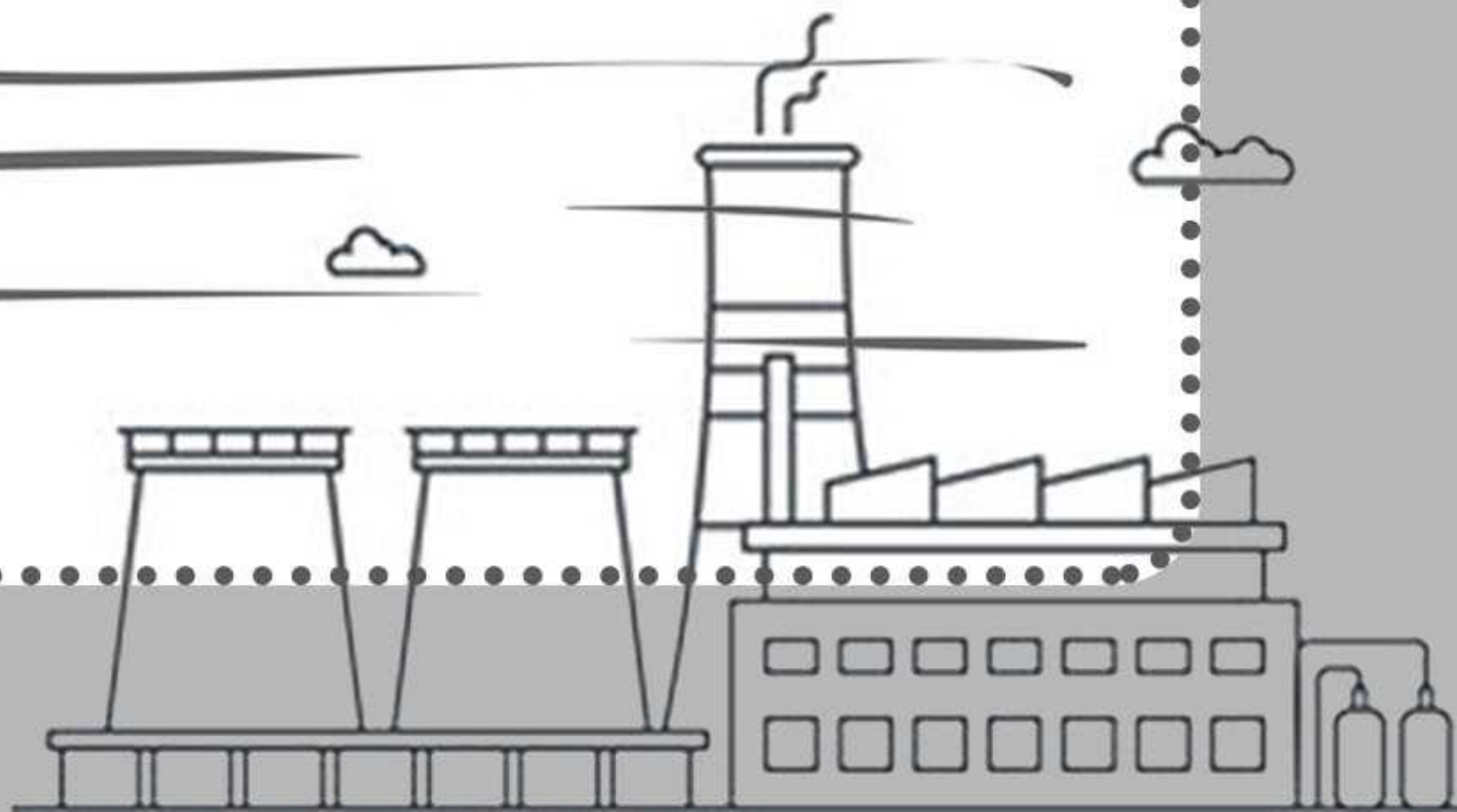


WEEKLY NOTEBOOK

Logistics

second semester
2021



* Topic 1: Introduction to Logistics and SC ::

Overview ::

1. What is the relationship between Supply Chain Management and Logistics System Management?
2. What is the meaning of Common Supply Chain and logistics terminology?
3. What are the main activities of a Logistics System?
4. What are the objectives and decisions in Logistics Systems?

Concept 1 ::

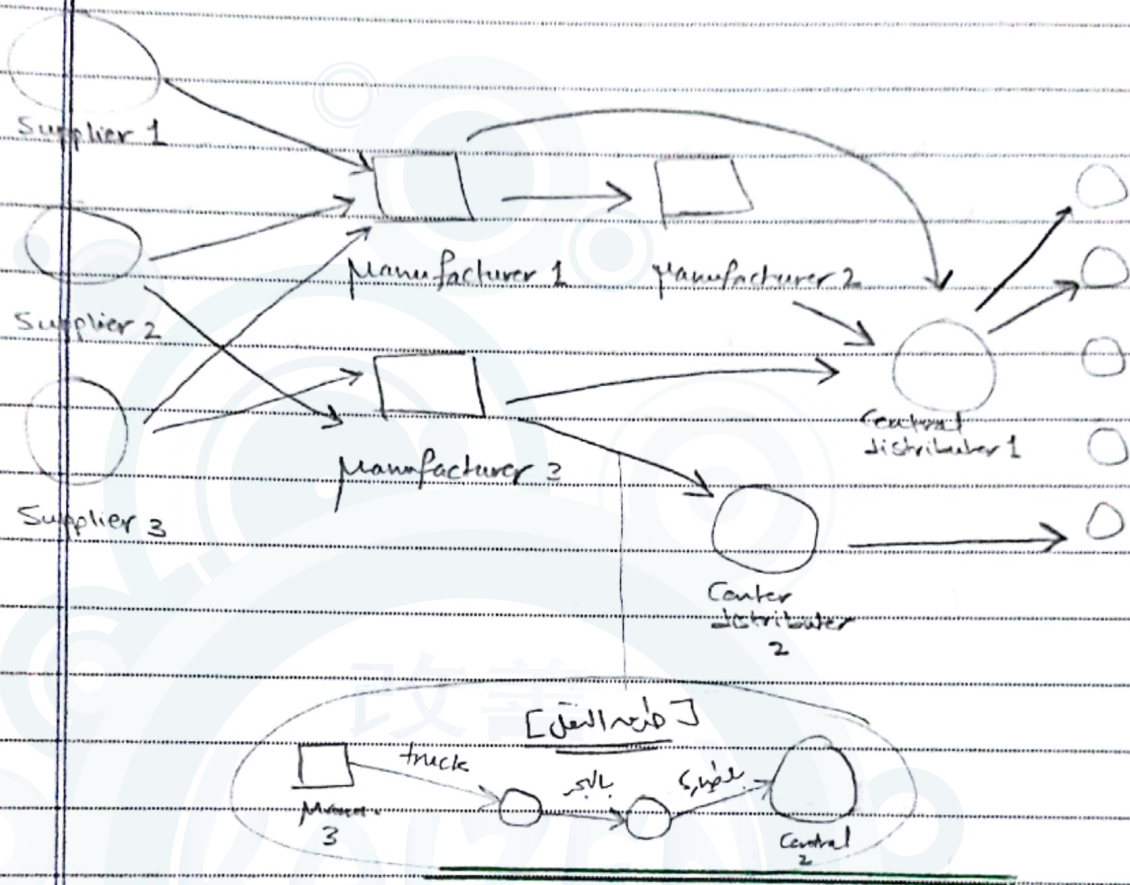
- Relationship ?!

