OPERATIONS RESEARCH

SUBMITTED TO MAHATMA GANDHI UNIVERSITY KOTTAYAM IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE MATHEMATICS

JOINTLY BY

FATHIMATHUL SAHALA N A (Reg.No:210021031517)

JYOTHIKA K JAYAN (Reg.No:210021031518)

VISHNU SIJU (Reg.No:210021031523)

UNDER THE SUPERVISION OF

Dr. LAKSHMI C

Estd.1965

DEPARTMENT OF MATHEMATICS BHARATA MATA COLLEGE, THRIKKAKARA, KOCHI AFFLIATED TO MAHATMA GANDHI UNIVERSITY KOTTAYAM

DECLERATION

We hereby declare that this project report entitled "OPERATIONS RESEARCH" is a bonafide record of work done by us under the supervision of Dr. LAKSHMI C and the work has not previously been formed the basis for the award of any academic qualification fellowship or other similar title of any other university or board.

FATHIMATHUL SAHALA N A

JYOTHIKA K JAYAN

VISHNU SIJU

Place: Thrikkakara

Date:

CERTIFICATE

This is to certify that the report of the project entitled "OPERATIONS RESEARCH" carried out and submitted jointly by Miss FATHIMATHUL SAHALA N A, Miss JYOTHIKA K JAYAN, and Mr. VISHNU SIJU in partial fulfillment of the requirements for the award of the B.Sc. Degree in Mathematics is a bonafide record of the studies undertaken by them under my supervision at the Department of Mathematics, Bharata Mata College, Thrikkakara during the academic year 2023-2024. This dissertation has not been submitted for any other degree elsewhere.

Dr. LAKSHMI C

(Supervisor and HOD)

Place: Thrikkakara

Date:

ACKNOWLEDGEMENT

We would like to express our sincere thanks to Dr. LAKSHMI C, Head of the Department of Mathematics, Bharata Mata College, Thrikkakara, whose instruction and inspiring guidance helped us to complete this work. We are also obliged to all teachers of the Department of Mathematics, Bharata Mata College for their encouragement and co-operation for completing this work.

FATHIMATHUL SAHALA N A

JYOTHIKA K JAYAN

VISHNU SIJU

Place: Thrikkakara

Date:

CONTENTS

CHAPTER 0

INTRODUCTION

Operations research (OR) is a methodical approach to problem-solving and decision-making in organizational management.

The process of operations research can be roughly divided into the following stages.

- Recognizing an issue that requires a solution.
- Building a model of the issue that takes factors and the real world into account.
- Solving the problem by applying the model.
- Analyzing the performance of each solution after testing it on the model.
- Putting the actual problem's answer into practice.

Characteristics commonly found in operations research include optimization, simulation, and the application of probability and statistics. Optimization aims to enhance performance within given constraints, often by narrowing down choices. Simulation entails creating models to test solutions before implementation. Probability and statistics are utilized to analyze data, identify risks, generate forecasts, and evaluate potential solutions.

Compared to standard software and data analytics tools, the field of operations research offers a more potent method for making decisions. Companies that hire operations research specialists can obtain more comprehensive datasets, take into account all of their alternatives, forecast all potential outcomes, and assess risk. Furthermore, in order to identify the best suitable strategies for resolving a given problem, operations research can be customized to certain business processes or use cases.

Applications of Operations Research

There are numerous application cases for operations research, such as:

Analyzing risks:

An operation research tool called risk analysis enables companies to identify and address any issues before they become obstacles to their goals or projects. Additionally, risk analysis can be used for non-business endeavors like event planning and property ownership.

Inventory evaluation:

Inventory is an asset on a balance sheet that represents the products a business plans to sell to clients in the future. It includes work-in-progress items and raw materials needed to make such goods in addition to the final products. Thus, inventory analysis helps companies decide how much stock to keep on hand to satisfy consumer demand and keep expenses associated with inventory storage to a minimum.

Methodical organizing:

Through the application of operations research, strategic planning enables leaders of organizations to clarify their future vision and pinpoint their aims and objectives. The procedure entails figuring out which goals should be completed first in order for the organization to realize its declared vision. Strategic planning is frequently used to represent long-term objectives that have a three- to five-year lifespan, however this can be extended.

Market analysis:

Businesses employ marketing research as a method or set of procedures to gather data that helps them better understand their target market. Businesses use this data to improve their user experience, make better products, and give customers better products. Typically, marketing research is done to learn what consumers want and how they respond to certain products or services

Logistics:

The management of the movement of goods between the point of origin and the point of consumption to satisfy the needs of businesses and consumers is known as logistics in the business world. Physical resources such as food, drink, materials, equipment, and animals can all be managed in logistics. It can also contain abstract objects like time and data.

Management of revenue:

Using a methodical analytics approach, revenue management forecasts micro-level customer behavior in order to maximize product availability and price while boosting revenue growth. Said another way, the main objective is to provide the appropriate product to the appropriate customer at the appropriate cost at the appropriate moment.

Analysis of sales:

The technique of analyzing sales data to find patterns and trends is known as sales analysis. Businesses can use sales data to help them make informed decisions about their products, prices, promotions, consumer requests, inventory, and other organizational aspects. Sales analysis in certain organizations is as simple as regularly reviewing the sales figures.

Making Plans:

A component of operations research called scheduling helps to plan, coordinate, and optimize tasks and workloads in a manufacturing or production process. It is employed in the planning of production procedures, material acquisitions, plant and machinery resources, and human resources.

Putting up an auction:

An auction is set up in order to sell properties and assets to potential buyers. An auction is a public method of buying and selling where customers are invited to submit bids for particular items that are owned by companies and owners.

Predicting:

It's a tactic that makes informed forecasts regarding the future course of trends by utilizing historical data as inputs. Businesses use forecasting to understand how to distribute funds or make plans for future expenditures.

Chain of supply:

Supply chain management is the management of the entire process of turning raw resources into a final product. It comprises establishing a centralized management mechanism to link a network of suppliers. Every supplier functions as a connection between producers and sellers in the manufacturing cycle.

CHAPTER 1

MATHEMATICAL PRELIMINARIES

Slack variable

Introduced in linear programming to convert constraints into an equality constraint. It represents the amount by which the LHS of constraints can be exceeded without violating the constraint.

Iteration:

Iteration is a process that repeats its step until a specific condition is met.

Initial feasible solution:

Satisfying the constraint of the problem is called the initial feasible solution.

Graph

A graph can be denoted as $G(V, U)$ where V is the set of elements V_j ; $j = 1,2, \dots, n$ which represents points and U is the set of pairs (v_j, v_k); where v_j, v_k are elements of V which represents the arcs joining the points of V.

V- vertices U- arcs

Directed arc

If (v_j, v_k) are ordered pairs, then we call it as directed arc.

Directed graph

A graph which contains directed arc is called directed graphs.

Finite Graph

If U and V are finite set, then graph $G(U, V)$ is finite graph.

Initial and terminal vertex

The arc incident from or going from is called initial vertex and the arc incident to or going to is called terminal vertex

Subgraph

A subgraph of $G(U, V)$ is defined as a graph $G_1(V_1, U_1)$with $V1$ subset of V and $U1$ containing all those arcs of G which connects the vertices of $G1$.

Partial graph

A partial graph of $G(U, V)$ is a graph $G_2(V, U_2)$ which contains all the vertices of G and some of its arcs.

$\boldsymbol{\Omega(V_k)}$

$\Omega(V_k)$ is the set of arcs of $G(V, U)$ incident with a subset V_k of V.

Chain

A sequence of arcs $\{u_1, u_2, \dots, u_k, u_{k+1}, \dots, u_q\}$ of a graph such that every intermediate arc u_k has one vertex common with the arc u_{k-1} and another common with u_{k+1}

Path

A path is a chain in which all the arcs are directed in the same sense such that the terminal vertex of the preceding arc is the initial vertex of the succeeding arc

Cycle

A cycle is a chain in which no arc is used twice and the first arc has a vertex common with the last arc

Circuit

In a cycle, if all the arc arcs are directed in the same sense then, it is called a circuit.

