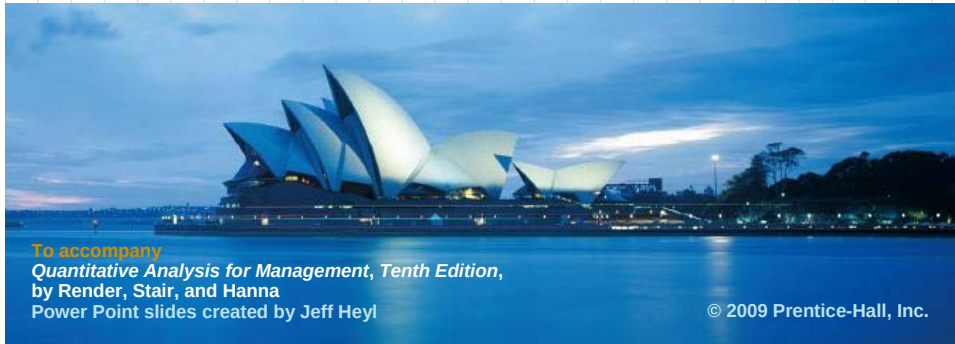


Chapter 11

Integer Programming, Goal Programming, and Nonlinear Programming



Chapter Outline

- 11.1 Introduction
- 11.2 Integer Programming
- 11.3 Modeling with 0-1 (Binary) Variables
- 11.4 Goal Programming
- 11.5 Nonlinear Programming

Learning Objectives

After completing this chapter, students will be able to:

1. Understand the difference between LP and integer programming
2. Understand and solve the three types of integer programming problems
3. Apply the branch and bound method to solve integer programming problems
4. Solve goal programming problems graphically and using a modified simplex technique
5. Formulate nonlinear programming problems and solve using Excel

Introduction

- A large number of business problems can be solved only if variables have integer values
- We will solve integer programming problems graphically and by using the branch and bound method
- Many business problems have multiple objectives
- Goal programming is an extension to LP that can permit multiple objectives
- Linear programming requires linear models
- Nonlinear programming allows objectives and constraints to be nonlinear

Integer Programming

- An integer programming model is one where one or more of the decision variables has to take on an integer value in the final solution
- There are three types of integer programming problems
 1. Pure integer programming where all variables have integer values
 2. Mixed-integer programming where some but not all of the variables will have integer values
 3. Zero-one integer programming are special cases in which all the decision variables must have integer solution values of 0 or 1

Harrison Electric Company Example of Integer Programming

- The Company produces two products popular with home renovators, old-fashioned chandeliers and ceiling fans
- Both the chandeliers and fans require a two-step production process involving wiring and assembly
- It takes about 2 hours to wire each chandelier and 3 hours to wire a ceiling fan
- Final assembly of the chandeliers and fans requires 6 and 5 hours respectively
- The production capability is such that only 12 hours of wiring time and 30 hours of assembly time are available

Integer Programming

- Solving an integer programming problem is much more difficult than solving an LP problem
- Even the fastest computers can take an excessively long time to solve big integer programming problems
- The most common technique used to solve integer programming problems is the branch and bound method

Harrison Electric Company Example of Integer Programming

- Each chandelier produced nets the firm \$7 and each fan \$6
- Harrison's production mix decision can be formulated using LP as follows

$$\begin{aligned}
 &\text{Maximize profit} = \$7X_1 + \$6X_2 \\
 &\text{subject to} \quad \begin{aligned}
 &2X_1 + 3X_2 \leq 12 \quad (\text{wiring hours}) \\
 &6X_1 + 5X_2 \leq 30 \quad (\text{assembly hours}) \\
 &X_1, X_2 \geq 0 \quad (\text{nonnegative})
 \end{aligned}
 \end{aligned}$$

where

X_1 = number of chandeliers produced
 X_2 = number of ceiling fans produced

Harrison Electric Company Example of Integer Programming

The Harrison Electric Problem

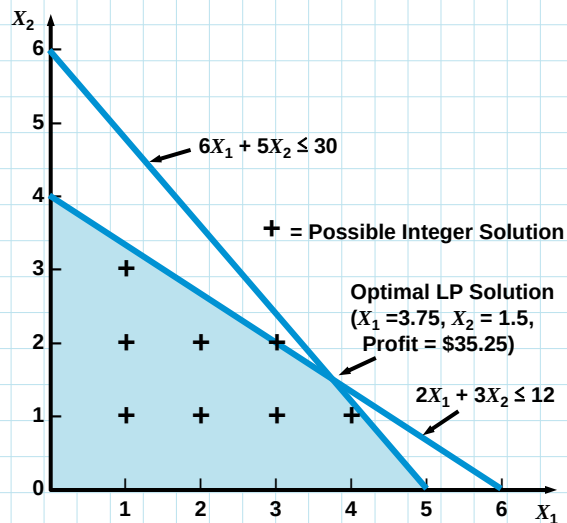


Figure 11.1

© 2009 Prentice-Hall, Inc. 11 - 9

Harrison Electric Company Example of Integer Programming

CHANDELIER (X₁) CEILING FANS (X₂) PROFIT (\$7X₁ + \$6X₂)

0	0	\$0
1	0	7
2	0	14
3	0	21
4	0	28
5	0	35
0	1	6
1	1	13
2	1	20
3	1	27
4	1	34
0	2	12
1	2	19
2	2	26
3	2	33
0	3	18
1	3	25
0	4	24

Integer solutions

Optimal solution to integer programming problem

Solution if rounding is used

Table 11.1

© 2009 Prentice-Hall, Inc. 11 - 11

Harrison Electric Company Example of Integer Programming

- The production planner Wes recognizes this is an integer problem
- His first attempt at solving it is to round the values to $X_1 = 4$ and $X_2 = 2$
- However, this is not feasible
- Rounding X_2 down to 1 gives a feasible solution, but it may not be **optimal**
- This could be solved using the **enumeration** method
- Enumeration is generally not possible for large problems

© 2009 Prentice-Hall, Inc. 11 - 10

Harrison Electric Company Example of Integer Programming

- The rounding solution of $X_1 = 4, X_2 = 1$ gives a profit of \$34
- The optimal solution of $X_1 = 5, X_2 = 0$ gives a profit of \$35
- The optimal integer solution is less than the optimal LP solution
- An integer solution can **never** be better than the LP solution and is **usually** a lesser solution

© 2009 Prentice-Hall, Inc. 11 - 12

Branch-and-Bound Method

- The most common algorithm for solving integer programming problems is the branch-and-bound method
- It starts by first allowing non-integer solutions
- If these values are integer valued, this must also be the solution to the integer problem
- If these variables are not integer valued, the feasible region is divided by adding constraints restricting the value of one of the variables that was not integer valued
- The divided feasible region results in subproblems that are then solved

Six Steps in Solving IP Maximization Problems by Branch and Bound

1. Solve the original problem using LP. If the answer satisfies the integer constraints, we are done. If not, this value provides an initial upper bound.
2. Find any feasible solution that meets the integer constraints for use as a lower bound. Usually, rounding down each variable will accomplish this.
3. Branch on one variable from step 1 that does not have an integer value. Split the problem into two subproblems based on integer values that are immediately above or below the noninteger value.
4. Create nodes at the top of these new branches by solving the new problem

Branch-and-Bound Method

- Bounds on the value of the objective function are found and used to help determine which subproblems can be eliminated and when the optimal solution has been found
- If a solution is not optimal, a new subproblem is selected and branching continues

Six Steps in Solving IP Maximization Problems by Branch and Bound

5. (a) If a branch yields a solution to the LP problem that is **not feasible**, terminate the branch
- (b) If a branch yields a solution to the LP problem that is feasible, but not an integer solution, go to step 6
- (c) If the branch yields a **feasible integer** solution, examine the value of the objective function. If this value equals the upper bound, an optimal solution has been reached. If it not equal to the upper bound, but exceeds the lower bound, terminate this branch.

