LINEAR PROGRAMMING

(MTS5 B08)

V SEMESTER

CORE COURSE

B Sc MATHEMATICS

(2019 Admission onwards)

UNIVERSITY OF CALICUT

School of Distance Education,

Calicut University P.O.

Malappuram - 673 635, Kerala.

19566

UNIVERSITY OF CALICUT

School of Distance Education

Study Material

V Semester

Core Course (MTS5 B08)

B Sc Mathematics

LINEAR PROGRAMMING

Prepared by:

Dr. Nithya A.A,
Asst. Professor, Dept. of Mathematics,
Mercy College, Palakkad.

Padma R.S,
Guest Faculty, NSS College,
Nenmara, Palakkad.

Scrutinized by:

Mr. Rajan T.K,
Associate. Professor, Dept. of Mathematics,
Yuvakshetra Institute of Management Studies,
Ezhakad, Palakkad.

Contents

Introduction

Linear programming is a mathematical technique for finding optimal solutions to problems that can be expressed using linear equations and inequalities. If a real-world problem can be represented accurately by the mathematical equations of a linear program, the method will find the best solution to the problem. Of course, few complex real-world problems can be expressed perfectly in terms of a set of linear functions. Nevertheless, linear programs can provide reasonably realistic representations of many real-world problems.

Business and industry widely use linear programming for sche duling and planning production, transportation and routing, and various types of scheduling. Delivery services use linear programs to schedule and route shipments to minimize shipment time and cost. Financial institutions use linear programming to determine the mix of financial products they offer or to schedule payments transferring funds between institutions. Airlines use linear programs to schedule their flights on the basis of availability of air crafts and staff. Retailers use linear programs to determine how to order products from manufacturers and organize deliveries with their stores. Health care institutions use linear programming to ensure the supply according to the need.

Let us see some model problems of linear programming

Model Problem 1

A furniture dealer deals in only two items- tables and chairs. He has Rs 1,00,000 to invest and has storage space of at most 100 pieces. A table costs Rs 3000 and a chair Rs 550. He estimates that from the sale of one table, he can make a profit of Rs 200 and that from the sale of one chair a profit of Rs 50. He wants to know how many tables and chairs he should buy from the available money so as to maximise his total profit, assuming that he can sell all the items which he buys. Such type of problems which seek to maximise (or, minimise) profit (or, cost) form a general class of problems called optimisation problems. Thus, an optimisation problem may involve finding maximum profit, minimum cost, or minimum use of resources etc. A special but a very important class of optimisation problems is linear programming problem. The above stated optimisation problem is an example of linear programming problem. Linear programming problems are of much interest because of their wide applicability in all fields.

Model Problem 2

A manufacturer produces two products X and Y with two machines A and B. The time of producing each unit of X for machine A is 50 minutes, and for machine B is 30 minutes. The time of producing each unit of Y for machine A is 24 minutes and for machine B is 33 minutes. Machines A and B work respectively 40 and 35 hours a weak. If a weak starts with a stock of 30 units of X and 90 of Y then how to plan the production in order to end the weak with the maximum stock to meet the demand of 75 units of X and 95 units of Y.

Model Problem 3

Maria has an on-line shop where she sells hand made paintings and cards. She sells the painting for Rs 5000 and the card for

Rs 2000. It takes her 2 hours to complete one painting and 45 minutes to make a single card. She also has a day job and makes paintings and cards in her free time. She cannot spend more than 15 hours a week to make paintings and cards. Additionally, she should make not more than 10 paintings and cards per week. She makes a profit of Rs 2500 on painting and Rs 1500 on each card. How many paintings and cards should she make each week to maximize her profit.

We discuss the graphical method of solving linear programming problems in the first chapter. After that an effective method called The Simplex Algorithm is discussed for solving the canonical maximization and canonical minimization linear programming problems. The importance of Canonical form is the non negativity of the independent variables and inequality of the main constraints. The third chapter is devoted to discuss general linear programming problems where these conditions are violated. Such problems are normally classified as non-canonical forms. As one of the main features of LPP is its dual nature, in the fourth chapter we discuss duality theory and related problems. Also the book focus on matrix games, transportation and assignment problems to an extend in the next two chapters.

Chapter 1

Geometric Linear Programming

In this chapter we discuss entirely about formulation of linear models and to find the solution of these linear programming problems by graphical and/or geometrical methods.

1.1 Profit Maximization and Cost Minimization

Linear functions are involved in linear programming problem(L.P.P) which are to be maximized or minimized according to the situation. Frequently, these functions represent profit (in the case of maximization) and cost (in the case of minimization).

1.2 Formulation of LPP

* Identify decision variables (an unknown in an optimization problem)
* Write objective function

* Formulate constraints

Example 1.1.

An appliance company manufactures heaters and air conditioners. The production of one heater requires 2 hours in the parts division of the company and 1 hour in the assembly division of the company; the production of one air conditioner requires 1 hour in the parts division of the company and 2 hours in the assembly division of the company. The parts division is operated for at most 8 hours per day and the assembly division is operated for at most 10 hours per day. If the profit realized upon sale is 30 per heater and 50 per air conditioner, how many heaters and air conditioners should the company manufacture per day so as to maximize profits?

Solution:

$$x = \text{Number of heaters per day}$$
$$y = \text{Number of air conditioners per day}$$

The quantity to be maximized, namely profit, is then given by

$$P(x, y) = 30x + 50y$$

Each heater requires 2 hours in the parts division and each air conditioner requires 1 hour in the parts division. Hence, the total amount of time required from the parts division per day is $2x+y$ hours and also as the parts division is available for at most 8 hours per day, we have the constraint

$$2x + y \leq 8$$

Similar reasoning applied to the assembly division yields the constraint

$$x + 2y \leq 10$$

We can also form table for better understanding

	Heater(x)	Air conditioner(y)	Max(atmost)
parts division	2	1	8
Assemble	1	2	10

Finally, we include the implied constraints

$$x \geq 0, y \geq 0$$

since a negative number of heaters or air conditioners manufactured per day is not possible. Hence we have, Maximize

$$P(x, y) = 30x + 50y \text{ (objective function)}$$

subject to constraints,

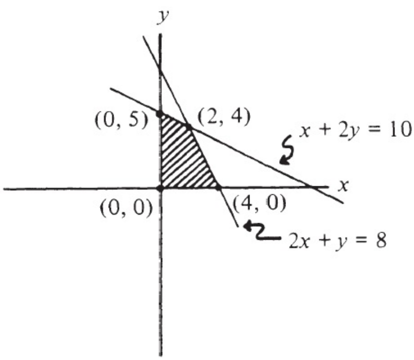
$$2x + y \leq 8,\ x + 2y \leq 10,\ x \geq 0,\ y \geq 0 \tag{1.1}$$

$$2x + y \leq 8$$

$$x + 2y \leq 10$$

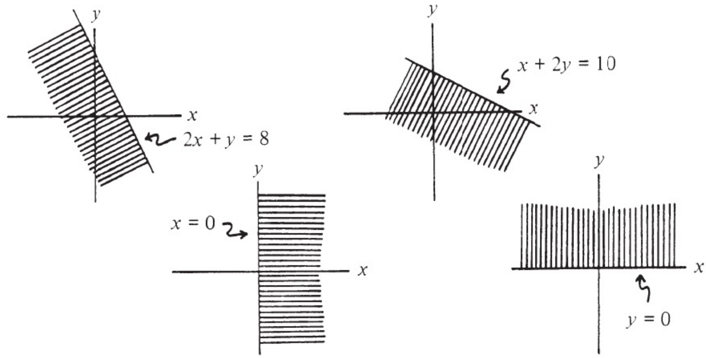
$$x \geq 0,\ y \geq 0$$

The set of points (x, y) (**Solution set**) satisfying all four constraints is the shaded region given below:



(2)

This region was obtained by graphing the equalities $2x + y = 8, x + 2y = 10, x = 0$ and $y = 0$, shading the solution set of the corresponding inequalities, and finding the mutual intersection of all such sets. In other words, the set of points common to the four shaded regions determined by the inequations 1.1



Example 1.2.

An oil company owns two refineries, say refinery A and refinery B. Refinery A is capable of producing 20 barrels of gasoline and 25 barrels of fuel oil per day; refinery B is capable of producing 40 barrels of gasoline and 20 barrels of fuel oil per day. The company requires at least 1000 barrels of gasoline and at least 800 barrels of fuel oil. If it costs 300 per day to operate refinery A and 500 per day to operate refinery B, how many days should each refinery be operated by the company so as to minimize costs?

Solution: If we put

$$x = \text{number of days for refinery A}$$
$$y = \text{number of days for refinery B}$$

then the quantity to be minimized, namely cost, is given by

$$C(x, y) = 300x + 500y.$$

Refinery A is capable of producing 20 barrels of gasoline per day and refinery B is capable of producing 40 barrels of gasoline per day. Hence, the total amount of gasoline produced is

$$20x + 40y$$

barrels. Since, at least 1000 barrels of gasoline is required by the company, we have the constraint

$$20x + 40y \geq 1000.$$

Similar reasoning applied to the fuel oil yields the constraint

$$25x + 20y \geq 800$$

	Refinery A(x)	Refinery B(y)	Min(atleast)
gasoline	20	40	1000
fuel oil	25	20	800

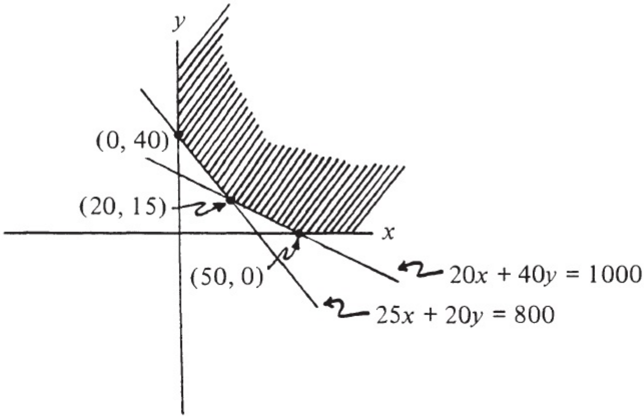
Again, we include the implied constraints $x \geq 0$ $y \geq 0$
the desired formulation is
Minimize

$$C(x, y) = 300x + 500y$$

subject to the constraints

$$20x + 40y \geq 1000, \ 25x + 20y \geq 800, \ x \geq 0, \ y \geq 0 \qquad (1.2)$$

The set of points (x, y) satisfying all four constraints in 1.2 is the shaded region given below.



(4)

Exercise 1.1. *Formulate the three model problems given in the Introduction as linear programming problems.*

1.3 Canonical Forms for LPP

Standard terminology of canonical maximization and minimization forms for Linear Programming Problems were discussed.

Definition 1.1.

Consider the canonical maximization linear programming problem
Maximize

$$f(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$$

subject to

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \leq b_n$$
$$x_1, x_2, ...x_n \geq 0$$

Definition 1.2.

Similarly, consider the canonical minimization linear programming problem
Minimize

$$g(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$$

subject to

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \geq b_m$$
$$x_1, x_2, ...x_n \geq 0$$

The first m constraints in each canonical form above are said to be **main constraints**; the second n constraints $x_1, x_2, ...x_n \geq 0$ in each canonical form above are said to be **non-negativity constraints**.

Definition 1.3.

The linear functions f and g in the above Definition 1.1 and 1.2 are said to be **objective functions**.

Definition 1.4.

The set of all points $(x_1, x_2, ...x_n)$ satisfying the $m+n$ constraints of a canonical maximization or a canonical minimization linear programming problem is said to be the **constraint set** of the problem. Any element of the constraint set is said to be a **feasible point** or **feasible solution**.

Definition 1.5.

Any feasible solution of a canonical maximization (respectively minimization) linear programming problem which maximizes (respectively minimizes) the objective function is said to be an **optimal solution**.

1.4 Polyhedral Convex Sets

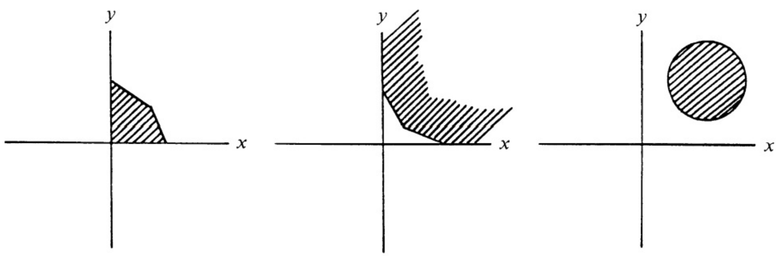
Definition 1.6.

Let $x = (x_1, x_2, ..., x_n), y = (y_1, y_2, ..., y_n) \in R^n$.
Then $tx + (1-t)y,\ 0 \le t \le 1$, is said to be the line segment between x and y

Definition 1.7.

Let S be a subset of R^n. S is said to be convex if, whenever $x = (x_l, x_2, ..., x_n), y = (y_l, y_2, ..., y_n) \in S$, then the line segment $tx + (1-t)y \in S, 0 \le t \le 1$.
i.e. A subset of S of R^n is convex if the line segment connecting any two points in S also lies entirely within S.

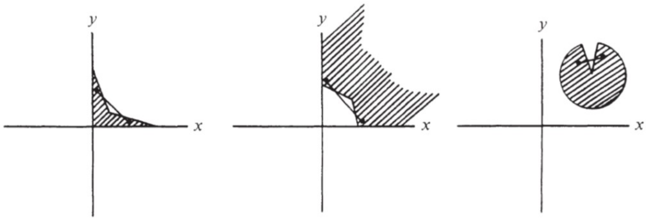
EXAMPLE 10.



Convex sets in $\mathbf{R}^2$.

Definition 1.8.

A subset of R^n is *non-convex set* if there exist two points in the set such that the line segment connecting them does not lie entirely within the subset. For example the circle $|z| = 1$ in R^2 is not a convex set as chord joining any two points of the set does not lie on the circle.



Nonconvex sets in $\mathbf{R}^2$.

Definition 1.9.

The set of points $(x_1, x_2, ..., x_n) \in R^n$ satisfying an equation of the form

$$a_1x_1 + a_2x_2 + ... + a_nx_n = b$$

is said to be a hyperplane of R^n. The set of points $(x_1, x_2, ..., x_n) \in R^n$ satisfying an inequality of the form

$$a_1x_1 + a_2x_2 + ... + a_nx_n \leq b$$

or

$$a_1x_1 + a_2x_2 + ... + a_nx_n \geq b$$

is said to be a closed half-space of R^n.

Example 1.3.

(i) A hyperplane in R^1 is the set of points x_1 satisfying an equation of the form $a_1x_1 = b$. If a_1 is non-zero, this set is simply the point b/a_1 in R^1 and the closed half-space $a_1x_1 \leq b$ or $a_1x_1 \geq b$ is closed ray in R^1.

(ii) A hyperplane in R^2 is the set of points (x_1, x_2) satisfying an equation of the form $a_1x_1 + a_2x_2 =$ b. If one of a_1 or a_2 is nonzero, this set is a line in R^2 and the closed half-space $a_1x_1 + a_2x_2 \leq b$ or $a_1x_1 + a_2x_2 \geq b$ is a closed half-plane in R^2.

Remarks 1.1.

The importance of closed half-spaces for linear programming is since each constraint of a canonical maximization or a canonical minimization linear programming problem describes a closed half-space and since the constraint set of the problem is the intersection of the solution sets of its constraints, we have that the constraint set of a canonical maximization or a canonical minimization linear programming problem is an intersection of closed half-spaces.

Theorem 1.1. *The constraint set of a canonical maximization or a canonical minimization linear programming problem is convex. Such a set is said to be a polyhedral convex set.*

Proof. Each constraint of a canonical maximization (minimization) problem is a convex set. Since intersection of convex set is convex,The constraint set of a canonical maximization or a canonical minimization linear programming problem is convex. □

Example 1.4.

The graph of constraint set of Example 1.1 and 1.2 are examples of polyhedral convex set.

Problem 1.1.

Find a convex set but not a polyhedral convex set?
Ans: Upper half plane in R^2 is convex but not a polyhedral convex set because it does not satisfy non-negativity constraint.

Definition 1.10.

Let x $= (x_1, x_2, \ldots, x_n) \in R^n$. The norm of x, denoted $\| x \|$, is

$$\| x \| = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$$

Note that the norm of a point in R^n is the usual Euclidean distance of that point from the origin.

Definition 1.11.

Let $r \geq 0$. The set of points x $= (x_1, x_2, \ldots, x_n) \in R^n$ such that $\| x \| \leq r$ is said to be the closed ball of radius r centered at the origin.

Example 1.5.

(i) The closed ball of radius 0 centered at the origin in R^n is simply the origin of R^n

(ii) The closed ball of radius $r > 0$ centered at the origin in R^1 is a line segment including the end points.

(iii) The closed ball of radius $r > 0$ centered at the origin in R^2 is a circle and its interior.

(iv) The closed ball of radius $r > 0$ centered at the origin in R^3 is a sphere and its interior.

Example 1.6.

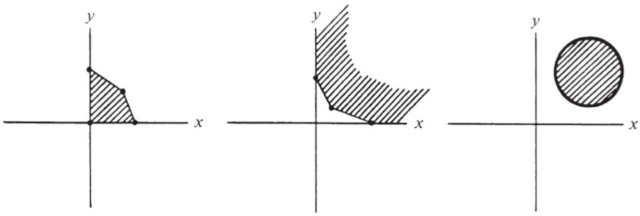
A subset S of $R^n$is said to be bounded if there exists $r \geq 0$ such that every element of S is contained in the closed ball of radius r centred at the origin. A subset S of R^n is said to be unbounded if it is not bounded.

Example 1.7.

The constraint set of Example 1.1 is a bounded polyhedral convex subset in R^2 because there exists a $r \geq 0$ (r > 5) such that every element of the constraint set is contained in the closed ball of radius r centered at the origin.
The constraint set of Example 1.2 is an unbounded polyhedral convex subset in R^2 because there does not exists a $r \geq 0$ such that every element of the constraint set is contained in the closed ball of radius r centered at the origin.

Example 1.8.

Let S be a convex set in R^n, $e \in S$ is said to be an extreme point of S if there do not exist $x, y \in S$ and t with $0 < t < 1$ such that $e = tx + (1 - t)y$.



Extreme points of convex sets in $\mathbf{R}^2$ (in bold).

Exercise 1.2. *Find the feasible region of the model linear programming problems given in the Introduction.*

Remarks 1.2.

A point $e \in S$ is an extreme point of S if no line segment within S contains e except at an endpoint. Extreme points in linear programming correspond to vertex or corner of polyhedral convex sets. However,extreme points of the closed unit disk in R^2 is the unit circle.

Theorem 1.2. *If the constraint set S of a canonical maximization or a canonical minimization linear programming problem is bounded, then the maximum or minimum value of the objective function is attained at an extreme point of S.*

Theorem 1.3. *(i) If the constraint set S of a canonical maximization linear programming problem is unbounded and there exists $M \in R$ such that the objective function f satisfies $f(x_1, x_2, \ldots, x_n) \leq M$ for all $(x_1, x_2, \ldots, x_n) \in S$, i.e.,$f$ is bounded above (by M), then the maximum value of the objective function is attained at an extreme point of S.*
(ii) If the constraint set S of a canonical minimization linear programming problem is unbounded and there exists $M \in R$ such that the objective function g satisfies $g((x_1, x_2, \ldots, x_n)) \geq M$ for all$(x_1, x_2, \ldots, x_n) \in S$, i.e., g is bounded below
(by M), then the minimum value of the objective function is attained at an extreme point of S.

Remarks 1.3.

Theorem 1.2 and Theorem 1.3 only assert the existence of optimal solutions at extreme points of the constraint set; there may be optimal solutions at points of the constraint set other than extreme points.
Also, any point on the line segment connecting two distinct optimal solutions of a canonical linear programming problem is an optimal solution and that any canonical linear programming problem has either zero, one, or infinitely many optimal solutions.

1.4.1 The Two Examples Revisited

Example 1.9.

Maximize

$$P(x, y) = 30x + 50y$$

subject to constraints

$$2x + y \leq 8$$

$$x + 2y \leq 10$$

$$x, y \geq 0$$

Solution:
Since the constraint set is bounded, the maximum value of P(x, y) is attained at an extreme point (Theorem 1.2) namely at $(0,0), (4,0), (0,5)$, or $(2,4)$. Hence we need to evaluate $P(x,y) = 30x + 50y$ at each of these points.

extreme points	value
(0,0)	0
(4,0)	120
(0,5)	250
(2,4)	260

The maximum is attained at the point (2,4). Hence the appliance company should manufacture 2 heaters and 4 air conditioners per day so as to obtain a maximum profit of 260 per day.

Example 1.10.

Minimize

$$C(x,y) = 300x + 500y$$

subject to constraints

$$20x + 40y \geq 1000$$

$$25x + 20y \geq 800$$

$$x, y \geq 0$$

Solution:
Since the constraint set is unbounded, we must show that C is

bounded below, i.e., we must find $M \in R$ such that $C(x, y) = 300x + 500y \geq M$
whenever (x, y) is a point of the constrained set. Now M=0 satisfies the condition. Then Theorem 1.3(ii) implies that the minimum value of $C(x, y)$ is attained at an extreme points, namely at $(50, 0), (0, 40)$, or $(20, 15)$. Hence we need to evaluate C at each of these points and note which evaluation is minimum.

extreme points	value
(50,0)	15,000
(0,40)	20,000
(20,15)	13,500

The minimum is attained at the point (20,15). Hence the oil company should operate refinery A for 20 days and refinery B for 15 days so as to attain a minimum cost of 13500.

1.5 A Geometric Method for Linear Programming

For solving linear programming problems according to the knowledge of the extreme points of the constraint set of the problem may be difficult to obtain in some cases. For example, consider the linear programming problem with three variables and five constrains.Then we find extreme difficulty to find the extreme points of the constraint set of a linear programming problem.
Another method to find extreme point without actually graphing the constraint set.
Maximize

$$f(x, y, z) = 2x + y - 2z$$

subject to

$$x + y + z \leq 1$$

$$y + 4z \leq 2$$

$$x, y, z \geq 0$$

The bounding surfaces of the constraint set are the planes

$$x + y + z = 1, y + 4z = 2, x = 0, y = 0, \text{ and } z = 0$$

Taking these five equations three at a time and solving the resulting systems of linear equations, we obtain

$$\begin{bmatrix} x + y + z = 1 \\ y + 4z = 2 \\ x = 0 \end{bmatrix} \tag{1.3}$$

solving we get $(x, y, z) = (0, 2/3, 1/3)$

$$\begin{bmatrix} x + y + z = 1 \\ y + 4z = 2 \\ y = 0 \end{bmatrix} \tag{1.4}$$

solving we get $(x, y, z) = (1/2, 0, 1/2)$

$$\begin{bmatrix} x + y + z = 1 \\ y + 4z = 2 \\ z = 0 \end{bmatrix} \tag{1.5}$$

solving we get $(x, y, z) = (-1, 2, 0)$

$$\begin{bmatrix} x + y + z = 1 \\ x = 0 \\ y = 0 \end{bmatrix} \tag{1.6}$$

solving we get $(x, y, z) = (0, 0, 1)$

$$\begin{bmatrix} x + y + z = 1 \\ x = 0 \\ z = 0 \end{bmatrix} \tag{1.7}$$

solving we get $(x, y, z) = (0, 1, 0)$

$$\begin{bmatrix} x + y + z = 1 \\ y = 0 \\ z = 0 \end{bmatrix} \tag{1.8}$$

solving we get $(x, y, z) = (1, 0, 0)$

$$\begin{bmatrix} y + 4z = 2 \\ x = 0 \\ y = 0 \end{bmatrix} \tag{1.9}$$

solving we get $(x, y, z) = (0, 0, 1/2)$

$$\begin{bmatrix} y + 4z = 2 \\ x = 0 \\ z = 0 \end{bmatrix} \tag{1.10}$$

solving we get $(x, y, z) = (0, 2, 0)$

$$\begin{bmatrix} y + 4z = 2 \\ y = 0 \\ z = 0 \end{bmatrix} \tag{1.11}$$

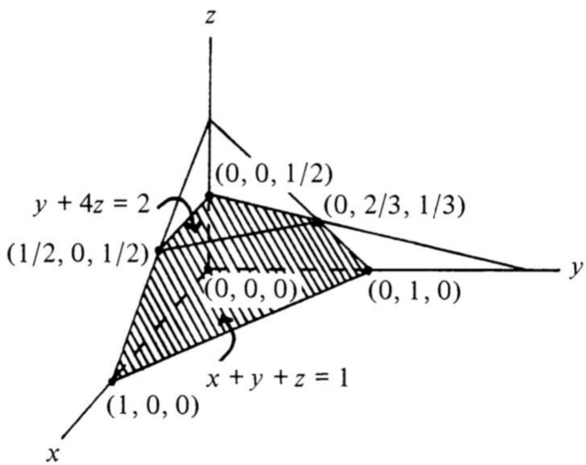
which implies INCONSISTENT

$$\begin{bmatrix} x = 0 \\ y = 0 \\ z = 0 \end{bmatrix} \tag{1.12}$$

solving we get $(x, y, z) = (0, 0, 0)$
Some of these points may violate one or more of the original constraints. We tabulate each candidate point along with the constraints violated (if any):

Candidate point	Constraint(s) of (5) violated
$(0, 2/3, 1/3)$	None
$(1/2, 0, 1/2)$	None
(-1,2,0)	x $\geq$ 0
(0,0,1)	y + 4z $\leq$ 2
(0,1,0)	None
(1,0,0)	None
$(0, 0, 1/2)$	None
(0,2,0)	x+ y+z $\leq$ 1
(0,0,0)	None

The extreme points of the constraint set are precisely those points above that violated none of the original constraints.
We can verify this by actually graphing the constraint set :



Remarks 1.4.

