Pace University
DigitalCommons@Pace

Faculty Working Papers | Lubin School of Business

11-1-1999

The Mystery of Linear Programming Explained: Second Edition

Jack Yurkiewicz
Pace University

Follow this and additional works at: http://digitalcommons.pace.edu/lubinfaculty_workingpapers

Part of the Management Sciences and Quantitative Methods Commons

Recommended Citation
Yurkiewicz, Jack, "The Mystery of Linear Programming Explained: Second Edition" (1999). *Faculty Working Papers.* Paper 21.
http://digitalcommons.pace.edu/lubinfaculty_workingpapers/21

This Article is brought to you for free and open access by the Lubin School of Business at DigitalCommons@Pace. It has been accepted for inclusion in Faculty Working Papers by an authorized administrator of DigitalCommons@Pace. For more information, please contact rracelis@pace.edu.

WORKING PAPERS

No. 191 November 1999

The Mystery of Linear Programming Explained: Second Edition

by

Jack Yurkewicz, Ph.D.
Professor of Management Science and
Director of the Advanced Professional
Certificate Program
Lubin School of Business
Pace University

THE MYSTERY OF LINEAR PROGRAMMING EXPLAINED- SECOND EDITION

As Recounted by John H. Watson, M.D.
Edited by Jack Yurkiewicz, Ph.D.

Jack Yurkiewicz is Professor of Management Science and Director of the Advanced Professional Certificate Program at the Lubin School of Business, Pace University.

INTRODUCTION

I hadn't seen my old friend and companion, Sir Sherlock Holmes (he had been knighted by the Queen and insisted that this appellation be used) in several months. Feeling that I had slighted him, I rang him up and told him I would be stopping by the next day to see how his new business venture was progressing. Holmes was truly glad to hear my voice and urged that we have lunch together. While he hinted that his firm, of which he was the sole employee, was doing well, he did indeed wish to discuss a business proposition with me. Needless to say, I was intrigued.

I won't bother to bore you, dear reader, with some of our mundane conversation the next day save to say that I found Holmes to be moodier than ever. The great detective had long ago become bored with crime solving, claiming there was little challenge in solving many of the violent crimes that seemed so prevalent in our society. The firm that he started provided consulting service to corporations, in which he would use his mental abilities to help managers solve their business-related problems. Holmes explained to me that he enjoyed the challenge of his new endeavors but was finding it increasingly difficult to converse with these "business types," as he called them. He just didn't have the patience to explain his findings to them, for he found them curiously dull. Holmes went on to say that he had no doubt that this fact was the cause of the country's lack of success in the international business environment. I didn't want to say it, but I thought to myself that the fault might not be solely with the business managers he was dealing with, but partially with him, for I myself frequently found him short-tempered and impatient, even during his ordinary conversations with me. In any case, his proposition to me was simple.

Holmes: Listen, Watson, I just can't bear the thought of explaining to these so-called "leaders" how they should be running their firms. They come to me with some specific problem, usually ill-defined, and then expect me to come up with a solution. I have found that getting the solution is fairly trivial, but I have the dickens of a time trying to explain it to them! You have a solid head on your shoulders, old fellow. Suppose you became my partner. Your only job would be to explain to them the solution that I propose, for I fear that my patience is not as long as yours is.

Watson: The thought intrigues me, for I have been cutting back on my practice lately and do find myself available with more free time. What procedures do you use to get your results?

Holmes: Mostly a branch of mathematics called "operations research," which incidentally has its roots in this country during the Second World War. I collect the data, enter it on a computer I have in my study, and in almost an instant, I get the results.

Watson: Well, I am afraid that tears it, Holmes. I don't know the first thing about operations research and thus could not possibly explain it to others. I did try to learn the subject several years back. I bought a college textbook on the subject and spent several frustrating days trying to learn linear programming. The subject seemed understandable enough, but I became hopelessly

lost on the mechanics of the simplex algorithm. I was working with equations and performing what the author called "elementary row operations." Believe me, there was nothing elementary about it. After three days I threw the book down in disgust and came to the conclusion that only masochists or the temporarily insane would ever try to fathom such a subject.

Holmes: I fear you overestimate the subject's complexity, but the point is irrelevant. No one really does these things by hand anymore, and so it was just your poor fortune to buy a book that was obsolete. All is now done with the aid of the computer. In fact, I am using a program called Solver, which is a part of Excel which is unusually easy to use and powerful. I will explain the computer output to you, along with its ramifications, and then you in turn will explain it to the manager in question. It really is quite simple, and if I may say, also lucrative.

The more Holmes spoke, the more interested I became. His power of persuasion was truly formidable, and in the end, against much misgiving on my part, I agreed to enter into this venture on a trial basis. Holmes was satisfied with my decision and proposed that we start immediately with a problem that he had completed the other day that he was due to report on the following week.

HOLMES DESCRIBES THE CASE

Holmes: The Maximus Computer Company (MCC) has four basic computers it sells to students and small business people. The first, called the **Starter**, is a basic, “no-frills” computer. It has most of the amenities that a new user or a buyer on a strict budget could want, including CD-ROM with sound, an entry-level processor, a small hard disk, a modem for Internet access, and a 15-inch monitor. The second model, called the **Midrange**, is for more demanding users. This model offers a faster processor, larger hard disk, more RAM, a DVD player, and a 17-inch monitor. The third model, the **Super**, provides just about all the computing power a user could want. It offers even more RAM, a very fast processor, a large hard disk, a DVD player with hardware decoder, and a 17-inch monitor. All but the most demanding users would be very happy with the Super. However, for those who want the very “best,” the company offers the **Extreme** which offers a state-of-the-art processor, a huge hard disk, the best multimedia package (the latest generation DVD ROM with a five-piece speaker system), a CD re-writable drive, a 19-inch monitor, etc.

Thus, while the company offers only four models, it feels there is enough flexibility to cover most of the target computer audience. It is a small start-up company and management knows it has to compete against the heavily entrenched products from Dell, IBM, Compaq, Hewlett-Packard, and Gateway. The company philosophy is to ship computers with brand-known components and offer superior service, all at a cost to consumers that is lower than the competition's.

Watson: Well, Holmes, I believe I know about the philosophy of the company and its immediate goals. Can you please tell me more about the four models of computers we offer?

Holmes: As I said, Maximus makes four models with different levels of features, all with the same customer support. They believe that whatever the sophistication of the machine, customer support should be "top notch" and not deteriorate with the price. In the long run, that will generate customer good will and promote their growth in the market.

Watson: I agree whole-heartedly. Tell me a bit more about the computers themselves.

HOW THE OPERATION WORKS

Holmes: As you probably know, the computer industry is currently very competitive, and profit margins are low for each model. The net profit on a Starter is $50, for a Midrange it is $120, for a Super it is $250, and for an Extreme it is $300. These figures already take into account material, labor, depreciation, taxes, shipping, etc. In other words, these are the net profits to the company for each computer sold.

Watson: Eventually we can delve into how these values are determined and perhaps how we can increase them, but for now, let's assume that they are sacrosanct. What else can you tell me about the operation?

Holmes: Management has, what they call "three operations" that make a computer. They call the first operation **manufacture**. This includes taking the customer's phone call and determining which computer and options he or she wants and getting that information to the management of the production staff. They in turn will get the necessary components and make them available to the workers on the assembly line. The second operation is referred to as **assembly**, where the workers on the assembly line actually put together the computer, according the specifications of the customer's order. These are skilled workers who take pride in their work, and even though they work on an assembly line, they do not have an "assembly line mentality," for they believe they are making custom products for specific customers. The last operation is called **inspection.** Here we install the software, run various diagnostic tests, and generally check out and pack up the computer before we send it to the customer.

Watson: I see. Do you have approximate time figures on how long it takes to do these operations?

Holmes: It is a function of the computer. However, they have been doing this for some time now and so the time values do not vary all that much from machine to machine. Thus, for instance, for a Starter, we can assume 0.1 hours for manufacture, 0.2 hours for assembly, and 0.1 hours for inspection. Our units are always in hours,I hope that doesn't confuse you.

Watson: Not at all. Actually, I'm used to minutes and so I am making mental transformations, but since the "company units" are always measured in hours, I will use those also. Tell me the corresponding values for the other computers.

Holmes: For a Midrange, we use 0.2 hours to manufacture, 0.5 hours to assemble, and 0.2 hours for inspection. Each Super requires 0.7 hours to manufacture, 0.25 hours to assemble, and 0.3 hours inspection and testing. Finally, the Extreme gets 0.8 hours to manufacture, 0.2 hours to assemble, and 0.5 hours for inspection. As I said, these numbers really don't vary all that much from machine to machine, but as you can see, they do differ from model to model.

Watson: Very well. What about your resources? How many people, or should I say, how many people-hours are available to do those three operations?

Holmes: I agree with you that we should talk in terms of people-hours. I have been using the term "man-hours" for many years and if I lapse into that gender-specific term, please forgive me. To avoid offending you, I will just use the word "hours" from now on, but I hope you know I mean "people-hours" when I say it.

