *Mutation Rate* as shown.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

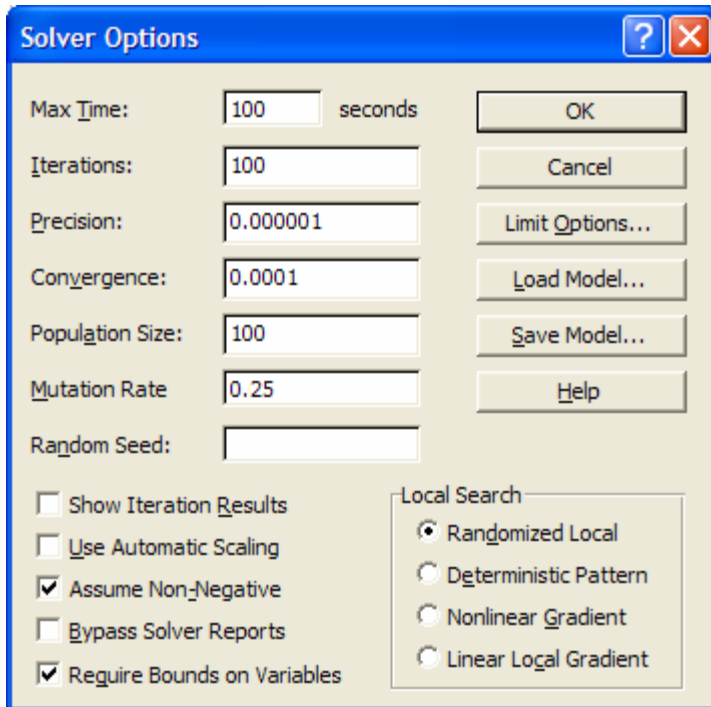


Figure 8.43 The Evolutionary Solver Options.

Now, when we operate the Solver, we find that several iterations of the Genetic Algorithm are being run. After several seconds, the message in Figure 8.44 appears. One of our option parameters has been reached, in this case the *Maximum Iteration*, and we are given the choice to *Continue* or *Stop*. In this case, we stop here, but we could continue for a more accurate solution.

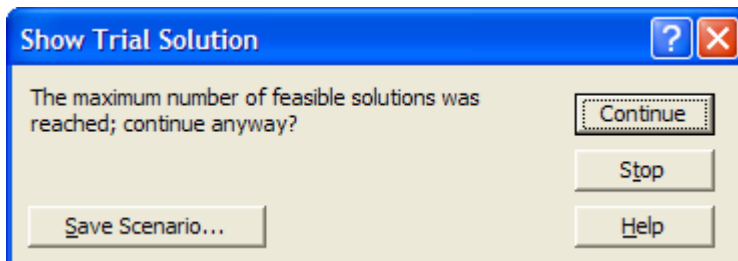


Figure 8.44 The *Show Trial Solution* message offers the choice to stop or continue.

Since we chose to stop, the *Solver Results* window indicates when and why we stopped (see Figure 8.45).

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

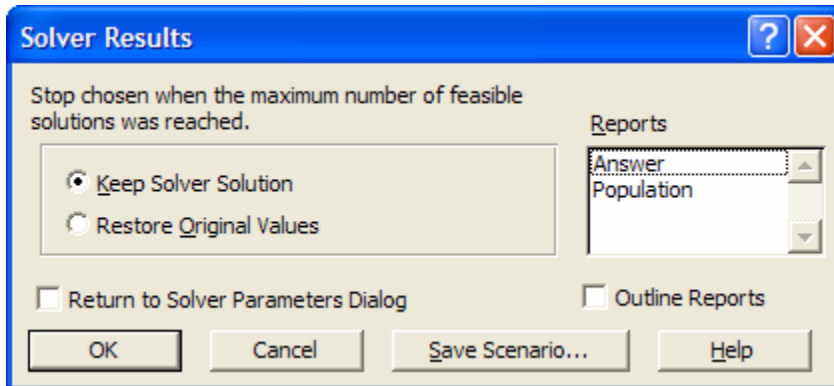


Figure 8.45 The *Solver Results* window explains when and why the Solver stopped.