An upper bound for the possible number of extreme point in this example is given by $5C_3 = \frac{5!}{3!2!} = 10$ since there are five constraints (considered as equations) and since two equations in three unknowns are needed to uniquely determine an ordered

triple. Of these 10 candidates, only 6 were actually extreme points of the desired constraint set.
In general $(m+n)C_n = \dfrac{(m+n)!}{m!n!}$

Problem 1.2.

Find the appropriate regions in R^2 constrained to the first quadrant of the Cartesian plane.

1. a bounded polyhedral convex subset. The set of all points (x,y) in the first quadrant which satisfies inequality $x+y \leq 1$.

2. an unbounded polyhedral convex subset. The set of all points (x,y) in the first quadrant which satisfies inequality $x+y \geq 1$.

3. a bounded non-convex subset. Any two disjoint squares in the first quadrant along with their interior.

4. an unbounded non-convex subset. Set of all points such that $\{(x,0) : x > 0\}$ and $\{(0,y) : y > 0\}$

5. a convex subset having no extreme points,Interior of a circle.

Problem 1.3.

Convert each of the linear programming problems below to canonical form as in Definition 1.1.

1. Maximize
$$f(x,y) = x + y$$
subject to
$$x - y \leq 3$$

$$2x + y \leq 12$$

$$0 \leq x \leq 4$$

$$0 \leq y \leq 6$$

Solution:
Maximize

$$f(x, y) = x + y$$

subject to

$$x - y \leq 3$$

$$2x + y \leq 12$$

$$x \leq 4$$

$$y \leq 6$$

$$x, y \geq 0$$

2. Maximize

$$f(x, y) = -2y - x$$

Subject to

$$2x - y \geq -1$$

$$3y - x \leq 8$$

$$x, y \geq 0$$

Solution:
Maximize

$$f(x, y) = -2y - x$$

Subject to

$$-2x + y \leq 1$$

$$3y - x \leq 8$$

$$x, y \geq 0$$

Problem 1.4.

A drug company sells three different formulations of vitamin complex and mineral complex. The first formulation consists entirely of vitamin complex and sells for Rs.1 per unit. The second formulation consists of 3/4 of a unit of vitamin complex and 1/4 of a unit of mineral complex and sells for Rs 2 per unit. The third formulation consists of 1/2 of a unit of each of the complexes and sells for Rs 3 per unit. If the company has 100 units of vitamin complex and 75 units of mineral complex available, how many units of each formulation should the company produce so as to maximize sales revenue?
Solution:
Let x be the number of units of the first formulation, y be the number of units of the second formulation and z be the number of units of the third formulation and let R be the total sales revenue. (i.e)
Maximize

$$R = x + 2y + 3z$$

subject to constraint

$$x + (3/4)y + (1/2)z \leq 100$$

$$(1/4)y + (1/2)z \leq 75$$

$$x \geq 0, y \geq 0, z \geq 0$$

Here we have 10 possible extreme point candidates. Out of these points, (25,0,150) and (0,50,125) are extreme points which maximize R.
Max $R = 475$ at $(x, y, z) = (25, 0, 150)$ or at $(x, y, z) = (0, 50, 125)$

Problem 1.5.

Find the upper bound for the number of extreme point candidates
Maximize

$$f(x, y, z, w) = 2x - y + z - w$$

subject to

$$x + w \leq 1$$
$$x - y \leq 2$$
$$z - 2w \leq 3$$
$$x \geq 0, y \geq 0, z \geq 0, w \geq 0$$

Solution :
Upper bound for the number of extreme points of the desired constraint set is $(m + n)C_n = \frac{(m+n)!}{m!n!}$
Therefore here we have 3+4 constraint set which has 35 extreme points candidates.

Problem 1.6.

Label each of the following statements TRUE or FALSE

1. Any unbounded linear programming problem has an unbounded constraint set - True
2. Any linear programming problem having an unbounded constraint set is unbounded - False

Problem 1.7.

Show that the linear programming problem
Maximize

$$f(x, y) = 3x + 2y$$

subject to

$$2x - y \leq -1$$
$$x - 2y \leq 0$$
$$x \geq 0, y \geq 0$$

is infeasible.
Solution:
Since it is impossible to find a point which satisfies both the constraint. Therefore it has a empty constraint set. Hence infeasible.

Chapter 2

The Simplex Algorithm

In this chapter, we present the simplex algorithm, an effective method for solving the canonical maximization and canonical minimization linear programming problems.It is capable of finding optimal solutions without finding and testing large numbers of extreme point candidates.

2.1 Canonical Slack Forms for LPP, Tucker Table

Consider the canonical maximization linear programming problem
Maximize $f(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$
subject to

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &\leq b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &\leq b_n \\ x_1, x_2, ...x_n &\geq 0 \end{aligned} \tag{1}$$

Let $t_1, t_2, ...t_m \geq 0$ be such that

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + t_1 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + t_2 &= b_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n + t_m &= b_m
\end{aligned}$$

Then

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - b_1 &= -t_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n - b_2 &= -t_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n - b_m &= -t_m
\end{aligned}$$

and we can reformulate (1) as

Problem 2.1.

Maximize $f(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$ subject to

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - b_1 &= -t_1 \\
a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n - b_2 &= -t_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n - b_m &= -t_m \\
t_1, t_2, ...t_m \geq 0 &\text{ and} \\
x_1, x_2, ...x_n &\geq 0
\end{aligned}$$

Similarly, consider the canonical minimization linear programming problem
Minimize $g(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$ subject

to

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n \geq b_n$$
$$x_1, x_2, ...x_n \geq 0$$

(2)

Let $t_1, t_2, ...t_m \geq 0$ be such that

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 + t_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 + t_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m + t_m$$

Then

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - b_1 = t_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n - b_2 = t_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n - b_m = t_m$$

and we can reformulate (2) as

Problem 2.2.

Minimize $g(x_1, x_2, ...x_n) = c_1x_1 + c_2x_2 + ... + c_nx_n - d$ subject to

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - b_1 = t_1$$
$$a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n - b_2 = t_2$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n - b_m = t_m$$
$$t_1, t_2, ...t_m \geq 0 \text{ and}$$
$$x_1, x_2, ...x_n \geq 0$$

Definition 2.1.

The linear programming problems 2.1 and 2.2 above are said to be **canonical slack maximization** and **canonical slack minimization** linear programming problems respectively. The variables $t_1, t_2, ...t_m$ are said to be **slack variables**.

Definition 2.2.

The tables

x_1	x_2	$\ldots$	x_n	-1	
a_{11}	a_{12}	$\ldots$	a_{1n}	b_1	$= -t_1$
a_{21}	a_{22}	$\ldots$	a_{2n}	b_2	$= -t_2$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m	$= -t_m$
c_1	c_2	$\ldots$	c_n	d	$= f$

and

x_1	a_{11}	$a_{21} \ldots$	a_{m1}	c_1
x_2	a_{12}	$a_{22} \ldots$	a_{m2}	c_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_n	a_{1n}	$a_{2n} \ldots$	a_{mn}	c_n
-1	b_1	$\ldots$	b_m	d
$= t_1$	$= t_2$	$\ldots$	$= t_m$	$= g$

are said to be **Tucker tableaus** or simply **tableaus** of the canonical slack maximization and the canonical slack minimization linear programming problems respectively. The variables to the north of the maximum tableau and to the west of the minimum tableau are said to be **independent variables** or **nonbasic variables**. The variables to the east of the maximum tableau and to the south of the minimum tableau are said to be *dependent variables* or *basic variables*. In this book, we will use the terms independent and dependent rather than non basic and basic respectively.

Note how the main constraints of each of the canonical slack linear programming problems are recorded in the corresponding Tucker tableaus. The coefficients of the main constraints of 2.1 appear as rows of the maximum tableau. The i^{th} main constraint may be reconstructed from the i^{th} row of coefficients by multiplying each coefficient by its corresponding independent variable (or multiplying by - 1 in the case of b_i, adding all such products, and setting the result equal to the corresponding dependent variable. The objective function of the problem is recorded by the last row of the maximum tableau in the same manner. Similarly, the main constraints and objective function of 2.1 are recorded in the minimum tableau above as columns instead of rows. (There is a reason for this difference which will be made clear in Chapter 3.) The reason for the choice of minus sign in front of the constant d in the problems 2.1 and 2.2 above is now seen to be one of convenience. Note that non-negativity constraints of a canonical slack maximization or a canonical slack minimization linear programming problem are not recorded in the corresponding Tucker tableaus. The non-negativity of all variables in a linear programming problem is assumed in the remainder of this chapter.

2.2 The Simplex Algorithm

Definition 2.3.

Any solution obtained by setting all of the independent variables of a tableau equal to zero is said to be a **basic solution**. For example consider the following table

x_1	x_2	-1	
1	2	20	$= -t_A$
2	2	30	$= -t_B$
2	1	25	$= -t_C$
200	150	0	$= P$

The basic solutions of this table are as follows $x_1 = 0, x_2 = 0, t_A = 20, t_B = 30, t_C = 25, P = 0$

Definition 2.4.

A basic solution is called a **basic feasible solution** if the values of all the independent and dependent variables of the table are greater than or equal to zero.

In the above example the obtained solution is also a basic feasible solution since the values of both independent and dependent variables are greater than or equal to zero.The following example shows that basic solution always need not be basic feasible solution.

x_1	-0.5	1.5	7.5
x_2	-1	1	5
t_P	0.5	0.5	12.5
-1	-100	50	-2500
	$= x_3$	$= t_F$	$= C$

The basic solutions of the above table are
$x_1 = x_2 = t_p = 0, x_3 = 100, t_F = -50, C = 2500$
Note that this solution is not a basic feasible solution since $t_F < 0$.

Definition 2.5.

Let

(	*indep*	*var's*	)	-1	
a_{11}	a_{12}	$\dots$	a_{1n}	b_1	
a_{21}	a_{22}	$\dots$	a_{2n}	b_2	*dep*
$\vdots$	$\vdots$		$\vdots$	$\vdots$	*var's*
a_{m1}	a_{m2}	$\dots$	a_{mn}	b_m	
c_1	c_2	$\dots$	c_n	d	$= f$

be a tableau of a canonical slack maximization linear programming problem. The tableau is said to be maximum basic feasible if $b_1, b_2, ...b_m \geq 0$
.

In a maximum basic feasible tableau, the basic solution is a feasible solution. Indeed, upon setting all of the independent variables equal to zero, all of the main constraints reduce to the form $-b_i = -$ (dep.var)[1], i.e. $b_i =$ (dep.var). Since all $b_i \geq 0$ for all i we have (dep.var) ≥ 0 for all dependent variables; such a solution satisfies all of the constraints of the original problem and is thus feasible.A similar argument shows the converse, namely that a maximum tableau whose basic solution is feasible must be a maximum basic feasible tableau.Since we are considering a basic feasible solution the values of all independent and dependent variables are greater than zero and hence value of $b_i \geq 0$ for all i.

Definition 2.6.

The pivot transformation is the operation by which we transform a tableau into a new tableau having exactly the same feasible solutions as the original.

[1]Dependent Variable

2.2.1 Simplex Algorithm for Maximum Basic Feasible Tableaus

Step 1: Consider the initial simplex table of our canonical slack maximization linear programming problem.

x_1	x_2	$\ldots$	x_n	-1	
a_{11}	a_{12}	$\ldots$	a_{1n}	b_1	$= -t_1$
a_{21}	a_{22}	$\ldots$	a_{2n}	b_2	$= -t_2$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m	$= -t_m$
c_1	c_2	$\ldots$	c_n	d	$= f$

Step 2: If the current tableau is maximum basic feasible,then proceed. If $c_1, c_2, ..., c_n \leq 0$ then STOP; the basic solution of the current table is optimal.Otherwise continue.

Step 3: Choose $c_j > 0$.(Refer anticycling rules given below for the most appropriate choice).

Step 4: If $a_{1j}, a_{2j}, ...a_{mj} \leq 0$ then STOP; the maximization problem is unbounded.

Step 5: Compute $\min_{1 \leq i \leq m} \{\frac{b_i}{a_{ij}} > 0\} = \frac{b_p}{a_{pj}}$. Pivot on a_{pj} and let that value be α. i.e let $a_{pj} = \alpha$

THE PIVOT TRANSFORMATION is as follows

Step 6: Interchange the variables corresponding to α's row and column, leaving the signs behind.

Step 7: Now replace α by $1/\alpha$ and replace every entry q in the same row as α by q/α.

Step 8: Replace every entry r in the same column as α by $-r/\alpha$.

Step 9: Every entry s not in the same row and not in the same column as α determines a unique entry q in the same row as α and in the same column as s and a unique entry r in the same column as α and in the same row as s. Replace s by $(\alpha s - qr)/\alpha$. Then you will obtain the new table.

Step 10: Again continue from step 2 until the process stops.

Remarks 2.1.

There may be more than one value of p for which b_p/a_{pj} is minimum. In that case proceed as follows. You can also use the following rules to choose your c_j which was mentioned in step 2.

2.2.2 Anticycling Rules

List all variables, both independent and dependent, appearing in the initial tableau. (The ordering of the variables in the list is not important as long as the rules below are implemented in a manner consistent with this list.) Any pivot entry is determined uniquely by a pivot row and a pivot column. The rules below determine this row and column.

Rule A (Determination of pivot row). Whenever there is more than one possible choice of pivot row in accordance with the simplex algorithm, choose the row corresponding to the variable that appears nearest the top (or front) of the list.

Rule B (Determination of pivot column). Whenever there is more than one possible choice of pivot column in accordance with the simplex algorithm, choose the column corresponding to the variable that appears nearest the top (or front) of the list.

Remarks 2.2.

If a problem is asked to find a table, by pivoting on a particular number in the canonical tableau, then we have to just follow steps from 6 to 9 from the above algorithm.

2.3 An Example: Profit Maximization

Example 2.1.

An electrical firm manufactures circuit boards in two configurations, say configuration 1 and configuration 2. Each circuit board in configuration 1 requires 1 A component, 2 B components, and 2 C components; each circuit board in configuration 2 requires 2 A components, 2 B components, and 1 C component. The firm has 20 A components, 30 B components, and 25 C components available. If the profit realized upon sale is Rs.200 per circuit board in configuration 1 and Rs.150 per circuit board in configuration 2, how many circuit boards of each configuration should the electrical firm manufacture so as to maximize profits?

Solution:
Let x_1 = No.of circuit boards in configuration 1 and x_2 = No.of circuit boards in configuration 2. Then the mathematical formulation of the problem is
Maximize

$$P(x_1, x_2) = 200x_1 + 150x_2$$

Subject to

$$\begin{aligned} x_1 + 2x_2 &\leq 20 \\ 2x_1 + 2x_2 &\leq 30 \\ 2x_1 + x_2 &\leq 25 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Put t_A = slack variable for A components, t_B = slack variable for B components, t_c = slack variable for C components. Then the canonical slack form of the problem is Maximize

$$P(x_1, x_2) = 200x_1 + 150x_2$$

Subject to

$$\begin{aligned} x_1 + 2x_2 - 20 &= -t_A \\ 2x_1 + 2x_2 - 30 &= -t_B \\ 2x_1 + x_2 - 25 &= -t_C \\ x_1, x_2, t_A, t_B, t_C &\geq 0. \end{aligned}$$

This is recorded in a Tucker tableau as

x_1	x_2	-1	
1	2	20	$= -t_A$
2	2	30	$= -t_B$
2	1	25	$= -t_C$
200	150	0	$= P$

x_1 and x_2 are independent variables in this tableau while t_A, t_B and t_C are dependent variables.

Definition 2.7.

A canonical maximization (respectively minimization) linear programming problem is said to be unbounded if the constraint set is unbounded and the objective function is not bounded above (respectively not bounded below) on this constraint set.
Hence, an unbounded linear programming problem has no maximum (or minimum) since there are feasible solutions that make the objective function of the problem arbitrarily large (or small).
If we STOP in step (4), the current tableau is of the form

$$\begin{array}{|cccccc|}
\hline
 & & x & -1 & & \\
\hline
 & & \leq & \geq & & \\
 & & \leq & \geq & & \\
\vdots & \vdots & \vdots & \vdots & & = (dep.var's) \\
 & & \leq & \geq & & \\
\hline
 & & > 0 & d & & f \\
\hline
\end{array}$$

Now set all of the independent variables except x equal to zero. All of the main constraints reduce to the form
$(\leq 0)x - (\geq 0) = -($ dep.var $)$
implies $(\leq 0)x + (\leq 0) = -($ dep.var $)$
Hence if x is non-negative, we have $(\leq 0) = -($ dep.var $)$ implies $($ dep.var $) \geq 0$
such a solution satisfies all of the constraints of the original problem and is thus feasible. But, as $x \to \infty$ we have $f = (> 0)x - d \to \infty$
i.e., we can make f arbitrarily large by increasing x.
The tableau form that terminates the simplex algorithm with an unbounded linear programming problem.

Example 2.2.

Consider the initial simplex table of a canonical maximization problem

x_1	x_2	-1	
2	1	8	$= -t_1$
1	2	10	$= -t_2$
30	50	0	$= f$

(1) The initial tableau is clearly maximum basic feasible.
(2) We proceed to choose c_j as per anticycling rules. So we choose $c_1 = 30$.
(3) Since here a_{11} and a_{21} are both positive we compute $\min\{b_1/a_{11} = 8/2, b_2/a_{21} = 10/1\} = b_1/a_{11}$. So pivot on a_{11} i.e. $\alpha = 2$

x_1	x_2	-1	
2^*	1	8	$= -t_1$
1	2	10	$= -t_2$
30	50	0	$= f$

Then by applying the remaining steps of simplex algorithm we obtain our new table as

t_1	x_2	-1	
$1/2$	$1/2$	4	$= -x_1$
$-1/2$	$3/2$	6	$= -t_2$
-15	35	-120	$= f$

(1) The current tableau is clearly maximum basic feasible.
(2) We proceed to choose c_j. So we choose $c_2 = 35$.
(3) Since here a_{12} and a_{22} are both positive we compute $Min\{b_1/a_{12} = 8, b_2/a_{22} = 4\} = b_2/a_{22}$. So pivot on a_{22} i.e. $\alpha = 3/2$

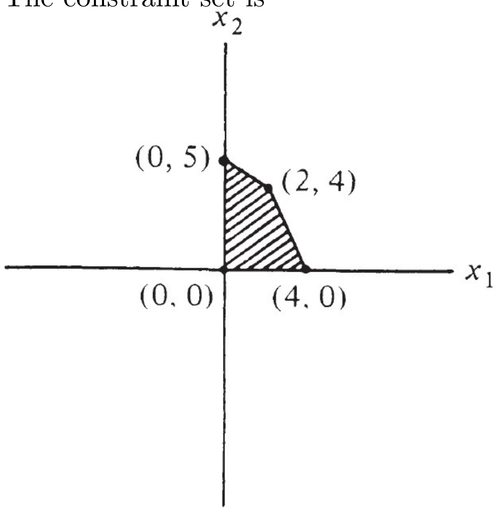
t_1	x_2	-1	
$1/2$	$1/2$	4	$= -x_1$
$-1/2$	$(3/2)^*$	6	$= -t_2$
-15	35	-120	$= f$

Then by applying the remaining steps of simplex algorithm we obtain our new table as

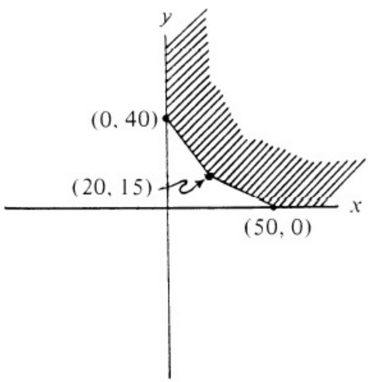
t_1	t_2	-1	
$2/3$	$-1/3$	2	$= -x_1$
$-1/3$	$2/3$	4	$= -x_2$
$-10/3$	$-70/3$	-260	$= f$

(1) The current tableau is clearly maximum basic feasible.
(2) Since $c_1, c_2 < 0$,we stop the algorithm. The basic solution of the current maximum tableau is optimal. This optimal solution is $t_1 = t_2 = 0, x_1 = 2, x_2 = 4, Max f = 260$

2.3 is the simplex algorithm solution of Example 1.1 of Chapter 1 The constraint set is



It is interesting to note what the simplex algorithm is doing geometrically here. Consider the x_1 and x_2 values of the basic solutions of the three tableaus in Example 2.3 are (0,0),(4,0) and (2,4) respectively.We see that the basic solutions correspond to extreme points of the constraint set and that each pivot simulates a movement from one extreme point to an adjacent extreme point along a connecting edge.



Such facts are true in general for maximum basic feasible tableaus. Basic feasible solutions correspond to extreme points of constraint sets in the geometric sense. The simplex algorithm is designed so that the transition between basic feasible solutions of any two successive maximum tableaus (simulated by movement from one extreme point to another along a connecting edge) does not decrease the value of the objective function. Hence each tableau transition maintains or increases the value of the objective function. Usually, after a finite number of tableau transitions, a maximum value of the objective function is reached or the simplex algorithm detects unboundedness. In rare instances, the objective function may maintain the same value repeatedly without the simplex algorithm terminating with a maximum value or a detection of unboundedness.

Example 2.3.

