Department of Mechanical and Production Engineering
Ahsanullah University of Science and Technology (AUST)

IPE 3114: Operations Research-I Sessional
Credit Hour: 1.5

General Guidelines:

1. Students must be present on time as per the scheduled time of the lab.
2. The report/assignment for each experiment must be submitted in the following lab session.
3. The structure of the report/assignment for each experiment will be provided by the respective teacher.
4. Students must form groups for the project and submit their project proposal before the midterm break.
5. The final project presentation will be conducted at the end of the semester, before the preparatory leave.
6. A final quiz on the experiments will be conducted at the end of the semester.
7. Marks Distribution:

Attendance	Report	Viva	Presentation	Final Quiz
10	30	10	10	40

Experiment-1:
Introduction to Linear Programming: A Graphical Approach

1. Introduction

Linear Programming (LP) is a core optimization methodology within Operations Research used for allocating scarce resources in an optimal manner. It provides a systematic way to express real-world resource-allocation problems mathematically through **decision variables, an objective function, and a set of constraints.** The most common application scenario involves distributing limited resources among competing activities to achieve the best possible (i.e., optimal) outcome.

More specifically, LP focuses on selecting activity levels where multiple activities compete for scarce resources required for their execution. Once these activity levels are chosen, the corresponding resource consumption is determined automatically. This structure captures a wide range of real-world applications: allocating production facilities to different products, distributing national resources among domestic needs, portfolio selection, designing shipping patterns, agricultural planning, radiation therapy planning, and numerous other contexts. Despite their diversity, these problems share a fundamental characteristic which is the need to allocate resources to activities by choosing appropriate activity levels.

LP uses a mathematical model to represent these allocation problems. The term linear indicates that all mathematical relationships (objective function and constraints) must be linear. The term programming refers to planning, not computer programming. Thus, linear programming represents the planning of activities to achieve the optimal result, that is, the solution that best satisfies the specified objective while adhering to all feasibility requirements.

The graphical approach serves as the simplest exact solution technique for LP models involving two decision variables. This method offers valuable geometric insight into the structure of LP problems by illustrating feasibility, optimality, and the interactions among constraints. We will use MATLAB to get the graphical solution.

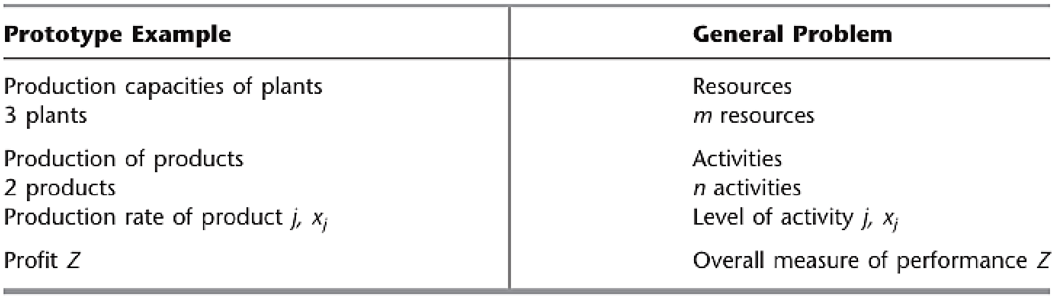
2. The Linear Programming Model

2.1 Linear Programming: Major Components

- **Objective Function**: LP problems seek to maximize or minimize some quantity (usually profit or cost). We refer to this property as the objective function of an LP problem.
- **Constraints:** The presence of restrictions or constraints, limits the degree to which we can pursue our objective. For example, deciding how many units of each product in a firm’s product line to manufacture is restricted by available labor and machinery.
- **Decision/Optimization variables:** The variables that change at each iteration of optimization. In any linear programming model, the decision variables should completely describe the decisions to be made.

Common Terminology

Table 2.1: Common Terminology for Linear Programming



Prototype Example	**General Problem**
Production capacities of plants	Resources
3 plants	*m* resources
Production of products	Activities
2 products	*n* activities
Production rate of product *j*, x_j	Level of activity *j*, x_j
Profit *Z*	Overall measure of performance *Z*

$Z =$ value of overall measure of performance

$x_j =$ level of activity j (for j = 1, 2, . . . , n).

$c_j =$ increase in Z that would result from each unit increase in level of activity j.

$b_i = amount\ of\ resource\ i\ that\ is\ available\ for\ allocation\ to\ activities\ (for\ i = 1,2, \ldots . . m)$

$a_{ij} =$ amount of resource i consumed by each unit of activity j.

2.2 A Standard Form of the Model

$$\text{Maximize } Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

Subject to the restrictions:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$$

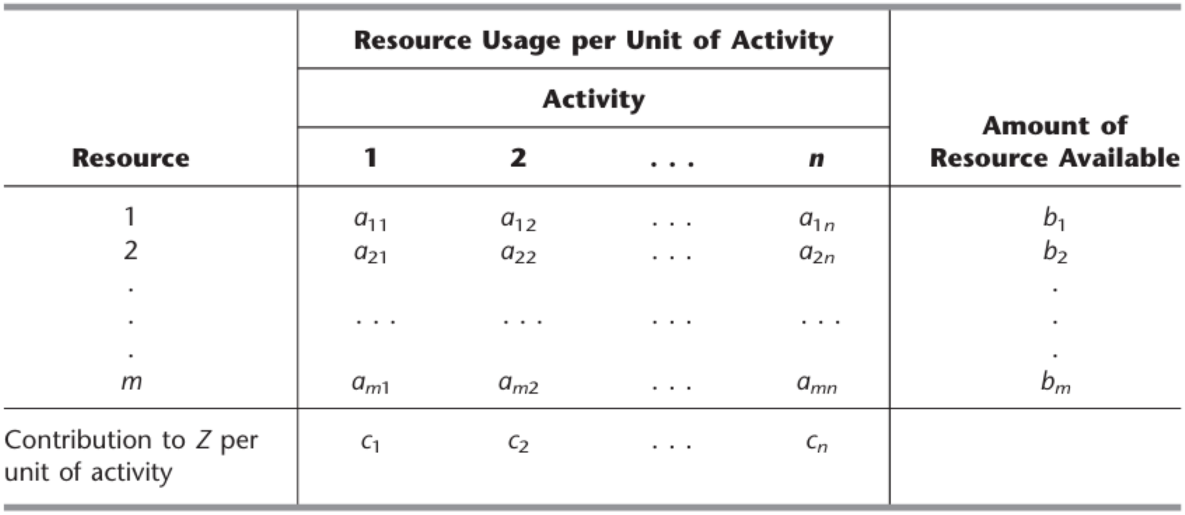
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2$$

……

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m$$

$$\text{and } x_1 \geq 0,\ x_2 \geq 0, \ldots\ x_n \geq 0$$

Table 2.2: Data needed for a linear programming model involving the allocation of resources to activities



	Resource Usage per Unit of Activity				
	Activity				
Resource	**1**	**2**	**. . .**	***n***	**Amount of Resource Available**
1	a_{11}	a_{12}	. . .	a_{1n}	b_1
2	a_{21}	a_{22}	. . .	a_{2n}	b_2
.					.
.	. . .	. . .	. . .	. . .	.
.					.
m	a_{m1}	a_{m2}	. . .	a_{mn}	b_m
Contribution to Z per unit of activity	c_1	c_2	. . .	c_n	

2.2.1 Other Forms

- Minimizing rather than maximizing the objective function
- Some functional constraints with a greater-than-or equal-to inequality
- Some functional constraints in equation form
- Deleting the nonnegativity constraints for some decision variables

2.2.2 Solution Terminology

- A ***feasible solution*** is a solution for which all the constraints are satisfied.

- An ***infeasible solution*** is a solution for which at least one constraint is violated.

- The ***feasible region*** is the collection of all feasible solutions.

- It is possible for a problem to have ***no feasible solutions***.

- An ***optimal solution*** is a feasible solution that has the most favorable value of the objective function. The most favorable value is the largest value if the objective function is to be maximized, whereas it is the smallest value if the objective function is to be minimized.

- Most problems will have just one optimal solution. However, it is possible to have ***more than one.***

- Another possibility is that a problem has ***no optimal solutions***. This occurs only if (1) **it has no feasible solutions or (2) the constraints do not prevent improving the value of the objective function (Z) indefinitely in the favorable direction (positive or negative**). The latter case is referred to as having an ***unbounded Z***.

- **Binding and Nonbinding Constraints:** A constraint is binding if the left-hand side and the right-hand side of the constraint are equal when the optimal values of the decision variables are substituted into the constraint.

A constraint is nonbinding if the left-hand side and the right-hand side of the constraint are unequal when the optimal values of the decision variables are substituted into the constraint.

- **Convex Set:** A set of points S is a convex set if the line segment joining any pair of points in S is wholly contained in S. Figure 3 gives four illustrations of this definition. In Figures 3a and 3b, each line segment joining two points in S contains only points in S. Thus, in both these figures, S is convex. In Figures 3c and 3d, S is not convex. In each figure, points A and B are in S, but there are points on the line segment AB that are not contained in S. In our study of linear programming, a certain type of point in a convex set (called an extreme point) is of great interest.

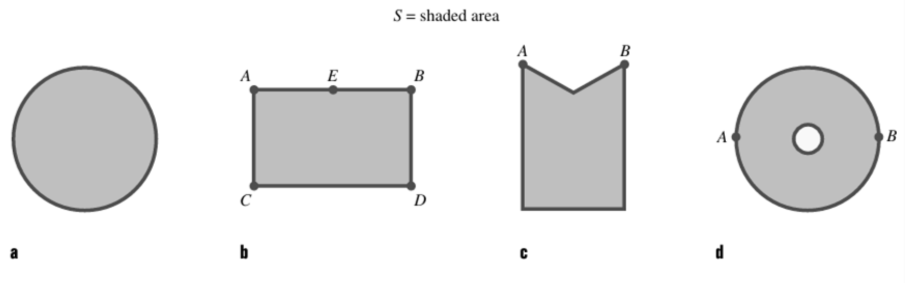


Figure 2.1: Convex and Non-convex sets

- **Extreme Point:** For any convex set S, a point P in S is an extreme point if each line segment that lies completely in S and contains the point P has P as an endpoint of the line segment. Extreme points are sometimes called corner points.

- A **corner-point feasible (CPF**) solution is a solution that lies at a corner of the feasible region. Relationship between optimal solutions and CPF solutions: For any linear programming problem with feasible solutions and a bounded feasible region:
  - The best CPF solution must be an optimal solution.
  - If the problem has multiple optimal solutions at least two must be CPF solutions

2.3 Prototype Example

The WYNDOR GLASS CO. produces high-quality glass products, including windows and glass doors. It has three plants. Aluminum frames and hardware are made in Plant 1, wood frames are made in Plant 2, and Plant 3 produces the glass and assembles the products. Because of declining earnings, top management has decided to revamp the company’s product line. Unprofitable products are being discontinued, releasing production capacity to launch two new products having large sales potential:

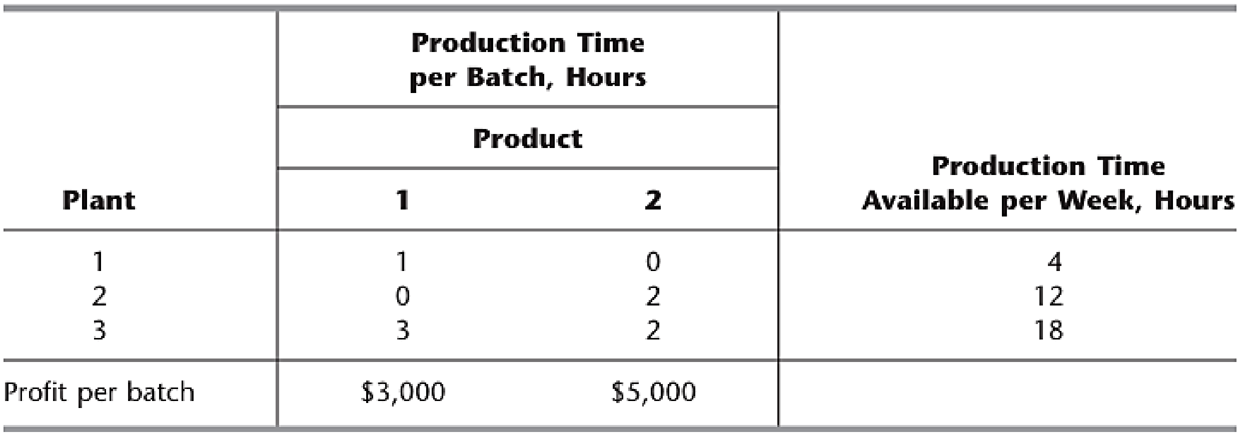
Product 1: An 8-foot glass door with aluminum framing

Product 2: A 4 6-foot double-hung wood-framed window

Product 1 requires some of the production capacity in Plants 1 and 3, but none in Plant 2. Product 2 needs only Plants 2 and 3. The marketing division has concluded that the company could sell as much of either product as could be produced by these plants. However, because both products would be competing for the same production capacity in Plant 3, it is not clear which mix of the two products would be most profitable. Therefore, an OR team has been formed to study this question. The OR team began by having discussions with upper management to identify management's objectives for the study. These discussions led to developing the following definition of the problem:

Determine what the production rates should be for the two products in order to maximize their total profit, subject to the restrictions imposed by the limited production capacities available in the three plants. (Each product will be produced in batches of 20, so the production rate is defined as the number of batches produced per week.) Any combination of production rates that satisfies these restrictions is permitted, including producing none of one product and as much as possible of the other.

Table 2.3: Data for Wyndor Glass Co. Problem



	Production Time per Batch, Hours		
	Product		
Plant	**1**	**2**	**Production Time Available per Week, Hours**
1	1	0	4
2	0	2	12
3	3	2	18
Profit per batch	$3,000	$5,000	

<u>Formulation as a Linear Programming Problem</u>

Decision Variables: To formulate the mathematical (linear programming) model for this problem, let

x_1 = number of batches of product 1 produced per week

x_2 = number of batches of product 2 produced per week

Objective Function:
Z = total profit per week (in thousands of dollars) from producing these two products

$$Z = 3x_1 + 5x_2$$

The coefficient of a variable in the objective function is called the objective function coefficient of the variable. In this example (and in many other problems), the objective function coefficient for each variable is simply the contribution of the variable to the company's profit.

Constraints:
Constraint 1: Each week no more than 4 hours of production time is available in plant 1.

$$x_1 \leq 4$$

Constraint 2: Each week no more than 12 hours of production time is available in plant 2.

$$2x_2 \leq 12$$

Constraint 3: Each week no more than 18 hours of production time is available in plant 3.

$$3x_1 + 2x_2 \leq 18$$

Sign Restrictions: To complete the formulation of a linear programming problem, the following question must be answered for each decision variable: Can the decision variable only assume nonnegative values, or is the decision variable allowed to assume both positive and negative values? If a decision variable x_i can only assume nonnegative values, then we add the sign restriction $x_i \geq 0$. If a variable x_i can assume both positive and negative (or zero) values, then we say that x_i is **unrestricted in sign (often abbreviated urs**). For this problem it is clear that:

$$x_1 \geq 0$$

$$x_2 \geq 0$$

To summarize, in the mathematical language of linear programming, the problem is to choose values of x_1 and x_2 so as to:

$$\text{Maximize } Z = 3x_1 + 5x_2$$

Subject to the restrictions:

$$x_1 \leq 4$$

$$2x_2 \leq 12$$

$$3x_1 + 2x_2 \leq 18$$

$$\text{and } x_1 \geq 0, x_2 \geq 0$$

2.4 Graphical Solution Using MATLAB

This very small problem has only two decision variables and therefore only two dimensions, so a graphical procedure can be used to solve it. This procedure involves constructing a two-dimensional graph with x_1 and x_2 as the axes.

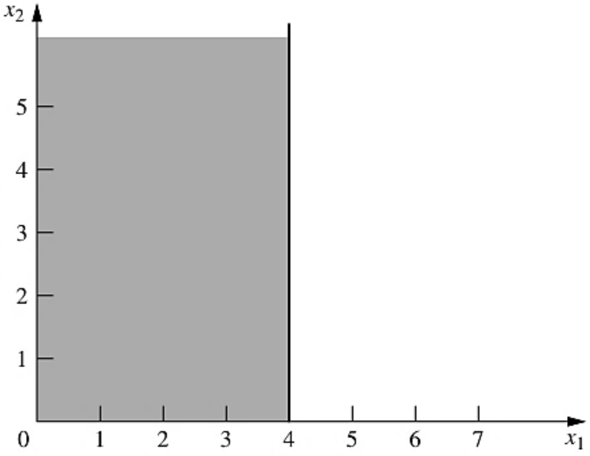


Figure 2.2: Shaded area shows values of (x_1, x_2) allowed by $x_1 \geq 0, x_2 \geq 0, x_1 \leq 4$.

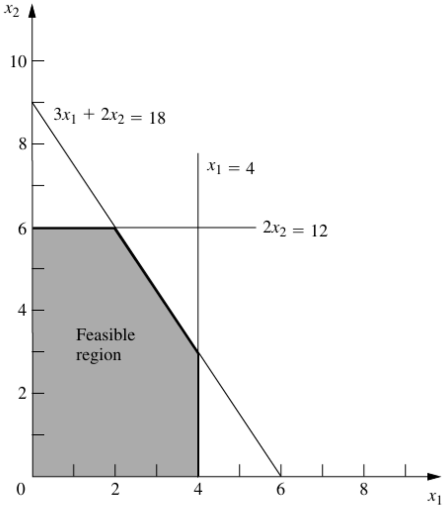
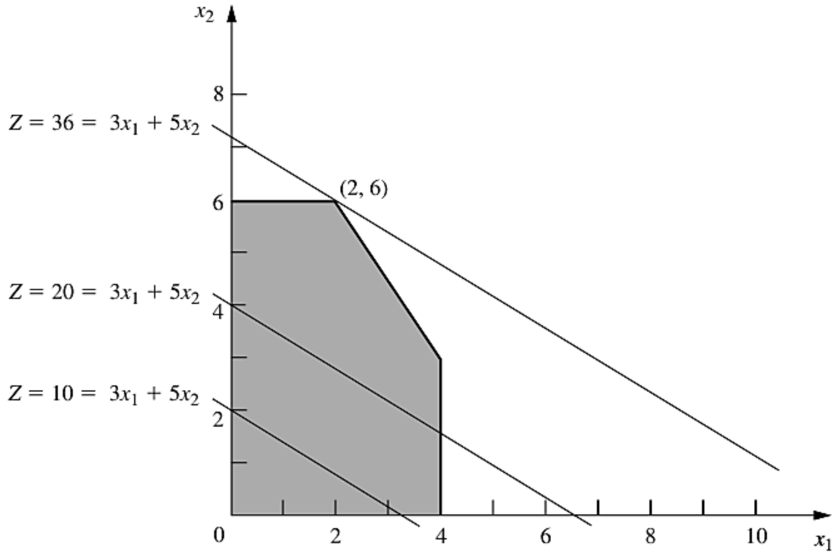


Figure 2.3: Shaded area shows the set of permissible values of (x_1, x_2), called the feasible region.

The feasible region for an LP is the set of all points that satisfies all the LP's constraints and sign restrictions. For a maximization problem, an optimal solution to an LP is a point in the feasible region with the largest objective function value. Similarly, for a minimization problem, an optimal solution is a point in the feasible region with the smallest objective function value.

To find the optimal solution, we need to graph a line on which all points have the same z-value. In a max problem, such a line is called an isoprofit line (in a min problem, an isocost line). The final step is to pick out the point in this feasible region that maximizes the value of $Z = 3x_1 + 5x_2$. To discover how to perform this step efficiently, begin by trial and error.



\

Figure 2.4: The value of (x_1, x_2) that maximizes $3x_1 + 5x_2$ is (2,6)

Now notice in Fig. 3.3 that the two lines just constructed are parallel. This is no co incidence, since any line constructed in this way has the form $Z = 3x_1 + 5x_2$ for the chosen value of Z, which implies that:

$$x_2 = \frac{-3}{5}x_1 + \frac{1}{5}Z$$

This last equation, called the **slope-intercept form** of the objective function, demonstrates that the slope of the line is $\frac{-3}{5}$ (since each unit increase in x_1 changes x_2 by $\frac{-3}{5}$), whereas the intercept of the line with the x_2 axis is $\frac{1}{5}Z$ (since $x_2 = \frac{1}{5}Z$ when x_1 is 0). In MATLAB we do not need to draw any isocost line, rather by simple coding we can get feasible region and an optimal solution.

```
% MATLAB Code for Graphical Method of Linear Programming Problem (Maximization)
% Maximize z = 3*x1 + 5*x2
% Subject to:
% x1 <= 4
% 2x2 <= 12
% 3x1+2x2 <=18
% x1 >= 0, x2 >= 0



clear; clc;

% Constraint matrix A and vector b (A*[x1; x2] <= b)
A = [1 0;   % x1 <= 4
     0 2;   % 2*x2 <= 12
     3 2];  % 3*x1 + 2*x2 <= 18
b = [4; 12; 18];

% Objective function coefficients
c = [3 5];  % z = 3*x1 + 5*x2

disp('Constraint matrix A:');
disp(A);
disp('Right-hand side b:');
disp(b);
disp('Objective coefficients c:');
disp(c);

% Find all intersection points of constraint lines (including axes)
points = [0 0];  % Origin is always a corner point
disp('Initial points (origin):');
disp(points);

% Intersections between pairs of constraints
n = size(A,1);
fprintf('Number of constraints n: %d\n', n);

for i = 1:n
    for j = i+1:n
        fprintf('\nProcessing pair: constraints %d and %d\n', i, j);
        Aij = A([i j], :);
        bij = b([i j]);
        disp('Submatrix Aij:');
        disp(Aij);
        disp('Subvector bij:');
        disp(bij);
```

```
        if rank(Aij) == 2  % Lines are not parallel
            disp('Rank == 2: Lines intersect');
            x = Aij \ bij;
            disp('Intersection point x:');
            disp(x);

            if all(x >= 0) && all(A*x <= b)  % Feasible and non-negative
                disp('Point is feasible and non-negative');
                points = [points; x'];
                disp('Updated points:');
                disp(points);
            else
                disp('Point is not feasible or negative - discarded');
            end
        else
            disp('Rank < 2: Lines parallel - skipped');
        end
    end
end

% Add axis intercepts (where one variable is zero)
disp('Adding axis intercepts:');
for i = 1:n
    % x2 = 0 intercept (x1 = b_i / A_{i1} if A_{i1} > 0)
    if A(i,1) > 0
        x1 = b(i) / A(i,1);
        pt = [x1; 0];
        fprintf('Intercept for constraint %d on x1-axis: [%f, %f]\n', i, pt(1), pt(2));
        if all(A*pt <= b )
            disp('Feasible - adding');
            points = [points; pt'];
        else
            disp('Not feasible - discarded');
        end
    end
    % x1 = 0 intercept (x2 = b_i / A_{i2} if A_{i2} > 0)
    if A(i,2) > 0
        x2 = b(i) / A(i,2);
        pt = [0; x2];
        fprintf('Intercept for constraint %d on x2-axis: [%f, %f]\n', i, pt(1), pt(2));
        if all(A*pt <= b)
            disp('Feasible - adding');
            points = [points; pt'];
        else
            disp('Not feasible - discarded');
        end
```

```
    end
end

% Remove duplicates and sort for convex hull
points = unique(points, 'rows');
disp('Unique points after removing duplicates:');
disp(points);

% Evaluate objective function at corner points
z_values = points * c';
disp('Objective values z at each point:');
for k = 1:size(points,1)
    pt = points(k,:);
    z = z_values(k);
    fprintf('Point [%f, %f]: z = %f\n', pt(1), pt(2), z);
end

[max_z, idx] = max(z_values);
optimal_point = points(idx, :);

fprintf('Maximum z: %f\n', max_z);
fprintf('Optimal point: [%f, %f]\n', optimal_point(1), optimal_point(2));

% Plot the feasible region (shaded)
figure;
K = convhull(points(:,1), points(:,2));
fill(points(K,1), points(K,2), 'c', 'FaceAlpha', 0.4, 'EdgeColor', 'k');
hold on;

% Plot constraint lines
x1_range = linspace(0, max(points(:,1)) + 1, 100);
for i = 1:n
    if A(i,2) ~= 0
        x2_line = (b(i) - A(i,1)*x1_range) / A(i,2);
    else
        x2_line = zeros(size(x1_range));  % Vertical line case (not here)
    end
    plot(x1_range, max(x2_line, 0), 'b--', 'LineWidth', 1.5);
end

% Plot corner points
plot(points(:,1), points(:,2), 'ko', 'MarkerFaceColor', 'k');

% Mark optimal point
plot(optimal_point(1), optimal_point(2), 'rp', 'MarkerSize', 20);
```

```
% Annotations
xlabel('x_1');
ylabel('x_2');
title('Graphical Method for LPP - Feasible Region Shaded');
grid on;
axis equal;
xlim([0 max(points(:,1)) + 1]);
ylim([0 max(points(:,2)) + 1]);

text(optimal_point(1)+0.2, optimal_point(2), ...
   sprintf('Optimal Point: (%.2f, %.2f)\nMax z = %.2f', optimal_point(1), optimal_point(2),
max_z), ...
   'FontSize', 10, 'BackgroundColor', 'w');

legend({'Feasible Region', 'Constraints', 'Corner Points', 'Optimal Solution'}, 'Location', 'best');
```

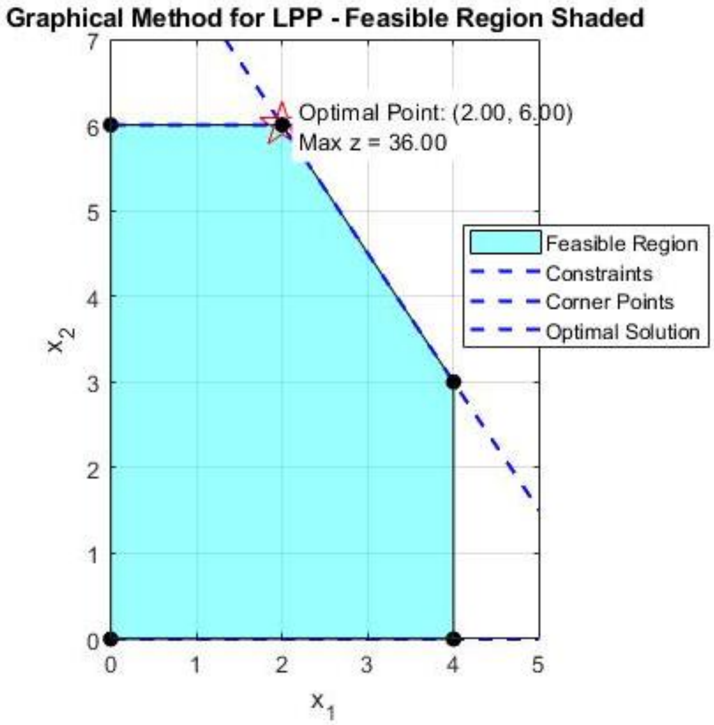


Figure 2.5: Graphical Solution Using MATLAB

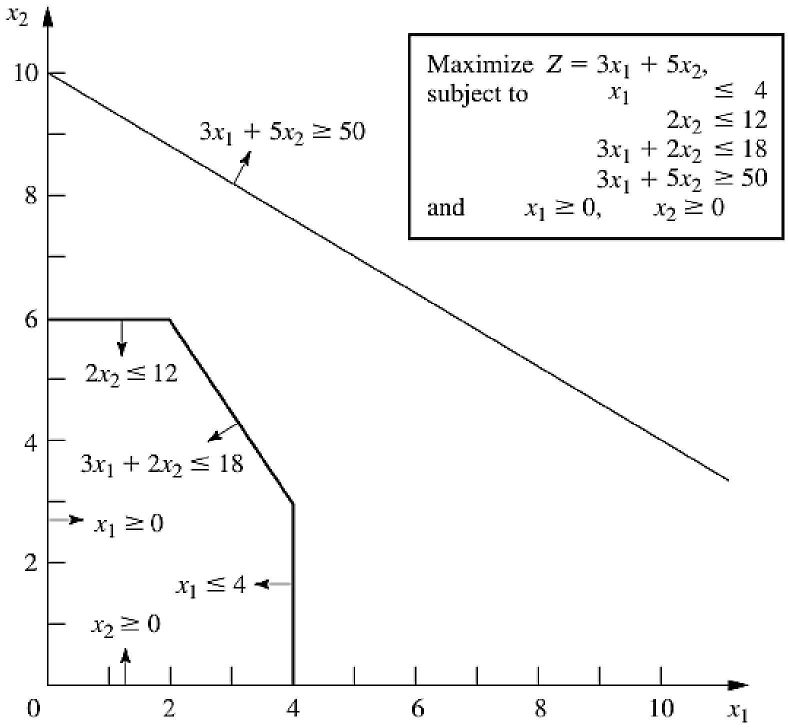


Figure 2.6: The Wyndor Glass Co. problem would have no feasible solution if the constraint $3x_1 + 5x_2 \geq 50$ were added to the problem

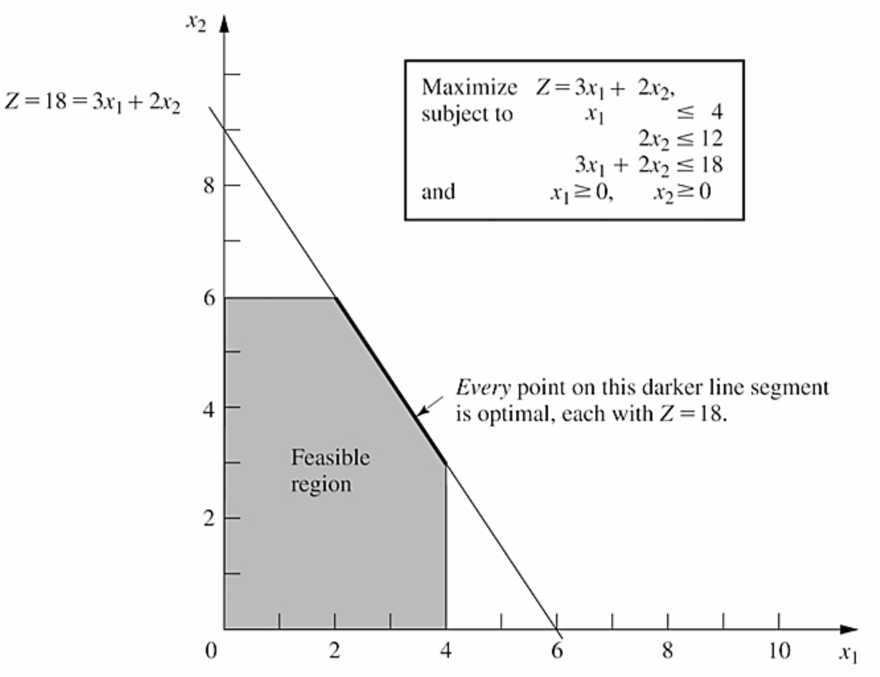


Figure 2.7: The Wyndor Glass Co. problem would have multiple optimal solutions if the objective function were changed to $Z = 3x_1 + 2x_2$.

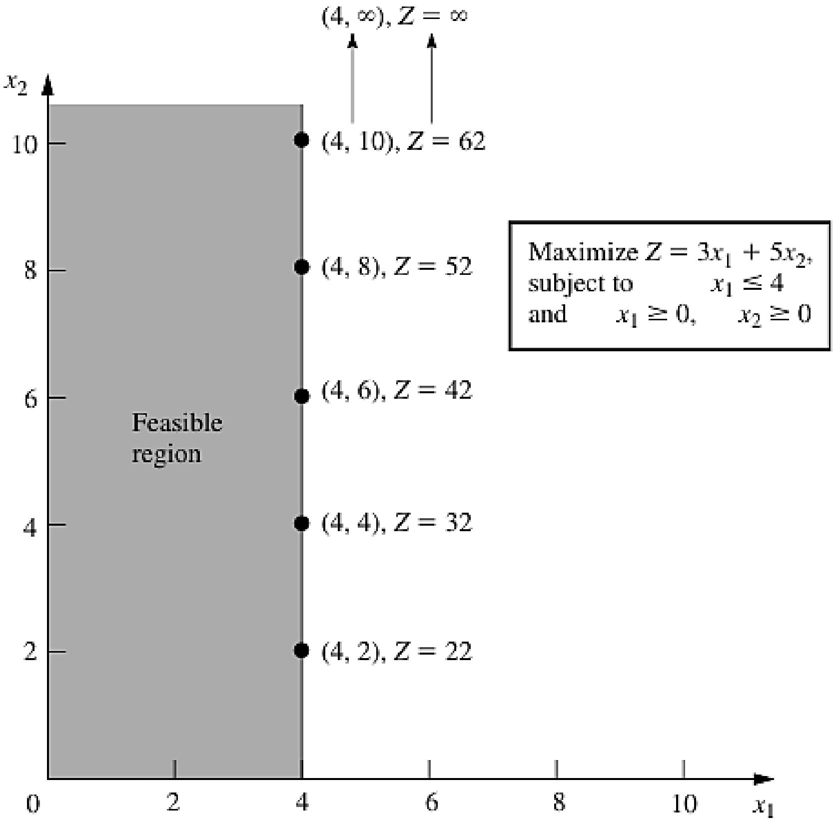


Figure 2.8: The Wyndor Glass Co. problem would have no optimal solutions if the only functional constraint were $x_1 \leq 4$, because x_2 then could be increased indefinitely in the feasible region without ever reaching the maximum value of $Z = 3x_1 + 5x_2$

2.5 LP Assumptions

- **Proportionality assumption**: The contribution of each activity to the value of the objective function Z (and LHS of functional constraint) is proportional to the level of the activity x_j.

Table 2.4: Examples of satisfying or violating proportionality

	Profit from Product 1 ($000 per Week)			
		Proportionality Violated		
x_1	**Proportionality Satisfied**	**Case 1**	**Case 2**	**Case 3**
0	0	0	0	0
1	3	2	3	3
2	6	5	7	5
3	9	8	12	6
4	12	11	18	6

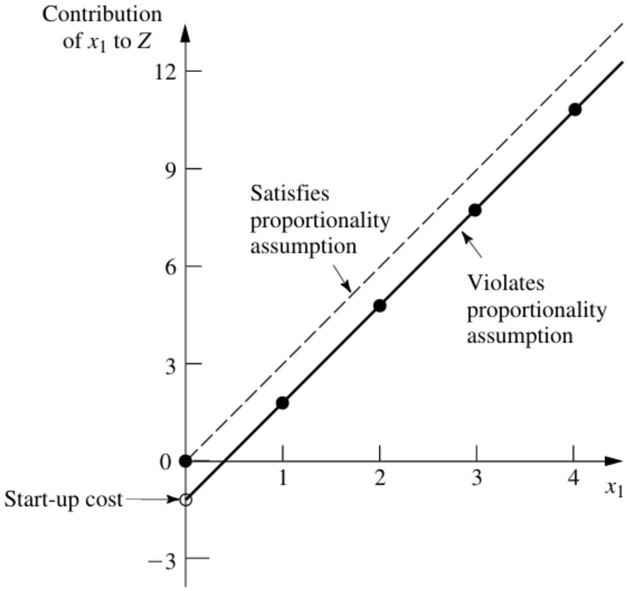


Figure 2.9: The solid curve violates the proportionality assumption because of the start-up cost that is incurred when x_1 is increased from 0.

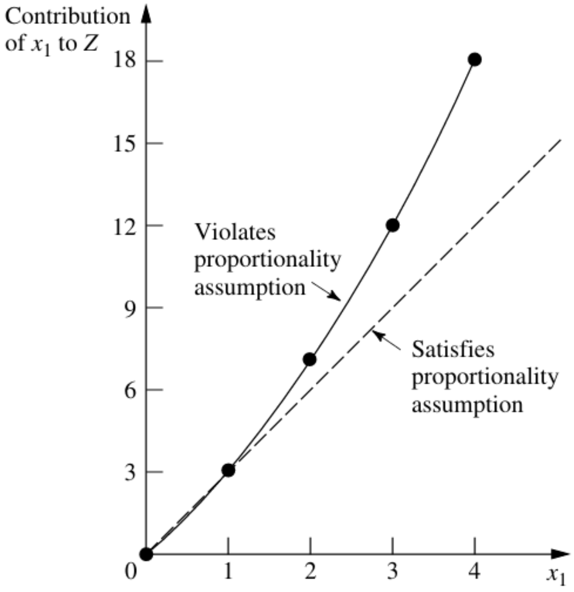


Figure 2.10: The solid curve violates the proportionality assumption because its slope (the marginal return from product 1 keeps increasing as x_1 is increased.

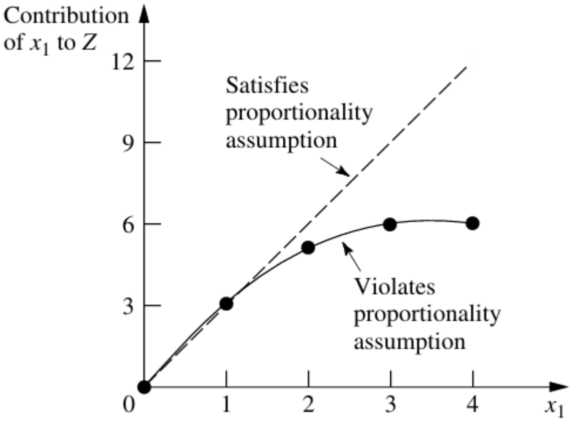


Figure 2.11: The solid curve violates the proportionality assumption because its slope (the marginal return from product 1 keeps decreasing as x_1 is increased.

- **Additivity assumption:** Every function in a linear programming model is the sum of the individual contributions of the respective activities.

Table 2.5: Examples of satisfying or violating additivity for the objective function

	Value of Z		
		Additivity Violated	
(x_1, x_2)	**Additivity Satisfied**	**Case 1**	**Case 2**
(1, 0)	3	3	3
(0, 1)	5	5	5
(1, 1)	8	9	7

Table 2.6: Examples of satisfying or violating additivity for a functional constraint

	Amount of Resource Used		
		Additivity Violated	
(x_1, x_2)	**Additivity Satisfied**	**Case 3**	**Case 4**
(2, 0)	6	6	6
(0, 3)	6	6	6
(2, 3)	12	15	10.8

- **Divisibility assumption**: Decision variables in a linear programming model are allowed to have any values, including non-integer values, that satisfy the functional and non-negativity constraints. In certain situations, the divisibility assumption does not hold because some of or all the decision variables must be restricted to integer values. Mathematical models with this restriction are called integer programming models,

For example, in the WYNDOR problem, the Divisibility Assumption implies that it is acceptable to produce 1.5 product 1 or 1.63 product 2. Because WYNDOR cannot actually produce a fractional number of products, the Divisibility Assumption is not satisfied in the WYNDOR problem.

- **Certainty assumption**: The value assigned to each parameter (objective function coefficient, right hand side, and functional constraint coefficient) of a linear programming model is assumed to be a known constant.

2.6 The Graphical Solution of Minimization Problems Using MATLAB

Dorian Auto manufactures luxury cars and trucks. The company believes that its most likely customers are high-income women and men. To reach these groups, Dorian Auto has embarked on an ambitious TV advertising campaign and has decided to purchase 1-minute commercial spots on two types of programs: comedy shows and football games. Each comedy commercial is seen by 7 million high-income women and 2 million high income men. Each football commercial is seen by 2 million high-income women and 12 million high-income men. A 1-minute comedy ad costs $50,000, and a 1-minute football ad costs $100,000. Dorian would like the commercials to be seen by at least 28 million high-income women and 24 million high-income men. Use linear programming to deter mine how Dorian Auto can meet its advertising requirements at minimum cost.

Solution:

Decision Variable:

$$x_1 = number\ of\ 1 - minute\ comedy\ ads\ purchased$$

$$x_2 = number\ of\ 1 - minute\ football\ ads\ purchased$$

Objective Function:

$$\text{Minimize } Z = 50x_1 + 100x_2$$

Constraint 1: Commercials must reach at least 28 million high-income women.

$$7x_1 + 2x_2 \geq 28$$

Constraint 2: Commercials must reach at least 24 million high-income men.

$$2x_1 + 12x_2 \geq 24$$

Non-negativity constraint:

$$x_1 \geq 0\,,x_2 \geq 0$$

```
% MATLAB Code for Graphical Method of Linear Programming Problem (Minimization)
% Minimize z = 50*x + 100*y (cost in thousands)
% Subject to:
% 7x + 2y >= 28  (high-income women, millions)
% 2x + 12y >= 24 (high-income men, millions)
% x >= 0, y >= 0

clear; clc;

% Original constraints (A_orig * x >= b_orig)
A_orig = [7 2; 2 12];
b_orig = [28; 24];

% Flip for standard <= form: A * x <= b
A = -A_orig;
b = -b_orig;

% Add artificial bounds to make feasible region bounded for shading
bound = 15;  % Large enough bound
A = [A; 1 0; 0 1];
b = [b; bound; bound];

% Objective function coefficients (minimize c' * x)
c = [50 100];

% Find all intersection points (corner points)
points = [];  % Do not assume origin is feasible

% Intersections between pairs of constraints
n = size(A,1);
for i = 1:n
    for j = i+1:n
        Aij = A([i j], :);
        bij = b([i j]);
        if rank(Aij) == 2
            x = Aij \ bij;
            if all(x >= 0) && all(A*x <= b + 1e-8)
                points = [points; x'];
```

```
            end
        end
    end
end

% Add axis intercepts if feasible
for i = 1:n
    % x2 = 0 intercept
    if A(i,1) ~= 0
        x1 = b(i)/A(i,1);
        if x1 >= 0
            pt = [x1; 0];
            if all(A*pt <= b + 1e-8)
                points = [points; pt'];
            end
        end
    end
    % x1 = 0 intercept
    if A(i,2) ~= 0
        x2 = b(i)/A(i,2);
        if x2 >= 0
            pt = [0; x2];
            if all(A*pt <= b + 1e-8)
                points = [points; pt'];
            end
        end
    end
end

% Remove duplicates
points = unique(points, 'rows');

% Evaluate objective at corner points
z_values = points * c';
[min_z, idx] = min(z_values);
optimal_point = points(idx, :);

% Plot the feasible region (shaded)
figure; hold on;
K = convhull(points(:,1), points(:,2));
fill(points(K,1), points(K,2), 'c', 'FaceAlpha', 0.4, 'EdgeColor', 'k');

% Plot original constraint lines only (not bounds)
x_range = linspace(0, bound, 100);
n_orig = size(A_orig,1);
for i = 1:n_orig
```

```
    if A_orig(i,2) ~= 0
        y_line = (b_orig(i) - A_orig(i,1)*x_range) / A_orig(i,2);
    else
        y_line = zeros(size(x_range));
    end
    plot(x_range, y_line, 'b--', 'LineWidth', 1.5);
end

% Plot corner points
plot(points(:,1), points(:,2), 'ko', 'MarkerFaceColor', 'k');

% Mark optimal point
plot(optimal_point(1), optimal_point(2), 'rp', 'MarkerSize', 12, 'MarkerFaceColor', 'r');

% Annotations
xlabel('x (Comedy Ads)');
ylabel('y (Football Ads)');
title('Graphical Method for Dorian Auto Advertising - Minimum Cost');
grid on;
axis equal;
xlim([0 bound]);
ylim([0 bound]);
text(optimal_point(1)+0.2, optimal_point(2), ...
    sprintf('Optimal: (%.1f, %.1f)\nMin Cost = $%.0fk', optimal_point(1), optimal_point(2),
min_z), ...
    'FontSize', 10, 'BackgroundColor', 'w');
legend({'Feasible Region (Bounded View)', 'Constraints', 'Corner Points', 'Optimal Solution'},
'Location', 'best');
hold off;
```

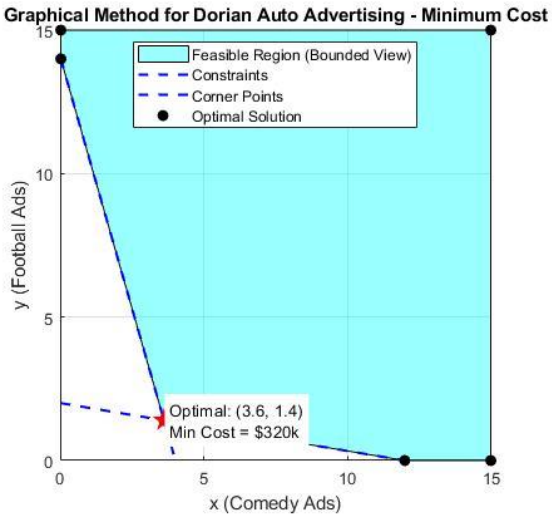


Figure 2.12: Graphical Solution Using MATLAB

The Dorian problem has a convex feasible region, but the feasible region for Dorian, contains points for which the value of at least one variable can assume arbitrarily large values. Such a feasible region is called an **unbounded feasible region.** Thus, the optimal z-value is $Z = 50 * 3.6 + 100 * 1.4 = 320$. This means the optimal z value is $320,000.

Does the Dorian model meet the four assumptions of linear programming?