↳ Logistics System is a Subsystem of the Supply chain.

- Supply Chain definition ?

↳ A Complex Logistics System in which raw materials are converted into finished products and then distributed to the final Users.

What is Supply Chain:-



Examples of Supply Chain:-

- Hewlett-Packard
 - Creating a product: Laser printers.
 - Raw Inputs: ink Cartridges, print heads, plastic.
- Delta Airlines
 - Delivering a Service: air travel bt. Cities.
 - Raw Inputs: Airplanes, Foods, Information.
- Amazon: Com
 - product or service?
 - fulfillment by Amazon.

What is Supply Chain Management?

- Council of Supply Chain Management professionals (CSCMP) Definition:-

* Supply chain management, def:-

Encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities.

- Key: Integrates Supply and demand management across Companies and functions within a Company.

What is Logistics:-

- A Continuous link that is always in Sync.
- Technology knows right where everything goes.
- parts for the line come precisely on time.

* ERP: Enterprise Resource planning

→ (Main Business) ←

Processes

What are Logistics Systems:-

Logistics System, def:-

↳ a set of facilities linked by transportation services.

Facilities:-

↳ Where materials are produced.

Transportation Services:-

↳ Move Material between facilities Using Vehicles and Equipment.

- Examples of Logistics in Services:

• Garbage Collection. (public) مصارف النفايات

• Mail delivery. توصيل البريد

Logistics ← إدارة سلسلة التوريد

Logistics ← إدارة سلسلة التوريد

What is Logistic management?

• Council of Supply Chain Management Professionals (CSCMP) definition:-

Logistics Management, Def:-

↳ That part of Supply Chain Management that plans, implements, and Control the efficient, effective forward and reverse flow and storage of goods, services and related information between the point of Origin and the point of Consumption in order to meet Customer's requirements.

Effective and Efficient Logistics Systems:-

• Key to the bottom line profitability of most firms.

• Effective: Customer Satisfaction How well does the Logistics System accomplish what the firm promises to do?

↳ E.g., Mail Service provider (USPS, FedEx, etc.) 2-3 day priority mail.

- What is an appropriate measure of the effectiveness of the priority mail Service?

a Measure - How many times the Company send emails as it is promises?

- Efficient: How well are firms resources used to achieve what the Company promises to do?

↳ - What did it cost to sort and deliver the packages?
 ↳ or time or fuel cars...

* Significance and State of Logistics:-

↳ Slide 14.

Assorted Supply Chain Concepts:-

الزمن
Supply Chain

• Pull / Make-to-order (MTO) Supply Chain:-

↳ Finished products manufactured only when Customer requires them.
 ↳ This products * القالب والمواد

- Little to no Inventories needed.

- Used when lead times short and/or products expensive and/or demand is highly variable.
 ↳ Services * Sij *

- Ex. Aircraft.
 ↳ Sij * Sij *

• Push / make-to-stock (MTS) Supply Chain:-

↳ production and distribution decisions based on forecasts.

- Inventories held in warehouses and at retailer.

- Appropriate when lead times long, products inexpensive, and demand easy to predict.
- **Ex.** Consumer electronics. Inventories low.

• Make-to-Assembly (MTA) Supply chain:

↳ Components and semi-finished products assembled in push-based manner and final assembly is pull-based.

- Sometimes, last minute Customization / product changes performed by distributor.

- **Ex.** Fast food Sandwiches.

• Product and Information flows in Supply Chain:

- Products flows forward from Suppliers to Customers. → Money : Backward

- Information flows in reverse from Customers to Suppliers. → Feedback.

- No flows instantaneous.

- **Note:** be aware of other types of flows and the reverse Logistics.

- Degree of Vertical Integration:

- A Supply chain is fully vertically Integrated If its Components [Raw material Sources, plants, transportation System, etc.] all belong to a single firm.

- 3rd party Logistics (3PL) :-

- Long-term Commitment to use an outside Company to perform all or part of a Company's product distribution.
- Allows Company to focus on its Core Competency.

- Retailer-Managed versus Vendor-Managed resupply:

- Retailer-Managed resupply:-

↳ Retailer in charge of Monitoring Inventory and placing orders for replenishment as needed.

- Vendor-Managed resupply:-

↳ vendors monitor Consumption and Inventories through electronic data Interchange (EDI) and decide how and when to replenish.

* Retailer → Customer المندرج والمستهلك

* Vendor → المورد → المورد البقالة

Concept 2 Refresher:-

Which type of Supply chain requires the least amount of Inventory to be held?

↳ Make-to-Order. → No Inventories.

* Stock → push

* assembly → Inventory in the

Concept 3:-

What are the main activities of a Logistics System?

↳ The main activities of Logistics systems:-

[1] Order processing

[2] Inventory Management.

[3] Freight transportation. التي هي

Order processing:-

- Includes Order transmission, verification of product availability and Customer's Credit Status, keeping Customer informed of status of Order.
- Bar Code technology, electronic data Interchange (EDI) Simplifying process.

Inventory Management :-

- **Inventories, Def:-** Stockpiles of goods waiting to be manufactured, transported, or sold.
- **Types of Inventory:-**
 - Work in Process (WIP) ^{→ any thing inside process or facility}
 - In-transit Inventory. ^{→ Inventory while in transit}
 - Finished products Stocked prior to being sold.
 - Finished products stocked by end-users for future needs.

Manufacturer ↔ Customer

Reasons to hold Inventories:-

- (1) Improve Service level. ^{→ تحسين مستوى الخدمة}
- (2) Reduce transportation Costs. ^{→ تقليل التكاليف}
- (3) Cope with Randomness [demand, lead time]. ^{→ التعامل مع العشوائية [الطلب، وقت التسليم]}
- (4) Speculate on price patterns. ^{→ التكهن بأنماط الأسعار}

Cost of Inventory:-

- Opportunity Cost.
- Warehousing Space. ^{← Rent or own}

* Service level → to be fulfilled with order of times

Inventory Management :-

PPC → Inventory Management, def :- determination of Stock levels in order to Minimize total Operation Cost while Satisfying Customer Service requirements.