The solution appears in Figure 8.46. The latitude and longitude for the warehouse location are displayed with the corresponding total minimal distance to be traveled to each city for all shipments.

	A	B	C	D	E
15		Phoenix	33.5	112.1	8
16		Salt Lake City	40.8	111.9	12
17		San Fransisco	37.8	122.6	16
18		Seattle	41.6	122.4	7
19					
20	Decision Variables		Latitude	Longitude	
21		Warehouse	37.45	102.54	
22					
23	Constraints				
24		Both Lat and Long must be ≤ 120 and ≥ 0			
25					
26	Objective Function				
27		Total Distance is distance from each city to the warehouse			
28					
29			Distance		
30		Atlanta	1277.1		
31		Boston	2202.1		
32		Chicago	1067.3		
33		Denver	229.4		
34		Houston	722.4		
35		LA	1118.2		
36		Miami	48269.1		
37		New Orleans	1012.8		
38		New York	1989.1		
39		Philadelphia	1901.8		
40		Phoenix	713.5		
41		Salt Lake City	685.6		
42		San Fransisco	1384.0		
43		Seattle	1399.6		
44					
45		Total Distance	424131.4		
46					

Figure 8.46 The Solver solution for the Warehouse Location problem.

8.5 Limitations and Manipulations of the Solver

Since the beginning of the Excel Solver in 1991, several improvements and modifications have been made to this optimization software. The software developers and users discovered some of the Standard Solver's limitations and errors, which we will now discuss briefly. There are some ways to manipulate the Standard Solver in order to avoid these problems; however, many of these limitations were removed in later versions of the Solver, such as the Premium Solver.

Let's first discuss problem size limitations. We are referring to the summarized table in Figure 8.47 (from www.solver.com). The Standard Solver has a limit of 200 variables and 200 constraints for LP problems, and 200 variables and 100 constraints for NLP problems. We have also noted that it takes a long time for IP problems to be solved with the Standard Solver.

The Premium Solver, on the other hand, has a limit of 1000 variables and 8000 constraints for LP problems, and 400 variables and 200 constraints for NLP problems. Therefore, all large problems should be solved with the Premium Solver. The time for solving IP problems with the Premium Solver is 10 to 20 times faster than the Standard Solver.

Problem Size:	Standard Solver	Premium Solver
Linear Variables x Constraints	200 x 200	1000 x 8000
Nonlinear Variables x Constraints	200 x 100	400 x 200
Speed (approximate):		
Problem Setup	1X	1-50X
Linear Problems	1X	3X
Nonlinear Problems	1X	1X
Mixed-Integer Problems	1X	10-20X

Figure 8.47 A summary of the size and speed comparisons between the Standard Solver and Premium Solver.

The Standard Solver faces other problems in addition to size limitations. One error that can occur involves scaling the related model parts. If there is a mixture of small and large coefficient values in the constraints or the objective function and the possible values that the decision variable can take, rescaling is necessary. For example, if we are solving a binary IP problem whose decision variable values can only be 0 or 1 and whose constraint coefficients are in the hundreds of thousands, the Solver will not be able to recognize the problem as an LP model if we choose *Assume Linear Model* from the Solver Options. To correct this error, we must first re-scale the problem by dividing the constraint coefficients by 1000 and adjusting the costs to maintain proportionality. The Standard Solver is then able to recognize the linearity of the problem and find a solution, which should then be re-scaled again to find a solution proportional to the initial coefficient values.

Another error that can occur with the Standard Solver involves the Sensitivity Report. If the decision variables or Changing Cells have additional bound constraints and the solution value of one of these variables is at its bound, then the reduced cost on the Sensitivity Report becomes equal to the dual values of these variables. The additional

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

bound constraints will not be listed and no shadow price for these constraints will appear either. For example, if we consider the Product Mix problem discussed in the Solver Example section above, the “Demand” Constraints are additional bound constraints on the decision variables. Therefore, if the solution to this problem has one of the decision variable values at its demand bound, its reduced cost will become its dual value in the Sensitivity Report and the “Demand” Constraint will not be listed. This is not correct, but it happens.