- After, say, 500 auto commercials have been aired, most people have probably seen one, so it does little good to air more commercials. Thus, the Proportionality Assumption is violated.
- In reality, many of the same people will see a Dorian comedy commercial and a Dorian football commercial. We are double-counting such people, and this creates an inaccurate picture. This violates the Additivity Assumption.
- If only 1-minute commercials are available, then it is unreasonable to say that Dorian should buy 3.6 comedy commercials and 1.4 football commercials, so the Divisibility Assumption is violated.
- There is no way to know with certainty how many viewers are added by each type of commercial, the Certainty Assumption is also violated.

3. Practice Problem

3.1 Mina's diet requires that all the food she eats come from one of the four "basic food groups" (chocolate cake, ice cream, soda, and cheesecake). At present, the following four foods are available for consumption: brownies, chocolate ice cream, cola, and pineapple cheese cake. Each brownie costs 50¢, each scoop of chocolate ice cream costs 20¢, each bottle of cola costs 30¢, and each piece of pineapple cheesecake costs 80¢. Each day, I must in gest at least 500 calories, 6 oz of chocolate, 10 oz of sugar, and 8 oz of fat. The nutritional content per unit of each food is shown in Table 2. Formulate a linear programming model that can be used to satisfy Mina's daily nutritional requirements at minimum cost. (Use MATLAB for graphical solution)

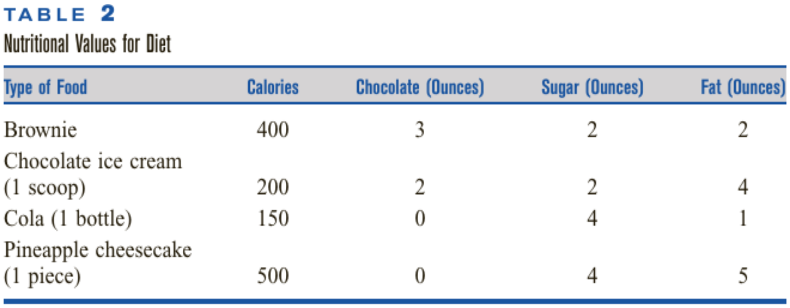


TABLE 2

Nutritional Values for Diet

Type of Food	Calories	Chocolate (Ounces)	Sugar (Ounces)	Fat (Ounces)
Brownie	400	3	2	2
Chocolate ice cream (1 scoop)	200	2	2	4
Cola (1 bottle)	150	0	4	1
Pineapple cheesecake (1 piece)	500	0	4	5

3.2 There are three factories on the Momiss River (1, 2, and 3). Each emits two types of pollutants (1 and 2) into the river. If the waste from each factory is processed, the pollution in the river can be reduced. It costs $15 to process a ton of factory 1 waste, and each ton processed reduces the amount of pollutant 1 by 0.10 ton and the amount of pollutant 2 by 0.45 ton. It costs $10 to process a ton of factory 2 waste, and each ton processed will reduce the amount of pollutant 1 by 0.20 ton and the amount of pollutant 2 by 0.25 ton. It costs $20 to process a ton of factory 3 waste, and each ton processed will reduce the amount of pollutant 1 by 0.40 ton and the amount of pollutant 2 by 0.30 ton. The state wants to reduce the amount of pollutant 1 in the river by at least 30 tons and the amount of pollutant 2 in the river by at least 40 tons.

(a) Formulate an LP that will minimize the cost of reducing pollution by the desired amounts.

(b) Do you think that the LP assumptions (Proportionality, Additivity, Divisibility, and Certainty) are reasonable for this problem?

(c) Use MATLAB for graphical solution

3.3 The World Light Company produces two light fixtures (products 1 and 2) that require both metal frame parts and electrical components. Management wants to determine how many units of each product to produce so as to maximize profit. For each unit of product 1, 1 unit of frame parts and 2 units of electrical components are required. For each unit of product 2, 3 units of frame parts and 2 units of electrical components are required. The company has 200 units of frame parts and 300 units of electrical com ponents. Each unit of product 1 gives a profit of $1, and each unit of product 2, up to 60 units, gives a profit of $2. Any excess over 60 units of product 2 brings no profit, so such an excess has been ruled out.

Formulate a linear programming model for this problem. Use MATLAB for graphical solution to solve this model. What is the resulting total profit?

Experiment-2: Solving Linear Programming Problems: Simplex Method

1. Objective

By the end of this lab, students will be able to:

- Solve LP problems in **LINDO**.
- Understand how LINDO handles artificial variables, degeneracy, and alternative optima.

2. Background

A Linear Programming (LP) problem has **infinitely many feasible points**, but only a **finite number of corner points (extreme points)**. A fundamental theorem of LP states:

If an LP has an optimal solution, at least one optimal solution occurs at a corner point of the feasible region.

The **Simplex Method** is a systematic algorithm that:

- Starts at one corner point (Basic Feasible Solution),
- Moves from corner to corner,
- ***Improves*** the objective value at each step,
- Stops when no further improvement is possible.

Decision Variables: Variables representing choices to be made.

- Denoted as: $X_1, X_2, \ldots, X_n$
- Example: number of products produced.

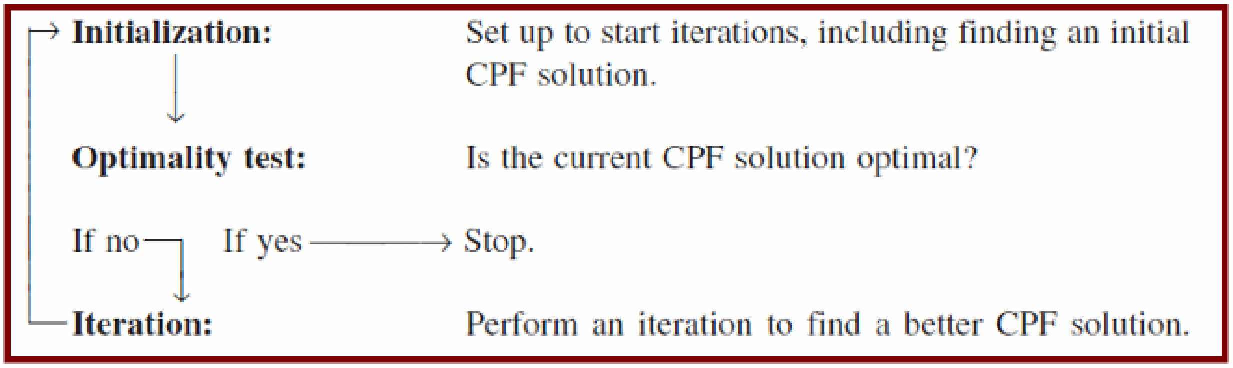
Objective Function: The function to be **maximized or minimized**.

$$Z = c_1X_1 + c_2X_2 + \cdots + c_nX_n$$

Constraints: Linear equations or inequalities restricting the decision variables. The dimension of the problem determines the number of constraints that intersect at each corner

$$a_1X_1 + a_2X_2 \leq b_1$$

Feasible Solution: Any solution that satisfies **all constraints and non-negativity conditions**. Simplex focuses only on CPF solutions, a finite set.



Initialization: Set up to start iterations, including finding an initial CPF solution.

Optimality test: Is the current CPF solution optimal?

If no — If yes → Stop.

Iteration: Perform an iteration to find a better CPF solution.

Basic Solution: A solution obtained by:

- Selecting m variables (number of constraints),
- Setting the remaining variables to zero,
- Solving the resulting equations.

Basic Feasible Solution (BFS): A **basic solution that is also feasible** (all variables ≥ 0). Each BFS corresponds to a **corner point** of the feasible region.

Slack Variable: Introduced to convert a ≤ **constraint** into an equation.

$$aX \leq b$$

$$\Longrightarrow aX + s = b$$

$$S \geq 0$$

Interpretation: **unused resource**

Negative RHS: To eliminate the negative value in the RHS, multiply through by -1. This operation reverses the sign of the inequality**.**

Surplus Variable: Introduced in a ≥ **constraint**, subtracted from LHS.

$$aX \geq b$$

$$\Longrightarrow aX - E = b$$

Artificial Variable: Introduced when:

- Constraint is = **or** ≥
- Needed only to obtain an initial BFS

Artificial variables **must be eliminated** from the final solution.

Simplex Tableau: A tabular representation of:

- Constraints
- Objective function
- Coefficients of all variables

This is the working structure of the simplex algorithm.

Optimality test: If a CPF solution has no adjacent CPF solution that is better (in terms of the value of the objective function) then it must be an optimal solution.

Entering Variable: A non-basic variable selected to **enter the basis** to improve the objective.

Leaving Variable: A basic variable selected to **leave the basis** to maintain feasibility.

Pivot Element: The element at the intersection of:

- Entering variable column
- Leaving variable row

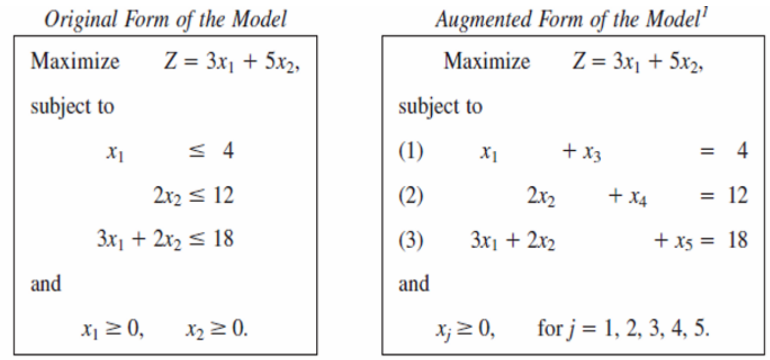
Used for Gauss–Jordan elimination.

Gauss-Jordan Elimination

- Used to update tableau during pivot.
- Steps:
  1. Select **pivot element**
  2. **Normalize pivot row:** divide row by pivot element.
  3. **Eliminate pivot column entries** in all other rows by making all other entries in the pivot column zero:
     New Pivot Row = Current Pivot Row / Pivot Element
     New Row = (Current Row) – (Respective Pivot Column Coefficient)*(New Pivot Row)

This ensures new variable becomes basic and old variable becomes non-basic.

<u>Example 1:</u>



Original Form of the Model

Maximize $Z = 3x_1 + 5x_2,$

subject to

$$x_1 \le 4$$

$$2x_2 \le 12$$

$$3x_1 + 2x_2 \le 18$$

and

$$x_1 \ge 0, \quad x_2 \ge 0.$$

Augmented Form of the Model[1]

Maximize $Z = 3x_1 + 5x_2,$

subject to

(1) $x_1 + x_3 = 4$

(2) $2x_2 + x_4 = 12$

(3) $3x_1 + 2x_2 + x_5 = 18$

and

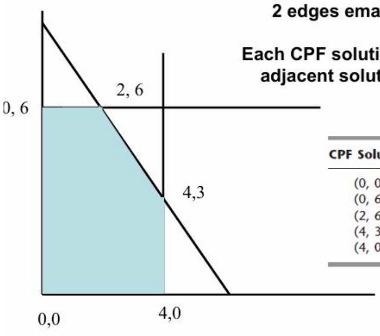
$$x_j \ge 0, \quad \text{for } j = 1, 2, 3, 4, 5.$$

Wyndor Problem

There are 5 edges.

From each CPF solution,
2 edges emanate.

Each CPF solution has 2
adjacent solutions.



CPF Solution	Its Adjacent CPF Solutions
(0, 0)	(0, 6) and (4, 0)
(0, 6)	(2, 6) and (0, 0)
(2, 6)	(4, 3) and (0, 6)
(4, 3)	(4, 0) and (2, 6)
(4, 0)	(0, 0) and (4, 3)

Maximize Z,

subject to

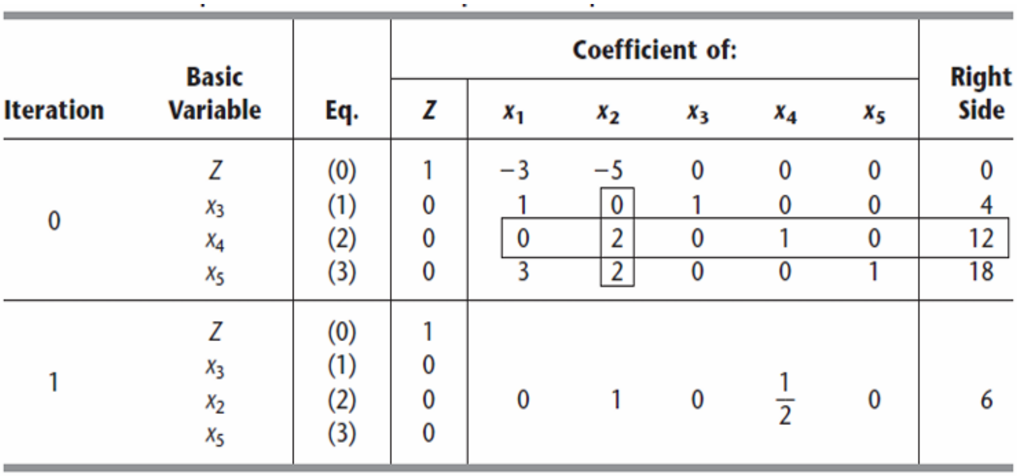
$$\begin{array}{llr} (0) & Z - 3x_1 - 5x_2 & = 0 \\ (1) & x_1 + x_3 & = 4 \\ (2) & 2x_2 + x_4 & = 12 \\ (3) & 3x_1 + 2x_2 + x_5 & = 18 \end{array}$$

and

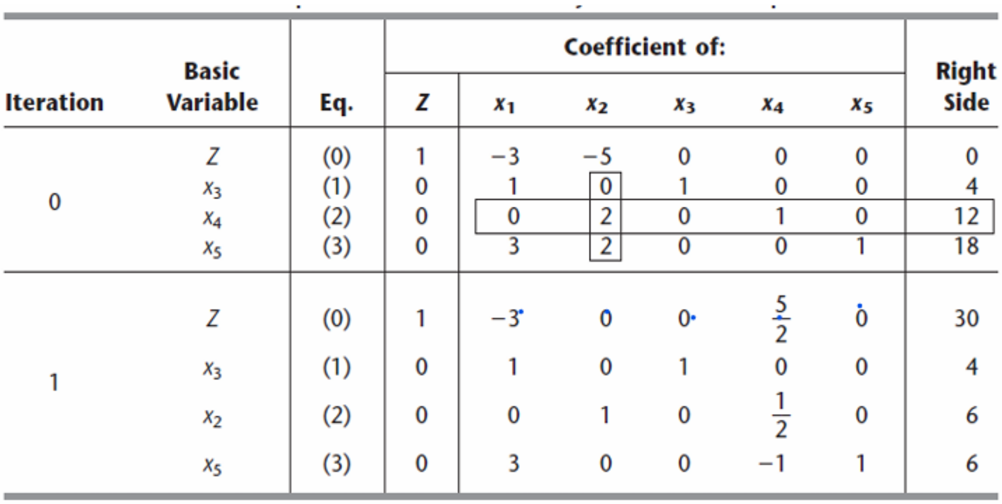
$$x_j \geq 0, \quad \text{for } j = 1, 2, \ldots, 5.$$

(a) Algebraic Form	(b) Tabular Form								
	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	x_5	
(0) $Z - 3x_1 - 5x_2 = 0$	Z	(0)	1	−3	−5	0	0	0	0
(1) $x_1 + x_3 = 4$	x_3	(1)	0	1	0	1	0	0	4
(2) $2x_2 + x_4 = 12$	x_4	(2)	0	0	2	0	1	0	12
(3) $3x_1 + 2x_2 + x_5 = 18$	x_5	(3)	0	3	2	0	0	1	18

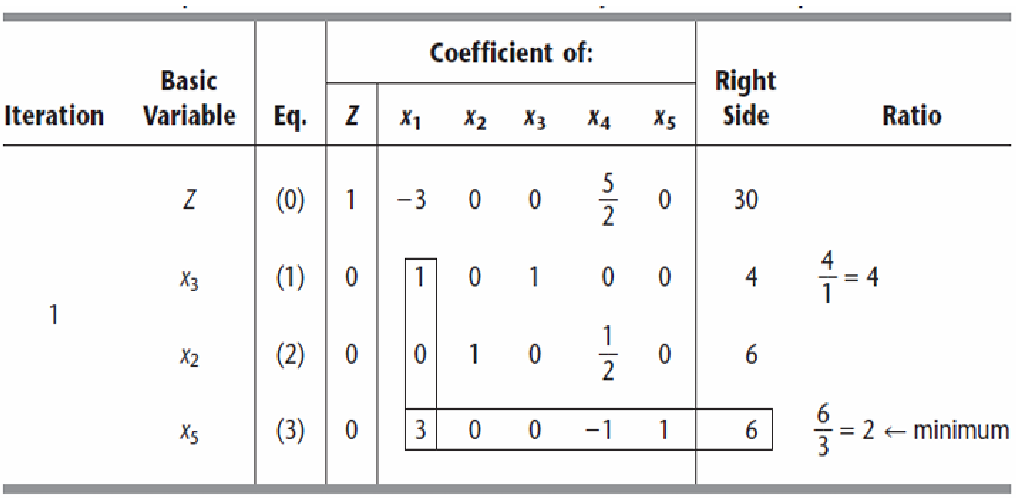
Basic Variable	Eq.	Coefficient of:						Right Side	Ratio
		Z	x_1	x_2	x_3	x_4	x_5		
Z	(0)	1	−3	−5	0	0	0	0	
x_3	(1)	0	1	$\boxed{0}$	1	0	0	4	
x_4	(2)	0	0	$\boxed{2}$	0	1	0	12	$12 \rightarrow \frac{12}{2} = 6 \leftarrow$ minimum
x_5	(3)	0	3	$\boxed{2}$	0	0	1	18	$18 \rightarrow \frac{18}{2} = 9$



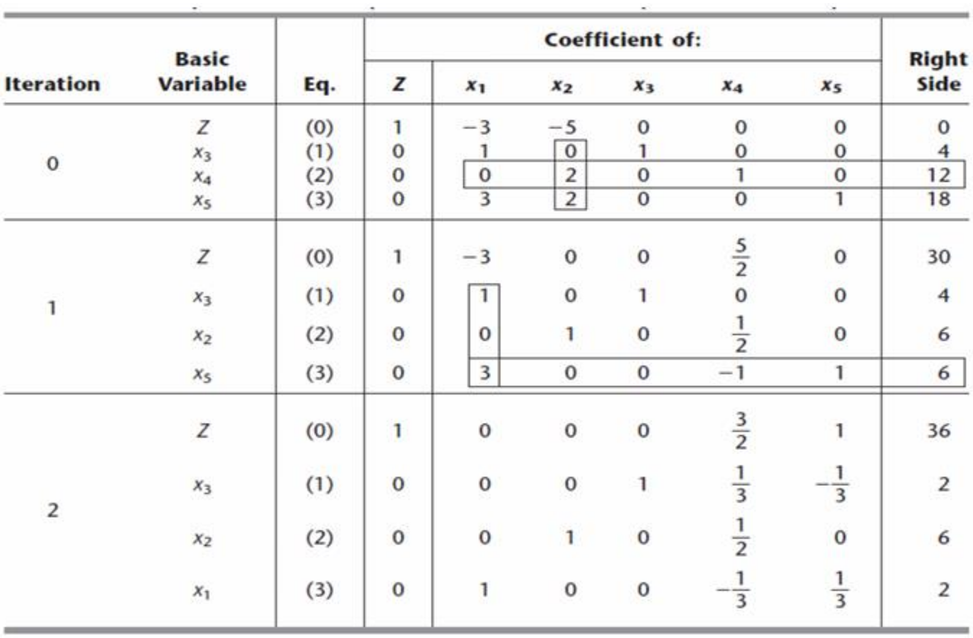
Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	x_5	
0	Z	(0)	1	−3	−5	0	0	0	0
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	x_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1						
	x_3	(1)	0						
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0						



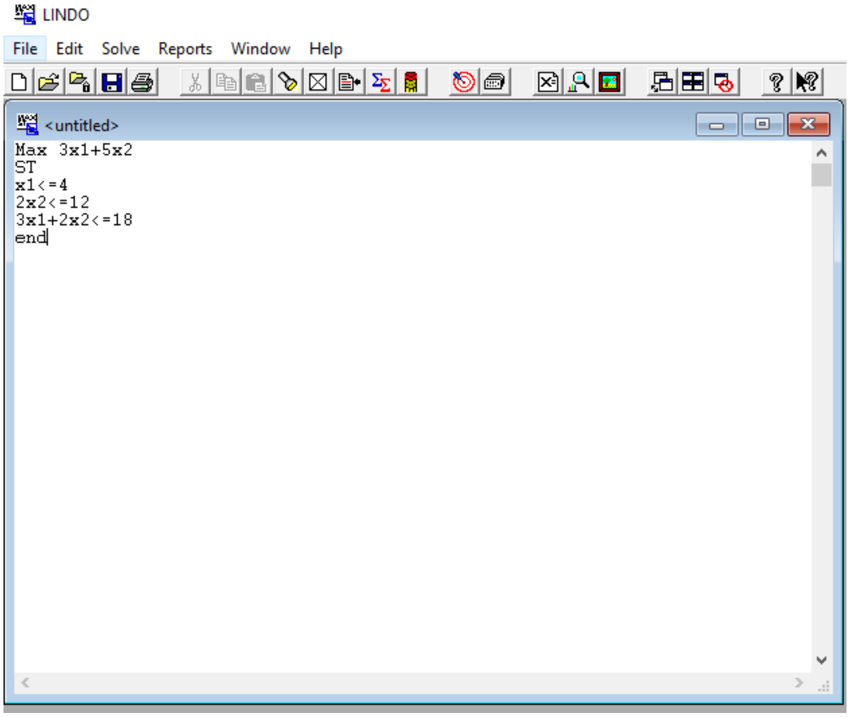
Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	x_5	
0	Z	(0)	1	−3	−5	0	0	0	0
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	x_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	−3	0	0	$\frac{5}{2}$	0	30
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0	3	0	0	−1	1	6



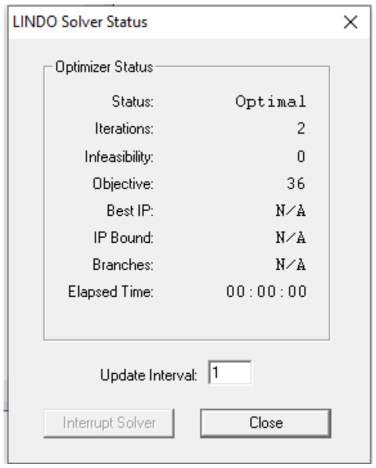
Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	Ratio
			Z	x_1	x_2	x_3	x_4	x_5		
1	Z	(0)	1	−3	0	0	$\frac{5}{2}$	0	30	
	x_3	(1)	0	1	0	1	0	0	4	$\frac{4}{1} = 4$
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6	
	x_5	(3)	0	3	0	0	−1	1	6	$\frac{6}{3} = 2 \leftarrow$ minimum



Iteration	Basic Variable	Eq.	Coefficient of: Z	x_1	x_2	x_3	x_4	x_5	Right Side
0	Z	(0)	1	−3	−5	0	0	0	0
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	x_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	−3	0	0	$\frac{5}{2}$	0	30
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0	3	0	0	−1	1	6
2	Z	(0)	1	0	0	0	$\frac{3}{2}$	1	36
	x_3	(1)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_1	(3)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2

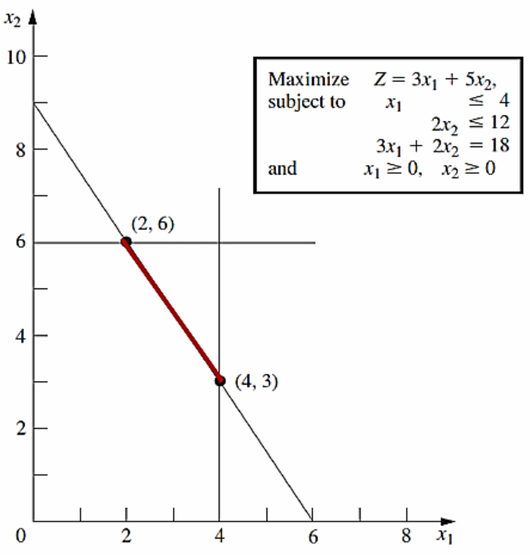


```
Max 3x1+5x2
ST
x1<=4
2x2<=12
3x1+2x2<=18
end
```



Status: Optimal
Iterations: 2
Infeasibility: 0
Objective: 36
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00
Update Interval: 1

Example 2:



$$\begin{array}{llll} \text{Maximize} & Z = 3x_1 + 5x_2, & & \\ \text{subject to} & x_1 & & \le 4 \\ & & 2x_2 & \le 12 \\ & 3x_1 + & 2x_2 & = 18 \\ \text{and} & x_1 \ge 0, & x_2 \ge 0 & \end{array}$$

$$\begin{array}{lllll} Z - 3X1 - 5X2 & & & & = 0 \\ X1 & + X3 & & & = 4 \\ 2X2 & & + X4 & & = 12 \\ 3X1 + 2X2 & & & & = 18 \end{array}$$

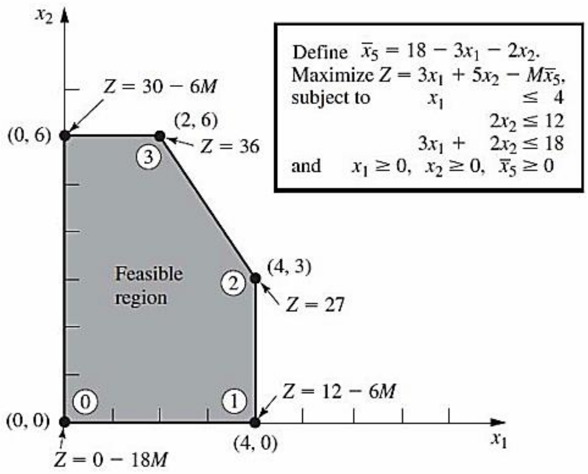
There is no obvious BFS because there is no slack in the third constraint.

The Real Problem

$$\begin{array}{ll} \text{Maximize} & Z = 3x_1 + 5x_2, \\ \text{subject to} & \\ & x_1 \le 4 \\ & 2x_2 \le 12 \\ & 3x_1 + 2x_2 = 18 \\ \text{and} & \\ & x_1 \ge 0, \quad x_2 \ge 0. \end{array}$$

The Artificial Problem

$$\begin{array}{ll} \text{Define} & \bar{x}_5 = 18 - 3x_1 - 2x_2. \\ \text{Maximize} & Z = 3x_1 + 5x_2 - M\bar{x}_5, \\ \text{subject to} & \\ & x_1 \le 4 \\ & 2x_2 \le 12 \\ & 3x_1 + 2x_2 \le 18 \\ (\text{so} & 3x_1 + 2x_2 + \bar{x}_5 = 18) \\ \text{and} & \\ & x_1 \ge 0, \quad x_2 \ge 0, \quad \bar{x}_5 \ge 0. \end{array}$$

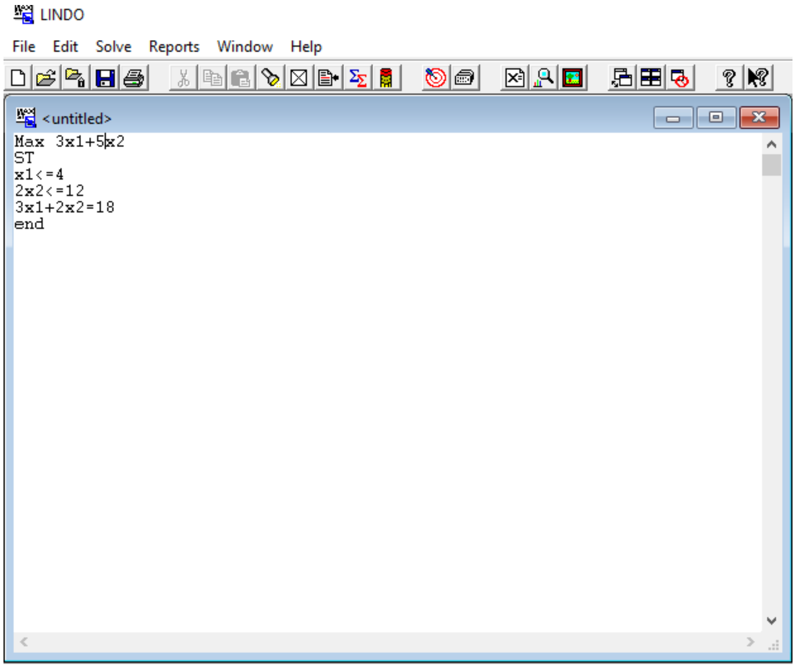


$$\begin{array}{llll} \text{Define} & \bar{x}_5 = 18 - 3x_1 - 2x_2. & & \\ \text{Maximize} & Z = 3x_1 + 5x_2 - M\bar{x}_5, & & \\ \text{subject to} & x_1 & & \le 4 \\ & & 2x_2 & \le 12 \\ & 3x_1 + & 2x_2 & \le 18 \\ \text{and} & x_1 \ge 0, & x_2 \ge 0, & \bar{x}_5 \ge 0 \end{array}$$

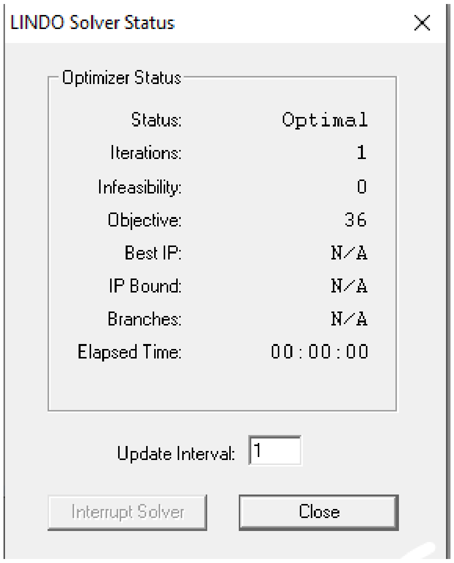
$$\begin{array}{llllll} (0) & Z - 3x_1 - 5x_2 & & & + M\bar{x}_5 = & 0 \\ (1) & x_1 & + x_3 & & = & 4 \\ (2) & 2x_2 & & + x_4 & = & 12 \\ (3) & 3x_1 + 2x_2 & & & + \quad \bar{x}_5 = & 18 \end{array}$$

$$\begin{array}{rr} & Z - 3x_1 - 5x_2 + M\bar{x}_5 = \quad 0 \\ & -M(3x_1 + 2x_2 + \quad \bar{x}_5 = 18) \\ \hline \text{New (0)} & Z - (3M + 3)x_1 - (2M + 5)x_2 = -18M. \end{array}$$

Iteration	Basic Variable	Eq.	Coefficient of:						Right Side
			Z	x_1	x_2	x_3	x_4	$\bar{x}_5$	
0	Z	(0)	1	$-3M-3$	$-2M-5$	0	0	0	$-18M$
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	$\bar{x}_5$	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	0	$-2M-5$	$3M+3$	0	0	$-6M+12$
	x_1	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	$\bar{x}_5$	(3)	0	0	2	-3	0	1	6
2	Z	(0)	1	0	0	$-\frac{9}{2}$	0	$M+\frac{5}{2}$	27
	x_1	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	0	3	1	-1	6
	x_2	(3)	0	0	1	$-\frac{3}{2}$	0	$\frac{1}{2}$	3
Extra	Z	(0)	1	0	0	0	$\frac{3}{2}$	$M+1$	36
	x_1	(1)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2
	x_3	(2)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
	x_2	(3)	0	0	1	0	$\frac{1}{2}$	0	6



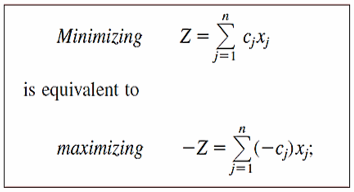
```
Max 3x1+5x2
ST
x1<=4
2x2<=12
3x1+2x2=18
end
```



Status: Optimal
Iterations: 1
Infeasibility: 0
Objective: 36
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00

Update Interval: 1

For minimization problems:

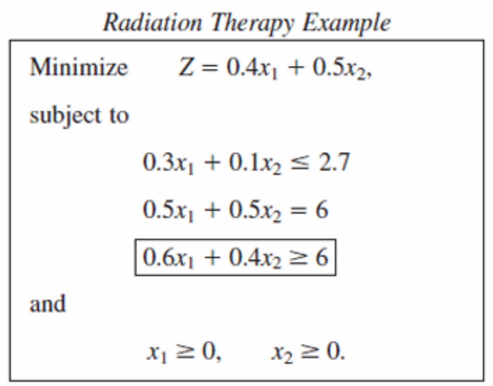


$$\text{Minimizing} \quad Z = \sum_{j=1}^{n} c_j x_j$$

is equivalent to

$$\text{maximizing} \quad -Z = \sum_{j=1}^{n} (-c_j) x_j;$$

Example 3: (Minimization Problem with Big M Method)



Radiation Therapy Example

Minimize $Z = 0.4x_1 + 0.5x_2,$

subject to

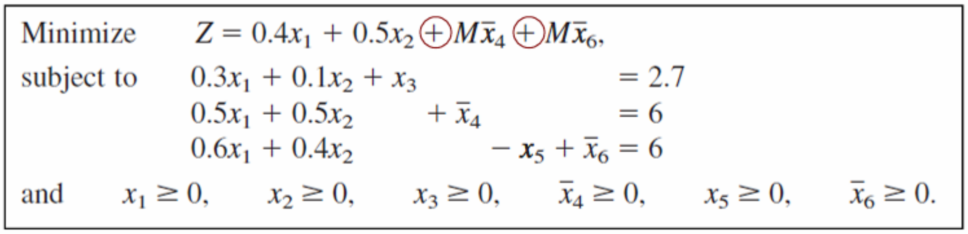
$$0.3x_1 + 0.1x_2 \le 2.7$$

$$0.5x_1 + 0.5x_2 = 6$$

$$\boxed{0.6x_1 + 0.4x_2 \ge 6}$$

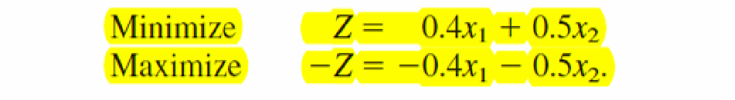
and

$$x_1 \ge 0, \qquad x_2 \ge 0.$$

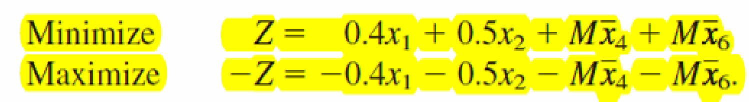


$$\begin{array}{lllll}
\text{Minimize} & Z = 0.4x_1 + 0.5x_2 \oplus M\bar{x}_4 \oplus M\bar{x}_6, & & & \\
\text{subject to} & 0.3x_1 + 0.1x_2 + x_3 & & & = 2.7 \\
 & 0.5x_1 + 0.5x_2 & + \bar{x}_4 & & = 6 \\
 & 0.6x_1 + 0.4x_2 & & - x_5 + \bar{x}_6 & = 6
\end{array}$$

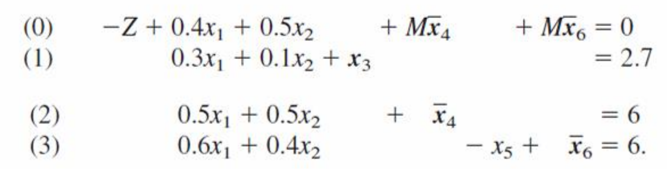
and $x_1 \ge 0, \quad x_2 \ge 0, \quad x_3 \ge 0, \quad \bar{x}_4 \ge 0, \quad x_5 \ge 0, \quad \bar{x}_6 \ge 0.$



$$\begin{array}{ll}
\text{Minimize} & Z = 0.4x_1 + 0.5x_2 \\
\text{Maximize} & -Z = -0.4x_1 - 0.5x_2.
\end{array}$$

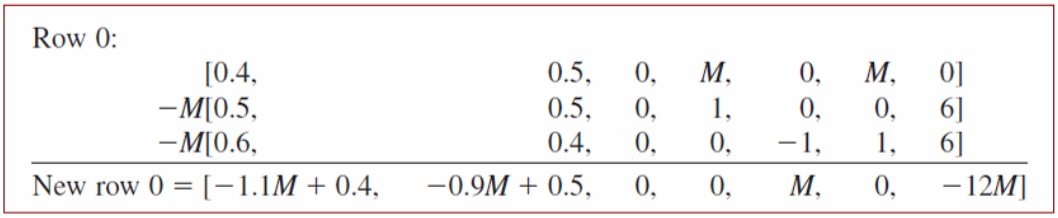


$$\begin{array}{ll}
\text{Minimize} & Z = 0.4x_1 + 0.5x_2 + M\bar{x}_4 + M\bar{x}_6 \\
\text{Maximize} & -Z = -0.4x_1 - 0.5x_2 - M\bar{x}_4 - M\bar{x}_6.
\end{array}$$



$$\begin{array}{lllll}
(0) & -Z + 0.4x_1 + 0.5x_2 & + M\bar{x}_4 & + M\bar{x}_6 & = 0 \\
(1) & 0.3x_1 + 0.1x_2 + x_3 & & & = 2.7 \\
(2) & 0.5x_1 + 0.5x_2 & + \bar{x}_4 & & = 6 \\
(3) & 0.6x_1 + 0.4x_2 & & - x_5 + \bar{x}_6 & = 6.
\end{array}$$

Proper Form



Row 0:

	[0.4,	0.5,	0,	M,	0,	M,	0]
	$-M$[0.5,	0.5,	0,	1,	0,	0,	6]
	$-M$[0.6,	0.4,	0,	0,	−1,	1,	6]
New row 0 =	[$-1.1M + 0.4$,	$-0.9M + 0.5$,	0,	0,	M,	0,	$-12M$]

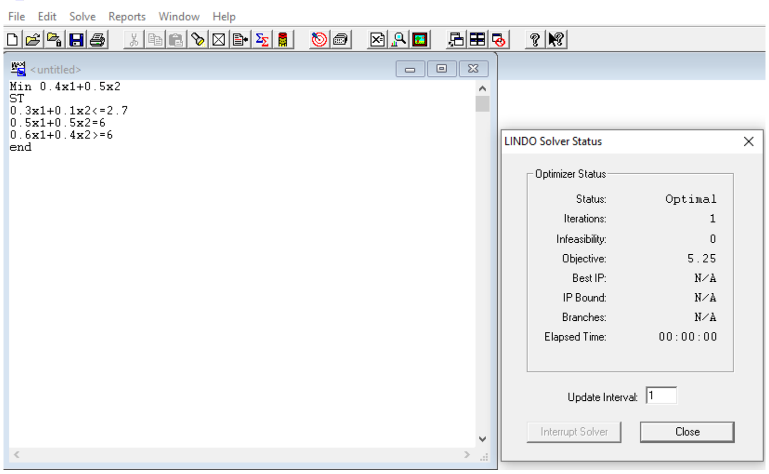
			Coefficient of:							
Iteration	Basic Variable	Eq.	Z	x_1	x_2	x_3	$\bar{x}_4$	x_5	$\bar{x}_6$	Right Side
0	Z	(0)	−1	$-1.1M + 0.4$	$-0.9M + 0.5$	0	0	M	0	$-12M$
	x_3	(1)	0	0.3	0.1	1	0	0	0	2.7
	$\bar{x}_4$	(2)	0	0.5	0.5	0	1	0	0	6
	$\bar{x}_6$	(3)	0	0.6	0.4	0	0	−1	1	6
1	Z	(0)	−1	0	$-\frac{16}{30}M + \frac{11}{30}$	$\frac{11}{3}M - \frac{4}{3}$	0	M	0	$-2.1M - 3.6$
	x_1	(1)	0	1	$\frac{1}{3}$	$\frac{10}{3}$	0	0	0	9
	$\bar{x}_4$	(2)	0	0	$\frac{1}{3}$	$-\frac{5}{3}$	1	0	0	1.5
	$\bar{x}_6$	(3)	0	0	0.2	−2	0	−1	1	0.6
2	Z	(0)	−1	0	0	$-\frac{5}{3}M + \frac{7}{3}$	0	$-\frac{5}{3}M + \frac{11}{6}$	$\frac{8}{3}M - \frac{11}{6}$	$-0.5M - 4.7$
	x_1	(1)	0	1	0	$\frac{20}{3}$	0	$\frac{5}{3}$	$-\frac{5}{3}$	8
	$\bar{x}_4$	(2)	0	0	0	$\frac{5}{3}$	1	$\frac{5}{3}$	$-\frac{5}{3}$	0.5
	x_2	(3)	0	0	1	−10	0	−5	5	3
3	Z	(0)	−1	0	0	0.5	$M - 1.1$	0	M	−5.25
	x_1	(1)	0	1	0	5	−1	0	0	7.5
	x_5	(2)	0	0	0	1	0.6	1	−1	0.3
	x_2	(3)	0	0	1	−5	3	0	0	4.5

The Big M method can be thought of as having two phases. In the first phase, all the artificial variables are driven to zero in order to reach an initial BF solution for the real problem. In the second phase, all the artificial variables are kept at zero while the simplex method generates a sequence of BF solutions for the real problem that leads to an optimal solution.

Two-phase method:

Phase 1: Minimize $Z = \bar{x}_4 + \bar{x}_6$ (until $\bar{x}_4 = 0,\ \bar{x}_6 = 0$).
Phase 2: Minimize $Z = 0.4x_1 + 0.5x_2$ (with $\bar{x}_4 = 0,\ \bar{x}_6 = 0$).

No feasible solution: If the original problem has no feasible solutions, then either the Big M method or phase 1 of the two-phase method yields a final solution that has at least one artificial variable greater than zero. Otherwise, they all equal zero.



```
Min 0.4x1+0.5x2
ST
0.3x1+0.1x2<=2.7
0.5x1+0.5x2=6
0.6x1+0.4x2>=6
end
```

Status: Optimal
Iterations: 1
Infeasibility: 0
Objective: 5.25
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00
Update Interval: 1

3. Special Cases

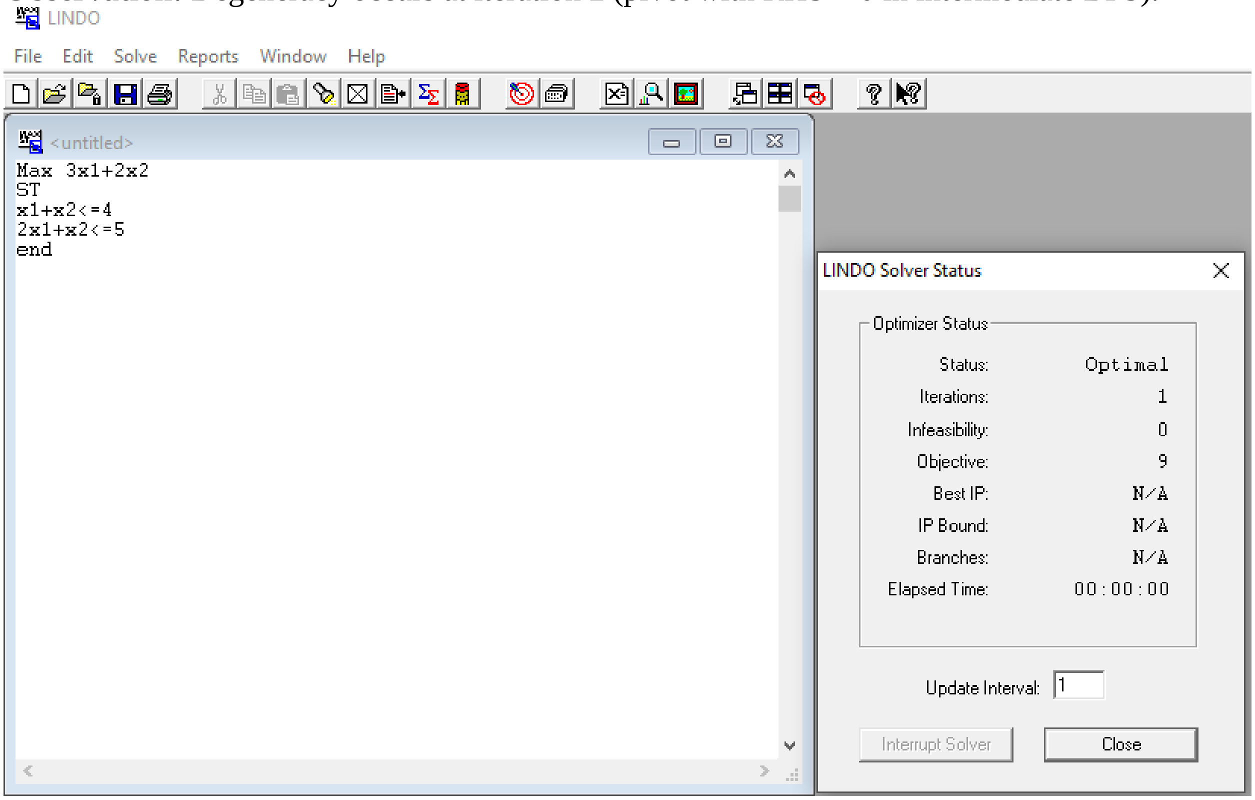
3.1 Case 1: Degeneracy

<u>**Tie for leaving basic variable(degeneracy):**</u> This is a tie for the **minimum ratio.** Whichever variable is picked to leave, the other variable will also be driven to zero in the pivot. Problems may ensue. If one of the degenerate basic variables (**basic variables with a value of zero**) retains its zero value until it is chosen at a subsequent iteration to be the leaving basic variable, the corresponding **entering variable will be stuck at zero** since it can't be increased without making the degenerate leaving variable negative, so the **value of Z won't change**. Simplex may go around in a loop, **repeating the same sequence** without advancing.