• Policy should Consider :-

- ① Relative Importance of Customers.
- ② Economic Significance of different products.
- ③ Transportation policies.
- ④ Production process flexibility.
- ⑤ Competitors policies.

↳ (Q, R), (T, S), (s, S)

Freight Transportation :-

• To be discussed in detail :-

- Costs.
- Transit Time
- Reliability
- Mode choice.
- Etc...

Concept 3 Refresher:-

- True/False: The Four main activities of Logistics Systems are:-
 - Order processing
 - Marketing X $\rightarrow$ Not one of them.
 - Inventory management
 - Freight transportation.

Concept 4:- Three main Objectives and decisions in Logistics Systems:-

- Capital Reduction: Level of Investment in Logistic System.
- Cost Reduction: minimize total Cost associated with transportation and storage.
- Service level Improvement:- lead time b. order placement + receipt of goods.

Distance $\rightarrow$ rate
 time $\rightarrow$ cost
 Cost $\rightarrow$

Strategic Level



Tactical Level



Operation Level

Long term 5-10
Years or more

Medium term
>1-5 Years

Short term
Day-to-day,
< 1 Year

- Strategic Level:-

- Number of Facilities.
- Location of Facilities.
- Capacity of Facilities.
- Resources. → to fulfill the order.

- Tactical Level:-

- Flow of materials
- Resource allocation
- Capacity allocation
- Inventory policies.

- Operation Level:-

- Scheduling
- Routing
- Truck loading
- Warehousing
- Tracking/tracing.

* Decisions Levels in Logistics *

Supply Chain Management activities:-

- All activities required to plan and Operate a Supply network, Including:-

Business { - Relationship Management.

- Finance

- Product Marketing

PPC { - SC inventory Management.

- Inventory Control

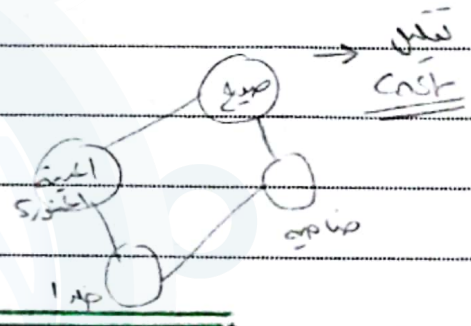
- Production planning and Scheduling

Facilities { - Facility design.

Logistics { - Transportation Management

- Network design.

MPS, MRP.



"End of Introduction"

* Topic 2:- Costs In Logistics System.

[Transportation Mode Selection]

* Part 1 :-

* Four essential types of Costs:-

1. Transportation.
2. Handling.
3. Holding (Cost of Capital / Opportunity Cost)
4. Facility Rent.

* Normalizing Costs:-

- Cost Per time period.
- Cost per Item.

* Comparing the Cost of Carriers or Modes:-

- **Objective:-** minimize total Cost when
Selecting mode of transportation.
→ often requires trading off inventories
+ transportation costs.

* Inventory Costs, 2 types:-

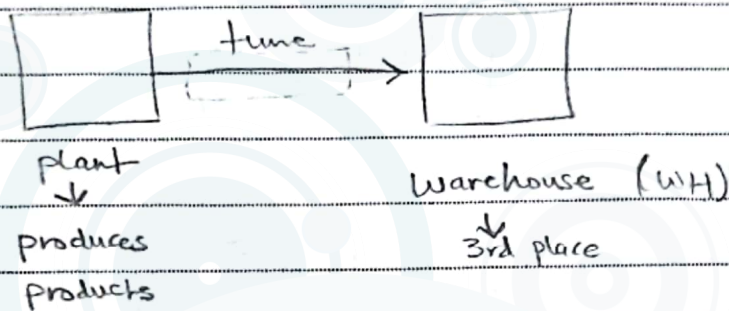
- 1] Holding / Cost of Capital.
 - 2] Rent Cost for space.
- f (value held, Interest rate)

* Inventory held in 3 places:-

[1] At plant

[2] At Warehouse $\rightarrow$ destination

[3] in Transit



Problem :-

A Company needs to select between two transportation modes: ^{1/60} rail and truck, to move goods from plant to warehouse with objective of minimizing total transportation and Inventory Costs. [On an Annual bases].

$\Rightarrow$ transportation cost $\cup$ Inventory Cost.

Data:-

	plant	Warehouse (WH)
Safety Stock:	0	100 Units
Value of Items:	\$500/unit	\$512/unit.
Space Cost:	\$20/Unit . year	\$25/Unit . year

- [Rail] $\rightarrow$ Car Capacity = 400 Units $\rightarrow$ Shipment.
 transit time = 20 days
 Cost per shipment = \$1200

- [truck] → Truck Capacity = 100 Units

Transit time = 3 days

Cost per Shipment = \$700

- production rate at plant 10 Units/day.

- demand at warehouse 10 Units/day.

- Inventory holding Cost rate 25% of value / year held.

* Goal: Determine the optimal shipment size, q (# of Units for each mode) and then the total annual cost associated with these shipment sizes.

Assumption:- Shipper will never ship more than a single car load [rail] or a single truck load [truck].

Steps:-

[1] Determine annual Inventory Space Cost at the plant.

[2] Determine annual Inventory holding Cost at the plant.

[3] " " " Space Cost at the WH.

[4] " " " Holding Cost at the WH.

[5] Determine in-transit Inventory Cost → Rail
↳ truck

* Holding Cost $\rightarrow$ Avg. $\times$ cost

* Space Cost $\rightarrow$ Max. " "

No.

(17)

[6] Determine transportation Costs $\rightarrow$ Rail

$\rightarrow$ truck.

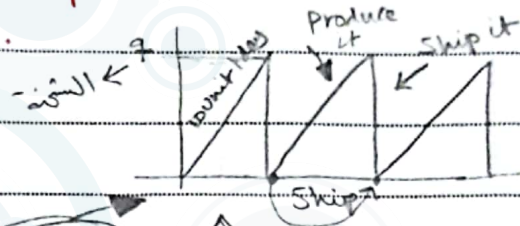
[7] Total annual Cost expression

$\rightarrow$ Rail

$\rightarrow$ truck

[8] Determine Optimal q for each mode.

[9] Choose a mode.