Six Steps in Solving IP Maximization Problems by Branch and Bound

6. Examine both branches again and set the upper bound equal to the maximum value of the objective function at all final nodes. If the upper bound equals the lower bound, stop. If not, go back to step 3.

Note: Minimization problems involved reversing the roles of the upper and lower bounds

Harrison Electric Company Revisited

- Since X_1 and X_2 are not integers, this solution is not valid
- The profit value of \$35.25 will provide the initial **upper bound**
- We can round down to $X_1 = 3$, $X_2 = 1$, profit = \$27, which provides a feasible **lower bound**
- The problem is now divided into two subproblems

Harrison Electric Company Revisited

- Recall that the Harrison Electric Company's integer programming formulation is

$$\begin{aligned} \text{Maximize profit} &= \$7X_1 + \$6X_2 \\ \text{subject to} \quad &2X_1 + 3X_2 \leq 12 \\ &6X_1 + 5X_2 \leq 30 \end{aligned}$$

where

X_1 = number of chandeliers produced
 X_2 = number of ceiling fans produced

- And the optimal noninteger solution is

$$X_1 = 3.75 \text{ chandeliers}, \quad X_2 = 1.5 \text{ ceiling fans} \\ \text{profit} = \$35.25$$

Harrison Electric Company Revisited

Subproblem A

$$\begin{aligned} \text{Maximize profit} &= \$7X_1 + \$6X_2 \\ \text{subject to} \quad &2X_1 + 3X_2 \leq 12 \\ &6X_1 + 5X_2 \leq 30 \\ &X_1 \geq 4 \end{aligned}$$

Subproblem B

$$\begin{aligned} \text{Maximize profit} &= \$7X_1 + \$6X_2 \\ \text{subject to} \quad &2X_1 + 3X_2 \leq 12 \\ &6X_1 + 5X_2 \leq 30 \\ &X_1 \leq 3 \end{aligned}$$

Harrison Electric Company Revisited

- If you solve both subproblems graphically

Subproblem A's optimal solution: $[X_1 = 4, X_2 = 1.2, \text{profit} = \$35.20]$

Subproblem B's optimal solution: $[X_1 = 3, X_2 = 2, \text{profit} = \$33.00]$

- We have completed steps 1 to 4 of the branch and bound method

Harrison Electric Company Revisited

- Subproblem A has branched into two new subproblems, C and D

Subproblem C

$$\begin{array}{ll} \text{Maximize profit} = & \$7X_1 + \$6X_2 \\ \text{subject to} & 2X_1 + 3X_2 \leq 12 \\ & 6X_1 + 5X_2 \leq 30 \\ & X_1 \geq 4 \\ & X_2 \geq 2 \end{array}$$

Subproblem D

$$\begin{array}{ll} \text{Maximize profit} = & \$7X_1 + \$6X_2 \\ \text{subject to} & 2X_1 + 3X_2 \leq 12 \\ & 6X_1 + 5X_2 \leq 30 \\ & X_1 \geq 4 \\ & X_2 \leq 1 \end{array}$$

Harrison Electric Company Revisited

- Harrison Electric's first branching: subproblems A and B

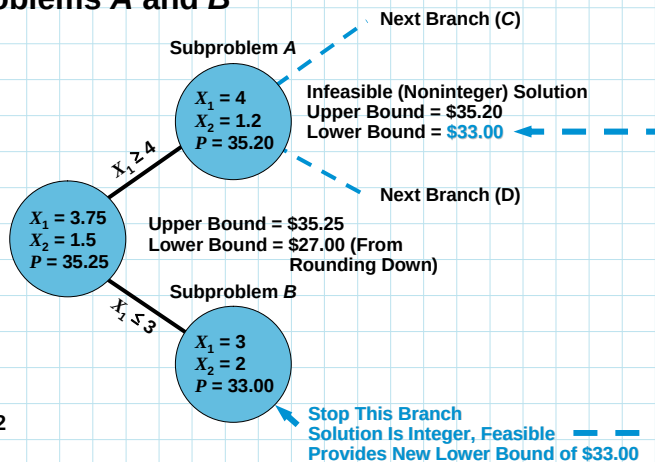


Figure 11.2

Harrison Electric Company Revisited

- Subproblem C has no feasible solution because the all the constraints can not be satisfied
- We terminate this branch and do not consider this solution
- Subproblem D's optimal solution is $X_1 = 4.17$, $X_2 = 1$, profit = \$35.16
- This noninteger solution yields a new upper bound of \$35.16

Harrison Electric Company Revisited

- Finally we create subproblems *E* and *F*

Subproblem E

$$\begin{array}{ll}\text{Maximize profit} = & \$7X_1 + \$6X_2 \\ \text{subject to} & 2X_1 + 3X_2 \leq 12 \\ & 6X_1 + 5X_2 \leq 30 \\ & X_1 \geq 4 \\ & X_1 \leq 4 \\ & X_2 \leq 1\end{array}$$

Optimal solution to *E*:
 $X_1 = 4, X_2 = 1$, profit = \$34

Subproblem D

$$\begin{array}{ll}\text{Maximize profit} = & \$7X_1 + \$6X_2 \\ \text{subject to} & 2X_1 + 3X_2 \leq 12 \\ & 6X_1 + 5X_2 \leq 30 \\ & X_1 \geq 4 \\ & X_1 \geq 5 \\ & X_2 \leq 1\end{array}$$

Optimal solution to *F*:
 $X_1 = 5, X_2 = 0$, profit = \$35

Harrison Electric Company Revisited

- Harrison Electric's full branch and bound solution

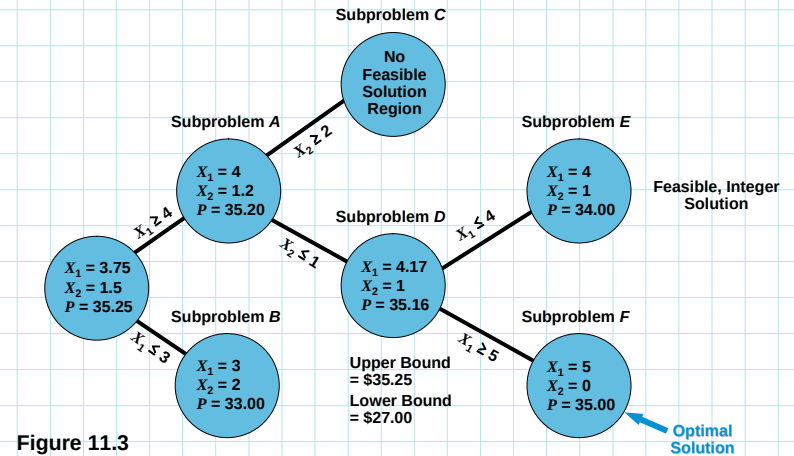


Figure 11.3

Harrison Electric Company Revisited

- The stopping rule for the branching process is that we continue until the new upper bound is less than or equal to the lower bound
- or no further branching is possible
- The later case applies here since both branches yielded feasible integer solutions
- The optimal solution is subproblem *F*'s node
- Computer solutions work well on small and medium problems
- For large problems the analyst may have to settle for a near-optimal solution

Using Software to Solve Harrison Integer Programming Problem

- QM for Windows input screen with Harrison Electric data

Program 11.1A

Using Software to Solve Harrison Integer Programming Problem

- QM for Windows solution screen for Harrison Electric data

Objective: ☒ Maximize ☐ Minimize Maximum number of iterations: 1000

Integer & Mixed Integer Programming Results

Variable	Type	Value
X1	Integer	5
X2	Integer	0
Solution value		35

Program 11.1B

Using Software to Solve Harrison Integer Programming Problem

- Using Excel's Solver to formulate Harrison's integer programming model

Program 11.2A

Enter the names for the objective function and constraints.