The Premium Solver has corrected some of these and other problems of the Standard Solver. There are additionally technical benefits to using the Premium Solver, including the algorithms. The main algorithmic difference between the Standard Solver and the Premium Solver is that the Standard Solver is limited by a slope-based algorithm for solving NLP problems. This algorithm works well for convex maximization or concave minimization problems; however, when neither of these is the structure of the problem, there can be multiple optima that the Standard Solver will not recognize. Such examples include problems with fixed charges or absolute values. Therefore, the Genetic Algorithm of the Premium Solver is much better for solving NLP problems.

Aside from the Excel Solver, there are other optimization software packages available to use with Excel. Frontline Systems, the developers of the Excel Solver, have developed improved software for solving IP problems, called the XPress Solver. They have also developed several large-scale solvers. Lindo Systems has additionally developed some solvers, including What’s Best. What’s Best is a popular alternative to the Excel Solver; however, it requires more optimization knowledge from the user in order to model the problem directly in the spreadsheet. We have chosen to work with the Excel Solver for this book since it is easier to use and learn in both the Excel spreadsheet and in VBA for Excel.

8.6 Summary

- The three parts of a mathematical model are decision variables, objective function, and constraints.
- The three primary types of mathematical models are linear, integer, and nonlinear programming problems.
- Using Solver involves three main steps: reading and interpreting the problem to determine the three parts of the model; preparing the spreadsheet so that Solver can read the data; and running the Solver.
- Several applications of mathematical modeling exist for which Solver can be a useful tool. Some LP examples are Transportation and Workforce Scheduling. An IP example is Capital Budgeting, and an NLP example is the Warehouse Location problem.
- We use the Premium Solver to solve NLP problems since the Genetic Algorithm is more accurate.

8.7 Exercises

8.7.1 Review Questions

1. What are the three components of a mathematical model?
2. How does Integer Programming differ from Linear programming?
3. How do we identify an NLP problem?
4. What are the three main steps involved in using the Solver?
5. How should constraint equations be entered into a spreadsheet when using the Solver?
6. How is the Target Cell used in the Solver?
7. How can you ensure that negative quantities are not produced in a Solver solution?
8. What additional constraint is necessary to change a Linear programming Solver model into an Integer Programming model?
9. What additional constraint is needed to enforce binary decision variables?
10. Give an example of an instance when the Solver model could be applied to solve a problem. State what the objective function, decision variables, and constraints would be for your example.
11. What are some of the main differences between the Standard Solver and the Premium Solver?
12. What are the three Solver versions offered in the Premium Solver?
13. What are some of the main differences in the Options offered, when comparing the standard with the premium Solver?
14. What algorithm solves NLP problems with the Standard Solver? For what type of problems will this cause inaccuracy?
15. What are the two main parameters for the Genetic Algorithm that are set using the Solver Options?

8.7.2 Hands-On Exercises

1. Use the Excel Solver to determine the solution to the following LP model:

$$\begin{aligned} \text{Maximize } Q &= 3X + 4Y - 5Z \\ \text{Subject to: } 5X + Z &\leq 150 \\ 2X + 4Y &\leq 100 \\ 10Z - 2X - 3Y &\geq 20 \\ X, Y, Z &\geq 0 \end{aligned}$$

2. A distribution center for a department store has four trucks available to deliver products to retail stores. The company accrues shipping costs for all boxes that it ships and losses for all boxes that cannot fit on one of the four trucks and must be shipped later. Use the information below to construct a model formulation that minimizes the total cost by determining the optimal number of boxes of each product to be delivered by each truck. Each truck has a trailer volume of 1000 ft³ and a weight limit of 50,000 lbs.