* Solution 2:

(1) Space Cost at plant:-

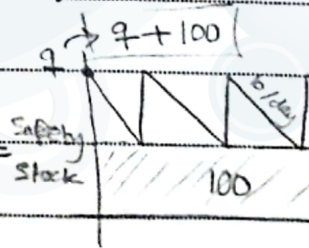
$\rightarrow$ Space Cost = \$20 $\times$ q Unit = \$20 q /Year
max. Inventory $\times$ cost $\times$ $\text{Unit} \cdot \text{Year}$
held at the facility

(2) Holding Cost at plant:-

$\rightarrow$ rate of charge per item of Inventory held at plant for a year = \$500 $\times$ 25%
= \$125 / Unit. Year

$\rightarrow$ Avg. Inventory level = $\frac{q}{2}$

$\rightarrow$ \$125 / Unit. Year $\times$ $\frac{q}{2}$ Units
= \$62.5 q / year.



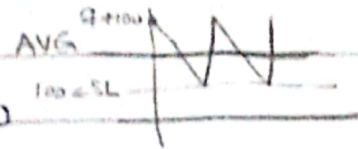
(3) Space Cost at WH:-

$\rightarrow$ Max. Inventory = $q + 100$
level Units

Space Cost = \$25 / Unit. Year $\times$ ($q + 100$) Unit
= (25 q + 2500) \$/year.

④ Holding Cost at WH^r

→ Avg. Inventory = $\frac{Q}{2} + 100$ Units



→ each item held one year is charged at a rate of $512 \times 25\%$
 $= \$128 / \text{Unit} \cdot \text{year}$

Holding Cost in WH = $\frac{\$128}{\text{Unit} \cdot \text{Year}} \times \left(100 + \frac{Q}{2}\right)$

* dimensional Analysis → $\frac{\text{cost}}{\text{unit} \cdot \text{year}}$

→ length (m)

length (m)

⑤ Determine Intransit Inventory Costs

→ Use avg. value of goods at plant and warehouse

Use $\frac{506 \$}{\text{Unit}} = \frac{500 + 512}{2}$



each item charged at $\frac{\$506}{\text{Unit}} \times 25\% / \text{yr}$

$= \$126.5 / \text{Unit} \cdot \text{year}$

→ Rail: each item spends 20 days

- Amount to be shipped at the whole

Year = Production rate $\times$ period

$= \frac{10 \text{ Unit}}{\text{day}} \times \frac{365 \text{ day}}{\text{Year}}$

$= 3650 \text{ Unit} / \text{year}$

(5) \$/unit *

→ 24. Pcs/1 yr *

No.

(19)

$$\begin{aligned}\text{Intransit Cost} &= \text{amount shipped yearly} * \text{intransit time} * \text{intransit holding cost} \\ &= \frac{3650 \text{ Unit}}{\text{Year}} * \frac{20 \text{ days}}{365 \text{ day}} * \frac{1 \text{ yr}}{\text{year}} * \frac{126.5 \$}{\text{unit} \cdot \text{yr}} \\ &= 25300 \$/\text{yr}.\end{aligned}$$

→ Part 2:

Truck: Intransit Cost =

$$\begin{aligned}&\left(\frac{3650 \text{ Unit}}{\text{Year}} \right) * \left(\frac{3 \text{ days}}{365 \text{ days}} * \frac{1 \text{ yr}}{\text{year}} \right) * \\ & (126.5 \$/\text{Unit} \cdot \text{yr}) = 3795 \$/\text{Year}.\end{aligned}$$

(6) Transportation Costs:

* Rail → every Shipment Cost us \$1200
of shipments = $\frac{3650 \text{ Ship}}{\text{yr}}$
(9 → 9 shipments)
5 no 15 10

Transportation Cost Rail = 7

$$\begin{aligned}&\left(\frac{\$1200}{\text{Shipment}} \right) * \left(\frac{3650 \text{ Shipment}}{9 \text{ yr}} \right) \\ &= \frac{4,380,000}{9} \$/\text{Year}.\end{aligned}$$

* Truck:-

Transportation Cost for the truck: =

$$\left(\frac{\$700}{\text{Ship}} \right) \times \left(\frac{3650}{q} \frac{\text{Ship}}{\text{Yr}} \right)$$

$$= \frac{2,555,000}{q} \$/\text{Yr.}$$

(7) Total annual Cost :-

$$\text{TCR}(q) = \frac{\text{terms of } q}{q}$$

total Cost $\swarrow \searrow$ Rail

$$20q + 62.5q + 25q + 2500 + 64q + 12800 + 25300 + \frac{4380000}{q}$$

SI Unit
Optimum

SI Unit

Equation

SI Unit

$$= 171.5q + 40600 + \frac{4380000}{q}$$

Inventory Cost

In transit Cost

truck:-

$\rightarrow$ truck

$$\Rightarrow \text{TCT}(q) = 20q + 62.5q + 25q + 2500 + 64q + 12800 + 3795 + \frac{2555000}{q}$$

$$= 171.5q + 19095 + \frac{2555000}{q}$$

Inventory

transit.

- Draw the equation by Excel sheet:-

q TCT TCF

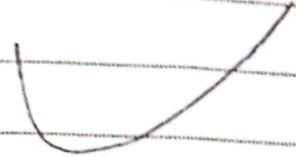
10 5000

5000

20

30

y



x

(2) Find Optimal :- for q for each mode.

$$TCR'(q) = 171.5 + 0 - \frac{4380000}{q^2} = 0$$

$\frac{1}{x} \rightarrow -\frac{1}{x^2}$

$$q_R = \sqrt{\frac{4380000}{171.5}} \approx 160 \text{ Units}$$

* the minimum shipment

→ the min cost possible by Unit by the Rail.

If you ship by the Rail

$$\rightarrow TCR(q_R) = \$95,415 / \text{yr} (\text{min possible}).$$

$$TCT'(q) = 171.5 - \frac{2555000}{q^2}$$

$$q_T = \sqrt{\frac{2555000}{171.5}} = 122 \text{ Units}$$

2 trucks

is not 2555000

$$\text{Shipment Cost} = 2 \times \frac{7008}{\text{Ship}} + \frac{340}{q}$$

$$= \frac{5110000}{q}$$

$$TCT(q_T = 122) = 171.5 \times 122 + 19095$$

$$+ \frac{5110000}{122}$$

الحد الأدنى من التكلفة
التي لا يمكن أن تكون أقل
(12) هو 11) = \$81,903

→ If the maximum truck Capacity was Selected $q_T = 100$.

$$TCT(q_T = 100) = \$61,795$$

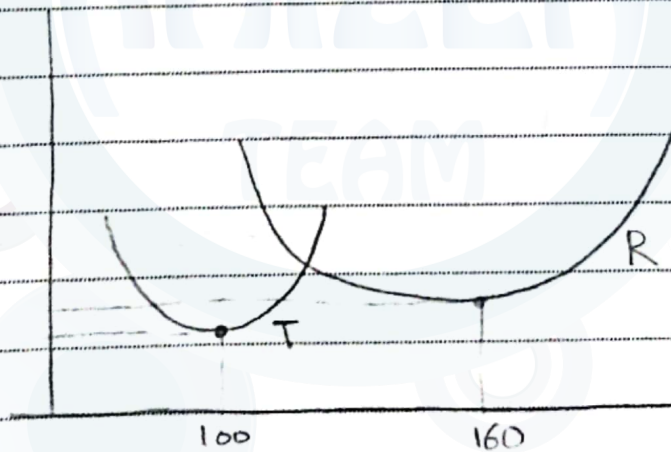
→ Minimum possible Cost

$$TCT(q_T = 99) = 16,881.58$$

251 1 *
100 1 W Cost

101 = ?!