Apply the simplex algorithm to the maximum tableau

x_1	x_2	-1	
-1	1	1	$= -t_1$
1	-1	3	$= -t_2$
1	2	0	$= f$

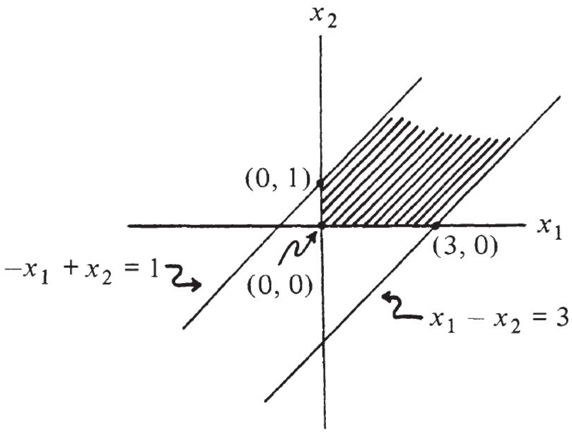
(1) The initial tableau is clearly maximum basic feasible.
(2) We proceed to choose c_j as per anticycling rules.So we choose $c_1 = 1$.
(3) Since here a_{21} is positive we compute $Min\{b_2/a_{21} = 3/1\} = b_2/a_{21}$.So pivot on a_{21} i.e. $\alpha = 1$

x_1	x_2	-1	
-1	1	1	$= -t_1$
1^*	-1	3	$= -t_2$
1	2	0	$= f$

Then by applying the remaining steps of simplex algorithm we obtain our new table as

t_2	x_2	-1	
1	0	4	$= -t_1$
1	-1	3	$= -x_1$
-1	3	-3	$= f$

(1) The current tableau is clearly maximum basic feasible.
(2) We proceed to choose c_j.We choose $c_2 = 3$
(3) Since $a_{12}, a_{21} \leq 0$ we STOP, the maximization problem is unbounded.
The constraint set of Example 2.3 is

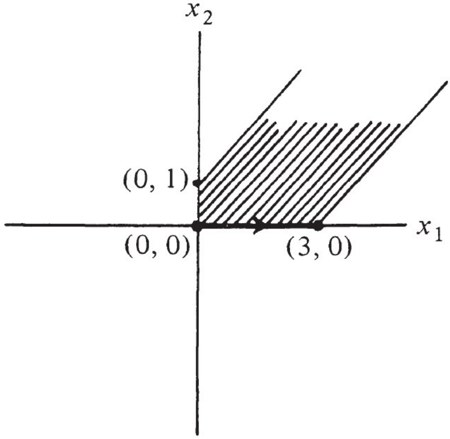


The objective function $f(x_1, x_2) = x_1 + 2x_2$ can be made arbitrarily large on this constraint set by considering feasible solutions farther and farther away from the origin.

Note that the unboundedness is detected at the point (3,0). The algorithm terminates because the objective function is not bounded above on the "infinite edge"of the line $x_1 - x_2 = 3$ (On the line $x_1 - x_2 = 3$,we have $f(x_1, x_2) = x_1 + 2x_2 = (x_2 + 3) + 2x_2 =$

$3x_2$

Now $f(x_1, x_2) \to \infty$ as $x_2 \to \infty$). The figure is given below



Remarks 2.3.

First of all, this algorithm preserves the maximum basic feasibility of a tableau, i.e., if an initial tableau is maximum basic feasible, then every subsequent tableau obtained via this algorithm will also be maximum basic feasible. This fact can be helpful in locating arithmetic mistakes in the b-columns of the tableaus. Secondly, when choosing the positive $c_j's$, it is advantageous to use some foresight and examine all positive c's and all a's above these c's. If you can find just one positive c corresponding to a column of nonpositive a's, then the linear programming problem is unbounded and the algorithm terminates. This is true even if there is some other positive c having positive a's in its column.

2.4 The Simplex Algorithm for Maximum Tableaus

So far, we have an algorithm for finding optimal solutions, if they exist, of canonical maximization linear programming problems having maximum basic feasible initial tableaus. What if the initial tableau of a canonical maximization problem is not maximum basic feasible?The answer to this question is given below.

The Simplex Algorithm for Maximum Tableaus
(1) The current tableau is of the form

$$\begin{array}{|cccc|c|c|}
\hline
(& \text{indep} & \text{var's} &) & -1 & \\
\hline
a_{11} & a_{12} & \dots & a_{1n} & b_1 & \\
a_{21} & a_{22} & \dots & a_{2n} & b_2 & \text{dep} \\
\vdots & \vdots & & \vdots & \vdots & \text{var's} \\
a_{m1} & a_{m2} & \dots & a_{mn} & b_m & \\
\hline
c_1 & c_2 & \dots & c_n & d & = f \\
\hline
\end{array}$$

(2) If $b_1, b_2, \ldots.., b_m \geq 0$ then continue to Simplex Algorithm for Maximum Basic Feasible Tableaus. Otherwise, continue.
(3) Choose $b_i < 0$ such that i is maximal.
(4) If $a_{i1}, a_{i2}, a_{i3}, \ldots., a_{in} \geq 0$ STOP; the maximization problem is infeasible. Otherwise, continue.
(5) If i = m, choose $a_{mj} < 0$,pivot on a_{mj}, and go to step(1). If $i < m$,choose $a_{ij} < 0$,
compute $\min\limits_{k>i}\{b_i/a_{ij}\} \cup \{(b_k/a_{kj}) : a_{kj} > 0\} = b_p/a_{pj}$,
pivot on a_{pj} and go to (1)

Remarks 2.4.

There may be more than one value of p for which b_p/a_{pj} is minimum, at that time use anticycling rules.

Example 2.4.

Apply the simplex algorithm above to the maximum tableau

x_1	x_2	-1	
-1	-2	-3	$= -t_1$
1	1	3	$= -t_2$
1	1	2	$= -t_3$
-2	4	0	$= f$

(1) The initial tableau is clearly a maximum tableau.

(2) We proceed to step (3) since $b_1 = -3$ is negative.

(3) We must choose $b_1 = -3$.

(4) We proceed to step (5) since $a_{11} = -1$ and $a_{12} = -2$ are both negative.

(5) By anticycling rules we choose $a_{11} = -1$. Since $1 = i < m = 3$, we compute

$$\min\{(b_1/a_{11} = 3/1), (b_2/a_{21} = 3/1), (b_3/a_{31} = 2/1)\} = b_3/a_{31}$$

Hence pivot on a_{31}

x_1	x_2	-1	
-1	-2	-3	$= -t_1$
1	1	3	$= -t_2$
1^*	1	2	$= -t_3$
-2	4	0	$= f$

t_3	x_2	-1	
1	-1	-1	$= -t_1$
-1	0	1	$= -t_2$
1	1	2	$= -x_1$
2	6	4	$= f$

Go to step (1).
(1) The initial tableau is clearly a maximum tableau.
(2) We proceed to step (3) since $b_1 = -1$ is negative.
(3) We must choose $b_1 = -1$
(4) We proceed to step (5) since a_{12} = - 1 is negative.
(5) We choose $a_{12} = -1$. Since $1 = i < m = 3$, we compute

$$min\{(b_1/a_{12}), (b_3/a_{32} = 2/1)\} = b_1/a_{12}$$

Hence pivot on a_{12}

t_3	x_2	-1	
1	-1^*	-1	$= -t_1$
-1	0	1	$= -t_2$
1	1	2	$= -x_1$
2	6	4	$= f$

t_3	t_1	-1	
-1	-1	1	$= -x_2$
-1	0	1	$= -t_2$
2	1	1	$= -x_1$
8	6	-2	$= f$

Go to step (1).
(1) The initial tableau is clearly a maximum tableau.
(2) Here $b_1, b_2, b_3 \geq 0$ i.e., the tableau is maximum basic feasible.So using Simplex Algorithm for Maximum Basic Feasible Tableaus we can solve the problem.

Definition 2.8.

A canonical maximization or canonical minimization linear programming problem is said to be **infeasible** if it has no feasible solutions.

Hence an infeasible linear programming problem has an empty constraint set. As with unboundedness, infeasibility in linear programming problems should be viewed as pathological. Even though most real-life problems have feasible solutions if properly posed, it is advantageous that our algorithm recognizes the existence of infeasibility and terminates. If we STOP in step (4), the current tableau is of the form

(	ind.	var's	)	-1	
≥ 0	≥ 0	.....	≥ 0	< 0	$-x$
					$= f$

The equation given by the i^{th} row of this tableau is
$(\geq 0)($ ind.var.$) + ... + (\geq 0)($ ind.var.$) - (< 0) = -x$
i.e.
$(\geq 0)($ ind.var.$) + ... + (\geq 0)($ ind.var.$) + (> 0) = -x$
Since all of the independent variables are nonnegative in any feasible solution,we have$-x > 0$ implies $x < 0$
which is impossible since dependent variables are nonnegative in any feasible solution. Hence the maximization problem has no feasible solutions.Hence the tableau form that terminates the simplex algorithm for maximum tableaus with an infeasible linear programming problem.

Example 2.5.

Apply the simplex algorithm to the maximum tableau

x_1	x_2	-1	
-1	-1	-3	$= -t_1$
1	1	2	$= -t_2$
2	-4	0	$= f$

(1) The initial tableau is clearly a maximum tableau.
(2) We proceed to step (3) since $b_1 = -3$ is negative.
(3) We must choose $b_1 = -3$.
(4) We proceed to step (5) since $a_{11} = -1$ and $a_{12} =$ - 1 is negative,by anticycling rules we choose $a_{11} = -1$
(5) We choose $a_{11} = -1$. Since $1 = i < m = 2$ we compute $\min\{b_1/a_{11}\} \cup \{b_2/a_{21} = 2/1\} = b_2/a_{21}$. i.e. pivot on a_{21}

x_1	x_2	-1	
-1	-1	-3	$= -t_1$
1^*	1	2	$= -t_2$
2	-4	0	$= f$

t_2	x_2	-1	
1	0	-1	$= -t_1$
1	1	2	$= -x_1$
-2	-6	-4	$= f$

Go to step (1).
(1) The initial tableau is clearly a maximum tableau.
(2) We proceed to step (3) since $b_1 = -1$ is negative.
(3) We must choose $b_1 = -1$.
(4) $a_{11}, a_{12} \geq 0$; STOP; the maximization problem is infeasible.
Hence the constraint set of this canonical maximization linear programming problem is empty. This can be seen without graphing by looking at the main constraints of the original problem, namely
$-x_1 - x_2 \leq 3$
$x_1 + x_2 \leq 2$
Multiplying both sides of the first constraint by -1, we obtain
$x_1 + x_2 \geq 3$
$x_1 + x_2 \leq 2$
Since a quantity can never by greater than or equal to 3 and less than or equal to 2 simultaneously, this canonical maximization

problem is infeasible.

Remarks 2.5.

On step (3) of the simplex algorithm for maximum tableaus,the choice of $b_i < 0$ with maximal i in this step assures that all non-negative b's below b_i remain non-negative in the new tableau after pivoting. This can be helpful in locating arithmetic mistakes in the b-columns of the tableaus. Note, though, that the maximality of i is unimportant for the determination of infeasibility in step (4).If you can find just one negative b corresponding to a row of non-negative a's, the linear programming problem is infeasible and the algorithm terminates.

2.5 The Simplex Algorithm for Minimum Tableaus

To obtain a simplex algorithm for canonical minimization linear programming problems, we use a simple trick to convert minimum tableaus into maximum tableaus whence the algorithm just discussed above can be implemented. This trick is called **negative transposition**.

Definition 2.9.

The negative transpose of the minimum tableau

$$
\begin{array}{|c|ccc|c|}
\hline
x_1 & a_{11} & a_{21}\ldots & a_{m1} & c_1 \\
x_2 & a_{12} & a_{22}\ldots & a_{m2} & c_2 \\
\vdots & \vdots & & \vdots & \vdots \\
x_n & a_{1n} & a_{2n}\ldots & a_{mn} & c_n \\
\hline
\text{-1} & b_1 & \ldots & b_m & d \\
\hline
 & = t_1 & \ldots & = t_m & = g \\
\hline
\end{array}
$$

is the maximum tableau

x_1	x_2	$\dots$	x_n	-1	
$-a_{11}$	$-a_{12}$	$\dots$	$-a_{1n}$	$-b_1$	$= -t_1$
$-a_{21}$	$-a_{22}$	$\dots$	$-a_{2n}$	$-b_2$	$= -t_2$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
$-a_{m1}$	$-a_{m2}$	$\dots$	$-a_{mn}$	$-b_m$	$= -t_m$
$-c_1$	$-c_2$	$\dots$	$-c_n$	$-d$	$= -g$

and vice versa.

Note that every column of the minimum tableau becomes a negated row in the maximum tableau except for the first column of the tableau containing the independent variables and - 1 which becomes a row but is not negated.By looking at the equations represented by these tableaus, we see that every equation of the minimum tableau has been multiplied by -1 in the maximum tableau. This is the only difference effected by this tableau transition multiplication of each equation of the minimum tableau by - 1 simply gives the form in a maximum tableau, namely the negated dependent variables to the east.
Now we can solve the maximum table obtained above using the simplex algorithm for maximum tableaus which maximizes the objective function $-g$.The original problem was to minimize the objective function g. The relationship between the two quantities max$(-g)$ and min g is -min g = - max$(-g)$.

2.5.1 The Simplex Algorithm for Minimum Tableaus

(1) The initial tableau is of the form

$(ind.var)$	a_{11} a_{12} $\vdots$ a_{1n}	$a_{21}\dots$ $a_{22}\dots$ $\vdots$ $a_{2n}\dots$	a_{m1} a_{m2} a_{mn}	c_1 c_2 $\vdots$ c_n
-1	b_1	$b_2\dots$	b_m	d
	$(dep.$	$var's$	$)$	$= g$

(2) Take the negative transpose of the tableau to obtain a maximum tableau.
(3) Apply the simplex algorithm for maximum tableaus.
(4) $\min g = \max(-g)$

Example 2.6.

Apply the simplex algorithm above to the minimum tableau

x_1	20	25	300
x_2	40	20	500
-1	1000	800	0
	$= t_1$	$= t_2$	$= g$

(1) The initial tableau is clearly a minimum tableau.
(2) Take the negative transpose of the tableau to obtain a maximum tableau.

x_1	x_2	-1	
-20	-40	-1000	$= -t_1$
-25	-20^*	-800	$= -t_2$
-300	-500	0	$-g$

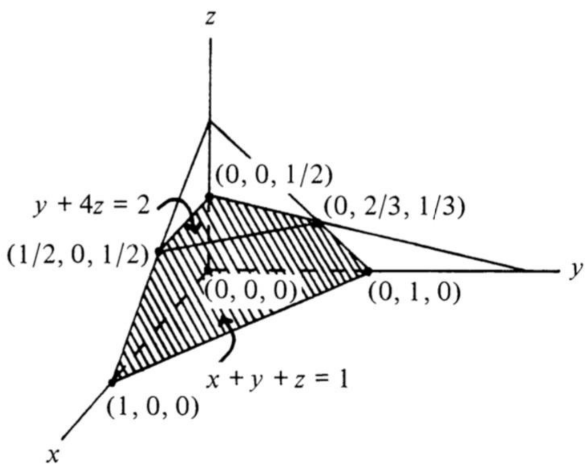
(3) Here choose b_2. Since $i = m = 2$ pivot on a_{22}.

x_1	t_2	-1	
30	-2	600	$= -t_1$
$5/4$	$-1/20$	40	$= -x_2$
325	-25	20000	$-g$

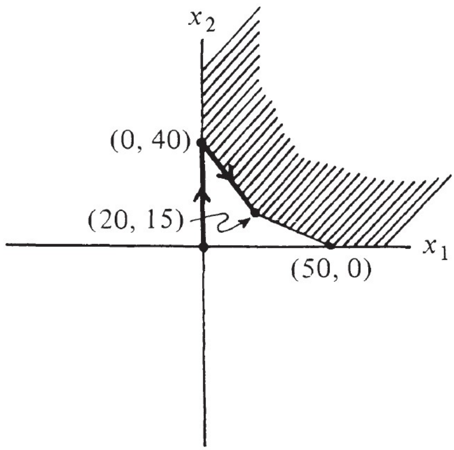
(4) Clearly this table is maximum basic feasible table. So applying simplex algorithm for maximum basic feasible table we choose $c_1 = 325$.
Now $\min\{600/30, 40/1.25\} = 600/30$.
So pivot on $a_{11} = 30$.

t_1	t_2	-1	
$1/30$	$-1/15$	20	$= -x_1$
$-1/24$	$1/30$	15	$= -x_2$
$-65/6$	$-10/3$	13500	$-g$

(5) since $c_1, c_2 \leq 0$.We STOP.
The optimal solution to the maximization problem is
$t_1 = t_2 = 0, x_1 = 20, x_2 = 15, \max(-g) = -13500$
Hence the optimal solution to the original minimization problem is $t_1 = t_2 = 0, x_1 = 20, x_2 = 15, \min g = -\max(-g) = 13500$.
Example 2.6 is the simplex algorithm solution of Example 1.2 of Chapter 1.



Consider the x_1 and x_2 values of the basic solutions of the three maximum tableaus. They are (0,0),(0,40) and (20,15) respectively.The movement exhibited by these basic solutions in the constraint set diagram is illustrated below:



Note that the basic solutions of the second and third tableaus are feasible while the basic solution of the first tableau is not feasible.There is a good reason for this-only the second and third tableaus are maximum basic feasible tableaus! Recall that, in general, the simplex algorithm transition between maximum basic feasible tableaus is designed so that the objective function (-9 in this particular case) is not decreased. Hence each such transition maintains or increases the value of the objective function, usually until either a maximum value is reached or the algorithm detects unboundedness.

SOLVED PROBLEMS:

Problem 2.3.

Minimize

$$g(x, y, z) = -x - y$$

subject to

$$3x + 6y + 2z \leq 6$$
$$y + z \geq 1$$
$$x, y, z \geq 0$$

Solution:
Canonical form of the given problem is
Minimize

$$g(x, y, z) = -x - y$$

subject to

$$-3x - 6y - 2z \geq -6$$
$$y + z \geq 1$$
$$x, y, z \geq 0$$

then the minimum table is given as

x	-3	0	-1
y	-6	1	-1
z	-2	1	0
-1	-6	1	0
	$=t_1$	$=t_2$	$= g$

Take negative transpose of the given table and convert to maximum table, then we get

x	y	z	-1	
3	6	2	6	$=-t_1$
0	-1	-1	-1	$=-t_2$
1	1	0	0	$=-g$

here b_2 is negative and i = m then a_{23} be the pivot

x	y	z	-1	
3	6	2	6	$= -t_1$
0	-1	-1*	-1	$= -t_2$
1	1	0	0	$= -g$

x	y	t_2	-1	
3	4	2	4	$= -t_1$
0	1	-1	1	$= -z$
1	1	0	0	$= -g$

According to anticyclic rule we choose c_1=1 and pivot a_{11}=3

x	y	t_2	-1	
3*	4	2	4	$= -t_1$
0	1	-1	1	$= -z$
1	1	0	0	$= -g$

t_1	y	t_2	-1	
1/3	4/3	2/3	4/3	$= -x$
0	1	-1	1	$= -z$
-1/3	-1/3	-1	-4/3	$= -g$

Since all b_i's are positive and all c_j's are negative, we have attained optimal solution as x= 4/3, y=0,z=1,t_1=0,t_2=0, g = -4/3.

Problem 2.4.

Solve

x	-2	1	-3
y	1	-2	-2
-1	1	0	0
	$=t_1$	$=t_2$	$= g$

Solution:
The given table is identified as Minimization problem, then we have to take negative transpose as

x	y	-1	
2	-1	-1	$= -t_1$
-1	2	0	$= -t_2$
3	2	0	$= -g$

Since b_1 is negative and i less than m ,by considering minimum ratio we set a_{22} to be pivot

x	y	-1	
2	-1	-1	$= -t_1$
-1	2*	0	$= -t_2$
3	2	0	$= -g$

x	t_2	-1	
3/2	1/2	-1	$= -t_1$
-1/2	1/2	0	$= -t_2$
4	-1	0	$= -g$

We have b_1 to be negative but all elements in that row is positive, So we can't continue the process and we conclude that the given system is infeasible.

Problem 2.5.

Solve

x	y	-1	
-1	-1	-2	$= -t_1$
1	-2	0	$= -t_2$
-2	1	1	$= -t_3$
-1	3	0	=f

Solution:
Here we have negative value -2 in the first row and a_{11} and a_{12} are same therefore by anticycling rules we choose x column and by finding minimum ratio we get a_{21}=1 as pivot

x	y	-1	
-1	-1	-2	$= -t_1$
1*	-2	0	$= -t_2$
-2	1	1	$= -t_3$
-1	3	0	=f

t_2	y	-1	
1	-3*	-2	$= -t_1$
1	-2	0	$= -x$
2	-3	1	$= -t_3$
1	1	0	=f

t_2	x	-1	
-1/3	-1/3	2/3	$= -t_1$
1/3	-2/3	4/3	$= y$
1	-1	3	$= -t_3$
4/3	1/3	-2/3	=f

The given table is unbounded since there is a column in which all the enteries are negative and the corresponding cj is positive.

Problem 2.6.

Solve the given table

x	y	-1	
1	-1	3	$= -t_1$
-2	1	2	$= -t_2$
2	-1	0	$= f$

Solution:

The given table is feasible and there exist a positive c_j , Therefore to eliminate this cj ,according to the rule of maximization problem, we get $a_{11} = 1$ as pivot

x	y	-1	
1*	-1	3	$= -t_1$
-2	1	2	$= -t_2$
2	-1	0	$= f$

t_1	y	-1	
1	-1	3	$= -x$
2	-1	8	$= -t_2$
-2	1	-6	$= f$

We have a positive c_j but all the elements in that column are negative. So we can't continue and the system is concluded to be unbounded system.

Chapter 3

Non Canonical LPP

3.1 Introduction

The important properties of a canonical linear programming problem are the non-negativity of the initial independent variables and the inequality form of the main constraints. However, these conditions may be violated in some problems and such types of problems are said to be non canonical linear programming problems. The concern of this chapter is the formalization of these modifications.

3.2 Unconstrained Variables

Definition 3.1.

A real variable in a linear programming problem is said to be *unconstrained* if there is no non-negativity constraint on the variable.

The first type of non-canonical linear programming problem has canonical maximization or canonical minimization form except that there may not be non-negativity constraints on all of the independent variables, i.e., some of these variables may be unconstrained.To solve such problems we use the following steps.

Maximization linear programming problems with unconstrained independent variables
(1) We begin by recording the problem in a Tucker tableau as usual. To record the fact that there are unconstrained variables, we put those variables in circles. Note that slack variables are always constrained to be non-negative.
(2) Pivot on a convenient entry that moves an unconstrained independent variable to the east. Never pivot on any entry in the - 1 column or the objective function row.
(3) File the equation corresponding to the row of the unconstrained variable pivoted, and delete that row from the tableau.
(4) If the table is in canonical form use the simplex algorithm to manipulate this canonical tableau. Otherwise go to step (2).
(5) Substitute these values in filed equations and obtain the required solution.

Remarks 3.1.

After moving all your unconstrained variables to east and deleting the corresponding pivoted row if you have obtained a single row then it is easy to analyse the row and get the solutions.
(a) If all the values on the row, except the value of objective function is negative or zero then you have obtained the optimal solution.Substitute these values in filed equations and obtain the required solution.
(b) If atleast one value of the row is positive then the objective function is unbounded.

Example 3.1.

Maximize

$$f(x, y) = x + 3y$$

subject to

$$x + 2y \leq 10$$
$$-3x - y \leq -15$$

In this problem, both x and y are unconstrained.
(1) We begin by recording the problem in a Tucker tableau as usual. To record the fact that x and y are unconstrained,we put them in circles or square brackets.

x	y	-1	
1	2	10	$= -t_1$
-3	-1	-15	$= -t_2$
1	3	0	$= f$

(2) The acceptable pivot entries are $a_{11}, a_{12}, a_{21}, a_{22}$; in a non-canonical maximum tableau.never pivot on any entry in the -1 column or the objective function row.The most convenient of these pivots for our purposes is $a_{11} = 1, a_{22} = -1$.We choose $a_{11} = 1$ for definiteness; pivoting on 1 moves x down to the east.

x	y	-1	
1^*	2	10	$= -t_1$
-3	-1	-15	$= -t_2$
1	3	0	$= f$

t_1	y	-1	
1	2	10	$= -$x
3	5	15	$= -t_2$
-1	1	-10	$= f$

(3) Since x is unconstrained, the equation represented by the first row of the new tableau represents no constraint on t_1 or y.