Example Problem:

Max Z = $3X_1 + 2X_2$
Constraints:

$X_1 + X_2 <= 4$
$2X_1 + X_2 <= 5$
$X_1, X_2 >= 0$
Observation: Degeneracy occurs at iteration 2 (pivot with RHS = 0 in intermediate BFS).



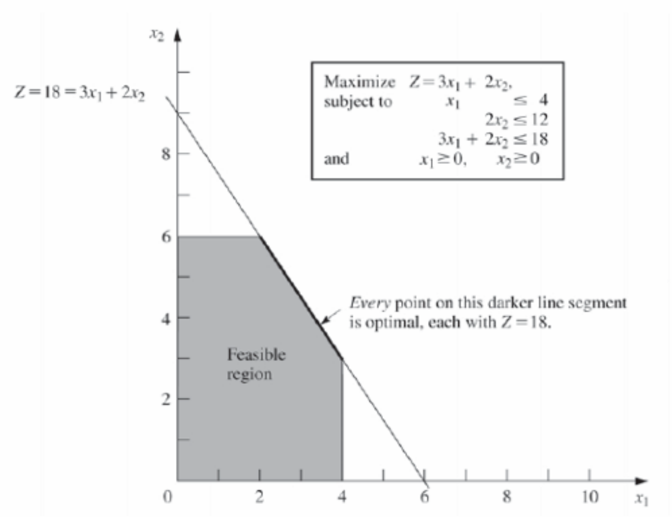
```
Max 3x1+2x2
ST
x1+x2<=4
2x1+x2<=5
end
```

Status: Optimal
Iterations: 1
Infeasibility: 0
Objective: 9
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00
Update Interval: 1

Practice: MAX $2X_1 + 2X_2$
ST
$X_1 + X_2 <= 4$
$2X_1 + 2X_2 <= 8$

3.2 Case 2: Alternative Optima

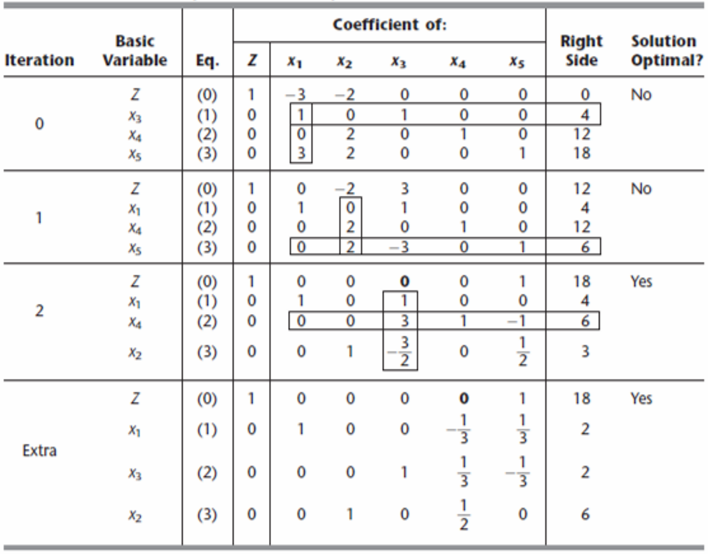
In this case, the **objective function has the same slope as one of the constraints**. Whenever a problem has more than one optimal BFS, ***at least one of the non-basic variables will have a coefficient of zero in the final objective function***. If a non-basic variable has ZERO reduced cost at optimality, then multiple optimal solutions exist. If the coefficient of non-basic is zero, it indicates that the non-basic can enter the basic solution without changing the value of z.



Every point on this darker line segment is optimal, each with $Z = 18$.

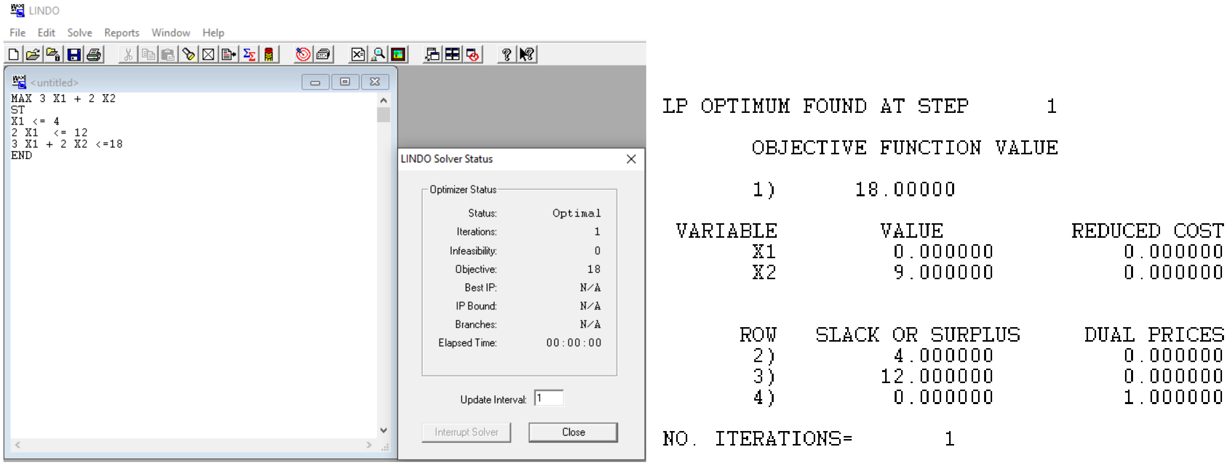
$$(x_1, x_2) = w_1(2, 6) + w_2(4, 3),$$

$$w_1 + w_2 = 1 \qquad \text{and} \qquad w_1 \geq 0, \qquad w_2 \geq 0.$$



Iteration	Basic Variable	Eq.	Coefficient of:						Right Side	Solution Optimal?
			Z	x_1	x_2	x_3	x_4	x_5		
0	Z	(0)	1	−3	−2	0	0	0	0	No
	x_3	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	2	0	1	0	12	
	x_5	(3)	0	3	2	0	0	1	18	
1	Z	(0)	1	0	−2	3	0	0	12	No
	x_1	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	2	0	1	0	12	
	x_5	(3)	0	0	2	−3	0	1	6	
2	Z	(0)	1	0	0	**0**	0	1	18	Yes
	x_1	(1)	0	1	0	1	0	0	4	
	x_4	(2)	0	0	0	3	1	−1	6	
	x_2	(3)	0	0	1	$-\frac{3}{2}$	0	$\frac{1}{2}$	3	
Extra	Z	(0)	1	0	0	0	**0**	1	18	Yes
	x_1	(1)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2	
	x_3	(2)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2	
	x_2	(3)	0	0	1	0	$\frac{1}{2}$	0	6	

$$(x_1, x_2, x_3, x_4, x_5) = w_1(2, 6, 2, 0, 0) + w_2(4, 3, 0, 6, 0),$$
$$w_1 + w_2 = 1, \qquad w_1 \geq 0, \qquad w_2 \geq 0.$$



```
MAX 3 X1 + 2 X2
ST
X1 <= 4
2 X1  <= 12
3 X1 + 2 X2 <=18
END
```

Optimizer Status

Status: Optimal
Iterations: 1
Infeasibility: 0
Objective: 18
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00

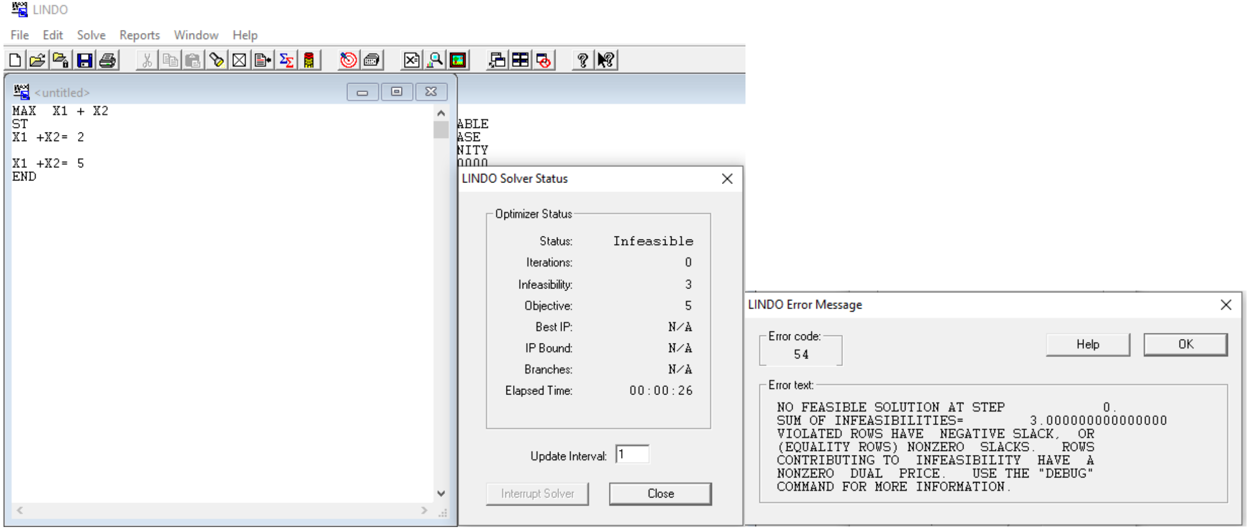
Update Interval: 1

```
LP OPTIMUM FOUND AT STEP      1

       OBJECTIVE FUNCTION VALUE

       1)      18.00000

 VARIABLE        VALUE          REDUCED COST
       X1         0.000000          0.000000
       X2         9.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         4.000000          0.000000
       3)        12.000000          0.000000
       4)         0.000000          1.000000

NO. ITERATIONS=       1
```

3.3 Case 3: Infeasible Solution

Example Problem: Max Z = $X_1 + X_2$

$X_1 + X_2 = 2$
$X_1 + X_2 = 5$
$X_1, X_2 >= 0$



```
MAX  X1 + X2
ST
X1 +X2= 2

X1 +X2= 5
END
```

Optimizer Status

Status: Infeasible
Iterations: 0
Infeasibility: 3
Objective: 5
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:26

Update Interval: 1

Error code: 54

Error text:

```
NO FEASIBLE SOLUTION AT STEP            0.
SUM OF INFEASIBILITIES=        3.000000000000000
VIOLATED ROWS HAVE  NEGATIVE SLACK,  OR
(EQUALITY ROWS) NONZERO  SLACKS.   ROWS
CONTRIBUTING TO  INFEASIBILITY  HAVE  A
NONZERO  DUAL  PRICE.   USE THE "DEBUG"
COMMAND FOR MORE INFORMATION.
```

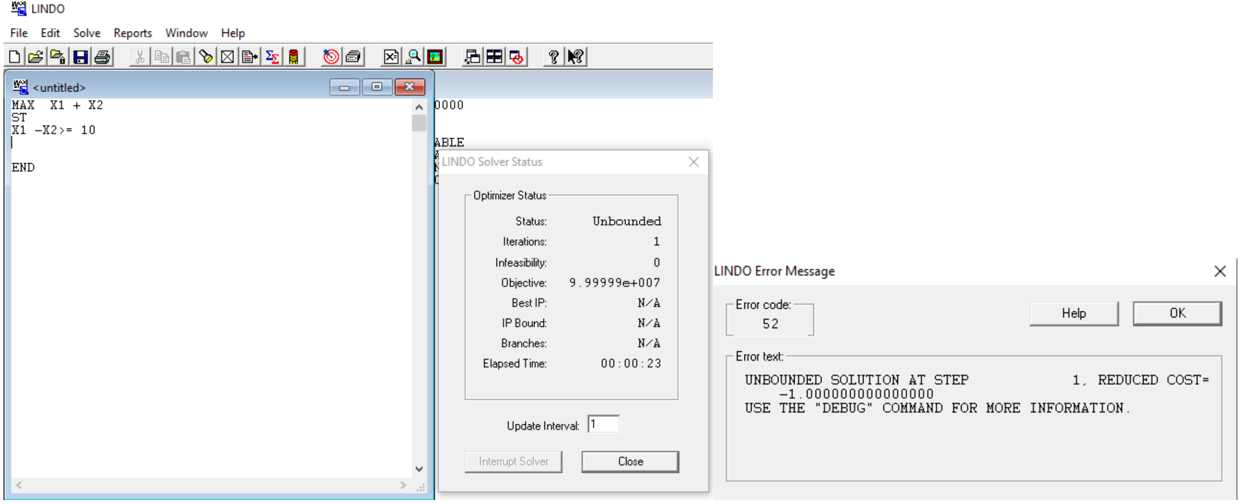
- LINDO reports **infeasible** automatically.
- Artificial variables remain non-zero at optimum.

3.4 Case 4: Unbounded Solution

If Z is unbounded, there will be no candidate for the leaving basic variable. If at any iteration any of the candidates for the entering variable has all negative or zero co-efficient in the constraints equation, it is an indication that the problem has an unbounded solution. LINDO reports unbounded without performing unnecessary iterations

Problem: Max Z = $X_1 + X_2$

$X_1 - X_2 >= 1$
$X_1, X_2 >= 0$



```
MAX   X1 + X2
ST
X1 -X2>= 10

END
```

Status: Unbounded
Iterations: 1
Infeasibility: 0
Objective: 9.99999e+007
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:23

Error code: 52

Error text:

```
UNBOUNDED SOLUTION AT STEP          1, REDUCED COST=
    -1.0000000000000000
USE THE "DEBUG" COMMAND FOR MORE INFORMATION.
```

4. Practice Problem:

1. Use the simplex algorithm to solve the following problem:

$$\begin{aligned} \max z &= 2x_1 - x_2 + x_3 \\ \text{s.t.} \quad 3x_1 + x_2 + x_3 &\le 60 \\ 2x_1 + x_2 + 2x_3 &\le 20 \\ 2x_1 + 2x_2 + x_3 &\le 20 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

2. Use the simplex algorithm to find the optimal solution to the following LP:

$$\begin{aligned} \min z &= -3x_1 + 8x_2 \\ \text{s.t.} \quad 4x_1 + 2x_2 &\le 12 \\ 2x_1 + 3x_2 &\le 6 \\ x_1, x_2 &\ge 0 \end{aligned}$$

3.

$$\begin{aligned} \max z &= -3x_1 + x_2 - 6x_3 \\ 9x_1 + x_2 - 9x_3 - 2x_4 &\le 0 \\ x_1 + \tfrac{x_2}{3} - 2x_3 - \tfrac{x_4}{3} &\le 0 \\ -9x_1 - x_2 + 9x_3 + 2x_4 &\le 1 \\ x_i \ge 0 \quad (i &= 1, 2, 3, 4) \end{aligned}$$

4.

$$\begin{aligned} \min z &= 4x_1 + 4x_2 + x_3 \\ \text{s.t.} \quad x_1 + x_2 + x_3 &\le 2 \\ 2x_1 + x_2 &\le 3 \\ 2x_1 + x_2 + 3x_3 &\ge 3 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

Experiment–3: Solving Linear Programming Problems: Duality and Sensitivity Analysis

1. Objective

- To understand the principles of duality and sensitivity analysis in linear programming.
- To apply LINDO software for analyzing optimal solutions of LP problems.
- To interpret reduced costs, shadow prices, and allowable ranges from LINDO output.
- To evaluate the impact of parameter changes using the 100% rule.
- To assess the stability and economic meaning of optimal LP solutions without re-solving the model.

2. Introduction

Linear programming models are extensively used in engineering, management science, and operations research to determine the optimal allocation of limited resources among competing activities. In practice, however, parameters such as cost coefficients, profit contributions, and resource availability are often subject to uncertainty and may vary due to market conditions, operational disruptions, or managerial decisions. As a result, an optimal solution obtained for a given set of parameters may not remain valid when these values change.

Sensitivity analysis provides a systematic framework to study how variations in model parameters influence the optimal solution, the objective function value, and the feasibility of the solution. It enables decision-makers to assess the robustness of an optimal plan and to determine the range within which parameter changes can occur without requiring re-optimization. Duality theory complements sensitivity analysis by offering an economic interpretation of linear programming solutions, particularly through concepts such as shadow prices and resource valuation. Together, sensitivity analysis and duality transform linear programming from a purely mathematical optimization tool into a powerful decision-support mechanism for real-world applications.

LINDO software provides built-in sensitivity and duality outputs that allow decision-makers to perform what-if analysis efficiently.

3. Sensitivity Analysis: Fundamental Concepts

Sensitivity analysis studies how variations in LP parameters influence:

- The optimal basis
- Decision variable values
- The optimal objective function value

It answers questions such as:

- How much is an additional unit of a resource worth?
- Within what range can parameters change without altering the optimal plan?

- Which parameters are critical and which are non-binding? (A constraint is nonbinding if the left-hand side and the right-hand side of the constraint are unequal when the optimal values of the decision variables are substituted into the constraint.)

Sensitivity analysis results are valid **only when the optimal basis remains unchanged**.

Example 1:

Winco sells four types of products. The resources needed to produce one unit of each and the sales prices are given in Table 2. Currently, 4,600 units of raw material and 5,000 labor hours are available. To meet customer demands, exactly 950 total units must be produced. Customers also demand that at least 400 units of product 4 be produced. Formulate an LP that can be used to maximize Winco's sales revenue.

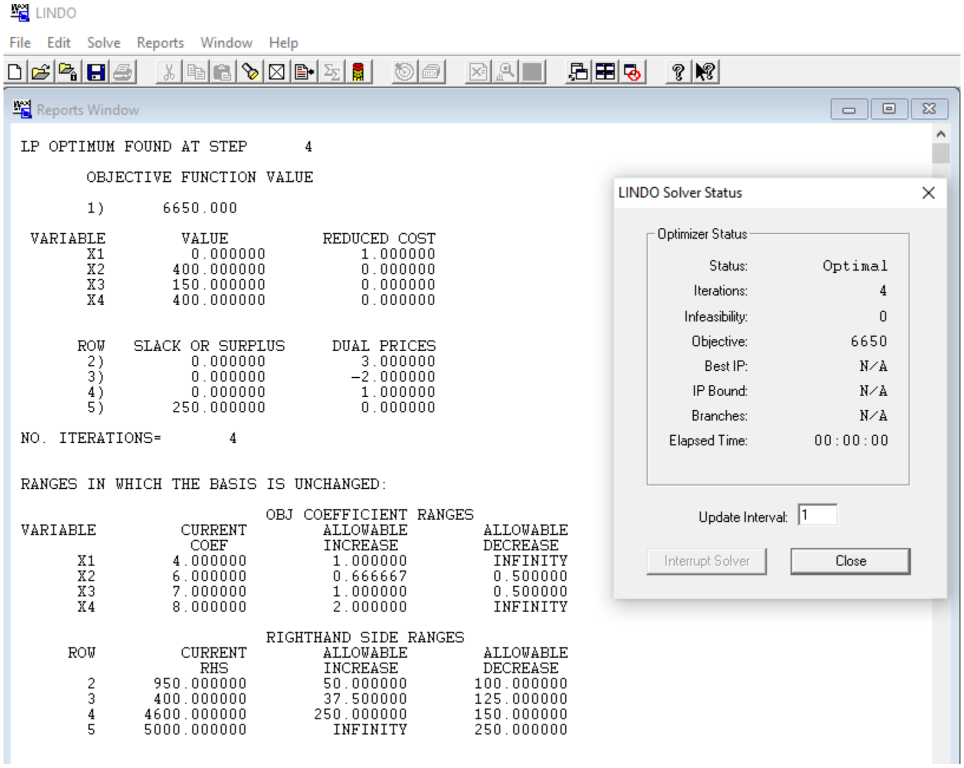
Let x_i = number of units of product i produced by Winco.

$$\begin{aligned}
\max z &= 4x_1 + 6x_2 + 7x_3 + 8x_4 \\
\text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 = 950 \\
& x_4 \geq 400 \\
& 2x_1 + 3x_2 + 4x_3 + 7x_4 \leq 4{,}600 \\
& 3x_1 + 4x_2 + 5x_3 + 6x_4 \leq 5{,}000 \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

TABLE 2

Costs and Resource Requirements for Winco

Resource	Product 1	Product 2	Product 3	Product 4
Raw material	2	3	4	7
Hours of labor	3	4	5	6
Sales price ($)	4	6	7	8



```
LP OPTIMUM FOUND AT STEP      4

       OBJECTIVE FUNCTION VALUE

       1)      6650.000

  VARIABLE        VALUE          REDUCED COST
        X1         0.000000          1.000000
        X2       400.000000          0.000000
        X3       150.000000          0.000000
        X4       400.000000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          3.000000
        3)         0.000000         -2.000000
        4)         0.000000          1.000000
        5)       250.000000          0.000000

NO. ITERATIONS=       4


RANGES IN WHICH THE BASIS IS UNCHANGED:

                         OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
      X1        4.000000         1.000000         INFINITY
      X2        6.000000         0.666667         0.500000
      X3        7.000000         1.000000         0.500000
      X4        8.000000         2.000000         INFINITY

                         RIGHTHAND SIDE RANGES
     ROW         CURRENT        ALLOWABLE        ALLOWABLE
                   RHS          INCREASE         DECREASE
       2       950.000000        50.000000       100.000000
       3       400.000000        37.500000       125.000000
       4      4600.000000       250.000000       150.000000
       5      5000.000000         INFINITY       250.000000
```

Status: Optimal
Iterations: 4
Infeasibility: 0
Objective: 6650
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00
Update Interval: 1

Example 2:

Tucker Inc. must produce 1,000 Tucker automobiles. The company has four production plants. The cost of producing a Tucker at each plant, along with the raw material and labor needed, is shown in Table 3. The autoworkers' labor union requires that at least 400 cars be produced at plant 3; 3,300 hours of labor and 4,000 units of raw material are available for allocation to the four plants. Formulate an LP whose solution will enable Tucker Inc. to minimize the cost of producing 1,000 cars.

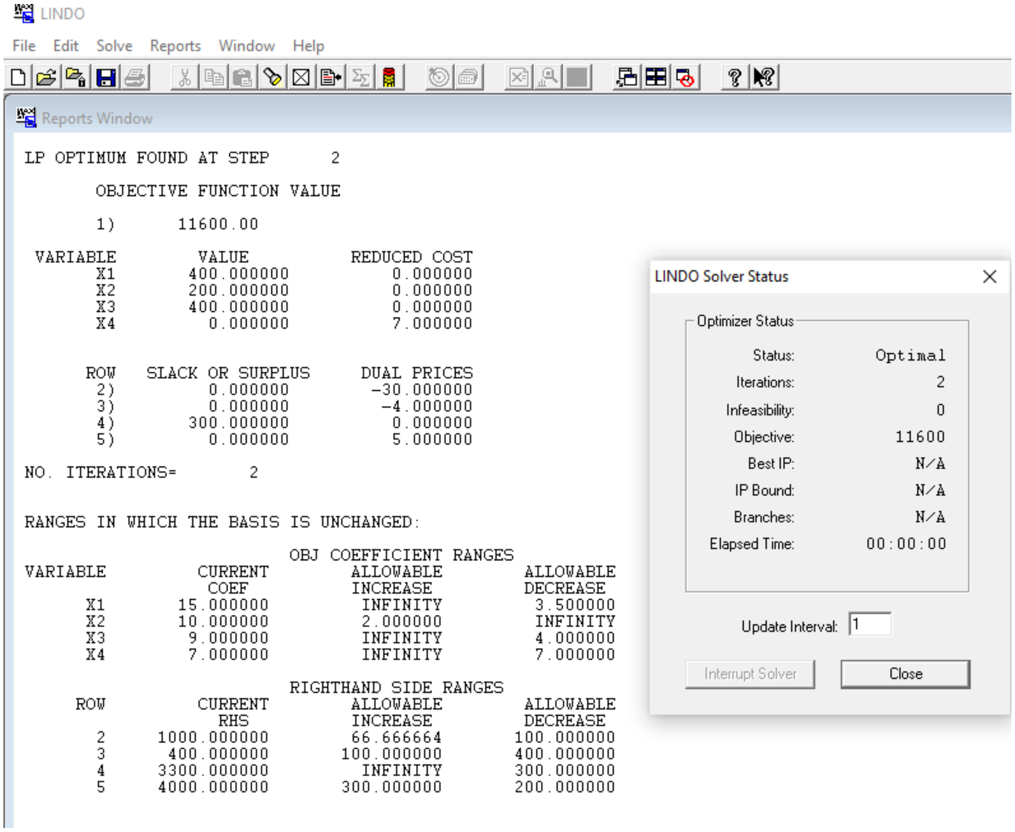
Let x_i = number of cars produced at plant i. Then, expressing the objective function in thousands of dollars, the appropriate LP is

$$\begin{aligned} \min z &= 15x_1 + 10x_2 + 9x_3 + 7x_4 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 = 1000 \\ & x_3 \ge 400 \\ & 2x_1 + 3x_2 + 4x_3 + 5x_4 \le 3300 \\ & 3x_1 + 4x_2 + 5x_3 + 6x_4 \le 4000 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

TABLE 3

Cost and Requirements for Producing a Tucker

Plant	Cost (in Thousands of Dollars)	Labor	Raw Material
1	15	2	3
2	10	3	4
3	9	4	5
4	7	5	6



```
 LP OPTIMUM FOUND AT STEP      2

        OBJECTIVE FUNCTION VALUE

        1)      11600.00

  VARIABLE        VALUE          REDUCED COST
        X1       400.000000          0.000000
        X2       200.000000          0.000000
        X3       400.000000          0.000000
        X4         0.000000          7.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000        -30.000000
        3)         0.000000         -4.000000
        4)       300.000000          0.000000
        5)         0.000000          5.000000

 NO. ITERATIONS=       2


 RANGES IN WHICH THE BASIS IS UNCHANGED:

                           OBJ COEFFICIENT RANGES
 VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                   COEF          INCREASE         DECREASE
       X1        15.000000        INFINITY         3.500000
       X2        10.000000        2.000000         INFINITY
       X3         9.000000        INFINITY         4.000000
       X4         7.000000        INFINITY         7.000000

                           RIGHTHAND SIDE RANGES
      ROW         CURRENT        ALLOWABLE        ALLOWABLE
                    RHS          INCREASE         DECREASE
        2      1000.000000        66.666664       100.000000
        3       400.000000       100.000000       400.000000
        4      3300.000000        INFINITY        300.000000
        5      4000.000000       300.000000       200.000000
```

Status: Optimal
Iterations: 2
Infeasibility: 0
Objective: 11600
Best IP: N/A
IP Bound: N/A
Branches: N/A
Elapsed Time: 00:00:00

Update Interval: 1

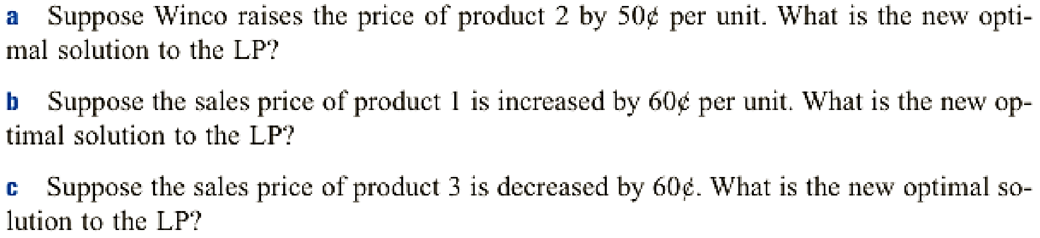
3.1 Sensitivity with Respect to Objective Function Coefficients

Each decision variable has an associated **range of optimality** for its objective function coefficient.

- Within the allowable range:
  - The optimal basis remains unchanged
  - Decision variable values remain unchanged
  - Only the objective value changes
- Outside the range:
  - The current basis may no longer be optimal
  - Re-optimization is required

LINDO reports for each variable:

- Current coefficient
- Allowable increase
- Allowable decrease



a Suppose Winco raises the price of product 2 by 50¢ per unit. What is the new optimal solution to the LP?

b Suppose the sales price of product 1 is increased by 60¢ per unit. What is the new optimal solution to the LP?

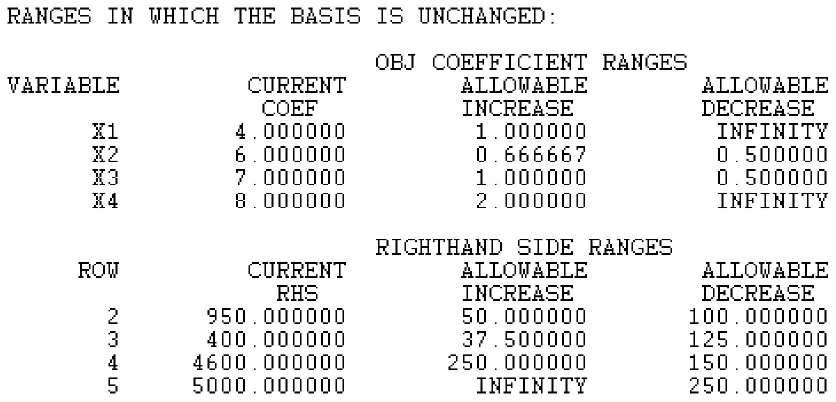
c Suppose the sales price of product 3 is decreased by 60¢. What is the new optimal solution to the LP?

3.2 Reduced Cost

The **reduced cost** of a variable **indicates how much its objective coefficient must improve before the variable can enter the optimal solution**. The REDUCED COST portion of the LINDO output gives us information about **how changing the objective function coefficient for a nonbasic variable will change the LP's optimal solution**. Reduced cost reflects **opportunity loss** due to excluding a variable from the basis.

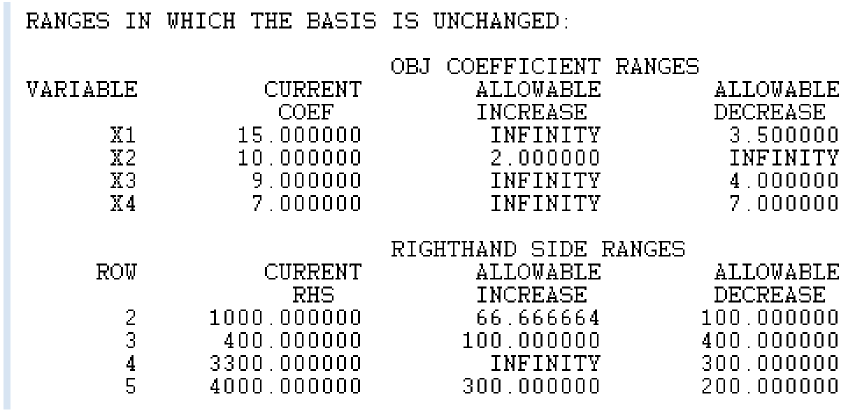
- Reduced cost = 0 → variable is basic
- Reduced cost ≠ 0 → variable is nonbasic

For a maximization problem a nonbasic variable with reduced cost *r* must increase its coefficient by *r* to become positive in the solution. For a minimization problem a nonbasic variable with reduced cost *r* must decrease its coefficient by *r* to become positive in the solution.



```
RANGES IN WHICH THE BASIS IS UNCHANGED:

                           OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
      X1        4.000000        1.000000         INFINITY
      X2        6.000000        0.666667         0.500000
      X3        7.000000        1.000000         0.500000
      X4        8.000000        2.000000         INFINITY

                           RIGHTHAND SIDE RANGES
     ROW         CURRENT        ALLOWABLE        ALLOWABLE
                   RHS          INCREASE         DECREASE
       2      950.000000       50.000000       100.000000
       3      400.000000       37.500000       125.000000
       4     4600.000000      250.000000       150.000000
       5     5000.000000        INFINITY       250.000000
```

The nonbasic variable x$_1$ has a reduced cost of \$1. This implies that if we increase x$_1$'s objective function coefficient (in this case, the sales price per unit of x$_1$) by exactly \$1, then there will be **alternative optimal solutions**, at least one of which will have x$_1$ as a basic variable. If we increase x$_1$'s objective function coefficient by more than \$1, then (because the current optimal bfs is nondegenerate) any optimal solution to the LP will have x$_1$ as a basic variable (with x$_1$ > 0). Thus, the reduced cost for x$_1$ is the amount by which x1 "misses the optimal basis."



```
RANGES IN WHICH THE BASIS IS UNCHANGED:

                           OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                  COEF          INCREASE         DECREASE
      X1       15.000000        INFINITY         3.500000
      X2       10.000000        2.000000         INFINITY
      X3        9.000000        INFINITY         4.000000
      X4        7.000000        INFINITY         7.000000

                           RIGHTHAND SIDE RANGES
     ROW         CURRENT        ALLOWABLE        ALLOWABLE
                   RHS          INCREASE         DECREASE
       2     1000.000000       66.666664       100.000000
       3      400.000000      100.000000       400.000000
       4     3300.000000        INFINITY       300.000000
       5     4000.000000      300.000000       200.000000
```

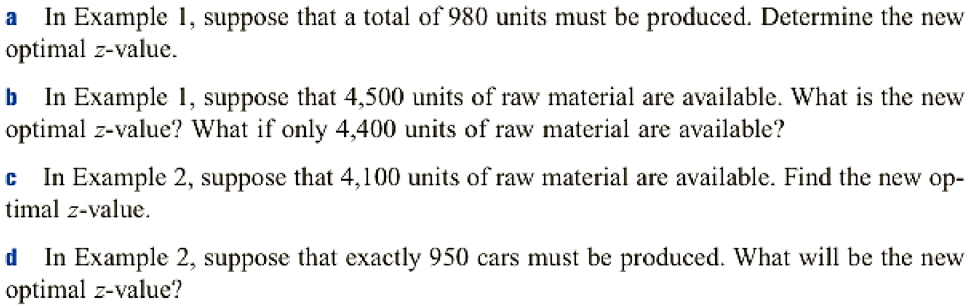
The nonbasic variable x$_4$ has a reduced cost of 7 (\$7,000), so we know that if the cost of producing x$_4$ is decreased by 7, then there will be **alternative optimal solutions**. In at least one of these optimal solutions, x$_4$ will be a basic variable. If the cost of producing x$_4$ is lowered by more than 7, then (because the current optimal solution is nondegenerate) any optimal solution to the LP will have x$_4$ as a basic variable (with x$_4$ > 0).

3.3 Sensitivity with Respect to Right-Hand-Side (RHS) Values

Each constraint has a **range of feasibility** for its RHS value.

- Within the allowable range:
    - Current basis remains optimal
    - Shadow prices remain valid
- **Decision variable values may change**
- **The objective value changes linearly**

If a constraint has positive slack, changes in its RHS (within the allowable range) do not affect the optimal solution.



a In Example 1, suppose that a total of 980 units must be produced. Determine the new optimal z-value.

b In Example 1, suppose that 4,500 units of raw material are available. What is the new optimal z-value? What if only 4,400 units of raw material are available?

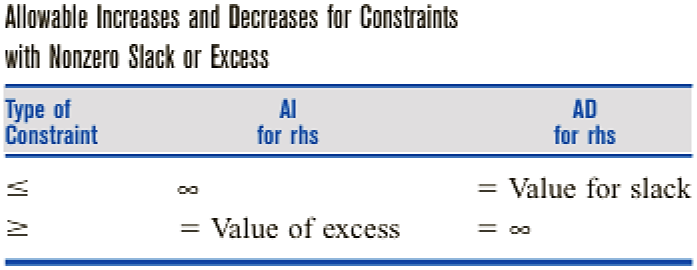
c In Example 2, suppose that 4,100 units of raw material are available. Find the new optimal z-value.

d In Example 2, suppose that exactly 950 cars must be produced. What will be the new optimal z-value?

3.4 Shadow Price

The **shadow price of a constraint** is the **change in the optimal objective** value resulting from a **one-unit increase in the RHS of that constraint**, provided the current basis remains optimal. If, after a change in a constraint's right-hand side, the current basis is no longer optimal, then the shadow prices of all constraints may change. The shadow price for each constraint is found in the **DUAL PRICES** section of the LINDO output

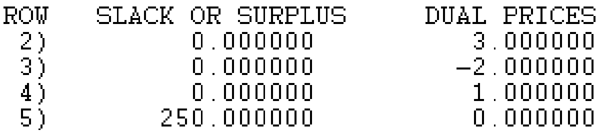
- Maximization problem → marginal increase in profit
- Minimization problem → marginal decrease in cost
- Shadow price is valid only within the allowable RHS range
- Binding constraint → nonzero shadow price
- Nonbinding constraint → zero shadow price
- A >= constraint will always have a **nonpositive** shadow price; a <= constraint will always have a **nonnegative** shadow price; and an equality constraint may have a positive, negative, or zero shadow price. An equality constraint's shadow price may be positive, negative, or zero.
- For any inequality constraint, the product of the values of the constraint's slack or excess variable and the constraint's shadow price must equal 0. This implies that any constraint whose slack or excess variable is 0 will have a zero shadow price. It also implies that any constraint with a nonzero shadow price must be binding (have slack or excess equal to 0).



Allowable Increases and Decreases for Constraints with Nonzero Slack or Excess

Type of Constraint	AI for rhs	AD for rhs
≤	∞	= Value for slack
≥	= Value of excess	= ∞

Let's give an interpretation to the shadow price for each constraint in Examples 1 and 2. Again, all discussions are assuming that we are within the allowable range where the current basis remains optimal. The shadow price of $3 for Constraint 1 in Example 1 implies that each one-unit increase in total demand will increase sales revenues by $3. The shadow price of $2 for Constraint 2 implies that each unit increase in the requirement for product 4 will decrease revenue by $2. The shadow price of $1 for Constraint 3 implies that an additional unit of raw material given to Winco (for no cost) increases total revenue by $1. Finally, the shadow price of $0 for Constraint 4 implies that an additional unit of labor given to Winco (at no cost) will not increase total revenue. This is reasonable.



```
ROW   SLACK OR SURPLUS     DUAL PRICES
 2)         0.000000          3.000000
 3)         0.000000         -2.000000
 4)         0.000000          1.000000
 5)       250.000000          0.000000
```

$$\text{(New optimal } z\text{-value)} = \text{(old optimal } z\text{-value)} + \text{(Constraint } i\text{'s shadow price) } \Delta b_i$$

For a minimization problem,

$$\text{(New optimal } z\text{-value)} = \text{(old optimal } z\text{-value)} - \text{(Constraint } i\text{'s shadow price) } \Delta b_i$$

3.5 Degeneracy and Sensitivity Analysis

An LP solution is **degenerate** if one or more basic variables have zero value.

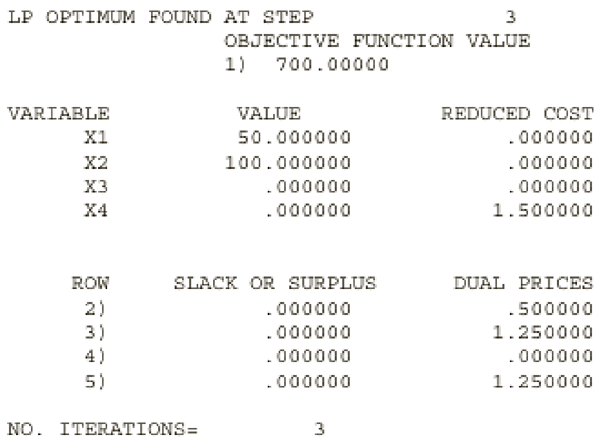
Effects:

- Allowable increases or decreases may be zero
- Reduced cost interpretation becomes less reliable
- The basis may change without changing the objective value

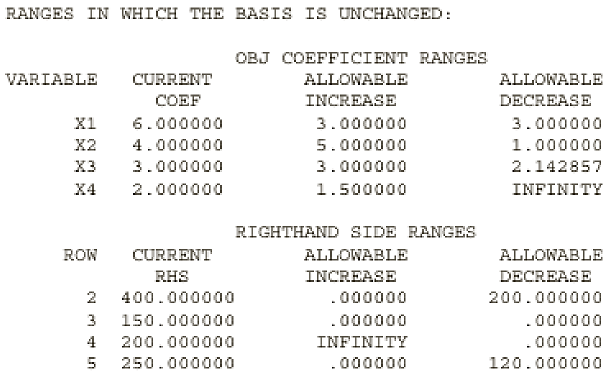
Sensitivity results must be interpreted cautiously for degenerate solutions.

$$\begin{aligned}
\max z = 6x_1 + 4x_2 + 3x_3 + 2x_4 & \\
\text{s.t.} \quad 2x_1 + 3x_2 + x_3 + 2x_4 &\le 400 \\
x_1 + x_2 + 2x_3 + x_4 &\le 150 \\
2x_1 + x_2 + x_3 + .5x_4 &\le 200 \\
3x_1 + x_2 \qquad\qquad + x_4 &\le 250 \\
x_1, x_2, x_3, x_4 &\ge 0
\end{aligned}$$

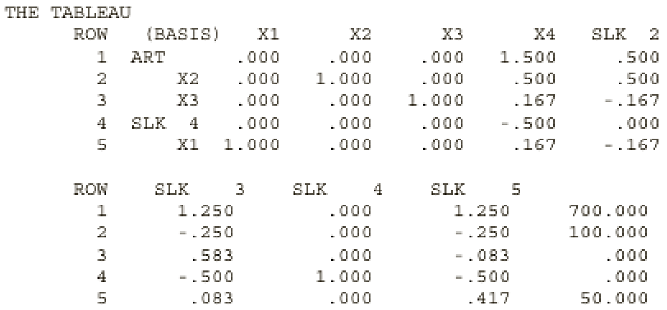
<u>Oddity 1</u> In the RANGES IN WHICH THE BASIS IS UNCHANGED, **at least one constraint** will have a 0 AI or AD. This means that for at least one constraint, the DUAL PRICE can tell us about the new z-value for either an increase or decrease in the right-hand side, but not both. To understand Oddity 1, consider the first constraint. Its AI is 0. This means that the first constraint's DUAL PRICE of .50 cannot be used to determine a new z-value resulting from any increase in the first constraint's right-hand side.



```
LP OPTIMUM FOUND AT STEP              3
                 OBJECTIVE FUNCTION VALUE
                 1)   700.00000

VARIABLE           VALUE            REDUCED COST
      X1          50.000000           .000000
      X2         100.000000           .000000
      X3            .000000           .000000
      X4            .000000          1.500000

     ROW     SLACK OR SURPLUS       DUAL PRICES
      2)            .000000           .500000
      3)            .000000          1.250000
      4)            .000000           .000000
      5)            .000000          1.250000

NO. ITERATIONS=          3
```



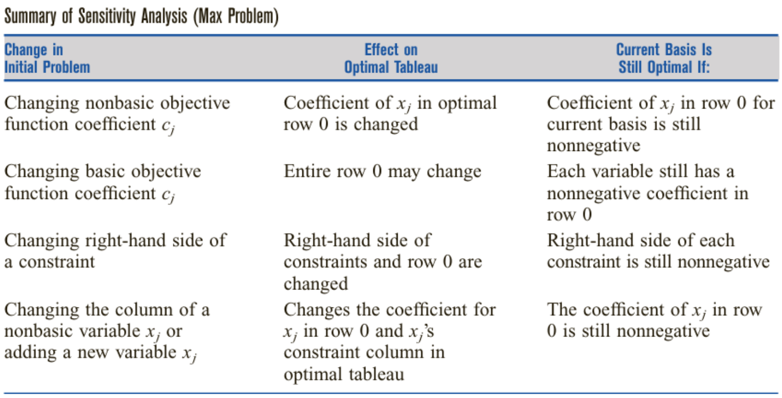
```
RANGES IN WHICH THE BASIS IS UNCHANGED:

                         OBJ COEFFICIENT RANGES
VARIABLE     CURRENT         ALLOWABLE          ALLOWABLE
              COEF           INCREASE           DECREASE
      X1     6.000000        3.000000           3.000000
      X2     4.000000        5.000000           1.000000
      X3     3.000000        3.000000           2.142857
      X4     2.000000        1.500000           INFINITY

                         RIGHTHAND SIDE RANGES
     ROW     CURRENT         ALLOWABLE          ALLOWABLE
              RHS            INCREASE           DECREASE
       2   400.000000         .000000         200.000000
       3   150.000000         .000000            .000000
       4   200.000000        INFINITY            .000000
       5   250.000000         .000000         120.000000
```



```
THE TABLEAU
     ROW   (BASIS)    X1       X2       X3       X4    SLK  2
      1   ART       .000     .000     .000    1.500      .500
      2       X2    .000    1.000     .000     .500      .500
      3       X3    .000     .000    1.000     .167     -.167
      4   SLK  4    .000     .000     .000    -.500      .000
      5       X1   1.000     .000     .000     .167     -.167

     ROW    SLK    3    SLK    4    SLK    5
      1      1.250        .000       1.250      700.000
      2      -.250        .000       -.250      100.000
      3       .583        .000       -.083         .000
      4      -.500       1.000       -.500         .000
      5       .083        .000        .417       50.000
```

Oddity 2 For a nonbasic variable to become positive, its objective function coefficient may have to be improved by more than its REDUCED COST. To understand Oddity 2, consider the nonbasic variable x_4; its REDUCED COST is 1.5. If we increase its objective function coefficient by 2, however, we still find that the new optimal solution has $x_4 = 0$. This oddity occurs because the increase changes the set of basic variables but not the LP's optimal solution.