→ 2 Shipments



Truck → Cost Mini ✓

Rail

Shipment	Optimum Inventory Cost	Trans.	Total
160, Rail	68,040	27,375	95,415
400	109,200	10,950	120,150
140	64,610	31,286	95,896

* Transportation Rates and Costs by Mode.

- Fixed Cost: Infrastructure, Right of way, vehicle, equipments

- Variable Cost: (Operating): fuel, Maintenance, driver, ...

Mode	Fixed Cost Components	Var. Cost Components
Water	High	Low
Rail	High	Low
Truck	Low	High
Air	High	High

* $Q_R = 160 < 400 \rightarrow 1 \text{ Car} \rightarrow \text{Cost}_R < \text{Cost}_T \rightarrow \text{Choose Rail}$

* $Q_T = 122 \rightarrow \text{More than 1 truck} \rightarrow \text{Assumption to violated the assumption} \rightarrow 1 \text{ truck}$

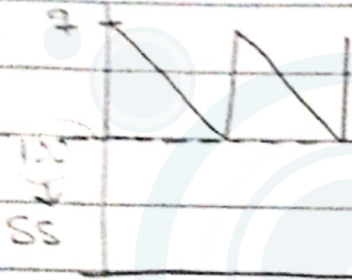
Assumption $\rightarrow Q_T = 122$

transportation cost. $\rightarrow$ jump also for 2 trucks

FIVE APPLE

Short hauls trans part 1:

Back ← Warehouse i-



$$\bullet \text{ Max.} = q + 100$$

$$\bullet \text{ Min.} = 100$$

$$\bullet \text{ Avg.} = \frac{(q + 100) + 100}{2}$$

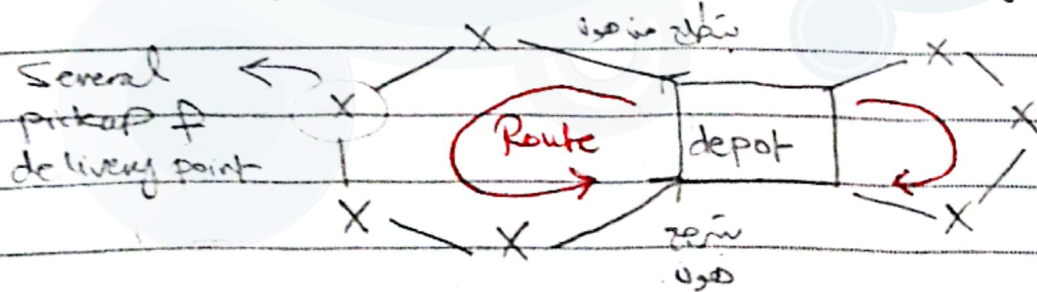
$$= \frac{q}{2} + 100$$

Short Haul transportation:

↳ Pickup, delivery of goods in relatively Small area Using fleet of vehicles.

- Unless Stated Otherwise, vehicles are Stationed at Single depot. → مركز / مركز

- Routes include Several pickup and/or delivery point.
- Routes performed in a single work shift



(depot) 1 مركز، 2 Routes

* Decisions in Short haul transportation:

- **Strategic:-** where to locate the depot / Ware house "terminal".

↳ [Location Problem / Facility location]

- **Tactical:-** fleet sizing (كمية) → How many trucks / cars to have.

- ✓ • **Operational:-** Vehicle Routing and Scheduling problem.

↳ **(VRSP)** → Prob. of routing & logistics is

truck & unit 2/3/4/5/6/7/8/9/10

point & point machine will

split & depot & 2/3/4/5/6/7/8/9/10

⇒ We will focus on the Operational Prob.

- **Imp.:** what data known beforehand?

Set 6 ← • **Static Vehicle Routing problem:-**

تغيير مسارات
الطلب اذا
طلبه من ال
WH
↳ All Customer requests (location, amount of goods) is known in advance.

تغيير مسارات
تغيير مسارات
• **Dynamic VRP:-**

الطلب اذا
الطلب اذا
↳ Not all information is known in advance.

(Routes planned for Set of requests known in advance, dynamically updated to incorporate new requests.)

Static → * إذا كان plan ثابت
الوقت من قبل ال WH

و طرح التوزيع ثابت / ثابت في مكانه

Dynamics → * إذا كان planned
مستمر ، طرح ساس متغير

يصل في حركة غير ال plan
التي كان في وقت سابق "كانت" ال
ملا التوزيع و كان ال plan متغير

الوقت! كمنه في سبب Request

[New Info]: سطر

الطلب إذا كان ثابت - Customer is static

Static ← proposed delivery ال
→ for one depot.

* Transportation problems defined on graphs
which are Symbolic representations of
Networks and their Connectivity.

• → Nodes

↓
depot, place,
WH, plant

Irbid

Jarash

Mafraq

→ Network

Salt

Amman

Madaba

→ Route,
Connection

Graph: $G = (V, E)$

↓ ↘
Set of Vertices (V) or nodes (n). Set of Edges (E) or arcs (A).

Connect them

→ Notation: $G = (N, A)$

Another one.

Network.

*Vertex: terminal point, intersection point of graph.

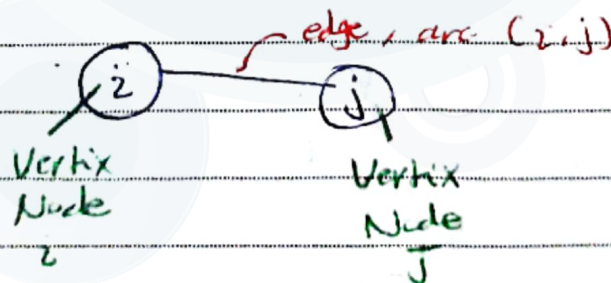
↑
WH, stop, etc



↳ Represents: City, Customer location, depot, road intersection.