Watson: Have no fear on my account, Holmes. My sensibilities will not be hurt if you use the older term "man-hours." What numbers do you have?

Holmes: On a daily basis, management informed me that the company has 250 hours available for manufacture, 350 hours available for assembly, and 150 hours to do the inspection and testing.

Watson: I presume that with all this information, we can proceed to model the problem as a linear program.

Holmes: Indeed, Watson. As I intimated earlier, we can solve linear programs with Excel. Excel comes with an add-on package called Solver that is easy to use and yet powerful enough to solve most mathematical programming problems. All we have to do is make a spreadsheet model of the problem and Solver will do the rest.

Watson: Well, that certainly is good news. I use Excel and I have made many models. But how do you make a linear program model in Excel?

MODELING THE PROBLEM IN EXCEL

Holmes: Have a look at the Excel spreadsheet model here on my computer, Watson. It is imperative that you first learn how to model a problem in a spreadsheet. Once you master this skill, we can then proceed to discuss how Solver can be used to get the answer for us. Cells B1, C1, D1, and E1 give the labels of our computers, which in linear programming terminology are called *decision variables*. We want the *values* of these variables to appear in cells B2, C2, D2, and E2. Excel's Solver will call these cells the *changing cells*. We put the per-unit profit of each computer into cells B4 through D4. These numbers are traditionally called the *objective function coefficients*. We must enter a formula into cell I2 (which we labeled as the Profit in cell I1) that will give the net profit for all the computer's made.

	A	B	C	D	E	F	G	H	I
1		**Starter**	**Midrange**	**Super**	**Extreme**				**Profit**
2		0	0	0	0				0
3									
4	**max**	50	120	250	300				
5						**Available**	**Used**	**Slack**	
6	**manufacture**	0.1	0.2	0.7	0.8	250	0	250	
7	**assembly**	0.2	0.5	0.25	0.2	350	0	350	
8	**inspection**	0.1	0.2	0.3	0.5	150	0	150	

Watson: I know how to do that. We would type in cell I2 the formula:

=B2*B4 + C2*C4 + D2*D4 + E2*E4

Algebraically, we are saying, with this formula:

50(Starters) + 120(Midrange) + 250(Super) + 300(Extreme)

Of course, the value in cell I2 is zero because we have zero values for the number of computers made in cells B2 through E2.

Holmes: Well done, Watson! You should know that there is a shortcut to this rather tedious formula. Excel has the built-in function, SUMPRODUCT, which will save us much typing. That is, in cell I2 we can simply type:

=SUMPRODUCT(B2:E2,B4:E4)

That says: multiply the values in cells B2 through E2 by the corresponding values in the cells B4 through E4 respectively, and then add up the results. That gives us the same result as your formula. However, it is easier to input, since we need just type in the **=sumproduct**, highlight the ranges, and type the parentheses. The result is our total profit, which Solver calls the target cell.

Watson: I suspect that we will be using that sumproduct function again. Tell me about the other items I see in your spreadsheet.

Holmes: Our model is incomplete. We need to specify the *constraints*. These account for the technological, economic, or other limitations of the system, and restrict the values of the decision variables to some feasible set. We have three constraints in our problem: manufacturing, assembly, and inspection. Our manufacturing constraint is:

0.1(Starter) + 0.2(Midrange) + 0.7(Super) + 0.8(Extreme) <= 250

Watson: That makes sense. The 250 represents the number of people-hours that are available daily. The expression on the left must represent the total number of manufacturing hours used by the four models of computers. In short, the number of manufacturing hours used by the company is less than or equal to the number of manufacturing hours available from the labor.

Holmes: Exactly.

Watson: Let me see if I can figure out what you entered in the spreadsheet. I see the manufacturing values in the cells B6 through E6. The number of hours of manufacturing available is the 250, in cell F6. I'll wager the formula in cell G6 is

=sumproduct(B2:E2,B6:E6)

Holmes: Watson, I am gratified that you are, to use the vernacular, a "quick learner." That is exactly what I entered.

Watson: I assume we need similar constraints for assembly and inspection.

Holmes: Precisely. Thus, the assembly constraint says

0.2(Starter) + 0.5(Midrange) + 0.25(Super) + 0.2(Extreme) <= 350

and this translates as: the number of hours used for assembly is less than or equal to the number of hours available for assembly, or 350.

Watson: Right, and that is why the formula in cell G7 is:

=sumproduct(B2:E2,B7:E7)

This gives us the number of hours used to assemble the computers. Finally, the last constraint should indicate the number of inspection hours used is less than or equal to 150 inspection hours available.

0.1(Starter) + 0.2(Midrange) + 0.3(Super) + 0.5(Extreme) <= 150

And the formula in cell G8 is:

=sumproduct(B2:E2,B8:E8

that represents the number of inspection hours that are used for the computers.

Holmes: Right again, Watson. Let us look at the spreadsheet again, only this time with the

formulas showing, instead of the values. You'll notice that the formulas in the G column have dollars signs. That is because I copied the formula from cell G6 to cells G7 and G8, and I needed to use the spreadsheet concept of "absolute addressing." This means the decision variable cells will remain fixed when I copied the formula. The only other thing I have added is a column called **Slack**. A *slack variable* for a less-than-or-equal constraint is the difference between the number of units of the resource *available* minus the number of units of the resource actually *used*. It tells us how many units of each resource we have left over in a less than or equal constraint.

	A	B	C	D	E	F	G	H	I
1		**Starter**	**Midrange**	**Super**	**Extreme**				**Profit**
2		0	0	0	0				=SUMPRODUCT(B2:E2,B4:E4)
3									
4	**max**	50	120	250	300				
5						**Avail**	**Used**	**Slack**	
6	**Man**	0.1	0.2	0.7	0.8	250	=SUMPRODUCT($B$2:$E$2,B6:E6)	=F6-G6	
7	**Asm**	0.2	0.5	0.25	0.2	350	=SUMPRODUCT($B$2:$E$2,B7:E7)	=F7-G7	
8	**Insp**	0.1	0.2	0.3	0.5	150	=SUMPRODUCT($B$2:$E$2,B8:E8)	=F8-G8	

Watson: What I would do, at this point, is just try different values for the computer models in cells B2 through E2, all the time monitoring the profit in cell I2. Eventually, I would pick a product mix that would give me the highest profit I obtained. I remember you said that we can get the answer using the technique of linear programming. Please refresh my memory, what does linear programming do?

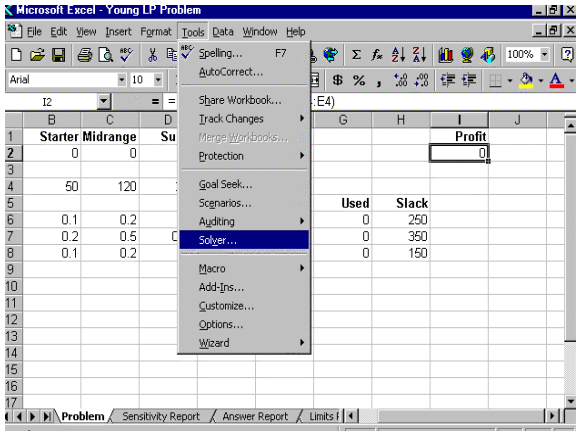
Holmes: It will find the optimal product mix. That is, it will give me the product mix that yields the highest profit and still not violate any constraints. That is called the **optimal solution**.

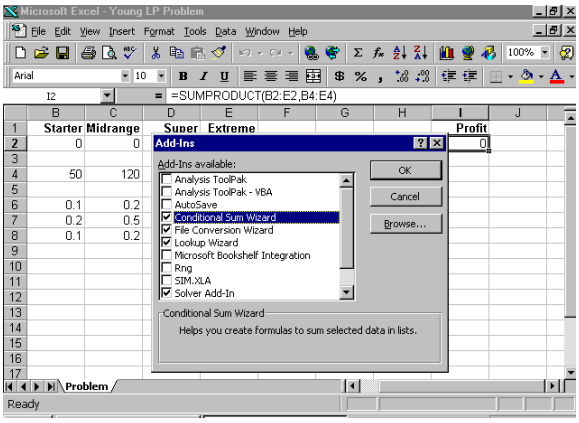
USING SOLVER TO GET THE OPTIMAL SOLUTION

Watson: And you said that Excel had a feature called Solver that can do that? I never even knew about it, and I have been using Excel for years. How does it work?

Holmes: We go to the TOOLS section on Excel's main menu. There you will see Solver. Just

click on it (Figure 1). If you don't see SOLVER listed, then you must make it available as an Excel Add-In. Go to TOOLS and then ADD-INS. You will see a





list of possible Add-Ins. Just make sure the SOLVER box is checked (Figure 2). Then click on OK. You will only have to do this once; the next time you start Excel, Solver will be available automatically in the Tools menu. If you do not see Solver listed in the Add-Ins section, unfortunately that means it was never installed when Microsoft Office was loaded on your hard drive. You must retrieve the original Office CD and do a "reinstall" which will allow you to pick those features that were not installed the first time. Make sure you find and check Solver and it will be installed. You only need to reinstall those missing features.