Enter the data, including the decision variable names, in columns B, C, and E.

We will place the answers here.

Calculate the total for the objective function and the constraints based on the product of the data coefficients and the variable values using the SUMPRODUCT function.

The objective is in cell D6. The cells to change are specified as B4 to C4.

The variables are integers (2 conditions).

Do not use more wiring or assembly hours than are available (2 constraints).

Harrison Electric IP Analysis

	Chandeliers	Fans	
Solution	1	1	
Profit	7	6	Total = \$UMPRODUCT(\$B\$4:\$C\$4, B6:C6)
wiring hours	2	3	Used = \$UMPRODUCT(\$B\$4:\$C\$4, B8:C8) <= Sign Limit
assembly hours	6	5	Used = \$UMPRODUCT(\$B\$4:\$C\$4, B9:C9) <= Sign Limit

Solver Parameters

Set Target Cell: \$D\$6 To: Max Value of: 0

By Changing Variable Cells: \$B\$4:\$C\$4

Subject to the Constraints:

\$B\$4:\$C\$4 = integer

\$B\$8:\$B\$9 <= \$F\$8:\$F\$9

Using Software to Solve Harrison Integer Programming Problem

- QM for Windows iteration results screen for Harrison Electric data

Program 11.1C

Objective: ☒ Maximize ☐ Minimize Maximum number of iterations: 1000

Iteration Results

Iteration	Level	Added constraint	Solution type	Solution Value	X1	X2
			Optimal	35	5	0
1	0		NONinteger	35.25	3.75	1.5
2	1	X1 <= 3	INTEGER	33	3	2
3	1	X1 >= 4	NONinteger	35.2	4	1.2
4	2	X2 <= 1	NONinteger	35.17	4.17	1
5	3	X1 <= 4	INTEGER	34	4	1
6	3	X1 >= 5	INTEGER	35	5	0
7	2	X2 >= 2	Infeasible			

Using Software to Solve Harrison Integer Programming Problem

- Integer variables are specified with a drop-down menu in Solver

Program 11.2B

Harrison Electric IP Analysis

	Chandeliers	Fans	
Solution	1	1	
Profit	7	6	Total = 13
wiring hours	2	3	Used = 5 <= Sign Limit 12
assembly hours	6	5	Used = 11 <= Sign Limit 30

Add Constraint

Cell Reference: \$B\$4:\$C\$4

Constraint: <=

int bin

Using Software to Solve Harrison Integer Programming Problem

- Excel solution to the Harrison Electric integer programming model

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	Harrison Electric IP Analysis							
2								
3		Chandeliers	Fans					
4	Solution	5	0					
5				Total				
6	Profit	7	6	35				
7				Used	Sign	Limit		
8	wiring hours	2	3	10	<	12		
9	assembly hours	6	5	30	<	30		

The Solver Results dialog box is open, showing: "Solver found a solution. All constraints and optimality conditions are satisfied." The "Keep Solver Solution" radio button is selected. The "Reports" section shows "Answer", "Sensitivity", and "Limits" are selected.

Program 11.2C

© 2009 Prentice-Hall, Inc. 11 - 33

Mixed-Integer Programming Problem Example

AMOUNT PER 50-POUND BAG OF XYLINE (LB)	AMOUNT PER POUND OF HEXALL (LB)	AMOUNT OF INGREDIENTS AVAILABLE
30	0.5	2,000 lb-ingredient A
18	0.4	800 lb-ingredient B
2	0.1	200 lb-ingredient C

- Bagwell wants to maximize profit
- We let X = number of 50-pound bags of xylene
- We let Y = number of pounds of hexall
- This is a mixed-integer programming problem as Y is not required to be an integer

© 2009 Prentice-Hall, Inc. 11 - 35

Mixed-Integer Programming Problem Example

- There are many situations in which some of the variables are restricted to be integers and some are not
- Bagwell Chemical Company produces two industrial chemicals
- Xylene must be produced in 50-pound bags
- Hexall is sold by the pound and can be produced in any quantity
- Both xylene and hexall are composed of three ingredients – A, B, and C
- Bagwell sells xylene for \$85 a bag and hexall for \$1.50 per pound

© 2009 Prentice-Hall, Inc. 11 - 34

Mixed-Integer Programming Problem Example

- The model is

$$\begin{aligned}
 &\text{Maximize profit} = \$85X + \$1.50Y \\
 &\text{subject to} \quad \begin{aligned} 30X &+ 0.5Y \leq 2,000 \\ 30X &+ 0.5Y \leq 800 \\ 30X &+ 0.5Y \leq 200 \end{aligned} \\
 &\quad \quad \quad X, Y \leq 0 \text{ and } X \text{ integer}
 \end{aligned}$$

© 2009 Prentice-Hall, Inc. 11 - 36

Mixed-Integer Programming Problem Example

- Using QM for Windows and Excel to solve Bagwell's IP model

Objective		Instruction		
☒ Maximize ☐ Minimize		There are more results available in additional windows. These may be opened by using the WINDOW option in the Main Menu.		
Bagwell Chemical Company Solution				
	Xylene	Hexall		RHS
Maximize	85.	1.5		
Ingredient A	30.	0.5	<=	2,000.
Ingredient B	18.	0.4	<=	800.
Ingredient C	2.	0.1	<=	200.
Variable type	Integer	Real		
Solution->	44.	20.	Optimal Z->	3,770.

Program 11.3

Mixed-Integer Programming Problem Example

- Excel solution to the Bagwell Chemical problem

	A	B	C	D	E	F	G	H	I
1	Bagwell Chemical Company								
2									
3		xyline (bags)	hexall (lbs)						
4	value	44	20						
5									
6	profit	85	1.5	3770					
7				used	sign	available			
8	ingredient a	30	0.5	1330	<=	2000			
9	ingredient b	18	0.4	800	<=	800			
10	ingredient c	2	0.1	90	<=	200			

Program 11.4B

Mixed-Integer Programming Problem Example

- Excel formulation of Bagwell's IP problem with Solver

Enter the names for the objective function and constraints.

We will place the answers here.

Enter the data, including the decision variable names, in columns B, C, and E.

The objective is in cell D6. The cells to change are specified as B4 to C4.

The number of bags is an integer.

Do not use more of any ingredient than is available (3 constraints).