*Edges: is the link between two vertices.

↳ edges/Arcs often referred to as (i, j) → Connected Vertex i to Vertex j



depot = first vertex = final vertex.

⇒ Arrow Representation direction:-

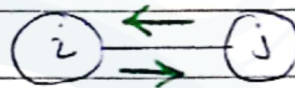
- Head node i Head Tail
- Tail node j $(i) \rightarrow (j)$

(i,j)

directional

Connection.

→ Assumed bidirectional if no arrow



bidirectional

→ Edges / Arcs have Number of Character-
istics.

- Cost to move from i to $j \rightarrow C_{ij}$

↳ Time, distance, expenses, fuel

Cost ⇒ Consumption, Risk, depreciation...

↳ Small, Small Cost ⇒ all are quality of

↳ *Cost $C_{ij} \rightarrow$ Cost (time, Risk, distance) associated with (i,j) .

- Capacity $U_{ij} \rightarrow$ Max. flow on edge (i,j) .

↳ [traffic, amount of goods].

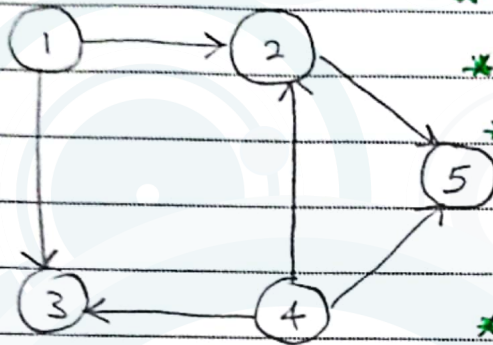
- Order of a graph: Number of Vertices

$|V| \rightarrow |V| \equiv \text{Number}$

$$V = \{1, 2, 3, 4, 5\} \rightarrow |V| = 5$$

$$V = \{\text{Amman, Aqaba, ...}\}$$

- Size of a graph: Number of edges, $|E|$.



$$* G = (V, E)$$

$$* V = \{1, 2, 3, 4, 5\}$$

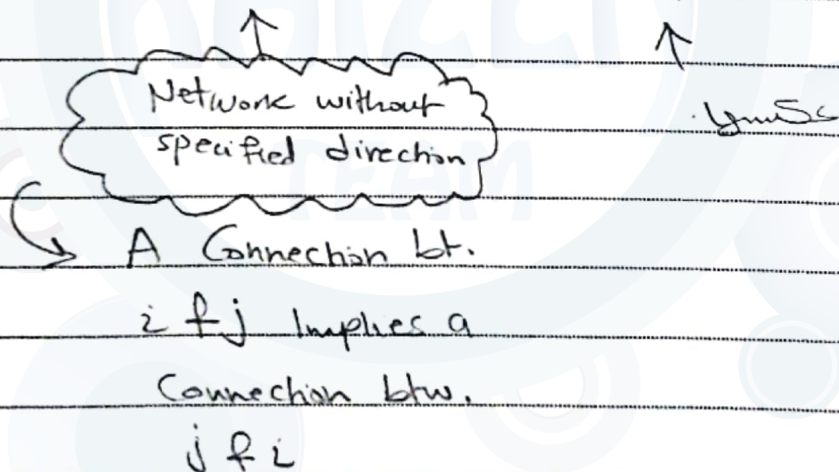
$$* E = \{(1, 2), (1, 3), (2, 5), (4, 3), (4, 5)\}$$

$$* \text{Order of } G = 5 = |V|$$

$$* \text{Size of } G = 6 = |E|$$

- Structural properties of graph:-

- Undirected vs. directed:



Also referred to
the Symmetrical
Network.

In transportation problems, Sometimes Symmetry Used to represent whether cost is same when travelling from i to j as when traveling from j to i .

⇒ A Symmetry example:-

Traffic maybe higher in one direction resulting in higher travel time.

15 min



30 min

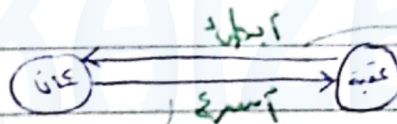
(cost)

Cost depends

⇒ in the direction

from j to i is Cost *

directional j to i is



Cost

Cost $\equiv$ time

* So Answer → Answer →

Cost



Symmetrical
bidirection



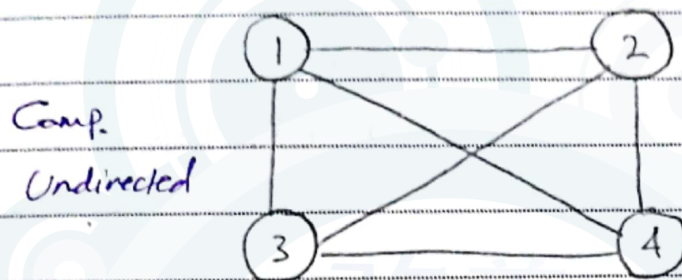
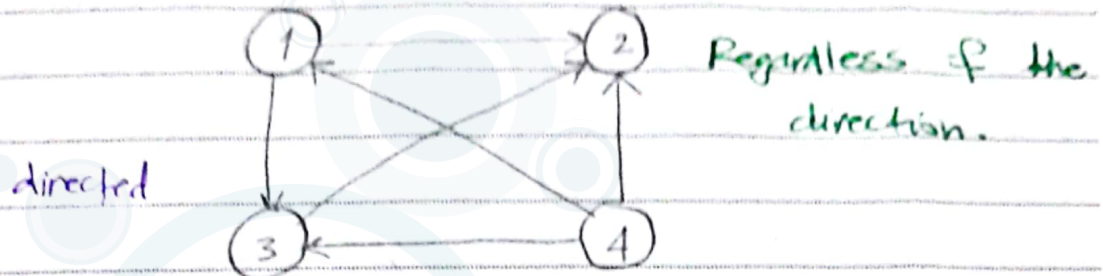
asymmetrical
bidirection.

Symmetrical ← bidirectional

* Symmetrical or asymmetrical = bidirectional

* bidirectional $\neq$ Symmetrical

Completeness:- a graph is Complete if an edge exists between every pair of vertices.



⇒ **Path**:- , Cycles , Trees:-

↳ For the following definitions, assume there are (n) vertices in V , and (m) edges in E .