Figure 2

Watson: Well, luckil[illegible] own notebook computer that Solver is available as an Add-In and so the reinstallation isn't necessary. What do I do now?

Holmes: Let us go through the sequence. You click on TOOLS on the main menu and then SOLVER. The Solver dialog box appears after a few seconds. The first thing you need to do is specify the **objective function**. This is the cell that specifies the problem's goal or objective and it *must* contain a formula. Solver calls this the **Target Cell** (Figure 3).

In our case, it is the cell that contains the profit formula and is cell I2. So we just click on cell I2. If the dialog box covers the cell, then just drag it out of the way and then click on I2.

Watson: The next item is pretty clear. I just choose what kind of problem I have, a maximization one, or a minimization one. I assume the "value of" cell allows me to put in some specified value?

Holmes: Correct, Watson. We will almost never use this for we want the *best* target cell value and we want Solver to get it for us. Thus, we can ignore this feature of Solver.

Watson: The next section is called "changing cells." What does that mean?

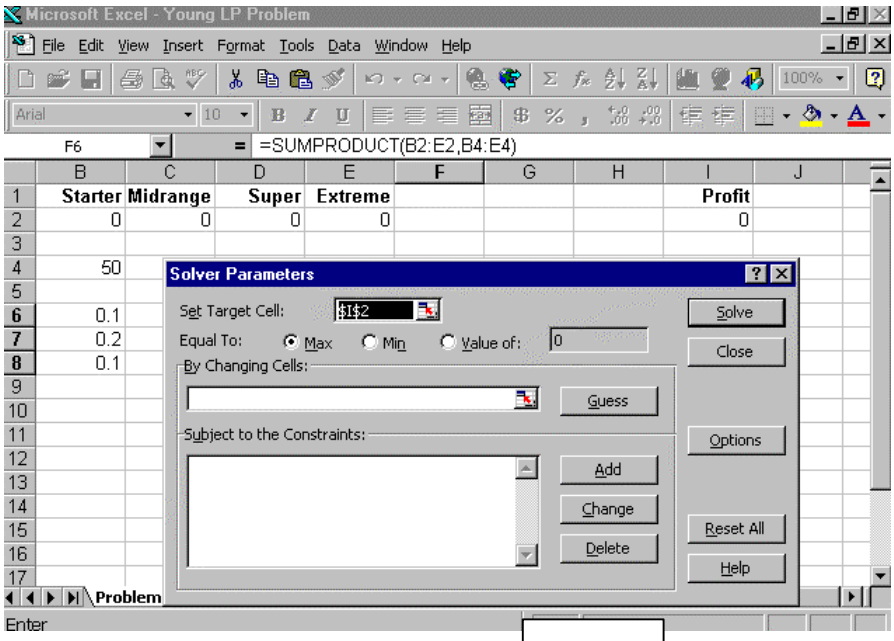


Figure 3

Holmes: These are the cells where the optimal solution will appear. That is, these cells show the number of computers that Maximus should make. In our problem, these are cells in the range B2 through E2. We can just highlight that range or type it in (Figure 4).

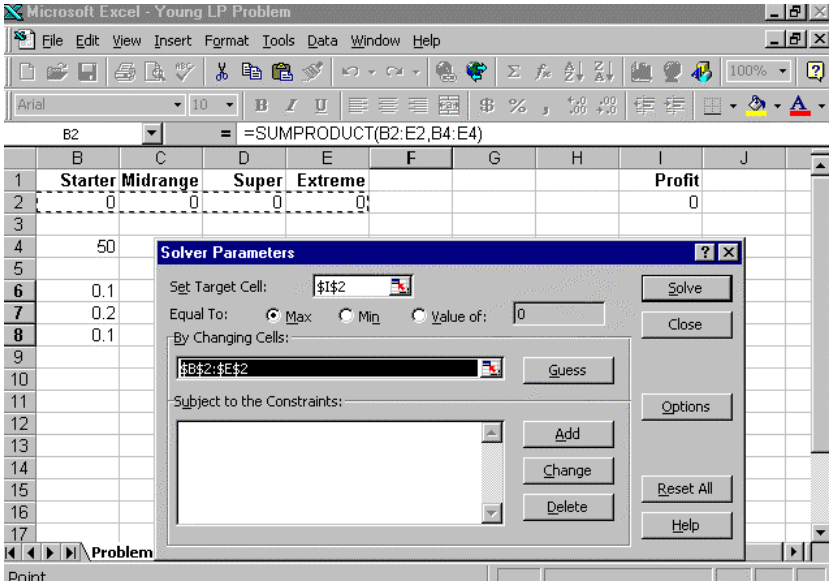


Figure 4

Watson: OK, that seems easy. Now what about the constraints? I see a section that says "Subject to the Constraints." Do I just type them in?

Holmes: Not quite. You click on ADD. After a few seconds another dialog box, called the "Add Constraint" Dialog Box, appears (Figure 5). The left field is the "*cell reference*." That is a cell or range of cells that tells Solver how much of the resource you actually use. It *must* contain a formula.

In our problem, it is the number of hours used for manufacturing, and was the formula in cell G6. We could just click on cell G6. Because the manufacturing constraint was a "less than or equal" constraint, we need to tell Solver that. That is the default inequality sense. Do you see it, the "<=" inequality sense in the second field? The last field is called "*Constraint*" and this refers to the cell or range of cells that tell Solver the amount of resources that are available. These are numbers. In our case, it is the number of manufacturing hours available and is the 250 that appears in cell G6. Thus, we just click on cell G6.

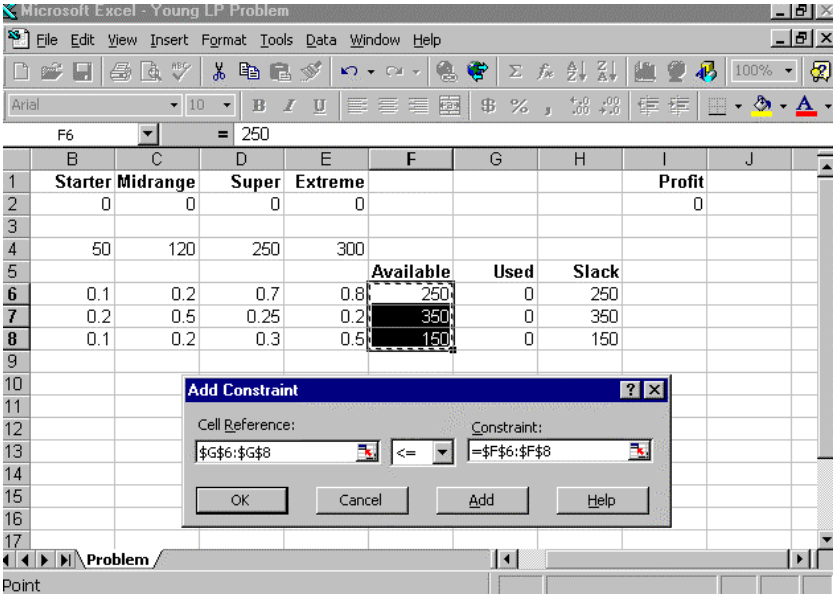


Figure 5

Watson: I see. I guess we have to do the same for the other constraints. This can get quite tedious if we have many constraints in our problem.

Holmes: True, but there is a shortcut. If all your constraints, or a subset of your constraints, are all of the "<=" form, you can add these all at one time. Make sure the cursor is in the Cell Reference field and just highlight the range of cells that indicate the resources used. In our case, we highlight the range G6 through G8. And then we…

Watson: I get it! They are all "<=" so all we have to do is move the cursor to the Constraints field and highlight the range that has the amount of resources that are available. These are cells F6 through F8. I guess we then click on OK.

Holmes: Right again. If you have additional constraints you click on ADD instead of OK. We finally click on "OK" only after you have entered all the constraints of your problem.

Watson: I remember that in linear programming we must also specify that the decision variables must all be nonnegative. That certainly makes sense for our problem, for we certainly cannot make a negative number of computer models. Do I have to enter those constraints in the same way I did the others?

Holmes: Not if you are using Office 97, Excel 8, or any later version. Do you see in the dialog box the button called OPTIONS?

Watson: Yes.