Connected graphs

If for every pair of vertices there exist a chain that connecting the two then that graph is said to be connected.

Strongly connected graphs

If there is a path connecting every pair of vertices then the graph is said to be strongly connected.

Tree

A tree is defined as a connected graph with at least 2 vertices and no cycles.

Centre of a graph

A vertex which is connected to every other vertex of a graph is called centre of a graph

A tree with a centre is called *arborescence*

CHAPTER 2
LINEAR PROGRAMMING

A technique for optimizing operations under certain limitations in mathematics is called linear programming. Maximizing or minimizing the numerical value is the primary goal of linear programming. It is composed of linear functions that are bound by formulas that are either linear equations or inequality. When determining the best way to use resources, linear programming is thought to be a crucial strategy. The two terms "linear" and "programming" make up the phrase "linear programming." A degree one link between several variables is said to be "linear." The process of identifying the optimal answer from a range of options is referred to as "programming."

Mathematicians and professionals in the commercial, economic, manufacturing, and telecommunications domains frequently utilize linear programming. Let's talk about what linear programming is, how it works, and many approaches to solving linear programming issues in this article.

2.1 Formulation of linear programming problems

Decision Variables: These represent the quantities a decision maker can control or decide upon. In an LP problem, decision variables are typically the unknown quantities we want to determine in order to optimize some objective. For example, in a production planning problem, decision variables might represent the quantities of different products to produce.

Structural Constraints: These are limitations or conditions that restrict the possible values for the decision variables. They define the feasible region within which the optimal solution must lie. These constraints are typically linear equations or inequalities derived from the problem's structure and requirements.

Non-negativity Constraints: In many optimization problems, including linear programming, decision variables cannot take negative values. This constraint is imposed to reflect the real-world meaning of the decision variables. For instance, quantities produced, purchased, or allocated cannot be negative. Non-negativity constraints are usually expressed as inequalities, where decision variables are greater than or equal to zero.

Single Objective Function: This is the function that we want to optimize (either maximize or minimize) based on the values of the decision variables. In linear programming, there's only one objective function.

Optimal solution: refers to the best possible outcome given certain constraints and an objective function. It's found at the vertex of the feasible region that maximizes or minimizes the objective function. This solution is determined using mathematical techniques like the simplex method or interior-point methods.

Feasible solution: This is a solution that satisfies all the constraints of the problem. It means that all the variables are within their defined ranges, and all the constraints are met simultaneously. However, a feasible solution may not necessarily be the best or most optimal solution; it just needs to adhere to the problem's constraints.

These elements together form the basis for constructing and solving linear programming problems, allowing decision makers to optimize their objectives subject to various constraints.

Let's illustrate this with an example:

A company manufactures two types of products A and B. Each product uses milling and drilling machine. the process time per unit of A on the milling is 10 hours and on the drilling machine is 8 hours. The processing time per unit of B on the milling is 15 hours and on the drilling machine is 10 hours. the maximum number of hours available per week on the milling and the drilling machine are 80 hours and 60 hours respectively. Also the profit per unit of selling A and B are Rs.25 and Rs.35 respectively.

Formulation of LP model to determine the production volume of each of the products such that the profit is maximized

STEP 1. IDENTIFY THE OBJECTIVE: Determine what you want to optimize, whether it's maximizing profit, minimizing cost, or something else.

OBJECTIVE**:** Maximize profit.

STEP 2. DEFINE DECISION VARIABLES: If you're optimizing production, decision variables could be the quantities of different products to produce.

DECISION VARIABLES: Let x and y be the production volumes of products A and B respectively.

STEP 3. WRITE THE OBJECTIVE FUNCTION: This function expresses the objective in terms of the decision variables. For instance, if you are maximizing profit, the objective function might be the total revenue minus the total cost.

OBJECTIVE FUNCTION: Maximize $Z = 25x + 35y$ (total profit)

STEP 4. SET CONSTRAINTS: Identify any limitations or restrictions on the decision variables. These could include resource constraints, demand constraints, or production capacity constraints.

STEP 5. FORMULATE CONSTRAINTS: Translate the limitations into mathematical equations or inequalities involving the decision variables.

CONSTRAINTS: $10x + 15y \leq 80$

$8x + 10y \leq 60$

STEP 6. NON-NEGATIVITY CONSTRAINTS: Typically, decision variables must be non-negative, meaning they cannot be negative quantities.

- Non-negativity constraint: $x, y \geq 0$

STEP 7. PUT IT ALL TOGETHER: Combine the objective function and constraints into a single mathematical model.

Put it All Together:

Maximize $25x + 35y$

Subject to: $10x + 15y \leq 80$

$8x + 10y \leq 60$

$x, y \geq 0$

This is the formulation of the linear programming problem.

2.2 SIMPLEX METHOD

Simplex method is an iteration algorithm for solving linear programming. It starts with an initial feasible solution and systematically moves towards an optimal solution by improving the objective function. It does this through a series of steps, where at each iteration, it identifies a variable to enter the solution (pivot) and a variable to leave the solution. This process continuous until no further improvement in the objective function can be achieved, indicating the current solution is optimal.

The following is the linear programming simplex technique algorithm:

Step 1: Identify the specific issue. Write the objective function and inequality restrictions, for example.

Step 2: Add the slack variable to each inequality expression in order to transform the provided inequalities into equations.

Step3: Make the first simplex tableau. In the bottom row, write the goal function. Each inequality constraint is shown here in a separate row. The problem can now be expressed as an augmented matrix, or what is known as the initial simplex tableau.

Step 4: To determine the pivot column, find the largest negative item in the bottom row. The highest coefficient in the objective function is defined by the largest negative entry in the bottom row, which will enable us to raise the objective function's value as quickly as feasible.

Step 5: Compute the quotients. We must divide the entries in the far right column by the entries in the first column, omitting the bottom row, in order to find the quotient. The row is identified by the least quotient. The pivot element will be the row found in this step and the element found in the step.

Step 6: Pivot the column such that all other entries are zero.

Step 7: Close the procedure if the bottom row contains no negative items. If not, go back to step 4.

Step 8: Lastly, ascertain the answer linked to the ultimate simplex tableau.

EXAMPLE:

Solve the following LPP using the Simplex method

Maximize: $Z = 12X_1 + 16X_2$

Subject to:

$$10X_1 + 20X_2 \leq 120$$

$$8X_1 + 8X_2 \leq 80$$

$$X_1 \; and \; X_2 \geq 0$$

Solution:

Max Z = $12X_1 + 16X_2 + 0S_1 + 0S_2$

Subject to

$$10X_1 + 20X_2 + S_1 = 120$$

$$8X_1 + 8X_2 + S_2 = 80$$

$$X_1, X_2, S_1 \; and \; S\_2 \geq 0$$

Initial Simplex Table

CBi	C_j	12	16	0	0	Solution	Ratio
	Basic variable	X_1	X_2	S_1	S_2		
0	S_1	10	20	1	0	120	6
0	S_2	8	8	0	1	80	10
	Z_j	0	0	0	0		
	$C_j - Z_j$	12	16	0	0		

Table 2.1

C_j - coefficient of objective function

CBi - coefficient of basic variable

Z_j row values are obtained by multiplying the CBi column by each column, element by element and summing.

Optimality condition:

For max : all $C_j - Z_j \leq 0$

For min : all $C_j - Z_j \geq 0$

Here we have got positive values. So we don't reach at optimality. So we need to proceed for further steps to reach optimality .

In order to proceed further, the first step is to select the maximum value in $C_j - Z_j$ corresponding column is called key column

To find the ratio :

solution column ÷ key column

Minimum value in Ratio - corresponding row is key row.