Program 11.4A

Modeling With 0-1 (Binary) Variables

- We can demonstrate how 0-1 variables can be used to model several diverse situations
- Typically a 0-1 variable is assigned a value of 0 if a certain condition is not met and a 1 if the condition is met
- This is also called a **binary variable**

Capital Budgeting Example

- A common capital budgeting problem is selecting from a set of possible projects when budget limitations make it impossible to select them all
- A 0-1 variable is defined for each project
- Qumo Chemical Company is considering three possible improvement projects for its plant
 - A new catalytic converter
 - A new software program for controlling operations
 - Expanding the storage warehouse
- It can not do them all
- They want to maximize net present value of projects undertaken

© 2009 Prentice-Hall, Inc. 11 - 41

Capital Budgeting Example

- The decision variables are

$$X_1 = \begin{cases} 1 & \text{if catalytic converter project is funded} \\ 0 & \text{otherwise} \end{cases}$$

$$X_2 = \begin{cases} 1 & \text{if software project is funded} \\ 0 & \text{otherwise} \end{cases}$$

$$X_3 = \begin{cases} 1 & \text{if warehouse expansion project is funded} \\ 0 & \text{otherwise} \end{cases}$$

- The mathematical statement of the integer programming problem becomes

$$\begin{aligned} \text{Maximize NPV} &= 25,000X_1 + 18,000X_2 + 32,000X_3 \\ \text{subject to} \quad &8,000X_1 + 6,000X_2 + 12,000X_3 \leq 20,000 \\ &7,000X_1 + 4,000X_2 + 8,000X_3 \leq 16,000 \\ &X_1, X_2, X_3 = 0 \text{ or } 1 \end{aligned}$$

© 2009 Prentice-Hall, Inc. 11 - 43

Capital Budgeting Example

- Qumo Chemical Company information

PROJECT	NET PRESENT VALUE	YEAR 1	YEAR 2
Catalytic Converter	\$25,000	\$8,000	\$7,000
Software	\$18,000	\$6,000	\$4,000
Warehouse expansion	\$32,000	\$12,000	\$8,000
Available funds		\$20,000	\$16,000

Table 11.2

- The basic model is
 Maximize net present value of projects undertaken
 subject to Total funds used in year 1 $\leq$ \$20,000
 Total funds used in year 2 $\leq$ \$16,000

© 2009 Prentice-Hall, Inc. 11 - 42

Capital Budgeting Example

- Solved with computer software, the optimal solution is $X_1 = 1$, $X_2 = 0$, and $X_3 = 1$ with an objective function value of \$57,000
- This means that Qumo Chemical should fund the catalytic converter and warehouse expansion projects only
- The net present value of these investments will be \$57,000

© 2009 Prentice-Hall, Inc. 11 - 44

Limiting the Number of Alternatives Selected

- One common use of 0-1 variables involves limiting the number of projects or items that are selected from a group
- Suppose Quemo Chemical is required to select no more than two of the three projects *regardless* of the funds available

- This would require adding a constraint

$$X_1 + X_2 + X_3 \leq 2$$

- If they had to fund *exactly* two projects the constraint would be

$$X_1 + X_2 + X_3 = 2$$

Fixed-Charge Problem Example

- Often businesses are faced with decisions involving a fixed charge that will affect the cost of future operations
- Sitka Manufacturing is planning to build at least one new plant and three cities are being considered in
 - Baytown, Texas
 - Lake Charles, Louisiana
 - Mobile, Alabama
- Once the plant or plants are built, the company want to have capacity to produce at least 38,000 units each year

Dependent Selections

- At times the selection of one project depends on the selection of another project
- Suppose Quemo's catalytic converter could only be purchased if the software was purchased
- The following constrain would force this to occur

$$X_1 \leq X_2 \quad \text{or} \quad X_1 - X_2 \leq 0$$

- If we wished for the catalytic converter and software projects to either both be selected or both not be selected, the constraint would be

$$X_1 = X_2 \quad \text{or} \quad X_1 - X_2 = 0$$

Fixed-Charge Problem Example

- Fixed and variable costs for Sitka Manufacturing

SITE	ANNUAL FIXED COST	VARIABLE COST PER UNIT	ANNUAL CAPACITY
Baytown, TX	\$340,000	\$32	21,000
Lake Charles, LA	\$270,000	\$33	20,000
Mobile, AL	\$290,000	\$30	19,000

Table 11.3

Fixed-Charge Problem Example

- We can define the decision variables as

$$X_1 = \begin{cases} 1 & \text{if factory is built in Baytown} \\ 0 & \text{otherwise} \end{cases}$$

$$X_2 = \begin{cases} 1 & \text{if factory is built in Lake Charles} \\ 0 & \text{otherwise} \end{cases}$$

$$X_3 = \begin{cases} 1 & \text{if factory is built in Mobile} \\ 0 & \text{otherwise} \end{cases}$$

$$X_4 = \text{number of units produced at Baytown plant}$$

$$X_5 = \text{number of units produced at Lake Charles plant}$$

$$X_6 = \text{number of units produced at Mobile plant}$$

Financial Investment Example

- Numerous financial applications exist with 0-1 variables
- Simkin, Simkin, and Steinberg specialize in recommending oil stock portfolios for wealthy clients
- One client has the following specifications
 - At least two Texas firms must be in the portfolio
 - No more than one investment can be made in a foreign oil company
 - One of the two California oil stocks must be purchased
- The client has \$3 million to invest and wants to buy large blocks of shares

Fixed-Charge Problem Example

- The integer programming formulation becomes

$$\text{Minimize cost} = 340,000X_1 + 270,000X_2 + 290,000X_3 + 32X_4 + 33X_5 + 30X_6$$

$$\begin{aligned} \text{subject to } X_4 + X_5 + X_6 &\geq 38,000 \\ X_4 &\leq 21,000X_1 \\ X_5 &\leq 20,000X_2 \\ X_6 &\leq 19,000X_3 \\ X_1, X_2, X_3 &= 0 \text{ or } 1; \\ X_4, X_5, X_6 &\geq 0 \text{ and integer} \end{aligned}$$

- The optimal solution is

$$X_1 = 0, X_2 = 1, X_3 = 1, X_4 = 0, X_5 = 19,000, X_6 = 19,000$$

Objective function value = \$1,757,000

Financial Investment Example

- Oil investment opportunities

STOCK	COMPANY NAME	EXPECTED ANNUAL RETURN (\$1,000s)	COST FOR BLOCK OF SHARES (\$1,000s)
1	Trans-Texas Oil	50	480
2	British Petroleum	80	540
3	Dutch Shell	90	680
4	Houston Drilling	120	1,000
5	Texas Petroleum	110	700
6	San Diego Oil	40	510
7	California Petro	75	900

Table 11.4

Financial Investment Example

Model formulation

Maximize return = $50X_1 + 80X_2 + 90X_3 + 120X_4 + 110X_5 + 40X_6 + 75X_7$

subject to

$$X_1 + X_4 + X_5 \geq 2 \quad (\text{Texas constraint})$$

$$X_2 + X_3 \leq 1 \quad (\text{foreign oil constraint})$$

$$X_6 + X_7 = 1 \quad (\text{California constraint})$$

$$480X_1 + 540X_2 + 680X_3 + 1,000X_4 + 700X_5 + 510X_6 + 900X_7 \leq 3,000 \quad (\text{\$3 million limit})$$

All variables must be 0 or 1

Using Excel to Solve the Simkin Example

Complete Solver input for Simkin's 0-1 integer programming problem

The objective function is specified as cell D12.

The answers will be in cells C4 to C10.

The constraints, including the binary conditions, are shown here.

Program 11.5B

Using Excel to Solve the Simkin Example

Solver input for Simkin's 0-1 variables

The binary conditions for the variables are set in the constraint window.