Notice, however, that this equation will enable us to solve for x if we know t_1 and y. We hence file the equation corresponding to the x row (for future use) and delete that row from the tableau.

t_1	y	-1	
3	5	15	$= -t_2$
-1	1	-10	$= f$

The corresponding filed equation is $t_1 + 2y - 10 = -x$.
(4) The table obtained is not in canonical form.So again using step (2) of our algorithm we have to move y down to the east.To pivot y down to the east, we must use $a_{12} = 5$

t_1	y	-1	
3	5^*	15	$= -t_2$
-1	1	-10	$= f$

t_1	t_2	-1	
$3/5$	$1/5$	3	$= -y$
$-8/5$	$-1/5$	-13	$= f$

Since y is unconstrained, the equation represented by the first row of the new tableau represents no constraint on t_1 or t_2. Notice, however, that this equation will enable us to solve for y if we know t_1 and t_2. We hence file the equation corresponding to the y row and delete that row from the tableau.

t_1	t_2	-1	
$-8/5$	$-1/5$	-13	$= f$

The corresponding file equation is $\frac{3}{5}t_1 + \frac{1}{5}t_2 - 3 = -y$.
(5) This tableau is now in canonical form since all independent variables are constrained to be non-negative.At this point, we would ordinarily use the simplex algorithm to manipulate this canonical tableau into basic solution optimal form. However, the basic solution of the final tableau above is clearly optimal since

any change in t_1 or t_2 would decrease f. Hence, the optimal solution to the non-canonical maximization linear programming problem is given by $t_1 = t_2 = 0, y = 3$ (from second filed equation), $x = 4$(from first filed equation).

Example 3.2.

Maximize

$$f(x, y) = x + 3y$$

subject to

$$\begin{aligned} x + 2y &\leq 10 \\ 3x + y &\leq 15 \end{aligned}$$

Here both x and y are unconstrained.
(1) We record the problem in a Tucker tableau.

x	y	-1	
1	2	10	$= -t_1$
3	1	15	$= -t_2$
1	3	0	$= f$

(2) Pivot on a convenient entry that moves an unconstrained independent variable to the east.Here the most convenient entry will be $a_{11} = 1$, which will move the variable x to east.

x	y	-1	
1^*	2	10	$= -t_1$
3	1	15	$= -t_2$
1	3	0	$= f$

t_1	y	-1	
1	2	10	$= -$x
-3	-5	-15	$= -t_2$
-1	1	-10	$= f$

(3) File the equation corresponding to the row of the unconstrained variable pivoted, and delete that row from the tableau.

t_1	y	-1	
-3	-5	-15	$= -t_2$
-1	1	-10	$= f$

The corresponding file equation is $t_1 + 2y - 10 = -x$. (4) Now pivot y down to the east, file the equation corresponding to the y row, and delete that row from the tableau.

t_1	y	-1	
-3	-5^*	-15	$= -t_2$
-1	1	-10	$= f$

t_1	t_2	-1	
$3/5$	$-1/5$	3	$= -y$
$-8/5$	$1/5$	-13	$= f$

t_1	t_2	-1	
$-8/5$	$1/5$	-13	$= f$

The corresponding file equation is $(3/5)t_1 - (1/5)t_2 - 3 = -y$. (5) The final tableau above is in canonical form since all independent variables are constrained to be non-negative. But the objective function f can clearly be made as large as we please by putting $t_1 = 0$ and letting $t_2 \to \infty$.Hence, the non-canonical maximization linear programming problem is unbounded.

Example 3.3.

Maximize

$$f(x, y) = x + 3y$$

subject to

$$x + 2y \leq 10$$
$$3x + y \leq 15, x \geq 0$$

Here, only y is unconstrained.
(1) We record the problem in a Tucker tableau.

x	y	-1	
1	2	10	$= -t_1$
3	1	15	$= -t_2$
1	3	0	$= f$

(2) Pivot on a convenient entry that moves an unconstrained independent variable to the east.Here the most convenient entry will be $a_{22} = 1$, which will move the variable y to east.

x	y	-1	
1	2	10	$= -t_1$
3	1^*	15	$= -t_2$
1	3	0	$= f$

x	t_2	-1	
-5	-2	-20	$= -t_1$
3	1	15	$= -y$
-8	-3	-45	$= f$

(3) File the equation corresponding to the row of the unconstrained variable pivoted, and delete that row from the tableau.

x	t_2	-1	
-5	-2	-20	$= -t_1$
-8	-3	-45	$= f$

The corresponding file equation is $3x + t_2 - 15 = -y$. (4) The final tableau above is in canonical form since all independent variables are constrained to be nonnegative. Hence, we apply the simplex algorithm.The above table is a maximum table.We have to convert it into maximum basic feasible table.For that choose $b_1 = -20$.Then our pivot entry will be $a_{12} = -2$.On pivoting we get

x	t_1	-1	
$5/2$	$-1/2$	10	$= -t_2$
$-1/2$	$-3/2$	-15	$= f$

(5) Now clearly the table is maximum basic feasible table.By applying simplex algorithm we find that the basic solution of the tableau above is optimal. Hence the optimal solution of the non-canonical maximization linear programming problem is given by

$$x = t_1 = 0, t_2 = 10, y = 5$$

(from filed equation), $\max f = 15$.

Minimization linear programming problems with unconstrained independent variables

(1) We begin by recording the problem in a Tucker tableau as usual. To record the fact that there are unconstrained, we put those variables in square brackets.Note that slack variables are always constrained to be non-negative.
(2) Pivot on a convenient entry that moves an unconstrained independent variable to the south.Never pivot on any entry in the - 1 row or the objective function column of any tableau.
(3) File the equation corresponding to the column of the unconstrained variable pivoted, and delete that column from the tableau.
(4)Once a canonical minimum tableau is obtained, the simplex algorithm is applied.Otherwise go to step (2)
(5)Using simplex algorithm obtain the value of variables in the table. Substitute these values in filed equations and obtain the required solution.

3.3 Equations of Constraint

The second type of non-canonical linear programming problem has canonical maximization or canonical minimization form except that there may be some equality constraints among the initial variables instead of inequality constraints only.

3.3.1 Maximisation linear programming problems with Equations of Constraint

(1) Equations of constraint are assigned slack “variable”0 when written in slack form and the initial Tucker table is prepared.It is highly recommended that the negative sign in front of the 0 be included; as with the sign of any other slack variable, this sign will stay behind when variables are interchanged during pivoting.
(2) Pivot on the most convenient entry that moves the slack “variable”0 to the north. Never pivot on any entry in the -1 column or the objective function row.
(3) Delete the column corresponding to this 0.
(4) Repeat Step (2) and Step (3) until you remove all the slack “variable”0.
(5) Use simplex algorithm to obtain your required solution.

Example 3.4.

Maximize

$$f(x, y, z) = 2x + y - 2z$$

subject to

$$\begin{aligned} x + y + z &\leq 1 \\ y + 4z &= 2 \\ x, y, z &\geq 0 \end{aligned}$$

Note the equation of constraint $y + 4z = 2$. This equation can be rewritten in the form $y + 4z - 2 = -0$ this form corresponds to the usual slack form of a main constraint of a canonical maximization linear programming problem except that the slack

"variable" is 0. It is highly recommended that the sign in front of the 0 be included; as with the sign of any other slack variable, this sign will stay behind when variables are interchanged during pivoting.
(1) We now record the problem in a Tucker tableau.

x	y	z	-1	
1	1	1	1	$= -t_1$
0	1	4	2	$= -0$
2	1	-2	0	$= f$

(2) Pivot the 0 up to the north. We merely need to choose the most convenient pivot for our purposes. The acceptable pivot entries are a_{22} and a_{23} ; again, never pivot on any entry in the -1 column or the objective function row of any maximum tableau. The most convenient of these pivots is $a_{22} = 1$ since all tableau entries will remain integral after pivoting:

x	y	z	-1	
1	1	1	1	$= -t_1$
0	1^*	4	2	$= -0$
2	1	-2	0	$= f$

x	0	z	-1	
1	-1	-3	-1	$= -t_1$
0	1	4	2	$= -y$
2	-1	-6	-2	$= f$

(3) Now the second column of the new tableau above never enters into the main constraints or objective function of this tableau since every entry is multiplied by 0. We may then delete this column without any loss of information.

x	z	-1	
1	-3	-1	$= -t_1$
0	4	2	$= -y$
2	-6	-2	$= f$

(4) This tableau is now in canonical form since all 0 "slack variables" have been removed from the tableau. Hence, we apply the simplex algorithm.

x	z	-1	
1	-3^*	-1	$= -t_1$
0	4	2	$= -y$
2	-6	-2	$= f$

x	t_1	-1	
$-1/3$	$-1/3$	$1/3$	$= -z$
$4/3$	$4/3$	$2/3$	$= -y$
0	-2	0	$= f$

The basic solution of the final tableau above is optimal. Hence an optimal solution of the non-canonical maximization linear programming problem is given by

$$x = t_1 = 0, z = 1/3, y = 2/3 \text{ Max } f = 0$$

Example 3.5.

Maximize

$$f(x, y, z) = x + 4y + 2z$$

subject to

$$\begin{aligned} x + 2y + 3z &\leq 6 \\ 4x - 7y &= 25 \\ x, y, z &\geq 0 \end{aligned}$$

(1) We record the problem in a Tucker tableau

x	y	z	-1	
1	2	3	6	$= -t_1$
4	-7	0	28	$= -0$
1	4	2	0	$= f$

(2) Pivot on the most convenient entry that moves the slack "variable"0 to the north.

x	y	z	-1	
1	2	3	6	$= -t_1$
4^*	-7	0	28	$= -0$
1	4	2	0	$= f$

0	y	z	-1	
$-1/4$	$15/4$	3	-1	$= -t_1$
$1/4$	$-7/4$	0	7	$= -x$
$-1/4$	$23/4$	2	-7	$= f$

(3) Delete the column corresponding to this 0.

y	z	-1	
$15/4$	3	-1	$= -t_1$
$-7/4$	0	7	$= -x$
$23/4$	2	-7	$= f$

(4) The final tableau above is in canonical form since all 0 "slack variables"have been removed from the tableau. The simplex algorithm applied to this tableau immediately yields infeasibility by the first row. Hence, the non-canonical maximization linear programming problem is infeasible.

Minimization linear programming problems with equations of constraint

(1) Equations of constraint are assigned slack "variable"0 when written in slack form and the initial Tucker table is prepared.
(2) Pivot on the most convenient entry that moves the slack "variable"0 to the west.Never pivot on any entry in the - 1 row or the objective function column.
(3) Delete the row corresponding to this 0.
(4) Repeat Step (2) and Step (3) until you remove all the slack "variable"0.

(5) Once a canonical minimum tableau is obtained, the simplex algorithm is applied.

Unconstrained variables and Equations of constraint

We conclude this section with an example of a non-canonical linear programming problem which combines the two types of non-canonical behaviour discussed in this chapter.

Example 3.6.

Maximize

$$f(x, y, z) = x + 2y + z$$

subject to

$$\begin{aligned} x + y + z &= 6 \\ x + y &\leq 1 \\ x, z &\geq 0 \end{aligned}$$

In this problem, we have an equation of constraint given by

$$x + y + z = 6$$

and an unconstrained independent variable y. Not surprisingly, we simultaneously use both of the methods that have been developed in this chapter.

(1) Recording the problem in a Tucker tableau, we obtain

x	y	z	-1	
1	1	1	6	$= -0$
1	1	0	1	$= -t_1$
1	2	1	0	$= f$

(2) Now, our goal is to pivot the unconstrained independent variable y down to the east and the slack “variable”0 up to the north. We can accomplish both of these at the same time by choosing the correct pivot, namely $a_{12} = 1$.

x	y	z	-1	
1	1^*	1	6	$=-0$
1	1	0	1	$=-t_1$
1	2	1	0	$=f$

x	0	z	-1	
1	1	1	6	$=-y$
0	-1	-1	-5	$=-t_1$
-1	-2	-1	-12	$=f$

(3) Delete y row and delete 0 column.Also file the equation corresponding to row y.

x	z	-1	
0	-1	-5	$=-t_1$
-1	-1	-12	$=f$

The corresponding file equation is $x+z-6=-y$ (4) Now the table is in canonical form.You can apply simplex method.The current table is a maximum table since $b_1 = -5$.So pivot on $a_{12}=-1$.

x	z	-1	
0	-1^*	-5	$=-t_1$
-1	-1	-12	$=f$

x	t_1	-1	
0	-1	5	$=-z$
-1	-1	-7	$=f$

(5) The basic solution of the final tableau above is optimal. Hence the optimal solution of the non-canonical maximization linear programming problem is given by

$$x=t_1=0, z=5,\ \text{Max}f=7, y=1$$

(from filed equation)

Remarks 3.2.

Never apply the simplex algorithm until the non-canonical tableau has been transformed into a canonical tableau. A canonical tableau is a tableau having no unconstrained independent variables and no slack "variable" of 0.

Solved Problems- unconstrained variables

Problem 3.1.

Maximize

$$f = x - y + z$$

subject to

$$x + y \geq 2 \tag{3.1}$$
$$z - y \geq 3 \tag{3.2}$$
$$2x + z \leq 8 \tag{3.3}$$

where x,y,z are unconstrained

Solution: First, we want to convert the given system as canonical form of Maximization problem
Maximize

$$f = x - y + z$$

subject to

$$-x - y \leq -2$$
$$-z + y \leq -3$$
$$2x + z \leq 8$$

where x,y,z are unconstrained variables

$[x]$	$[y]$	$[z]$	-1	
-1*	-1	0	-2	$= -t_1$
0	1	-1	-3	$= -t_2$
2	0	1	8	$= -t_3$
1	-1	1	0	$= f$

t_1	$[y]$	$[z]$	-1	
-1	1	0	2	$= -[x]$
0	1	-1	-3	$= -t_2$
2	-2	1	4	$= -t_3$
1	-2	1	-2	$= f$

Delete first row then we get

t_1	$[y]$	$[z]$	-1	
0	1	-1	-3	$= -t_2$
2	-2	1	4	$= -t_3$
1	-2	1	-2	$= f$

Choosing 1 as pivot ,

t_1	$[y]$	$[z]$	-1	
0	1*	-1	-3	$= -t_2$
2	-2	1	4	$= -t_3$
1	-2	1	-2	$= f$

t_1	t_2	$[z]$	-1	
0	1	-1	-3	$= -[y]$
2	2	-1	-2	$= -t_3$
1	2	-1	-8	$= f$

t_1	t_2	$[z]$	-1	
2	2	-1	-2	$= -t_3$
1	2	-1	-8	$= f$

t_1	t_2	$[z]$	-1	
2	2	-1*	-2	$= -t_3$
1	2	-1	-8	$= f$

t_1	t_2	t_3	-1	
-2	-2	-1	2	$= -[z]$
-1	0	-1	-6	$= f$

t_1	t_2	t_3	-1	
-1	0	-1	-6	$= f$

The filed equations of the above tableaus are

$$-t_1 + y - 2 = -x$$
$$t_2 - z + 3 = -y$$
$$-2t_1 - 2t_2 - t_3 - 2 = -z$$

Since $t_1 = t_3 = 0$ which implies $-2t_2 - 2 = -z$.
Then we get $y = t_2 - 1$. Substituting the values of y and t_1 we get $x = 3 - t_2$, since $t_2 \geq 0$ implies $x \leq 3$. Therefore we have

$$x \leq 3, y = 2 - x, z = 8 - 2x \text{ and Max } f = 6$$

Minimization with equation of constraint

Problem 3.2.

Solve the minimization problem
Minimize

$$g(x, y, z) = 3x + y + 2z$$

subject to

$$x + 2y + 3z \geq 24$$
$$2x + 4y + 3z = 36$$
$$x, y, z \geq 0$$

Solution: Create the initial canonical table

x	1	2	3
y	2	4	1
z	3	3	2
-1	24	36	0
	$= t_1$	=0	=g

we have to move 0 variable from south to west, so we choose 2 as pivot

x	1	2*	3
y	2	4	1
z	3	3	2
-1	24	36	0
	$= t_1$	=0	=g

0	1/2	1/2	3/2
y	0	-2	-5
z	3/2	-3/2	-5/2
-1	6	-18	-54
	$= t_1$	=x	=g

Delete the first row, then we get

y	0	-2	-5
z	3/2	-3/2	-5/2
-1	6	-18	-54
	$= t_1$	=x	=g

Since we have no 0 variable, we now convert the given minimum problem to maximization problem

y	z	-1	
0	-3/2	-6	$=-t_1$
2	3/2	18	=-x
5	5/2	54	=-g

Since b_1 is not positive we have to set pivot as -3/2 to make the table as feasible

y	z	-1	
0	-3/2*	-6	$=-t_1$
2	3/2	18	=-x
5	5/2	54	=-g

y	t_1	-1	
0	-2/3	4	$=-z$
2*	1	12	=-x
5	5/3	44	=-g

x	t_1	-1	
0	-2/3	4	=-z
1/2	1/2	6	=-y
-5/2	-5/6	14	=-g

Thus we attain optimal solution

$$x = 0, y = 6, z = 4, t_1 = 0 \text{ Min } g = 14$$

Maximization with equation of constraint

Problem 3.3.

Maximize

$$f(x, y) = x + y$$

subject to

$$2x + y = 5$$
$$x - y = -2$$
$$x + 3y = 6$$
$$x, y \geq 0$$

Solution:

x	y	-1	
2	1	5	=-0_1
1	-1	-2	=-0_2
1	3	6	=-0_3
1	1	0	=f

x	y	-1	
2*	1	5	=-0_1
1	-1	-2	=-0_2
1	3	6	=-0_3
1	1	0	=f

0_1	y	-1	
1/2	1/2	5/2	=-x
-1/2	-3/2	-9/2	=-0_2
-1/2	5/2	7/2	=-0_3
-1/2	1/2	-5/2	=f

deleting the first column we get,

y	-1	
1/2	5/2	=-x
-3/2*	-9/2	=-0_2
5/2	7/2	=-0_3
1/2	-5/2	=f

0_2	-1	
1/3	1	=-x
-2/3	3	=-y
5/3	-4	=-0_3
1/3	-4	=f

-1	
1	=-x
3	=-y
-4	=-0_3
-4	=f

Thus the given problem is infeasible.

Problem 3.4.

Maximize

$$f(x, y, z) = 3x - 2y + 3z$$

Subject to

$$\begin{aligned} x - y + 2z &= 6 \\ x + 2z &= 8 \\ y + 2z &\geq 2 \\ y, z &\geq 0 \end{aligned}$$

Solution:

x	y	z	-1	
1*	-1	2	6	=-0_1
1	0	2	8	=-0_2
0	-1	-2	-2	=-t_1
3	-2	3	0	=f

0_1	y	z	-1	
1	-1	2	6	=-x
-1	1	0	2	=-0_2
0	1	2	2	=-t_1
-3	1	-3	-18	=f

Delete first column and first row.Also file the equation corresponding to row x, then we get,

y	z	-1	
1*	0	2	=-0_2
1	2	2	=-t_1
1	-3	-18	=f

0_2	z	-1	
1*	0	2	=-y
-1	2	0	=-t_1
-1	-3	-20	=f

delete first column, then we get,

z	-1	
0	2	=-y
2	0	=-t_1
-3	-20	=f

All bi's are positive and all cj's are negative. Therefore we attain maximal optimal table $x = 8$ (from the filed equation),$y = 2, z = 0, t_1 = 0$ and Max$f = 20$

Problem 3.5.

Maximize

$$f(x, y, z) = 2x + y - 2z$$

subject to

$$\begin{aligned} x + y + z &\leq 1 \\ y + 4z &= 2 \\ x, y, z &\geq 0 \end{aligned}$$

Solution:
Construct the initial table

x	y	z	-1	
1	1	1	1	=-t_1
0	1	4	2	=-0
2	1	-2	0	=f

choose 1 as pivot

x	y	z	-1	
1	1	1	1	=-t_1
0	1*	4	2	=-0
2	1	-2	0	=f

x	0	z	-1	
1	-1	-3	-1	=-t_1
0	1	4	2	=-y
2	-1	-6	-2	=f

deleting column, we get

x	z	-1	
1	-3	-1	=-t_1
0	4	2	=-y
2	-6	-2	=f

x	z	-1	
1	-3*	-1	=-t_1
0	4	2	=-y
2	-6	-2	=f

x	t_1	-1	
-1/3	-1/3	1/3	=-z
4/3	4/3	2/3	=-y
0	-2	0	=f

Max $f = 0, t_1 = 0, y = 2/3 - (4/3)x, z = 1/3 + (1/3)x,\ 0 \leq x \leq 1/2$

Chapter 4

Duality Theory

4.1 Introduction

The concept of duality is of fundamental importance in linear programming. This chapter is devoted to duality with the goal of gaining a greater understanding of the relationships that exist between dual linear programming problems. Here we discuss duality both in canonical and non canonical linear programming problem.

4.2 Duality in Canonical Tableaus

Any canonical tableau can be interpreted both as a canonical maximization linear programming problem and a canonical minimization linear programming problem. For example, the canonical tableau

	x_1	x_2	$\ldots$	x_n	-1	
y_1	a_{11}	a_{12}	$\ldots$	a_{1n}	b_1	$= -t_1$
y_2	a_{21}	a_{22}	$\ldots$	a_{2n}	b_2	$= -t_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_m	a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m	$= -t_m$
-1	c_1	c_2	$\ldots$	c_n	d	$= f$
	$= s_1$	$= s_2$	$\ldots$	$= s_n$	$= g$	

represents the canonical slack maximization linear programming problem given by the variables to the north and east (ignoring the variables to the west and south) and the canonical slack minimization linear programming problem given by the variables to the west and south (ignoring the variables to the north and east).

Definition 4.1
Any pair of canonical maximization and canonical minimization linear programming problems corresponding to the same tableau as above are said to exhibit *duality* or be *duals* of one another. The tableau of dual canonical linear programming problems is said to be a *dual canonical tableau.*

4.3 The Dual Simplex Algorithm

The simplex algorithm views all canonical linear programming pro blems from the perspective of canonical maximization linear programming problems, i.e., canonical maximization linear programming problems are handled directly and canonical minimization linear programming problems are transformed into equivalent canonical maximization linear programming problems by negative transposition.

Definition 4.2 Let

	a_{11}	a_{12}	$\ldots$	a_{1n}	b_1
	a_{21}	a_{22}	$\ldots$	a_{2n}	b_2
$(ind.var's)$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m
-1	c_1	c_2	$\ldots$	c_n	d
	$=($	$dep.$	$var's$	$)$	$= g$

be a tableau of a canonical slack minimization linear programming problem. The tableau is said to be **minimum basic feasible** if $c_1, c_2, \ldots, c_n \leq 0$.

4.4 The Dual Simplex Algorithm for Minimum Tableaus

First formulate the initial table of our canonical minimization linear programming problem.
(1) The current tableau is of the form

	a_{11}	a_{12}	$\ldots$	a_{1n}	b_1
	a_{21}	a_{22}	$\ldots$	a_{2n}	b_2
$(ind.var's)$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m
-1	c_1	c_2	$\ldots$	c_n	d
	$=($	$dep.$	$var's$	$)$	$= g$

(2) If $c_1, c_2, \ldots, c_n \leq 0$, go to (6), otherwise continue.
(3) Choose $c_j > 0$ such that j is maximal.
(4) If $a_{1j}, a_{2j}, ..., a_{mj} \leq 0$ STOP; the minimization problem is infeasible.Otherwise, continue.
(5) If $j = n$, choose $a_{in} > 0$, pivot on a_{in},and go to (1)if $j < n$,choose $a_{ij} > 0$, compute $\min_{k>j}\{c_j/a_{ij}\} \cup \{c_k/a_{ik} : a_{ik} < 0\} = c_p/a_{ip}$, pivot on a_{ip}, and go to (1).

(6) The current tableau is minimum basic feasible, i.e., of the form

	a_{11}	a_{12}	$\ldots$	a_{1n}	b_1
	a_{21}	a_{22}	$\ldots$	a_{2n}	b_2
$(ind.var's)$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m
-1	c_1	c_2	$\ldots$	c_n	d
	$=($	$dep.$	$var's$	$)$	$=g$

(7) If $b_1, b_2, \ldots\ldots\ldots b_m \geq 0$, STOP;the basic solution of the current minimum tableau is optimal. Otherwise, continue.
(8) Choose $b_i < 0$.
(9) If $a_{i1}, a_{i2}, \ldots\ldots, a_{in} \geq 0$, STOP; the minimization problem is unbounded. Otherwise, continue.
(10) Compute $\min\limits_{1 \leq j \leq n} \{(c_j/a_{ij} : a_{ij} < 0) = c_p/a_{ip}\}$
pivot on a_{ip}, and go to (6).