Here 20 is the key element.

X_2 is an Entering variable and S_1 is leaving variable.

- 1^{st} Iteration Table

CBi	C_j	12	16	0	0	Solution	Ratio
	Basic variable	X_1	X_2	S_1	S_2		
16	X_2	1/2	1	1/20	0	6	12
0	S_2	4	0	-2/5	1	32	8
	Z_j	8	16	4/5	0		
	$C_j - Z_j$	4	0	-4/5	0		

Table 2.2

R1 = R1 ÷ key element

R2 = old value - (corresponding key column value × corresponding key row value) ÷ key element

For max: all $C_j - Z_j \leq 0$

Here we have one positive value. So we need to proceed for further steps to reach optimality. Repeat as above

Now 4 is the key element

X_1 is entering the variable and the S_2 is leaving the variable.

- 2^{nd} Iteration Table

CBi	C_j	12	16	0	0	Solution
	Basic variable	X_1	X_2	S_1	S_2	
16	X_2	0	1	1/10	-1/8	2
12	X_1	1	0	-1/10	1/4	8
	Z_j	12	16	2/5	1	128
	$C_j - Z_j$	0	0	-2/5	-1	

Table 2.3

Here $C_j - Z_j \leq 0$

Now, we reach at optimal solution.

$X_1 = 12, X_2 = 16$

Optimal solution = 128

CHAPTER 3

FLOW AND POTENTIAL IN NETWORKS

An easy way to visualize networks is in transportation or communication systems, such as highways, railroads, pipelines, nerves, or blood arteries. Network diagrams are easily recognized in electrical theory. Networks provide a wide range of mathematical difficulties, from kid-friendly puzzles to complex issues that stump mathematicians. Diagrammatically, many problems are conveniently stated as networks, especially those that include sequential actions or separate but linked states or stages. An issue that doesn't seem to have any structure might occasionally take on a mathematical shape that can best be understood and resolved by viewing it as a network.

Graphs are networks in the more abstract and broad sense. Mathematicians have been studying and researching graph theory a lot lately, and it is finding more and more applications in a variety of fields. Graph theory is particularly significant in the subject of operations research because, in many cases, selecting the best sequence of operations from a finite set of possibilities that can be represented as a graph is the problem of finding an optimal solution.

This chapter will cover a few particular types of linear programming problems that can be solved with the aid of graph theory concepts. Instead of introducing graphs in their abstract form, we will address specific issues and demonstrate how they can be viewed and resolved as networks or graphs.

3.1 MINIMUM PATH PROBLEM

Assume that a graph $G(V, U)$ has two vertices, v_a and v_b , and that each arc (v_j, v_k) has a number x_{jk} . There are numerous paths to go from v_a to v_b. Every path has a length defined as $\sum x_{jk}$, where $\sum x_{jk}$ is the total of all the arcs that comprise the path. The challenge is to find the path with the smallest length.

Here the term "length" is used in a generic sense of any real number connected with the arc and should not be regarded as a geometrical distance. A road map connecting towns is a graph and the distance along a road between any two towns is the length of a path within the present definition of the term, but this is only a particular case. The time or the cost involved in going from one town to another is also a "length" under the present definition. There may bring more abstract situation in which the length is not even non-negative. In general x_{jk} is a real number, unrestricted in sign.

Many methods are there to solve the problems of minimum path. Here we discuss about two of them. One is applicable only to the case where $x_{jk} \geq 0$. And the other is used in general cases where x_{jk} is unrestricted

3.1.1 All arc lengths non-negative

The minimum path from v_a to v_j can be indicated by f_j . We have to find f_b . Clearly $f_a = 0$

Let v_p be a subset of V such that v_a is in V_p and v_b is not in V_p. Assume further that f_j has been found for each v_j in v_p. For any v_j in V_p and not in V_p, find $f_j + x_{jk}$ so that (v_j, v_k) is an arc incident from V_p.

Let $f_r + x_{rs} = \min (f_j + x_{jk})$

Where v_r is an element of V_p and v_s is not an element of V_p

Then the minimum path from v_a to v_s is given by

$$f_s = f_r + x_{rs}.$$

This is the case because $f_r + x_{rs}$ is the least path that leaves V_p along a single arc, and we must leave V_p in order to reach v_s. Any additional path to v_s can either follow a different, larger single arc that leaves V_p and ends at v_s, or it can follow a different, larger arc that leaves V_p, travels to another point, and then returns to v_s.

Repeat the process to create an expanded subset V_{p+1} of V, which is defined as $V_{p+1} = Vp\ U\ \{v_s\}$. Assume that V_0 consists of a single vertex v_a and $f_a = 0$ and that we begin with $p = 0$. The set $V_1, V_2, \dots, V_p, V_{p+1}, \dots$ are constructed in accordance with the above-described approach. We have located f_b as soon as we reach a set in this sequence that contains v_b. There is no path that connects v_a to v_b if no such set can be found.

Example: With the provided graph, find the shortest path between v_0 and v_8, where the length of the journey is indicated by the number along a directed arc.

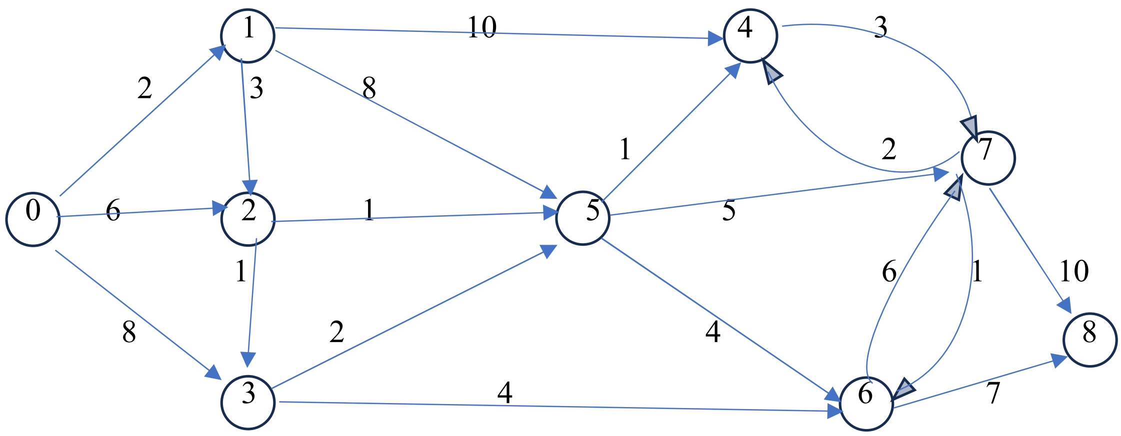

Fig. 3.1

The table below displays the iterations based on the previously described algorithm. The vertices in the subset , V_p are listed in the , V_p column. The shortest paths from v_0 to these vertices are indicated under f. The arcs incident from , V_p are $\Omega^-(v_p)$, where (v_j, v_k) is represented as (j, k). The arc lengths are listed under x. The vertex to which this minimal distance leads, v_s, is the minimum of $f + x$ and is presented in the expanded subset V_{p+1} in the subsequent iteration.

Table 3.1

p	**V_P**	**f**	**$\Omega^-(v_p)$**	**x**	**f + x**	**f_s**	**v_s**

0	0	0	(0,1) (0,2) (0,3)	2 6 8	2 6 8	2	1
1	0 1	0 2	(0,2) (0,3) (1,2) (1,4) (1,5)	6 8 3 10 8	6 8 5 12 10	 5	 2
2	0 1 2	0 2 5	(0,3) (1,4) (1,5) (2,3) (2,5)	8 10 8 1 1	8 12 10 6 6	 6 6	 3 5
3	0 1 2 3 5	0 2 5 6 6	 (1,4) (3,6) (5,4) (5,7)	 10 4 1 5	 12 10 7 11	 7	 4
4	0 1 2 3 4 5	0 2 5 6 7 6	 (3,6) (4,7) (5,7)	 4 3 5	 10 10 11	 10 10	 6 7
5	0 1 2 3 4 5 6 7	0 2 5 6 7 6 10 10	 (6,8) (7,8)	 7 10	 17 20	 17	 8

The shortest path, which passes through the vertices (0,1,2,3,6,8), is determined to be 17 in length.