Holmes: Just click on that, and you can see the two boxes that must be checked (Figure 6). The first is "ASSUME LINEAR MODEL" and the second is "ASSUME NON-NEGATIVE." The first one is

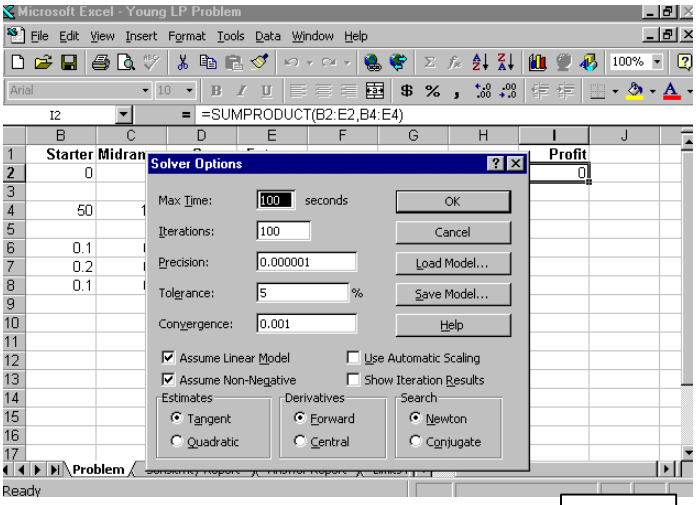


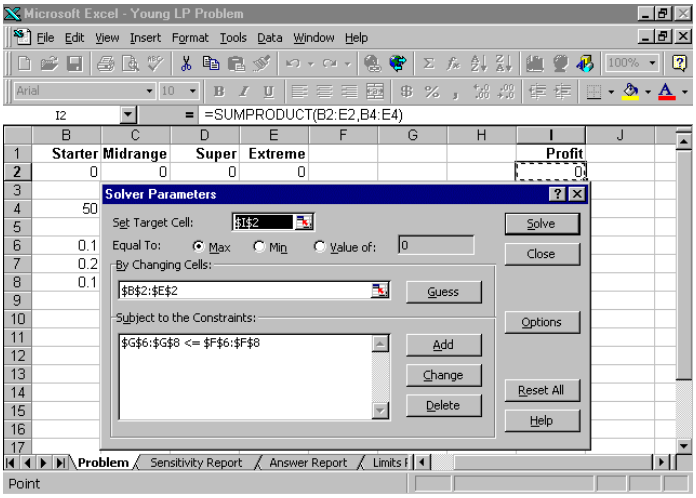
Figure 6

important because it will give us important **sensitivity analysis** that we will discuss later, and the second tells Solver to look for solutions in which the variables are all greater than or equal to zero. By the way,if you are using an older version of Excel, this last option is not available, and you have to enter the non-negativity constraints just as you entered all the other constraints. Then

you click on OK. The completed Solver dialog box is now in front of you and you are ready to go (Figure 7).

Watson: I guess that will do it for entering the model into Solver. I see the "SOLVE" button and so I will just click on it. Let us see what happens.

Figure 7



Holmes: Fine. Do you see that in a few seconds the computer tells us that Solver

found a solution. All constraints and optimality conditions are satisfied." It then asks if you want to "Keep Solver Solution." The obvious answer will be yes, and so that box is checked. But Solver also offers three reports, called the **Answer, Sensitivity**, and **Limits Reports**. Frankly, we just need the first two, the Answer and Sensitivity one. So when we solve linear programs we always specify that we want these two reports. The Limits Report has limited utility for our needs (no pun intended, Watson, so please do not snicker), therefore, we will not ask for it. Thus, highlight the two reports you want and click on OK. Solver adds two new worksheets to our workbook, each giving a separate report. Also, notice that we got the optimal solution to appear in our specified cells, B2 through E2.

The Original Worksheet

	A	B	C	D	E	F	G	H	I
1		**Starter**	**Midrange**	**Super**	**Extreme**				**Profit**
2		0	375	250	0				107500
3									
4	**max**	50	120	250	300				
5						**Available**	**Used**	**Slack**	
6	**manufacture**	0.1	0.2	0.7	0.8	250	250	0	
7	**assembly**	0.2	0.5	0.25	0.2	350	250	100	
8	**inspection**	0.1	0.2	0.3	0.5	150	150	0	

The Sensitivity Report

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$2	Starter	0	-10	50	10	1E+30
$C$2	Midrange	375	0	120	46.67	20
$D$2	Super	250	0	250	170	70.00
$E$2	Extreme	0	-52.50	300	52.50	1E+30

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$G$6	manufacture	250	175	250	100	80
$G$7	assembly	250	0	350	1E+30	100
$G$8	inspection	150	425	150	26.67	42.86

The Answer Report

Target Cell (Max)

Cell	Name	Original Value	Final Value
$I$2	Profit	107500	107500

Adjustable Cells

Cell	Name	Original Value	Final Value
$B$2	Starter	0	0
$C$2	Midrange	0	375
$D$2	Super	0	250
$E$2	Extreme	0	0

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$G$6	manufacture	250	$G$6<=$F$6	Binding	0
$G$7	assembly	250	$G$7<=$F$7	Not Binding	100
$G$8	inspection	150	$G$8<=$F$8	Binding	0

THE SOLUTION IS REVEALED

Watson: Well, I see the optimal solution. Solver tells us to make no Starters, 375 Midranges, 250 Supers, and no Extremes on a daily basis. I must admit that is not what we are currently doing, but Solver says that if we make that daily product mix, our daily profit will be $107,500, which is considerably more than we are currently making. It is interesting that Solver is telling us that we should abandon the "low-end" and "high-end" of the market as compared to offering a "complete product line" for the entire spectrum of customers. We will have to take that up with upper management at our next weekly meeting.

Holmes: I can tell you that I could have predicted something like this. It is a fact of "linear programming life" that the optimal solution would tell us not to make at least one of our computer styles. The reason is that our model had only three constraints, and therefore it is impossible for us to make more than three products. *In other words, the number of positive decision variables can never be more than the number of constraints in a linear programming optimal solution.* Thus, I knew, since we had just three constraints, that at least one of our products, the Starter, Midrange, Super, or Extreme, should not be made. However, I did not know which one. The fact that Solver tells us not to make two of these products is a slight surprise, but quite reasonable.

Watson: Well, as I said, we can take up the solution with upper management. But what does the rest of the output tell us. Is it important?

Holmes: Yes, but before we get into that, I should point out to you that Solver gave us integer optimal values for the number of computer models to make. That will not always happen. In other words, the solution may not be integer-valued. If we want integer solutions, we need to use another technique called **integer programming**, which Solver can also do. However, we will not discuss it now.

Watson: Very well. I assume that I can ask for your help on that when I have need it. Let us get back to all that additional output. Tell me if I need to understand or worry about it.

Holmes: The additional output is what makes linear programming such a rich topic. It is all related to the concept of sensitivity analysis, which tells us how sensitive is our solution to any changes in the data.

HOLMES DESCRIBES SLACK VARIABLES

Watson: Holmes, in the original spreadsheet as well as in the Answer Report, I see a column labeled "Slack" and there are numbers in that column. Since "value" means the values of the decision variables, what does slack value imply?

Holmes: That has to do with the Simplex Algorithm, the procedure that solves linear programs. This procedure requires that the constraints be equalities, rather than inequalities. Hence, all inequality constraints must first be transformed into equality ones with aid of new variables called **slack variables**. Nonnegative slack variables are added to less-than-or-equal-to constraints, and similar variables, which are usually called surplus variables, are subtracted from greater than or equal to constraints in order to achieve the desired equalities.

Watson: I think I understand. Because our three constraints were of the form "less than or equal," we needed three slack variables, for each constraint. Is it correct to say that the first constraint, for manufacture, can be written as:

0.1(Starter) + 0.2(Midrange) + 0.7(Super) + 0.8(Extreme) + slack 1 = 250

Holmes: Splendid, Watson. Since this is the first constraint, the computer adds a new variable, traditionally called slack 1, to the left-hand side of the constraint. The inequality is gone, and the desired equality is achieved. A similar thing is done for the other constraints as well. The program does this automatically. Because these are new variables in the formulation, then their solution must be obtained, in addition to the solution of the original decision variables. The computer gives us the optimal solution to all these variables.

Watson: The computer output informs us that the value of slack variable 1 is zero. The first constraint had to do with how many person-hours of manufacture we are using. How does one then know the final amount of manufacture person-hours that actually are used?

Holmes: The answer is obvious, dear fellow. If we substitute the optimal values of the decision variables into the equality form of the manufacture constraint, we shall see. Let me show you.

0.1(Starter) + 0.2(Midrange) + 0.7(Super) + 0.8(Extreme) + slack 1 = 250

or

0.1(0) + 0.2(375) + 0.7(250) + 0.8(0) + 0 = 250

Subject to our rounding, this is exactly the number of person-hour of manufacture that are available, and hence the number of person-hours of manufacture to be used.

Watson: I see, Holmes. The optimal slack variable 1, corresponding to the manufacture constraint, is zero, which tells us that exactly 250 hours of manufacture are utilized in the optimal solution!

Holmes: Precisely. In technical terms, such a constraint is said to be **tight** or **binding**.

Watson: I'm sorry, Holmes, but I am not sure I understand what you mean.

Holmes: A tight or binding constraint is one in which an optimal slack or surplus variable is zero. That is, such a constraint is satisfied exactly if we substitute the optimal solution in it. There is no slack or surplus.

Watson: I see. That explains the column called "Status" in the Answer Report. Excel is telling us which constraints are satisfied exactly or not, at the optimal solution.

Holmes: Correct, Watson.

Watson: Let me see if I have all this information correct in my mind. Since slack variable 3 is zero, then we are using all 150 hours of the inspection time available. The same reasoning for slack variable 2 tells us that not all hours of assembly were used. In fact, the 100 slack for assembly indicates that we will have 100 assembly hours left over when we implement our optimal solution. Assembly is thus not a tight constraint. Now I understand, only 250 of the 350 assembly hours are actually utilized.