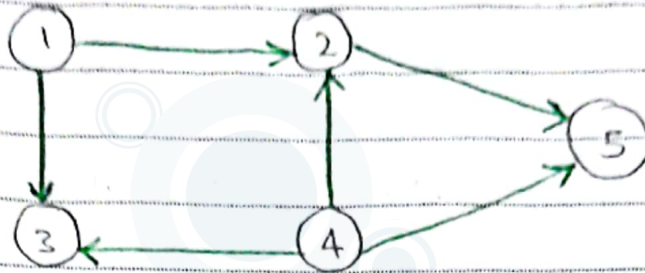
→ **Path**:- a path P in G is a sequence of vertices $\{z_1, z_2, \dots, z_n\}$ such that $z_k \in V \quad \forall k=1, \dots, P$ and z_k Unique $\forall k=1, \dots, P$.

↳ No vertex visited more than once.

• (z_k, z_{k+1}) or $(z_{k+1}, z_k) \in E$

↳ It is Not necessary that the path should be feasible

* Set of edges



Not Completeness, directed Network.

* Path from 1-5:-

↳ 1-2-5 → {1, 2, 5} → Feasible Path

{1, 2, 4, 5} → Unfeasible Path

{1, 3, 5}

{1, 4, 5} } Not a path.

- Directed path:

↳ additional Condition: $(z_k, z_{k+1}) \in E \ (\forall k=1, \dots, p-1)$
 * Which are include path?

is A feasible sequence of decision for users.

[feasible ← path] ← is a sequence of nodes

→ In the previous Example: {1, 2, 5} is a directed path → feasible.

* Related Concept: Connectedness.

↳ Connected: 2 vertices i & j are Connected in G if there exists path from i to j .

↳ Regardless of direction.

edge (pair of
vertices)
Example

* Adjacency: Connected (2 & 4) by a path of length 1.

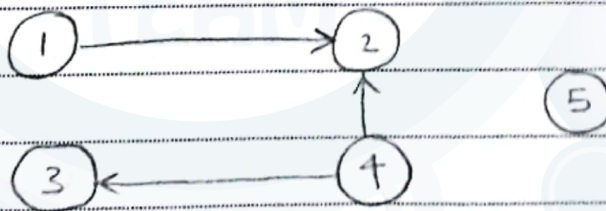


→ nodes 1 and 4 are Connected but not adjacency.

→ nodes 1 and 2 are Connected And adjacent.

* Adjacent $\equiv$ Connected (by default)
but $\rightarrow$ Connected $\neq$ Adjacent

→ A graph is Connected if every pair of vertices in the graph are Connected.

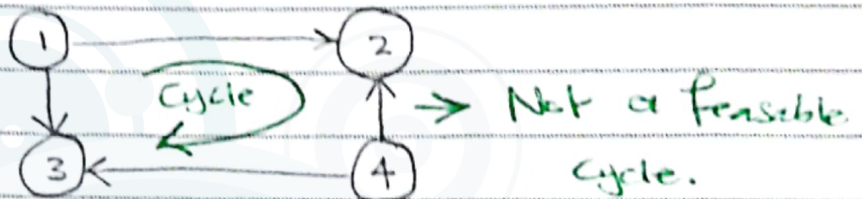


↳ Not Connected

Connected $\leftarrow \underline{5} \approx \text{dps } \underline{2}$ "as well"

Complete graph

* Cycle Set of edges that form a closed loop in G , you can begin walking at some vertex, follow only edges in the cycle regardless of their direction, then return to original vertex.



→ Sequence of vertices $\{z_1, z_2, \dots, z_p, z_1\}$ such that →

- $z_k \in V \quad \forall k=1, \dots, p$
- z_k Unique $\forall k=1, \dots, p$ } like a path.

→ (z_k, z_{k+1}) or $(z_{k+1}, z_k) \in E$
 $\forall k=1, \dots, p-1$

Is it
feasible or
not

• either (z_p, z_1) or $(z_1, z_p) \in E$

↳ still satisfied
• yes, it will

Predecessor

Successor

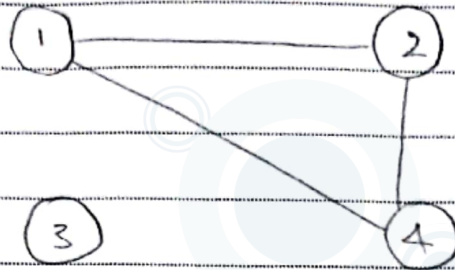


z_1, z_2, z_4, z_3

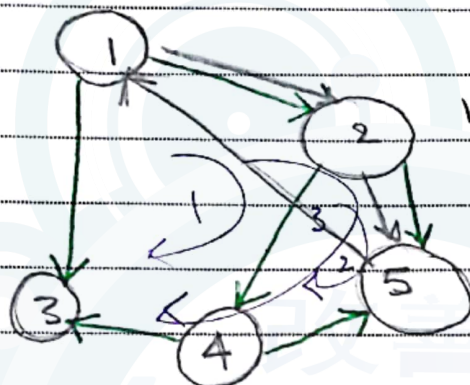
↳ cycle ✓

↳ Not feasible

↓
Directed
Cycle


 $\{1, 2, 4, 1\}$

↳ Undirected Cycle ✓
↳ feasible



1 Cycle, Feasible

3 Cycles, Not feasible.

Path → $\{1, 2, 5\}$

$\{1, 2, 4, 5\}$

$1, 2, 5, 1 \rightarrow$ Cycle

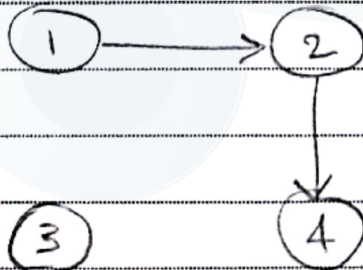
$2, 5, 4, 3, 1, 2 \rightarrow$ Cycle

$1, 2, 5, 4, 2 \rightarrow$ Not a Cycle

$2, 5, 4, 2 \rightarrow$ Cycle. ~~Not~~ Not Feasible

*Not include all the nodes

Path Include a Cycle.



→ $1, 4 \rightarrow$ Connected.

there is a path.

Not adjacent

but $1-2 \rightarrow$ Connected + adjacent

$2-4 \rightarrow$ Connected + adjacent.

شبكة غير متصلة
واحدة من العقد متصلة
عليا بين العقد

max = Visited max *
Sum = Sum Visited

No. (36)

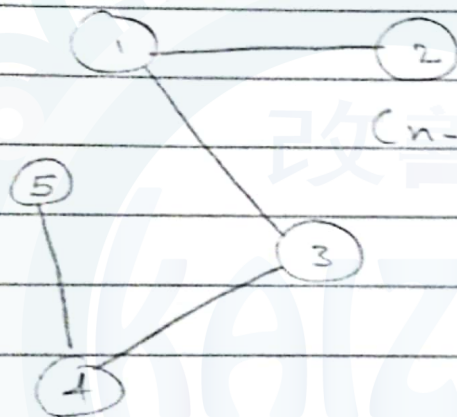
Tree :- is a Collection of edges such that vertices are connected and no cycle exists.