It should be understood that neither the problem description nor the solution depends on the graph's actual drawing. If all of the vertices, arcs, and arc lengths are defined, the problem is fully stated. Drawing a figure may not be necessary or practical in a huge problem with many vertices and arcs.

3.1.2 Arc lengths unrestricted in sign.

In the graph $G(V,U)$, let v_a and v_b be two vertices with real, positive, or negative arc lengths. We need to determine the shortest route between v_a and v_b. It is assumed that the graph does not contain any circuits whose total arc length is negative. Because if such a circuit exists, it can be circled endlessly, allowing one to find an infinite solution by reducing the path's length.

Create an arborescence with the centre v_a, V_1 containing all the vertices of V that can be reached from v_a along a path, and U_1 containing some of the arcs of U that are required to construct the arborescence, $A_1(V_1,U_1), V_1 \subseteq V, U_1 \subseteq U.$

A path links v_a and v_b if V_1 contains v_b. This pathway is distinct inside a certain arborescence. Numerous arborescence could imply numerous pathways. Any single arborescence is A_1. There is only one path from v_a to v_b if there is only one arborescence that can occur in any given situation, and that path is the solution. There is no path from v_a to v_b and no solution to the issues if V_1 contains no v_b.

The arborescence's construction and methodology are simple. Draw the arcs that lead from v_a. Draw arcs (not necessarily all of them) connecting the vertices that have been so reached to the other vertices.

It is imperative that no vertex be approached by more than one arc, meaning that no vertex should have more than one arc incident to it. A vertex that has no incident arc is excluded because it cannot be reached from v_a. No arc incident to v_a should be drawn.

Let f_j represent the path length in the arborescence from v_a to any vertex v_j. For every v_j in V_1 , the arborescence uniquely defines f_j ; nevertheless, f_j need not be minimal. Consider an arc (v_k, v_j) in G that isn't in A_1 . Compare f_j with the length $f_k + x_{kj}$. Don't alter if $f_j \leq f_k + x_{kj}$

If $f_j > f_k + x_{kj}$, add the arc (v_k, v_j) in place of the arc incident to v_j in A_1 . Because of this, the arborescence is changed from A_1 to A_2, and f_j is reduced to $f_k + x_{kj}$, or $f_j - f_k - x_{kj}$. The same amount is likewise subtracted from the lengths of the pathways leading to the vertices and the lengths of the paths to the vertices going through v_j are also reduced to the same amount. After making these modifications, the new values of f_j for each v_j in A_2 are computed.

Proceed as you did in A_2, choosing a vertex and determining whether a different arc leads to it via a shorter path. If so, change A_2 to A_3 and change f_j appropriately.

At some point, an arborescence A_r is attained, and the process described above cannot alter it further. In this arborescence, f_b is the shortest path to v_b, while a_r indicates the minimum path of each v_j from v_a. Here is the proof.

Proof: Let any path in G from v_a to v_b be represented by $(v_a, v_1, v_2, \dots, v_b)$. $x_{a1} + x_{12} + \ldots + x_{pb}$ is its length. Since A_r contains all of the vertices of G that are accessible from v_a, the vertices in this path are also in A_r. According to the last paragraph's description of $A_r's$ attribute, for each vertex v_j in A_r and for each arc (v_k, v_j) in G,

$$f_j \leq f_k + x_{jk},$$

or

$$f_j - f_k \leq x_{jk}$$

because otherwise A_r could have been further modified. Writing these inequalities for all vertices of the above path,

$$f_1 - f_a \leq x_{a1},$$

$$f_2 - f_1 \leq x_{12},$$

………………..

$$f_b - f_p \leq x_{pb},$$

Adding, we get

$$f_b - f_a \leq x_{a1} + x_{12} + x_{23} + \ldots + x_{pb}$$

Or, since $f_a = 0$

$$f_b \leq x_{a1} + x_{12} + x_{23} + \ldots + x_{pb}$$

We can thus demonstrate that there is no path in G that can be smaller than f_b from v_a to v_b. This path is the minimum because the path of length f_b is also in G.

proved

The maximum length path can be determined by either reversing the inequality $f_j > f_k + x_{kj}$ *to* $f_j < f_k + x_{kj}$ as the criterion for changing an arc in the arborescence, so that at every stage a greater path is selected against a smaller one, or by changing the sign of the length of all arcs and then determining the minimum path.

Example: In the provided graph G of Fig. 3.2, find the minimal path from v_0 to v_7 .

With a centre v_0 made up of all the graph's vertices that can be accessed from v_0 (v_8 is therefore omitted), draw an arborescence A_1 (Fig. 3.3) and the required number of arcs. Observe that these arborescence can occur in large numbers. A_1 is one among them.

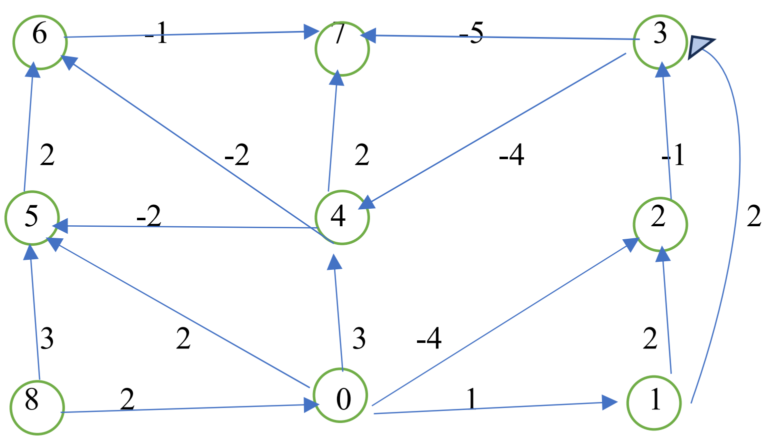

Fig 3.2

The length f_j of the paths from v_0 to different vertices v_j of A_1 are as follows.

$$f_0 = 0, f_1 = 1, f_2 = -4, f_3 = 3, f_4 = 3, f_5 = 2, f_6 = 4, f_7 = 5$$

consider the vertex v_2 . There is an arc (v_1, v_2) in G which is not in A_1 , Such that

$$f_2 = -4 < f_1 + x_{12} = 1 + 2 = 3.$$

So we leave A_1 unchanged.

Now consider the vertex v_3 . There is an arc (v_2, v_3) in G which is not in A_1 such that

$$f_3 = 3 > f_2 + x_{23} = -4 - 1 = -5.$$

So we delete the arc(v_1, v_3) which is incident to v_3 in A_1 and instead include the arc (v_2, v_3). This gives us a new arborescence $A_2$with the $f_3 = 5$ since no vertex is reached in A_1through v_3, all other f_j remain unchanged.