Holmes: Splendid, Watson. I could not have said it better myself.

VARIABLE STATUS—BASIC AND NONBASIC

Watson: Let me return to the solution. Solver tells us that we should produce only Midrange and Super model computers, and not make Starters or Extremes. Since, as you implied, slack variables are new variables that the computer adds to the problem in order to ascertain the optimal solution, then their values are also important and are part of the optimal solution. Thus, we have a slack of 100 hours in constraint 2, the assembly time constraint. I shall not ask you how the computer arrived at these answers for I fear you will tell me that the solution involves much mathematics. However, I do notice that some of our variables are positive and some are zero. Obviously those that are positive are important. Do they have a special name?

Holmes: A well-noted point, Watson. They indeed do. They are frequently called **Basic Variables**. Basic is a technical term that comes from the simplex algorithm. It refers to the fact that, in the absence of a special case called *degeneracy*, a *basic variable is one whose value is positive*. Also, the number of variables that are positive in the optimal solution equals the number of constraints of the problem if degeneracy is not present.

Watson: I fear this is getting too intricate for me. Let us forget this notion of degeneracy for now, since you say it is a special case.

Holmes: Quite right, Watson. As I was saying, in the absence of this special case called degeneracy, a basic variable is simply one whose solution is positive. A listing of these variables, and their values, is sometimes called a **basic feasible solution**.

Watson: I noticed, Holmes, that there are three (if you count slack variable 2) positive or basic variables, and there were three constraints in our problem, ignoring the nonnegativity of the variables. Is that a coincidence?

Holmes: A keen observation. No, it is not a coincidence. I alluded to this issue earlier in our discussion. *The number of positive variables is always equal to the number of constraints in the absence of degeneracy*. By the way, those variables that are not positive, that is, that are equal to zero, are called **nonbasic** variables. Once again, we are ignoring degeneracy and we are not counting the nonnegativity constraints.

Watson: I am afraid that sooner or later I must learn the meaning of that ominous term.

Holmes: Yes, if you are to truly understand all the concepts. But we shall defer its discussion to a later time.

SHADOW PRICES ARE DISCUSSED

Watson: Very well, Holmes. In the interim, I would like to ask you another question. I recall reading an article in *The Times* last week concerning Maximus Computers. The gist of the article was that upper management wanted to achieve a daily profit of $130,000 per day. If we follow the computer's advice and implement the optimal solution, we would achieve a daily profit of $107,500. We are still $22,500 short. I presume that if the company had more resources for manufacture, assembly, and inspection, they could overcome the shortfall?

Holmes: Before I answer your question directly, Watson, I would like to first propose a *hypothetical* example. Now stay with me on this. While it *isn't* the case here, suppose Solver told us that not all 250 available hours of manufacture would be utilized in the optimal product mix. That would imply that there would be some manufacturing time left over; it would be going to waste. Recall we called such a constraint non-binding. If we were able to get possession of some additional hours of manufacture, all we would achieve is to have even more waste, for that many more hours of manufacture would not be used in our optimal product mix. That is, we would be obtaining the same product mix as before, and the same objective profit, but we would have additional hours of manufacture not used. Hence, obtaining more units of a particular not-fully-utilized resource, such as manufacture time, in our hypothetical case, would not help us increase our profit. We are "loosening" or "relaxing" a constraint, or getting more resources for that constraint and achieving nothing.

Watson: That makes sense, Holmes. What you are implying is that if we want to increase our profit, we must always concern ourselves with loosening a constraint that is tight or binding. As I recall, such a constraint is easily recognized because it has zero slack or surplus. In our present problem, all the constraints, except for assembly time, are tight and are thus candidates for relaxing.

Holmes: Precisely, Watson.

Watson: But which ones should I be relaxing? And by how much?

Holmes: Do you notice in the Sensitivity Report that there is a column called **Shadow Price**? In linear programming, each constraint has a number associated with it. That number is called a "shadow price" or "dual variable." A shadow price for a particular constraint refers to how much the objective function changes for a unit increase in the right-hand side value of that constraint. For example, for each additional hour of manufacturing time the firm obtains above the already available 250 hours, its profit will increase by $175. Similarly, for every hour of inspection time the firm gets above the available 150, its profit will increase by $425. The **Constraint RHS** column in the output reminds us how much available resource we have, and the **Shadow Price** column informs us by how much our profit would change if we had more of a particular resource.

Watson: One more time, Holmes. I see that the shadow price for that constraint is 175, or I assume $175. This tells me that…

Holmes: What it tells us is this: for each additional hour of manufacturing the company obtains above the already available 250 daily hours, its profit will increase by $175. The shadow price is sometimes called the constraint's *marginal value*, because it tells us the most the company would be willing to pay, above and beyond what it currently does pay, for an additional unit of the resource. For instance, certainly the firm would not pay $180 beyond what it usually pays for an additional hour of manufacturing time, because its profit increases by only $175 for that extra hour.

RIGHT-HAND SIDE SENSITIVITY ANALYSIS

Watson: I think I get it. If I am correct, we would like to have the company get an additional $22,500 per day. For each hour of extra manufacturing time we hire, the company gets an additional $175 profit. So if the company could hire an additional 22,500/175 or about 128.57 hours of manufacturing time, we would achieve our profit goal.

Holmes: Almost. Your calculations are correct except for one key detail. These shadow prices are valid only over some limited range of the available value. The Sensitivity Report gives that range.

Watson: I see it. The allowable increase for manufacturing is 100 and the allowable decrease is 80. Please be specific. What exactly do these numbers tell us?

Holmes: The shadow prices for the constraints, as given in the Sensitivity Report, will remain constant as long as the range for the available resource stays between those two values. Thus, if the number of daily manufacturing hours is between 170 (250 – 80) and 350 (250 + 100), then the shadow prices are fixed. Since your calculation called for getting an additional 128.57 hours of manufacturing time, which is above the allowable increase of 100 hours for that constraint, we cannot be certain that the shadow prices are as given. Therefore, your calculation is incorrect.

Watson: Now I understand. Let me follow your reasoning. I notice that the shadow price for inspection is $425. If we got an additional $22,500/425 or 52.94 hours, we would get to the profit's goal. But, because the allowable increase for inspection is only 26.67 hours, this calculation is also incorrect, because 52.94 is more than the allowable increase of 26.67. I have answered my own potential question. We can't get the additional $22,500 that way either.

Holmes: Unfortunately, no. However, your analysis is correct.

Watson: Let me try a different approach. Suppose we got some additional hours of manufacture *and* inspection? What if we get enough of *both* resources to achieve the additional $22,500, but still do not go outside the allowable range for either one. That should do it.

Holmes: Almost. Unfortunately, our discussion of sensitivity analysis of an available resource infers that we change just *one* resource amount. You are changing *two.* There is something called the **100% Rule** which tells us how to analyze making changes to more than one data item, but we will save that discussion for later.

HOLMES DESCRIBES THE 100% RULE FOR RESOURCES

Watson: You told me, in our early detective days, that we should not postpone the obvious and to forge ahead to where the evidence leads. Please tell me about the 100% Rule.

Holmes: Very well, Watson. The 100% Rule was developed approximately 30 years after the Simplex Algorithm, and delves into the concept of making more than one change to a resource constraint at a time.

Watson: Is it complicated to follow?

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$2	Starter	0	-10	50	10	1E+30
$C$2	Midrange	375	0	120	46.67	20
$D$2	Super	250	0	250	170	70.00
$E$2	Extreme	0	-52.50	300	52.50	1E+30

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$G$6	manufacture	250	175	250	100	80
$G$7	assembly	250	0	350	1E+30	100
$G$8	inspection	150	425	150	26.67	42.86

Holmes: Not really, Watson. The best way to describe the rule is to work out a specific example. Recall that you correctly deduced that we would not be able to achieve the shortfall of $22,500. It may be that we will never be able to overcome that deficiency, but let us try. Look at the Sensitivity Report once again, Watson. Suppose we somehow manage to increase the number of manufacturing hours to 300, or 50 more hours than we currently have. The maximum number of hours we *could* increase manufacture time, and still be assured that the shadow prices do not change, is 100. Our increase of 50 can be thought as an increase of 50/100 = 0.5 or 50% of the permissible increase.

Let us *also* increase the number of inspection hours from the assumed 150 to 160, an increase of 10 hours. The allowable increase is 26.67, so this increase of 10 represents 10/26.67 = 0.37 or 37% of the permissible increase. The 100% rule states that *if we add the percent changes (increase or decrease) and if these are less than 100%, then the shadow prices do not change.* Because we had a total change of 50% + 37% = 87% < 100%, then all the shadow prices of our Sensitivity Report do not change.

Watson: I think I understand. However, I believe that these changes still do not help us achieve the $22,500 shortfall. My calculations show that, with 300 manufacture hours (an increase of 50) and 160 inspection hours (an increase of 10), by the 100% Rule, the shadow prices are valid, then our profit will increase by just:

175*50 + 425*10 = $13,000

This is well short of the extra $22,500 needed. In fact, I am not sure that, even with the 100% Rule giving us the option of changing more than one resource simultaneously, we can *ever* get the needed $22,500.