Product Box	Boxes	Size	Weight	Cost of shipping	Loss if shipped late
Clothing Box	400	5 ft ³	75 lbs.	\$55	\$500

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

Jewelry Box	400	2 ft ³	20 lbs.	\$70	\$300
Perfume Box	400	2 ft ³	20 lbs.	\$65	\$275
Shoes Box	400	6 ft ³	45 lbs.	\$50	\$250

- Referring to the model formulated in the previous problem, use the Solver to find the optimal number of boxes of each product to ship in each truck. Adjust the values for amount, size, weight, cost of shipping, and loss if shipped late, and use the Solver again to find the optimal solution.
- A toy company is expanding its toy vehicle product line. The company formerly produced only toy trains but now is expanding the line to include toy cars, trucks, and airplanes. The amount of each type of vehicle to produce must now be determined. The table below displays the expected production cost, sales price, required machine hours, and required labor hours to produce a single unit of each type of toy vehicle. It costs \$200 an hour to run the machine that produces cars, trucks, and trains and \$250 an hour to run the machine that produces airplanes. All toy assembly workers are paid a wage of \$7.25 an hour. Based on historical data, the product line manager forecasts that the demand for trains, cars and trucks will be at least 500 units, while the demand for airplanes will be at least 250 units. The production cost of all toy vehicles cannot exceed \$10,000, and no more than 1,000 labor hours can be spent on production. Formulate this problem as an integer programming model that will maximize the profit earned by the company's toy vehicle product line.

	Production Cost	Sales Price	Machine Hours	Labor Hours
car	\$2.15	\$20.99	0.080	0.17
truck	\$3.45	\$22.99	0.065	0.25
train	\$4.65	\$27.95	0.100	0.19
airplane	\$3.50	\$30.00	0.075	0.23

Now use this formulation to create an Excel spreadsheet of your model. Then, use the Solver to find the optimal number of each type of toy vehicle to produce.

- An agricultural supply company is developing a livestock feed mix that will consist of three ingredients: ingredient A, ingredient B, and ingredient C. The table below displays nutrition and cost information per ounce of each of these ingredients. The company wants to create a mix that contains no more than 750 calories per ounce and no more than 10 grams of fat. The desired mix should also meet at least 25% of the Recommended Daily Allowance (RDA) of each of the following nutrients: Vitamin A, Vitamin D, and Protein. The company wants to develop the feed mix as cheaply as possible. Formulate this problem as a linear programming model, and use the solver to find the optimal percentages of each ingredient to include in the feed mix.

<i>Ingredient</i>	Calories	Fat Grams	% RDA Vitamin A	%RDA Vitamin D	%RDA Protein	Cost
A	830	4	0.35	0.5	0.35	\$ 0.75
B	600	6	0.25	0.41	0.21	\$ 0.63
C	750	3	0.04	0.15	0.22	\$ 0.20

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

6. Using the Solver model that you developed in the previous exercise, perform the following:
 - a. Trace the dependents of each decision variable.
 - b. Trace the precedents of the target cell and the constraint cells.
7. A hardware manufacturer uses four workstations to produce nuts and bolts. The following table provides the number of minutes required to create a batch of nuts or bolts at each workstation:

	Workstation			
	1	2	3	4
Nuts	2	8	7	8
Bolts	3	9	7	5

The following table lists the cost of machining a batch of nuts or bolts:

	Cost per Batch			
	1	2	3	4
Nuts	\$ 0.62	\$ 0.61	\$ 0.66	\$ 0.61
Bolts	\$ 0.82	\$ 0.80	\$ 0.79	\$ 0.77

- The machines at each workstation run for 16 hours a day, 5 days a week. A minimum of 700 batches of nuts and 1000 batches of bolts must be produced each week. Use the Solver to determine the optimal number of batches of nuts and bolts to produce at each workstation in order to minimize the cost of machining.
8. Suppose you have \$0.97 worth of coins in your pocket. You know that you have three times as many nickels as there are dimes. You also know that you have at least five pennies and no more than two quarters. Use the solver to determine the optimal number of batches of nuts and bolts to be produced per week at each workstation in order to minimize production costs.
 9. A venture capitalist is trying to determine which of three projects to finance: project A, project B, and/or project C. She plans to finance as many projects as necessary to maximize her total return. She has a total of \$400,000 to invest in the first year and \$200,000 to invest in each subsequent year. The following table displays the implementation cost per year (in thousands of dollars) of each project:

Project	Implementation Cost per Year		
	1	2	3
A	175	85	72
B	210	105	89
C	195	90	73

Projects A, B, C will yield an estimated return of \$550,000, \$750,000, and \$675,000, respectively, at the end of three years. Formulate this problem as a

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

binary programming model, and use the solver to find the optimal combination of projects for the capitalist to finance.