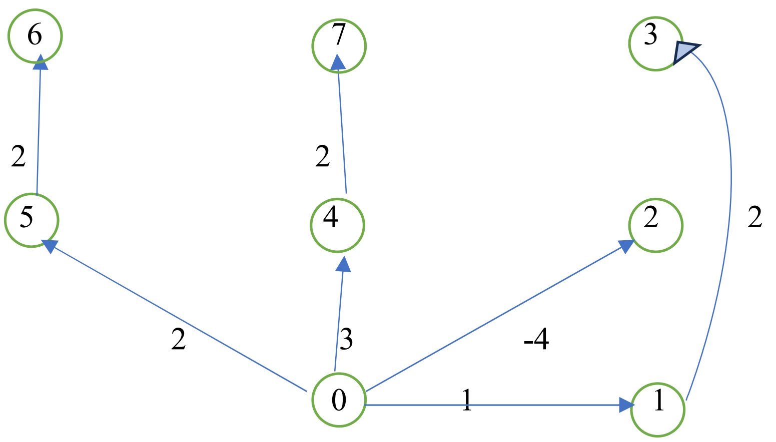

Fig. 3.3

Coming now to v_4 in A_2 , arc is in G but not in A_2 such that

$$f_4 = 3 > f_3 + x_{34} = -5 - 4 = -9.$$

Thus, we include (v_3, v_4), remove the arc (v_0, v_4), and obtain another arborescence A_3 with $f_4 = -9$ and, as a result, $f_7 = -7$.

If we keep going in this manner, we eventually reach the arborescence, which is unchangeable. The path from v_0 to any vertex cannot be made shorter by any other arc. Testing for each potential alternate arc reveals this. The shortest path, with length -12, is $(v_0, v_2, v_3, v_4, v_6, v_7)$ from v_0 to v_7.

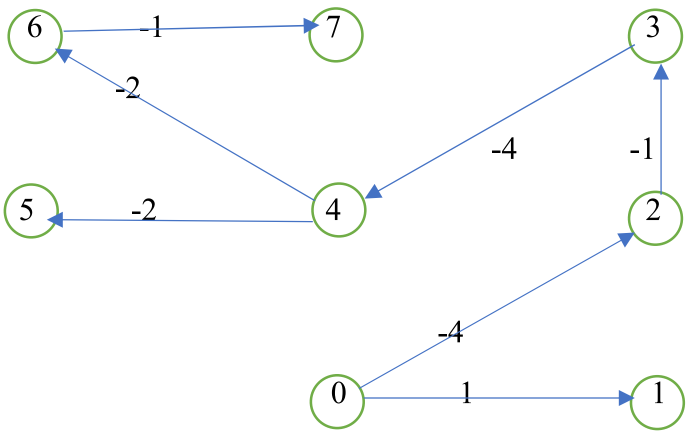

Fig. 3.4

3.2 SPANNING TREE OF MINIMUM LENGTH

Let $T(V, U')$ be a tree such that $U' \subseteq U$ and let $G(V, U)$ be a connected graph with undirected arcs. T shares the same set of vertices as G, and all of T's arcs are also arcs of G. Then, T is said to span G, or $T(V, U')$ is said to be a spanning tree of $G(V, U)$. It is clear that a graph's spanning tree is not unique. For example , in Figure 3.5, a spanning tree remains, even after removing u_1 and adding u_2 to the tree.

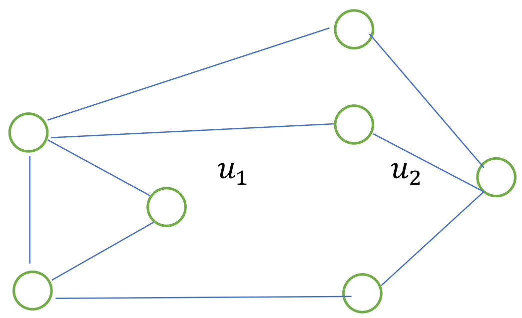

Fig. 3.5

If the length of each arc in G is supplied as a non-negative value, the task is to identify the spanning tree with the smallest length possible. The issue arises while constructing telephone wires or pipes that connect towns.

By x_{jk} , we indicate the length of the arc between v_j and v_k. Since all of the arcs are undirected, $x_{jk} = x_{kj}$, For all arcs $x_{jk} \geq 0$

We present a strategy for resolving this issue, which requires that none of G's arcs have the same length. We therefore presume that this requirement is met .If there are several arcs that are the same length, it is possible to add little length variations to them without affecting the optimal solution. For example, if

$$x_{jk} = x_{pq} = x_{rs}$$

we put $$x_{pq} = x_{jk} + e, \quad x_{rs} = x_{jk} + 2e, \quad e > 0,$$

Keeping e small enough to prevent x_{rs} from being more than or equal to any other arc length that is greater than x_{jk}.

Here is the algorithm. Consider the graph $G(V, U)$ and its collection of vertices, V. G is connected , therefore each vertex has an additional arcs incident associated with it. Additionally, because every arc in U has a different length, there is always one arc that is the shortest for every vertex among all the arcs that intersect with it. We'll refer to this Arc as a connection between the vertex and its closest neighbour.

One by one, go over each vertex in V and connect it to the neighbour that is closest to it. In this manner, we will obtain a partial graph $G_1(V, U_1)$ of G that includes all of its vertices as well as a portion of its arcs. Generally speaking, G_1 could be an unconnected graph made up of multiple components, each of which is connected within itself.

Let's handle each of these parts as though it were a single vertex. We will have a new collection of "vertices," each of which is a part of the graph G_1. Arcs of the graph G connect each and every one of these "vertices" to one or more of the other "vertices." G would not be connected if this were not the case. The 'distance' between two 'vertices' will be determined by taking the arc of the smallest length between them. Once more, we go over each of these "vertices" one at a time and adjust each one to its nearest neighbour -that is, the one that is the closest to it. After this operation, another graph $G_2(V, U_2)$ will be produced, which again may have components

Once more, treat each of these elements as a "vertex," and connect each "vertex" to its closest neighbour to create a graph $G_3(V, U_3)$This process is repeated until a connected graph $G_p(V, U_p)$is produced. The required spanning tree is this graph.

To make the above-described approach clear, we figure out an example before providing a proof of this algorithm.

Example: Determine the graph $G(V, U)$ of Figure 3.6's minimum spanning tree. It is an undirected graph, as you can see.

Graph $G_1(V, U_1)$ in figure 3.7 is obtained by traversing all vertices from v_1 to v_8 and drawing the arc connecting each to its closest neighbour. The closest neighbours for $v_1$are v_3, v_2 are v_3, v_3 are v_2, and so forth. There is no connectivity in graph G_1 . There are three parts: A_1 , A_2, and A_3. They are regarded as three 'vertices'. The lengths of the arcs in G that connect A_1 to A_2, are 14, 18, 8, 16, 11, and so the separation between A_1 and A_2 is 8. In the same way, A_2 and A_3 are separated by 9. Furthermore, the distance between A_1 and A_3 is infinity because there isn't an arc connecting them. $A_1's$ and $A_2's$ closest neighbours are A_2 and A_1 1, respectively. So we connect the two by arc (v_2, v_5) which measures the distance between the two.