Holmes: I am afraid you are correct in the latter analysis Watson. While the 100% Rule may not perhaps help us in this particular problem, it is useful to know for other problems we may encounter in the future.

MORE RIGHT-HAND SIDE SENSITIVITY ANALYSIS

Watson: Very well, Holmes. Let me ignore the 100% Rule and the idea of changing more than one resource at a time. Let me return to the concept of a shadow price for a constraint.

Holmes: Very well. What do you want to discuss?

Watson: The shadow price for assembly is zero. I don't understand that. Can you explain it please?

Holmes: Remember that the shadow price of a constraint tells us how much more we would be willing to pay, above and beyond what we already pay, for an additional unit of that resource. Let me ask you something. Did we use up all of our available 350 hours of assembly?

Watson: No, because if we make the number of computer models that Solver advises, we would only use up 250 daily hours of assembly. In fact, we would have 100 hours left over, and the slack value of 100 tells me that, correct?

Holmes: Indeed. So, if you have 100 such hours not used, why would you want to buy additional assembly hours? Let me rephrase that. What's the most you would pay for an additional hour of assembly?

Watson: I daresay I wouldn't pay anything for it, because I don't need it.

Holmes: Precisely. Assembly time is an example of a **non-binding constraint**. That means that we do not use all the available resources of that constraint at the optimal solution. *The shadow price of a non-binding constraint is zero.* On the other hand, if a constraint is **binding**, or **tight**, then we are using up all the available resources for that constraint. If you look at Solver's Answer Report, it will tell you which constraints are binding or tight, and which constraints are not binding.

Watson: I get it. We would be willing to pay for some more of that resource, because that resource was all used up at the optimal solution. So *the shadow price for a tight constraint would not be zero*.

Holmes: Again, correct. You just stated a key theorem in the subject of linear programming, called the **Complementary Slackness Theorem**. Generally stated, if a constraint is tight, then its slack value is zero, and its shadow price will not be zero. If a constraint is not tight, its slack variable is positive and its shadow price is zero. This is true if the problem is not "degenerate."

Watson: I don't like the tone of that word. What is a degenerate problem?

Holmes: It is not as bad as the term sounds. I will get into its definition later, but I can tell you now that you need not worry about it.

Watson: Well, that's a relief. Let me just go over this concept of shadow price another time to confirm I got it right.

Holmes: Fine.

Watson: Suppose a person comes into my office looking for employment. This person says she is skilled in manufacturing. She is willing to work 10 hours a day, and she wants $50 an hour in salary. I normally pay manufacturing workers $30 an hour (which we don't see explicitly in our model, but it is there implicitly, as part of the net profit coefficients of the objective function), so, she really wants $20 more than I ordinarily pay my workers. Still, I would hire her because, for each hour she works, the firm gets an additional $175, which is what the shadow price tells me. Therefore, the net profit for the company would be 175 minus the $20 "rip-off" of her extra salary, or $155. We are still "ahead" and therefore, I would hire her.

Holmes: I'm not sure I like your choice of words, "rip-off," but your analysis is indeed sound. Notice that you can hire her to work the 10 hours, because the additional 10 daily hours is less than the allowable increase of 100 hours.

Watson: I do understand it, however, another thought just occurred to me. What good is hiring her to do just manufacturing? To build a computer, don't we need manufacturing, assembly, and inspection?

Holmes: A good question, Watson. The computer sees that you have increased just one available resource, manufacturing, for instance, and will come up with a new and different product mix based on the available resources. *When you change the amount of available resource to a different value, even if you are in the allowable range, the optimal solution to the problem will change. All that stays the same are the shadow prices.* We generally don't know what the new optimal solution would be, and would have to re-run the problem with Solver to get the new optimal solution. However, I can tell you that the product mix will be different.

Watson: Very well, let me continue with my example. If I hire this hypothetical person to work an additional 10 hours in manufacturing at $50 per hour, which is $20 more than the $30 per hour I ordinarily pay for manufacturing time, the company would achieve an additional daily profit of $175 - $20 or $155 per hour. Therefore, for the 10 additional hours she works, the company would gain an additional $1550. However, the product mix, zero Starters, 375 Midranges, 250 Supers, and zero Extremes would no longer be the optimal solution. Moreover, if I wanted to know what that new solution is, I would have to re-run the problem in Solver to get it. Is that right?

Holmes: Correct. However, I can tell you something about the new solution, even if we don't run the problem again on Solver. If we change the value of an available resource, and we stay within the allowable range, the shadow prices will remain the same. What also remains the same is the "make-up" of the product mix. Those variables that were *basic* before are still basic, and those variables that were *nonbasic* before are still non-basic. What that means is we would still

make the same products before we made the change of the resource, although in different amounts. Also, those products we did not make prior to the change, we still would not make. In other words, without re-running this problem, if you hire this woman to work the additional 10 manufacturing hours, I can tell you for a fact that we should still make Midrange and Super models, but no Starters and Extremes. However, the number of Midrange and Supers will indeed be different from the original optimal values of 375 and 250 respectively, and if we want those amounts, we would have to run the problem again in Solver.

Watson: That is useful to know. What about the slack for our assembly constraint? Will slack still exist there?

Holmes: Yes. The **optimal solution** means the values of the original variables and any slack variables. Once again, if we make a change in the amount of available resource, and that change is within the allowable range, the optimal solution values will be different. However, those that were positive before will still be positive, and those that were zero before will still be zero. That includes the original variables and the slack variables.

Watson: I see. Can you tell me what the rest of the output is all about?

OBJECTIVE FUNCTION SENSITIVITY ANALYSIS

Holmes: Certainly, Watson. It the called the "**Sensitivity to Objective Function Changes**." Let me propose a hypothetical scenario to you. Consider the Midrange computer at the moment the firm assesses a net profit from it of $120. The computer tells us we should produce 375 of them. Now suppose that for whatever reason the per-unit profit of this model Midrange computer were to increase above the given $120 profit. If that profit became higher and higher, what do you think might happen?

Watson: It seems to me, Holmes, based on your story, that I would like to make more than the 375 Midrange models that Solver says I should make. Since I presume to have the same limited resources of manufacture, assembly and inspection; if I make more Midrange computers, apparently I would make fewer of some the other model type, perhaps the Super. Is this some kind of trick question?

Holmes: Not at all, Watson, for I marvel at your insight. You are correct indeed. If the profit of model Midrange computers increased, sooner or later we would have a profit in which the optimal solution given would no longer be optimal. Let me reverse the story. Now suppose the per-unit profit of the Midrange decreased below the given $120. What do you think would happen then?

Watson: Clearly, if the profit on this model continued to fall, there would be some value at which I would no longer be happy making the 375 Midrange computers called for in the optimal solution, but would prefer making fewer, or perhaps even none at all. And, indeed, if I do make fewer or none, then I would probably make more of some other model to compensate.

Holmes: Correct again, Watson.

Watson: You seem to be implying, Holmes, that there is some range of values of the profit of Midrange over which we would be content making the 375 called for by the Solver.

Holmes: Not only the 375 Midrange's, but the entire product mix will remain the same. You can see the answer in the section involving sensitivity analysis of the variables. There you see, for Midrange, the current profit of $120, an allowable increase of $46.67 (to a maximum allowable profit of $166.67), and an allowable decrease of $20 (to a minimum allowable profit of $100). This is the range over which the profit of Midrange computers can vary.

Watson: Without changing the number of Midrange models we will make?

Holmes: More, Watson. Without changing the optimal solution of any of the variables. That is, the entire product mix as called for will still be the same so long as the change in profit of Midrange's is within a decrease of $20 and an increase of $46.67. Equivalently, the product mix will not change if our profit of model Midrange computer is somewhere between $100 and $166.67.

Watson: How do we know that? I don't understand.

Holmes: It comes from the theory of linear programming and the mathematics behind it.

Watson: Very well. Let me get a bit hypothetical to see if I understood what you said. What if the company discovered that their formulation was in error, and, in fact, that the per-unit profit of the Super computer was really $200 instead of the assumed $250, an error of minus $50, but less than the allowable increase of $70? Then the answer to the problem is still produce no Starters, 375 Midranges, 250 Supers, and no Extremes. But what about the total profit of $107,500? Surely that can't remain the same.

Holmes: Clearly not. Think, Watson. If we making the same number of Super's, but are making less money for each one, then the total profit must be lower. In fact, we can easily find it:

$107,500 - 250($250 - $200) = $107,500 - 250($50) = $95,000

Watson: That seems fairly obvious, Holmes. But I just thought of something. If the profit of an objective function coefficient stays within the allowable interval, then you said all the answers

found by the computer stay the same. Does that mean that the slack or surplus variable values stay the same also? And all variables that are zero, or nonbasic, stay that way too?

Holmes: Precisely.

Watson: That would imply then that the constraints that are tight remain tight and those that are nonbinding remain nonbinding.