4.5 The Dual Simplex Algorithm for Maximum Tableaus

(1) The current tableau is of the form

$($	$ind.$	$var.'s$	$)$	-1	
a_{11}	a_{12}	$\ldots$	a_{1n}	b_1	
a_{21}	a_{22}	$\ldots$	a_{2n}	b_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$=-(dep.var's)$
a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m	
c_1	c_2	$\ldots$	c_n	d	$=f$

(2) Take the negative transpose of the tableau to obtain a minimum tableau.
(3) Apply the dual simplex algorithm for minimum tableaus.
(4) Max $f = -\min(-g)$.

Theorem 4.1. *If a canonical maximization linear programming problem has an optimal solution, then the dual canonical minimization linear programming problem has an optimal solution and vice versa. Furthermore, dual canonical linear programming problems with optimal solutions have $f = g$ at these solutions.*

Proof. Assume that we have a pair of dual canonical linear programming problems recorded in a dual canonical tableau. Note that any single pivot transformation transforms the maximization problem into an equivalent maximization problem having the same feasible solutions as the original and transforms the minimization problem into an equivalent minimization problem having the same feasible solutions as the original. Assume that the maximization problem is not infeasible or unbounded. If the simplex algorithm is applied to this maximization problem and we make sure that variables corresponding to pivot entries are interchanged in the dual minimization problem as well as the maximization problem, the algorithm terminates in a basic optimal solution for the maximization problem with the tableau in the form

	$($ $ind.$ $var's$ $)$	-1	
$(ind.var's)$		≥ 0 ≥ 0 $\vdots$ ≥ 0	$= -(dep.var's)$
-1	≤ 0 ≤ 0 $\ldots$ ≤ 0	d	$= f$
	$= ($ dep $var's$ $)$	$= g$	

But the basic solution of the minimization problem is also optimal since the dual simplex algorithm terminates with the tableau in the same form.Hence the existence of an optimal solution for a canonical maximization linear programming problem implies the existence of an optimal solution for its dual canonical minimization linear programming problem.

Now to prove the converse assume that we have a pair of dual canonical linear programming problems recorded in a dual canonical tableau and assume that the minimization problem is not infeasible or unbounded. If the dual simplex algorithm is applied to this minimization problem and we make sure that variables corresponding to pivot entries are interchanged in the dual maximization problem as well as the minimization problem, the algorithm terminates in a basic optimal solution for the minimization problem with the tableau in the form above. But the basic solution of the maximization problem is also optimal since the simplex algorithm for maximization also terminates with the tableau in the same form. Hence the existence of an optimal solution for a canonical minimization linear programming problem implies the existence of an optimal solution for its dual canonical maximization linear programming problem.

Hence, the existence of an optimal solution for one canonical linear programming problem implies the existence of an optimal solution for its dual linear programming problem.

Now assume that we have a pair of optimal solutions of dual canonical linear programming problems. Since all optimal solutions of a linear programming problem yield the same objective function value (otherwise, some of the solutions would not be optimal) and since one such pair of optimal solutions for the maximization and minimization problems arises from the basic solutions of a dual canonical tableau in the form above, we have $f = g = d$. □

Remarks 4.1.

As an additional consequence of the agreement of the tableau forms that terminate both the simplex algorithm of Chapter 2 and the dual simplex algorithm of this chapter in optimal solutions, we note that canonical minimization linear programming problems with optimal solutions can be solved directly by applying the simplex algorithm of Chapter 2 to the dual canonical maximization linear programming problem-no negative transposition is required. Two linear programming problems are being solved for the price of one.

Example 4.1.

Solve the dual canonical linear programming problems below:

	x_1	x_2	-1	
y_1	20	25	300	$= -t_1$
y_2	40	20	500	$= -t_2$
-1	1000	800	0	$= f$
	s_1	s_2	$= g$	

We apply the simplex algorithm of Chapter 2 to the maximization linear programming problem, making sure that variables corresponding to pivot entries are interchanged in the dual minimization problem as well as the maximization problem.

	x_1	x_2	-1	
y_1	20	25	300	$= -t_1$
y_2	40^*	20	500	$= -t_2$
-1	1000	800	0	$= f$
	s_1	s_2	$= g$	

	t_2	x_2	-1	
y_1	$-1/2$	15^*	50	$= -t_1$
s_1	$1/40$	$1/2$	$25/2$	$= -x_1$
-1	-25	300	-12500	$= f$
	y_2	s_2	$= g$	

	t_2	t_1	-1	
s_2	$-1/30$	$1/15$	$10/3$	$= -x_2$
s_1	$1/24$	$-1/30$	$65/6$	$= -x_1$
-1	-15	-20	-13500	$= f$
	y_2	y_1	$= g$	

The basic solution of each linear programming problem in the final tableau above is optimal:

$$t_2 = t_1 = 0, x_2 = 10/3, x_1 = 65/6, \max f = 13500$$

$$s_2 = s_1 = 0, y_2 = 15, y_1 = 20, \min g = 13500$$

4.6 Matrix Formulation of Canonical Tableaus

Consider the dual canonical tableau

	x_1	x_2	$\dots$	x_n	-1	
y_1	a_{11}	a_{12}	$\dots$	a_{1n}	b_1	$=-t_1$
y_2	a_{21}	a_{22}	$\dots$	a_{2n}	b_2	$=-t_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_m	a_{m1}	a_{m2}	$\dots$	a_{mn}	b_m	$=-t_m$
-1	c_1	c_2	$\dots$	c_n	d	$=f$
	$=s_1$	$=s_2$	$\dots$	$=s_n$	$=g$	

Letting

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \mathbf{B}=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \mathbf{C}=\begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}, \mathbf{D}=\begin{bmatrix} d \end{bmatrix}$$

$$\mathbf{X}=\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}, \mathbf{Y}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \mathbf{S}=\begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix}, \mathbf{T}=\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}$$

We can now reformulate the dual canonical linear programming problems in terms of matrix equations.
Matrix reformulation of canonical maximization linear programming problem is
Maximize

$$f = CX^t - D$$

subject to

$$AX^t - B = -T$$

$$X, T \geq 0$$

Matrix reformulation of canonical minimization linear programming problem is
Minimize

$$g = Y^t B - D$$

subject to

$$Y^t A - C = S$$

$$Y, S \geq 0$$

Here, ^{t} denotes the transpose of a matrix. Also, a matrix being greater than or equal to zero is to be interpreted as every entry of that matrix being greater than or equal to zero.

4.7 The Duality Equation

Theorem 4.2.

(The Duality Equation). For any pair of feasible solutions of dual canonical linear programming problems, we have

$$g - f = X^t + Y^t T$$

.

Proof: From the matrix reformulations of the canonical maximization and the canonical minimization linear programming problems we obtain the equations

$$f = CX^t - D$$

$$B = AX^t + T$$

$$g = Y^t B - D$$

$$C = Y^t A - S.$$

Now

$$
\begin{aligned}
g - f &= (Y^tB - D) - (CX^t - D) \\
&= Y^tB - CX^t \\
&= Y^t(AX^t + T) - (Y^tA - S)X^t \\
&= Y^tAX^t + Y^tT - Y^tAX^t - SX^t \\
&= Y^tT + SX^t \\
&= SX^t + Y^tT.
\end{aligned}
$$

Note that only the matrix forms of the objective function and the main constraints were used in the proof of the duality equation above; the non-negativity constraints were not used. Hence the duality equation is also true for solutions to dual canonical linear programming problems that are infeasible in terms of the non-negativity constraints only, i.e., solutions which satisfy all main constraints but which violate one or more non-negativity constraints. The duality equation does not hold if an infeasible solution violates a main constraint.

Corollary 4.1.

For any pair of feasible solutions of dual canonical linear programming problems, we have $g \geq f$.

Proof: Since the solutions of the dual canonical linear programming problems are feasible, every entry of S, X^t, Y^t, and T is non-negative.
Hence, $SX^t + Y^tT \geq 0$.
Since $g - f = SX^t + Y^tT \geq 0$
by the duality equation, we have $g - f \geq 0$ or $g \geq f$.

Corollary 4.2.

(i) If a canonical maximization linear programming problem is unbounded, then the dual canonical minimization linear programming problem is infeasible.

(ii) If a canonical minimization linear programming problem is unbounded, then the dual canonical maximization linear programming problem is infeasible.

proof

(i) Assume, by way of contradiction, that the maximization linear programming problem is unbounded and that the dual minimization linear programming problem has a feasible solution. By Corollary 4.1, this feasible solution must yield a value for g that is greater than or equal to the value of f for any feasible solution to the maximization problem. But since the maximization problem is unbounded, feasible solutions for this problem exist for which $f \to \infty$.Hence no such feasible solution corresponding to a value for g can exist and the minimization linear programming problem is infeasible.

(ii) The proof is similar to (i).

Note that Corollary 4.2 rules out the possibility of ever encountering dual unbounded canonical linear programming problems.

Corollary 4.3.

Any pair of feasible solutions of dual canonical linear programming problems for which $f = g$ are optimal solutions.

Proof: No other feasible solution can increase the value of f since, if it could, we would have $f > g$, contradicting Corollary 4.1. Similarly, no other feasible solution can decrease the value of g. Hence the given feasible solutions are optimal solutions.

Definition 4.1.

Any pair of feasible solutions of the dual canonical linear programming problems

	x_1	x_2	$\ldots$	x_n	-1	
y_1	a_{11}	a_{12}	$\ldots$	a_{1n}	b_1	$=-t_1$
y_2	a_{21}	a_{22}	$\ldots$	a_{2n}	b_2	$=-t_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_m	a_{m1}	a_{m2}	$\ldots$	a_{mn}	b_m	$=-t_m$
-1	c_1	c_2	$\ldots$	c_n	d	$=f$
	$=s_1$	$=s_2$	$\ldots$	$=s_n$	$=g$	

for which
(i) $x_j \neq 0 \Rightarrow s_j = 0, j = 1, 2, ..., n$ and
(ii) $y_i \neq 0 \Rightarrow t_i = 0, i = 1, 2, ...m$
are said to exhibit *complementary slackness*.

Example 4.2.

Consider the pair of optimal solutions of the dual canonical linear programming problems of Example 4.1. The dual tableau has optimal maximization solution $x_1 = 65/6, x_2 = 10/3, t_2 = t_1 = 0$, $\max f = 13500$,and optimal minimization solution $y_1 = 20, y_2 = 15, s_2 = s_1 = 0$, $\min g = 13500$. Whenever x_j is not equal to zero, s_j is equal to zero and similarly for y_i and t_i Hence complementary slackness is exhibited in these optimal solutions.

Theorem 4.3.

A pair of feasible solutions of dual canonical linear programming problems exhibit complementary slackness if and only if they are optimal solutions.

Proof: Assume that a pair of feasible solutions of dual canonical linear programming problems exhibit complementary slackness.Then
$s_j x_j = 0, j = 1, 2, ..., n$ and $y_i t_i = 0, i = 1, 2, ...m$.

Now

$$\mathbf{SX^t} = \begin{bmatrix} s_1 & s_2 & \dots & s_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \textstyle\sum_{j=1}^{n} s_j x_j = 0$$

and

$$\mathbf{Y^tT} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} = \textstyle\sum_{i=1}^{m} y_i t_i = 0$$

i.e.$SX^t + Y^tT = 0$. But $gf - f = SX^t + Y^tT$ by the duality equation. So $g - f = 0$ i.e $g = f$.
So by Corollary 4.3 the given feasible solutions are optimal solutions.
Now conversely assume that we have a pair of optimal solutions of dual canonical linear programming problems. Then $f = g$ at these solutions by Theorem 4.1. Hence $SX^t + Y^tT = 0$ by the duality equation. Now, since every entry of the matrices S, X^t, Y^t and T must be non-negative, we have $SX^t = Y^tT = 0$.

$$SX^t = \sum_{j=1}^{n} s_j x_j = 0$$

and

$$Y^tT = \sum_{i=1}^{m} y_i t_i = 0$$

Again, since $s_j, x_j, y_i,$ and t_i are non-negative for all i and j,we have that at least one of the factors in every term of each summation must be zero. This is equivalent to complementary slackness.

4.8 The Duality Theorem

Theorem 4.4.

(The Duality Theorem) Given dual canonical linear programming problems, exactly one of the following is true:

(i) Both problems have optimal solutions; for these solutions, $f = g$

(ii) The maximization problem is unbounded and the minimization problem is infeasible

(iii) The minimization problem is unbounded and the maximization problem is infeasible

(iv) Both problems are infeasible.

Proof:

1. Refer proof of Theorem 4.1
2. Refer proof of corollary 4.2
3. Refer proof of corollary 4.2
4. Refer example 4.4

Example 4.3.

Solve the dual canonical linear programming problems below:

	x_1	x_2	-1	
y_1	-1	-1	-3	$= -t_1$
y_2	1	1	2	$= -t_2$
-1	2	-4	0	$= f$
	$= s_1$	$= s_2$	$= g$	

Soltion:

	x_1	x_2	-1	
y_1	-1	-1	-3	$= -t_1$
y_2	1^*	1	2	$= -t_2$
-1	2	-4	0	$= f$
	$= s_1$	$= s_2$	$= g$	

	t_2	x_2	-1	
y_1	1	0	-1	$= -t_1$
s_1	1	1	2	$= -x_1$
-1	-2	-6	-4	$= f$
	$= y_2$	$= s_2$	$= g$	

Clearly the maximization problem is infeasible since $b_1 < 0$ and the corresponding $a_{11} \geq 0$ and $a_{12} \geq 0$. Now to solve the minimization problem as we are using simplex algorithm for maximum tables we convert the minimization problem into maximization by taking the negative transpose. Then we get

y_1	s_1	-1	
-1	-1	2	$= -y_2$
0	-1	6	$= -s_2$
1	-2	4	$= -g$

Clearly the minimization problem is unbounded since $c_1 > 0$ and the corresponding $a_{11} \leq 0$ and $a_{21} \leq 0$.

Example 4.4.

Solve the dual canonical linear programming problems below

	x_1	x_2	-1	
y_1	-1	1	-1	$= -t_1$
y_2	1	-1	-1	$= -t_2$
-1	1	1	0	$= f$
	$= s_1$	$= s_2$	$= g$	

Solution:

	x_1	x_2	-1	
y_1	-1	1	-1	$= -t_1$
y_2	1	-1^*	-1	$= -t_2$
-1	1	1	0	$= f$
	$= s_1$	$= s_2$	$= g$	

	x_1	t_2	-1	
y_1	0	1	-2	$= -t_1$
s_2	-1	-1	1	$= -x_2$
-1	2	1	-1	$= f$
	$= s_1$	$= y_2$	$= g$	

Clearly the maximization problem is infeasible since $b_1 < 0$ and the corresponding $a_{11} \geq 0$ and $a_{12} \geq 0$. Now to solve the minimization problem as we are using simplex algorithm for maximum tables we convert the minimization problem into maximization by taking the negative transpose. Then we get

	y_1	s_2	-1	
	0	1	-2	$= -s_1$
	-1	1	-1	$= -y_2$
	2	-1	1	$= -g$

Clearly the minimization problem is infeasible.
In both examples above, the dual canonical minimization linear programming problem was immediately solved after negative transposition. This will not happen in general-it may be necessary to perform pivots on the transposed tableau.

4.9 Duality in Non Canonical Forms

The duality theory presented in the preceding sections can be extended to accommodate dual non-canonical tableaus corresponding to the non-canonical linear programming problems of Chapter 3. We will be content in this section with giving the form of a dual non-canonical tableau and illustrating, via examples, the solution procedure for reducing it to a dual canonical tableau.

Definition 4.2.

A **dual non-canonical tableau** is a non-canonical tableau of the form

	x_1	$\ldots$	x_j	x_{j+1}	$\ldots$	x_n	-1	
y_1	a_{11}	$\ldots$	a_{1j}	a_{1j+1}	$\ldots$	a_{1n}	b_1	$= -0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_i	a_{i1}	$\ldots$	a_{ij}	a_{ij+1}	$\ldots$	a_{in}	b_i	$= -0$
y_{i+1}	a_{i+11}	$\ldots$	a_{i+1j}	a_{i+1j+1}	$\ldots$	a_{i+1n}	b_{i+1}	$= -t_{i+1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_m	a_{m1}	$\ldots$	a_{mj}	a_{mj+1}	$\ldots$	a_{mn}	b_m	$= -t_m$
-1	c_1	$\ldots$	c_j	c_{j+1}	$\ldots$	c_n	d	$= f$
	$= 0$	$\ldots$	$= 0$	s_{j+1}	$\ldots$	$= s_n$	$= g$	

Note that each unconstrained independent variable in the maximization linear programming problem corresponds to an equation of constraint in the dual minimization linear programming problem and each unconstrained independent variable in the minimization linear programming problem corresponds to an equation of constraint in the dual maximization linear programming problem. These unconstrained variables and 0 slack "variables" lie opposite each other in the tableau. This property is crucial and allows one to solve dual non-canonical linear programming problems by using the techniques developed in Chapter 3.

Example 4.5.

Solve the dual non-canonical linear programming problems below:

	x_1	x_2	x_3	-1	
y_1	1	-1	2	1	$=-0$
y_2	2	0	2	-1	$=-t_1$
y_3	0	1	-1	-1	$=-t_2$
-1	1	-1	3	0	$=f$
	$=0$	$=0$	$=s_1$	$=g$	

We apply the techniques of Chapter 3 to the dual non-canonical tableau, making sure that unconstrained independent variables and 0 slack "variables" are handled accordingly in both problems.

	x_1	x_2	x_3	-1	
y_1	1^*	-1	2	1	$=-0$
y_2	2	0	2	-1	$=-t_1$
y_3	0	1	-1	-1	$=-t_2$
-1	1	-1	3	0	$=f$
	$=0$	$=0$	$=s_1$	$=g$	

In maximization pivot x_1 down and pivot 0 up and in minimization pivot y_1 down and pivot 0 up.

	0	x_2	x_3	-1	
0	1	-1	2	1	$=-x_1$
y_2	-2	2	-2	-3	$=-t_1$
y_3	0	1	-1	-1	$=-t_2$
-1	-1	0	1	-1	$=f$
	$=y_1$	$=0$	s_1	$=g$	

In maximization we file $-x_2 + 2x_3 - 1 = -x_1$. Delete x_1 row and 0 column.In minimization file $-2y_2 + 1 = y_1$. Delete y_1 column and

0 row.

	x_2	x_3	-1	
y_2	2	-2	-3	$=-t_1$
y_3	1^*	-1	-1	$=-t_2$
-1	0	1	-1	$=f$
	$=0$	$=s_1$	$=g$	

In maximization problem pivot x_2 down and in minimization problem pivot 0 up.

	t_2	x_3	-1	
y_2	-2	0	-1	$=-t_1$
0	1	-1	-1	=-x_2
-1	0	1	-1	$=f$
	$=y_3$	$=s_1$	$=g$	

In maximization problem file the equation $t_2-x_3+1=-x_2$; delete x_2 row.In minimization problem delete 0 row.

	t_2	x_3	-1	
y_2	-2^*	0	-1	$=t_1$
-1	0	1	-1	$=f$
	$=y_3$	$=s_1$	$=g$	

Now applying simplex algorithm to the above table we get

	t_1	x_3	-1	
y_3	$-1/2$	0	$1/2$	$=-t_2$
-1	0	1	-1	$=f$
	$=y_2$	$=s_1$	$=g$	

Clearly the maximization problem is unbounded and hence minimization problem is infeasible.

Example 4.6.

Solve the dual non-canonical linear programming problems below:

	x_1	x_2	x_3	-1	
y_1	0	-1	-1	-1	$=-0$
y_2	-1	-3	4	0	$=-t_1$
y_3	-1	2	-3	0	$=-t_2$
-1	-1	0	0	0	$=f$
	$=0$	$==s_1$	$=s_2$	$=g$	

Solution:

	x_1	x_2	x_3	-1	
y_1	0	-1^*	-1	-1	$=-0$
y_2	-1	-3	4	0	$=-t_1$
y_3	-1	2	-3	0	$=-t_2$
-1	-1	0	0	0	$=f$
	$=0$	$=s_1$	$=s_2$	$=g$	

max:pivot 0 up and min: pivot y_1 down

	x_1	0	x_3	-1	
s_1	0	-1	1	1	$=-x_2$
y_2	-1	-3	7	3	$=-t_1$
y_3	-1	2	-5	-2	$=-t_2$
-1	-1	0	0	0	$=f$
	$=0$	$=y_1$	$=s_2$	$=g$	

max:delete 0 column; min: file $-s_1-3y_2+2y_3=y_1$; delete y_1 column

	x_1	x_3	-1	
s_1	0	1	1	$=-x_2$
y_2	-1	7	3	$=-t_1$
y_3	-1^*	-5	-2	$=-t_2$
-1	-1	0	0	$=f$
	$=0$	$=s_2$	$=g$	

max: pivot x_1 down, min: pivot 0 up

	t_2	x_3	-1	
s_1	0	1	1	$=-x_2$
y_2	-1	12	5	$=-t_1$
0	-1	5	2	=-x_1
-1	-1	5	2	$=f$
	$=y_3$	$=s_2$	$=g$	

max:file $-t_2+5x_3-2=-x_1$; delete x_1 row; min:delete 0 row;

	t_2	x_3	-1	
s_1	0	1	1	$=-x_2$
y_2	-1	12^*	5	$=-t_1$
-1	-1	5	2	$=f$
	$=y_3$	$=s_2$	$=g$	

Now, we can do simplex algorithm

	t_2	t_1	-1	
s_1	$1/12$	$-1/12$	$7/12$	$=-x_2$
s_2	$-1/12$	$1/12$	$5/12$	$=-x_3$
-1	$-7/12$	$-5/12$	$-1/12$	$=f$
	$=y_3$	$=s_2$	$=g$	

The basic solution of each linear programming problem in the final table is optimal.

$t_2 = t_1 = 0$, $x_2 = 7/12$, $x_3 = 5/12$, max f =1/12,x_1=-1/12(from second filed equation),
$s_2 = s_1 = 0$, $y_2 = 5/12$, $y_3 = 7/12$, min g =1/12,y_1=-1/12(from first filed equation)

Solved Problems

Problem 4.1.

Max f(x_1,x_2) $= x_1 + x_2$
subject to $x_1 + 2x_2 \leq 4$
$3x_1 + x_2 \leq 6$, $x_1, x_2 \geq 0$. Write the dual canonical minimization linear programming problem and solve using dual table
Solution: Minimize g(y_1,y_2) = 4 y_1 + $6y_2$
subject to y_1 +3y_2 $\geq$ 1
2y_1 + y_2 $\geq$ 1,
y_1,y_2 $\geq$ 0
Dual table is

	x_1	x_2	-1	
y_1	1	2	4	=-t_1
y_2	3	1	6	=-t_2
-1	1	1	0	=f
	=s_1	=s_2	=g	

	x_1	x_2	-1	
y_1	1	2	4	=-t_1
y_2	3*	1	6	=-t_2
-1	1	1	0	=f
	=s_1	=s_2	=g	

	t_2	x_2	-1	
y_1	-1/3	5/3*	2	=-t_1
s_1	1/3	1/3	2	=-x_1
-1	-1/3	2/3	-2	=f
	=y_2	=s_2	=g	

	t_2	t_1	-1	
s_2	-1/5	3/5	6/5	=-x_2
s_1	2/5	-1/5	8/5	=-x_1
-1	-1/5	-2/5	-14/5	=f
	=y_2	=y_1	=g	

x_1=8/5, x_2=6/5 ,t_1=0, t_2=0, Max f = 14/5
y_1=2/5, y_2=1/5, s_2=0,s_1=0, Min g = 14/5

Problem 4.2.