Oddity 3 Increasing a variable's objective function coefficient by more than its AI or decreasing it by more than its AD may leave the optimal solution to the LP the same. Oddity 3 is similar to Oddity 2. To understand it, consider the nonbasic variable x_4. Its AI is 1.5. If we increase its objective function coefficient by 2, however, we still find that the new optimal solution is **unchanged**. This oddity occurs because the **increase changes the set of basic variables but not the LP's optimal solution**



Summary of Sensitivity Analysis (Max Problem)

Change in Initial Problem	Effect on Optimal Tableau	Current Basis Is Still Optimal If:
Changing nonbasic objective function coefficient c_j	Coefficient of x_j in optimal row 0 is changed	Coefficient of x_j in row 0 for current basis is still nonnegative
Changing basic objective function coefficient c_j	Entire row 0 may change	Each variable still has a nonnegative coefficient in row 0
Changing right-hand side of a constraint	Right-hand side of constraints and row 0 are changed	Right-hand side of each constraint is still nonnegative
Changing the column of a nonbasic variable x_j or adding a new variable x_j	Changes the coefficient for x_j in row 0 and x_j's constraint column in optimal tableau	The coefficient of x_j in row 0 is still nonnegative

Sensitivity analysis results are valid only when:

- One parameter changes at a time
- Changes remain within allowable ranges
- The optimal basis remains unchanged

If multiple parameters change simultaneously, the validity of results must be checked using the **100% rule**.

3.6 The 100% Rule

The **100% rule** determines whether the current optimal basis remains valid when **multiple parameters change at the same time**. Depending on whether the objective function coefficient of any variable with a zero reduced cost in the optimal tableau is changed, there are two cases to consider:

Case 1 All variables whose objective function coefficients are changed have nonzero reduced costs in the optimal row 0. In Case 1, the current basis remains optimal if and only if the objective function co-efficient for each variable remains within the allowable range.

Case 2 At least one variable whose objective function coefficient is changed has a **reduced cost of zero**. In Case 2, we can often show that the current basis remains optimal by using the 100% Rule. Let

c_j = original objective function coefficient for x_j

Δc_j = change in c_j

I_j = maximum allowable increase in c_j for which current basis remains optimal (from LINDO output)

D_j = maximum allowable decrease in c_j for which current basis remains optimal (from LINDO output)

For each variable x_j, we define the ratio r_j:

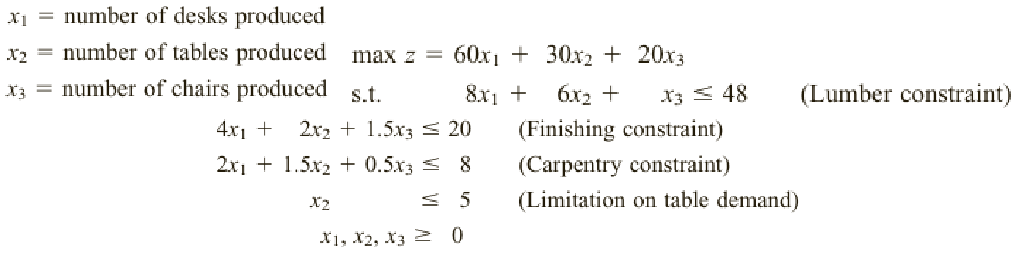
$$\text{If} \quad \Delta c_j \geq 0, \qquad r_j = \frac{\Delta c_j}{I_j}$$

$$\text{If} \quad \Delta c_j \leq 0, \qquad r_j = \frac{-\Delta c_j}{D_j}$$

If c_j is unchanged, then $r_j = 0$.

For objective function coefficients:

- Compute the fraction of change for each coefficient
- Sum of all fractions ≤ 1 → basis remains optimal

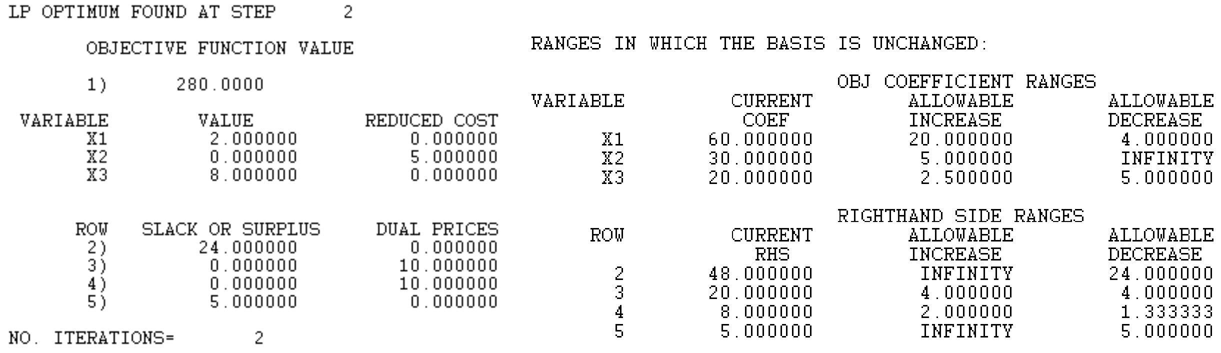


x_1 = number of desks produced

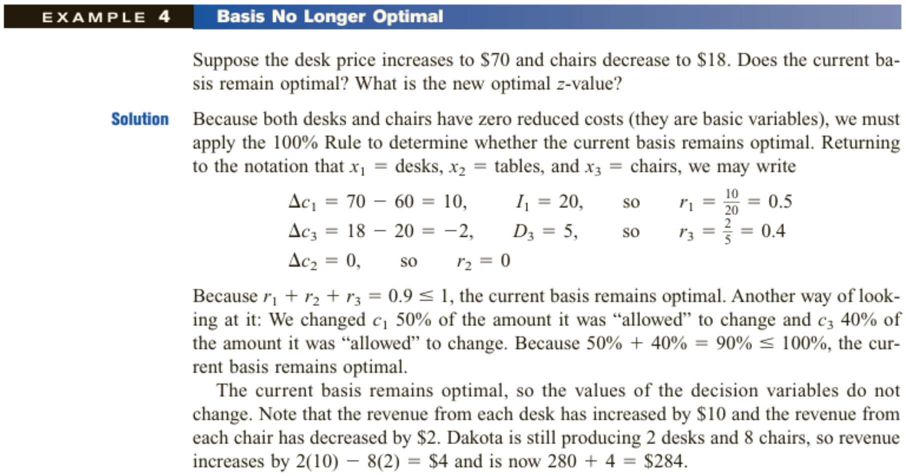
x_2 = number of tables produced

x_3 = number of chairs produced

$$\begin{array}{lrl}
\max z = & 60x_1 + 30x_2 + 20x_3 & \\
\text{s.t.} & 8x_1 + 6x_2 + x_3 \le 48 & \text{(Lumber constraint)} \\
& 4x_1 + 2x_2 + 1.5x_3 \le 20 & \text{(Finishing constraint)} \\
& 2x_1 + 1.5x_2 + 0.5x_3 \le 8 & \text{(Carpentry constraint)} \\
& x_2 \le 5 & \text{(Limitation on table demand)} \\
& x_1, x_2, x_3 \ge 0 &
\end{array}$$



```
LP OPTIMUM FOUND AT STEP      2

       OBJECTIVE FUNCTION VALUE              RANGES IN WHICH THE BASIS IS UNCHANGED:

       1)      280.0000                                               OBJ COEFFICIENT RANGES
                                             VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
 VARIABLE        VALUE          REDUCED COST                    COEF          INCREASE         DECREASE
       X1         2.000000          0.000000        X1      60.000000        20.000000         4.000000
       X2         0.000000          5.000000        X2      30.000000         5.000000         INFINITY
       X3         8.000000          0.000000        X3      20.000000         2.500000         5.000000

                                                                      RIGHTHAND SIDE RANGES
      ROW   SLACK OR SURPLUS     DUAL PRICES        ROW       CURRENT        ALLOWABLE        ALLOWABLE
       2)        24.000000          0.000000                    RHS           INCREASE         DECREASE
       3)         0.000000         10.000000         2      48.000000         INFINITY        24.000000
       4)         0.000000         10.000000         3      20.000000         4.000000         4.000000
       5)         5.000000          0.000000         4       8.000000         2.000000         1.333333
                                                     5       5.000000         INFINITY         5.000000
NO. ITERATIONS=        2
```



EXAMPLE 4 **Basis No Longer Optimal**

Suppose the desk price increases to \$70 and chairs decrease to \$18. Does the current basis remain optimal? What is the new optimal z-value?

Solution Because both desks and chairs have zero reduced costs (they are basic variables), we must apply the 100% Rule to determine whether the current basis remains optimal. Returning to the notation that x_1 = desks, x_2 = tables, and x_3 = chairs, we may write

$$\Delta c_1 = 70 - 60 = 10, \quad I_1 = 20, \quad \text{so} \quad r_1 = \frac{10}{20} = 0.5$$
$$\Delta c_3 = 18 - 20 = -2, \quad D_3 = 5, \quad \text{so} \quad r_3 = \frac{2}{5} = 0.4$$
$$\Delta c_2 = 0, \quad \text{so} \quad r_2 = 0$$

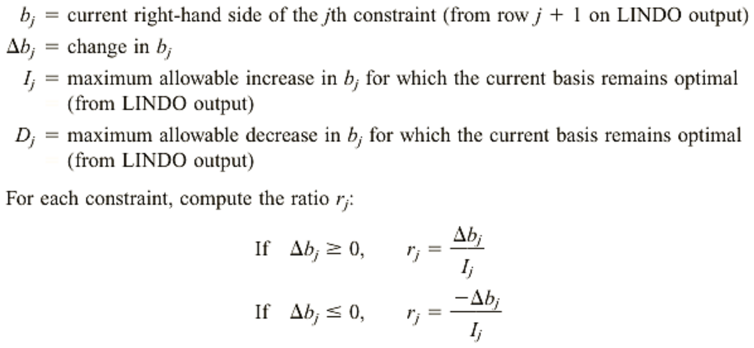
Because $r_1 + r_2 + r_3 = 0.9 \le 1$, the current basis remains optimal. Another way of looking at it: We changed c_1 50% of the amount it was “allowed” to change and c_3 40% of the amount it was “allowed” to change. Because 50% + 40% = 90% ≤ 100%, the current basis remains optimal.

The current basis remains optimal, so the values of the decision variables do not change. Note that the revenue from each desk has increased by \$10 and the revenue from each chair has decreased by \$2. Dakota is still producing 2 desks and 8 chairs, so revenue increases by $2(10) - 8(2) = \$4$ and is now $280 + 4 = \$284$.

Depending on whether any of the constraints whose right-hand sides are being modified are **binding** constraints, there are two cases to consider:

<u>Case 1</u> All constraints whose right-hand sides are being modified are **nonbinding constraints**. In Case 1, the current basis remains optimal if and only if each right-hand side remains within its allowable range.

<u>Case 2</u> At least one of the constraints whose right-hand side is being modified is a binding constraint (that is, ***has zero slack or zero excess***). In Case 2, we can often show that the current basis remains optimal via another version of the 100% Rule. Let



b_j = current right-hand side of the jth constraint (from row $j + 1$ on LINDO output)

Δb_j = change in b_j

I_j = maximum allowable increase in b_j for which the current basis remains optimal (from LINDO output)

D_j = maximum allowable decrease in b_j for which the current basis remains optimal (from LINDO output)

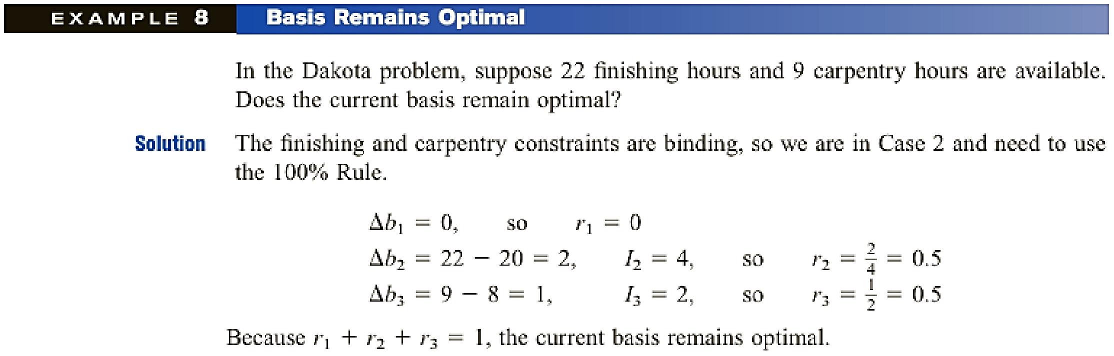
For each constraint, compute the ratio r_j:

$$\text{If} \quad \Delta b_j \geq 0, \qquad r_j = \frac{\Delta b_j}{I_j}$$

$$\text{If} \quad \Delta b_j \leq 0, \qquad r_j = \frac{-\Delta b_j}{I_j}$$

For RHS values:

- Compute the fraction of change for each RHS
- Sum of all fractions ≤ 1 → basis remains feasible and optimal



EXAMPLE 8 **Basis Remains Optimal**

In the Dakota problem, suppose 22 finishing hours and 9 carpentry hours are available. Does the current basis remain optimal?

Solution The finishing and carpentry constraints are binding, so we are in Case 2 and need to use the 100% Rule.

$$\Delta b_1 = 0, \quad \text{so} \quad r_1 = 0$$

$$\Delta b_2 = 22 - 20 = 2, \quad I_2 = 4, \quad \text{so} \quad r_2 = \tfrac{2}{4} = 0.5$$

$$\Delta b_3 = 9 - 8 = 1, \quad I_3 = 2, \quad \text{so} \quad r_3 = \tfrac{1}{2} = 0.5$$

Because $r_1 + r_2 + r_3 = 1$, the current basis remains optimal.

4. Duality in Linear Programming

In linear programming, every optimization problem, referred to as the **primal problem**, has an associated **dual problem** that is derived systematically from the primal formulation. The primal problem primarily concentrates on determining the optimal values of decision variables in order to achieve a specified objective, such as maximizing profit or minimizing cost, subject to a set of constraints. In contrast, the dual problem shifts the perspective from decision-making to **resource valuation**, where the objective is to evaluate the implicit worth or marginal value of the resources represented by the primal constraints.

A fundamental principle of duality is that the primal and dual problems are closely interconnected and convey the same economic information from different viewpoints. When both the primal and dual problems are feasible and bounded, their optimal objective function values are equal. This equivalence provides the theoretical foundation for interpreting shadow prices, reduced costs, and sensitivity analysis results, and it ensures consistency between decision-variable optimization and resource-based evaluation in linear programming models.

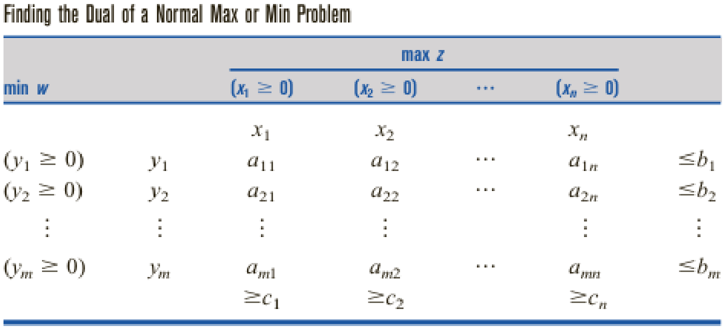
4.1 Finding the Dual of a Normal Max or Min Problem

For a maximization primal:

- Each primal constraint corresponds to a dual variable
- Each primal variable corresponds to a dual constraint
- Objective coefficients of the primal become RHS values of the dual

Sign rules:

- $\leq$ constraint → dual variable ≥ 0
- $\geq$ constraint → dual variable ≤ 0
- Equality → dual variable unrestricted



Finding the Dual of a Normal Max or Min Problem

		max z				
min w		$(x_1 \geq 0)$	$(x_2 \geq 0)$	$\cdots$	$(x_n \geq 0)$	
		x_1	x_2		x_n	
$(y_1 \geq 0)$	y_1	a_{11}	a_{12}	$\cdots$	a_{1n}	$\leq b_1$
$(y_2 \geq 0)$	y_2	a_{21}	a_{22}	$\cdots$	a_{2n}	$\leq b_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$(y_m \geq 0)$	y_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	$\leq b_m$
		$\geq c_1$	$\geq c_2$		$\geq c_n$	

Primal:

$$\begin{array}{llr}
\max z = 60x_1 + 30x_2 + 20x_3 & & \\
\text{s.t.} \quad 8x_1 + 6x_2 + x_3 \leq 48 & & \text{(Lumber constraint)} \\
\quad 4x_1 + 2x_2 + 1.5x_3 \leq 20 & & \text{(Finishing constraint)} \\
\quad 2x_1 + 1.5x_2 + 0.5x_3 \leq 8 & & \text{(Carpentry constraint)} \\
\quad x_1, x_2, x_3 \geq 0 & &
\end{array}$$

x_1 = number of desks manufactured

x_2 = number of tables manufactured

x_3 = number of chairs manufactured

Dual:

$$\begin{array}{l}
\min w = 48y_1 + 20y_2 + 8y_3 \\
\text{s.t.} \quad 8y_1 + 4y_2 + 2y_3 \geq 60 \\
\quad 6y_1 + 2y_2 + 1.5y_3 \geq 30 \\
\quad y_1 + 1.5y_2 + 0.5y_3 \geq 20 \\
\quad y_1, y_2, y_3 \geq 0
\end{array}$$

Primal:

$$\begin{array}{llr}
\min w = 50y_1 + 20y_2 + 30y_3 + 80y_4 & & \\
\text{s.t.} \quad 400y_1 + 200y_2 + 150y_3 + 500y_4 \geq 500 & & \text{(Calorie constraint)} \\
\quad 3y_1 + 2y_2 \geq 6 & & \text{(Chocolate constraint)} \\
\quad 2y_1 + 2y_2 + 4y_3 + 4y_4 \geq 10 & & \text{(Sugar constraint)} \\
\quad 2y_1 + 4y_2 + y_3 + 5y_4 \geq 8 & & \text{(Fat constraint)} \\
\quad y_1, y_2, y_3, y_4 \geq 0 & &
\end{array}$$

y_1 = number of brownies eaten daily

y_2 = number of scoops of chocolate ice cream eaten daily

y_3 = bottles of soda drunk daily

y_4 = pieces of pineapple cheesecake eaten daily

Dual:

$$\begin{array}{l}
\max z = 500x_1 + 6x_2 + 10x_3 + 8x_4 \\
\text{s.t.} \quad 400x_1 + 3x_2 + 2x_3 + 2x_4 \leq 50 \\
\quad 200x_1 + 2x_2 + 2x_3 + 4x_4 \leq 20 \\
\quad 150x_1 + 4x_3 + x_4 \leq 30 \\
\quad 500x_1 + 4x_3 + 5x_4 \leq 80 \\
\quad x_1, x_2, x_3, x_4 \geq 0
\end{array}$$

4.2 Finding the Dual of a Nonnormal Max or Min Problem

Fortunately, an LP can be transformed into normal form. To place a max problem into normal form, we proceed as follows:

Step 1 Multiply each >= constraint by -1, converting it into a <= constraint.

Step 2 Replace each equality constraint by two inequality constraints (a <= constraint and a >= constraint). Then convert the >= constraint to a <= constraint.

Step 3 Replace each urs variable x_i by $x_i = x_i' - x_i''$, where $x_i' >= 0$ and $x_i'' >= 0$.

If the primal is not a normal min problem, then we can transform it into a normal min problem as follows

Step 1 Convert each <= constraint into a >= constraint by multiplying through by -1.

Step 2 Replace each equality constraint by a <= constraint and a >= constraint. Then transform the >= constraint to a <= constraint

Step 3 Replace any urs variable y_i by $y_i = y_i' - y_i''$, where $y_i' >= 0$ and $y_i'' >= 0$.

$$
\begin{array}{ll}
\min w = 2y_1 + 4y_2 + 6y_3 & \max z = 2x_1 + x_2' - x_2'' \\
\text{s.t.} \quad y_1 + 2y_2 + y_3 \geq 2 & \text{s.t.} \quad x_1 + x_2' - x_2'' \leq 2 \\
\quad y_1 - y_3 \geq 1 & \quad -x_1 - x_2' + x_2'' \leq -2 \\
\quad y_2 + y_3 = 1 & \quad -2x_1 + x_2' - x_2'' \leq -3 \\
\quad 2y_1 + y_2 \leq 3 & \quad x_1 - x_2' + x_2'' \leq 1 \\
\quad y_1 \text{ urs}, y_2, y_3 \geq 0 & \quad x_1, x_2', x_2'' \geq 0
\end{array}
$$

4.3 Economic Interpretation of the Dual

In linear programming, the dual problem provides a meaningful economic interpretation of the primal problem by assigning values to the resources represented by the constraints. The variables of the dual problem, known as dual variables**, represent the implicit or marginal value of resources in the optimal solution.** These values indicate how much the objective function of the primal problem would improve if the availability of a particular resource were increased by one unit, assuming the current optimal basis remains unchanged.

Dual variables are numerically equal to the shadow prices of the corresponding primal constraints and therefore play a critical role in sensitivity analysis. **A positive dual variable indicates that the associated resource is scarce and fully utilized, while a zero dual variable implies that the resource has excess capacity and does not contribute to improving the objective value**. The objective function of the dual problem represents the minimum total cost of acquiring the required resources needed to satisfy the constraints of the primal problem. Thus, the dual formulation provides valuable insight into resource allocation efficiency and supports managerial decision-making by quantifying the economic worth of limited resources.

Putting everything together, we see that the solution to the dual of the Dakota problem does yield prices for lumber, finishing hours, and carpentry hours.

The dual of the Dakota problem is

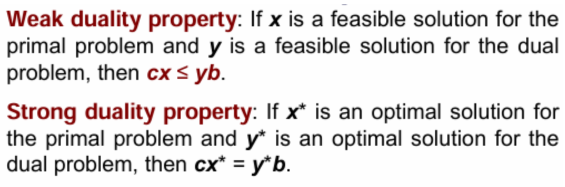
$$
\begin{array}{lll}
\min w = 48y_1 + 20y_2 + 8y_3 & & \\
\text{s.t.} \quad 8y_1 + 4y_2 + 2y_3 \geq 60 & \text{(Desk constraint)} & \\
\quad 6y_1 + 2y_2 + 1.5y_3 \geq 30 & \text{(Table constraint)} & y_1 = \text{price paid for 1 board ft of lumber} \\
\quad y_1 + 1.5y_2 + 0.5y_3 \geq 20 & \text{(Chair constraint)} & y_2 = \text{price paid for 1 finishing hour} \\
\quad y_1, y_2, y_3 \geq 0 & & y_3 = \text{price paid for 1 carpentry hour}
\end{array}
$$

Complementary slackness conditions state:

- If a primal constraint has slack, the corresponding dual variable is zero
- If a dual constraint has slack, the corresponding primal variable is zero

This explains why only binding constraints have nonzero shadow prices.

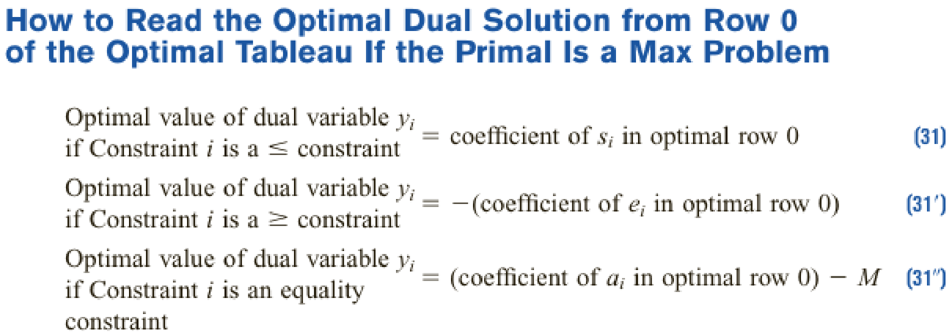
4.4 Duality Theorems



Weak duality property: If ***x*** is a feasible solution for the primal problem and ***y*** is a feasible solution for the dual problem, then $\boldsymbol{cx} \le \boldsymbol{yb}$.

Strong duality property: If $\boldsymbol{x}^*$ is an optimal solution for the primal problem and $\boldsymbol{y}^*$ is an optimal solution for the dual problem, then $\boldsymbol{cx}^* = \boldsymbol{y}^*\boldsymbol{b}$.

- If one problem has feasible solutions and a bounded objective function (and so has an optimal solution), then so does the other problem, so both the weak and strong duality properties are applicable.
- If one problem has feasible solutions and an unbounded objective function (and so no optimal solution), then the other problem has no feasible solutions.
- If one problem has no feasible solutions, then the other problem has either no feasible solutions or an unbounded objective function.

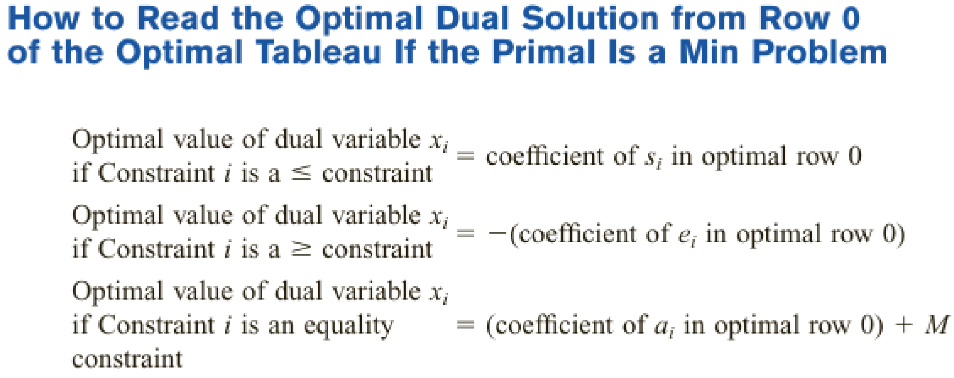


How to Read the Optimal Dual Solution from Row 0 of the Optimal Tableau If the Primal Is a Max Problem

$$\begin{matrix}\text{Optimal value of dual variable } y_i \\ \text{if Constraint } i \text{ is a} \le \text{constraint}\end{matrix} = \text{coefficient of } s_i \text{ in optimal row 0} \tag{31}$$

$$\begin{matrix}\text{Optimal value of dual variable } y_i \\ \text{if Constraint } i \text{ is a} \ge \text{constraint}\end{matrix} = -(\text{coefficient of } e_i \text{ in optimal row 0}) \tag{31$'$}$$

$$\begin{matrix}\text{Optimal value of dual variable } y_i \\ \text{if Constraint } i \text{ is an equality} \\ \text{constraint}\end{matrix} = (\text{coefficient of } a_i \text{ in optimal row 0}) - M \tag{31$''$}$$



How to Read the Optimal Dual Solution from Row 0 of the Optimal Tableau If the Primal Is a Min Problem

$$\begin{matrix}\text{Optimal value of dual variable } x_i \\ \text{if Constraint } i \text{ is a} \le \text{constraint}\end{matrix} = \text{coefficient of } s_i \text{ in optimal row 0}$$

$$\begin{matrix}\text{Optimal value of dual variable } x_i \\ \text{if Constraint } i \text{ is a} \ge \text{constraint}\end{matrix} = -(\text{coefficient of } e_i \text{ in optimal row 0})$$

$$\begin{matrix}\text{Optimal value of dual variable } x_i \\ \text{if Constraint } i \text{ is an equality} \\ \text{constraint}\end{matrix} = (\text{coefficient of } a_i \text{ in optimal row 0}) + M$$

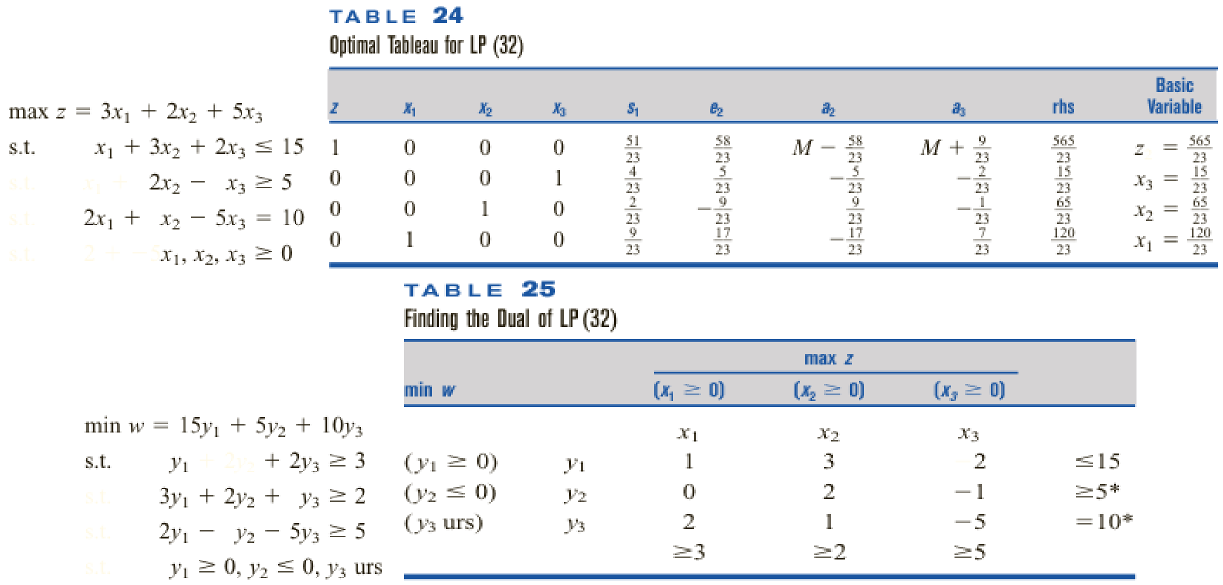


TABLE 24

Optimal Tableau for LP (32)

$$\begin{aligned} \max z &= 3x_1 + 2x_2 + 5x_3 \\ \text{s.t.} \quad x_1 + 3x_2 + 2x_3 &\le 15 \\ 2x_2 - x_3 &\ge 5 \\ 2x_1 + x_2 - 5x_3 &= 10 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

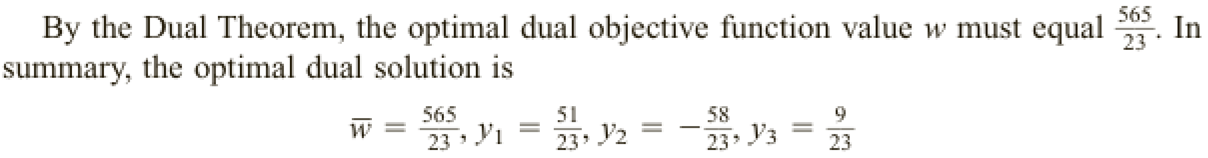
z	x_1	x_2	x_3	s_1	e_2	a_2	a_3	rhs	Basic Variable
1	0	0	0	$\frac{51}{23}$	$\frac{58}{23}$	$M - \frac{58}{23}$	$M + \frac{9}{23}$	$\frac{565}{23}$	$z = \frac{565}{23}$
0	0	0	1	$\frac{4}{23}$	$\frac{5}{23}$	$-\frac{5}{23}$	$-\frac{2}{23}$	$\frac{15}{23}$	$x_3 = \frac{15}{23}$
0	0	1	0	$\frac{2}{23}$	$-\frac{9}{23}$	$\frac{9}{23}$	$-\frac{1}{23}$	$\frac{65}{23}$	$x_2 = \frac{65}{23}$
0	1	0	0	$\frac{9}{23}$	$\frac{17}{23}$	$-\frac{17}{23}$	$\frac{7}{23}$	$\frac{120}{23}$	$x_1 = \frac{120}{23}$

TABLE 25

Finding the Dual of LP (32)

$$\begin{aligned} \min w &= 15y_1 + 5y_2 + 10y_3 \\ \text{s.t.} \quad y_1 + 2y_3 &\ge 3 \\ 3y_1 + 2y_2 + y_3 &\ge 2 \\ 2y_1 - y_2 - 5y_3 &\ge 5 \\ y_1 \ge 0, y_2 \le 0, &\ y_3 \text{ urs} \end{aligned}$$

		max z			
min w		($x_1 \ge 0$)	($x_2 \ge 0$)	($x_3 \ge 0$)	
		x_1	x_2	x_3	
($y_1 \ge 0$)	y_1	1	3	2	≤ 15
($y_2 \le 0$)	y_2	0	2	-1	≥ 5*
(y_3 urs)	y_3	2	1	-5	$= 10$*
		≥ 3	≥ 2	≥ 5	



By the Dual Theorem, the optimal dual objective function value w must equal $\frac{565}{23}$. In summary, the optimal dual solution is

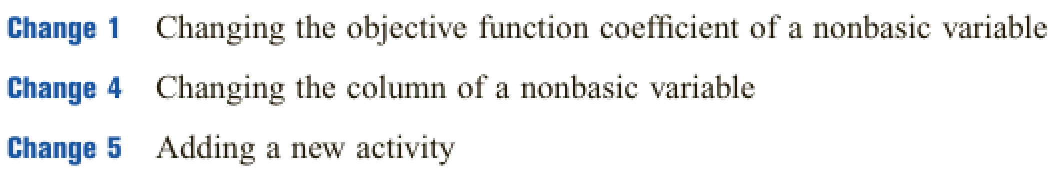
$$\bar{w} = \frac{565}{23}, y_1 = \frac{51}{23}, y_2 = -\frac{58}{23}, y_3 = \frac{9}{23}$$

4.5 Sensitivity Analysis from the Dual Perspective

- Changes in objective coefficients affect **dual constraint feasibility**
- Changes in RHS values affect the **dual objective function**
- Allowable ranges ensure dual feasibility, preserving primal optimality

This provides the theoretical foundation for LINDO’s sensitivity output.

Our proof of the Dual Theorem demonstrated the following result: Assuming that a set of basic variables BV is feasible, then BV is optimal (that is, each variable in row0 has a non-negative coefficient) if and only if the associated dual solution is dual feasible.



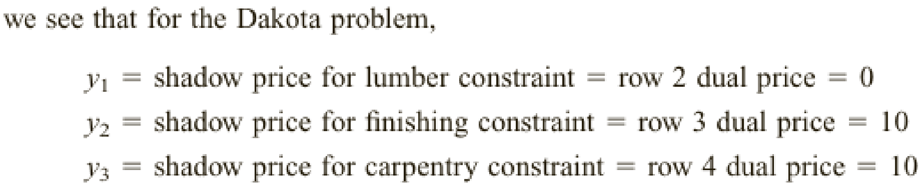
Change 1 Changing the objective function coefficient of a nonbasic variable

Change 4 Changing the column of a nonbasic variable

Change 5 Adding a new activity

4.6 Interpretation of Duality in LINDO Output

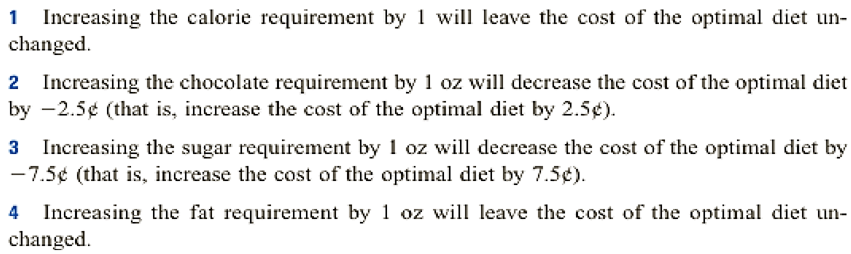
For a max problem, LINDO gives the values of the shadow prices in the DUAL PRICES column of the output. The dual price for row i+1 on the LINDO output is the shadow price for the ith constraint and the optimal value for the ith dual variable.



we see that for the Dakota problem,

$$y_1 = \text{shadow price for lumber constraint} = \text{row 2 dual price} = 0$$
$$y_2 = \text{shadow price for finishing constraint} = \text{row 3 dual price} = 10$$
$$y_3 = \text{shadow price for carpentry constraint} = \text{row 4 dual price} = 10$$

For a minimization problem, the entry in the DUAL PRICE column for any constraint is the shadow price. Thus, from the LINDO printout in Figure 6, we find that the shadow prices for the constraints in the diet problem are as follows: calorie = 0; chocolate = 2.5¢; sugar = 7.5¢; and fat = 0. This implies that

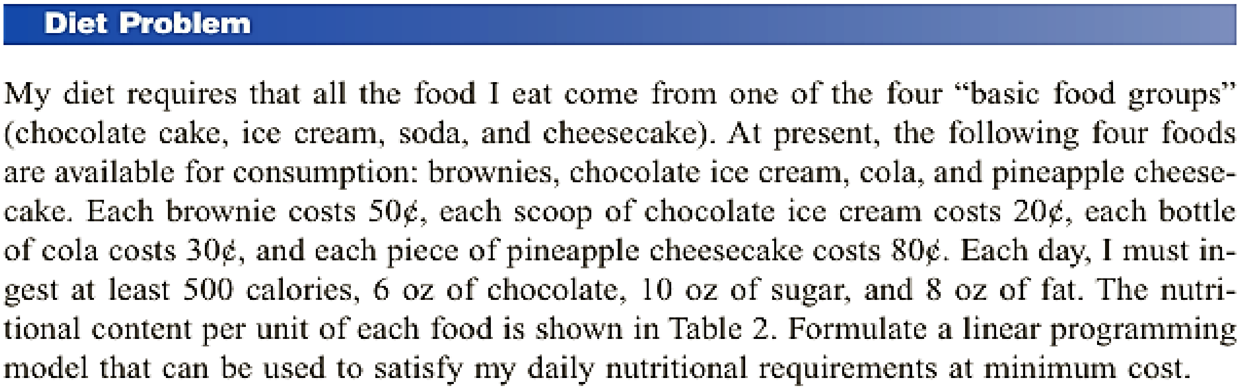


1 Increasing the calorie requirement by 1 will leave the cost of the optimal diet unchanged.

2 Increasing the chocolate requirement by 1 oz will decrease the cost of the optimal diet by −2.5¢ (that is, increase the cost of the optimal diet by 2.5¢).

3 Increasing the sugar requirement by 1 oz will decrease the cost of the optimal diet by −7.5¢ (that is, increase the cost of the optimal diet by 7.5¢).

4 Increasing the fat requirement by 1 oz will leave the cost of the optimal diet unchanged.



Diet Problem

My diet requires that all the food I eat come from one of the four "basic food groups" (chocolate cake, ice cream, soda, and cheesecake). At present, the following four foods are available for consumption: brownies, chocolate ice cream, cola, and pineapple cheesecake. Each brownie costs 50¢, each scoop of chocolate ice cream costs 20¢, each bottle of cola costs 30¢, and each piece of pineapple cheesecake costs 80¢. Each day, I must ingest at least 500 calories, 6 oz of chocolate, 10 oz of sugar, and 8 oz of fat. The nutritional content per unit of each food is shown in Table 2. Formulate a linear programming model that can be used to satisfy my daily nutritional requirements at minimum cost.

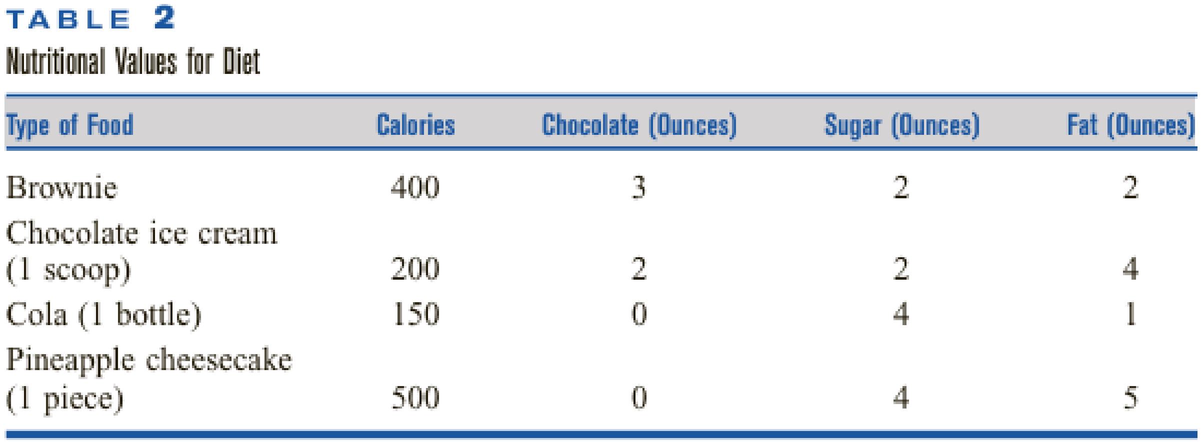
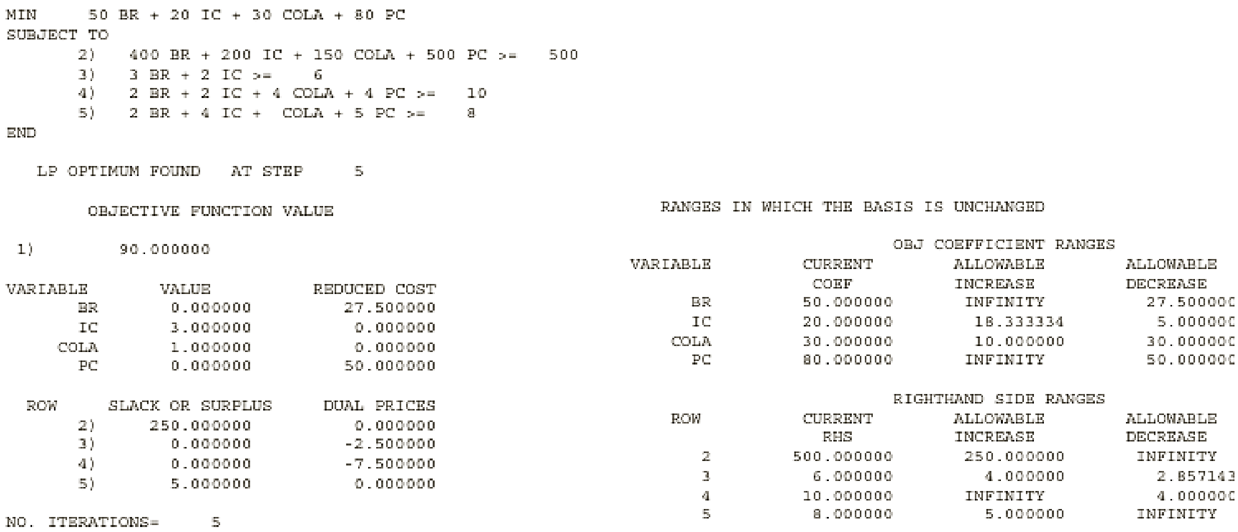


TABLE 2
Nutritional Values for Diet

Type of Food	Calories	Chocolate (Ounces)	Sugar (Ounces)	Fat (Ounces)
Brownie	400	3	2	2
Chocolate ice cream (1 scoop)	200	2	2	4
Cola (1 bottle)	150	0	4	1
Pineapple cheesecake (1 piece)	500	0	4	5



```
MIN      50 BR + 20 IC + 30 COLA + 80 PC
SUBJECT TO
       2)    400 BR + 200 IC + 150 COLA + 500 PC >=    500
       3)    3 BR + 2 IC >=     6
       4)    2 BR + 2 IC + 4 COLA + 4 PC >=    10
       5)    2 BR + 4 IC +  COLA + 5 PC >=     8
END

   LP OPTIMUM FOUND   AT STEP      5

        OBJECTIVE FUNCTION VALUE

 1)         90.000000

VARIABLE        VALUE          REDUCED COST
       BR        0.000000         27.500000
       IC        3.000000          0.000000
     COLA        1.000000          0.000000
       PC        0.000000         50.000000

  ROW     SLACK OR SURPLUS     DUAL PRICES
       2)      250.000000          0.000000
       3)        0.000000         -2.500000
       4)        0.000000         -7.500000
       5)        5.000000          0.000000

NO. ITERATIONS=      5
```

```
 RANGES IN WHICH THE BASIS IS UNCHANGED

                           OBJ COEFFICIENT RANGES
VARIABLE         CURRENT         ALLOWABLE         ALLOWABLE
                  COEF           INCREASE          DECREASE
      BR        50.000000        INFINITY           27.50000
      IC        20.000000         18.333334          5.00000
    COLA        30.000000         10.000000         30.00000
      PC        80.000000        INFINITY           50.00000

                           RIGHTHAND SIDE RANGES
     ROW         CURRENT         ALLOWABLE         ALLOWABLE
                  RHS            INCREASE          DECREASE
       2       500.000000        250.000000        INFINITY
       3         6.000000          4.000000          2.85714
       4        10.000000        INFINITY            4.00000
       5         8.000000          5.000000        INFINITY
```

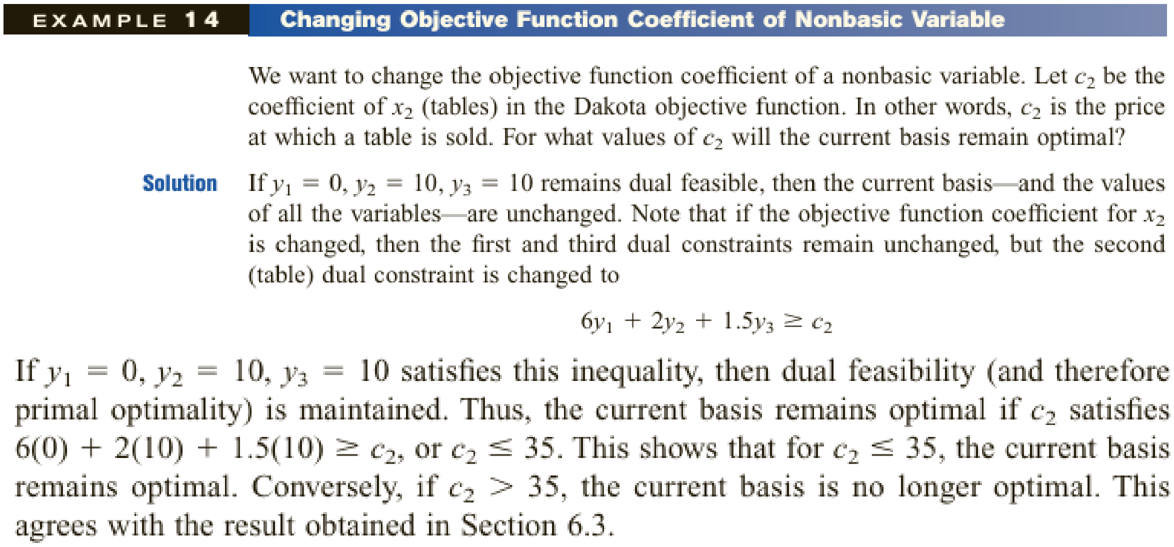
In LINDO:

- **Dual prices** are the optimal values of dual variables
- Dual prices = shadow prices
- Slack values verify complementary slackness numerically

If allowable ranges are violated:

- Dual feasibility may be lost
- Shadow prices and reduced costs become invalid
- Re-optimization is required

For Dakota problem:



EXAMPLE 14 **Changing Objective Function Coefficient of Nonbasic Variable**

We want to change the objective function coefficient of a nonbasic variable. Let c_2 be the coefficient of x_2 (tables) in the Dakota objective function. In other words, c_2 is the price at which a table is sold. For what values of c_2 will the current basis remain optimal?