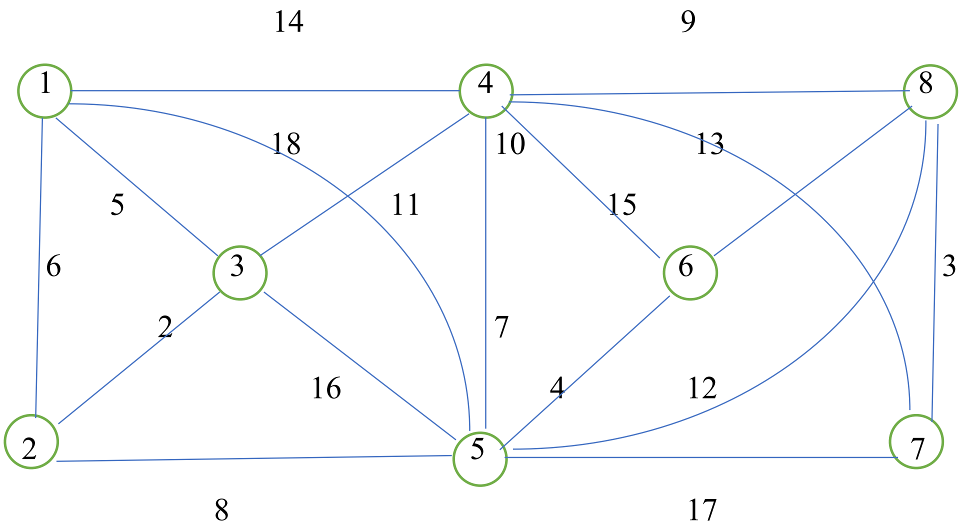

Fig. 3.6

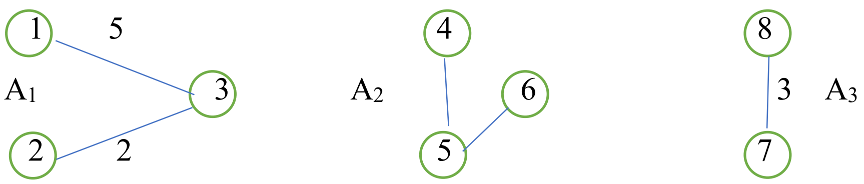

Fig. 3.7

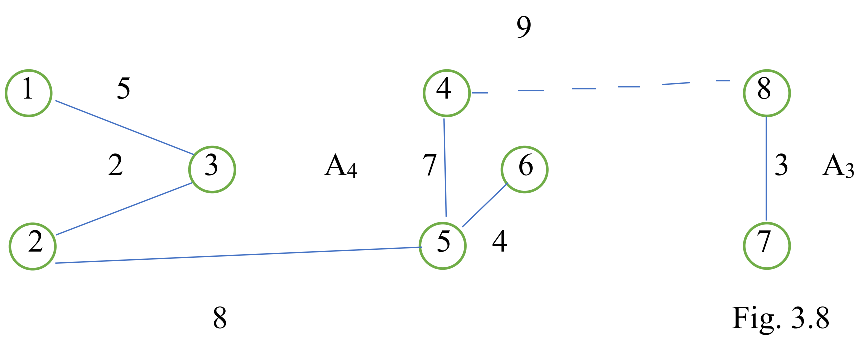

Fig. 3.8

As a result, we obtain the graph $G_2(V, U_2)$ in Figure 3.8, which has the components A4 and A3. Since there are just two of them, one is the other's closest neighbour. As a result, we connect them using the dotted arc (v_4, v_8), which shows the distance between them. The shortest spanning tree is thus produced, which is a single connected graph. The tree measures 38 in length.

We now prove the algorithm.

Every vertex in G has a single nearest neighbour since $G(V, U)$ is connected and no two arcs have the same length. Every vertex can be connected to its neighbour to create the component-rich graph $G_1(V, U_1)$. Another graph with components, $G_2(V, U_2)$, is created when these components are connected to each other to their nearest neighbours. After doing this process once again, we eventually obtain a graph $G_p(V, U_p)$with a single component; as a result, the finished result is a connected graph.

There are no cycles on this graph. For, if possible, let it have a cycle $(v_a, v_b, v_c, \ldots, v_d, v_a)$. Let's draw arrows that point to each vertex's closest neighbour on the arcs in this cycle (keep in mind that they were initially undirected). For example, we would mark the arrow from v_b to v_c if v_c was v_b's closest neighbour. In this manner, an arrow will be placed on each arc in the cycle. For example, if (v_a, v_b) does not receive an arrow, then neither v_b nor v_a are the closest neighbours of v_a and v_b respectively , and so the arc (v_a, v_b) should not have existed at all.

One of the following two scenarios will occur once the cycle's arcs have all been marked.

(1) Every arc is labelled with the same sense. Let us suppose they have been marked $as\ v_a \rightarrow v_b, v_b \rightarrow v_c, \ldots, v_d \rightarrow v_a$. Then since v_c and not v_a is the nearest neighbour of v_b ,$x_{ab} > x_{bc}$. Similarly for other vertices; so we find that

$$x_{ab} > x_{bc} > \ldots > x_{da}.$$

But $x_{ab} > x_{da}$ means that the nearest neighbour of $v_a$is not v_b. This is a contradiction. Therefore all the arcs cannot have arrow marks in the same sense.

(2) A few arcs have one sense indicated, while others have the opposing sense. The arrows pointing to a vertex, let's say v_b, must then be pointed away from v_b in a path of its neighbours, v_a andv_c . This would imply that v_b is closest to both v_c and v_a. This is not feasible due to the disparity in length between the arcs. Thus there is a contradiction as well in this situation.

Therefore, we draw the conclusion that the assumption of a cycle's existence is false because it always results in a contradiction. For this reason, the graph $Gp(V, U_p)$ is cycle-free.

As a result, we demonstrate that the graph produced by this technique is a tree ; that is a connected graph without cycles. It spans G since it has every vertex in G.

Let us assume, if possible, that the smallest spanning tree $T(V, U')$ is not the same as $Gp(V, U_p)$, in order to demonstrate that it is the smallest spanning tree. In order for the two to differ, there needs to be an arc in Gp that is not in T. Assume that this arc is(v_a, v_b), with v_b being v_a's closest neighbour. Allow us to present these arcs in T. Because every extra arc in a tree creates a cycle. Let $(v_a, v_b, v_c, \dots, v_d, v_a)$ be the cycle. The length of the arc (v_d, v_a) is bigger than that of (v_a, v_b) since v_b is v_a's closest neighbour. Thus, the length of the resulting tree will decrease if we remove the arc (v_d, v_a) from T and add (v_a, v_b), which would indicate that T is not a minimum length, contradicting the hypothesis. Thus, there isn't a Gp arc that isn't in T. However, since both trees are spanning the same graph and should have $n - 1$ arcs, where n is the number of vertices in G, the total number of arcs in Gp is the same as in T. Consequently, Gp and T, G's minimum spanning tree, are equal.

Proved.

CHAPTER 4

TRANSPORTATION AND ASSIGNMENT PROBLEMS

In this section, we will examine a few linear programming problems with unique mathematical structures. These can be solved using the simplex approach, which is a general strategy for handling LP problems. However, the solution to their problem has been found in simpler methods thanks to their unique properties. Due to the prevalence of a sizable number of physical scenarios whose mathematical formulation either fits into or can be forced to fit into these unique structures, these issues have gained significant attention and been given unique names, such as the assignment problem and the transportation problem. Although the titles do point to the physical circumstances in which the issues most plainly occur, they actually refer to specific types of mathematical models rather than any physical circumstances at all.

4.1 TRANSPORTATION PROBLEMS:

Finding the best way to move resources to different locations and from one site to another while minimizing costs is the focus of a transportation problem, which is a type of linear programming problem.

To put it simply, the primary goal of the transportation problem is to convey resources at the lowest possible cost from the source to the destination.

- Another name for it is the Hitchcock Problem.
- One commodity must be transported from "m" sources (origin) to "n" destinations.
- Assumptions: Every source's supply level and every destination's demand are known.
- Objective: To reduce the overall cost of transportation.

LINEAR PROGRAMMING FORMULATION:

We define x_{ij} as the quantity sent from warehouse i to market j in order to simplify the transportation problem into a linear program. Given that i can assume values from

$1, 2, \ldots m$ and j from $1, 2, \ldots n$, the product of m and n determines the number of decision variables**.**

Minimize:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} x_{ij} \qquad \text{(total cost of transportation)}$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} \leq a_i \qquad \text{(supply restriction at warehouse i)}\ for\ i = 1, 2, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} \geq bi \qquad \text{(demand requirement at market j)}\ for\ i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{(non negative restrictions) for all pairs } (i, j)$$

The constraints imposed by supply ensure that no warehouse ships more than it can hold. Demand constraints ensure that the overall quantity sent to a market satisfies the market's minimal demand. There are $(m + n)$ constraints in total, excluding the nonnegativity constraints. It is clear that the demands of the market can only be satisfied if and when the whole supply at the warehouses equals the total demand at the marketplaces.