Program 11.5A

Using Excel to Solve the Simkin Example

Excel solution to Simkin's 0-1 integer programming problem

Stock	Company Name	Invest	Return	Cost
1	Trans-Texas Oil	0	50	480
2	British Petroleum	0	80	540
3	Dutch Shell	1	90	680
4	Houston Drilling	1	120	1000
5	Texas Petroleum	1	110	700
6	San Diego Oil	1	40	510
7	California Petro	0	75	900
Total			360	2890
Limit				3000
Texas Constraint		2	>=	2
Foreign oil constraint		1	<=	1
California Constraint		1	=	1

Program 11.5C

Goal Programming

- Firms often have more than one goal
- They may want to achieve several, sometimes contradictory, goals
- In linear and integer programming methods the objective function is measured in one dimension only
- It is not possible for LP to have **multiple goals** unless they are all measured in the same units, and this is a highly unusual situation
- An important technique that has been developed to supplement LP is called **goal programming**

Example of Goal Programming Harrison Electric Revisited

- The LP formulation for the Harrison Electric problem is

$$\begin{aligned} \text{Maximize profit} &= \$7X_1 + \$6X_2 \\ \text{subject to} \quad &2X_1 + 3X_2 \leq 12 \text{ (wiring hours)} \\ &6X_1 + 5X_2 \leq 30 \text{ (assembly hours)} \\ &X_1, X_2 \geq 0 \end{aligned}$$

where

X_1 = number of chandeliers produced

X_2 = number of ceiling fans produced

Goal Programming

- Typically goals set by management can be achieved only at the expense of other goals
- A hierarchy of importance needs to be established so that higher-priority goals are satisfied before lower-priority goals are addressed
- It is not always possible to satisfy every goal so goal programming attempts to reach a satisfactory level of multiple objectives
- The main difference is in the objective function where goal programming tries to minimize the **deviations** between goals and what we can actually achieve within the given constraints

Example of Goal Programming Harrison Electric Revisited

- Harrison is moving to a new location and feels that maximizing profit is not a realistic objective
- Management sets a profit level of \$30 that would be satisfactory during this period
- The goal programming problem is to find the production mix that achieves this goal as closely as possible given the production time constraints
- We need to define two deviational variables
 - d_1^- = underachievement of the profit target
 - d_1^+ = overachievement of the profit target

Example of Goal Programming Harrison Electric Revisited

- We can now state the Harrison Electric problem as a single-goal programming model

Minimize under or overachievement of profit target $= d_1^- + d_1^+$

subject to $\begin{array}{rcl} \$7X_1 + \$6X_2 + d_1^- - d_1^+ & = & \$30 \quad (\text{profit goal constraint}) \\ 2X_1 + 3X_2 & \leq & 12 \quad (\text{wiring hours}) \\ 6X_1 + 5X_2 & \leq & 30 \quad (\text{assembly hours}) \\ X_1, X_2, d_1^-, d_1^+ & \geq & 0 \end{array}$

Extension to Equally Important Multiple Goals

- The deviational variables are
 d_1^- = underachievement of the profit target
 d_1^+ = overachievement of the profit target
 d_2^- = idle time in the wiring department (underutilization)
 d_2^+ = overtime in the wiring department (overutilization)
 d_3^- = idle time in the assembly department (underutilization)
 d_3^+ = overtime in the assembly department (overutilization)
 d_4^- = underachievement of the ceiling fan goal
 d_4^+ = overachievement of the ceiling fan goal

Extension to Equally Important Multiple Goals

- Now Harrison's management wants to achieve several goals of equal in priority

Goal 1: to produce a profit of \$30 if possible during the production period

Goal 2: to fully utilize the available wiring department hours

Goal 3: to avoid overtime in the assembly department

Goal 4: to meet a contract requirement to produce at least seven ceiling fans

Extension to Equally Important Multiple Goals

- Because management is unconcerned about d_1^+ , d_2^+ , d_3^- , and d_4^+ these may be omitted from the objective function
- The new objective function and constraints are

Minimize total deviation $= d_1^- + d_2^- + d_3^+ + d_4^-$

subject to $\begin{array}{rcl} 7X_1 + 6X_2 + d_1^- - d_1^+ & = & 30 \quad (\text{profit constraint}) \\ 2X_1 + 3X_2 + d_2^- - d_2^+ & = & 12 \quad (\text{wiring hours}) \\ 6X_1 + 5X_2 + d_3^- - d_3^+ & = & 30 \quad (\text{assembly hours}) \\ X_2 + d_4^- - d_4^+ & = & 7 \quad (\text{ceiling fan constraint}) \\ \text{All } X_i, d_i \text{ variables} & \geq & 0 \end{array}$

Ranking Goals with Priority Levels

- In most goal programming problems, one goal will be more important than another, which will in turn be more important than a third
- Goals can be ranked with respect to their importance in management's eyes
- Higher-order goals are satisfied before lower-order goals
- Priorities (P_i 's) are assigned to each deviational variable with the ranking so that P_1 is the most important goal, P_2 the next most important, P_3 the third, and so on

Ranking Goals with Priority Levels

- This effectively means that each goal is infinitely more important than the next lower goal
- With the ranking of goals considered, the new objective function is

$$\text{Minimize total deviation} = P_1d_1^- + P_2d_2^- + P_3d_3^+ + P_4d_4^-$$

- The constraints remain identical to the previous ones

Ranking Goals with Priority Levels

- Harrison Electric has set the following priorities for their four goals

GOAL	PRIORITY
Reach a profit as much above \$30 as possible	P_1
Fully use wiring department hours available	P_2
Avoid assembly department overtime	P_3
Produce at least seven ceiling fans	P_4

Solving Goal Programming Problems Graphically

- We can analyze goal programming problems graphically
- We must be aware of three characteristics of goal programming problems
 1. Goal programming models are all minimization problems
 2. There is no single objective, but multiple goals to be attained
 3. The deviation from the high-priority goal must be minimized to the greatest extent possible before the next-highest-priority goal is considered

Solving Goal Programming Problems Graphically

- Recall the Harrison Electric goal programming model

Minimize total deviation = $P_1d_1^- + P_2d_2^- + P_3d_3^+ + P_4d_4^-$

subject to $7X_1 + 6X_2 + d_1^- - d_1^+ = 30$ (profit)

$2X_1 + 3X_2 + d_2^- - d_2^+ = 12$ (wiring)

$6X_1 + 5X_2 + d_3^- - d_3^+ = 30$ (assembly)

$X_2 + d_4^- - d_4^+ = 7$ (ceiling fans)

All X_i, d_i variables ≥ 0 (nonnegativity)

where

X_1 = number of chandeliers produced

X_2 = number of ceiling fans produced

© 2009 Prentice-Hall, Inc. 11 – 69

Solving Goal Programming Problems Graphically

- Analysis of the first goal

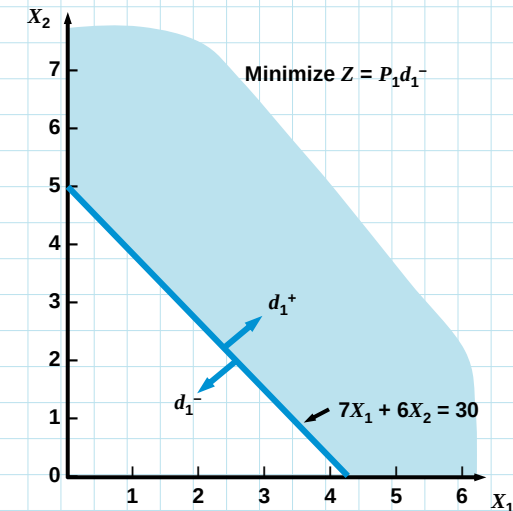


Figure 11.4

© 2009 Prentice-Hall, Inc. 11 – 71

Solving Goal Programming Problems Graphically

- To solve this we graph one constraint at a time starting with the constraint with the highest-priority deviational variables
- In this case we start with the profit constraint as it has the variable d_1^- with a priority of P_1
- Note that in graphing this constraint the deviational variables are ignored
- To minimize d_1^- the feasible area is the shaded region

© 2009 Prentice-Hall, Inc. 11 – 70

Solving Goal Programming Problems Graphically

- The next graph is of the second priority goal of minimizing d_2^-
- The region below the constraint line $2X_1 + 3X_2 = 12$ represents the values for d_2^- while the region above the line stands for d_2^+
- To avoid underutilizing wiring department hours the area below the line is eliminated
- This goal must be attained within the feasible region already defined by satisfying the first goal