Solution If $y_1 = 0$, $y_2 = 10$, $y_3 = 10$ remains dual feasible, then the current basis—and the values of all the variables—are unchanged. Note that if the objective function coefficient for x_2 is changed, then the first and third dual constraints remain unchanged, but the second (table) dual constraint is changed to

$$6y_1 + 2y_2 + 1.5y_3 \geq c_2$$

If $y_1 = 0$, $y_2 = 10$, $y_3 = 10$ satisfies this inequality, then dual feasibility (and therefore primal optimality) is maintained. Thus, the current basis remains optimal if c_2 satisfies $6(0) + 2(10) + 1.5(10) \geq c_2$, or $c_2 \leq 35$. This shows that for $c_2 \leq 35$, the current basis remains optimal. Conversely, if $c_2 > 35$, the current basis is no longer optimal. This agrees with the result obtained in Section 6.3.

Using shadow prices, we may give an alternative interpretation of this result. We can use shadow prices to compute the implied value of the resources needed to construct a table (see Table 30). A table uses \$35 worth of resources, so the only way producing tables can increase Dakota's revenues is if a table sells for more than \$35. Thus, the current basis fails to be optimal if $c_2 > 35$, and the current basis remains optimal if $c_2 \leq 35$.

TABLE 30

Why a Table Is Profitable at > \$35/Table

Resource in a Table	Shadow Price of Resource (\$)	Amount of Resource Used	Value of Resource Used
Lumber	0	6 board ft	0(6) = \$0
Finishing	10	2 hours	10(2) = \$20
Carpentry	10	1.5 hours	10(1.5) = \$15
			Total: = \$35

EXAMPLE 15 Changing a Nonbasic Variable

We want to change the column for a nonbasic activity. Suppose a table sells for \$43 and uses 5 board feet of lumber, 2 finishing hours, and 2 carpentry hours. Does the current basis remain optimal?

Solution Changing the column for the nonbasic variable "tables" leaves the first and third dual constraints unchanged but changes the second to

$$5y_1 + 2y_2 + 2y_3 \geq 43$$

Because $y_1 = 0$, $y_2 = 10$, $y_3 = 10$ does not satisfy the new second dual constraint, dual feasibility is not maintained, and the current basis is no longer optimal. In terms of shadow prices, this result is reasonable (see Table 31). Each table uses \$40 worth of resources and sells for \$43, so Dakota can increase its revenue by $43 - 40 = \$3$ for each table that is produced. Thus, the current basis is no longer optimal, and x_2 (tables) will be basic in the new optimal solution.

TABLE 31

Shadow Price Interpretation of Table Production Decision (\$40/Table)

Resource in a Table	Shadow Price of Resource (\$)	Amount of Resource Used	Value of Resource Used (\$)
Lumber	0	5 board ft	0(5) = \$0
Finishing	10	2 hours	10(2) = \$20
Carpentry	10	2 hours	10(2) = \$20
			Total: = \$40

EXAMPLE 16 Adding a New Activity

We want to add a new activity. Suppose Dakota is considering manufacturing footstools (x_4). A footstool sells for \$15 and uses 1 board foot of lumber, 1 finishing hour, and 1 carpentry hour. Does the current basis remain optimal?

Solution Introducing the new activity (footstools) leaves the three dual constraints unchanged, but the new variable x_4 adds a new dual constraint (corresponding to footstools). The new dual constraint will be

$$y_1 + y_2 + y_3 \geq 15$$

The current basis remains optimal if $y_1 = 0$, $y_2 = 10$, $y_3 = 10$ satisfies the new dual constraint. Because $0 + 10 + 10 \geq 15$, the current basis remains optimal. In terms of shadow prices, a stool utilizes $1(0) = \$0$ worth of lumber, $1(10) = \$10$ worth of finishing hours, and $1(10) = \$10$ worth of carpentry time. A stool uses $0 + 10 + 10 = \$20$ worth of resources and sells for only \$15, so Dakota should not make footstools, and the current basis remains optimal.

5. Practice

6 Sugarco can manufacture three types of candy bar. Each candy bar consists totally of sugar and chocolate. The compositions of each type of candy bar and the profit earned from each candy bar are shown in Table 10. Fifty oz of sugar and 100 oz of chocolate are available. After defining x_i to be the number of Type i candy bars manufactured, Sugarco should solve the following LP:

$$\begin{array}{lll} \max z = 3x_1 + 7x_2 + 5x_3 & & \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 50 & \text{(Sugar constraint)} \\ & 2x_1 + 3x_2 + x_3 \leq 100 & \text{(Chocolate constraint)} \\ & x_1, x_2, x_3 \geq 0 & \end{array}$$

TABLE 10

Bar	Amount of Sugar (Ounces)	Amount of Chocolate (Ounces)	Profit (Cents)
1	1	2	3
2	1	3	7
3	1	1	5

2 The following questions refer to the Sugarco problem (Problem 6 of Section 6.3):

- **a** For what values of profit on a Type 1 candy bar does the current basis remain optimal?
- **b** If a Type 1 candy bar used 0.5 oz of sugar and 0.75 oz of chocolate, would the current basis remain optimal?
- **c** A Type 4 candy bar is under consideration. A Type 4 candy bar yields a 10¢ profit and uses 2 oz of sugar and 1 oz of chocolate. Does the current basis remain optimal?

1 Farmer Leary grows wheat and corn on his 45-acre farm. He can sell at most 140 bushels of wheat and 120 bushels of corn. Each acre planted with wheat yields 5 bushels, and each acre planted with corn yields 4 bushels. Wheat sells for \$30 per bushel, and corn sells for \$50 per bushel. To harvest an acre of wheat requires 6 hours of labor; 10 hours are needed to harvest an acre of corn. Up to 350 hours of labor can be purchased at \$10 per hour. Let A1 = acres planted with wheat; A2 = acres planted with corn; and L = hours of labor that are purchased. To maximize profits, Leary should solve the following LP:

$$\begin{array}{ll} \max z = 150\text{A1} + 200\text{A2} - 10\text{L} & \\ \text{s.t.} & \text{A1} + \text{A2} \leq 45 \\ & 6\text{A1} + 10\text{A2} - \text{L} \leq 0 \\ & \text{L} \leq 350 \\ & 5\text{A1} \leq 140 \\ & 4\text{A2} \leq 120 \\ & \text{A1, A2, L} \geq 0 \end{array}$$

Use the LINDO output in Figure 6 to answer the following questions:

- **a** If only 40 acres of land were available, what would Leary's profit be?
- **b** If the price of wheat dropped to \$26, what would be the new optimal solution to Leary's problem?
- **c** Use the SLACK portion of the output to determine the allowable increase and allowable decrease for the amount of wheat that can be sold. If only 130 bushels of wheat could be sold, then would the answer to the problem change?

6 Steelco uses coal, iron, and labor to produce three types of steel. The inputs (and sales price) for one ton of each type of steel are shown in Table 8. Up to 200 tons of coal can be purchased at a price of \$10 per ton. Up to 60 tons of iron can be purchased at \$8 per ton, and up to 100 labor hours can be purchased at \$5 per hour. Let x_1 = tons of steel 1 produced; x_2 = tons of steel 2 produced; and x_3 = tons of steel 3 produced.

The LINDO output that yields a maximum profit for the company is given in Figure 11. Use the output to answer the following questions.

a What would profit be if only 40 tons of iron could be purchased?

b What is the smallest price per ton for steel 3 that would make it desirable to produce it?

c Find the new optimal solution if steel 1 sold for \$55 per ton.

11 Consider the LP:

$$\begin{array}{ll}
\max z = 9x_1 + 8x_2 + 5x_3 + 4x_4 & \\
\text{s.t.} \quad x_1 + x_4 \le 200 & \\
\quad x_2 + x_3 \le 150 & \\
\quad x_1 + x_2 + x_3 \le 350 & \\
\quad 2x_1 + x_2 + x_3 + x_4 \le 550 & \\
\quad x_1, x_2, x_3, x_4 \ge 0 &
\end{array}$$

a Solve this LP with LINDO and use your output to show that the optimal solution is degenerate.

Experiment- 4: Solving unconstrained and constrained nonlinear optimization problem

1. Objective

By the end of this experiment, students will be able to:

- Formulate nonlinear optimization problems.
- Solve unconstrained nonlinear optimization problems using LINGO and MATLAB.
- Solve constrained nonlinear optimization problems using LINGO and MATLAB

2. Nonlinear Optimization

Nonlinear Programming (NLP) deals with optimization problems in which the objective function and/or at least one constraint is nonlinear. Unlike linear programming, NLP problems may have multiple local optima, saddle points, or no optimal solution at all.

A general nonlinear programming problem is expressed as:

Minimize or Maximize:

$f(x) = f(x_1, x_2, \ldots, x_n)$

Subject to:

$g_i(x) \leq 0, i = 1, 2, \ldots, m$
$h_j(x) = 0, j = 1, 2, \ldots, p$

where $x \in \mathbb{R}^n$ is the vector of decision variables.

3. Nonlinear Unconstrained Optimization

In unconstrained optimization, the objective function is optimized without any explicit constraints.

3.1 Necessary Condition

If f(x) has a local minimum at x* then ∇f(x*) = 0, where ∇f is the gradient vector. Geometrically, the gradient vector is normal to the tangent plane at the point x* as shown in Figure 4.1 for a function of three variables. Also, it points in the direction of maximum increase in the function.

$$\nabla f(\mathbf{x}^*) = \begin{bmatrix} \dfrac{\partial f(\mathbf{x}^*)}{\partial x_1} \\ \dfrac{\partial f(\mathbf{x}^*)}{\partial x_2} \\ \vdots \\ \dfrac{\partial f(\mathbf{x}^*)}{\partial x_n} \end{bmatrix}$$

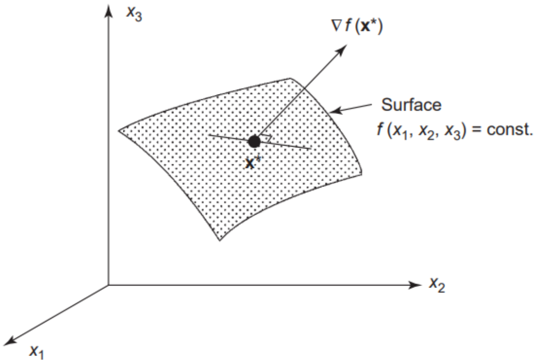


Figure 3.1.1: Visualization of gradient vector

3.2 Second-Order Conditions

Let H be the Hessian matrix of *f(x)*, where H is represented by,

$$\frac{\partial^2 f}{\partial \mathbf{x} \partial \mathbf{x}} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

- If H is positive definite at x*, then x* is a local minimum.
- If H is negative definite at x*, then x* is a local maximum.
- If H is indefinite, then x* is a saddle point.

4. Nonlinear Constrained Optimization

When constraints are present, stationary points must satisfy additional optimality conditions.

4.1 KKT Condition

Let x* be a regular point of the feasible set that is a local minimum for f(x) subject to $h_i(x) = 0$; i = 1 to p; $g_j(x) <= 0$; j = 1 to m. Then there exist Lagrange multipliers *v** (a p-vector) and *u** (an m-vector) such that the Lagrangian function is stationary with respect to x_j, v_i, u_j, and s_j at the point x*.

1. Lagrangian Function

$$L(\mathbf{x},\mathbf{v},\mathbf{u},\mathbf{s}) = f(\mathbf{x}) + \sum_{i=1}^{p} v_i h_i(\mathbf{x}) + \sum_{j=1}^{m} u_j (g_j(\mathbf{x}) + s_j^2) = f(\mathbf{x}) + \mathbf{v}^{\mathrm{T}}\mathbf{h}(\mathbf{x}) + \mathbf{u}^{\mathrm{T}}(\mathbf{g}(\mathbf{x}) + \mathbf{s}^2)$$

2. Gradient Conditions

$$\frac{\partial L}{\partial x_k} = \frac{\partial f}{\partial x_k} + \sum_{i=1}^{p} v_i^* \frac{\partial h_i}{\partial x_k} + \sum_{j=1}^{m} u_j^* \frac{\partial g_j}{\partial x_k} = 0; \quad k = 1 \text{ to } n$$

$$\frac{\partial L}{\partial v_i} = 0 \Rightarrow h_i(\mathbf{x}^*) = 0; \quad i = 1 \text{ to } p$$

$$\frac{\partial L}{\partial u_j} = 0 \Rightarrow (g_j(\mathbf{x}^*) + s_j^2) = 0; \quad j = 1 \text{ to } m$$

3. Feasibility Check for Inequalities

$$s_j^2 \geq 0; \text{ or equivalently } g_j \leq 0; \quad j = 1 \text{ to } m$$

4. Switching Conditions

$$\frac{\partial L}{\partial s_j} = 0 \Rightarrow 2u_j^* s_j = 0; \quad j = 1 \text{ to } m$$

5. Nonnegativity of Lagrange Multipliers for Inequalities

$$u_j^* \geq 0; \quad j = 1 \text{ to } m$$

6. Regularity Check
   Gradients of active constraints should be linearly independent. In such a case the Lagrange multipliers for the constraints are unique.

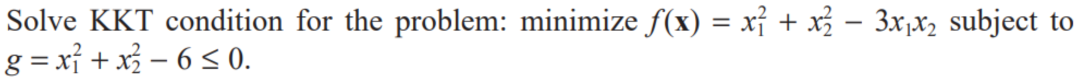
Exercise Problems

Exercise 4.1 – NLP of several variables using MATLAB: A monopolist producing a single product has two types of customers. If q1 units are produced for customer 1, then customer 1 is willing to pay a price of 70 - 4q1 dollars. If q2 units are produced for customer 2, then customer 2 is willing to pay a price of 150 - 15q2 dollars. For q >0, the cost of manufacturing q units is 100 +15q dollars. To maximize profit, how much should the monopolist sell to each customer? Use MATLAB to determine your result.

Solution:

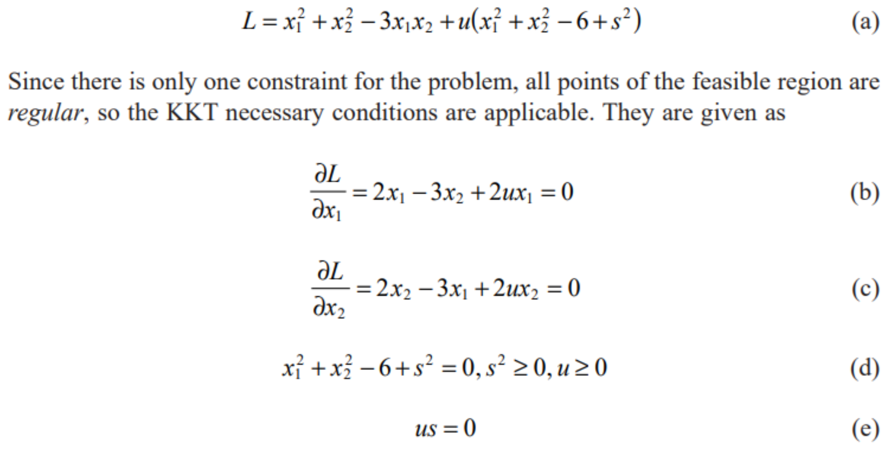
```
%% Monopolist Profit Maximization
clc; clear;
syms q1 q2

% Profit function
pi = (70*q1 - 4*q1^2) + (150*q2 - 15*q2^2) - (100 + 15*(q1+q2));

% Gradient and stationary point
grad = gradient(pi, [q1,q2]);
sol = solve(grad==0, [q1,q2]);
q_star = double([sol.q1, sol.q2]);

% Hessian and eigenvalues
H_val = double(subs(hessian(pi,[q1,q2]), {q1,q2}, {q_star(1), q_star(2)}));
eig_H = eig(H_val);

% Display results if concave
disp('Gradient:'); disp(grad);
disp('Hessian:'); disp(H_val);
disp('Eigenvalues:'); disp(eig_H);

if all(eig_H<0)
    disp('Global maximum confirmed.');
    disp(['Optimal quantities: q1*=', num2str(q_star(1)), ', q2*=', num2str(q_star(2))]);

end
```

Exercise 4.2 Solving for KKT points in MATLAB



Solve KKT condition for the problem: minimize $f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$ subject to $g = x_1^2 + x_2^2 - 6 \le 0$.

Solution:



$$L = x_1^2 + x_2^2 - 3x_1x_2 + u(x_1^2 + x_2^2 - 6 + s^2) \qquad \text{(a)}$$

Since there is only one constraint for the problem, all points of the feasible region are *regular*, so the KKT necessary conditions are applicable. They are given as

$$\frac{\partial L}{\partial x_1} = 2x_1 - 3x_2 + 2ux_1 = 0 \qquad \text{(b)}$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - 3x_1 + 2ux_2 = 0 \qquad \text{(c)}$$

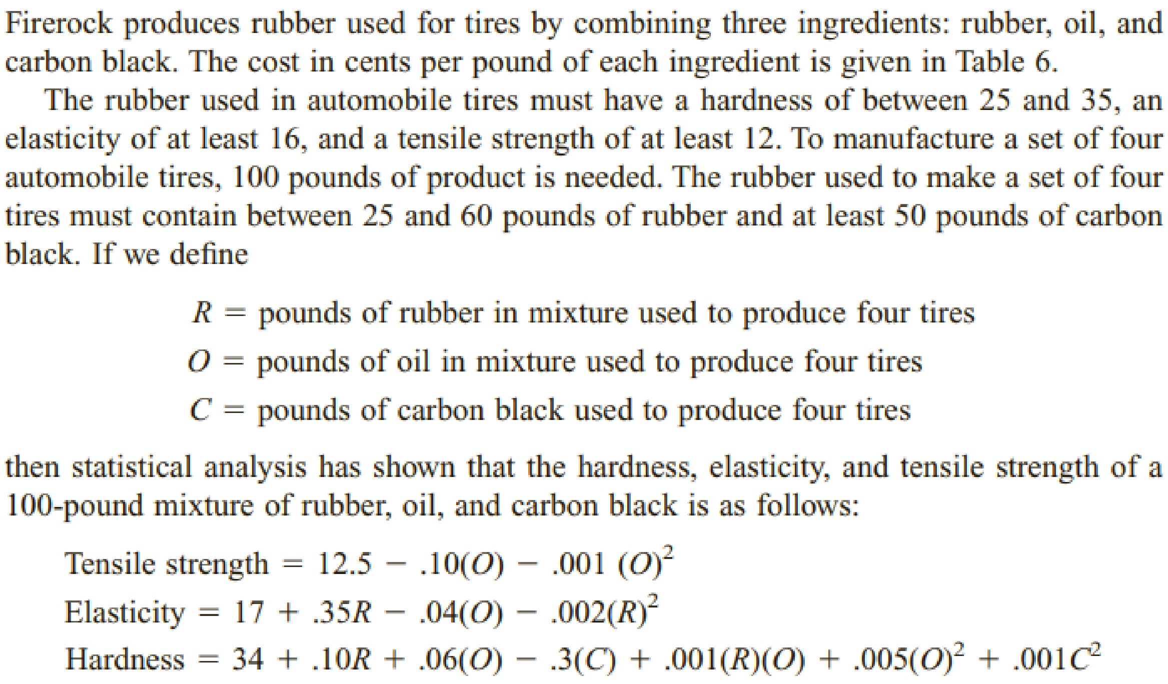
$$x_1^2 + x_2^2 - 6 + s^2 = 0, s^2 \ge 0, u \ge 0 \qquad \text{(d)}$$

$$us = 0 \qquad \text{(e)}$$

For the present example, components of the vector x are defined as x(1) = x1, x(2) = x2, x(3) = u, and x(4) = s. In terms of these variables, the KKT conditions of Eqs. (b) to (e) are written as a system of functions in MATLAB.

```
%% Solve KKT system using fsolve

clc; clear;

% Define the system as a function
fun = @(x) [2*x(1) - 3*x(2) + 2*x(3)*x(1);
            2*x(2) - 3*x(1) + 2*x(3)*x(2);
            x(1)^2 + x(2)^2 - 6 + x(4)^2;
            x(3)*x(4)];

% Initial guess
x0 = [1; 1; 1; 1];

% Solve using fsolve
[x_sol, fval] = fsolve(fun, x0);

% Display solution
disp('KKT point:');
disp(x_sol);
```

Exercise 4.3 Solving NLP using LINGO



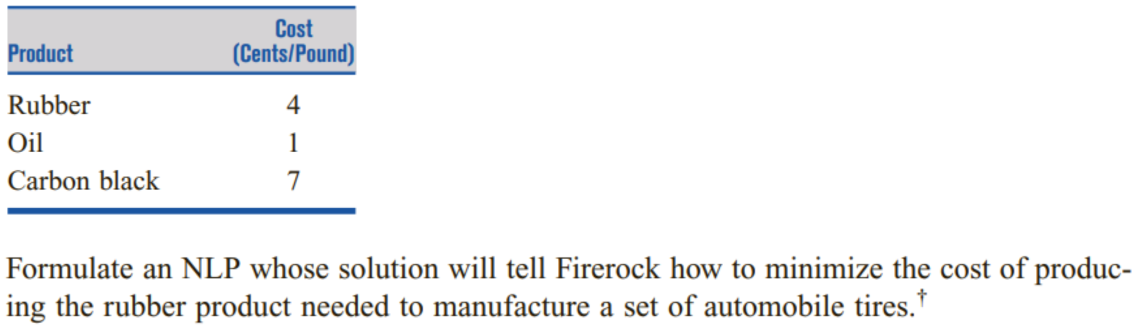
Firerock produces rubber used for tires by combining three ingredients: rubber, oil, and carbon black. The cost in cents per pound of each ingredient is given in Table 6.

The rubber used in automobile tires must have a hardness of between 25 and 35, an elasticity of at least 16, and a tensile strength of at least 12. To manufacture a set of four automobile tires, 100 pounds of product is needed. The rubber used to make a set of four tires must contain between 25 and 60 pounds of rubber and at least 50 pounds of carbon black. If we define

R = pounds of rubber in mixture used to produce four tires
O = pounds of oil in mixture used to produce four tires
C = pounds of carbon black used to produce four tires

then statistical analysis has shown that the hardness, elasticity, and tensile strength of a 100-pound mixture of rubber, oil, and carbon black is as follows:

$$\text{Tensile strength} = 12.5 - .10(O) - .001\ (O)^2$$
$$\text{Elasticity} = 17 + .35R - .04(O) - .002(R)^2$$
$$\text{Hardness} = 34 + .10R + .06(O) - .3(C) + .001(R)(O) + .005(O)^2 + .001C^2$$



Product	Cost (Cents/Pound)
Rubber	4
Oil	1
Carbon black	7

Formulate an NLP whose solution will tell Firerock how to minimize the cost of producing the rubber product needed to manufacture a set of automobile tires.[†]

Solution

LINGO CODE

```
MODEL:
! Decision variables (lb per 100-lb mixture);
! R = rubber, O = oil, C = carbon black;

MIN = 4*R + 1*O + 7*C;

! Mixture constraint;
R + O + C = 100;
! Bounds;
R >= 25;
R <= 60;
O >= 0;
C >= 50;
! Tensile strength constraint;
12.5 - 0.10*O - 0.001*O^2 >= 12;
! Elasticity constraint;
17 + 0.35*R - 0.04*O - 0.002*R^2 >= 16;

! Hardness (MINIMUM only — no upper bound);
34 + 0.10*R + 0.06*O - 0.3*C
 + 0.001*R*O + 0.005*O^2 + 0.001*C^2 >= 25;
34 + 0.10*R + 0.06*O - 0.3*C
 + 0.001*R*O + 0.005*O^2 + 0.001*C^2 <= 35;
END
```

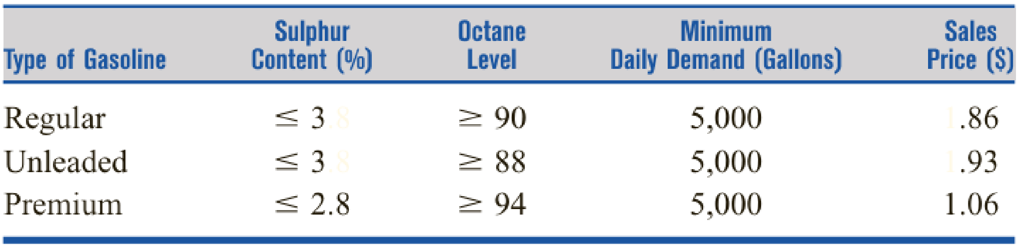
As it is not a convex programming problem, we cannot be sure that LINGO has found the optimal solution. To see if LINGO has actually found the optimal solution to the NLP, we have to use the INIT command to input a wide variety of starting solutions. however, extensive use of the INIT command failed to turn up any better solutions, so we are fairly confident that LINGO has found the optimal solution

```
! Manually add INIT points;
INIT: R = 20; O = 5; C = 75; ENDINIT
```

Practice problems

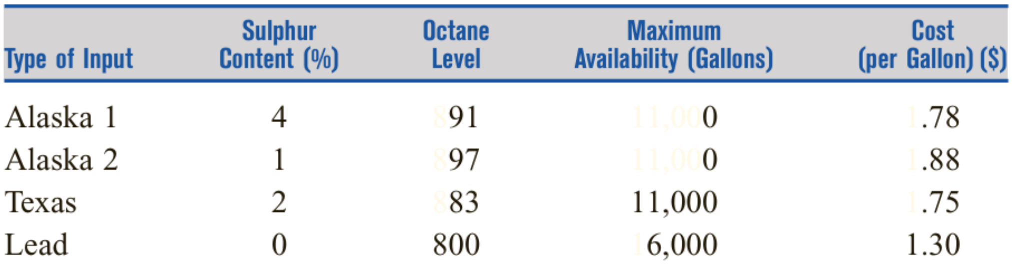
1. Oilco produces three types of gasoline: regular, unleaded, and premium. All three are produced by combining lead and crude oil brought in from Alaska and Texas. The required sulphur content, octane levels, minimum daily demand (in gallons), and sales price per gallon of each type of gasoline are given in the Table 4.1. The crude brought in from Alaska is made by blending two types of crude: Alaska1 and Alaska2. The Alaska crude is blended in Alaska and shipped via pipeline to Oilco's Texas refinery. At most, 10,000 gallons of crude per day can be shipped from Alaska. The sulphur content, octane level, daily maximum amount available (in gallons) and purchase cost (per gallon) for each type of Alaska crude, Texas crude, and lead are given in table 4.2. Of course, unleaded gasoline can contain no lead. Formulate an NLP to help Oilco maximize the daily profit obtained from selling gasoline. Use LINGO to determine your result.

Table 4.1: Gasoline Production data



Type of Gasoline	Sulphur Content (%)	Octane Level	Minimum Daily Demand (Gallons)	Sales Price ($)
Regular	≤ 3	≥ 90	5,000	.86
Unleaded	≤ 3	≥ 88	5,000	.93
Premium	≤ 2.8	≥ 94	5,000	1.06

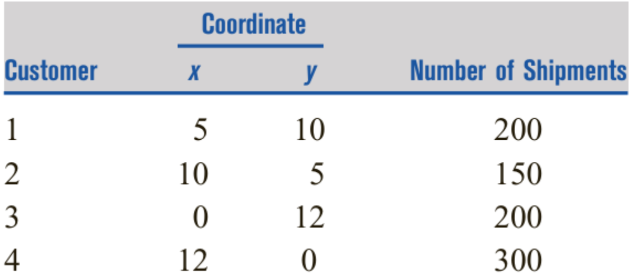
Table 4.2: Crude data



Type of Input	Sulphur Content (%)	Octane Level	Maximum Availability (Gallons)	Cost (per Gallon) ($)
Alaska 1	4	91	0	.78
Alaska 2	1	97	0	.88
Texas	2	83	11,000	.75
Lead	0	800	6,000	1.30

2**.** Truckco is trying to determine where it should locate a single warehouse. The positions in the x–y plane (in miles) of four customers and the number of shipments made annually to each customer are given in Table 5. Truckco wants to locate the warehouse to minimize the total distance trucks must travel annually from the warehouse to the four customers.

Table 4.3: Shipping distance and amount



	Coordinate		
Customer	x	y	Number of Shipments
1	5	10	200
2	10	5	150
3	0	12	200
4	12	0	300

Experiment-5: Solving the Transportation Problem Using LINDO.

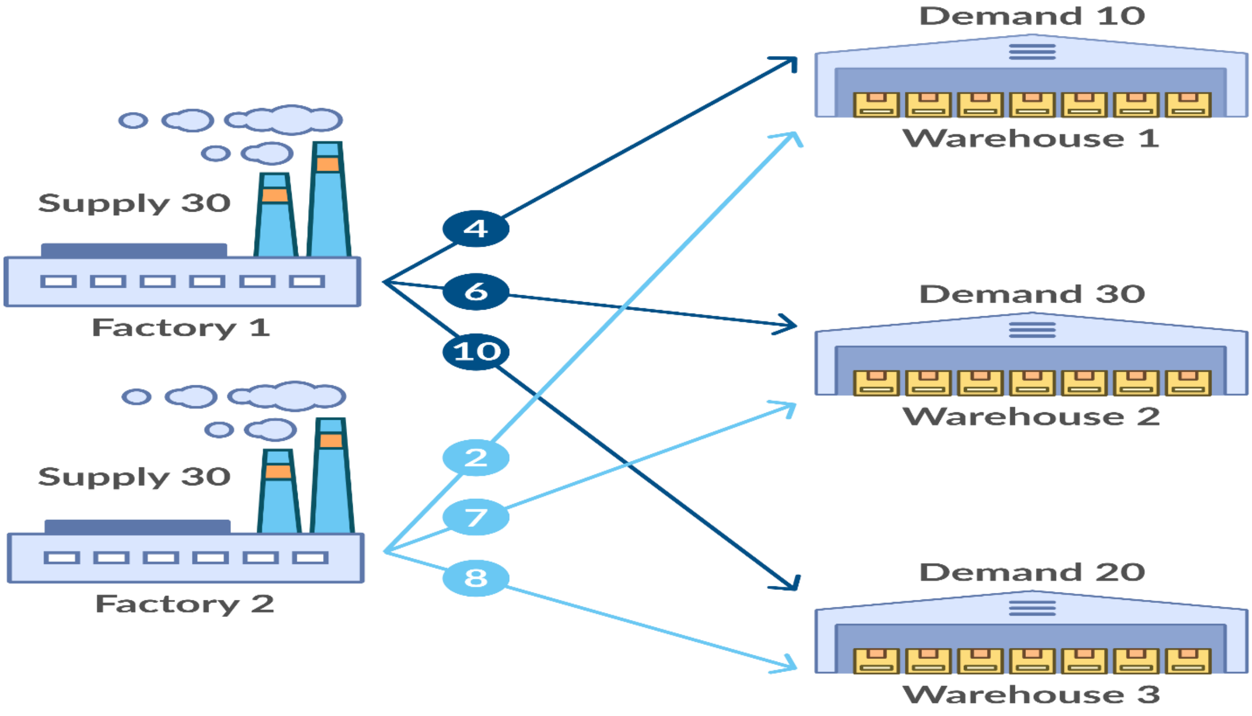
1. Objective

By the end of this experiment, students will be able to:

- Formulate transportation problems.
- Solve balanced and unbalanced transportation problems using LINDO.
- Solve different methods for initial basic feasible solution.

2. Introduction to Transportation Problem

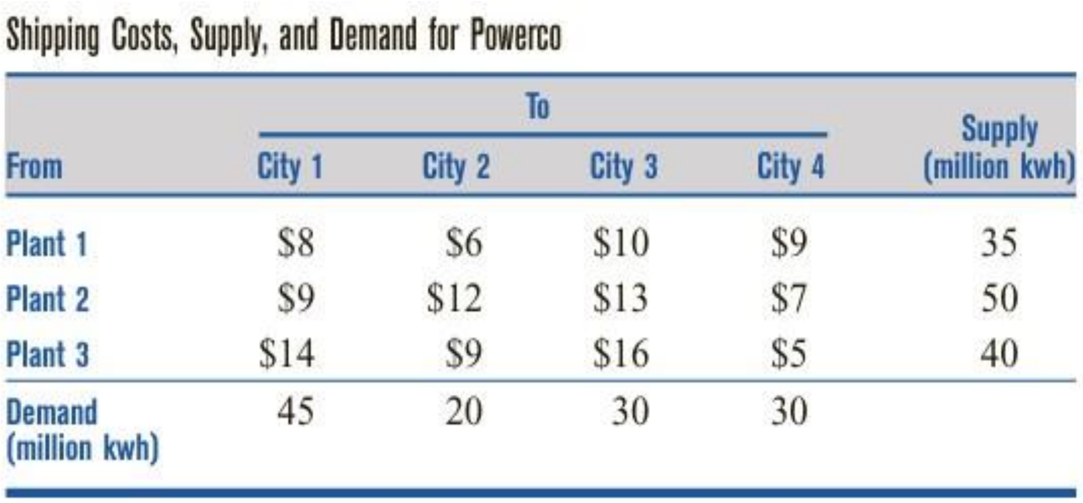
The transportation problem is to transport various amounts of a single homogeneous commodity that are initially stored at various origins, to different destinations in such a way that the total transportation cost is a minimum. It can also be defined as to ship goods from various origins to various destinations in such a manner that the transportation cost is a minimum. The availability as well as the requirements is finite. It is assumed that the cost of shipping is linear.



3. Balanced Transportation Problem Formulation

Example: Power Co has three electric power plants that supply the needs of four cities. Each power plant can supply the following numbers of kilowatt-hours (kwh) of electricity: plant 1— 35 million; plant 2—50 million; plant 3—40 million (see Table 1). The peak power demands in these cities, which occur at the same time (2 P.M.), are as follows (in kwh): city 1—45 million; city 2—20 million; city 3—30 million; city 4—30 million. The costs of sending 1 million kwh of electricity from plant to city depend on the distance the electricity must travel. Formulate an LP to minimize the cost of meeting each city's peak power demand.

Solution:



Shipping Costs, Supply, and Demand for Powerco

	To				
From	City 1	City 2	City 3	City 4	Supply (million kwh)
Plant 1	\$8	\$6	\$10	\$9	35
Plant 2	\$9	\$12	\$13	\$7	50
Plant 3	\$14	\$9	\$16	\$5	40
Demand (million kwh)	45	20	30	30	

To formulate Powerco's problem as an LP, we begin by defining a variable for each decision that Powerco must make. Because Powerco must determine how much power is sent from each plant to each city, we define (for $i = 1, 2, 3$ and $j = 1, 2, 3, 4$)

$$x_{ij} = \text{number of (million) kwh produced at plant } i \text{ and sent to city } j$$

In terms of these variables, the total cost of supplying the peak power demands to cities 1–4 may be written as

$$\begin{aligned} & 8x_{11} + 6x_{12} + 10x_{13} + 9x_{14} && \text{(Cost of shipping power from plant 1)} \\ + \; & 9x_{21} + 12x_{22} + 13x_{23} + 7x_{24} && \text{(Cost of shipping power from plant 2)} \\ + \; & 14x_{31} + 9x_{32} + 16x_{33} + 5x_{34} && \text{(Cost of shipping power from plant 3)} \end{aligned}$$

Powerco faces two types of constraints. First, the total power supplied by each plant cannot exceed the plant's capacity. For example, the total amount of power sent from plant

1 to the four cities cannot exceed 35 million kwh. Each variable with first subscript 1 represents a shipment of power from plant 1, so we may express this restriction by the LP constraint

$$x_{11} + x_{12} + x_{13} + x_{14} \leq 35$$

In a similar fashion, we can find constraints that reflect plant 2's and plant 3's capacities. Because power is supplied by the power plants, each is a **supply point.** Analogously, a constraint that ensures that the total quantity shipped from a plant does not exceed plant capacity is a **supply constraint.** The LP formulation of Powerco's problem contains the following three supply constraints:

$$\begin{aligned} &x_{11} + x_{12} + x_{13} + x_{14} \leq 35 \qquad &&\text{(Plant 1 supply constraint)} \\ &x_{21} + x_{22} + x_{23} + x_{24} \leq 50 \qquad &&\text{(Plant 2 supply constraint)} \\ &x_{31} + x_{32} + x_{33} + x_{34} \leq 40 \qquad &&\text{(Plant 3 supply constraint)} \end{aligned}$$

Second, we need constraints that ensure that each city will receive sufficient power to meet its peak demand. Each city demands power, so each is a **demand point.** For example, city 1 must receive at least 45 million kwh. Each variable with second subscript 1 represents a shipment of power to city 1, so we obtain the following constraint:

$$x_{11} + x_{21} + x_{31} \geq 45$$

Similarly, we obtain a constraint for each of cities 2, 3, and 4. A constraint that ensures that a location receives its demand is a **demand constraint.** Powerco must satisfy the following four demand constraints:

$$\begin{aligned} &x_{11} + x_{21} + x_{31} \geq 45 \qquad &&\text{(City 1 demand constraint)} \\ &x_{12} + x_{22} + x_{32} \geq 20 \qquad &&\text{(City 2 demand constraint)} \\ &x_{13} + x_{23} + x_{33} \geq 30 \qquad &&\text{(City 3 demand constraint)} \\ &x_{14} + x_{24} + x_{34} \geq 30 \qquad &&\text{(City 4 demand constraint)} \end{aligned}$$

Because all the x_{ij}'s must be nonnegative, we add the sign restrictions $x_{ij} \geq 0$ ($i = 1, 2, 3; j = 1, 2, 3, 4$).

Combining the objective function, supply constraints, demand constraints, and sign restrictions yields the following LP formulation of Powerco's problem:

$$\begin{aligned} \min z = {} & 8x_{11} + 6x_{12} + 10x_{13} + 9x_{14} + 9x_{21} + 12x_{22} + 13x_{23} + 7x_{24} \\ & + 14x_{31} + 9x_{32} + 16x_{33} + 5x_{34} \\ \text{s.t.}\quad & x_{11} + x_{12} + x_{13} + x_{14} \leq 35 \qquad \text{(Supply constraints)} \\ & x_{21} + x_{22} + x_{23} + x_{24} \leq 50 \\ & x_{31} + x_{32} + x_{33} + x_{34} \leq 40 \end{aligned}$$

$$x_{11} + x_{21} + x_{31} \geq 45 \qquad \text{(Demand constraints)}$$
$$x_{12} + x_{22} + x_{32} \geq 20$$
$$x_{13} + x_{23} + x_{33} \geq 30$$
$$x_{14} + x_{24} + x_{34} \geq 30$$
$$x_{ij} \geq 0 \quad (i = 1, 2, 3; j = 1, 2, 3, 4)$$

TABLE 3
Transportation Tableau for Powerco

	City 1	City 2	City 3	City 4	Supply
Plant 1	8	6 10	10 25	9	35
Plant 2	9 45	12	13 5	7	50
Plant 3	14	9 10	16	5 30	40
Demand	45	20	30	30	

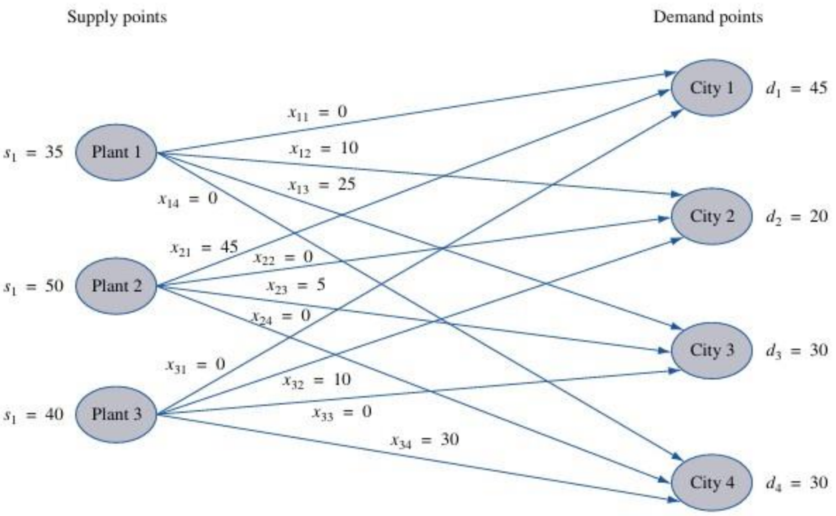
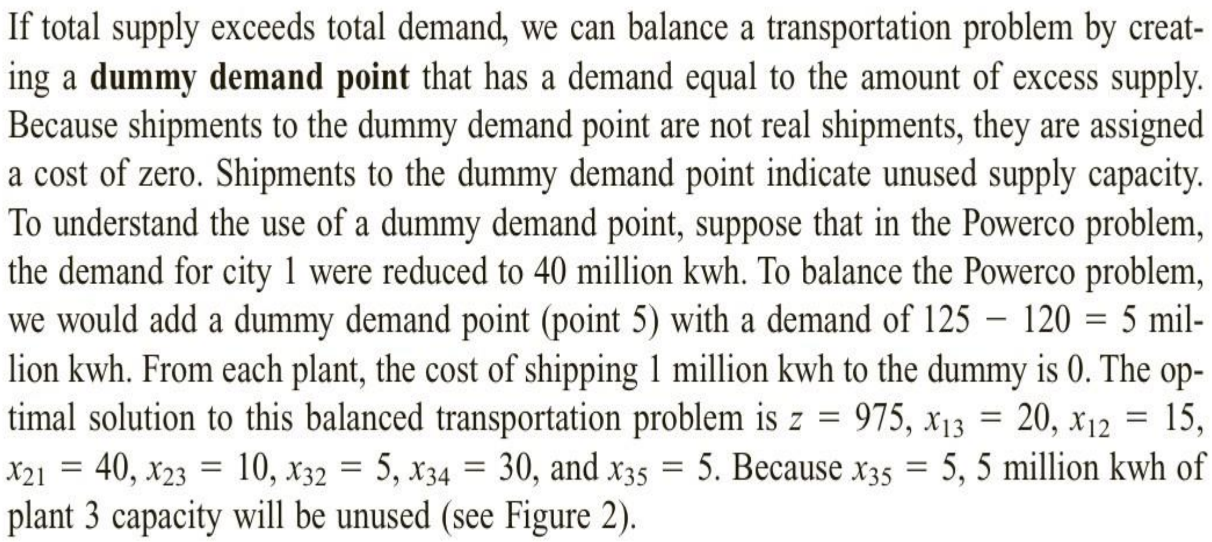


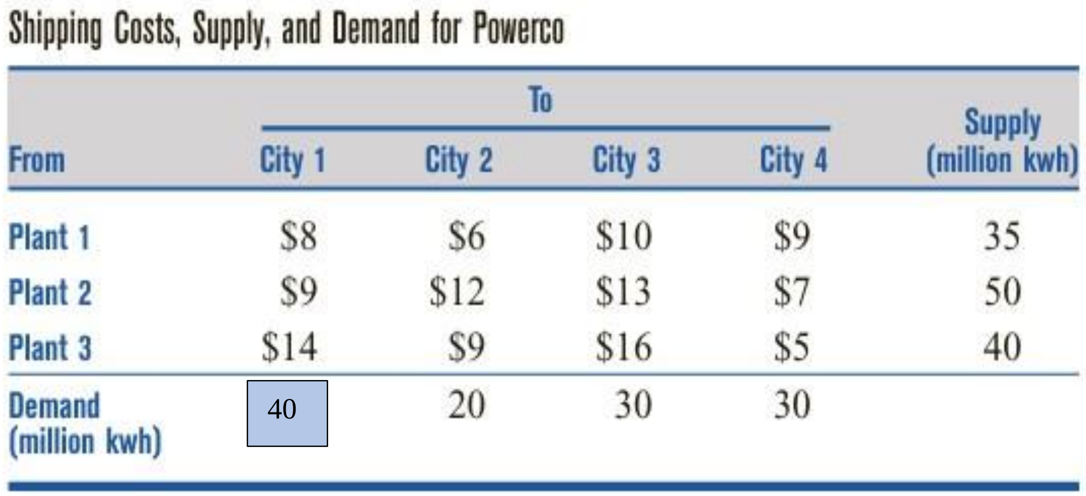
FIGURE 1
Graphical Representation of Powerco Problem and Its Optimal Solution

optimal solution to this LP is $z = 1020$, $x_{12} = 10$, $x_{13} = 25$, $x_{21} = 45$, $x_{23} = 5$, $x_{32} = 10$, $x_{34} = 30$. Figure 1 is a graphical representation of the Powerco problem and its optimal solution. The variable x_{ij} is represented by a line, or arc, joining the ith supply point (plant i) and the jth demand point (city j).