Put otherwise,

$$\sum_{i=1}^{m} a_i \geq \sum_{j=1}^{n} b_j.$$

Every supply that is available in the warehouses will be dispatched to satisfy the minimum demands at the marketplaces when the total supply and demand are equal. All of the supply and demand constraints would then turn into strict equations in this situation.

And we have a standard transportation problem given by,

Minimize: $z = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} x_{ij}$

Subject to: $\sum_{j=1}^{n} x_{ij} = a_i \quad for\ i = 1, 2, \ldots, m$

$\sum_{i=1}^{m} x_{ij} = b_i \quad for\ j = 1, 2, \ldots, n$

$x_{ij} \geq 0 \quad for\ all\ (i, j)$

We will create a method solely for resolving a typical transportation issue. This indicates that before a nonstandard transportation problem where supplies and demands are out of balance can be solved, it has to be transformed into a standard transportation problem. A mock market or dummy warehouse can be used to accomplish this conversion, as demonstrated below:

1.Examine an imbalanced transportation situation in which the total supply surpasses the total demand. This is turned into a regular problem by creating a phony market to take in the surplus product that is kept in the warehouses. As there is no actual physical movement of goods and the dummy market doesn't exist, the unit cost of shipping from any warehouse to the dummy market is believed to be 0. Therefore, the following standard problem is similar to the unbalanced transportation problem:

Minimize: $z = \sum_{i=1}^{m} \sum_{j=1}^{n+1} C_{ij} x_{ij}$

Subject to: $\sum_{j=1}^{n+1} x_{ij} = a_i \quad for\ i = 1, 2, \dots, m$

$\sum_{i=1}^{m} x_{ij} = b_i \quad for\ j = 1, 2, \dots, n, n+1$

$x_{ij} \geq 0 \quad for\ all\ (i, j)$

Where $j = n + 1$ is the dummy market with demand $b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$, and $c_{i,\, n+1}$ = 0 for all $i = 1, 2, \dots, m$. Note that the value of $x_{i,n+1}$ denotes the unused supply at warehouse i.

2. Now consider the situation in which total demand surpasses total supply. It may be desirable to identify a least-cost shipping plan that will provide the markets with as much as feasible, even in cases where all of the demands cannot be satisfied. In this case, a fake warehouse is built to fill the void. The corresponding standard problem turns into

Minimize: $z = \sum_{i=1}^{m+1} \sum_{j=1}^{n} C_{ij} x_{ij}$

Subject to: $\sum_{j=1}^{n} x_{ij} = a_i \quad for\ i = 1, 2, \dots, m, m+1$

$\sum_{i=1}^{m+1} x_{ij} = b_i \quad for\ j = 1, 2, \dots, n$

$x_{ij} \geq 0 \quad for\ all\ (i, j)$

Where $i = m + 1$ denotes the dummy warehouse with supply $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$, and $c_{m+1,j} = 0\ for\ all\ j = 1, 2, \dots, n$. Here x_{m+1}, j denotes the amount of storage at market j.

The classic transportation problem has a significant feature: it can be expressed as a table that shows the values of all the data coefficients (a_{ij}, b_{ij}, c_{ij}) related to the problem. It is actually possible to read the transportation model's objectives and constraints straight from the table. On Table 4.1, there is a transportation table for three warehouses and four markets.

By simply equating the total of all the variables in each row to the warehouse capacities, the supply constraints may be determined. In a similar manner, the market demands are equal to the total of all the variables in each column to determine the demand restrictions.

In theory, we will use the more straightforward linear programming approach to solve the transportation problem.

However, the unique form of the transportation matrix all of the variable coefficients are either 0 or 1 provides straightforward guidelines for selecting a non basic variable or eliminating a basic variable.

	Markets				
	M_1	M_2	M_3	M_4	
W1	X_{11} C_{11}	X_{12} C_{12}	X_{13} C_{13}	X_{14} C_{14}	a_1
Ware houses W2	X_{21} C_{21}	X_{22} C_{22}	X_{23} C_{33}	X_{24} C_{44}	a_2
W3	X_{31} C_{31}	X_{32} C_{32}	X_{33} C_{33}	X_{34} C_{34}	a_3
Demands	b_1	b_2	b_3	b_4	

Table 4.1

FINDING AN INITIAL BASIC FEASIBLE SOLUTION:

There are $(m + n)$ constraints and (mn) variables in the typical transportation problem. Generally speaking, the number of restrictions determines the number of fundamental variables in a basic feasible solution. However, because one of the restrictions in the transportation problem is duplicated, the total number of variables that can have positive values is just (m+n-1). Add up all of the supply restrictions to see this. This gives

$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} a_i$, summing up the demand constraints, we get $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} bj$. Since $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, these two equations are identical, and we have only

(m+n-1) independent constraints.

The unique nature of the transportation matrix makes it very straightforward to identify a workable first solution to launch the simplex approach without the need for artificial variables. We'll provide an example to show this.

EXAMPLE

Think of a transportation problem where there are four markets and three warehouses. Warehouse capacity are as follows: $a_1 = 3, a_2 = 7, and\ a_3 = 5$. Market demands are as follows: $b_1 = 4, b_2 = 3, b_3 = 4\ and\ b_4 = 4$. The following table provides the shipping unit cost:

	M_1	M_2	M_3	M_4
W_1	2	2	2	1
W_2	10	8	5	4
W_3	7	6	6	8

Table 4.2

Since $\sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j = 15$, we have a standard transportation problem. The transportation table is given by

	Markets M1	M2	M3	M4	
W1	X_{11} 2	X_{12} 2	X_{13} 2	X_{14} 1	3
Ware houses W2	X_{21} 10	X_{22} 8	X_{23} 5	X_{24} 4	7
W3	X_{31} 7	X_{32} 6	X_{33} 6	X_{34} 8	5
Demands	4	3	4	4	

Table 4.3

basic feasible solution to this example will have at most six (i.e., $3 + 4 - 1$) positive variables. There are a number of ways to find an initial basic feasible solution.

4.2 ASSIGNMENT PROBLEMS

Machines $M_1, M_2, \ldots, M_n$ represent the n machines in a particular machine shop. For these machines, a set of n distinct jobs $(J_1, J_2, \ldots, J_n)$ must be assigned. The machine to which a job is assigned determines the machining cost for that job. The cost of performing a job on machine i is represented by c_{ij} . (If a specific job cannot be done on a machine, we set the appropriate c_{ij}, to a very big amount). Every machine is limited to one task at a time. The challenge lies in allocating tasks to the machines in a way that minimises the overall cost of machining.

A simplistic method of addressing this issue would be to list every possible assignment of jobs to machines. The best assignment is determined by calculating the total cost for each assignment and selecting the one with the lowest cost. Given that there are n alternative assignments, this will be an inefficient and costly strategy! There are 3,628,800 potential assignments, even for $n = 10!$

In order to solve this using linear programming, first define

$$X_{ij} = \begin{cases} 1 & \text{if job } j \text{ is assigned to machine } i \\ 0 & \text{otherwise} \end{cases}$$

We have since every machine is specifically designated for a certain task.

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad for\ i = 1, 2, \dots, n$$

In the same way, a single machine is designated for each job.

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad for\ j = 1, 2, \dots, n$$

The objective is to minimize

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

The above is actually the formulation of a standard transportation problem with n warehouses and n markets where $a_i = 1,\ for\ i = 1, 2, \dots, n$, and the demand $b_j = 1$, $for\ j = 1, 2, \dots, n$. The corresponding transportation matrix is given below:

Jobs

	J_1	J_2	. . . .	J_n	
M_1	C_{11}	C_{12}		C_{1n}	1
M_2	C_{21}	C_{22}		C_{23}	1
.					.
.					.
.					.
Mn	C_{n1}	C_{n2}		C_{nn}	1
	1	1	. . .	1	

Table 4.4

Non standard assignment problems

Think of a machine shop that has M machines and N jobs. We construct dummy jobs or dummy machines in order to turn this into a (equivalent) standard assignment problem with an equal number of jobs and machines.