Holmes: Brilliant deduction, Watson.

Watson: Holmes, what happens if the profit on, say, Supers was increased by exactly $170, to $420?

Holmes: At that point, Watson, in the absence of degeneracy, all that happens is that we have alternative optima. That is, there is our current optimal solution set and some other set. Both sets of solutions will have the same objective function value:

$107,500 + 250(170) = $150,000.

Watson: There is that ominous word, degeneracy, again.

Holmes: Ignore it for now, Watson. Do you see how really useful the objective function sensitivity analysis is? Once we run the problem on Solver, we can change values on the per-unit profit and if our new values are within the allowable range, we know that we should make the same number of computer models that Solver found before we made the change. And we didn't have to re-run the problem to get this information.

Watson: I agree that this is all useful. Let me take the discussion on a different tact. Is my reasoning for the following correct? At the moment the per-unit profit of Starters is $50. The computer says not to produce any, presumably because this profit is too low. If the profit were even lower than the $50, it seems obvious that we would still not produce those Starter computers. In fact, the profit can go as low as we please below the $50; the optimal product mix would not change. Yet the Solver output shows an allowable decrease of **1.0E+30**. What does that mean?

Holmes: The software is using scientific notation, my good fellow. That number is one followed by 30 zeros, an exceedingly large number to be sure. As your common sense surmised, we can have the profit of a Starter go as low as we want, even to "negative infinity," and the product mix would stay the same. That implies that we would still make no Starter computers.

Watson: Very well. On the other hand, if the per-unit profit of a Starter drill were to increase beyond the given $50, then there must a point at which the profit is high enough where it would pay to produce the Starters. In other words, the current optimal solution, which says produce no Starters, can no longer be optimal. I am guessing that this per-unit profit for the Starters would have to increase by $10, to $60. What do you say, Holmes?

Holmes: Your analysis is indeed correct. At $60 per-unit profit for a Starter (meaning an increase of $10 from the assumed profit of $50) we have **alternative optima**. That means there is another optimal solution to the problem (another product mix, if you will) with the same profit. Both sets of answers are equally good. That is, we would be indifferent between the two solutions, for the profit would be identical in either case. Once the profit change exceeds this value, we have a new optimal solution in which the Starter model is indeed produced, resulting in perhaps fewer amounts made of some other model drill. What exactly does happen we cannot say. In order to find out we must run the problem again with the new profit of Starter in the objective function. Very nicely done, Watson. By the way, if the per-unit profit increase of a Starter were exactly $10, because of the nondegeneracy of our problem, we would have alternative optima. But what is the objective function value, the total profit of our firm?

Watson: Since the alternative optima means another solution with the same objective function value, then the profit would be the same as that of the current optimal solution in which no Starters are produced. Since we are increasing the per-unit profit of these Starters upward from $50 to the allowable increase of $10, to a limit of $60, and we don't produce any in that range, then the total profit would not change. It remains at $107,500.

Holmes: Excellent, Watson. Your analysis is right on the mark.

Watson: So, if the coefficient of an objective function value is within the allowable interval shown, the optimal solution does not change, but the optimal objective function value or total profit may or may not change. I see. However, you mentioned that word “degeneracy” several times. Just what is a degenerate linear program? It certainly sounds ominous.

DEGENERACY IS REVEALED

Holmes: It is no so threatening as you might imagine. Do you remember that I told you that the number of positive variables in an optimal solution can never exceed the number of constraints? Briefly, the optimal solution to a linear program is called ***degenerate*** *if the number of positive variables is less than the number of constraints in the problem.* The problem is not degenerate if the number of positive variables equals the number of constraints. When we count positive variables, we look at the original and slack variables.

Watson: OK. I understand the definition. That says, for example, that our solution is not degenerate because there are three positive variables in the optimal solution (Midrange, Super, and slack for Assembly time) and there are three constraints in the problem. So what does this imply?

Holmes: Actually, for all practical purposes, nothing. If a problem is degenerate, we glean a little less information from the computer output. For example, if the per unit profit of a Starter were $60, then we would have alternative optima. If the problem had been a degenerate one, we would have to rephrase to that to read “we might have alternative optima.” If the problem was

not degenerate (as ours is), we can change that to the more definitive "we have alternative optima." Also, the 100% rule, which we mentioned earlier, does not work precisely as stated if the problem is degenerate. We shall leave these issues to a more advanced analysis, perhaps in the near future.

Watson: In other words, I really shouldn't worry too much about it, but I should at least be aware of the concept, is that it? I must say, I was worried needlessly.

Holmes: That's right.

THE 100% RULE FOR OBJECTIVE FUNCTION COEFFICIENTS

Watson: By the way, Holmes, you mentioned the 100% Rule, which allowed us to change more than one resource value or right-hand side value, at a time. Is there a comparable 100% Rule for changing more than the profit coefficient or objective function coefficient simultaneously?

Holmes: Yes, there is, Watson. When we change an objective function coefficient, and we are still in the allowable range as given by Solver, the optimal solution stays the same. The 100% Rule for objective function coefficients says: if we consider the ratios of the changes made (positive or negative) to the maximum possible change in that direction (positive or negative), then the sum of the ratios must be less than or equal to one (or 100 percent if the ratios are in percents). If these ration are indeed less than or equal to one, than the optimal solution to the problem will not change.

Watson: Indeed, the rule is the same as that for the resources or right-hand side, except that now the optimal solution stays the same, while there the shadow prices stayed the same.

Holmes: Indeed, Watson.

Watson: Let me try a hypothetical example to verify if my understanding is sound. Suppose upper management informs us that the per-unit profit of a Midrange computer dropped from \$120 to \$110, while at the same time the per-unit profit of a Super computer increased from \$250 to \$300. The change in Midrange is \$10, and the allowable decrease is \$20. This represents a ratio of 10/20 or ½ or 50%. The change in Super's profit is 50 and the allowable increase for it is 170. This represents a ratio of 50/170 or 5/17 or 29%. The sum of the ratios is ½ plus 5/17 which is less than one, or better yet, 50% plus 29% or 89% which is less than 100%. That tells me that the optimal solution we had is still optimal. That is, make no Starters, 375 Midrange, 250 Super, and no Extreme computers. The optimal profit can be determined as follows. We still make 375 Midranges, but we are making \$10 less on each, for a drop of 375*10 = \$3750. We still make 250 Supers, but are making \$50 more on each, for a gain of 250*50 = \$12,500. Our net gain is thus 12,500 – 3750 = \$8,750. We add this \$8,750 to the original profit of \$107,500 to get a total profit of \$116,250. Any errors, Holmes?

Holmes: None, Watson. I could not have said it better myself. Excellent analysis.

REDUCED COSTS ARE EXPLAINED AT LAST

Watson: Well, Holmes, I must admit the analysis is not as daunting as I had originally thought. The only item that remains left unexplained is the Reduced Cost column of the output. You said we would get to it in due time. It that time nigh? Perhaps you saved it for last because it is the most complex subject?

Holmes: Yes and no, Watson. Yes the time is nigh and no the subject is not very complex at all.

Watson: Very well, Holmes. Please proceed.

Holmes: If you examine the output you will notice that the reduced cost of a variable is a number other than zero whenever the value of the variable itself is zero, or to put it technically, the reduced cost is not zero when the variable is nonbasic. For our problem, we have nonzero reduced costs for the Starters and Extremes. One interpretation of the reduced cost is this: it tells us how much the per-unit profit of that variable would have to increase before it would pay to produce any. That is, how much the per-unit profit must increase before that nonbasic variable becomes basic. We can see that the current profit of Starters is $50. Not until that per-unit increases by $10 to a total of $60 will we produce Starters. For the Extreme computer, its current profit is $300. Its profit would have to increase by $52.50 to $352.50.

Watson: Really Holmes, all that you have just told me could have been obtained from examining the sensitivity analysis of the objective function coefficients of the output. We see those figures you calculated in the “Allowable Increase” column for the two variables in question.

Holmes: Right again, Watson. However, it is our second interpretation of the reduced costs that is even more meaningful. The reduced cost for a nonbasic decision variable tells us how much the objective function value would change per rate of increase of that variable. Hence in our example if, for whatever reason, the firm insisted on making Starters in its product mix, its objective function total profit would *decrease* by $10 for every such computer made. We may think of it as though it is "costing" the company $10 per Starter. That is, this is how much money we are getting less than our optimal $107,500 we could make by producing the optimal product mix. Remember, our optimal mix says to make no Starters. Similarly, if the company wanted to produce Extreme computers, for every Extreme made, the firm would be decreasing its profit $52.50 from the optimal $107,500.

Watson: So really these reduced costs give us an indication of what the firm's total profit would be if the company produced a non-optimal or sub-optimal product mix.

Holmes: That is one way of stating it, Watson.

Watson: I think I get it. Are you saying that if I insist on producing an Extreme computer line, for each one I make and sell, the company would be losing $52.50?

Holmes: Right. Of course, the company doesn't "lose" money if it makes an Extreme. It would be making $52.50 less than it could if it had not made and sold the Extreme computer.