© 2009 Prentice-Hall, Inc. 11 – 72

Solving Goal Programming Problems Graphically

- Analysis of first and second goals

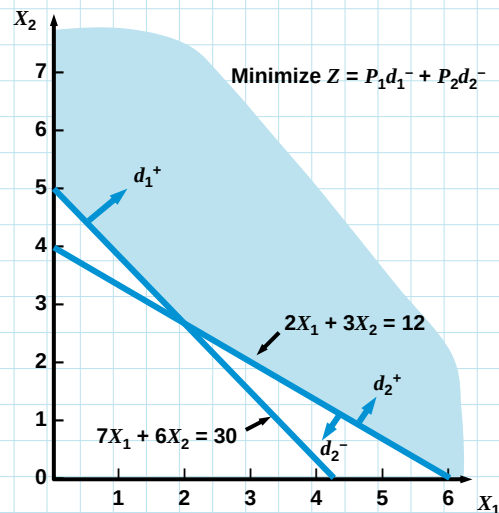


Figure 11.5

© 2009 Prentice-Hall, Inc. 11 - 73

Solving Goal Programming Problems Graphically

- Analysis of all four priority goals

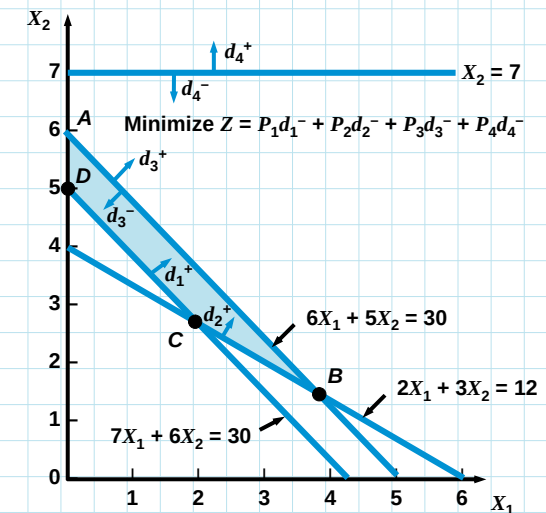


Figure 11.6

© 2009 Prentice-Hall, Inc. 11 - 75

Solving Goal Programming Problems Graphically

- The third goal is to avoid overtime in the assembly department
- We want d_3^+ to be as close to zero as possible
- This goal can be obtained
- Any point inside the feasible region bounded by the first three constraints will meet the three most critical goals
- The fourth constraint seeks to minimize d_4^-
- To do this requires eliminating the area below the constraint line $X_2 = 7$ which is not possible given the previous, higher priority, constraints

© 2009 Prentice-Hall, Inc. 11 - 74

Solving Goal Programming Problems Graphically

- The optimal solution must satisfy the first three goals and come as close as possible to satisfying the fourth goal
- This would be point A on the graph with coordinates of $X_1 = 0$ and $X_2 = 6$
- Substituting into the constraints we find

$d_1^- = \$0$	$d_1^+ = \$6$
$d_2^- = 0$ hours	$d_2^+ = 6$ hours
$d_3^- = 0$ hours	$d_3^+ = 0$ hours
$d_4^- = 1$ ceiling fan	$d_4^+ = 0$ ceiling fans
- A profit of \$36 was achieved exceeding the goal

© 2009 Prentice-Hall, Inc. 11 - 76

Modified Simplex Method for Goal Programming

- The modified simplex method can be used to solve problems with more than two real variables
- Recall the Harrison Electric model

$$\text{Minimize} = P_1 d_1^- + P_2 d_2^- + P_3 d_3^+ + P_4 d_4^-$$

$$\text{subject to } 7X_1 + 6X_2 + d_1^- - d_1^+ = 30$$

$$2X_1 + 3X_2 + d_2^- - d_2^+ = 12$$

$$6X_1 + 5X_2 + d_3^- - d_3^+ = 30$$

$$X_2 + d_4^- - d_4^+ = 7$$

$$\text{All } X_i, d_i \text{ variables} \geq 0$$

Modified Simplex Method for Goal Programming

- There are four features of the modified simplex tableau that differ from earlier simplex tableaus
1. The variables in the problem are listed at the top, with the decision variables (X_1 and X_2) first, then the negative deviational variables and, finally, the positive deviational variables. The priority level of each variable is assigned on the very top row.
 2. The negative deviational variables for each constraint provide the initial basic solution. This is analogous to the use of slack variables in the earlier simplex tableaus. The priority level of each variable in the current solution mix is entered in the C_j column.

Modified Simplex Method for Goal Programming

- Initial goal programming tableau

C_j		0	0	P_1	P_2	0	P_4	0	0	P_3	0	
	SOLUTION MIX	X_1	X_2	d_1^-	d_2^-	d_3^-	d_4^-	d_1^+	d_2^+	d_3^+	d_4^+	QUANTITY
P_1	d_1^-	7	6	1	0	0	0	-1	0	0	0	30
P_2	d_2^-	2	3	0	1	0	0	0	-1	0	0	12
0	d_3^-	6	5	0	0	1	0	0	0	-1	0	30
P_4	d_4^-	0	1	0	0	0	1	0	0	0	-1	7
Z_j	P_4	0	1	0	0	0	1	0	0	0	-1	7
	P_3	0	0	0	0	0	0	0	0	0	0	0
	P_2	2	3	0	1	0	0	0	-1	0	0	12
	P_1	7	6	1	0	0	0	-1	0	0	0	30
$C_j - Z_j$	P_4	0	-1	0	0	0	0	0	0	0	1	
	P_3	0	0	0	0	0	0	0	0	1	0	
	P_2	-2	-3	0	0	0	0	0	1	0	0	
	P_1	-7	-6	0	0	0	0	1	0	0	0	

Pivot column

Table 11.5

Modified Simplex Method for Goal Programming

3. There is a separate X_j and $C_j - Z_j$ row for each of the P_i priorities because different units of measurement are used for each goal. The bottom row of the tableau contains the highest ranked (P_1) goal, the next row has the P_2 goal, and so forth. The rows are computed exactly as in the regular simplex method, but they are done for each priority level.

Modified Simplex Method for Goal Programming

4. In selecting the variable to enter the solution mix, we start with the highest-priority row, P_1 , and select the most negative $C_j - Z_j$ value in it. If there was no negative number for P_1 , we would move on to priority P_2 's $C_j - Z_j$ row and select the largest negative number there. A negative $C_j - Z_j$ that has a positive number in the P row underneath it, however, is ignored. This means that deviations from a more important goal (one in a lower row) would be **increased** if that variable were brought into the solution.