Solve using duality, Max f(x_1, x_2) = x_1
Subject to x_1 + x_2 $\leq$ 1
x_1 - x_2 $\geq$ 1
x_2 - 2x_1 $\geq$ 1
x_1,x_2 $\geq$ 0
Solution
Max f(x_1, x_2) = x_1
Subject to x_1 + x_2 $\leq$ 1
-x_1 + x_2 $\leq$ -1
-x_2 + 2x_1 $\leq$ -1
x_1,x_2 $\geq$ 0

	x_1	x_2	-1	
y_1	1	1	1	=-t_1
y_2	-1	1	-1	=-t_2
y_3	2	-1*	-1	=-t_3
-1	1	0	0	=f
	=s_1	=s_2	=g	

	x_1	t_3	-1	
y_1	3	1	0	=-t_1
y_2	1	1	-2	=-t_2
s_2	-2	-1	1	=-x_2
-1	1	0	0	=f
	=s_1	=y_3	=g	

thus Maximization problem is infeasible.
Now consider only the minimization problem, and take negative transpose,then we get

y_1	y_2	s_2	-1	
-3	-1	2	-1	=-s_1
-1	-1	1	0	=-y_3
0	2	-1	0	=-g

pivot on a_{11}then we get

y_1	y_2	s_2	-1	
-3*	-1	2	-1	=-s_1
-1	-1	1	0	=-y_3
0	2	-1	0	=-g

s_1	y_2	s_2	-1	
-1/3	1/3*	-2/3	1/3	=-y_1
-1/3	-2/3	1/3	1/3	=-y_3
0	2	-1	0	=-g

s_1	y_1	s_2	-1	
-1	3	-2	1	=-y_2
-1	2	-1	1	=-y_3
2	-6	3	-2	=-g

from the table its clear that the given minimization problem is unbounded.
Thus maximization problem is infeasible and minimization problem is unbounded.

Problem 4.3.

Solve the dual canonical linear programming

	x_1	x_2	-1	
y_1	1	-1	-1	=-t_1
y_2	-1	-1	-1	=-t_2
-1	1	-2	0	=f
	=s_1	=s_2	=g	

Solution:

	x_1	x_2	-1	
y_1	1	-1	-1	=-t_1
y_2	-1	-1*	-1	=-t_2
-1	1	-2	0	=f
	=s_1	=s_2	=g	

	x_1	t_2	-1	
y_1	2*	-1	0	=-t_1
s_2	1	-1	1	=-x_2
-1	3	-2	2	=f
	=s_1	=y_2	=g	

	t_1	t_2	-1	
s_1	1/2	-1/2	0	=-x_1
s_2	-1/2	-1/2	1	=-x_2
-1	-3/2	-1/2	2	=f
	=y_1	=y_2	=g	

max f =-2, x_1=0, x_2=1, t_1=0, t_2=0
min g = -2, s_2=0, $0 \leq s_1 \leq 1$, $y_1 = 1/2s_1 + 3/2$, $y_2 = 1/2 - 1/2s_1$

Problem 4.4.

Solve the dual canonical linear programming

	x_1	x_2	-1	
y_1	2	-2	-1	$=-t_1$
y_2	-1	1	-1	$=-t_2$
-1	2	1	0	=f
	$=s_1$	$=s_2$	=g	

Solution:

	x_1	x_2	-1	
y_1	2	-2	-1	$=-t_1$
y_2	-1*	1	-1	$=-t_2$
-1	2	1	0	=f
	$=s_1$	$=s_2$	=g	

	t_2	x_2	-1	
y_1	2	0	-3	$=-t_1$
s_1	-1	-1	1	$=-x_1$
-1	2	3	-2	=f
	$=y_2$	$=s_2$	=g	

Now the maximization problem is infeasible, then by taking negative transposition, we get

y_1	s_2	-1	
0	1	-3	$=-s_1$
-2	-1	1	$=-y_2$
3	1	-1	=-g

Therefore, minimization problem is infeasible.

Problem 4.5.

Solve the dual non canonical linear programming table

	$[x_1]$	x_2	-1	
$[y_1]$	2	-1	-1	=-0
y_2	-1	1	-1	=-t_1
-1	2	1	0	=f
	=0	=s_1	=g	

Solution:

	$[x_1]$	x_2	-1	
$[y_1]$	2*	-1	-1	=-0
y_2	-1	1	-1	=-t_1
-1	2	1	0	=f
	=0	=s_1	=g	

	0	x_2	-1	
0	1/2	-1/2	-1/2	=-$[x_1]$
y_2	1/2	1/2	-3/2	=-t_1
-1	-1	2	1	=f
	=$[y_1]$	=s_1	=g	

Deleting 0 variable's row and column ,we get

	x_2	-1	
y_2	1/2	-3/2	=-t_1
-1	2	1	=f
	=s_1	=g	

From this above table , it is clear that the maximization problem is infeasible
Now consider the minimization problem

y_2	1/2	-3/2
-1	2	1
	=s_1	=g

negative transpose of the table is

y_2	-1	
-1/2	-2	=-s_1
3/2	-1	=-g

s_1	-1	
-2	4	=-y_2
3	-7	=-g

From this above table , it is clear that the minimization problem is unbounded
Therefore, the given maximize table is infeasible and the minimum table is unbounded.

Problem 4.6.

Solve the dual non canonical linear programming problem with unconstrained variables.

	[x_1]	x_2	-1	
[y_1]	1	2	2	=-0
y_2	-1	-2	-2	=-t_1
-1	-1	-2	0	=f
	=s_1	=s_1	=g	

Solution:

	[x_1]	x_2	-1	
[y_1]	1*	2	2	=-0
y_2	-1	-2	-2	=-t_1
-1	-1	-2	0	=f
	=s_1	=s_1	=g	

	0	x_2	-1	
s_1	1	2	2	=-[x_1]
y_2	1	0	0	=-t_1
-1	1	0	2	=f
	=[y_1]	=s_2	=g	

Deleting corresponding column, we get

	x_2	-1	
y_2	0	0	=-t_1
-1	0	2	=f
	=s_2	=g	

Thus $t_1 = 0$,$x_1 \leq 2, x_2 = 1 - 1/2x_1$, max f = - 2
$s_1 = 0,\ y_1 \geq -1, y_2 = y_1 + 1$, min g = -2

Chapter 5

Matrix Games

5.1 Introduction

Life is full of conflict and competition. Numerous examples involving adversaries in conflict include parlour games, military battles, political campaigns, advertising and marketing campaigns by competing business firms, and so forth. A basic feature in many of these situations is that the final outcome depends primarily upon the combination of strategies selected by the adversaries. Game theory is a mathematical theory that deals with the general features of competitive situations like these in a formal, abstract way. It places particular emphasis on the decision-making processes of the adversaries. Because competitive situations are so ubiquitous, game theory has applications in a variety of areas, including in business and economics.By a game we mean roughly a situation of conflict between two or more people, in which each contestant, player, or participant has some, but not total, control over the outcome of the conflict. We assume that all players have complete knowledge of all actions, moves, or choices available to themselves and their opponents, and knowledge of the results of the conflict associated with any given selection of actions. Assuming that each player acts rationally to maximize his or her gain, our basic problem is to develop a theory that will help us to understand and predict human behaviour or economic phenomena.

Definition 5.1.

Two-person, zero-sum games
As the name implies, these games involve only two players (who may be armies, teams, firms, and so on). They are called zero-sum games because one player wins whatever the other one loses, so that the sum of their net winnings is zero.

5.2 Formulation of Two-person, Zero sum game

Example 5.1.

We begin with a typical example.Two players, say an "even" player and an "odd" player, each secretly think of an integer between 1 and 3 inclusive. Both players reveal their numbers simultaneously. If the sum of the numbers is even, the "even" player wins a number of dollars from the "odd" player equal to the difference of the numbers provided that the numbers are distinct. If the numbers are the same (in which case the sum is also even), the "even" player wins a number of dollars from the "odd" player equal to the sum of the numbers. If the sum of the numbers is odd, the "odd" player wins 3 dollars from the "even" player.
Note that, in any round of the game, each player has three choices, a 1, a 2, or a 3. By using a matrix to tabulate all of the possible combinations of choices by the players as well as the payoffs associated with these choices, we can obtain a pay off matrix for the game. This payoff matrix, in terms of winnings for the "even" player (i.e., negative entries in the matrix are interpreted as losses for the "even" player or, equivalently, winnings

for the "odd" player), is given by

$$
\begin{array}{ccc}
 & & \text{"ODD" PLAYER'S CHOICE} \\
 & & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\
\text{"EVEN"} & 1 & \\
\text{PLAYER'S} & 2 & \begin{bmatrix} 2 & -3 & 2 \\ -3 & 4 & -3 \\ 2 & -3 & 6 \end{bmatrix} \\
\text{CHOICE} & 3 &
\end{array}
$$

For example, if the "even" player chooses 1 (first row) and the "odd" player chooses 1 (first column), the "even" player wins 2 dollars from the "odd" player since 1 + 1 = 2. This is recorded as a 2 (since the "even" player wins 2 dollars) in the first row and first column of the payoff matrix. If the "even" player chooses 1 (first row) and the "odd" player chooses 2 (second column), the "odd" player wins 3 dollars from the "even" player since 1 + 2 = 3 is odd. This is recorded as a - 3 (since the "even" player loses 3 dollars) in the first row and second column of the payoff matrix. If the "even" player chooses 1 (first row) and the "odd" player chooses 3 (third column), the "even" player wins 2 dollars from the "odd" player since 3 - 1 = 2. This is recorded as a 2 (since the "even" player wins 2 dollars) in the first row and third column of the payoff matrix. This completes the computation of the entries of the payoff matrix in the first row; the other entries of the payoff matrix are computed similarly.
This is an example of a two-person zero-sum matrix game (hereafter referred to simply as a matrix game).

In a matrix game, we have an m x n payoff matrix, a row player, and a column player. In each round of the game, the row player chooses a row of the payoff matrix and the column player chooses a column of the payoff matrix. These choices are then cross-indexed to find the payoff for the round. In this book, the payoffs in the matrix are always listed as winnings for the row player; winnings for the column player appear as negative entries in the payoff matrix. Hence, the row player wishes to maximize the payoff and the column player wishes to minimize the payoff.

Definition 5.2.

A **strategy** is a predetermined rule that specifies completely how one intends to respond to each possible circumstance at each stage of the game.
A strategy can be a pure strategy or mixed one.When a row player plays one row all the time (or one column in case of column player) then we say that the player is playing a pure strategy game.In a mixed strategy game row player will play each of his row certain portion of time and column player will play each of his column a certain portion of time.Let's look at their formal definitions.

Definition 5.3.

Let $A = [a_{ij}]_{m\times n}$ be an $m \times n$ matrix game. A **mixed (probabilistic) strategy** for the row player is a column vector

$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}$$

such that $p_i \geq 0$ for all i and $\sum_{i=1}^{m} p_i = 1$.
A *mixed (probabilistic) strategy* for the column player is a row vector

$$\mathbf{Q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}$$

such that $q_j \geq 0$ for all j and $\sum_{j=1}^{n} q_j = 1$.
Any mixed strategy containing an entry of 1 (whence all of the other entries are necessarily 0) is said to be a **pure strategy**.

The interpretation of the mixed strategy for the row player is that, if the row player uses strategy P, he will choose row i of the matrix with probability p_i Similarly, the interpretation of the mixed strategy for the column player is that, if the column player uses strategy Q, he will choose column j of the matrix with probability q_j. If a player uses a pure strategy, he will constantly choose the same row or column, namely the row or column corresponding to the probability 1.

Definition 5.4.

An **optimal strategy** for a player of a matrix game, we mean a strategy whereby a player can maximize his winnings or minimize his losses, assuming that the other player will have perfect knowledge of this strategy and also play so as to maximize his winnings or minimize his losses subject to this strategy.

5.3 Domination in a Matrix Game

Whenever one row of a payoff matrix is term-by-term less than or equal to another row, delete the smaller row from the game (since the row player is trying to maximize the outcome). Whenever one column of a payoff matrix is term-by-term greater than or equal to another column, delete the larger column from the game (since the column player is trying to minimize the outcome). Continue deleting rows and/or columns until no row or column "dominates" another.

5.4 LP Formulation of Matrix Games

Let $A = [a_{ij}]_{m\times n}$ be an $m\times n$ matrix game.Assume, for the moment, that the column player always chooses column j of the matrix, i.e., the column player is using the pure strategy

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}$$

If the row player uses mixed strategy

$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}$$

then the expected value of his winnings, denoted $E_j(P)$, is
$E_j(P) = p_1a_{1j} + p_2a_{2j} + \ldots + p_ma_{mj}$
by elementary probability theory. Now the row player's optimal strategy would assure that the expected value of his winnings is maximal no matter what column the column player chooses. Stated a bit differently, the row player's optimal strategy would assure that his minimum expected winnings are as large as possible. Hence, the optimal strategy for the row player is to choose strategy P such that $\min_{1 \leq j \leq n} E_j(P)$ is maximal. Similarly, assume, for the moment, that the row player always chooses row i of the matrix, i.e., the row player is using the pure strategy

$$\mathbf{P} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

If the column player uses mixed strategy

$$\mathbf{Q} = \begin{bmatrix} q_1 & q_2 & \ldots & q_n \end{bmatrix}$$

then the expected value of his losses (remember that matrix game entries are in terms of winnings for the row player!), denoted $F_i(Q)$, is
$F_i(Q) = q_1a_{i1} + q_2a_{i2} + \ldots + q_na_{in}$

by elementary probability theory. Now the column player's optimal strategy would assure that the expected value of his losses is minimal no matter what row the row player chooses. Stated a bit differently, the column player's optimal strategy would assure that his maximum expected losses are as small as possible. Hence, the optimal strategy for the column player is to choose strategy Q such that $\max_{1\leq i\leq m} F_i(Q)$ is minimal.

The expected winnings of the row player per round of the game provided that both players play their optimal strategies is $u = \max_{P} \min_{1\leq j\leq n} E_j(P)$.

The expected losses of the column player per round of the game provided that both players play their optimal strategies is $v = \min_{Q} \max_{1\leq i\leq m} F_i(Q)$.

Theorem 5.1. *Let $A = [a_{ij}]$ be an $m \times n$ matrix game. Then the mixed strategies*

$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}$$

and

$$\mathbf{Q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}$$

obtained from the solution of the dual non-canonical linear programming problems

	v	q_1	q_2	$\dots$	q_n	-1	
u	*0*	-1	-1	$\dots$	-1	-1	$= -0$
p_1	-1	a_{11}	a_{12}	$\dots$	a_{1n}	0	$= -t_1$
p_2	-1	a_{21}	a_{22}	$\dots$	a_{2n}	0	$= -t_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_m	-1	a_{m1}	a_{m2}	$\dots$	a_{mn}	0	$= -t_m$
-1	-1	*0*	*0*	$\dots$	*0*	*0*	$= f$
	$= 0$	$= s_1$	$= s_2$	$\dots$	$= s_n$	$= g$	

are optimal for the row and column player respectively. The dual non canonical tableau above is called the game tableau for A.

Proof. We show that the maximization problem ofthe game tableau yields the optimal strategy for the column player; the proof that the minimization problem yields the optimal strategy for the row player is similar. The first equation of the maximization problem is

$$\sum_{j=1}^{n} q_j = 1$$

along with $q_j \geq 0$ for all j, we have that

$$\mathbf{Q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}$$

is a mixed strategy. The next m equations of the maximization problem are

$$-v + F_i(Q) = -t_i, i = 1, 2, \dots, m$$

; in non-slack form, these m equations become the inequalities

$$F_i(Q) \leq v, i = 1, 2, \dots, m$$

. Finally, the last equation of the maximization problem says to maximize $f = -v$ or, equivalently, to minimize v.Hence, the maximization problem of the game tableau finds the mixed strategy Q so that the maximum value of $F_i(Q) \leq v, i = 1, 2, \dots, m$ is minimal; this is precisely the optimal strategy for the column player. □

Remarks 5.1.

This theorem gives us a procedure for solving a matrix game - having reduced the determination of optimal strategies to dual non-canonical linear programming problems, we simply apply the techniques and theory of Chapter 2, Chapter 3, and Chapter 4 to solve the problem. Notice that this theorem does not guarantee the existence of a pair of optimal solutions for the dual non-canonical linear programming problems arising from a matrix game. Fortunately, an important theorem of game theory called *the Von Neumann minimax theorem* is an existence theorem; that assures the existence of optimal strategies for both players of a matrix game. The statement of the von Neumann minimax theorem is given without proof

Theorem 5.2. ***von Neumann Minimax Theorem.*** *Let $A = [a_{ij}]_{m\times n}$ be an $m \times n$ matrix game. Then there exist optimal mixed strategies P^* and Q^* for the row player and the column player respectively. Furthermore*

$$\begin{aligned}\min_{1\le j\le n} E_j(P^*) &= \max_{P} \min_{1\le j\le n} E_j(P)\\ &= \min_{Q} \max_{1\le i\le m} F_i(Q)\\ &= \max_{1\le i\le m} F_i(Q^*).\end{aligned}$$

This common value is said to be the von Neumann value of the game.

The von Neumann value of a matrix game is the expected winnings of the row player and the expected losses of the column player per round of the game provided that both players play their optimal strategies. A positive Von Neumann value hence indicates that the game favours the row player, a negative von Neumann value indicates that the game favours the column player, and a von Neumann value of 0 indicates that the game is fair.

Chapter 6

Transportation Problems

6.1 Introduction

Transportation and assignment problems are traditional examples of linear programming problems. Although these problems are solvable by using the techniques of Chapters 2-4 directly, the solution procedure is cumbersome; hence, we develop much more efficient algorithms for handling these problems. In the case of transportation problems, the algorithm is essentially a disguised form of the dual simplex algorithm. Assignment problems, which are special cases of transportation problems, pose difficulties for the transportation algorithm and require the development of an algorithm which takes advantage of the simpler nature of these problems.

6.2 Transportation Problem

A manufacturer of a certain good owns m warehouses $W_1, W_2, ...W_m$ and sells to n markets $M_1, M_2, ...M_n$. Let $s_i, i = 1, 2, ...m$, be the supply of W_i, let $d_j, j = 1, 2, ..., n$, be the demand of M_j and let c_{ij} be the unit shipping cost from W_i to M_j, then the general *balanced transportation problem* is

Minimize

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = s_i, i = 1, 2, ...m\}$$

Warehouse constraints

$$\sum_{i=1}^{m} x_{ij} = d_j, j = 1, 2, ...n\}$$

Market constraints

$$x_{ij} \geq 0, \forall i, j$$

Here, total supply is equal to total demand:

$$\begin{aligned} \text{Total supply } &= \sum_{i=1}^{m} s_i = \sum_{i=1}^{m} (\sum_{j=1}^{n} x_{ij}) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \\ &= \sum_{j=1}^{n} (\sum_{i=1}^{m} x_{ij}) = \sum_{j=1}^{n} d_j = \text{ Total demand} \end{aligned}$$

It is, in this sense, the transportation problem is said to be balanced. If total demand and total supply are not equal then it is said to be an unbalanced transportation problem. The relevant balanced transportation tableau is given by

	M_1	M_2	$\dots$	M_n	
W_1	c_{11}	c_{12}	$\dots$	c_{1n}	s_1
W_2	c_{21}	c_{22}	$\dots$	c_{2n}	s_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
W_m	c_{m1}	c_{m2}	$\dots$	c_{mn}	s_m
	d_1	d_2	$\dots$	d_n	$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$

The entries of the transportation tableau are called **cells**.

6.3 The Vogel Advanced-Start Method

The Vogel Advanced-Start Method (VAM), named after *W.R. Vogel* implements the reduction of the minimum Tucker tableau of a transportation problem to minimum basic feasible form. This algorithm would hence correspond to the transformation of the Tucker tableau to canonical form and the subsequent application of steps of the dual simplex algorithm of Chapter 4. We should remark here that VAM is not the only algorithm that implements this reduction. Other methods include, for example, the minimum-entry method and the north west-corner method.In fact, the use of VAM in large transportation problems can be quite cumbersome. But V AM is usually considered to be superior for smaller transportation problems and is especially suitable for hand computations since it generally results in a feasible solution that is closer to being optimal than the other procedures.

VAM Steps

(0) Given: An initial balanced transportation tableau.

(1) Compute the difference of the two smallest entries in every row and column of the tableau and write this difference opposite the row or column. (If there is only one entry in any row or column, write that entry.)

(2) Choose the largest difference and use the smallest cost in the corresponding row or column to empty a warehouse or completely fill a market demand. (If there is a tie for the largest difference, use the smallest entry in the corresponding rows and/or columns. If there is a tie for the smallest entry, use any such entry.) Circle the cost used and write above the circle the amount of goods shipped by that route. Reduce the supply and demand in the row and column containing the cost used.

(3) Delete the row or column corresponding to the emptied warehouse or fully supplied market; if both happen simultaneously, delete the row unless that row is the only row remaining in which case delete the column.

(4) If all tableau entries are deleted, STOP; otherwise, go to (1).

Example 6.1.

A manufacturer of widgets owns three warehouses and sells to three markets. The supply of each warehouse, the demand of each market, and the shipping cost per ton of widgets from each warehouse to each market are as follows:

	Market 1	Market 2	Market 3	warehouse supplies ↓	
Warehouse 1	\$2/ton	\$1/ton	\$2/ton	40 tons	
Warehouse 2	\$9/ton	\$4/ton	\$7/ton	60 tons	
Warehouse 3	\$1/ton	\$2/ton	\$9/ton	10 tons	
market demands →	40 tons	50 tons	20 tons	110 tons	← total demand
				↑ total supply	

How should the manufacturer ship the widgets so as to minimize total transportation cost?
Let's use VAM to the above problem.
(0)

	M_1	M_2	M_3	
W_1	2	1	2	40
W_2	9	4	7	60
W_3	1	2	9	10
	40	50	20	110

The W_i's and the M_j's will be suppressed computationally.
(1)

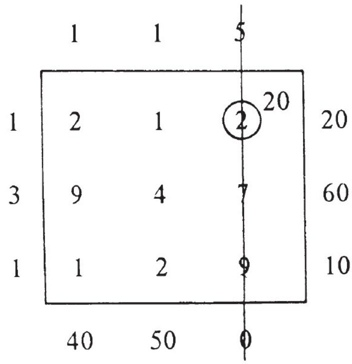
	1	1	5	← column differences
	M_1	M_2	M_3	
1	2	1	2	40
3	9	4	7	60
1	1	2	9	10
↑ row differences	40	50	20	110

(2) The largest difference in the tableau above is 5 corresponding to the third column. Hence we use the smallest cost in the third column, namely 2, to empty a warehouse or completely fill a market demand. 2 denotes the unit cost of shipping widgets from $W_1$to M_3; since M_3 only needs 20 tons of the possible 40 tons in W_1,we ship 20 tons of widgets from W_1 to M_3 and therefore fulfil the demand of M_3.We then adjust the supply of W_1 and the demand of M_3 accordingly, i.e., after such a shipment, the current supply of W_1 is $40-20=20$ and the current demand ofM_3 is $20-20=0$. The entire transaction is recorded as follows:

	1	1	5	
1	2	1	2^{20}	~~40~~ 20
3	9	4	7	60
1	1	2	9	10

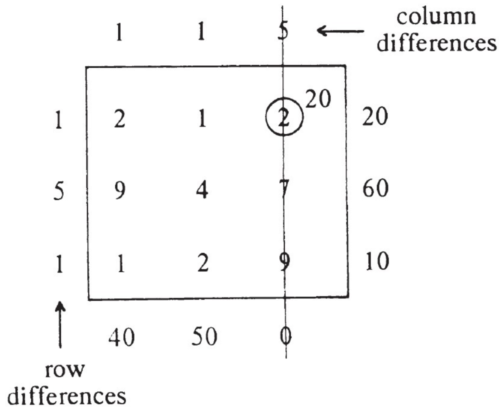
What is the rationale behind the choice of the largest difference and the subsequent choice of the smallest cost in the corresponding row or column? The largest difference at any stage of VAM is a measure of the "regret" we would have for not using the smallest cost possible in the row or column of this difference. Referring to the tableau above, if we do not take advantage of the 2 in the third column, then, because of the larger difference, we will eventually have to use a much larger cost to fulfil the demand of M_3. Hence we ship as many tons of widgets as we can via the cost of 2 in the hope of avoiding the much larger costs in the same column.