Watson: I was wondering about that. Often, companies would not want to drop a product line as it might alienate customers or send a negative message to the marketplace. In particular, I'm not so sure upper management would want to eliminate the Extreme computer line, despite what the optimal solution says.

Holmes: Fine. Then they should know that for every Extreme they sell at the current profit of $300, they are making $52.50 less than the optimal daily profit of $107,500. It is quite common in business for a company to implement a sub-optimal solution instead of the optimal solution.

INTRODUCING A NEW DECISION VARIABLE OR PRODUCT

Watson: Speaking of a non-optimal solution, Holmes, I read in *The Times* that the marketing people and the R&D people want to produce and sell a new model computer for game players, called the **Ultimate**. This would be the current state-of-the-art desktop computer. The article said that each Ultimate computer would need 0.9 hours of manufacturing time, 0.4 hours of assembly time, and 0.7 hours of inspection and testing. Could we use what we have learned about linear programming to determine what the minimum per unit profit on such a computer would have to be before it would be beneficial for the company to make any?

Holmes: Yes. Understand that to produce this new computer, we are diverting the valuable resources that we allocated to make the other models. We know what the "loss" of those resources are the shadow prices. Let us calculate, in dollars, how much the resources "eat up" to make just one Ultimate computer.

Manufacturing:	(0.9 hour)*($175/hour) =	$157.50
Assembly:	(0.4 hour)*($0/hour) =	0
Inspection:	(0.7 hour)*($425/hour) =	297.50
		$455.00

The concept is called **pricing out a variable**. We see that to make a single Ultimate computer, we need $455 of resources. We want to find the profit minus the cost of an Ultimate, and we want that result to be positive, otherwise why bother to make an Ultimate? Clearly, for the profit minus the cost to be positive, the profit should be at least $455 for each machine. If the per unit profit of an Ultimate is over that amount, we should pursue the new product line. If the per unit profit is below the $455, we should not introduce the Ultimate line. Needless to say, if the profit on an Ultimate were, say $500, if we ran Solver, we would find that Solver recommends that some Ultimates to be made, and some other computer would drop to zero (not be made) to

compensate. Recall that we have three constraints; therefore, no more than three products can be made in the final product mix.

HOLMES GIVES A DOSE OF THEORY

Watson: "Pricing Out" a variable certainly has its use if we want to introduce a new product to our product line. Does it have any other applications, Holmes?

Holmes: Indeed, Watson, but more from a theoretical point.

Watson: Theory? That conjures up complicated mathematics and is a road I am hesitant to take at this point. Still, it must be important, so pray, Holmes, go slowly down this road and I will attempt to follow.

Holmes: Watson, you underestimate yourself. Still, I will take your plea seriously. If you followed the calculations of pricing out, then you should realize what the plan was. We were trying to figure out if it would pay to produce just one new model computer, the Ultimate. Let's ignore the new model for now. We have a model and Solver gave us a solution: in particular, make no Starters, 375 Midranges, 250 Supers, and no Extremes. Suppose I insisted that we make just one Extreme computer in our product mix. To make this one Extreme we need resources, resources that are used to make the Midrange and Super computers too. In particular, to make just one Extreme, we need 0.8 hours of manufacture, 0.2 hours of assembly, and 0.5 hours of inspection. By taking these resources away from the available resources to make Midrange and Super computers, we can think of the situation as one in which Midrange and Super computers had *fewer* amounts of resources than we originally thought. But, if you recall, we know the value of these resources; they are called the shadow prices. So, to make one Extreme, we need:

Manufacturing:	(0.8 hour)*($175/hour) =	$140.00
Assembly:	(0.2 hour)*($0/hour) =	0
Inspection:	(0.5 hour)*($425/hour) =	212.50
		$352.50

That is, we are using up $352.50 worth of valuable resources (which we used to make the other computers) to make just one Extreme. The profit of an Extreme is $300, and so the profit minus the cost of an Extreme is:

Profit-Cost = $300 - $352.50 = -$52.50

Watson: Wait, Holmes. I see that same number as the reduced cost for Extreme computers. Is that a coincidence?

Holmes: Not at all, Watson. In fact, that is how Solver finds the reduced cost of a variable, by pricing it out. Think of Solver now talking to you, saying, "So you insist on making an Extreme computer. Why bother? If you do, you will lose $52.50 for each one you insist on making. Why would you want to do something like that?" That's how Solver knows that it has found an optimal solution. For a maximization problem, the reduced costs of variables are always zero or negative. A negative reduced cost tells Solver not to make that product. If a reduced cost comes up positive, Solver does another iteration, making that product while another product leaves the product mix. Eventually, Solver comes up with a solution in which the reduced costs are all zero or negative, and that implies an optimal solution has been found.

Watson: I see, that does make sense. What I don't understand is why those variables we are making, Midrange and Super, have reduced costs of zero. Can you explain that?

Holmes: Certainly, Watson. Remember the concept of pricing out a variable. We want to make just one additional unit of that product. So consider the Super computer. Solver tells us to make 250 units. Suppose, for whatever reason, that you insist on making one more, or 251 all together. That additional Super computer needs 0.7 hours of manufacture, 0.25 hours of assembly, and 0.3 hours of inspection. Let us price out this computer:

Manufacturing:	(0.7 hour)*($175/hour) =	$122.50
Assembly:	(0.25 hour)*($0/hour) =	0
Inspection:	(0.3 hour)*($425/hour) =	127.50
		$250.00

So this extra Super needs $250 worth of resources. However, the profit of a Super is $250, and so the profit – cost = $250 - $250 = 0. The reduced cost is zero! Think of it this way. Solver is now talking to you, saying, "Look, I recommended that you make 250 Super computers. You want to make one more. Why? You need $250 worth of resources to make it, and you are bringing in a profit of $250. It is a tie. You are gaining nothing! So why are you going against my recommendation of making 250 such computers? If you want to make one more, you gain nothing, so why bother? In other words, don't do it. Just make the number of Super computers that I recommend, for that is the optimal amount."

Watson: Solver sounds like a "pushy fellow," but I get your point. Thus, you are saying that the reduced cost of a variable that we are making will always be zero.

Holmes: Indeed, Watson. That is what is going on "behind the scenes" as Solver tries to find the optimal solution for us.

ON RECOGNIZING ALTERNATIVE OPTIMA

Watson: Let me ask you another question. You mentioned alternative optima before. That means there is another optimal solution to the linear program with the same profit. If I look at Solver's output, how do I know if the optimal solution that Solver found is unique, or that there are alternative optima?

Holmes: Simple, Watson. We just have to look at the objective sensitivity analysis in the Sensitivity Report. *In the absence of degeneracy, if we simultaneously see the reduced cost and the value of a variable are zero, we have alternative optima*. Unfortunately, the computer does not tell us what these other solutions are.

Watson: That says that our problem does have a unique solution. There is no other optimal solution because I don't have any variable whose optimal value is zero and whose reduced cost is zero also. That's pretty easy. What if the problem were degenerate?

Holmes: If you saw that both the optimal value and the reduced cost of a variable were simultaneously zero, then we might have alternative optima. That is the best we could say.

Watson: Very well, let us forget about degeneracy. I think I even understand *why* your criteria for recognizing alternative optima works. The reduced cost tells us how much our profit would decrease if we made a product that Solver recommends we do not make. But if that value were already zero, and its reduced cost is zero, then we would be indifferent on whether we made it or not. The reason is that the profit would not change. In other words, there is another possible solution to the problem with the same optimal total profit. That is what alternative optima means.

Holmes: Once again, I compliment you on your perception. That is exactly right.

ADDING NEW CONSTRAINTS

Watson: I think that is about enough for now, because I am starting to get a headache. Let me ask you one last question, though; what would occur in such a case: the firm made its formulation, ran the program, and interpreted the output as we have, then it discovered that it had omitted an important constraint. Must the entire analysis be performed once again?

Holmes: No, Watson. All we must do, when adding a new constraint to a problem, is substitute the current optimal solution into that constraint and see what happens. If the constraint is not violated, then the original optimal solution is still optimal.

Watson: What would occur if the constraint is violated?

Holmes: In that case, I am afraid we would have to run the problem over again to get the new optimal solution.

Watson: I see. Well, Holmes, I feel that I do understand linear programming much more. If that textbook that I first investigated had treated the subject as you have, from the computer-output point of view, rather that confusing me with tedious calculations involving equations, I might have known the subject much sooner. Thank you for the discussion.

Holmes: Better late than never. Make an appointment with the management of Maximus Computers to explain all that was said here, and get our fee. I think this may be the beginning of a worthwhile business arrangement between us.

EPILOGUE

I did indeed make an appointment for that week. The discussion with upper management was very similar to the one Holmes had with me. In the end they thanked Holmes and myself for our help, indicated that they would take our advice, and handed me a check for a hefty amount that surprised me greatly. The business venture with Holmes promised to be, at least monetarily, much more rewarding than our previous chosen professions. That night I told my wife to start looking in *The Times* for a new house in a better neighborhood, for we shall be more wealthy very soon.