Modified Simplex Method for Goal Programming

Second goal programming tableau

C_j		0	0	P_1	P_2	0	P_4	0	0	P_3	0	
	SOLUTION MIX	X_1	X_2	d_1^-	d_2^-	d_3^-	d_4^-	d_1^+	d_2^+	d_3^+	d_4^+	QUANTITY
0	X_1	1	6/7	1/7	0	0	0	-1/7	0	0	0	30/7
P_2	d_2^-	0	9/7	-2/7	1	0	0	2/7	-1	0	0	24/7
0	d_3^-	0	-1/7	-6/7	0	1	0	6/7	0	-1	0	30/7
P_4	d_4^-	0	1	0	0	0	1	0	0	0	-1	7
Z_j	P_4	0	1	0	0	0	1	0	0	0	-1	7
	P_3	0	0	0	0	0	0	0	0	0	0	0
	P_2	0	9/7	-2/7	1	0	0	2/7	-1	0	0	24/7
	P_1	0	0	0	0	0	0	0	0	0	0	0
$C_j - Z_j$	P_4	0	-1	0	0	0	0	0	0	0	1	
	P_3	0	0	0	0	0	0	0	0	1	0	
	P_2	0	-9/7	2/7	0	0	0	-2/7	1	0	0	
	P_1	0	0	1	0	0	0	0	0	0	0	

Pivot column

Table 11.6

Modified Simplex Method for Goal Programming

- We move towards the optimal solution just as with the regular minimization simplex method
- We find the pivot row by dividing the quantity values by their corresponding pivot column (X_1) values and picking the one with the smallest positive ratio
- In this case, d_1^- leaves the basis and is replaced by X_1
- We continue this process until an optimal solution is reached

Modified Simplex Method for Goal Programming

Final solution to Harrison Electric's goal program

C_j		0	0	P_1	P_2	0	P_4	0	0	P_3	0	
	SOLUTION MIX	X_1	X_2	d_1^-	d_2^-	d_3^-	d_4^-	d_1^+	d_2^+	d_3^+	d_4^+	QUANTITY
0	d_2^+	8/5	0	0	-1	3/5	0	0	1	-3/5	0	6
0	X_2	6/5	1	0	0	1/5	0	0	0	-1/5	0	6
0	d_1^+	1/5	0	-1	0	6/5	0	1	0	-6/5	0	6
P_4	d_4^-	-6/5	0	0	0	-1/5	1	0	0	1/5	-1	1
Z_j	P_4	-6/5	0	0	0	-1/5	1	0	0	1/5	-1	1
	P_3	0	0	0	0	0	0	0	0	0	0	0
	P_2	0	0	0	0	0	0	0	0	0	0	0
	P_1	0	0	0	0	0	0	0	0	1	0	0
$C_j - Z_j$	P_4	6/5	0	0	0	1/5	0	0	0	-1/5	1	
	P_3	0	0	0	0	0	0	0	0	1	0	
	P_2	0	0	0	1	0	0	0	0	0	0	
	P_1	0	0	1	0	0	0	0	0	0	0	

Table 11.7

Modified Simplex Method for Goal Programming

- In the final solution the first three goals have been fully achieved with no negative entries in their $C_j - Z_j$ rows
- A negative value appears in the d_3^+ column in the priority 4 row indicating this goal has not been fully attained
- But the positive number in the d_3^+ at the P_3 priority level (shaded cell) tells us that if we try to force d_3^+ into the solution mix, it will be at the expense of the P_3 goal which has already been satisfied

Goal Programming with Weighted Goals

- Normally priority levels in goal programming assume that each level is infinitely more important than the level below it
- Sometimes this may not be desirable
- A goal may be only two or three times more important than another
- Instead of placing these goals on different levels, we place them on the same level but with different weights
- The coefficients of the deviation variables in the objective function include both the priority level and the weight

Modified Simplex Method for Goal Programming

- The final solution is

$X_1 = 0$ chandeliers produced

$X_2 = 6$ ceiling fans produced

$d_1^+ = \$6$ over the profit goal

$d_2^+ = 6$ wiring hours over the minimum set

$d_4^- = 1$ fewer fan than desired

Goal Programming with Weighted Goals

- Suppose Harrison decides to add another goal of producing at least two chandeliers
- The goal of producing seven ceiling fans is considered twice as important as this goal
- The goal of two chandeliers is assigned a weight of 1 and the goal of seven ceiling fans is assigned a weight of 2 and both of these will be priority level 4
- The new constraint and objective function are

$$X_1 + d_5^- - d_5^+ = 2 \text{ (chandeliers)}$$

$$\text{Minimize} = P_1 d_1^- + P_2 d_2^- + P_3 d_3^+ + P_4 (2d_4^-) + P_4 d_5^-$$

Using QM for Windows to Solve Harrison's Problem

- Harrison Electric's goal programming analysis using QM for Windows
- This table refers to the formulation as presented in Table 11.5

Harrison Electric Company							
	Wt(d+)	Prty(d+)	Wt(d-)	Prty(d-)	X1	X2	RHS
Constraint 1	0	0	1	1	7	6	= 30
Constraint 2	0	0	1	2	2	3	= 12
Constraint 3	1	3	0	0	6	5	= 30
Constraint 4	0	0	1	4	0	1	= 7

Program 11.6A

Using QM for Windows to Solve Harrison's Problem

- Summary solution screen for Harrison Electric's goal programming problem using QM for Windows

Harrison Electric Company Solution			
Item			
Decision variable analysis	Value		
X1	0.		
X2	6.		
Priority analysis	Nonachievement		
Priority 1	0.		
Priority 2	0.		
Priority 3	0.		
Priority 4	1.		
Constraint Analysis	RHS	d+ (row i)	d- (row i)
Constraint 1	30.	6.	0.
Constraint 2	12.	6.	0.
Constraint 3	30.	0.	0.
Constraint 4	7.	0.	1.

Program 11.6C

Using QM for Windows to Solve Harrison's Problem

- Final tableau for Harrison Electric using QM for Windows

Harrison Electric Company Solution											
	X1	X2	d- 1	d- 2	d- 3	d- 4	d+ 1	d+ 2	d+ 3	d+ 4	RHS
Constraint 1	1.6	0.	0.	-1.	0.6	0.	0.	1.	-0.6	0.	6.
Constraint 2	1.2	1.	0.	0.	0.2	0.	0.	0.	-0.2	0.	6.
Constraint 3	0.2	0.	-1.	0.	1.2	0.	1.	0.	-1.2	0.	6.
Constraint 4	-1.2	0.	0.	0.	-0.2	1.	0.	0.	0.2	-1.	1.
Priority 4	-1.2	0.	0.	0.	-0.2	0.	0.	0.	0.2	-1.	1.
Priority 3	0.	0.	0.	0.	0.	0.	0.	0.	-1.	0.	0.
Priority 2	0.	0.	0.	-1.	0.	0.	0.	0.	0.	0.	0.
Priority 1	0.	0.	-1.	0.	0.	0.	0.	0.	0.	0.	0.

Program 11.6B

Nonlinear Programming

- The methods seen so far have all assumed that the problem's objective function and constraints are linear
- Terms such as X_1^3 , $1/X_2$, $\log X_3$, or $5X_1X_2$ are not allowed
- But there are many nonlinear relationships in the real world that would require the objective function, constraint equations, or both to be nonlinear
- Excel can be used to solve these **nonlinear programming (NLP)** problems

Nonlinear Objective Function and Linear Constraints

- The Great Western Appliance Company sells two models of toaster ovens, the Microtoaster (X1) and the Self-Clean Toaster Oven (X2)
- They earn a profit of \$28 for each Microtoaster no matter the number of units sold
- For the Self-Clean oven, profits increase as more units are sold due to a fixed overhead
- The profit function for the Self-Clean over may be expressed as

$$21X_2 + 0.25X_2^2$$

Nonlinear Objective Function and Linear Constraints

- An Excel formulation of Great Western's problem

Great Western Appliance

Variables are located here.

Microtoast Self-clean Total

Number 2 2 =B5+C5 < 1000

Profit =28*B5 =21*C5+0.25*C5^2 =B6+C6

Profit is nonlinear.

used Sign capacity

Hours 0.5 0.4 =SUMPRODUCT(\$B\$5:\$C\$5,B9:C9) < 500

Solver Parameters

Set Target Cell: \$D\$6

Equal To: Max Min Value of: 0

By Changing Variable Cells: \$B\$5:\$C\$5

Subject to the Constraints:

\$D\$5 <= \$F\$5

\$D\$9 <= \$F\$9

Guarantee that we do not make more than 1,000 units.