4. Balancing a Transportation Problem if total supply exceeds total demand



If total supply exceeds total demand, we can balance a transportation problem by creating a **dummy demand point** that has a demand equal to the amount of excess supply. Because shipments to the dummy demand point are not real shipments, they are assigned a cost of zero. Shipments to the dummy demand point indicate unused supply capacity. To understand the use of a dummy demand point, suppose that in the Powerco problem, the demand for city 1 were reduced to 40 million kwh. To balance the Powerco problem, we would add a dummy demand point (point 5) with a demand of $125 - 120 = 5$ million kwh. From each plant, the cost of shipping 1 million kwh to the dummy is 0. The optimal solution to this balanced transportation problem is $z = 975$, $x_{13} = 20$, $x_{12} = 15$, $x_{21} = 40$, $x_{23} = 10$, $x_{32} = 5$, $x_{34} = 30$, and $x_{35} = 5$. Because $x_{35} = 5$, 5 million kwh of plant 3 capacity will be unused (see Figure 2).



Shipping Costs, Supply, and Demand for Powerco

	To				
From	**City 1**	**City 2**	**City 3**	**City 4**	**Supply (million kwh)**
Plant 1	\$8	\$6	\$10	\$9	35
Plant 2	\$9	\$12	\$13	\$7	50
Plant 3	\$14	\$9	\$16	\$5	40
Demand (million kwh)	40	20	30	30	

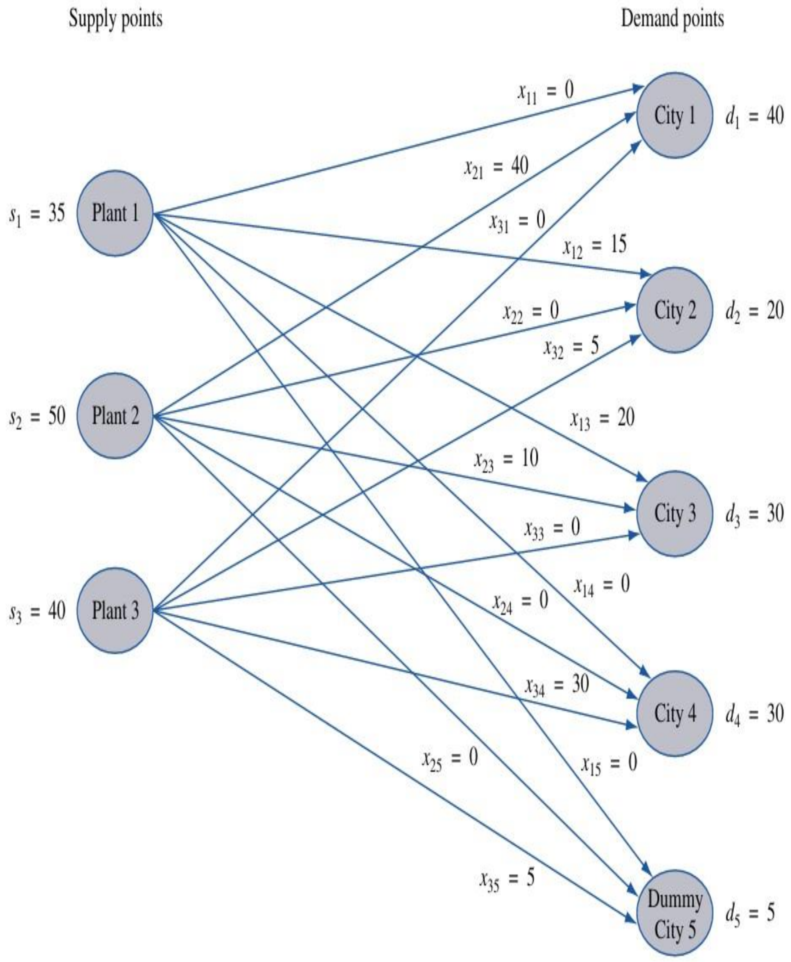


FIGURE 2
Graphical Representation of Unbalanced Powerco Problem and Its Optimal Solution (with Dummy Demand Point)

5. Balancing a Transportation Problem if total supply is less than total demand

Problem: Two reservoirs are available to supply the water needs of three cities. Each reservoir can supply up to 50 million gallons of water per day. Each city would like to receive 40 million gallons per day. For each million gallons per day of unmet demand, there is a penalty. At city 1, the penalty is \$20; at city 2, the penalty is \$22; and at city 3, the penalty is \$23. The cost of transporting 1 million gallons of water from each reservoir to each city is shown in Table 4. Formulate a balanced transportation problem that can be used to minimize the sum of shortage and transport costs.

Solution In this problem,

$$\text{Daily supply} = 50 + 50 = 100 \text{ million gallons per day}$$
$$\text{Daily demand} = 40 + 40 + 40 = 120 \text{ million gallons per day}$$

To balance the problem, we add a dummy (or shortage) *supply point* having a supply of 120 − 100 = 20 million gallons per day. The cost of shipping 1 million gallons from the dummy supply point to a city is just the shortage cost per million gallons for that city. Table 5 shows the balanced transportation problem and its optimal solution. Reservoir 1 should send 20 million gallons per day to city 1 and 30 million gallons per day to city 2, whereas reservoir 2 should send 10 million gallons per day to city 2 and 40 million gallons per day to city 3. Twenty million gallons per day of city 1's demand will be unsatisfied.

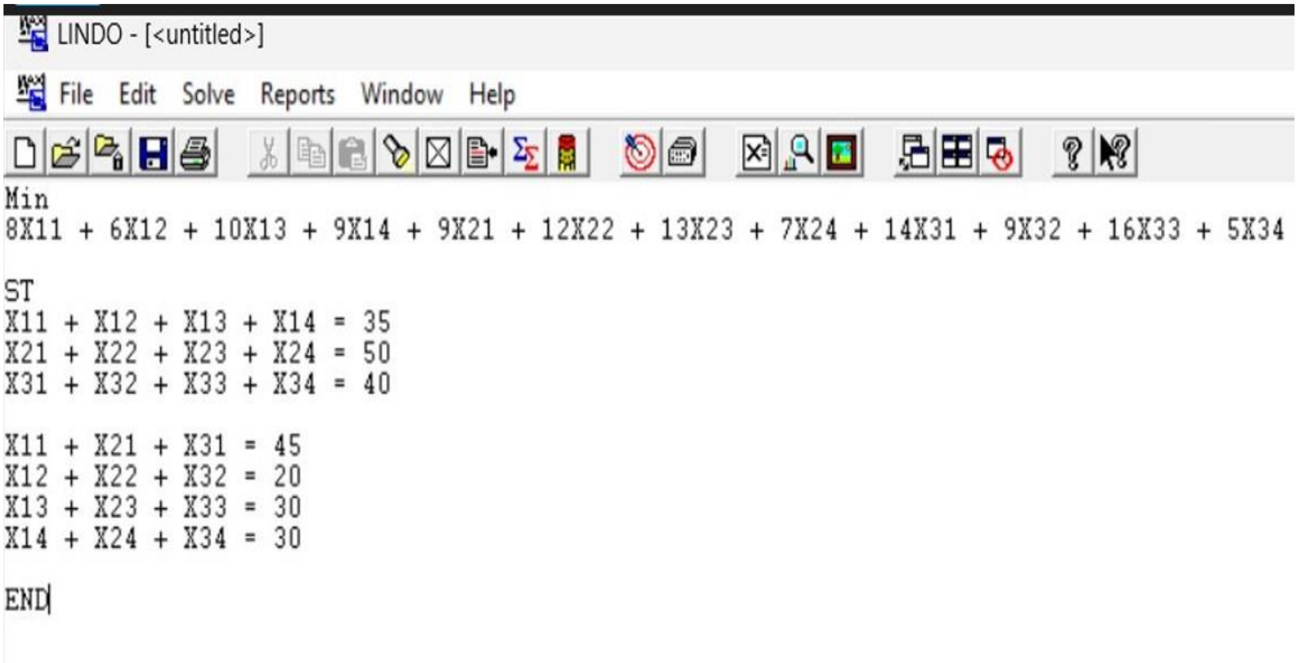
TABLE 4
Shipping Costs for Reservoir

	To		
From	City 1	City 2	City 3
Reservoir 1	$7	$8	$10
Reservoir 2	$9	$7	$8

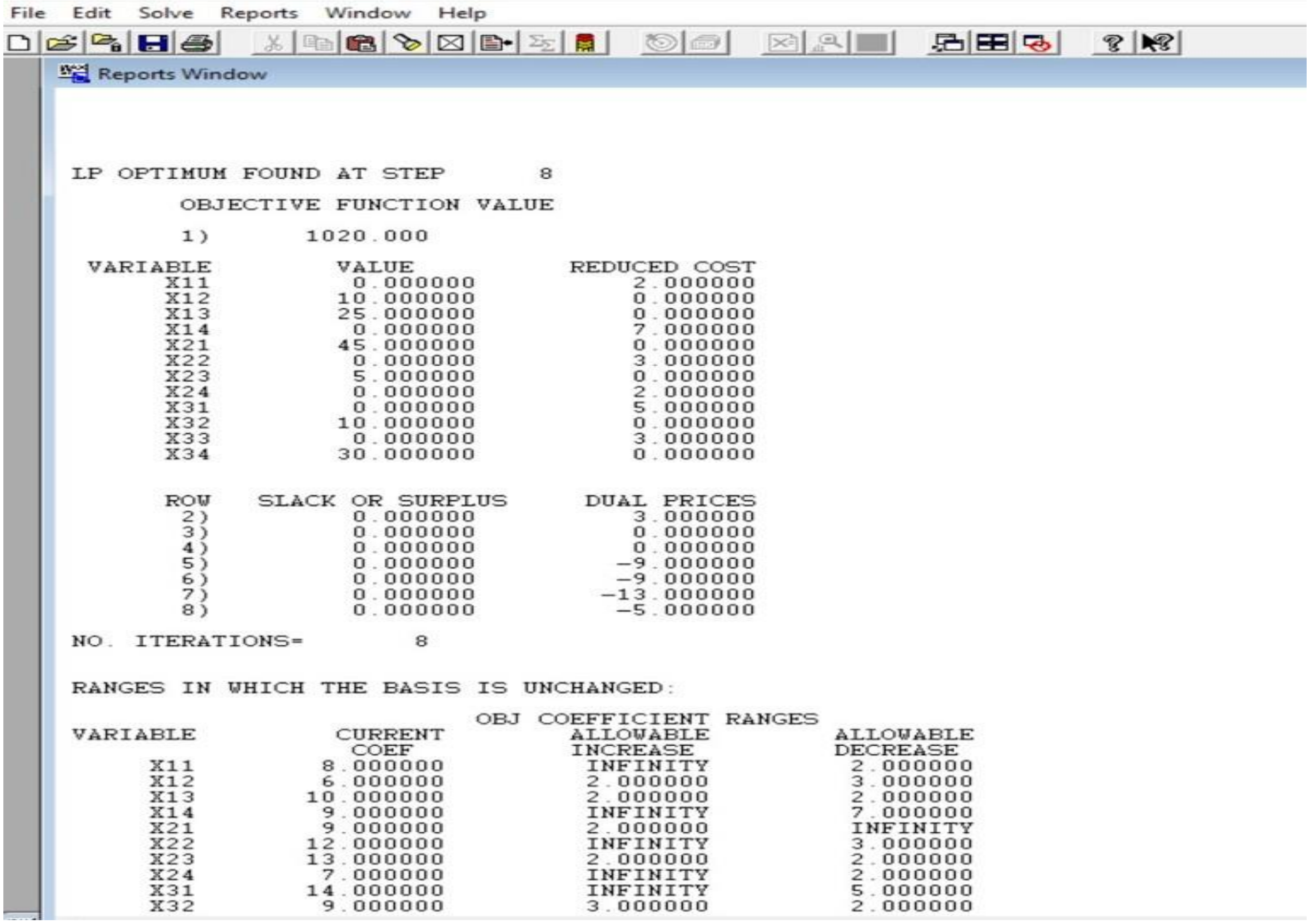
TABLE 5
Transportation Tableau for Reservoir

	City 1	City 2	City 3	Supply
Reservoir 1	7 20	8 30	10	50
Reservoir 2	9	7 10	8 40	50
Dummy (shortage)	20 20	22	23	20
Demand	40	40	40	

6. Balanced transportation problems (PowerCo) using LINDO:



```
Min
8X11 + 6X12 + 10X13 + 9X14 + 9X21 + 12X22 + 13X23 + 7X24 + 14X31 + 9X32 + 16X33 + 5X34

ST
X11 + X12 + X13 + X14 = 35
X21 + X22 + X23 + X24 = 50
X31 + X32 + X33 + X34 = 40

X11 + X21 + X31 = 45
X12 + X22 + X32 = 20
X13 + X23 + X33 = 30
X14 + X24 + X34 = 30

END
```



```
LP OPTIMUM FOUND AT STEP       8

       OBJECTIVE FUNCTION VALUE

       1)       1020.000

 VARIABLE        VALUE          REDUCED COST
      X11         0.000000          2.000000
      X12        10.000000          0.000000
      X13        25.000000          0.000000
      X14         0.000000          7.000000
      X21        45.000000          0.000000
      X22         0.000000          3.000000
      X23         5.000000          0.000000
      X24         0.000000          2.000000
      X31         0.000000          5.000000
      X32        10.000000          0.000000
      X33         0.000000          3.000000
      X34        30.000000          0.000000

      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          3.000000
       3)         0.000000          0.000000
       4)         0.000000          0.000000
       5)         0.000000         -9.000000
       6)         0.000000         -9.000000
       7)         0.000000        -13.000000
       8)         0.000000         -5.000000

NO. ITERATIONS=       8

RANGES IN WHICH THE BASIS IS UNCHANGED:

                          OBJ COEFFICIENT RANGES
VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                   COEF          INCREASE         DECREASE
     X11         8.000000        INFINITY         2.000000
     X12         6.000000        2.000000         3.000000
     X13        10.000000        2.000000         2.000000
     X14         9.000000        INFINITY         7.000000
     X21         9.000000        2.000000         INFINITY
     X22        12.000000        INFINITY         3.000000
     X23        13.000000        2.000000         2.000000
     X24         7.000000        INFINITY         2.000000
     X31        14.000000        INFINITY         5.000000
     X32         9.000000        3.000000         2.000000
```

7. Methods to obtain the initial basic feasible solution:

Tabular Representation

Let 'm' denote number of factories (f1,f2…fm)

Let 'n' denote number of warehouse (w1,w2….wn)

w f	W1	W2	..	Wn	Capacities (availability)
F1	C11	C12	..	C1n	A1
F2	C21	C22	..	C2n	A2
.	.	.	.	.	..
.	.	.	.	.	..
Fm	Cm1	Cm2	.	Cmn	Am
Required	B1	B2	..	Bn	$\sum ai=\sum bj$

w f	W1	W2	..	Wn	Capacities (availability)
F1	X11	X12	..	W1n	A1
F2	X21	X22	..	X2n	A2
.	.	.	.	.	..
.	.	.	.	.	..
Fm	Xm1	Xm2	.	Xmn	Am
Required	B1	B2	..	Bn	$\sum ai=\sum bj$

In general these two tables are combined by inserting each unit cost c_{ij} with the corresponding amount x_{ij} in the cell (I,j). the product c_{ij} x_{ij} gives the net cost of shipping units from the factory fi to warehouse w_j.

7.1 North –west corner rule:

Step-1:

- The first assignment is made in the cell occupying the upper left- hand (north-west) corner of the table.
- The maximum possible amount is allocated here i.e. $x11=min (a1,b1)$. This value of $x11$ is then entered in the cell (1,1) of the transportation table.

Step-2:

- If $b1 > a1$, move vertically downwards to the second row and make the second allocation of amount $x21=min(a2,b1-x11)$ in the cell (2,1).
- If $b1 < a1$, move horizontally right side to the second column and make the second allocation of amount $x12=min(a1- x11,b2)$ in the cell(1,2).
- If $b1=a1$, there is tie for the second allocation. One can make a second allocation of magnitude $x12=min(a1-a1,b2)$ in the cell(1,2) or $x21=min(a2,b1-b1)$in the cell(2,1).

Step-3:

- Start from the new north-west corner of the transportation table and repeat steps 1 and step 2 until all the requirements are satisfied.

Example: Find the initial basic feasible solution by using **north-west corner** rule.

w f	W1	W2	W3	W4	Factory capacity
F1	19	30	50	10	7
F2	70	30	40	60	9
F3	40	8	70	20	18
Warehouse requirement	5	8	7	14	34

	W1	W2	W3	W4	Availability

F1	5 (19)	2 (30)			7	2	0
F2		6 (30)	3 (40)		9	3	0
F3			4 (70)	14 (20)	18	14	0
Requirements	5	8	7	14			
	0	6	4	0			
		0	0				

Solution:

Initial basic feasible solution

X11=5, X12=2, X22-6, X23=3, X33=4, X34=14

The transportation cost is

5(19)+2(30)+6(30)+3(40)+4(70)+14(20)=$1015

7.2 Vogel's Approximation Method (VAM):

Step-1:

- For each row of the table, identify the smallest and the next to smallest cost. Determine the different between them for each row. These are called penalties. Put them aside by enclosing them in the parenthesis against the respective rows. Similarly compute penalties for each column.

Step-2:

- Identify the row or column with the largest penalty. If a tie occurs then use an arbitrary choice. Let the largest penalty corresponding to the i^{th} row have the cost cij allocate the largest possible amount xij=min(ai,bj) in the cell(I,j) and cross out either i^{th} row or j^{th} column in the usual manner.

Step-3:

- Again compute the row and column penalties for the reduced table and then go to step 2. Repeat the procedure until all the requirements are satisfied.

Example: Find the initial basic feasible solution by using **Vogel's Approximation Method (VAM)**.

	W1	W2	W3	W4	Availability
F1	19	30	50	10	7
F2	70	30	40	60	9
F3	40	8	70	20	18
Requirements	5	8	7	14	

Solution:

	W1	W2	W3	W4	Availability	
F1	19	30	50	10	7	19-10=9
F2	70	30	40	60	9	40-30=10
F3	40	8	70	20	18	20-8=12
	5	8	7	14		
	40-19=21	30-8=22	50-40=10	20-10=10		

	W1	W2	W3	W4	Availability	
F1	19	30	50	10	7	9
F2	70	30	40	60	9	10
F3	40	8 8	70	20	10	12
	5	0	7	14		
	21	x	10	10		

	W1	W2	W3	W4	Availability	
F1	19 5	30	50	10	2	9
F2	70	30	40	60	9	20
F3	40	8 8	70	20	10	20
	0	0	7	14		
	X	X	10	10		

	W1	W2	W3	W4	Availability	
F1	19 5	30	50	10	2	40
F2	70	30	40	60	9	20
F3	40	8 8	70	20 10	0	x
	0	0	7	4		
	X	X	10	50		

	W1	W2	W3	W4	Availability	
F1	19 5	30	50	10 2	0	X
F2	70	30	40	60	9	20
F3	40	8 8	70	20 10	0	X
	0	0	7	2		
	X	X	10	50		

	W1	W2	W3	W4	Availability	
F1	19 5	30	50	10 2	0	X
F2	70	30	40 7	60 2	0	X
F3	40	8 8	70	20 10	0	X
	0	0	0	0		
	X	X	X	X		

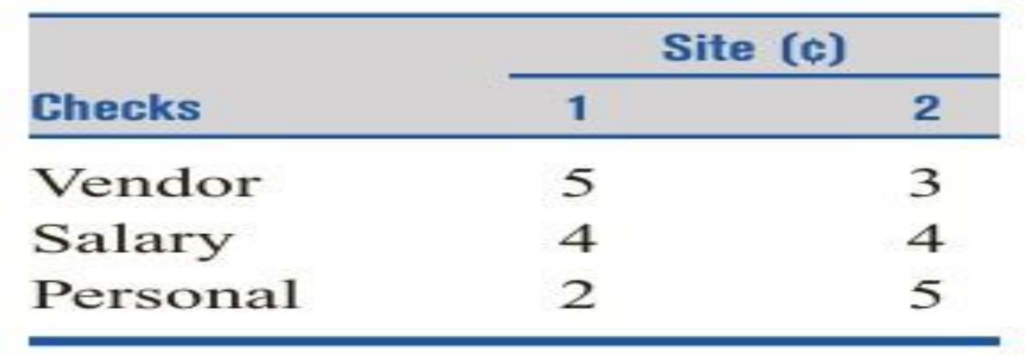
Initial basic feasible solution X11=5,X14=2,X23=7,X24=2,X32=8,X34=10.

The transportation cost is

5(19)+2(10)+7(40)+2(60)+8(8)+10(20)=$779.

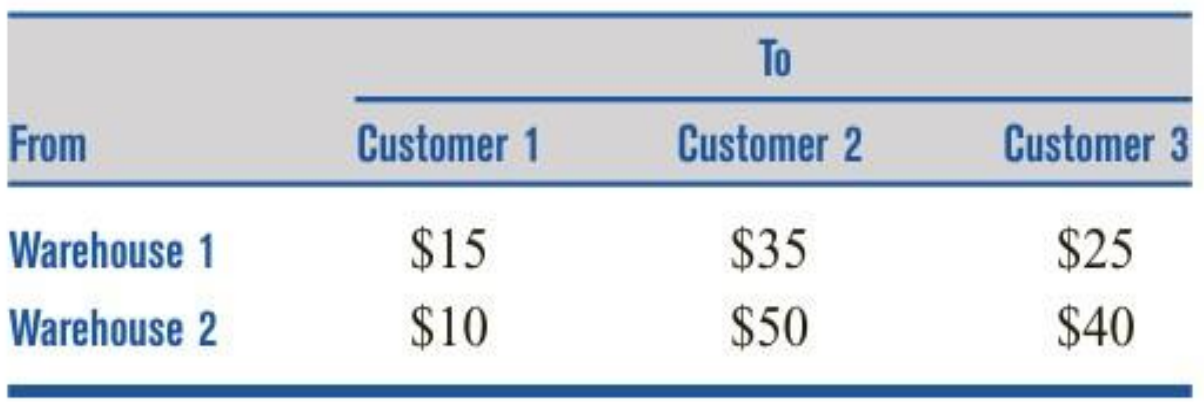
Practice Problem:

1.A bank has two sites at which checks are processed. Site 1 can process 10,000 checks per day, and site 2 can process 6,000 checks per day. The bank processes three types of checks: vendor, salary, and personal. The processing cost per check depends on the site. Each day, 5,000 checks of each type must be processed. Formulate a balanced transportation problem to minimize the daily cost of processing checks.



	Site (¢)	
Checks	**1**	**2**
Vendor	5	3
Salary	4	4
Personal	2	5

2. A company supplies goods to three customers, who each require 30 units. The company has two warehouses. Warehouse 1 has 40 units available, and warehouse 2 has 30 units available. The costs of shipping 1 unit from warehouse to customer. There is a penalty for each unmet customer unit of demand: With customer 1, a penalty cost of $90 is incurred; with customer 2, $80; and with customer 3, $110. Formulate a balanced transportation.



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

Experiment-6: Solving the Assignment Problem Using LINDO.

1. Objective

By the end of this experiment, students will be able to:

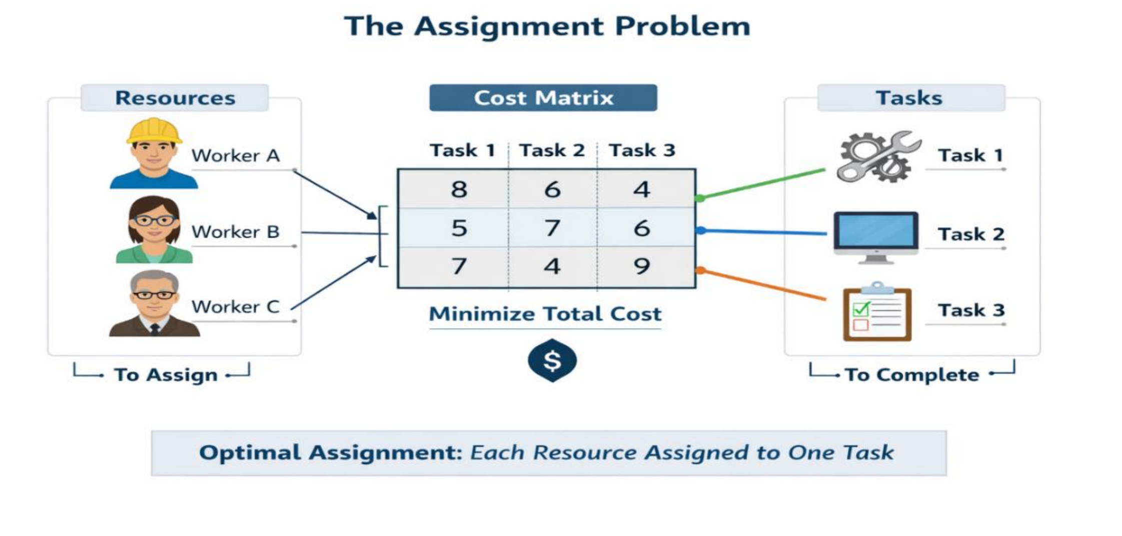
- Formulate assignment problems.
- Solve balanced and unbalanced assignment problems by The Hungarian Method.
- Solve balanced and unbalanced assignment problems using LINDO.

2. Introduction to Assignment Problem

The **Assignment Problem** is a special type of linear programming problem in Operations Research that deals with allocating a set of resources to a set of tasks in an optimal way. The main objective is to assign each resource to exactly one task such that the total cost is minimized or the total profit is maximized. Typically, the number of resources and tasks are equal, forming a square cost matrix. Each element of the matrix represents the cost (or time, or profit) of assigning a particular resource to a specific task. The solution must satisfy two main constraints:

1. Each resource is assigned to only one task.
2. Each task is performed by only one resource.

The Assignment Problem is actually a special case of the Transportation Problem, where supply and demand for each source and destination are equal to one unit. Because of this structure, it can be solved efficiently using specialized methods such as the Hungarian Method. This problem has wide practical applications in real life, such as assigning workers to jobs, machines to tasks, salespeople to territories, or teachers to courses. By finding the optimal assignment, organizations can minimize operational costs, improve efficiency, and ensure better utilization of resources.



Task 1	Task 2	Task 3
8	6	4
5	7	6
7	4	9

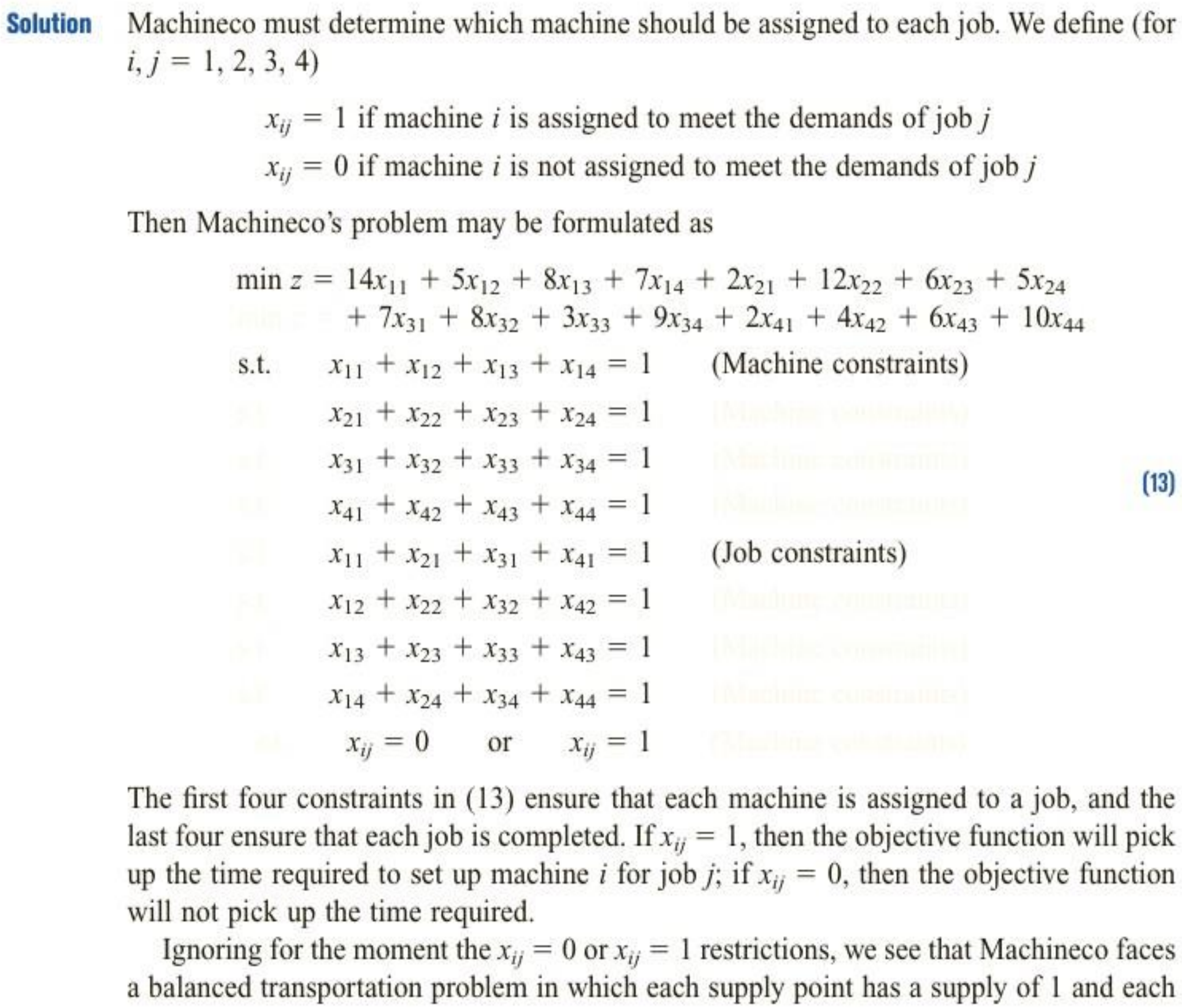
3. Balanced Assignment Problem Formulation

Example: Machineco has four machines and four jobs to be completed. Each machine must be assigned to complete one job. The time required to set up each machine for completing each job is shown in **Table 43**. Machineco wants to minimize the total setup time needed to complete the four jobs. Use linear programming to solve this problem.

TABLE 43

Setup Times for Machineco

Machine	Time (Hours)			
	Job 1	Job 2	Job 3	Job 4
1	14	5	8	7
2	2	12	6	5
3	7	8	3	9
4	2	4	6	10



Solution Machineco must determine which machine should be assigned to each job. We define (for $i, j = 1, 2, 3, 4$)

$$x_{ij} = 1 \text{ if machine } i \text{ is assigned to meet the demands of job } j$$

$$x_{ij} = 0 \text{ if machine } i \text{ is not assigned to meet the demands of job } j$$

Then Machineco's problem may be formulated as

$$\begin{aligned}
\min z = {} & 14x_{11} + 5x_{12} + 8x_{13} + 7x_{14} + 2x_{21} + 12x_{22} + 6x_{23} + 5x_{24} \\
& + 7x_{31} + 8x_{32} + 3x_{33} + 9x_{34} + 2x_{41} + 4x_{42} + 6x_{43} + 10x_{44} \\
\text{s.t.} \quad & x_{11} + x_{12} + x_{13} + x_{14} = 1 \qquad \text{(Machine constraints)} \\
& x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 1 \\
& x_{41} + x_{42} + x_{43} + x_{44} = 1 \\
& x_{11} + x_{21} + x_{31} + x_{41} = 1 \qquad \text{(Job constraints)} \\
& x_{12} + x_{22} + x_{32} + x_{42} = 1 \\
& x_{13} + x_{23} + x_{33} + x_{43} = 1 \\
& x_{14} + x_{24} + x_{34} + x_{44} = 1 \\
& x_{ij} = 0 \quad \text{or} \quad x_{ij} = 1
\end{aligned} \tag{13}$$

The first four constraints in (13) ensure that each machine is assigned to a job, and the last four ensure that each job is completed. If $x_{ij} = 1$, then the objective function will pick up the time required to set up machine i for job j; if $x_{ij} = 0$, then the objective function will not pick up the time required.

Ignoring for the moment the $x_{ij} = 0$ or $x_{ij} = 1$ restrictions, we see that Machineco faces a balanced transportation problem in which each supply point has a supply of 1 and each demand point has a demand of 1. In general, an **assignment problem** is a balanced transportation problem in which all supplies and demands are equal to 1. Thus, an assignment problem is characterized by knowledge of the cost of assigning each supply point to each demand point. The assignment problem's matrix of costs is its **cost matrix.**

All the supplies and demands for the Machineco problem (and for any assignment problem) are integers, so our discussion in Section 7.3 implies that all variables in Machineco's optimal solution must be integers. Because the right-hand side of each constraint is equal to 1, each x_{ij} must be a nonnegative integer that is no larger than 1, so each x_{ij} must equal 0 or 1. This means that we can ignore the restrictions that $x_{ij} = 0$ or 1 and solve (13) as a balanced transportation problem. By the minimum cost method, we obtain the bfs in Table 44. The current bfs is highly degenerate. (In any bfs to an $m \times m$ assignment problem, there will always be m basic variables that equal 1 and $m - 1$ basic variables that equal 0.)

We find that $\bar{c}_{43} = 1$ is the only positive $\bar{c}_{ij}$. We therefore enter x_{43} into the basis. The loop involving x_{43} and some of the basic variables is (4, 3)–(1, 3)–(1, 2)–(4, 2). The odd variables in the loop are x_{13} and x_{42}. Because $x_{13} = x_{42} = 0$, either x_{13} or x_{42} will leave the basis. We arbitrarily choose x_{13} to leave the basis. After performing the pivot, we obtain the bfs in Table 45. All $\bar{c}_{ij}$'s are now nonpositive, so we have obtained an optimal assignment: $x_{12} = 1$, $x_{24} = 1$, $x_{33} = 1$, and $x_{41} = 1$. Thus, machine 1 is assigned to job 2, machine 2 is assigned to job 4, machine 3 is assigned to job 3, and machine 4 is assigned to job 1. A total setup time of $5 + 5 + 3 + 2 = 15$ hours is required.

TABLE 44
Basic Feasible Solution for Machineco

		Job 1	Job 2	Job 3	Job 4	
	$v_j =$	3	4	8	7	
Machine 1	$u_i = 0$	14	5 1	8 0	7 0	1
Machine 2	−2	2	12	6	5 1	1
Machine 3	−5	7	8	3 1	9	1
Machine 4	−1	2 1	4 0	6	10	1
		1	1	1	1	

TABLE 45
x_{43} Has Entered the Basis

		Job 1	Job 2	Job 3	Job 4	
	$v_j =$	3	5	7	7	
Machine 1	$u_i = 0$	14	5 1	8	7 0	1
Machine 2	−2	2	12	6	5 1	1
Machine 3	−4	7	8	3 1	9	1
Machine 4	−1	2 1	4 0	6 0	10	1
		1	1	1	1	

4. Balanced Assignment Problem Solution by The Hungarian Method

Looking back at our initial bfs, we see that it was an optimal solution. We did not know that it was optimal, however, until performing one iteration of the transportation simplex. This suggests that the high degree of degeneracy in an assignment problem may cause the transportation simplex to be an inefficient way of solving assignment problems. For this reason (and the fact that the algorithm is even simpler than the transportation simplex), the Hungarian method is usually used to solve assignment (min) problems:

Step 1 Find the minimum element in each row of the $m \times m$ cost matrix. Construct a new matrix by subtracting from each cost the minimum cost in its row. For this new matrix, find the minimum cost in each column. Construct a new matrix (called the reduced cost matrix) by subtracting from each cost the minimum cost in its column.

Step 2 Draw the minimum number of lines (horizontal, vertical, or both) that are needed to cover all the zeros in the reduced cost matrix. If m lines are required, then an optimal solution is available among the covered zeros in the matrix. If fewer than m lines are needed, then proceed to step 3.

Step 3 Find the smallest nonzero element (call its value k) in the reduced cost matrix that is uncovered by the lines drawn in step 2. Now subtract k from each uncovered element of the reduced cost matrix and add k to each element that is covered by two lines. Return to step 2.

1 To solve an assignment problem in which the goal is to maximize the objective function, multiply the profits matrix through by -1 and solve the problem as a minimization problem.

2 If the number of rows and columns in the cost matrix are unequal, then the assignment problem is unbalanced. The Hungarian method may yield an incorrect solution if the problem is unbalanced. Thus, any assignment problem should be balanced (by the addition of one or more dummy points) before it is solved by the Hungarian method.

3 In a large problem, it may not be easy to find the minimum number of lines needed to cover all zeros in the current cost matrix. For a discussion of how to find the minimum number of lines needed, see Gillett (1976). It can be shown that if j lines are required, then only j "jobs" can be assigned to zero costs in the current matrix. This explains why the algorithm terminates when m lines are required.

Solution of Machineco Example by the Hungarian Method

We illustrate the Hungarian method by solving the Machineco problem (see Table 46).

Step 1 For each row, we subtract the row minimum from each element in the row, obtaining Table 47. We now subtract 2 from each cost in column 4, obtaining Table 48.

Step 2 As shown, lines through row 1, row 3, and column 1 cover all the zeros in the reduced cost matrix. From remark 3, it follows that only three jobs can be assigned to zero costs in the current cost matrix. Fewer than four lines are required to cover all the zeros, so we proceed to step 3.

TABLE 46
Cost Matrix for Machineco

				Row Minimum
14	5	8	7	5
2	12	6	5	2
7	8	3	9	3
2	4	6	10	2

TABLE 47
Cost Matrix After Row Minimums Are Subtracted

9	0	3	2
0	10	4	3
4	5	0	6
0	2	4	8
Column Minimum	0	0	2

TABLE 48
Cost Matrix After Column Minimums Are Subtracted

9	0	3	0
0	10	4	1
4	5	0	4
0	2	4	6

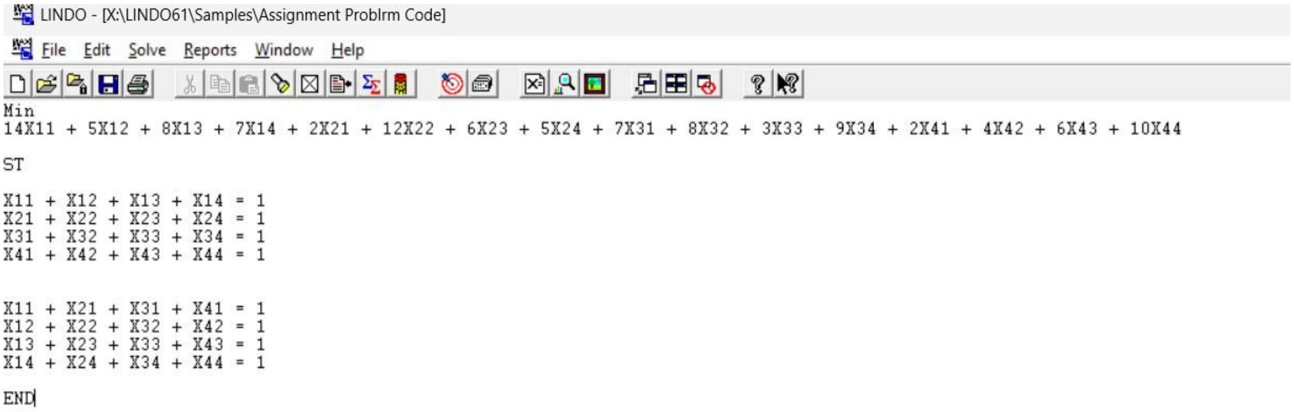
Step 3 The smallest uncovered element equals 1, so we now subtract 1 from each uncovered element in the reduced cost matrix and add 1 to each twice-covered element. The resulting matrix is Table 49. Four lines are now required to cover all the zeros. Thus, an optimal solution is available. To find an optimal assignment, observe that the only covered 0 in column 3 is x_{33}, so we must have $x_{33} = 1$. Also, the only available covered zero in column 2 is x_{12}, so we set $x_{12} = 1$ and observe that neither row 1 nor column 2 can be used again. Now the only available covered zero in column 4 is x_{24}. Thus, we choose $x_{24} = 1$ (which now excludes both row 2 and column 4 from further use). Finally, we choose $x_{41} = 1$.

TABLE 49
Four Lines Required; Optimal Solution Is Available

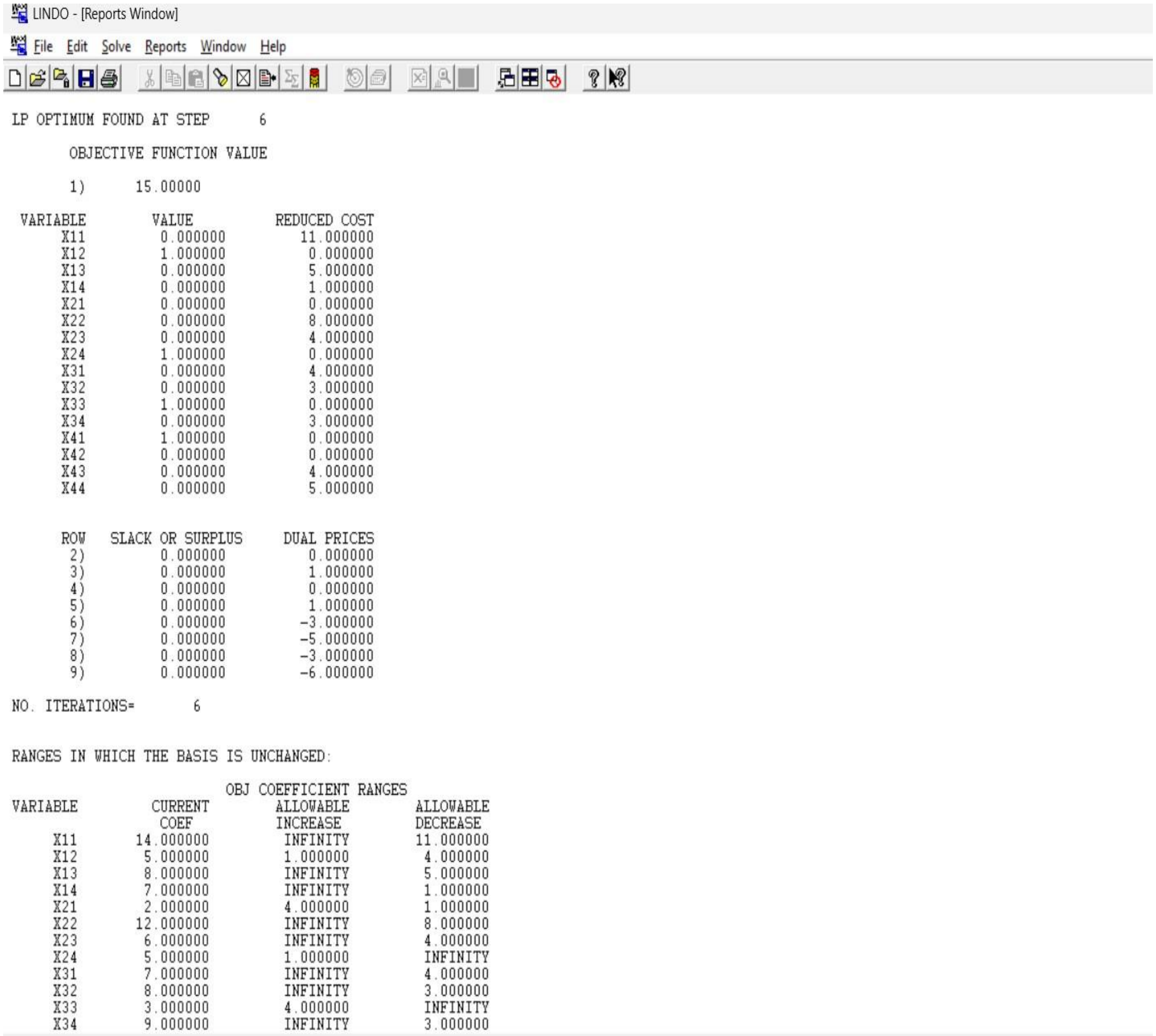
10	0	3	0
0	9	3	0
5	5	0	4
0	1	3	5

Thus, we have found the optimal assignment $x_{12} = 1$, $x_{24} = 1$, $x_{33} = 1$, and $x_{41} = 1$. Of course, this agrees with the result obtained by the transportation simplex.