10. An engineering student is trying to determine how many hours of studying to devote to each of his subjects in order to maximize his overall grade point average this semester. To do so, he predicts the grade average he will receive for studying different amounts of time in each of his classes. The table below displays his predictions. He wants to study no more than a total of 40 hours per week. He estimates that the amount of time he should study physics is double the amount of time he should study economics, and the amount of time he should study calculus is in between those two values. He also estimates that he will devote equal amounts of time to calculus and chemistry. Formulate this problem as a linear programming model, and use the solver to find the optimal solution.

Hours	Calculus	Chemistry	Physics	Economics
0-5	75	76	65	85
6-10	84	87	81	92
11-15	93	94	91	97

11. During each 4-hour period, a small town's police force requires the following number of on-duty police officers: 8 from midnight to 4 AM, 7 from 4 AM to 8 AM, 6 from 8 AM to noon, 6 from noon to 4 PM, 5 from 4 PM to 8 PM, and 4 from 8 PM to midnight. Each police officer works two consecutive 4-hour shifts. Formulate and solve an LP that can be used to minimize the number of police officers needed to meet the daily requirements.
12. A retailer store accepts orders made by telephone 7 days a week, from 8 a.m. to 5 p.m. The management has estimated that the following number of people is needed daily in the call center to cover incoming orders:

Mon	Tue	Wed	Thu	Fri	Sat	Sun
18	10	12	13	15	10	15

The employees work 5 consecutive days per week. The salary is \$100/day to work on Monday through Friday and \$150/day to work on the weekend. Formulate the problem as an integer programming problem that minimizes the cost of staffing the call center. Use Excel solver to optimize this problem.

13. Read hands-on exercise 12. Suppose part-time staff working 3 consecutive days during Monday to Friday can be hired at a cost of \$110/day. The higher cost reflects the higher training and turnover costs associated with part-time employees. The number of such staff cannot exceed 5. Extend the integer programming model to incorporate this option. Use Excel solver to optimize the problem.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

14. A production company blends silicon and nitrogen to produce two types of fertilizers. Fertilizer 1 contains 40% nitrogen and 60% silicon. Fertilizer 2 contains 30% nitrogen and 70% silicon. The selling price for fertilizer 1 is \$70/lb and for fertilizer 2 is \$40/lb. The company can purchase up to 80lbs of nitrogen at \$15/lb and up to 100lbs of silicon at \$10/lb. The company should produce at least 80lbs of fertilizer 1 and at least 30lbs of fertilizer 2.

Identify the amount of fertilizer 1 and 2 that maximize the profits. Formulate this problem as a linear programming problem and use Excel solver to find the solution

15. A local bakery sells blueberry and chocolate muffins in packs of four. In a week the bakery bakes at most 65 packs of muffins. The cost and demands per pack are presented in the table below. It costs \$.50 to hold a pack of blueberry muffins and \$.40 to hold a pack of chocolate muffins in inventory for a week. Formulate and solve an LP to minimize total cost of meeting next three weeks' demands.

Item	Demand Week 1	Cost/Cake Week 1	Demand Week 2	Cost/Cake Week 2	Demand Week 3	Cost/Cake Week 3
Blueberry	40	\$3.00	30	\$3.40	20	\$3.80
Chocolate	20	\$2.50	30	\$2.80	10	\$3.40

16. A company supplies goods to three customers, each of whom require 30 units. The company has two warehouses. Warehouse 1 has 40 units available and warehouse 2 has 30 units available. The costs of shipping 1 unit from the warehouse to a customer are shown below. There is a penalty for each unmet customer unit of demand. With customer 1, a penalty cost of \$90 is incurred, with customer 2, \$80, and with customer 3, \$110. Formulate and solve a transportation problem to minimize the sum of shortage and shipping costs.