Guarantee that we do not use more than 500 hours.

The objective is in cell D6. The cells to change are specified as B5 and C5.

NOTE: In the Options dialog box for this problem, assume Linear Model is not checked.

Program 11.7A

Nonlinear Objective Function and Linear Constraints

- The objective function is nonlinear and there are two linear constraints on production capacity and sales time available

Maximize profit = $28X_1 + 21X_2 + 0.25X_2^2$

subject to $X_1 + 21X_2 \leq 1,000$ (units of production capacity)

$0.5X_1 + 0.4X_2 \leq 500$ (hours of sales time available)

$X_1, X_2 \geq 0$

- When an objective function contains a squared term and the problem constraints are linear, it is called a **quadratic programming** problem

Nonlinear Objective Function and Linear Constraints

- Solution to Great Western Appliance's NLP problem using Excel Solver

	A	B	C	D	E	F	G	H	I
1	Great Western Appliance								
2									
3									
4		Microtoast	Self-clean	Total					
5	Number	0	1000	1000	<	1000			
6	Profit	0	271000	\$271,000.00					
7									
8				used	Sign	capacity			
9	Hours	0.5	0.4	400	<	500			

Program 11.7B

Both Nonlinear Objective Function and Nonlinear Constraints

- The annual profit at a medium-sized (200-400 beds) Hospicare Corporation hospital depends on the number of medical patients admitted (X_1) and the number of surgical patients admitted (X_2)
- The objective function for the hospital is nonlinear
- They have identified three constraints, two of which are nonlinear
 - Nursing capacity - nonlinear
 - X-ray capacity - nonlinear
 - Marketing budget required

Both Nonlinear Objective Function and Nonlinear Constraints

- An Excel formulation of Hospicare's NLP problem

The variables are in these cells. Because of the squared terms, it makes more sense to initialize them to 2 rather than 1 to check that the formulas are correct.

Enter the objective and constraint coefficients.

Set up one column for each term that is needed in either the objective function or a constraint. Profit is nonlinear.

The goal is in H8, and the cells to change are in B3 to C3.

The constraints are all of the form "≤" (3 constraints).

NOTE: In the Options dialog box for this problem, assume Linear Model is not checked.

Program 11.8A

Both Nonlinear Objective Function and Nonlinear Constraints

- The objective function and constraint equations for this problem are

$$\begin{aligned} \text{Maximize profit} &= \$13X_1 + \$6X_1X_2 + \$5X_2 + \$1/X_2 \\ \text{subject to} \quad 2X_1^2 + 4X_2 &\leq 90 \quad (\text{nursing capacity in thousands of labor-days}) \\ X_1 + X_2^3 &\leq 75 \quad (\text{x-ray capacity in thousands}) \\ 8X_1 - 2X_2 &\leq 61 \quad (\text{marketing budget required in thousands of \$}) \end{aligned}$$

Both Nonlinear Objective Function and Nonlinear Constraints

- Excel solution to the Hospicare Corp. NLP problem using Solver

	A	B	C	D	E	F	G	H	I	J
1	Hospicare Corp									
2		x1	x2							
3	value	6.066259	4.100253							
4										
5	terms	x1	x1^2	x1*x2	x2	x2^3	1/x2			
6	values	6.066259	36.79949	24.87319	4.100253	68.93374	0.243887			
7										
8	revenue		13		6	5		1	total	248.85
9										
10	constraint 1			2		4			90 <	90
11	constraint 2		1				1		75 <	75
12	constraint 3		8			-2			40.33 <	61

Program 11.8B

Linear Objective Function and Nonlinear Constraints

- Thermlock Corp. produces massive rubber washers and gaskets like the type used to seal joints on the NASA Space Shuttles
- It combines two ingredients, rubber (X_1) and oil (X_2)
- The cost of the industrial quality rubber is \$5 per pound and the cost of high viscosity oil is \$7 per pound
- Two of the three constraints are nonlinear

Linear Objective Function and Nonlinear Constraints

- Excel formulation of Thermlock's NLP problem

The variables are in these cells. Because of the nonlinear terms, it makes more sense to initialize them to 2 rather than 1 to check that the formulas are correct.

Enter the objective and constraint coefficients.

Set up one column for each term that is needed in either the objective function or a constraint.

The goal is in E6, and the cells to change are in B4 to C4.

The constraints are all of the form $\geq$ (3 constraints).

The constraints are nonlinear. NOTE: In the Options dialog box for this problem, assume Linear Model is not checked.

	A	B	C	D	E	F	G	H	I	J	
1	Thermlock Gaskets										
2											
3		x1	x2								
4	value	2	2								
5					total						
6	cost	5	7		=SUMPRODUCT(\$B\$4:\$C\$4,B5:C6)						
7	constraints										
8		x1	x1^2	x1^3	x2	x2^2					
9	value	=B4	=B4^2	=B4^3	=C4	=C4^2	Total				
10	Constraint 1	3	0.25		4	0.3	=SUMPRODUCT(\$B\$10:\$F\$10,B\$11:F\$11)	>	125		
11	Constraint 2	13			1		=SUMPRODUCT(\$B\$10:\$F\$10,B\$12:F\$12)	>	80		
12	Constraint 3	0.7			1		=SUMPRODUCT(\$B\$10:\$F\$10,B\$13:F\$13)	>	17		

Solver Parameters

Set Target Cell: \$E\$6 To: Max Of Value of: 0

By Changing Variable Cells: \$B\$4:\$C\$4

Subject to the Constraints:

\$G\$11:\$G\$13 >= \$H\$11:\$H\$13

Program 11.9A

Linear Objective Function and Nonlinear Constraints

- The firm's objective function and constraints are

$$\text{Minimize costs} = \$5X_1 + \$7X_2$$

$$\text{subject to } \$5X_1 + 0.25X_1^2 + 4X_2 + 0.3X_2^2 \geq 125 \quad (\text{hardness constraint})$$

$$13X_1 + X_1^3 \geq 80 \quad (\text{tensile strength})$$

$$0.7X_1 + X_2 \geq 17 \quad (\text{elasticity})$$

Linear Objective Function and Nonlinear Constraints

- Solution to Thermlock's NLP problem using Excel Solver

	A	B	C	D	E	F	G	H	I
1	Thermlock Gaskets								
2									
3		x1	x2						
4	value	3.325326	14.67227						
5					total				
6	cost	5	7		119.3325				
7									
8	constraints								
9		x1	x1^2	x1^3	x2	x2^2			
10	value	3.325326	11.05779	36.77076	14.67227	215.2756	Total		
11	Constraint 1	3	0.25		4	0.3	136.0122	>	125
12	Constraint 2	13		1			80	>	80
13	Constraint 3	0.7			1		17	>	17

Program 11.9B

Computational Procedures for Nonlinear Programming

- Computational procedures for nonlinear problems do not always yield an optimal solution in a finite number of steps
- There is no general method for solving all nonlinear problems
- **Classical optimization** techniques based on calculus can handle some simpler problems
- The **gradient method** (sometimes called the **steepest ascent method**) is an iterative procedure that moves from one feasible solution to the next improving the objective function

Computational Procedures for Nonlinear Programming

- The best way to deal with nonlinear problems may be to reduce them to a linear or near-linear form
- Separable programming deals with a class of problems in which the objective and constraints are approximated by linear functions
- The simplex algorithm may then be applied
- In general, work in the area of NLP is the most difficult of all the quantitative analysis models