5. Balanced Assignment problems (Machineco) using LINDO:



```
Min
14X11 + 5X12 + 8X13 + 7X14 + 2X21 + 12X22 + 6X23 + 5X24 + 7X31 + 8X32 + 3X33 + 9X34 + 2X41 + 4X42 + 6X43 + 10X44

ST

X11 + X12 + X13 + X14 = 1
X21 + X22 + X23 + X24 = 1
X31 + X32 + X33 + X34 = 1
X41 + X42 + X43 + X44 = 1


X11 + X21 + X31 + X41 = 1
X12 + X22 + X32 + X42 = 1
X13 + X23 + X33 + X43 = 1
X14 + X24 + X34 + X44 = 1

END
```



```
 LP OPTIMUM FOUND AT STEP      6

        OBJECTIVE FUNCTION VALUE

        1)      15.00000

  VARIABLE        VALUE          REDUCED COST
        X11         0.000000         11.000000
        X12         1.000000          0.000000
        X13         0.000000          5.000000
        X14         0.000000          1.000000
        X21         0.000000          0.000000
        X22         0.000000          8.000000
        X23         0.000000          4.000000
        X24         1.000000          0.000000
        X31         0.000000          4.000000
        X32         0.000000          3.000000
        X33         1.000000          0.000000
        X34         0.000000          3.000000
        X41         1.000000          0.000000
        X42         0.000000          0.000000
        X43         0.000000          4.000000
        X44         0.000000          5.000000


        ROW   SLACK OR SURPLUS     DUAL PRICES
         2)         0.000000          0.000000
         3)         0.000000          1.000000
         4)         0.000000          0.000000
         5)         0.000000          1.000000
         6)         0.000000         -3.000000
         7)         0.000000         -5.000000
         8)         0.000000         -3.000000
         9)         0.000000         -6.000000

 NO. ITERATIONS=       6


 RANGES IN WHICH THE BASIS IS UNCHANGED:

                           OBJ COEFFICIENT RANGES
 VARIABLE         CURRENT        ALLOWABLE        ALLOWABLE
                   COEF          INCREASE         DECREASE
      X11        14.000000        INFINITY        11.000000
      X12         5.000000        1.000000         4.000000
      X13         8.000000        INFINITY         5.000000
      X14         7.000000        INFINITY         1.000000
      X21         2.000000        4.000000         1.000000
      X22        12.000000        INFINITY         8.000000
      X23         6.000000        INFINITY         4.000000
      X24         5.000000        1.000000         INFINITY
      X31         7.000000        INFINITY         4.000000
      X32         8.000000        INFINITY         3.000000
      X33         3.000000        4.000000         INFINITY
      X34         9.000000        INFINITY         3.000000
```

6. Unbalanced Assignment Problem Solution by The Hungarian Method

Example: A company has five machines that are used for four jobs. Each job can be assigned to one and only one machine. The cost of each job on each machine is given on the following table.

$$\begin{array}{cc|ccccc} & & \multicolumn{5}{c}{\textbf{Machines}} \\ & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\ \hline & \textbf{1} & 5 & 7 & 11 & 6 & 7 \\ & \textbf{2} & 8 & 5 & 5 & 6 & 5 \\ \textbf{Job} & \textbf{3} & 6 & 7 & 10 & 7 & 3 \\ & \textbf{4} & 10 & 4 & 8 & 4 & 2 \end{array}$$

Solution:

Convert the 4*5 matrix into a square matrix by adding a dummy row D5.

Dummy Row D5 added

$$\begin{array}{cc|ccccc} & & \multicolumn{5}{c}{\textbf{Machines}} \\ & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\ \hline & \textbf{1} & 5 & 7 & 11 & 6 & 7 \\ & \textbf{2} & 8 & 5 & 5 & 6 & 5 \\ \textbf{Job} & \textbf{3} & 6 & 7 & 10 & 7 & 3 \\ & \textbf{4} & 10 & 4 & 8 & 4 & 2 \\ & \textbf{D}_5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Row-wise Reduction of the Matrix

$$\begin{array}{cc|ccccc|}
 & & \multicolumn{5}{c}{\textbf{Machines}} \\
 & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & 0 & 2 & 6 & 1 & 2 \\
 & \textbf{2} & 3 & 0 & 0 & 1 & 0 \\
\textbf{Job} & \textbf{3} & 3 & 4 & 7 & 4 & 0 \\
 & \textbf{4} & 8 & 2 & 6 & 2 & 0 \\
 & \textbf{D}_5 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Column-wise reduction is not necessary since all columns contain a single zero. Now, draw minimum number of lines to cover all the zeros, as shown in Table.

All Zeros in the Matrix Covered

$$\begin{array}{cc|ccccc|}
 & & \multicolumn{5}{c}{\textbf{Machines}} \\
 & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & 0 & 2 & 6 & 1 & 2 \\
 & \textbf{2} & 3 & 0 & 0 & 1 & 0 \\
\textbf{Job} & \textbf{3} & 3 & 4 & 7 & 4 & 0 \\
 & \textbf{4} & 8 & 2 & 6 & 2 & 0 \\
 & \textbf{D}_5 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Number of lines drawn ≠ Order of matrix. Hence not optimal.

Subtracted or Added to Elements

$$\begin{array}{cc|ccccc|}
 & & \multicolumn{5}{c}{\textbf{Machines}} \\
 & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & 0 & 1 & 5 & 0 & 2 \\
 & \textbf{2} & 4 & 0 & 0 & 1 & 1 \\
\textbf{Job} & \textbf{3} & 3 & 3 & 6 & 3 & 0 \\
 & \textbf{4} & 8 & 1 & 5 & 1 & 0 \\
 & \textbf{D}_5 & 1 & 0 & 0 & 0 & 1
\end{array}$$

Number of lines drawn ≠ Order of matrix. Hence not optimal.

Again Added or Subtracted 1 from Elements

$$\begin{array}{ccc|ccccc|}
 & & & \multicolumn{5}{c}{\textbf{Machines}} \\
 & & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & & 0 & 1 & 5 & 0 & 3 \\
 & \textbf{2} & & 4 & 0 & 0 & 1 & 2 \\
\textbf{Job} & \textbf{3} & & 2 & 2 & 5 & 2 & 0 \\
 & \textbf{4} & & 7 & 0 & 4 & 0 & 0 \\
 & \textbf{D}_5 & & 1 & 0 & 0 & 0 & 2
\end{array}$$

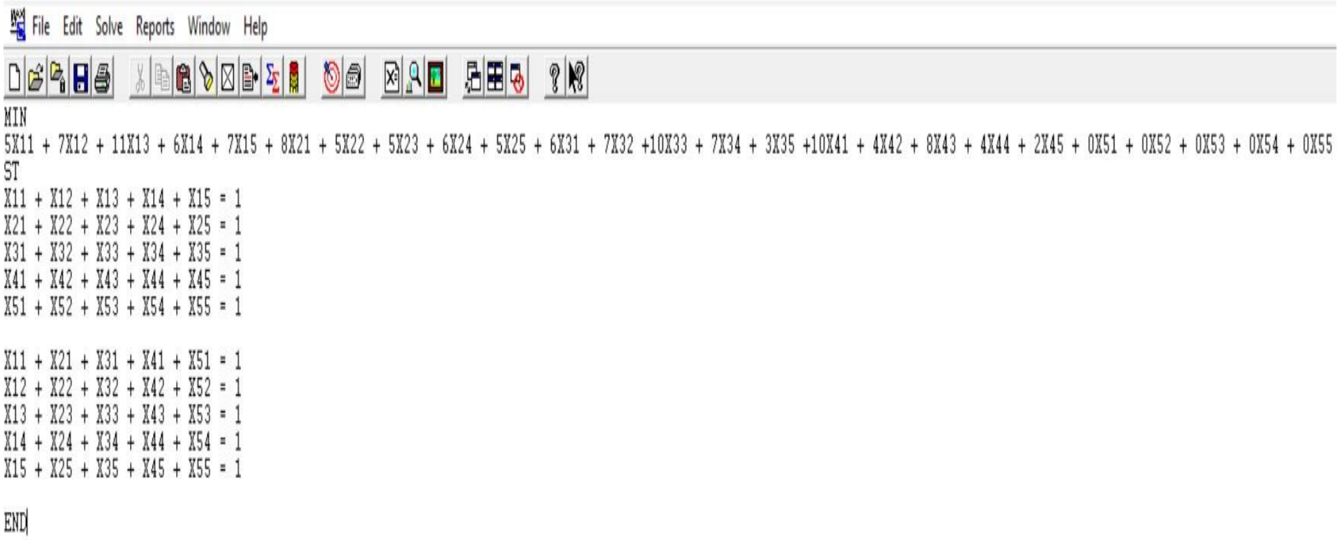
Number of lines drawn = Order of matrix. Hence optimality is reached. Now assign the jobs to machines, as shown in Table.

Assigning Jobs to Machines

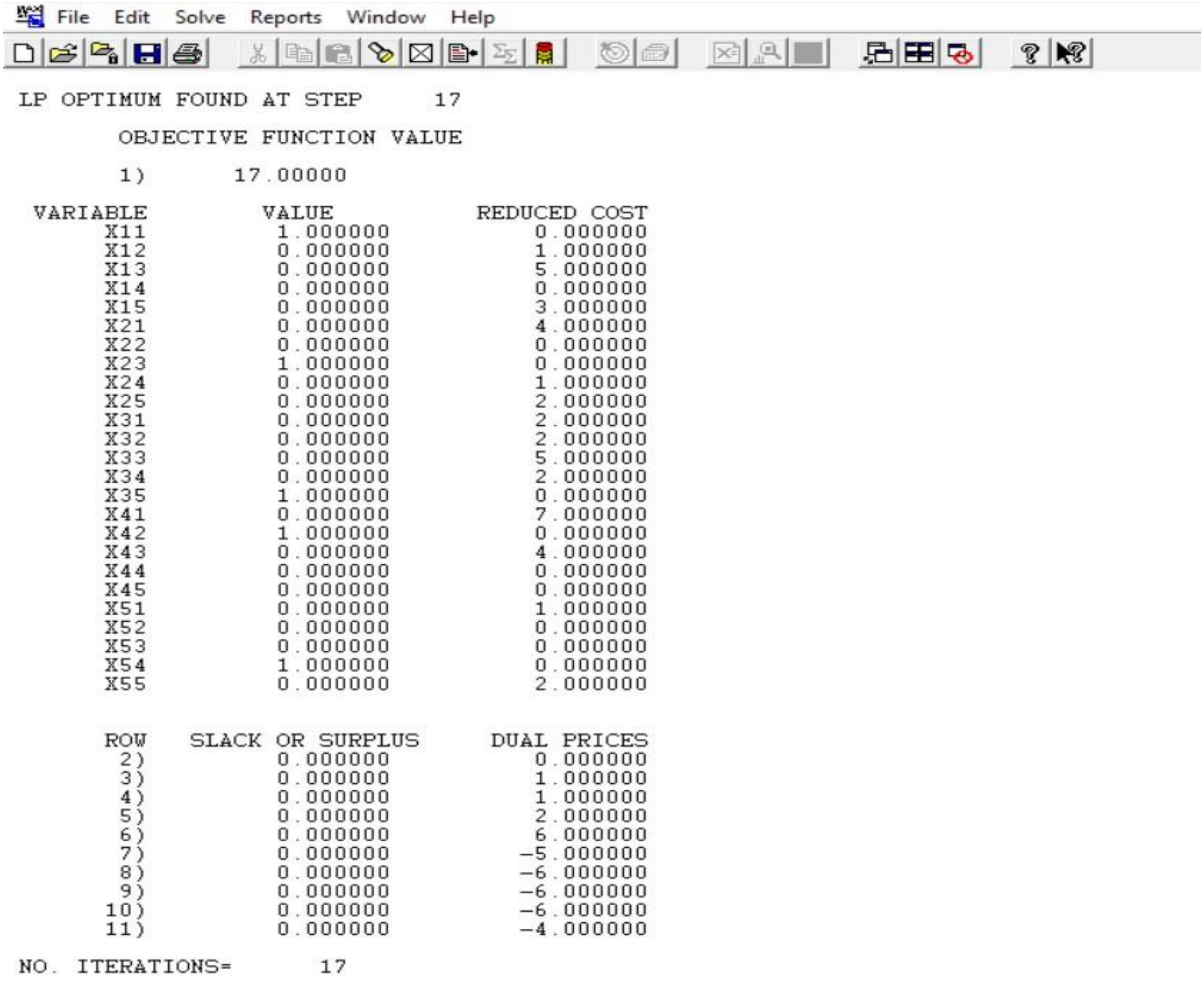
$$\begin{array}{cc|ccccc|}
 & & \multicolumn{5}{c}{\textbf{Machines}} \\
 & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & \boxed{0} & 1 & 5 & \cancel{0} & 3 \\
 & \textbf{2} & 4 & \boxed{0} & \cancel{0} & 1 & 2 \\
\textbf{Job} & \textbf{3} & 2 & 2 & 5 & 2 & \boxed{0} \\
 & \textbf{4} & 7 & \cancel{0} & 4 & \boxed{0} & \cancel{0} \\
 & \textbf{D}_5 & 1 & \cancel{0} & \boxed{0} & \cancel{0} & 2
\end{array}$$

Job	Machine	Cost
1	A	5
2	B	5
3	E	3
4	D	4
D5	C	0
	Total Cost	= Rs. 17

7. Unbalanced Assignment problems using LINDO:



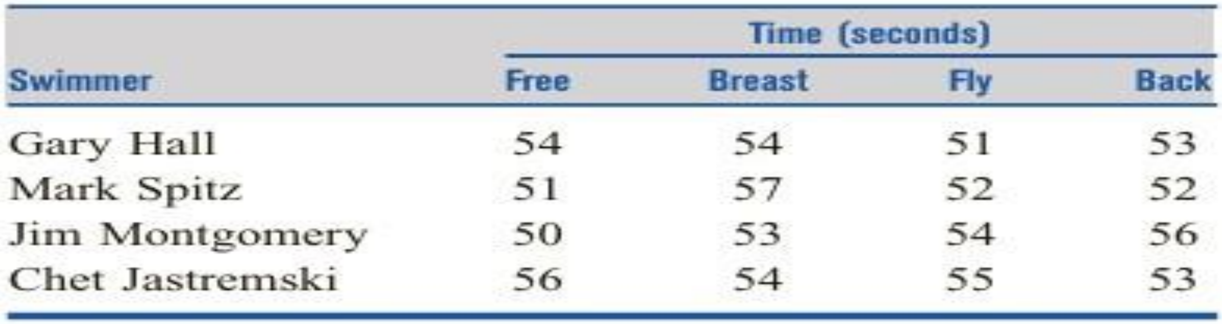
```
MIN
5X11 + 7X12 + 11X13 + 6X14 + 7X15 + 8X21 + 5X22 + 5X23 + 6X24 + 5X25 + 6X31 + 7X32 +10X33 + 7X34 + 3X35 +10X41 + 4X42 + 8X43 + 4X44 + 2X45 + 0X51 + 0X52 + 0X53 + 0X54 + 0X55
ST
X11 + X12 + X13 + X14 + X15 = 1
X21 + X22 + X23 + X24 + X25 = 1
X31 + X32 + X33 + X34 + X35 = 1
X41 + X42 + X43 + X44 + X45 = 1
X51 + X52 + X53 + X54 + X55 = 1

X11 + X21 + X31 + X41 + X51 = 1
X12 + X22 + X32 + X42 + X52 = 1
X13 + X23 + X33 + X43 + X53 = 1
X14 + X24 + X34 + X44 + X54 = 1
X15 + X25 + X35 + X45 + X55 = 1

END
```



```
 LP OPTIMUM FOUND AT STEP      17

        OBJECTIVE FUNCTION VALUE

        1)      17.00000

  VARIABLE        VALUE          REDUCED COST
       X11         1.000000          0.000000
       X12         0.000000          1.000000
       X13         0.000000          5.000000
       X14         0.000000          0.000000
       X15         0.000000          3.000000
       X21         0.000000          4.000000
       X22         0.000000          0.000000
       X23         1.000000          0.000000
       X24         0.000000          1.000000
       X25         0.000000          2.000000
       X31         0.000000          2.000000
       X32         0.000000          2.000000
       X33         0.000000          5.000000
       X34         0.000000          2.000000
       X35         1.000000          0.000000
       X41         0.000000          7.000000
       X42         1.000000          0.000000
       X43         0.000000          4.000000
       X44         0.000000          0.000000
       X45         0.000000          0.000000
       X51         0.000000          1.000000
       X52         0.000000          0.000000
       X53         0.000000          0.000000
       X54         1.000000          0.000000
       X55         0.000000          2.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          0.000000
        3)         0.000000          1.000000
        4)         0.000000          1.000000
        5)         0.000000          2.000000
        6)         0.000000          6.000000
        7)         0.000000         -5.000000
        8)         0.000000         -6.000000
        9)         0.000000         -6.000000
       10)         0.000000         -6.000000
       11)         0.000000         -4.000000

 NO. ITERATIONS=      17
```

Practice Problem:

1. Doc Councillman is putting together a relay team for the 400-meter relay. Each swimmer must swim 100 meters of breaststroke, backstroke, butterfly, or freestyle. Doc believes that each swimmer will attain the times given in Table. To minimize the team's time for the race, which swimmer should swim which stroke?



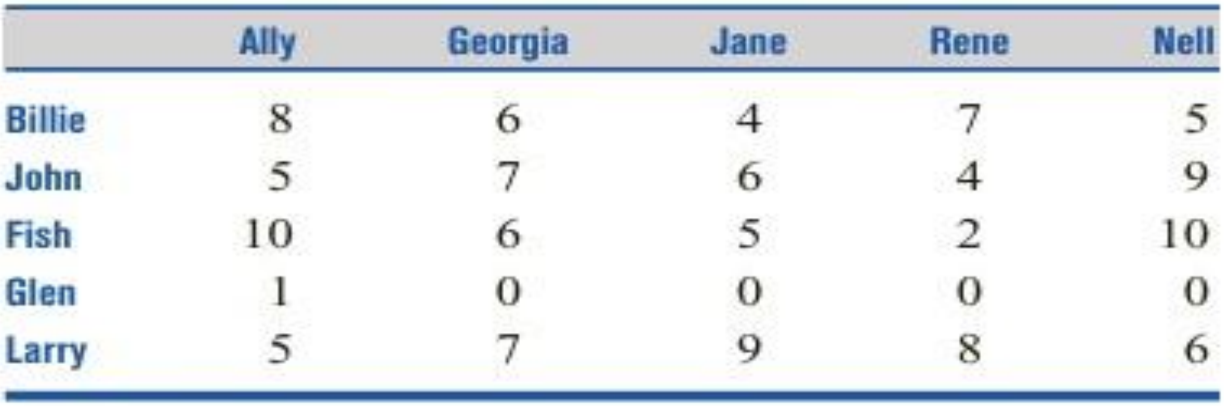
	Time (seconds)			
Swimmer	**Free**	**Breast**	**Fly**	**Back**
Gary Hall	54	54	51	53
Mark Spitz	51	57	52	52
Jim Montgomery	50	53	54	56
Chet Jastremski	56	54	55	53

2. Five male characters (Billie, John, Fish, Glen, and Larry) and five female characters (Ally, Georgia, Jane, Rene, and Nell) from Ally McBeal are marooned on a desert island. The problem is to determine what percentage of time each woman on the island should spend with each man. For example, Ally could spend 100% of her time with John or she could "play the field" by spending 20% of her time with each man. Table shows a "happiness index" for each potential pairing of a man and woman. For example, if Larry and Rene spend all their time together, they earn 8 units of happiness for the island.

i. Play matchmaker and determine an allocation of each man and woman's time that earns the maximum total happiness for the island. Assume that happiness earned by a couple is proportional to the amount of time they spend together.

ii. Explain why the optimal solution to this problem will, for any matrix of "happiness indices," always in volve each woman spending all her time with one man.

iii. What assumption made in the problem is needed for the Marriage Theorem to hold?



	Ally	**Georgia**	**Jane**	**Rene**	**Nell**
Billie	8	6	4	7	5
John	5	7	6	4	9
Fish	10	6	5	2	10
Glen	1	0	0	0	0
Larry	5	7	9	8	6

Experiment-7: Solving the Game Theory Problem Using LINDO.

1. Objective

By the end of this experiment, students will be able to:

- Formulate game theory problems.
- Solve Maximin-Minimax Principal problems.
- Solve Two-person zero-sum games mixed strategy problems using LINDO.

2. Introduction to Game Theory Problem

Game theory is a type of decision theory in which one's choice of action is determined after considering all possible alternatives available to an opponent playing the same game, rather than just by the possibilities of several outcome results. Game theory does not insist on how a game should be played but tells the procedure and principles by which action should be selected. Thus, it is a decision theory useful in competitive situations. Games are defined as an activity between two or more people according to a set of rules at the end of which each perceive some benefit or suffers loss. The set of rules defines the game. Going through the set of rules once by the participants defines a play.

3. Properties of Game Theory Problem

a) There are finite numbers of competitors called 'players'

b) Each player has a finite number of possible courses of action called 'strategies'

c) All the strategies and their effects are known to the players but player does not know which strategy is to be chosen.

d) A game is played when each player chooses one of his strategies. The strategies are assumed to be made simultaneously with an outcome such that no player knows his opponent's strategy until he decides his own strategy.

e) The game is a combination of the strategies and in certain units which determines the gain or loss.

f) The figures shown as the outcomes of strategies in a matrix form are called 'pay-off matrix'.

g) The player playing the game always tries to choose the best course of action which results in optimal pay off called 'optimal strategy'.

h) The expected pay off when all the players of the game follow their optimal strategies is known as ‘value of the game’. The main objective of a problem of a game is to find the value of the game.

4. Characteristics of Game Theory Problem

a) Competitive game: A competitive situation is called a competitive game if it has the following four properties

- There are finite number of competitors such that n ≥ 2. In case n = 2, it is called a two- person game and in case n > 2, it is referred to as n-person game.
- Each player has a list of finite number of possible activities.
- A play is said to occur when each player chooses one of his activities. The choices are assumed to be made simultaneously i.e. no player knows the choice of the other until he has decided on his own.
- Every combination of activities determines an outcome which results in a gain of payments to each player, provided each player is playing uncompromisingly to get as much as possible. Negative gain implies the loss of same amount.

b) Strategy: The strategy of a player is the predetermined rule by which player decides his course of action from his own list during the game.

The two types of strategy are

- Pure strategy
- Mixed strategy

Pure Strategy: If a player knows exactly what the other player is going to do, a deterministic situation is obtained, and objective function is to maximize the gain. Therefore, pure strategy is a decision rule always to select a particular course of action.
Mixed Strategy: If a player is guessing as to which activity is to be selected by the other on any particular occasion, a probabilistic situation is obtained, and objective function is to maximize the expected gain. Thus, the mixed strategy is a selection among pure strategies with fixed probabilities.

c) Number of persons: A game is called ‘n’ person game if the number of persons playing is ‘n’. The person means an individual or a group aiming at a particular objective.

Two-person, zero-sum game: A game with only two players (player A and player B) is called a ‘two-person, zero-sum game’, if the losses of one player are equivalent to the gains of the other so that the sum of their net gains is zero.

Two-person, zero-sum games are also called rectangular games as these are usually represented by a payoff matrix in a rectangular form.

d) Number of activities: The activities may be finite or infinite.
e) Payoff: The quantitative measure of satisfaction a person gets at the end of each play is called a payoff
f) Payoff matrix: Suppose the player A has ‘m’ activities and player B has ‘n’ activities. Then a payoff matrix can be formed by adopting the following rules
   i. Row designations for each matrix are the activities available to player A
   ii. Column designations for each matrix are the activities available to player B
   iii. Cell entry Vij is the payment to player A in A’s payoff matrix when A chooses the activity i and B chooses the activity j.
   iv. With a zero-sum, two-person game, the cell entry in the player B’s payoff matrix will be negative of the corresponding cell entry Vij in the player A’s payoff matrix so that sum of payoff matrices for player A and player B is ultimately zero.
i) Value of the game: Value of the game is the maximum guaranteed game to player A (maximizing player) if both the players uses their best strategies. It is generally denoted by ‘V’ and it is unique.

5. Classifications of Game Theory Problem

All games are classified into

- Pure strategy games
- Mixed strategy games

Strategy: It is the pre-determined rule by which each player decides his course of action from his list available to him. How one course of action is selected out of various courses available to him is known as strategy of the game.

Types of Strategy: Generally, two types of strategy are employed

(i) Pure Strategy
(ii) Mixed Strategy

(i) Pure Strategy: It is the predetermined course of action to be employed by the player. The players knew it in advance. It is usually represented by a number with which the course of action is associated.

(ii) Mixed Strategy: In mixed strategy the player decides his course of action in accordance with some fixed probability distribution. Probability are associated with each course of action and the selection is done as per these probabilities.

In mixed strategy the opponent cannot be sure of the course of action to be taken on any particular occasion. Pure strategy games can be solved by saddle point method.

Decision of a Game. In Game theory, best strategy for each player is determined on the basis of some rule. Since both the players are expected to be rational in their approach this is known as the criteria of optimality. Each player lists the possible outcomes from his action and selects the best action to achieve his objectives. This criterion of optimality is expressed as Maximin for the maximizing player and Minimax for the minimizing player.

6. The Maximin-Minimax Principal Problem

j) Maximin Criteria: The maximizing player lists his minimum gains from each strategy and selects the strategy which gives the maximum out of these minimum gains.

k) Minimax Criteria: The minimizing player lists his maximum loss from each strategy and selects the strategy which gives him the minimum loss out of these maximum losses.

For Example, consider a two-person zero sum game involving the set of pure strategy for Maximizing player A say Al A2 & As and for player B, B1 & B2, with the following payoff

Player B

		B1	B2	**Row Maximin**
	A1	9	2	2
Player A	A2	8	6	6* Maximin
	A3	6	4	4
	Column Minimax	9	6* Minimax	

Since Maximin = Minimax V = 6

Suppose that player A starts the game knowing fully well that whatever strategy he adopts B will select that particular counter strategy which will minimize the payoff to A. If A selects the strategy Al then B will select B2 so that A may get minimum gain. Similarly, if A chooses A2 than B will adopt the strategy of B2. Naturally A would like to maximize his maximin gain which is just the largest of row minima. Which is called 'maximin strategy'. Similarly, B will minimize his minimum loss which is called 'minimax strategy'. We observe that in the above example, the maximum if row minima and minimum of column maxima are equal. In symbols.

$$\text{Maxi [Min.]} = \text{Mini [Max]}$$

The strategies followed by both the p1ayersarecalled ‘optimum strategy’.

Value of Game. In game theory, the concept value of game is considered very important. The value of game is the maximum guaranteed gain to the maximizing player if both the players use their best strategy. It refers to the average payoff per play of the game over a period of time. Consider the following games.

$$\begin{array}{cc} & \text{Player Y} \\ \text{Player X} & \begin{pmatrix} 3 & 4 \\ -6 & -2 \end{pmatrix} \end{array} \qquad \begin{array}{cc} & \text{Player Y} \\ \text{Player X} & \begin{pmatrix} -7 & 2 \\ -3 & -1 \end{pmatrix} \end{array}$$

(With Positive Value) (With Negative Value)

In the first game player X wins 3 points and the value of the value is three with positive sign and in the second game the player Y wins 3 points, and the value of the game is -ve which indicates that Y is the Winner. The value is denoted by 'v.

The different methods for solving a mixed strategy game are

- Analytical method
- Graphical method
- Dominance rule

7. Two Person and Zero-Sum Game Problem

Two-person zero-sum games may be deterministic or probabilistic. The deterministic games will have saddle points and pure strategies exist in such games. In contrast, the probabilistic games will have no saddle points and mixed strategies are taken with the help of probabilities.

Definition of saddle point

A saddle point of a matrix is the position of such an element in the payoff matrix, which is minimum in its row and the maximum in its column

Procedure to find the saddle point

- Select the minimum element of each row of the payoff matrix and mark them with circles.
- Select the maximum element of each column of the payoff matrix and mark them with squares.
- If them appears an element in the payoff matrix with a circle and a square together then that position is called saddle point and the element is the value of the game.

Solution of games with saddle point

To obtain a solution of a game with a saddle point, it is feasible to find out

- Best strategy for player A
- Best strategy for player B
- The value of the game

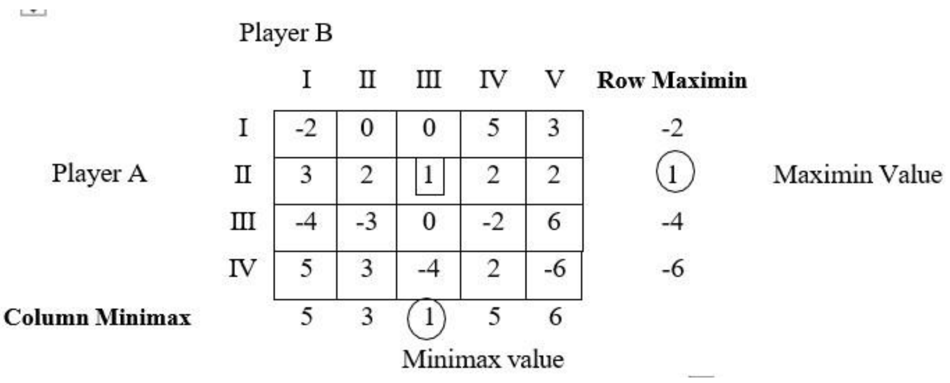
The best strategies for player A and B will be those which correspond to the row and column respectively through the saddle point.

Example 1: Solve the payoff matrix

Player B

		I	II	III	IV	V
Player A	I	-2	0	0	5	3
	II	3	2	1	2	2
	III	-4	-3	0	-2	6
	IV	5	3	-4	2	-6

Solution:



Player B

		I	II	III	IV	V	**Row Maximin**	
	I	-2	0	0	5	3	-2	
Player A	II	3	2	$\boxed{1}$	2	2	①	Maximin Value
	III	-4	-3	0	-2	6	-4	
	IV	5	3	-4	2	-6	-6	
Column Minimax		5	3	①	5	6		
				Minimax value				

Strategy of player A – II
Strategy of player B - III
Value of the game = 1

Example -2: Solve the payoff matrix

	B1	**B2**	**B3**	**B4**
A1	1	7	3	4
A2	5	6	4	5
A3	7	2	0	3

Solution

	B1	**B2**	**B3**	**B4**	**Row Maximin**	
A1	1	7	3	4	**1**	
A2	5	6	$\boxed{4}$	5	**④**	**Maximin Value**
A3	7	2	0	3	**0**	
Column Minimax	**7**	**7**	**④**	**5**		

Minimax Value

Strategy of player A= A2

Strategy of player B = B3

Value of the game = 4

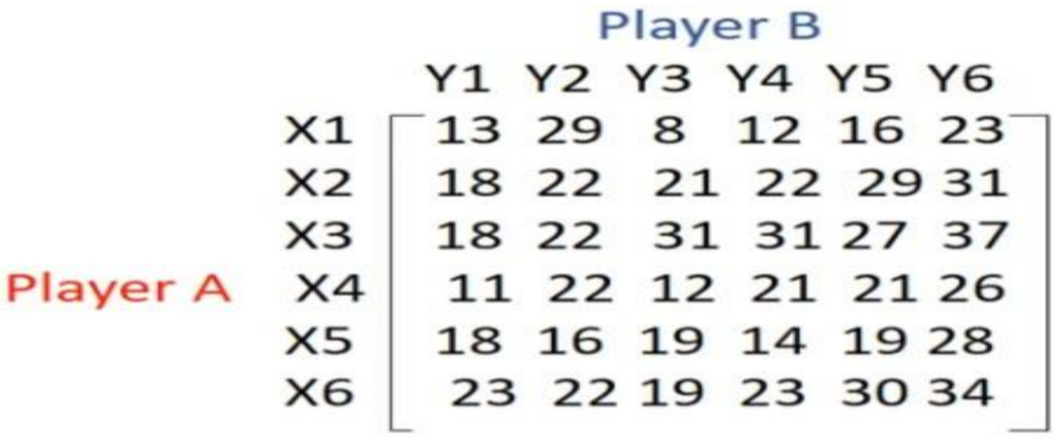
Points to remember:

I. Saddle Points may or may not exist in a given game.
II. There may be more than one saddle point then there will be more than one solution. (Such situation is rare in the rea1life).

III. The value of game may be +ve or -ve.
IV. The value of game may be zero which means 'fair game'.

7. Two-person zero-sum games mixed strategy problems using LINDO:

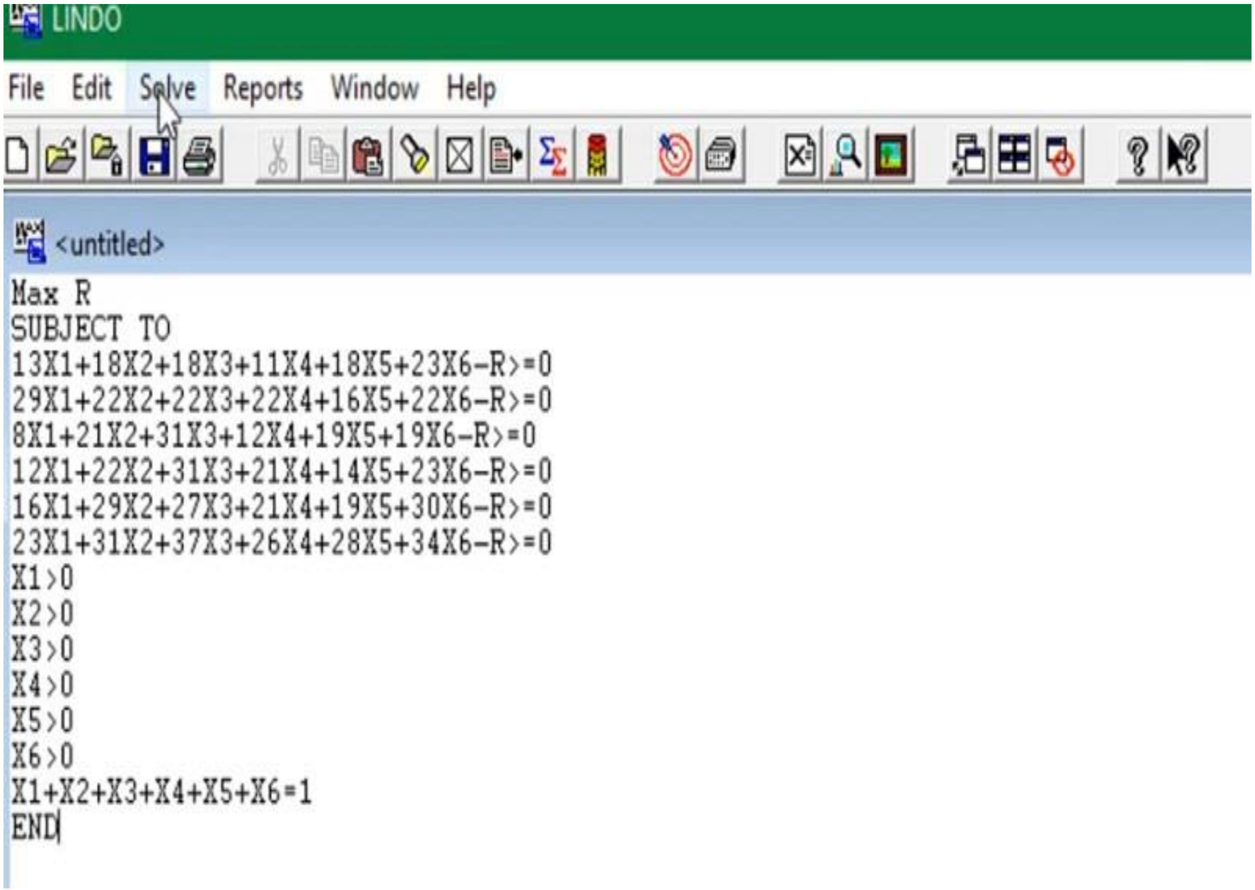
Example -1: Solve the payoff matrix using LINDO



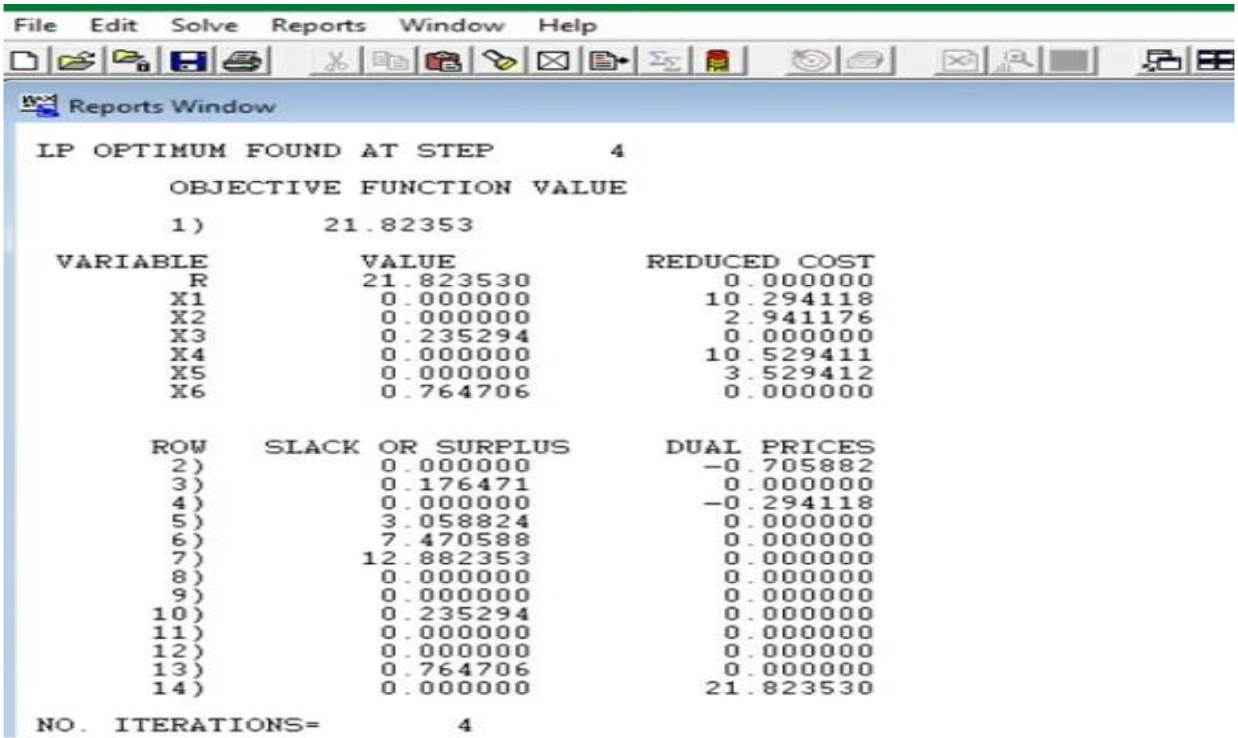
$$\begin{array}{cc|cccccc|} & & \multicolumn{6}{c}{\text{Player B}} \\ & & Y1 & Y2 & Y3 & Y4 & Y5 & Y6 \\ & X1 & 13 & 29 & 8 & 12 & 16 & 23 \\ & X2 & 18 & 22 & 21 & 22 & 29 & 31 \\ & X3 & 18 & 22 & 31 & 31 & 27 & 37 \\ \text{Player A} & X4 & 11 & 22 & 12 & 21 & 21 & 26 \\ & X5 & 18 & 16 & 19 & 14 & 19 & 28 \\ & X6 & 23 & 22 & 19 & 23 & 30 & 34 \end{array}$$

Solution:

For player A

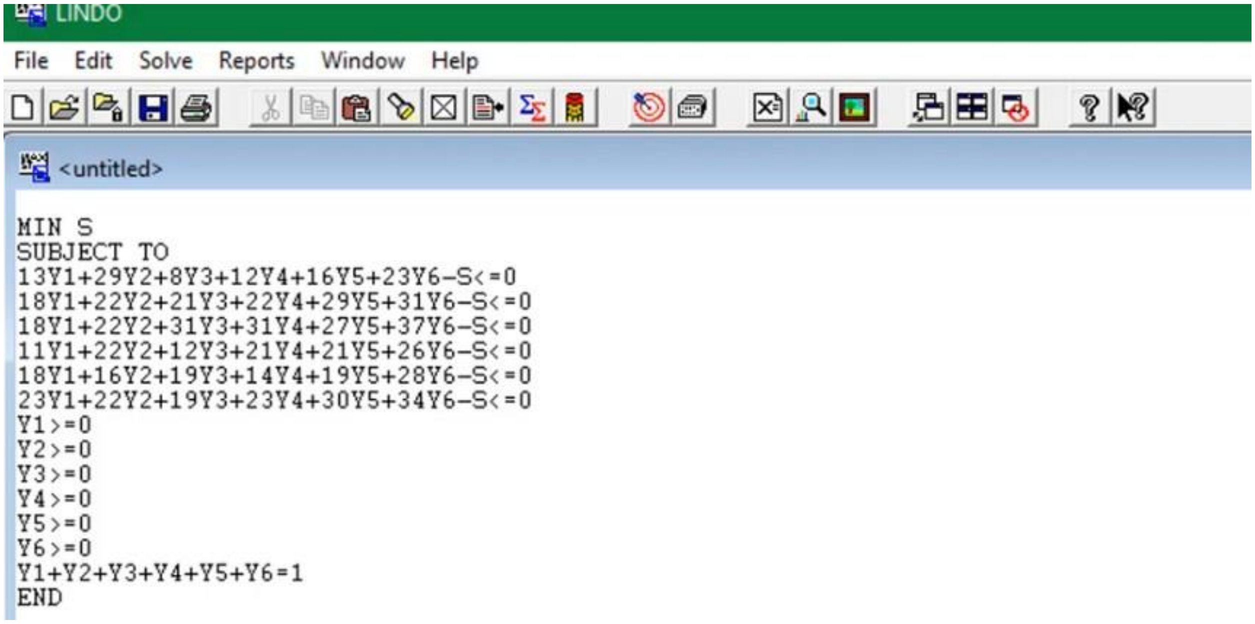


```
Max R
SUBJECT TO
13X1+18X2+18X3+11X4+18X5+23X6-R>=0
29X1+22X2+22X3+22X4+16X5+22X6-R>=0
8X1+21X2+31X3+12X4+19X5+19X6-R>=0
12X1+22X2+31X3+21X4+14X5+23X6-R>=0
16X1+29X2+27X3+21X4+19X5+30X6-R>=0
23X1+31X2+37X3+26X4+28X5+34X6-R>=0
X1>0
X2>0
X3>0
X4>0
X5>0
X6>0
X1+X2+X3+X4+X5+X6=1
END
```

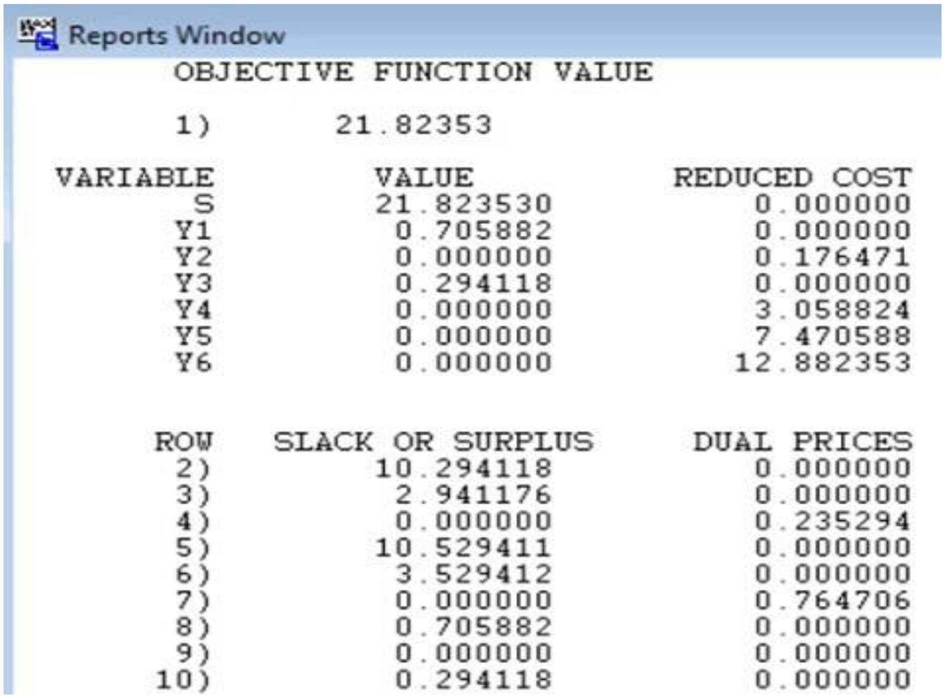


```
 LP OPTIMUM FOUND AT STEP        4

        OBJECTIVE FUNCTION VALUE

        1)       21.82353

  VARIABLE        VALUE          REDUCED COST
         R        21.823530          0.000000
        X1         0.000000         10.294118
        X2         0.000000          2.941176
        X3         0.235294          0.000000
        X4         0.000000         10.529411
        X5         0.000000          3.529412
        X6         0.764706          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000         -0.705882
        3)         0.176471          0.000000
        4)         0.000000         -0.294118
        5)         3.058824          0.000000
        6)         7.470588          0.000000
        7)        12.882353          0.000000
        8)         0.000000          0.000000
        9)         0.000000          0.000000
       10)         0.235294          0.000000
       11)         0.000000          0.000000
       12)         0.000000          0.000000
       13)         0.764706          0.000000
       14)         0.000000         21.823530

 NO. ITERATIONS=        4
```

For Player B



```
MIN S
SUBJECT TO
13Y1+29Y2+8Y3+12Y4+16Y5+23Y6-S<=0
18Y1+22Y2+21Y3+22Y4+29Y5+31Y6-S<=0
18Y1+22Y2+31Y3+31Y4+27Y5+37Y6-S<=0
11Y1+22Y2+12Y3+21Y4+21Y5+26Y6-S<=0
18Y1+16Y2+19Y3+14Y4+19Y5+28Y6-S<=0
23Y1+22Y2+19Y3+23Y4+30Y5+34Y6-S<=0
Y1>=0
Y2>=0
Y3>=0
Y4>=0
Y5>=0
Y6>=0
Y1+Y2+Y3+Y4+Y5+Y6=1
END
```



```
        OBJECTIVE FUNCTION VALUE

        1)      21.82353

  VARIABLE        VALUE          REDUCED COST
         S       21.823530          0.000000
        Y1        0.705882          0.000000
        Y2        0.000000          0.176471
        Y3        0.294118          0.000000
        Y4        0.000000          3.058824
        Y5        0.000000          7.470588
        Y6        0.000000         12.882353


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)       10.294118          0.000000
        3)        2.941176          0.000000
        4)        0.000000          0.235294
        5)       10.529411          0.000000
        6)        3.529412          0.000000
        7)        0.000000          0.764706
        8)        0.705882          0.000000
        9)        0.000000          0.000000
       10)        0.294118          0.000000
```

Practice Problem:

1. Find each player's optimal strategy and the value of the two-person zero-sum game in Table.

4	5	1	4
2	1	6	3
1	0	0	2

2. Find each player's optimal strategy and the value of the two-person zero-sum game in Table.

2	4	6
3	1	5

Experiment-8: Solving Binary Integer Programming (BIP) problems using LINDO

1. Objective

By the end of this experiment, students will be able to:

- Differentiate different type of Integer Programming (IP) Problem
- Formulate Binary Integer Programming problems.
- Solve BIP problems using LINDO

2. Integer Programming

If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an integer programming (IP) problem.