(3) M 3 has been fully supplied in step (2) above; hence we delete the third column of the tableau:

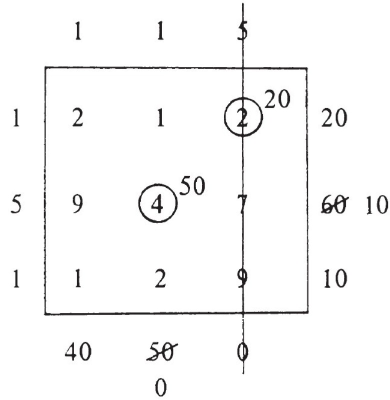


(4) All tableau entries have not been deleted so we go to step (1).

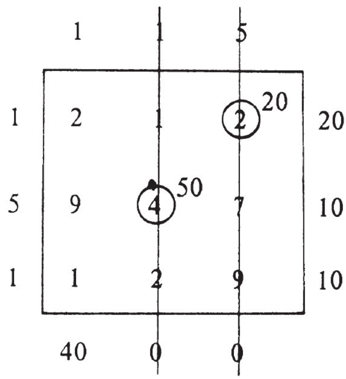
(1)



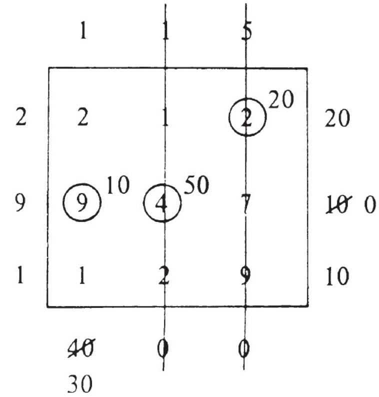
(2) The largest difference in the tableau above is 5 corresponding to the second row. Hence we use the smallest cost in the second row, namely 4, to ship as many tons of widgets as we can from W_2 toM_2 - this amount is 50 tons (fulfilling the demand of M_2)' The supply of W2 is adjusted to 60 - 50 = 10 and the demand of M_2 is adjusted to 50 - 50 = 0;



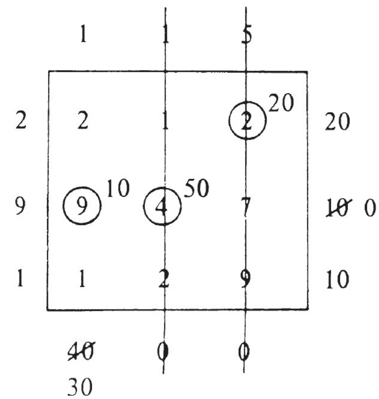
(3)M_2 has been fully supplied in step (2) above; hence we delete the second column of the tableau:



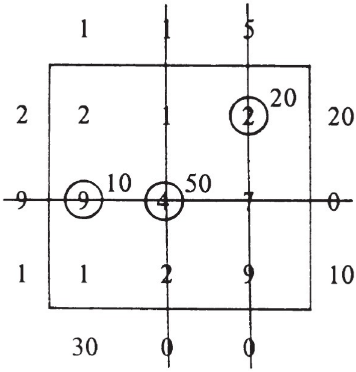
(4) All tableau entries have not been deleted so we go to step (1).



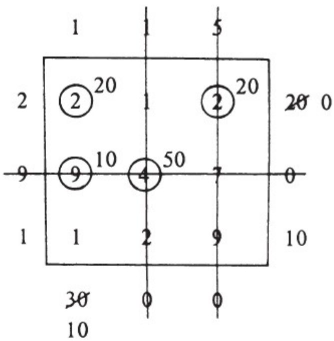
(Remember: If a row or column contains only one entry, write that entry when computing differences.) (2) The largest difference in the tableau above is 9 corresponding to the second row. Hence we use the smallest cost in the second row, namely 9, to ship as many tons of widgets as we can from W_2 to M_1 - this amount is 10 tons (emptying W_2). The supply of W_2 and the demand of M_1 are adjusted accordingly:

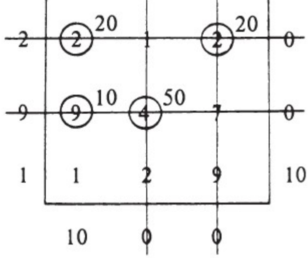


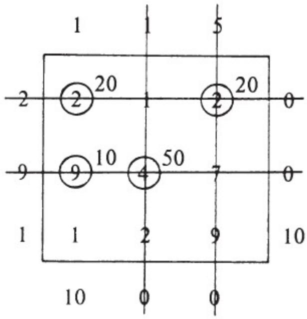
(3) W2 has been emptied in step (2) above; hence we delete the second row of the tableau:

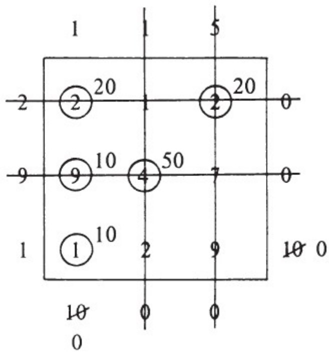


Again by same arguments we can prepare the following tables.

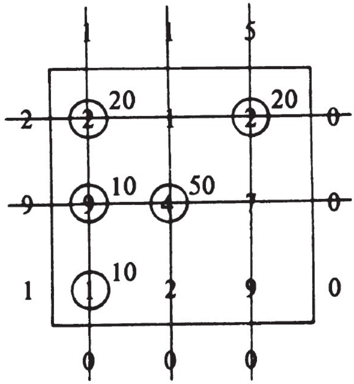








(3) A warehouse has been emptied and a market has been fully supplied simultaneously in step (2) above. Although it really does not matter at this point whether we delete the row or column (since all tableau entries will have been deleted in any event and YAM will terminate in step (4)), we delete the column in accordance with step (3) as stated previously:



(4) All tableau entries have been deleted-STOP. Our transportation tableau after VAM is

2 ^{20}	1	2 ^{20}	40
9 ^{10}	4 ^{50}	7	60
1 ^{10}	2	9	10

Note that the solution obtained by VAM is a feasible solution. To see this, check that the total number of tons of widgets shipped from each warehouse is equal to the total supply of that warehouse and that the total number of tons of widgets shipped to each market is equal to the total demand of that market.This feasible solution is given by

$x_{11} = 20, x_{12} = 0, x_{13} = 20, x_{21} = 10, x_{22} = 50, x_{23} = 0, x_{31}, x_{32} = 0, x_{33} = 0$

with corresponding total transportation cost

$C = 2(20) + 2(20) + 9(10) + 4(50) + 1(10) = 380$

Definition 6.1.

A feasible solution of a balanced transportation problem is said to be a **basic feasible solution** if at most $m+n-1$ of the x_{ij}'s are positive where m is the number of warehouses and n is the number of markets.

Theorem 6.1.

VAM produces a basic feasible solution for any balanced transportation problem. Furthermore, the basic feasible solution corresponds to exactly $m+n-1$ distinguished (circled) cells of the transportation tableau where m is the number of warehouses and n is the number of markets. These distinguished cells are said to constitute a basis for the basic feasible solution.

Remarks 6.1.

VAM will always terminate with exactly $m+n-1$ circled cells where m is the number of warehouses and n is the number of markets. Theorem 6.1 does not say that exactly $m+n-1$ of the x_{ij}'s will be positive-even though exactly $m+n-1$ of the cells will be circled (and all non-circled cells will have x_{ij}-value 0), it is possible for one (or more) circled cells to have x_{ij}-value 0.

6.4 The Transportation Algorithm

Definition 6.2.

Let T be the tableau of a balanced transportation problem. A cycle C in T is a subset of cells of T such that each row and each column of T contains exactly zero or two cells of C.

Example 6.2.

In each table T below, the circled cells form a cycle C in T.

$$\begin{array}{|ccc|}\hline c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ \hline\end{array}$$

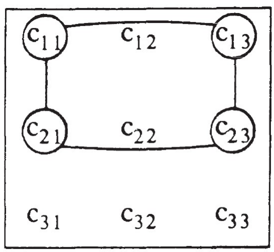
$$\begin{array}{|ccc|}\hline c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ \hline\end{array}$$

$$\begin{array}{|ccc|}\hline c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ \hline\end{array}$$

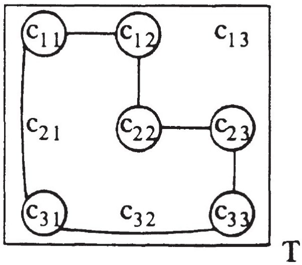
By connecting the cells of a cycle using horizontal and vertical movement only, we may visualize cycles in transportation tableaus as usual graph theoretic cycles. Never use diagonal movement in this regard (even though it may yield a graph-theoretic cycle).

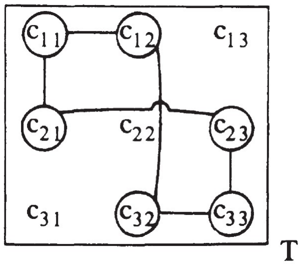
Definition 6.3.

The cycles of the tableaus in Example 5 above may be visualized as

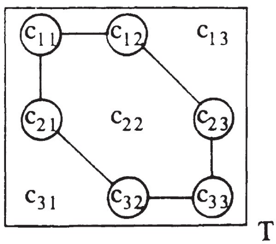


T





respectively. The cycle in the final tableau above should not be visualized as



since diagonal movement is used here.

The Transportation Algorithm

(0) Given: An initial balanced transportation tableau.
(1) Apply V AM to obtain a basic feasible solution and a corresponding basis.
(2) Let $b_1 = 0$.Determine $a_1, a_2, ..., a_m, b_1, b_2, ...b_n$ uniquely

such that $a_i + b_j = c_{ij}$ for all basis cells c_{ij}.
(3) Replace each cell c_{ij} by $c_{ij} - a_i - b_j$; these are the new cells c_{ij} .
(4) If $c_{ij} = 0$ for all i and j, STOP; replace all cells with their original costs from (0); the basic feasible solution given by the current basis cells is optimal. Otherwise, continue.
(5) Choose $c_{ij} < 0$.(Whenever there is more than one possible choice of negative c_{ij} in step (5) of the transportation algorithm, choose the north west-most negative c_{ij}, i.e., choose the negative c_{ij} with minimal i and, if more than one such c_{ij} has minimal i, choose the c_{ij} among those cells with minimal j.)Label this cell as a "getter"cell (+). (By convention, this cell is distinguished by squaring it instead of circling it.) Find the unique cycle C in the tableau determined by this (squared) cell and basis cells. Label the cells in C alternately as "giver"cells (-) and "getter"cells (+).
Choose the "giver"cell associated with the smallest amount of goods. (If there is a tie among certain "giver"cells for the smallest amount of goods, choose any such cell.)
(6) Add the squared cell of (5) to the basis, i.e., circle it in a new tableau. Remove the chosen "giver"cell of(5) from the basis, i.e., do not circle it in a new tableau. Add the amount of goods given up by this "giver"cell to all amounts of goods of "getter"cells in C; subtract the amount of goods given up by this "giver"cell from all amounts of goods of "giver"cells in C. Go to (2).
Before we illustrate the transportation algorithm, we make two remarks. First of all, balanced transportation problems, by their very nature, are never infeasible or unbounded. VAM produces an initial feasible solution ruling out infeasibility; the constraints force the x_{ij}'s to have finite upper bounds ruling out unboundedness. Secondly, recall that VAM applied to any balanced transportation problem results in exactly $m + n - 1$ basis cells. If this basic feasible solution is not optimal, steps (5) and (6) then determine a new basis for a new basic feasible solution-one of the

old basis cells leaves the basis (namely the "giver" cell associated with the smallest amount of goods) and a new cell enters the basis (namely the squared cell). If all cells are replaced with their original costs, this new basic feasible solution generally has lower total transportation cost than the original VAM basic feasible solution.If this new basic feasible solution is not optimal, then a second "sweep" through the transportation algorithm will be necessary to try to improve the solution further by determining another new basis for another new basic feasible solution. This process of constructing basic feasible solutions which successively improve the objective function is completely analogous to the procedure used by the simplex algorithm and usually terminates with an optimal solution.

Example 6.1 (continued)

(0), (1) These steps have already been performed in §2. The tableau obtained by VAM is

2^{20}	1	2^{20}	40
9^{10}	4^{50}	7	60
1^{10}	2	9	10
40	50	20	

Since the warehouse supplies and market demands are no longer needed, we suppress these quantities hereafter.

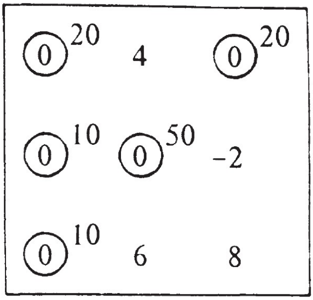
(2) Given $b_1 = 0$ we wish to find a_1, a_2, a_3, b_2 and b_3 uniquely such that $a_i + b_j = c_{ij}$ for all basis cells c_{ij}.

	$b_1(=0)$	b_2	b_3
a_1	2^{20}	1	2^{20}
a_2	9^{10}	4^{50}	7
a_3	1^{10}	2	9

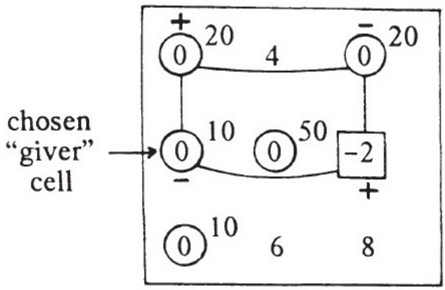
Now
$a_1 + b_1 = c_{11} = 2 \implies a_1 = 2$
$a_2 + b_1 = c_{21} = 9 \implies a_2 = 9$
$a_3 + b_1 = c_{31} = 1 \implies a_3 = 1$
$a_2 + b_2 = c_{22} = 4 \implies b_2 = -5$
$a_1 + b_3 = c_{13} = 2 \implies b_3 = 0.$
Hence we have

	0	-5	0
2	2^{20}	1	2^{20}
9	9^{10}	4^{50}	7
1	1^{10}	2	9

(3) Every cell c_{ij} gets replaced by a new cost, namely the old cost less the a_i indexing the row of the cost less the b_j indexing the column of the cost. Note that all basis cells will necessarily have cost 0 after this replacement; for basis cells, the a_i's and b_j's satisfy $a_i + b_j = c_{ij}$ forcing $c_{ij} - a_i - b_j = 0$ for these cells. The new cells are given below:



(4) Since $c_{23} < 0$, we continue with the transportation algorithm.



Note that the unique cycle must involve the squared cell but, in general, will not involve all of the current basis cells.
Here, c_{23} (the squared cell of step (5)) has entered the basis and c_{21} (the chosen "giver"cell of step (5)) has left the basis. In addition, goods have been redistributed around the cycle of step (5) the "getter"cells c_{11} and c_{23} have increased their amounts by 10 and the "giver"cells c_{13} and c_{21} have decreased their amounts by 10. c_{21} leaves the basis, we do not record the superscripted 0 amount of goods here. Go to step (2).

20+10 20−10 0+10

0^{30}	4	0^{10}
0	0^{50}	-2^{10}
0^{10}	6	8

	(b_1) 0	(b_2) 2	(b_3) 0
(a_1) 0	0^{30}	4	0^{10}
(a_2) −2	0	0^{50}	-2^{10}
(a_3) 0	0^{10}	6	8

0^{30}	2	0^{10}
2	0^{50}	0^{10}
0^{10}	4	8

(4) Since $c_{ij} \geq 0$ for all i and j, we STOP. Replacing all cells with their original costs from step (0), we obtain

2^{30}	1	2^{10}
9	4^{50}	7^{10}
1^{10}	2	9

and corresponding optimal solution
$x_{11} = 30, x_{12} = 0, x_{13} = 10,$
$x_{21} = 0, x_{22} = 50, x_{23} = 10,$
$x_{31} = 10, x_{32} = 0, x_{33} = 0,$
$\min C = 2(30) + 2(10) + 4(50) + 7(10) + 1(10) = 360.$

6.5 Unbalanced Transportation Problems

In this section, we consider transportation problems with tableaus as follows:

	M_1	M_2	$\dots$	M_n	
W_1	c_{11}	c_{12}	$\dots$	c_{1n}	s_1
W_2	c_{21}	c_{22}	$\dots$	c_{2n}	s_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
W_m	c_{m1}	c_{m2}	$\dots$	c_{mn}	s_m
	d_1	d_2	$\dots$	d_n	$\sum_{i=1}^{m} s_i \neq \sum_{j=1}^{n} d_j$

Such transportation problems are said to be unbalanced. We consider two cases.

Case I: $\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j$

Here, the current demand of the markets exceeds the current supply of the warehouses. This is a rationing problem where goods must be allocated among the markets. We introduce a fictitious warehouse W_{m+1} with supply s_{m+1} such that $\sum_{i=1}^{m+1} s_i = \sum_{j=1}^{n} d_j$ i.e., the $(m+1)^{st}$ warehouse supplies the excess demand. This creates a balanced transportation problem. When a market receives goods from the fictitious warehouse, it doesn't really receive any goods at all. The transportation cost of such non-shipment will be assumed negligible compared to other transportation costs and assigned a value of 0. (In reality, the assigned cost would reflect, for example, loss in sales as well as costs associated with expediting the shipment of the goods from another source.)

Case II: $\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$

Here, the current supply of the warehouses exceeds the current demand of the markets. This is usually the case in well-managed inventory situations. We introduce a fictitious market M_{n+1} with

demand d_{n+1} such that $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n+1} d_j$
i.e., the $n+1^{st}$ market demands the excess inventory. This creates a balanced transportation problem. When a warehouse is instructed to ship goods to the fictitious market, it "ships to itself", i.e., it retains the goods. The transportation cost of such self-shipment will be assumed negligible compared to other transportation costs and assigned a value of 0. (In reality, the assigned cost would include, for example, spoilage costs (if the good is perishable) and storage costs.

Example 6.3.

Solve the transportation problem below:

	M_1	M_2	M_3	
W_1	2	1	2	40
W_2	9	4	7	60
W_3	1	2	9	10
	50	60	30	

Note that the transportation problem is unbalanced since
$110 = \sum_{i=1}^{3} s_i < \sum_{j=1}^{3} d_j$=140
Always check this-the transportation algorithm only works with balanced transportation problems! Since demand exceeds supply (Case I), we add a fictitious warehouse W_4 (with associated transportation costs of 0) to supply the excess demand of 30:

	M_1	M_2	M_3	
W_1	2	1	2	40
W_2	9	4	7	60
W_3	1	2	9	10
W_4	0	0	0	30
	50	60	30	

The transportation algorithm is now applied to this balanced transportation problem.It is easy to verify that VAM yields the basic feasible solution.

2^{40}	1	2^{0}	40
9	4^{60}	7	60
1^{10}	2^{0}	9	10
0	0	0^{30}	30
50	60	30	

and that such a shipping schedule is optimal(perform this verification).In this optimal solution, notice that M_3 does not receive any goods since it receives all of its 30 units from the fictitious warehouse W_4.

Example 6.4.

Solve the transportation problem below:

	M_1	M_2	M_3	
W_1	2	1	2	50
W_2	9	4	7	70
W_3	1	2	9	20
	40	50	20	

Note that the transportation problem is unbalanced since
$140 = \sum_{i=1}^{3} s_i < \sum_{j=1}^{3} d_j$=110
Again, always check this-the transportation algorithm only works with balanced transportation problems! Since supply exceeds demand (Case II), we add a fictitious fourth market M4 (with associated transportation costs of 0) to demand the excess inventory

of 30:

	M_1	M_2	M_3	M_4	
W_1	2	1	2	0	50
W_2	9	4	7	0	70
W_3	1	2	9	0	20
	40	50	20	30	

The transportation algorithm is now applied to this balanced transportation tableau. As an exercise, verify that VAM yields the basic feasible solution.

2^{20}	1^{10}	2^{20}	0	50
9	4^{40}	7	0^{30}	70
1^{20}	2	9	0	20
40	50	20	30	

and that such a shipping schedule is optimal. In this optimal solution, notice that W_2 retains 30 units of the good since it has been instructed to ship 30 units to the fictitious fourth market M_4.

The Northwest-Corner Method

(0) Given: An initial balanced transportation tableau.

(1) Use the north west-most cost in the tableau to empty a warehouse or completely fill a market demand. (The north west-most cost is that cost in the top left position of the tableau.) Circle the cost used and write above the circle the amount of goods shipped by that route. Reduce the supply and demand in the row and column containing the cost used.

(2) Delete the row or column corresponding to the emptied warehouse or fully supplied market; if both happen simultaneously,

delete the row unless that row is the only row remaining in which case delete the column.
(3) If all tableau entries are deleted, STOP; otherwise go to (1).

Problem 6.1.

Solve the table using North-West Corner Method

	M1	M2	M3	
W1	2	1	2	40
W2	9	4	7	60
W3	1	2	9	10
	40	50	20	110

Solution: In the given problem the north west corner cell $c_{1,1}$=2 and the demand and supply corresponding to this cell is same. Hence allocate 40 to cell and delete corresponding row and column

	M1	M2	M3	
W1	2 40	1	2	40
W2	9	4	7	60
W3	1	2	9	10
	40	50	20	110

	M2	M3	
W2	4	7	60
W3	2	9	10
	50	20	

From the new table consider the north east corner cell $c_{2,2}$=4 and the demand and supply corresponding to this cell is 50 and 60 respectively. The minimum of these values is 50. Hence allocate 50 to cell and subtract the same from the supply and demand values of the cell$c_{2,2}$. Hence delete corresponding column,

	M2	M3	
W2	4 50	7	60
W3	2	9	10
	50	20	

	M3	
W2	7	10
W3	9	10
	20	

From the new table consider the north east corner cell $c_{2,3}$=7 and the demand and supply corresponding to this cell is 20 and 10 respectively. The minimum of these values is 10. Hence allocate 10 to cell and subtract the same from the supply and demand values of the cell$c_{2,3}$. Hence delete corresponding row,

	M3	
W2	7 10	10
W3	9	10
	20	

Finally assign 10 to the cell $c_{3,3}$=9.

	M1	M2	M3	
W1	2 40	1	2	40
W2	9	4 50	7 10	60
W3	1	2	9 10	10
	40	50	20	110

Hence the initial feasible solution has been obtained
Total cost = $(40\times2)+(50\times4)+(10\times7)+(10\times9) = 440$ (Instead of deleting rows and columns we can also cross the corresponding rows and columns).

The Minimum-Entry Method

(0) Given: An initial balanced transportation tableau.

(1) Use the smallest cost in the tableau to empty a warehouse or completely fill a market demand. (If there is a tie for the smallest entry, use any such entry.) Circle the cost used and write above the circle the amount of goods shipped by that route. Reduce the supply and demand in the row and column containing the cost used.
(2) Delete the row or column corresponding to the emptied warehouse or fully supplied market; if both happen simultaneously, delete the row unless that row is the only row remaining in which case delete the column.
(3) If all tableau entries are deleted, STOP; otherwise go to (I).

Problem 6.2.

Solve the given table by minimum entry method

	M1	M2	M3	
W1	2	1	2	40
W2	9	4	7	60
W3	1	2	9	10
	40	50	20	110

Solution:
The minimum cost of the given table is 1 and it is in two cells $c_{1,2}$ and $c_{3,1}$.Choose $c_{1,2}$ where demand is 50 and supply is 40. So maximum possible allocation is 40.Now cross off row and decrease the demand to be 50 - 40 =10.
Now, the next lowest uncrossed cost is 1 in cell $c_{3,1}$ where the demand 40 and supply is 10. So maximum possible allocation is 10. Now cross off row 3 and decrease the demand to be 30.
Next the lowest uncrossed cost is 4 in cell $c_{2,2}$ where the demand 50 and supply is 60. So maximum possible allocation is 50. Now cross off column 2 and decrease the supply by 10.
Next the lowest uncrossed cost is 7 in cell $c_{2,3}$ where the demand

20 and supply is 10. So maximum possible allocation is 10. Now cross off column 2 and decrease the supply by 10.
Next the lowest uncrossed cost is 9 in cell $c_{2,1}$ where the demand 30 and supply is 300. So maximum possible allocation is 30.