Assuming $M > N$, we have more machines than jobs. Next, in order to have M machines and M jobs, we generate $(M - N)$ dummy jobs. In order to maintain the objective function, we set the dummy job's machining costs to zero. When a machine is given a dummy job, it remains inactive. Comparably, some jobs cannot be assigned when there are more machines than jobs $(N > M)$. In this instance, $(N - M)$ dummy machines are made, and every job will have a zero machining cost.

Hungarian method

The method's name comes from the Hungarian mathematician König, who discovered a mathematical feature that allowed for a more effective solution to the assignment problem. For the purposes of the Hungarian technique, we'll assume that cost element (c_{ij}) is nonnegative. The fundamental idea behind the approach is that changing any row or column of the typical assignment cost matrix by a constant will not change the optimal assignment. For example, if the cost of doing any job on machine 1 is lowered by \$k thenthe assignment problem's objective function becomes

Minimize: $z = \sum_{j=1}^{n}(c_j - k)x_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij}$

$= \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij} - \mathrm{k}\sum_{j=1}^{n} x_{ij}$

But since machine 1 has to be precisely assigned to one of the jobs, $\sum_{j=1}^{n} x_{ij}$ equals 1. Thus, the newly defined objective function is

Z = (original objective) - k

We know that the objective function of a linear program, remain unchanged when a constant is added to it.

Essentially, the process of finding a solution involves deducting sufficiently large cost from each row or column such that an optimal assignment can be identified through inspection. In order to find the smallest element, we start the process by looking at each row (column) of the cost matrix. After that, this value is deducted from each and every element in that row (column). This generates a cost matrix where each row (column) has at least one zero element in it. Try to create a workable assignment now by using the costless cells. We have the best potential task if it is feasible. This is due to the fact that the goal function's minimum value $\sum_i \sum_j c_{ij} x_{ij}$ cannot be less than zero and the cost elements (c) are nonnegative. Therefore, the optimal must be one that costs zero.

EXAMPLE

When the following table indicates the cost of assignment, determine the optimal assignment of four jobs and four machines:

	J_1	J_2	J_3	J_4
M_1	10	9	8	7
M_2	3	4	5	6
M_3	2	1	1	2
M_4	4	3	5	6

Table 4.5

solution

We are faced with a standard assignment problem because the number of jobs and machines is equal. By initially examining at the rows, subtracting can be used to get a reduced cost matrix.

1. 7 from the first row,

2. 3 from the second row,

3. 1 from the third row, and

4. 3 from the fourth row.

The resulting cost matrix is as follows:

	J_1	J_2	J_3	J_4
M_1	3	2	1	0
M_2	0	1	2	3
M_3	1	0	0	1
M_4	1	0	2	3

Table 4.6

using only the cells with zero costs,(M_j)is a feasible assignment based on the given table. As a result, this is an optimal assignment. Generally speaking, finding a feasible assignment using cells with zero cost might not always be possible . Take a look at the following assignment problem to demonstrate this:

EXAMPLE

Using the following cost matrix, determine the optimal solution to an assignment problem:

	J_1	J_2	J_3	J_4
M_1	10	9	7	8
M_2	5	8	7	7
M_3	5	4	6	5
M_4	2	3	4	5

Table 4.7

Solution

Each row's minimal element is first deducted from all of the row's elements. provides the reduced-cost matrix that follows:

	J_1	J_2	J_3	J_4
M_1	3	2	0	1
M_2	0	3	2	2
M_3	1	0	2	1
M_4	0	1	2	3

Table 4.8

It is not possible to complete this feasible assignment with just cells with zero costs because machine M_2 and M_4 both have zero costs that correspond to job J1. Subtract the fourth column's minimal element from each of the column's elements to obtain more zeros.

	J_1	J_2	J_3	J_4
M_1	3	2	0	0
M_2	0	3	2	1
M_3	1	0	2	0
M_4	0	1	2	2

Table 4.9

A feasible assignment is still impossible since the zero cells can only be used to allocate three jobs. In these situations, the process selects a minimum number of rows and columns and draws lines through them so that the lines cover all of the cells that have zero costs. The maximum number of jobs that can be allocated using the zero cells equals the minimum number of lines required.

In our case, three lines can accomplish this in the manner shown below.

	J_1	J_2	J_3	J_4
M_1	3	2	0	0
M_2	0	3	2	1
M_3	1	0	2	0
M_4	0	1	2	2

Table 4.10

Choose the smallest element that is not covered by the lines. For us, it is 1. Take this amount and deduct it from all the uncovered elements. After that, add this number to each and every covered element at the point where two lines overlap. This yields the reduced cost matrix that is shown below.

	J_1	J_2	J_3	J_4
M_1	4	2	0	0
M_2	0	2	1	0
M_3	2	0	2	0
M_4	0	0	1	1

Table 4.11

Keep in mind that the process described above is equal to adding element I to the first column of the cost matrix and subtracting it from the second and fourth rows. As a result, the optimal assignment remains unchanged once more.

Now that a feasible assignment is achievable, and an optimal solution is to assign

$M_1 \rightarrow J_3,\ M_2 \rightarrow J_1,\ M_3 \rightarrow J_4$ and $M_4 \rightarrow J_2$. The equation $7 + 5 + 5 + 3 = 20$ gives the total cost. A different, equally good option is $M_1 \rightarrow J_3,\ M_2 \rightarrow J_4,\ M_3 \rightarrow J_2\ and\ M_4 \rightarrow J_1$. If a feasible set could not obtained at this stage, the process of drawing lines to cover the zeros must be repeated until a feasible is reached.

CONCLUSION

The discipline of operations research is relatively recent, having emerged during World War II and gained global recognition. One of the few nations that used operations research from the beginning was India. Operations research is successfully applied in business, government, and industry in addition to military and army operations. These days, practically every field uses operations research.

It is challenging to define operations research since its parameters and subject matter are ambiguous. In order to select potential alternative courses of action, the subject areas of economics, engineering, mathematics, statistics, psychology, and so on give the instruments for operations search. Linear programming, non-linear programming, dynamic programming, integer programming, Markov process, and queuing theory etc. are some of the operations research tools and approaches.

Applications for operations research are numerous. In a similar vein, it has several drawbacks, most of which are associated with the costs, time, and difficulties associated with model making. Daily operations research is becoming more and more popular since it increases manager's ability to make good decisions. Operations research is used to make decisions in almost every aspect of company.

REFERENCE

1. K. V. Mital, C. Mohan *Optimization Methods in operations research and systems analysis*. New Age International, 1996.
2. Ravindran, A., Don T. Phillips, and James J. Solberg. "Operations research: principles and practice." (1987).
3. Hillier, Frederick S., and Gerald J. Lieberman. *Introduction to operations research*. McGraw-Hill, 2015.
4. Duckworth, Walter E. *A guide to operational research*. Springer Science & Business Media, 2012.
5. Barnhart, Cynthia, and Gilbert Laporte, eds. *Handbooks in operations research and management science: Transportation*. Elsevier, 2006.
6. Hira, D. S. *Operations research*. S. Chand Publishing, 1992.
7. Murthy, P. Rama. *Operations research (linear programming)*. bohem press, 2005.
8. Marlow, William H. *Mathematics for operations research*. Courier Corporation, 1993.
9. Coyle, Doug, et al. *Application of operations research to funding decisions for treatments with rare disease*. Springer New York, 2013.
10. Kasana, Harvir Singh, and Krishna Dev Kumar. *Introductory operations research: theory and applications*. Springer Science & business media, 2013.