From	To Customer 1	To Customer 2	To Customer 3
Warehouse 1	\$15	\$35	\$25
Warehouse 2	\$10	\$50	\$40

17. Referring to the above problem, suppose that extra units could be purchased and shipped to either warehouse for a total cost of \$100 per unit and that all customer demand must be met. Formulate and solve this transportation problem to minimize the sum of purchasing and shipping costs.
18. Currently, the exchange rate is 100 yen per dollar. In Japan, you sell a product that costs \$5 to produce at a price of 700 yen. The product has an elasticity of 3. With exchange rates varying from 70 to 130 yen per dollar, determine the optimal product price in Japan and the profit in US dollars. Assume a Linear demand curve. The current demand is 100.

A currency trader faces the following 1-day currency exchange problem that involves U.S. dollars, English pounds and Japanese yen. In the beginning of the day, he has an inventory of 40,000 dollars, 90,000 pounds and 100,000 yen. By the end of the day, the he must have an inventory of at least 50,000 dollars,

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

95,000 pounds and 90,000 yen. The exchange rates are given below (for example, one can exchange 1 U.S. dollars for 0.61 English pounds).

	Dollars	Pounds	Yen
Dollars	1	0.61	1.11
Pounds	1.6	1	1.82
Yen	0.88	0.55	1

During the day, the currency trader exchanges the starting inventory for different currencies to create the required ending inventory while maximizing the surplus inventory of U.S. dollars. Formulate this problem as a linear program and use the Excel solver to find the solution.

19. A computer company must purchase 700 customized hard drivers for the new model that is planning to launch next year. The following table presents the quotes received from three different vendors:

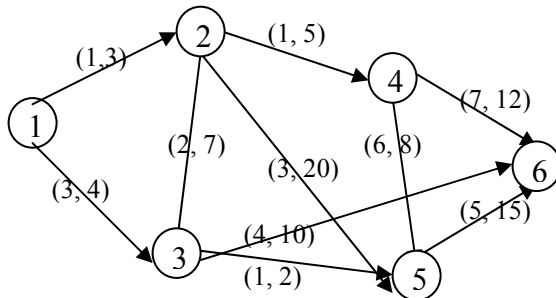
Company	Fixed cost	Unit Cost	Max. Available
A	16,000	250	300
B	0	370	450
C	9,000	270	550

For example, Company C will require a fixed cost of \$9,000 to set-up the machines to produce the drivers. In addition, it will charge \$270 per unit sold.

Formulate the vendor selection problem as an integer programming model. Use Excel solver to solve the problem.

20. Read hands-on exercise 8.19. Reformulate the problem and re-optimize using Excel solver under the additional constraint that no vendor is allowed to supply more than 65% of the total number of units required.

21. A manufacturing company uses trucks to ship products from the production plant to the warehouse. The following network represents the available routes between the plant and the warehouse. The numbers in brackets (d, t) present the length of the route d and the time t it takes to cross the road segment.



Find the shortest path (in terms of distance) from the plant (node 1) to the warehouse (node 6). Formulate the problem as an integer program and use Excel solver to find the solution.

Chapter 8: Using The Excel Solver To Solve Mathematical Programs

22. Read hands-on exercise 8.22. Find the longest path (in terms of time) that connects the plant (node 1) with the warehouse (node 6). Formulate the problem as an integer programming problem and use Excel solver to find the solution.
23. The Markowitz problem provides the foundations for single-period investment theory. The problem is stated as follow: "Given that an investor has n assets. The corresponding mean rates of return are: $\bar{r}_1, \bar{r}_2, \dots, \bar{r}_n$ and the co-variances are σ_{ij} for $i, j = 1, \dots, n$. The problem is to find a minimum-variance portfolio for a given fixed mean value returns (\bar{r})." A portfolio is defined by a set of weights w_i , $i = 1, \dots, n$, that sum to 1 (Luenberger 1998). The following is a NLP formulation of the problem:

$$\text{Min} : \frac{1}{2} \sum_{i,j=1}^n w_i w_j \sigma_{ij}$$

Subject to :

$$\sum_{i=1}^n w_i \bar{r}_i = \bar{r}$$

$$\sum_{i=1}^n w_i = 1$$

Given the covariance matrix and the rates of return for three assets, find the minimum variance portfolio that gives an expected return equal to 0.5.

$$V = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$