	M1	M2	M3	
W1	2	1 40	2	40
W2	9 30	4 10	7 20	60
W3	1 10	2	9	10
	40	50	20	110

Hence the initial solution is
Total cost = (1 x 40) + (1 x 10) + (4 x 10) + (7 x 20) + (9 x 30) = 500

Problem 6.3.

Find the initial solution for the given table using North West Corner Method and Minimum Entry Method.

7	2	4	10
10	5	9	20
7	3	5	30
20	10	30	60

Solution:

7 10	2	4	10
10	5	9	20
7	3	5	30
20	10	30	60

The North West corner cell $c_{1,1}$= 7,allocate 10 to this cell and delete row 1 and reduce 20 by 10.

10 10	5	9	20
7	3	5	30
10	10	30	60

Now, the next North west corner cell is $c_{2,1}$=10, allocate 10 to this cell and delete column 1 and reduce 20 by 10

5 10	9	10
3	5	30
10	30	60

Now, the next North west corner cell is $c_{2,2}$=5, allocate 10 to this cell and delete column 2 and row 2.

5 30	30
30	60

Finally we allocate 30 to the cell$c_{3,3}$. Hence the initial feasible solution has been obtained by
Total cost $= (7 \times 10) + (10 \times 10) + (5 \times 10) + (5 \times 30) = 370$
ii) Solution using Minimum Entry Method
The minimum cost of the given table is 2, allocate 10 to this cell and cross off the row 1 and column 2,
Next, lowest uncrossed cost is 5 in cell $c_{3,3}$, allocate 30 to this cell and cross off the row 3 and column 3 since the values are same.
Finally the minimum cost is 10 in cell $c_{2,1}$, allocate 20 to this cell.

7	2 10	4	10
10 20	5	9	20
7	3	5 30	30
20	10	30	60

Thus the initial feasible solution is
Total cost $(2 \times 10) + (5 \times 30) + (10 \times 20) = 370$

Problem 6.4.

Solve the transportation problem

7	2	4	10
10	5	9	20
7	3	5	30
20	10	30	60

Here we are using VAM method
The Column difference and row difference of the table is given below.

	0	1	1	← column difference
2	7	2	4	10
4	10	5	9	20
2	7	3	5	30
↑ row difference	20	10	30	60

The highest difference in the above is 4 which corresponds to second row and the minimum cost in the second column is 5.The supply and demand is 20 and 10. Choose the minimum and here it is 10, so assign 10 to 5 and we can cross off second column and change the second row (supply) value as $20 - 10 = 10$. Again we have to find row difference and column difference,then we get,

	0		1	
3	7	2	4	10
1	10	5^{10}	9	10
2	7	3	5	30
	20	0	30	

Repeat the same step as before,then we assign 10 to 4 in cell $c_{1,3}$ and cross off the first row

	0		1	
3	7	2	4^{10}	0
1	10	5^{10}	9	10
2	7	3	5	30
	20	0	20	

Repeat the same step , then we assign 20 to 5 in cell $c_{3,3}$ then we cross off the third column

	3		4	
	7	2	4^{10}	0
1	10	5^{10}	9	10
2	7	3	5^{20}	10
	20	0	0	

	3			
	7	2	4^{10}	0
10	10^{10}	5^{10}	9	0
7	7	3	5^{20}	10
	10	0	0	

Finally, we get

7	2	4^{10}
10^{10}	5^{10}	9
7^{10}	3	5^{20}

To get optimal solution Set $b_1 = 0$ then using relation $a_i+b_j = c_{i,j}$ then we get $a_2 = 10, a_3 = 7, a_1 = 6, b_2 = -5, b_3 = -2$

$\downarrow a_i \ddots \rightarrow b_j$	0	-5	-2
6	7	2	4^{10}
10	10^{10}	5^{10}	9
7	7^{10}	3	5^{20}

Every cell cost is replaced by $c_{i,j} - a_i - b_j$ then we get

1	1	0^{10}
0^{10}	0^{10}	1
0^{10}	1	0^{20}

Since all $c_{i,j}$ are non-negative we attain optimal solution. Replacing all cells, by its original cost. Therefore Total cost $= (4 \times 10) + (10 \times 10) + (5 \times 10) + (7 \times 10) + (5 \times 20) = 360$

Problem 6.5.

Solve the transportation problem

8	2	3	7	42
9	4	5	6	17
7	1	6	5	17
9	14	24	29	

Solution:
VAM Method to find initial feasible solution
First we have find the row difference and column difference of the given problem

	1	1	2	1	
1	8	2	3	7	42
1	9	4	5	6	17
4	7	1	6	5	17
	9	14	24	29	

4 is the highest difference and 1 is the least cost, then assign14 to 1 then cross off the second column.

	1	1	2	1	
1	8	2	3	7	42
1	9	4	5	6	17
4	7	1^{14}	6	5	3
	9	0	24	29	

	1		2	1	
4	8	2	3^{24}	7	18
1	9	4	5	6	17
1	7	1^{14}	6	5	3
	9	0	0	29	

4 is the highest difference and it occurs in first row and assign 24 to 3 and cross off third column.
Repeating the same process

	1			1	
1	8	2	3^{24}	7	18
3	9	4	5	6^{17}	0
2	7	1^{14}	6	5	3
	9	0	0	12	

	1			2	
1	8	2	3^{24}	7	18
	9	4	5	6^{17}	0
2	7	1^{14}	6	5^{3}	0
	9	0	0	9	

	8			7	
1	8^{9}	2	3^{24}	7	9
	9	4	5	6^{17}	0
	7	1^{14}	6	5^{3}	0
	0	0	0	9	

Finally we get

8^{9}	2	3^{24}	7^{9}
9	4	5	6^{17}
7	1^{14}	6	5^{3}

To get optimal solution Set b_1=0 then using relation a_i +b_j=$c_{i,j}$ then we get $a_2 = 7, a_3 = 6, a_1 = 8, b_2 = -5, b_3 = -5, b_4 = -1,$

	0	-5	-5	-1
8	8^{9}	2	3^{24}	7^{9}
7	9	4	5	6^{17}
6	7	1^{14}	6	5^{3}

Every cell cost is replaced by $c_{i,j} - a_i - b_j$ then we get

0^{9}	-1	0^{24}	0^{9}
2	2	3	0^{17}
1	0^{14}	5	0^{3}

Since $c_{1,2}$=-1 is negative, we have to form a cycle
Here we form cycle using $c_{1,2} = -1, c_{1,4} = 0, c_{3,4} = 0$ and $c_{3,2} = 0$ and choose $c_{1,4}$ as giver cell. Add 9 to $c_{1,2}$ and $c_{3,4}$ and subtract 9 from $c_{1,4}$ and $c_{3,2}$ then we get,

0^{9}	-1^{9}	0^{24}	0
2	2	3	0^{17}
1	0^{5}	5	0^{12}

Again Set $b_1 = 0$ then using relation $a_i + b_j = c_{i,j}$ then we get $a_2 = 1, a_3 = 1, a_1 = 0, b_2 = -1, b_3 = 0, b_4 = -1,$

	0	-1	0	-1
0	0^{9}	-1^{9}	0^{24}	0
1	2	2	3	0^{17}
1	1	0^{5}	5	0^{12}

Every cell cost is replaced by $c_{i,j} - a_i - b_j$ then we get

0^{9}	0^{9}	0^{24}	1
1	2	2	0^{17}
0	0^{5}	4	0^{12}

since all $c_{i,j}$ are positive, we attain optimal solution.Now replace the original cost

8^{9}	2^{9}	3^{24}	7
9	4	5	6^{17}
7	1^{5}	6	5^{12}

Total cost $= (8\times9)+(2\times9)+(3\times24)+(6\times17)+(1\times5)+(5\times12) =$ 329.

6.6 Another Example

Solve the transportation problem:

	M_1	M_2	M_3	M_4	
W_1	5	12	8	50	26
W_2	11	4	10	8	20
W_3	14	50	1	9	30
	15	20	26	15	

Solution:

	6	8	7	1	← column difference
3	5	12	8	50	26
4	11	4	10	8	20
8	14	50	1	9	30
↑ row difference	15	20	26	15	

The highest difference is 8 and it is in third row and the smallest value in this row is 1 assign 26 to this cell and reduce supply of W_3 by $30 - 26 = 4$. Hence delete corresponding column three, then we get

	6	8	1	
3	5	12	50	26
4	11	4	8	20
8	14	50	9	4
	15	20	15	

Thus the final basic feasible solution obtained by VAM must be

5^{15}	12^{0}	8	50^{11}
11	4^{20}	10	8
14	50	$1^{26}1$	9^{4}

The superscripted 0 on the cost of $C_{12} = 12$ be recorded since VAM results in $3 + 4 - 1 = 6$ circled cells.
according to step (2) we have

	0	7	37	45
5	5^{15}	12^{0}	8	50^{11}
-3	11	4^{20}	10	8
-36	14	50	1^{26}	9^{4}

by (3) we have

0^{15}	0^{0}	-34	0^{11}
14	0^{20}	-24	-34
50	79	0^{26}	0^{4}

(4) Since there are several negative $c'_{ij}s$, we continue with the transportation algorithm.

Note that there are three choices for a negative c_{ij} here. Our choice corresponds to the most negative c_{ij} but, more importantly, emphasizes the importance of including the superscripted 0 cell as a basis cell.
Repeating the same procedure until all c_{ij}b$\geq$ 0 for all i and j, then replacing all cells with their original costs, we obtain

5^{15}	12^{11}	8	50
11	4^{9}	10	8^{11}
14	50	1^{26}	9^{4}

and corresponding optimal solution
$x_{11} = 15$, $x_{12} = 11$, $x_{13} = x_{14} = 0$, $x_{21} = 0$, $x_{22} = 9$,
$x_{23} = 0$, $x_{24} = 11$, $x_{31} = x_{32} = 0$, $x_{33} = 26$, $x_{34} = 4$
$\min C = (5\times15)+(12\times11)+(I4\times9)+(8\times11)+(1\times26)+(9\times4) =$ 393.

6.7 Assignment Problem

The assignment problem is a special type of transportation problem. We will consider only balanced assignment problems in this section; the treatment of unbalanced assignment problems is similar to the treatment of unbalanced transportation problems. The general balanced assignment problem is the general balanced transportation problem in which $m = n$ and s_i=d_j $= 1, i, j = 1, 2, ..., n$ i.e.,

Minimize

$$C = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, ..., n$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, ..., n$$

$$x_{ij} \geq 0 \; \forall \; i, j$$

In balanced assignment problems, i indexes a set of n persons and j indexes a set of n jobs, $c_{i,j}$ is the cost of assigning person i to job j. It is desired to assign each person to exactly one job and each job to exactly one person so that the total cost of assignment is minimized. Here,

$$\begin{aligned} x_{i,j} &= 1, \text{ if person i is assigned to job j} \\ &= 0, \text{ otherwise} \end{aligned}$$

Now, since the balanced assignment problem is a special case of the balanced transportation problem we should be able to use the transportation algorithm to solve balanced assignment problems. However, due to the degenerate nature of balanced assignment problems, the transportation algorithm becomes inefficient and tedious to use.

Definition 6.4.

Let T be the tableau of a balanced assignment problem. A permutation set of zeros Z is a subset of zero cells of T such that every row and every column of T contains exactly one zero cell of T.

Example of a permutation set is

1	0	0*
0*	1	0
1	0*	1

The Hungarian Algorithm

(0) Given: An initial balanced ($n \times n$) assignment tableau.

(1) Convert all c_{ij}'s to non-negative integers if necessary by application of one or both of the following steps:

   (i) (Non-negativity) If$cij \leq 0$ for some i and j, compute $k_1 = \max_{i,j} \{|c_{ij}| \leq 0\}$,
   and add k_1 to every entry of the tableau.

   (ii) (Integrability) If $c_{ij} \notin$ Z and $c_{ij} \in$Q for some i and j, form the set
   S $= \{c_{ij} = p_{ij}/q_{ij} : c_{ij} \notin Z,\ c_{ij} \in Q, p_{ij}, q_{ij} \in Z, q_{ij} \geq 0\}$
   and compute $k_2 = \text{lcm}\{q_{ij} : c_{ij} = p_{ij}/q_{ij} \in S\}$. (Here, lcm denotes the least common multiple.) Multiply every entry of the tableau by k_2.

(2) Subtract the smallest entry in each row from every entry of the row to obtain a new tableau. Subtract the smallest entry in each column of the new tableau from every entry of the column to obtain the reduced tableau.

(3) Draw a minimum number k of horizontal and/or vertical lines to cover all zero entries of the reduced tableau.

(4) If $k = n$, STOP; a permutation set of zeros can be found among the zero entries of the reduced tableau; the optimal solution corresponds to this permutation set of zeros when all cells are replaced with their original entries from (0). If $k <$ n, choose the smallest uncovered entry. Subtract this entry from all uncovered entries (including itself) and add this entry to all covered entries corresponding to intersections of horizontal and vertical lines (all other covered entries remain unchanged), hence obtaining a new reduced tableau. Go to (3).

Example 6.5.

Illustrate step (1) (ii) of the Hungarian algorithm by converting the assignment tableau below to non-negative integers.

	J_1	J_2	J_3	
P_1	0.5	2	1	1
P_2	1.2	1/6	7	1
P_3	5/9	0	3.14	1
	1	1	1	

We first write all rational numbers in the tableau as quotients of integers with positive denominators. In the tableau below, all fractions have been reduced. Although this is not necessary, it will result in a smaller value of k_2,

1/2	2	1
6/5	1/6	7
5/9	0	157/50

Then $k_2 = \text{lcm}\,\{2, 5, 6, 9, 50\} = 450$; multiplying every entry of the tableau above by $k_2 = 450$, we obtain

225	900	450
540	75	3150
250	0	1413

Example 6.6.

Solve the assignment problem using Hungarian algorithm

	J_1	J_2	J_3	
P_1	8	7	10	1
P_2	7	7	8	1
P_3	8	5	7	1
	1	1	1	

The parenthetical numbers below correspond to the steps of the Hungarian algorithm.
(0) The initial tableau is balanced. Note that the supplies and demands are really unnecessary in an assignment tableau. Hereafter, all supplies and demands will be suppressed in assignment problems.
(1) Each c_{ij} is a non-negative integer
(2)

$$\boxed{\begin{array}{ccc} 8 & 7 & 10 \\ 7 & 7 & 8 \\ 8 & 5 & 7 \end{array}}$$

$$\boxed{\begin{array}{ccc} 1 & 0 & 3 \\ 0 & 0 & 1 \\ 3 & 0 & 2 \end{array}}$$

$$\boxed{\begin{array}{ccc} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 3 & 0 & 1 \end{array}} \text{ reduced tableau}$$

by step (3) we get

$$\boxed{\begin{array}{ccc} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 3 & 0 & 1 \end{array}}$$

Here k=2
So by step (4) check the condition,
since 2 = k < n = 3, we have to reduce the table

$$\boxed{\begin{array}{ccc} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 3 & 0 & 1 \end{array}} \longrightarrow$$

0	0	1
0	1	0
2	0	0

Go to step (3)

~~0~~	~~0~~	~~1~~
~~0~~	~~1~~	~~0~~
~~2~~	~~0~~	~~0~~

(4) $k = n = 3$,; hence, a permutation set of zeros can be found among the zero entries of the reduced tableau. In fact, there are two such permutation sets of zeros in the tableau, denoted by * and ** below:

0*	0**	1
0**	1	0*
2	0*	0**

Replacing all cells with their original entries, we obtain

8*	7**	10
7**	7	8*
8	5*	7**

and corresponding optimal solutions are
$*: x_{11}{=}x_{23}{=}x_{32} = 1$ all other x_{ij}'s $= 0$ and $\min C = 8 + 8 + 5 = 21$.
$**: x_{12}{=}x_{21}{=}x_{33}{=}1$, all other x_{ij}'s $= 0$ and $\min C = 7 + 7 + 7 = 21$.

Example 6.7.

A company wishes to assign five of its workers to five different jobs (one worker to each job and vice versa). The rating of each worker with respect to each job on a scale of 0 to 10 (10 being a high rating) is given by the following table:

	J_1	J_2	J_3	J_4	J_5
W_1	5	4	2	8	5
W_2	7	6	4	6	9
W_3	5	5	3	3	2
W_4	4	3	5	5	4
W_5	3	6	4	10	2

If the company wishes to maximize the total rating of the assignment, find the optimal assignment plan and the corresponding maximum total rating.

The problem as stated above is a maximization problem. But the Hungarian algorithm solves assignment problems which are minimization problems. Here by transforming the maximization problem into an equivalent minimization problem to which the Hungarian algorithm can be applied. This is easy-maximizing the assignment of the given tableau is equivalent to minimizing the assignment of the tableau consisting of the negatives of the given tableau entries.

-5	-4	-2	-8	-5
-7	-6	-4	-6	-9
-5	-5	-3	-3	-2
-4	-3	-5	-5	-4
-3	-6	-4	-10	-2

The Hungarian algorithm is now applied to this tableau. The parenthetical numbers below correspond to the steps of the algorithm.

(0) The tableau above is balanced.

(1) (i) $k_1 = |-10| = 10$:

-5	-4	-2	-8	-5
-7	-6	-4	-6	-9
-5	-5	-3	-3	-2
-4	-3	-5	-5	-4
-3	-6	-4	-10	-2

$\longrightarrow$

Add 10 to all entries, then we get

5	6	8	2	5
3	4	6	4	1
5	5	7	7	8
6	7	5	5	6
7	4	6	0	8

(2)

5	6	8	2	5
3	4	6	4	1
5	5	7	7	8
6	7	5	5	6
7	4	6	0	8

$\longrightarrow$

3	4	6	0	3
2	3	5	3	0
0	0	2	2	3
1	2	0	0	1
7	4	6	0	8

reduced tableau

continue step(3) and step(4) according to the need, finally we get

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

Replacing all cells with their original entries in the statement of the problem, we obtain

5*	4	2	8	5
7	6	4	6	9*
5	5*	3	3	2
4	3	5*	5	4
3	6	4	10*	2

to attain optimal solution as x_{11}=x_{25}=x_{32}=x_{43}=x_{54} = 1, all other x_{ij}'s = 0 Maximum rating = $5+9+5+5+10 = 34$.

Solved Problems

Problem 6.6.

Solve the assignment problem

38	21	34
41	14	36
28	20	25

Solution:
Given problem is a balanced 3×3 assignment problem. Subtract the smallest entry in each row from every entry of the row to obtain a new tableau. Subtract the smallest entry in each column of the new tableau from every entry of the column to obtain the reduced tableau.

9	0	8
19	0	17
0	0	0

Cross off a minimum number of horizontal and/or vertical lines to cover all zero entries of the reduced tableau. Here we have to cross off second column and third row to cover all zero entries.So minimum horizontal or vertical lines i.e $k = 2$ and $n = 3$ so the given table is not optimal.
choose the smallest uncovered entry. Subtract this entry from all uncovered entries and add this entry to all covered entries corresponding to intersections of horizontal and vertical lines hence obtaining a new reduced tableau.
Here 8 is the minimum,so subtract from all uncovered entries and add to point of intersection, then

1	0	0
11	0	9
0	8	0

Now minimum number of horizontal and/or vertical lines to cover all zero entries is $k = 3$ which is equal to n. Hence we attain optimal solution.

1	0	0*
11	0*	9
0*	8	0

Replacing all cells with their original entries, we obtain

38	21	34*
41	14*	36
28*	20	25

Thus one of the optimal solution is, 34+14+28=76.

Problem 6.7.

Solve the Assignment problem

4	6	5	10
10	9	7	13
7	11	8	13
12	13	12	17

Solution:
Given problem is a balanced 4×4 assignment problem. Subtract the smallest entry in each row from every entry of the row to obtain a new tableau. Subtract the smallest entry in each column of the new tableau from every entry of the column to obtain the reduced tableau.

0	1	1	1
3	1	0	1
0	3	1	1
0	0	0	0

Cross off a minimum number of horizontal and/or vertical lines to cover all zero entries of the reduced tableau. Here we have to cross off second column and third row to cover all zero entries. So minimum horizontal or vertical lines i.e $k = 3$ and $n = 4$ so the given table is not optimal.
choose the smallest uncovered entry. Subtract this entry from all uncovered entries and add this entry to all covered entries corresponding to intersections of horizontal and vertical lines hence obtaining a new reduced tableau.
Here 1 is the minimum,so subtract from all uncovered entries and add to point of intersection(i.e) $c_{4,1}$ and $c_{4,3}$, then

0	0	1	0
3	0	0	0
0	2	1	0
1	0	1	0

Now minimum number of horizontal and/or vertical lines to cover all zero entries is $k = 4$ which is equal to n.Hence we attain optimal solution.

0	0	1	0*
3	0	0*	0
0*	2	1	0
1	0*	1	0

Replacing the original cost, we get

4	6	5	10*
10	9	7*	13
7*	11	8	13
12	13*	12	17

Thus one of the optimal solution is given by
$7 + 13 + 7 + 10 = 37$

Problem 6.8.

Solve the Assignment Problem:

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

Solution:
Given problem is a balanced 5×5 assignment problem. Subtract the smallest entry in each row from every entry of the row to obtain a new tableau. Subtract the smallest entry in each column of the new tableau from every entry of the column to obtain the reduced tableau.

0	4	1	3	3
0	1	1	1	2
0	2	1	2	2
0	0	0	0	2
0	2	1	1	0

0	3	0	2	2
0	0	0	0	1
0	1	0	1	1
1	0	0	0	2
1	2	1	1	0

now we obtain optimal solution. here two optimal solutions were given

0*	3	0**	2	2
0	0*	0	0**	1
0**	1	0*	1	1
1	0**	0	0*	2
1	2	1	1	$0_{**}*$

Replacing original cost we get

1*	6	2**	7	5
2	4*	3	6**	5
3**	6	4*	8	6
4	5**	4	7*	7
5	8	6	9	$6_{**}*$

Two optimal solutions are
1)$1 + 4 + 4 + 7 + 6 = 22$
2)$2 + 6 + 3 + 5 + 6 = 22$

Problem 6.9.

Solve the unbalanced assignment problem

9	7	8	6	8
10	8	7	9	6
9	6	9	7	8
8	9	10	7	6

Solution:
Given, a 4×5 unbalanced problem, to make a balanced problem insert a row of 0's then we get,

9	7	8	6	8
10	8	7	9	6
9	6	9	7	8
8	9	10	7	6
0	0	0	0	0

Subtract the smallest entry in each row from every entry of the row to obtain a new tableau. Subtract the smallest entry in each column of the new tableau from every entry of the column to obtain the reduced tableau.

3	1	2	0	2
4	2	1	3	0
3	0	3	1	2
2	3	4	1	0
0	0	0	0	0

Here we by cross off fifth row, second, fourth and fifth column to cover all zeros(i.e) $k = 4$ less than n. 1 is the minimum of all uncovered cells so subtract 1 from all uncovered entries and add to point of intersection(i.e) $c_{5,2}$,$c_{5,4}$ and $c_{5,5}$, then we get

2	1	1	0	2
3	2	0	3	0
2	0	2	1	2
1	3	3	1	0
0	1	0	1	1

Now minimum number of horizontal and/or vertical lines to cover all zero entries is $k = 5$ which is equal to n. Hence we attain optimal solution.

2	1	1	0*	2
3	2	0*	3	0
2	0*	2	1	2
1	3	3	1	0*
0*	1	0	1	1

Replacing original cost, we get
Thus optimal solution is $6 + 7 + 6 + 6 = 25$.