2.1 Types of Integer Programming Problems

Pure Integer Programming Problem: An IP in which all variables are required to be integers is called a pure integer programming problem.

$$\begin{aligned} &\max z = 3x_1 + 2x_2 \\ &\text{s.t.} \qquad x_1 + x_2 \le 6 \\ &x_1, x_2 \ge 0,\ x_1, x_2 \text{ integer} \end{aligned}$$

Mixed Integer Programming Problem (MIP): An IP in which only some of the variables are required to be integers is called a mixed integer programming problem.

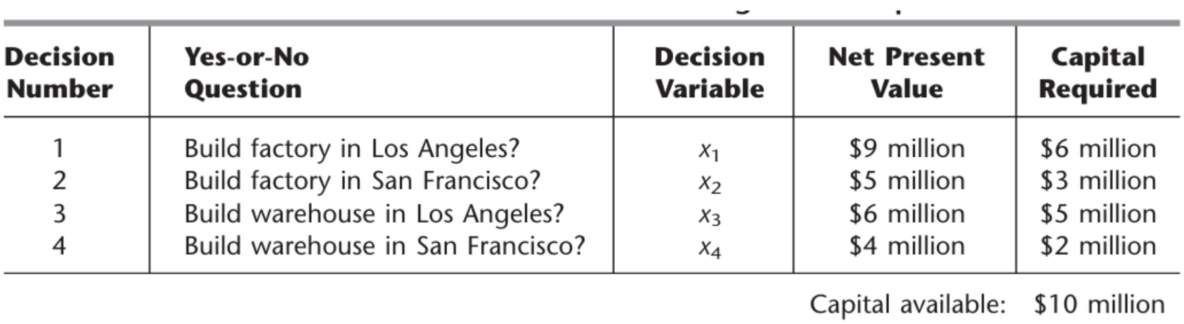
$$\begin{aligned} &\max z = 3x_1 + 2x_2 \\ &\text{s.t.} \qquad x_1 + x_2 \le 6 \\ &x_1, x_2 \ge 0,\ x_1 \text{ integer} \end{aligned}$$

Binary Integer Programming Problem (BIP): An integer programming problem in which all the variables must equal 0 or 1 is called a 0–1 IP. Such variables are called binary variables (or 0–1 variables). Consequently, IP problems that contain only binary variables sometimes are called binary integer programming (BIP) problems

2.2 BIP Model Formulation:

The CALIFORNIA MANUFACTURING COMPANY is considering expansion by building a new factory in either Los Angeles or San Francisco or perhaps even in both cities. It also is considering building at most one new warehouse, but the choice of location is restricted to a city where a new factory is being built. The net present value (total profitability considering the time value of money) of each of these alternatives is shown in the table 2.2.1. The rightmost column gives the capital required (already included in the net present value) for the respective investments, where the total capital available is $10 million. The objective is to find the feasible combination of alternatives that maximizes the total net present value.

Table 2.2.1: NPV and capital requirement of the facilities



Decision Number	Yes-or-No Question	Decision Variable	Net Present Value	Capital Required
1	Build factory in Los Angeles?	x_1	$9 million	$6 million
2	Build factory in San Francisco?	x_2	$5 million	$3 million
3	Build warehouse in Los Angeles?	x_3	$6 million	$5 million
4	Build warehouse in San Francisco?	x_4	$4 million	$2 million

Capital available: $10 million

Solution:

As all the decision variables have the binary form, we can represent the variables as,

$$x_j = \begin{cases} 1 & \text{if decision } j \text{ is yes,} \\ 0 & \text{if decision } j \text{ is no,} \end{cases} \qquad (j = 1, 2, 3, 4).$$

Let

Z = total net present value of these decisions.

using units of millions of dollars,

$$Z = 9x_1 + 5x_2 + 6x_3 + 4x_4.$$

the amount of capital expended on the four facilities cannot exceed $10 million. Consequently, continuing to use units of mil lions of dollars, one constraint in the model is

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10.$$

Because the last two decisions represent *mutually exclusive alternatives* (the company wants *at most* one new warehouse), we also need the constraint

$$x_3 + x_4 \leq 1.$$

Furthermore, decisions 3 and 4 are contingent decisions, because they are contingent on decisions 1 and 2, respectively (the company would consider building a warehouse in a city only if a new factory also were going there). Thus, in the case of decision 3, we require that $x_3 = 0$ if $x_1=0$. This restriction on x_3 is represented by

$$x_3 \leq x_1.$$

Similarly, the requirement that $x_4=0$ if $x_2=0$ is imposed by adding the constraint.

$$x_4 \leq x_2.$$

Therefore, after we rewrite these two constraints to bring all variables to the left-hand side, the complete BIP model is,

$$\text{Maximize} \qquad Z = 9x_1 + 5x_2 + 6x_3 + 4x_4,$$

subject to

$$\begin{array}{rrrrl}
6x_1 & + 3x_2 & + 5x_3 & + 2x_4 & \leq 10 \\
 & & x_3 & + \ x_4 & \leq \ 1 \\
-x_1 & & + \ x_3 & & \leq \ 0 \\
 & - \ x_2 & & + \ x_4 & \leq \ 0 \\
 & & & x_j & \leq \ 1 \\
 & & & x_j & \geq \ 0
\end{array}$$

and

$$x_j \text{ is integer}, \qquad \text{for } j = 1, 2, 3, 4.$$

Equivalently, the last three lines of this model can be replaced by the single restriction,

$$x_j \text{ is binary}, \qquad \text{for } j = 1, 2, 3, 4.$$

3. Solving BIP Model

3.1 LP Relaxation: The LP obtained by omitting all integer or 0–1 constraint on variables is called the LP relaxation of the IP. the LP relaxation is a less constrained, or more relaxed, version of the IP. This means that the feasible region for any IP must be contained in the feasible region for the corresponding LP relaxation. For any IP that is a max problem, this implies that,

Optimal z-value for LP relaxation > = Optimal z-value for IP

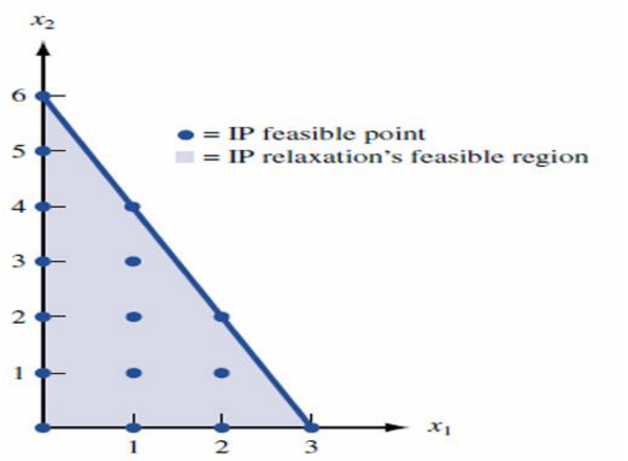


Figure 3.1.1: Feasible area of IP problems

3.2 Branch and Bound algorithm to solve BIP

Because any bounded pure IP problem has only a finite number of feasible solutions, it is natural to consider using some kind of enumeration procedure for finding an optimal solution. The basic concept underlying the branch-and-bound technique is to divide and conquer. Since the original “large” problem is too difficult to be solved directly, it is divided into smaller and smaller subproblems until these subproblems can be conquered. The dividing (branching) is done by partitioning the entire set of feasible solutions into smaller and smaller subsets. The conquering (fathoming) is done partially by bounding how good the best solution in the subset can be and then discarding the subset if its bound indicates that it cannot possibly contain an optimal solution for the original problem.

Branching: When you are dealing with binary variables, the most straightforward way to partition the set of feasible solutions into subsets is to fix the value of one of the variables (say, x1) at x1=0 for one subset and at x1=1 for the other subset. Doing this for a BIP problem divides the whole problem into the two smaller subproblems

Bounding: For each of these subproblems, we now need to obtain a bound on how good its best feasible solution can be. The standard way of doing this is to quickly solve a simpler relaxation of the subproblem. In most cases, a relaxation of a problem is obtained simply by deleting (“relaxing”) one set of constraints that had made the problem difficult to solve.

Fathoming: A subproblem can be conquered (fathomed), and thereby dismissed from further consideration, in the three ways described below.

Test 1: Its bound <= Z*, or

Test 2: Its LP relaxation has no feasible solutions

Test 3: The optimal solution for its LP relaxation is integer. (If this solution is better than the incumbent, it becomes the new incumbent, and test 1 is reapplied to all unfathomed subproblems with the new larger Z*.)

3.3 Solving the California Manufacturing Company problem with branch and bound technique

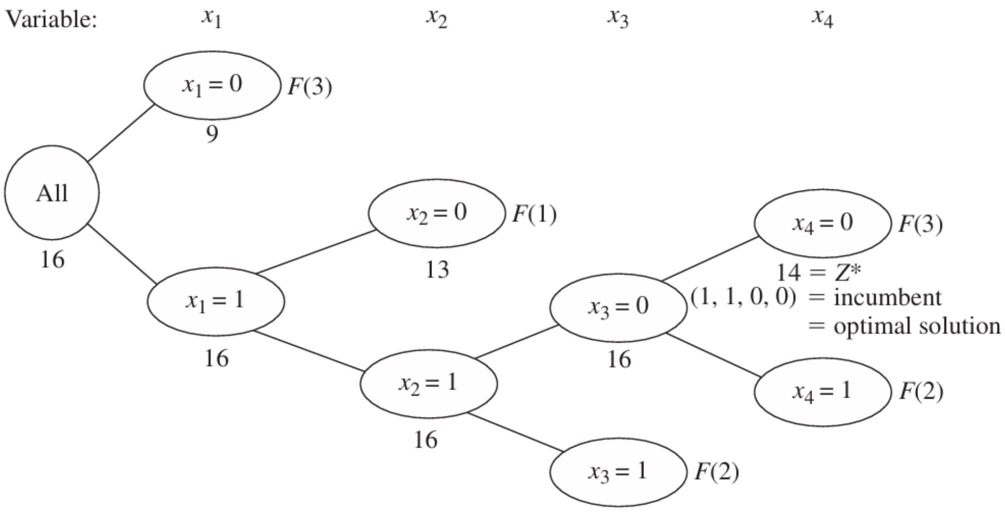


Figure 3.3.1: Solution of the California Manufacturing Company problem

4. Solving IP with LINDO/LINGO

To use LINDO to solve an IP, begin by entering the problem as if it were an LP. After typing in the END statement (to designate the end of the LP constraints), type for each 0–1 variable *x* the following statement:

INTE *x*

To tell LINDO that the first n variables appearing in the formulation must be 0–1 variable, use the command

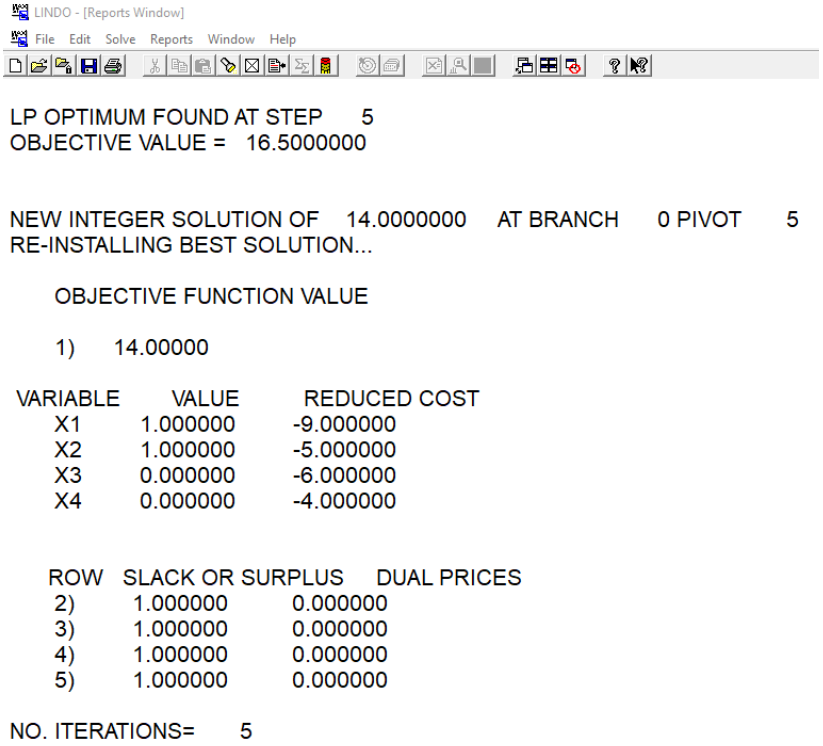
INT n.

LINGO can also be used to solve IPs. To indicate that a variable must equal 0 or 1 use the @BIN operator

4.1 Solving the California Manufacturing Company problem using LINDO

```
MAX
   9x1 + 5x2 + 6x3 + 4x4
ST
6x1 + 3x2 + 5x3 + 2x4 <= 10
x3 + x4 <= 1
- x1 + x3 <= 0
- x2 + x4 <= 0
END
INTE x1
INTE x2
INTE x3
INTE x4
```

Output:



```
LP OPTIMUM FOUND AT STEP      5
OBJECTIVE VALUE =   16.5000000


NEW INTEGER SOLUTION OF    14.0000000    AT BRANCH      0 PIVOT      5
RE-INSTALLING BEST SOLUTION...

      OBJECTIVE FUNCTION VALUE

      1)      14.00000

 VARIABLE        VALUE          REDUCED COST
      X1         1.000000        -9.000000
      X2         1.000000        -5.000000
      X3         0.000000        -6.000000
      X4         0.000000        -4.000000


     ROW   SLACK OR SURPLUS     DUAL PRICES
      2)         1.000000         0.000000
      3)         1.000000         0.000000
      4)         1.000000         0.000000
      5)         1.000000         0.000000

NO. ITERATIONS=       5
```

Figure 4.1.1: LINDO output of California Manufacturing Company problem

5. Exercise: Solving the lockbox problem using BIP Model

J. C. Nickles receives credit card payments from four regions of the country (West, Midwest, East, and South). The average daily value of payments mailed by customers from each region is as follows: the West, $70,000; the Midwest, $50,000; the East, $60,000; the South, $40,000. Nickles must decide where customers should mail their payments. Because Nickles can earn 20% annual interest by investing these revenues, it would like to receive payments as quickly as possible. Nickles is considering setting up operations to process payments (often referred to as lockboxes) in four different cities: Los Angeles, Chicago, New York, and Atlanta. The average number of days (from time payment is sent) until a check clears and Nickles can deposit the money depends on the city to which the payment is mailed, as shown in the table. For example, if a check is mailed from the West to Atlanta, it would take an average of 8 days before Nickles could earn interest on the check. The annual cost of running a lockbox in any city is $50,000.

Table 5.1: Time delay data

	To			
From	City 1 (Los Angeles)	City 2 (Chicago)	City 3 (New York)	City 4 (Atlanta)
Region 1 West	2	6	8	8
Region 2 Midwest	6	2	5	5
Region 3 East	8	5	2	5
Region 4 South	8	5	5	2

Formulate an IP that Nickles can use to minimize the sum of costs due to lost interest and lockbox operations. Assume that each region must send all its money to a single city and that there is no limit on the amount of money that each lockbox can handle.

Solution:

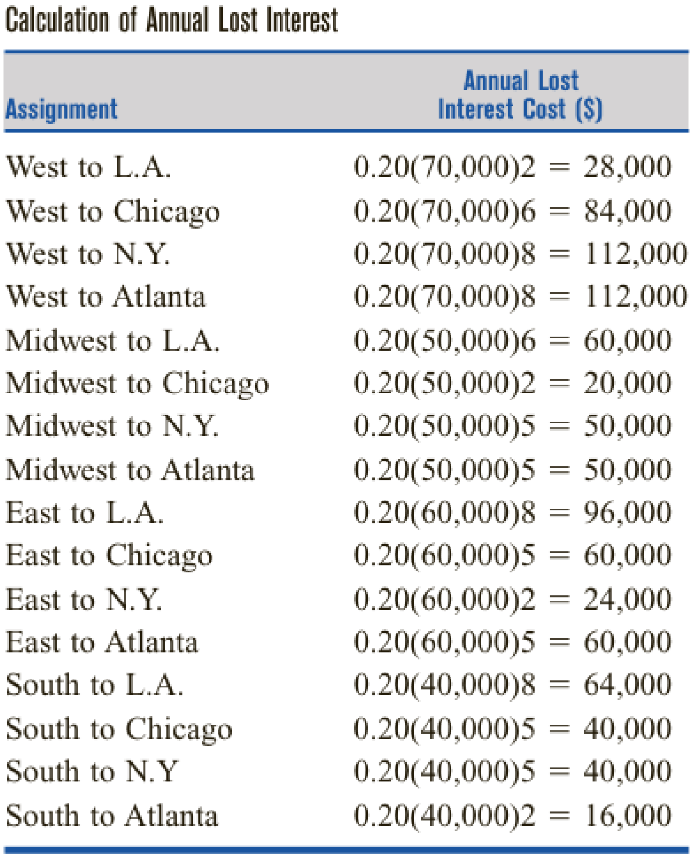
Nickles must make two types of decisions. First, Nickles must decide where to operate lockboxes. We define, *j* =1,2,3,4

$$y_j = \begin{cases} 1 & \text{if a lockbox is operated in city } j \\ 0 & \text{otherwise} \end{cases}$$

Second, Nickles must determine where each region of the country should send payments. We define (*i*,*j* = 1,2,3,4)

$$x_{ij} = \begin{cases} 1 & \text{if region } i \text{ sends payments to city } j \\ 0 & \text{otherwise} \end{cases}$$

Nickles wants to minimize total annual cost and annual lost interest cost. To determine how much interest Nickles loses annually, we must determine how much revenue would be lost if payments from region *i* were sent to region *j*. For example, how much in annual interest would Nickles lose if customers from the West region sent payments to New York? On any given day, 8 days' worth, or 8(70,000) $560,000 of West payments will be in the mail and will not be earning interest.



Calculation of Annual Lost Interest

Assignment	Annual Lost Interest Cost ($)
West to L.A.	0.20(70,000)2 = 28,000
West to Chicago	0.20(70,000)6 = 84,000
West to N.Y.	0.20(70,000)8 = 112,000
West to Atlanta	0.20(70,000)8 = 112,000
Midwest to L.A.	0.20(50,000)6 = 60,000
Midwest to Chicago	0.20(50,000)2 = 20,000
Midwest to N.Y.	0.20(50,000)5 = 50,000
Midwest to Atlanta	0.20(50,000)5 = 50,000
East to L.A.	0.20(60,000)8 = 96,000
East to Chicago	0.20(60,000)5 = 60,000
East to N.Y.	0.20(60,000)2 = 24,000
East to Atlanta	0.20(60,000)5 = 60,000
South to L.A.	0.20(40,000)8 = 64,000
South to Chicago	0.20(40,000)5 = 40,000
South to N.Y	0.20(40,000)5 = 40,000
South to Atlanta	0.20(40,000)2 = 16,000

Because Nickles can earn 20% annually, each year West funds will result in 0.20(560,000) = \$112,000 in lost interest. Similar calculations for the annual cost of lost interest for each possible assignment of a region to a city yield the results shown in the Table. The lost interest cost from sending region i's payments to city j is only incurred if $x_{ij} = 1$.

The type 1 constraints state that for region *i* (*i* = 1, 2, 3, 4) exactly one of x_{i1}, x_{i2}, x_{i3}, and x_{i4} must equal 1 and the others must equal 0. This can be accomplished by including the following four constraints

$$
\begin{aligned}
&x_{11} + x_{12} + x_{13} + x_{14} = 1 && \text{(West region constraint)}\\
&x_{21} + x_{22} + x_{23} + x_{24} = 1 && \text{(Midwest region constraint)}\\
&x_{31} + x_{32} + x_{33} + x_{34} = 1 && \text{(East region constraint)}\\
&x_{41} + x_{42} + x_{43} + x_{44} = 1 && \text{(South region constraint)}
\end{aligned}
$$

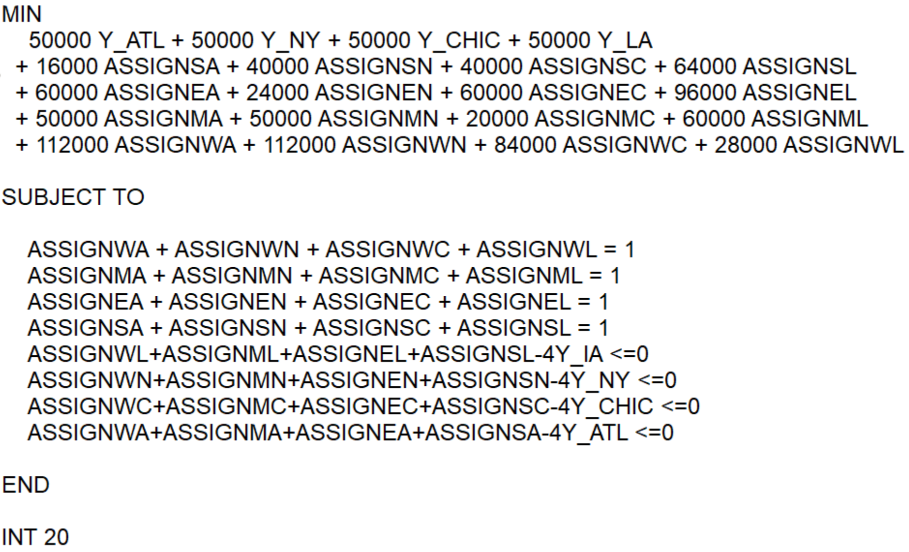
The type 2 constraints state that if

$$x_{ij} = 1 \qquad \text{(that is, customers in region } i \text{ send payments to city } j\text{)}$$

$$
\begin{aligned}
&x_{11} + x_{21} + x_{31} + x_{41} \le 4y_1 && \text{(Los Angeles constraint)}\\
&x_{12} + x_{22} + x_{32} + x_{42} \le 4y_2 && \text{(Chicago constraint)}\\
&x_{13} + x_{23} + x_{33} + x_{43} \le 4y_3 && \text{(New York constraint)}\\
&x_{14} + x_{24} + x_{34} + x_{44} \le 4y_4 && \text{(Atlanta constraint)}
\end{aligned}
$$

LINDO Code:

Thus, Nickles

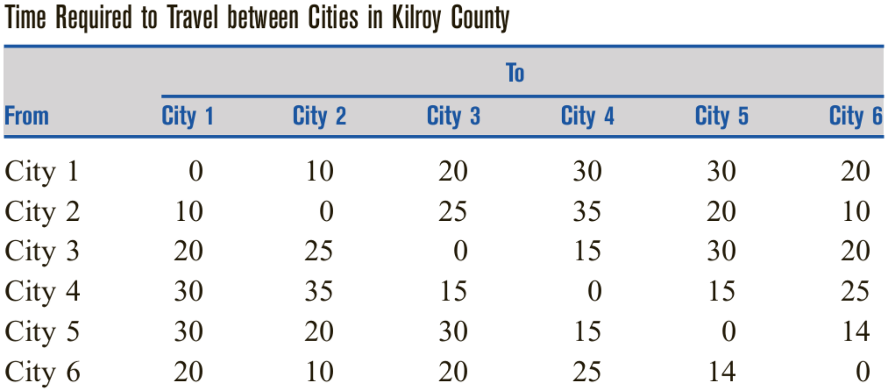


```
MIN
   50000 Y_ATL + 50000 Y_NY + 50000 Y_CHIC + 50000 Y_LA
 + 16000 ASSIGNSA + 40000 ASSIGNSN + 40000 ASSIGNSC + 64000 ASSIGNSL
 + 60000 ASSIGNEA + 24000 ASSIGNEN + 60000 ASSIGNEC + 96000 ASSIGNEL
 + 50000 ASSIGNMA + 50000 ASSIGNMN + 20000 ASSIGNMC + 60000 ASSIGNML
 + 112000 ASSIGNWA + 112000 ASSIGNWN + 84000 ASSIGNWC + 28000 ASSIGNWL

SUBJECT TO

   ASSIGNWA + ASSIGNWN + ASSIGNWC + ASSIGNWL = 1
   ASSIGNMA + ASSIGNMN + ASSIGNMC + ASSIGNML = 1
   ASSIGNEA + ASSIGNEN + ASSIGNEC + ASSIGNEL = 1
   ASSIGNSA + ASSIGNSN + ASSIGNSC + ASSIGNSL = 1
   ASSIGNWL+ASSIGNML+ASSIGNEL+ASSIGNSL-4Y_IA <=0
   ASSIGNWN+ASSIGNMN+ASSIGNEN+ASSIGNSN-4Y_NY <=0
   ASSIGNWC+ASSIGNMC+ASSIGNEC+ASSIGNSC-4Y_CHIC <=0
   ASSIGNWA+ASSIGNMA+ASSIGNEA+ASSIGNSA-4Y_ATL <=0

END

INT 20
```

Practice Problem

There are six cities (cities 1–6) in Kilroy County. The county must determine where to build fire stations. The county wants to build the minimum number of fire stations needed to ensure that at least one fire station is within 15 minutes (driving time) of each city. The times (in minutes) required to drive between the cities in Kilroy County are shown in Table 5.2. Formulate an IP that will tell Kilroy how many fire stations should be built and where they should be located.

Table 5.2: Time required to travel between cities



Time Required to Travel between Cities in Kilroy County

	To					
From	**City 1**	**City 2**	**City 3**	**City 4**	**City 5**	**City 6**
City 1	0	10	20	30	30	20
City 2	10	0	25	35	20	10
City 3	20	25	0	15	30	20
City 4	30	35	15	0	15	25
City 5	30	20	30	15	0	14
City 6	20	10	20	25	14	0

Experiment-9: Solving Mixed Integer Programming (MIP) problems using LINDO

1. Objective

By the end of this experiment, students will be able to:

- Differentiate different type of Integer Programming (IP) Problem
- Formulate Mixed Integer Programming problems.
- Solve MIP problems using LINDO

2. Integer Programming

If requiring integer values is the only way in which a problem deviates from a linear programming formulation, then it is an integer programming (IP) problem.

2.1 Mixed Integer Programming

Mixed Integer Programming Problem (MIP): An IP in which only some of the variables are required to be integers is called a mixed integer programming problem. The following problem can be classified as a Mixed Integer Programming problem as x_1,x_2 and x_3 is required to be integer whereas x4 is not.

$$\text{Maximize} \qquad Z = 4x_1 - 2x_2 + 7x_3 - x_4,$$

subject to

$$\begin{aligned}
x_1 \qquad\quad + 5x_3 \qquad\quad &\le 10\\
x_1 + x_2 - x_3 \qquad\quad &\le 1\\
6x_1 - 5x_2 \qquad\qquad\quad &\le 0\\
-x_1 \qquad\quad + 2x_3 - 2x_4 &\le 3
\end{aligned}$$

and

$$\begin{aligned}
&x_j \ge 0, \qquad \text{for } j = 1, 2, 3, 4\\
&x_j \text{ is an integer}, \qquad \text{for } j = 1, 2, 3.
\end{aligned}$$

3 Solving Mixed Integer Programming problems

3.1 LP Relaxation: The LP for MIP is obtained by omitting all integer constraint on the integer variables. The LP relaxation is less constrained, or more relaxed, version of the IP. This means that the feasible region for any IP must be contained in the feasible region for the corresponding LP relaxation. For any IP that is a max problem, this implies that,

Optimal z-value for LP relaxation > = Optimal z-value for IP

3.2 Branch and Bound algorithm to solve MIP

We shall now consider the general MIP problem, where some of the variables (say, I of them) are restricted to integer values (but not necessarily just 0 and 1) but the rest are ordinary continuous variables. For notational convenience, we shall order the variables so that the first I variables are the integer-restricted variables. Therefore, the general form of the problem being considered is

$$\text{Maximize} \qquad Z = \sum_{j=1}^{n} c_j x_j,$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad \text{for } i = 1, 2, \ldots, m,$$

and

$$x_j \geq 0, \qquad \text{for } j = 1, 2, \ldots, n,$$
$$x_j \text{ is integer}, \qquad \text{for } j = 1, 2, \ldots, I;\ I \leq n.$$

(When $I = n$, this problem becomes the pure IP problem.)

One change involves the choice of the branching variable. Before, the next variable in the natural ordering — $x_1, x_2, \ldots, xn$ was chosen automatically. Now, the only variables considered are the integer-restricted variables that have a non-integer value in the optimal solution for the LP relaxation of the current subproblem. Our rule for choosing among these variables is to select the first one in the natural ordering. (Production codes generally use a more sophisticated rule.)

The second change involves the values assigned to the branching variable for creating the new smaller subproblems. Before, the binary variable was fixed at 0 and 1, respectively, for the two new subproblems. Now, the general integer-restricted variable could have a very large number of possible integer values, and it would be inefficient to create and analyze many subproblems by fixing the variable at its individual integer values. Therefore, what is done instead is to create just two new subproblems (as before) by specifying two ranges of values for the variable.

To spell out how this is done, let x_j be the current branching variable, and let x_j^* be its (non-integer) value in the optimal solution for the LP relaxation of the current sub problem. Using square brackets to denote

$$[x_j^*] = \text{greatest integer} \leq x_j^*,$$

we have for the range of values for the two new subproblems

$$x_j \leq [x_j^*] \qquad \text{and} \qquad x_j \geq [x_j^*] + 1,$$

respectively. Each inequality becomes an *additional constraint* for that new subproblem. For example, if $x_j^* = 3\frac{1}{2}$, then

$$x_j \leq 3 \qquad \text{and} \qquad x_j \geq 4$$

The third change involves the bounding step. Before, with a pure IP problem and integer coefficients in the objective function, the value of Z for the optimal solution for the subproblem's LP relaxation was rounded down to obtain the bound, because any feasible solution for the subproblem must have an integer Z. Now, with some of the variables not integer-restricted, the bound is the value of Z without rounding down.

The fourth (and final) change to the BIP algorithm to obtain our MIP algorithm in volves fathoming test 3. Before, with a pure IP problem, the test was that the optimal solution for the subproblem's LP relaxation is integer, since this ensures that the solution is feasible, and therefore optimal, for the subproblem. Now, with a mixed IP problem, the test requires only that the integer-restricted variables be integer in the optimal solution for the subproblem's LP relaxation, because this suffices to ensure that the solution is feasible, and therefore optimal, for the subproblem.

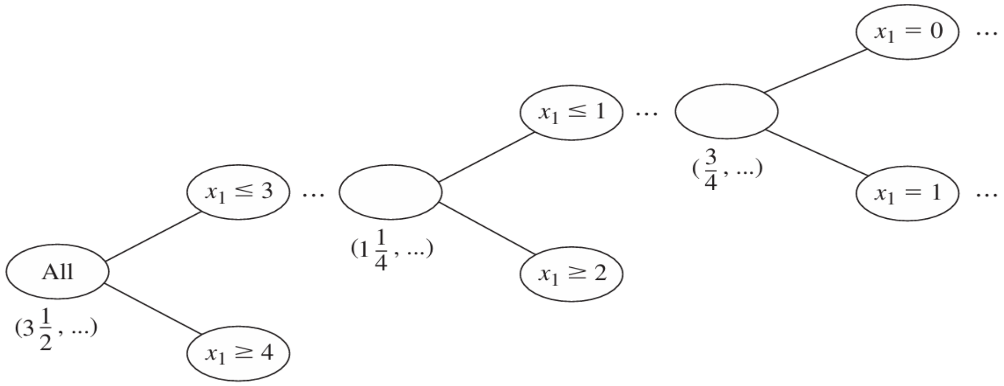


Figure 3.2.1: Branching in MIP

4. Solving IP with LINDO/LINGO

A variable (say, w) that can assume any non-negative integer value is indicated by the GIN statement. Thus, if an integer variable w can assume the values 0, 1, 2,..., we would type the following statement after the END statement:

```
GIN w
```

To tell LINDO that the first n variables appearing in the formulation may assume any non-negative integer value, use the command

```
GIN n.
```

To indicate that a variable must equal a non-negative integer in LINGO, use the @GIN operator.

Let us solve the problem described in section 2.1 using LINDO

LINDO Code

```
MAX 4 X1 - 2 X2 + 7 X3 - X4

ST
```

```
X1 + 5 X3 <= 10
X1 + X2 - X3 <= 1
6 X1 - 5 X2 <= 0
- X1 + 2 X3 - 2 X4 <= 3
END
GIN X1
GIN X2
GIN X3
```

5. Formulating & Solving MIP

The Research and Development Division of the GOOD PRODUCTS COMPANY has developed three possible new products. However, to avoid undue diversification of the company's product line, management has imposed the following restriction.

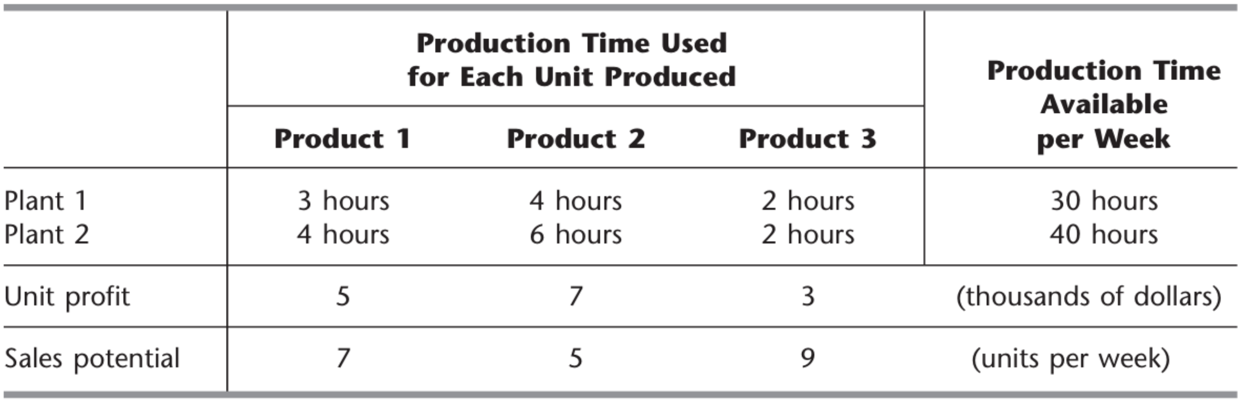
Restriction 1: From the three possible new products, at most two should be chosen to be produced. Each of these products can be produced in either of two plants.

For administrative reasons, management has imposed a second restriction in this regard.

Restriction 2: Just one of the two plants should be chosen to be the sole producer of the new products.

The production cost per unit of each product would be essentially the same in the two plants. However, because of differences in their production facilities, the number of hours of pro duction time needed per unit of each product might differ between the two plants. These data are given in table 5.1, along with other relevant information, including marketing estimates of the number of units of each product that could be sold per week if it is produced. The objective is to choose the products, the plant, and the production rates of the chosen products so as to maximize total profit.

Table 5.1: Production data



	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	(thousands of dollars)
Sales potential	7	5	9	(units per week)

Solution

If we let x_1, x_2, x_3 be the production rates of the respective products, the model then becomes,

$$\text{Maximize} \qquad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\le 30 \\ 4x_1 + 6x_2 + 2x_3 &\le 40 \\ x_1 \qquad\qquad &\le 7 \\ x_2 \qquad &\le 5 \\ x_3 &\le 9 \end{aligned}$$

and

$$x_1 \ge 0, \qquad x_2 \ge 0, \qquad x_3 \ge 0.$$

To deal with requirement 1, we introduce three auxiliary binary variables (y_1, y_2, y_3) with the interpretation

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \text{ can hold (can produce product } j) \\ 0 & \text{if } x_j = 0 \text{ must hold (cannot produce product } j), \end{cases}$$

for $j = 1,2,3$. To enforce this interpretation in the model with the help of M (an extremely large positive number, assume M=1000), we add the constraints

$$\begin{aligned} &x_1 \le My_1 \\ &x_2 \le My_2 \\ &x_3 \le My_3 \\ &y_1 + y_2 + y_3 \le 2 \\ &y_j \text{ is binary,} \qquad \text{for } j = 1, 2, 3. \end{aligned}$$

The either-or constraint and nonnegativity constraints give a *bounded* feasible region for the decision variables (so each $x_j \le M$ throughout this region). Therefore, in each $x_j \le My_j$ constraint, $y_j = 1$ allows any value of x_j in the feasible region, whereas $y_j = 0$ forces $x_j = 0$. (Conversely, $x_j > 0$ forces $y_j = 1$, whereas $x_j = 0$ allows either value of y_j.)

To deal with requirement 2, we introduce another auxiliary binary variable y4with the interpretation

$$y_4 = \begin{cases} 1 & \text{if } 4x_1 + 6x_2 + 2x_3 \le 40 \text{ must hold (choose Plant 2)} \\ 0 & \text{if } 3x_1 + 4x_2 + 2x_3 \le 30 \text{ must hold (choose Plant 1).} \end{cases}$$

this interpretation is enforced by adding the constraints,

$$\begin{aligned}
&3x_1 + 4x_2 + 2x_3 \le 30 + My_4 \\
&4x_1 + 6x_2 + 2x_3 \le 40 + M(1 - y_4) \\
&y_4 \text{ is binary.}
\end{aligned}$$

The final MIP Formulation is given below

$$\text{Maximize} \qquad Z = 5x_1 + 7x_2 + 3x_3,$$

subject to

$$\begin{aligned}
x_1 &\le 7 \\
x_2 &\le 5 \\
x_3 &\le 9 \\
x_1 - My_1 &\le 0 \\
x_2 - My_2 &\le 0 \\
x_3 - My_3 &\le 0 \\
y_1 + y_2 + y_3 &\le 2 \\
3x_1 + 4x_2 + 2x_3 - My_4 &\le 30 \\
4x_1 + 6x_2 + 2x_3 + My_4 &\le 40 + M
\end{aligned}$$

and

$$\begin{aligned}
&x_1 \ge 0, \qquad x_2 \ge 0, \qquad x_3 \ge 0 \\
&y_j \text{ is binary,} \qquad \text{for } j = 1, 2, 3, 4.
\end{aligned}$$

LINDO Code

```
MAX 5 X1 + 7 X2 + 3 X3

SUBJECT TO

M=1000

X1 <= 7

X2 <= 5

X3 <= 9

X1 - 1000Y1 <= 0

X2 - 1000Y2 <= 0

X3 - 1000Y3 <= 0

Y1 + Y2 + Y3 <= 2

3 X1 + 4 X2 + 2 X3 - 1000Y4 <= 30

4 X1 + 6 X2 + 2 X3 + 1000Y4 <= 1040
```

```
END

INT Y1

INT Y2

INT Y3

INT Y4
```

Exercise

The Toys-R-4-U Company has developed two new toys for possible inclusion in its product line for the upcoming Christmas season. Once production is started, the toys would generate a unit profit of $10 for toy 1 and $15 for toy 2. The company has two factories that are capable of producing these toys. However, to avoid doubling the start-up costs, just one factory would be used, where the choice would be based on maximizing weekly profit. For administrative reasons, the same factory would be used for both new toys if both are produced. Toy 1 can be produced at the rate of 50 per hour in factory 1 and 40 per hour in factory 2. Toy 2 can be produced at the rate of 40 per hour in factory 1 and 25 per hour in factory 2. Factories 1 and 2, respectively, have 50 hours and 70 hours of production time available per week. Determine the optimal production rate to maximize weekly profit.

Experiment-10:

Operational Research Lab Project: Application of Optimization Techniques Using LINDO, LINGO, and MATLAB.

Objective:

The objective of this project is to apply operational research (OR) techniques to solve real-life problems using software tools such as LINDO, LINGO, and MATLAB. Students will develop practical skills in modeling, data collection, analysis, and optimization.

Project Description:

Students are required to select a real-world problem from any industry or service sector (e.g., manufacturing, supply chain, transportation, healthcare, retail, etc.). They must collect actual data from a reliable source and formulate the problem using appropriate OR techniques.

Requirements:

1. **Problem Selection:**

Choose a practical problem that can be solved using OR methods (e.g., linear programming, transportation model, assignment problem, inventory model, queuing theory, etc.).

2. **Data Collection:**

- Gather real data from industries, businesses, or credible online sources.
- Clearly mention the source of data.

3. **Model Formulation:**

- Define decision variables, objective function, and constraints.
- Convert the real-world problem into a mathematical model.

4. **Software Implementation:**

- Solve the model using at least one of the following tools:
    - LINDO
    - LINGO
    - MATLAB

5. **Analysis and Interpretation:**

- Interpret the results obtained from the software.

- Provide insights and recommendations based on the solution.

6. **Report Preparation:**
   The final report must include:
   - Introduction of the problem
   - Literature/background (if applicable)
   - Data collection method
   - Model formulation
   - Software output (with screenshots)
   - Result analysis
   - Conclusion and recommendations
7. **Presentation & Viva:**
   - Students will present their project findings.
   - Each group/member must be able to explain their model and results clearly.