Cycles 2 loop Connected 10/50 Vertices 1/15

→ Equivalent definitions

↳ n vertices in the Network

↳ A tree is a set of $n-1$ edges such that all vertices are connected.



(n-1) edges

1. Loop

2. Node 3

Not Connect.



Subtour

Node 2

Loop 20/10/2020

Not a tree



$n=4$

$m=n-1=3$

∴ tree ✓



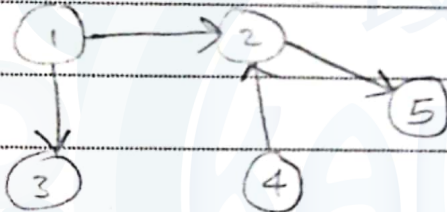
- Important property of a tree:- defines for each pair of vertices, a Unique (Not necessarily directed) path between them.

directed Network

G:



Not a tree



Tree

Undirected Network

G:



Not a tree



trees



trees

- What happens if you remove edge from a tree.

↳ Some Vertex is no longer Connected $\Rightarrow$ graph is Not Connected
 $\Rightarrow$ tree - edge = UnConnected graph

If you add an edge to tree?

↳ Creates an edge → No longer a tree.

Minimum Cost path Models:-

→ Many transportation decision model require finding the min. Cost path(s) between pair(s) vertices.

⇒ Cost Can be time, distance, \$, risk --



Applications:-

- Personal travel: road System (Network) ⇒ determine the minimum time & Path between origin & destination.

✓ Global Supply Chain: Given the Network model representing shipper its full set of transportation services for a particular Commodity, determine the min. Cost (time) transport Option from Origin to destination.

* Given the Network model for a shipper Representing its full set of transportation services for a particular Commodity, determine the minimum Cost (time) transport Option from Origin to destination.

[A] Single - pair Shortest path problem:-

[A] Single - pair Shortest path problem:-

↳ Find the Shortest path between a Single Source and destination

$[Node(s)]$ $[Node(t)]$

* مع لفة الدخان و في ارجل لفة من - لفة من - لفة من

الحل: $\frac{1}{2} \times 100 = 50\%$

B Single - Source Shortest path problem:-

↳ Find the Shortest path between a Single Source [Node (s)] and all other vertices/nodes.

3 Single - destination shortest path problem:-

↳ Find the shortest path from all vertices to a given destination. [Node (t)].

III All-pairs Shortest path problem:-

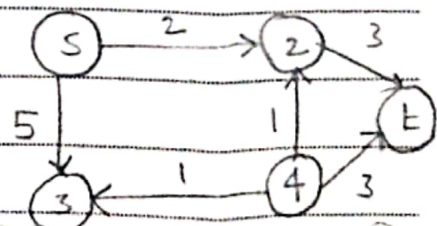
↳ Find the shortest path between all pairs of vertices.

* **Formulation definition:** Given a graph $G = (V, E)$ with costs c_{ij} for each $(i, j) \in E$, a source node s , and a destination node t , Find the path from $\underline{s}$ to $\underline{t}$ that minimizes the total costs for the path.

$$\sum C_{ij} \rightarrow \min \sum C_{EL}$$

path if main arc is d

⇒ paths from $S \rightarrow t$:-

 $\{5, 2, 6\} \quad \text{Coq} = 5$
$$\{5, 3, 4, 6\} \quad G_d = 9$$
$$\{5, 2, 4, 1\} \quad \text{Git} = 6$$


✓ feasible ممكن

feasible if we can find

* Solving Minimum Cost path problems (Models):



→ Dijkstra's Algorithm: Single-pair, Single-source, Single-destination

↳ Finding the Single Source (SP) Shortest Path.

- requires $C_{ij} > 0$ (+ve)

* pros: • Fast for dense Network

$m \rightarrow n^2$

• Can be truncated



If path of interest has been found

* Cons: • Alternative methods faster for Sparse Network

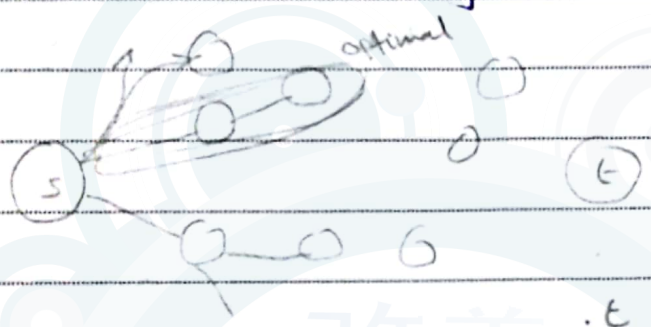
Additional Programming

• programming is needed for efficient implementation.

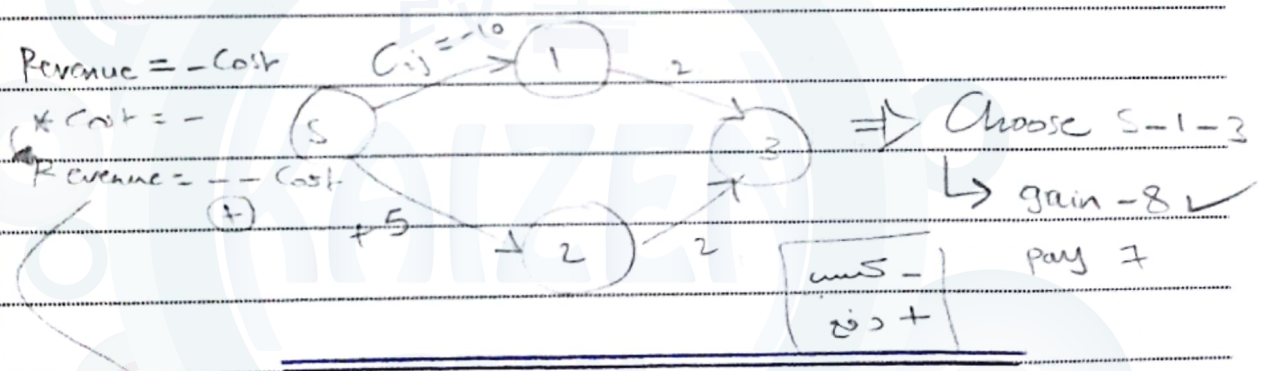
Pay Loss

→ **Bellman-Ford**: Single-Source, allows $C_{ij} \leq 0$
 ↳ paper's Modification → performance ↑.

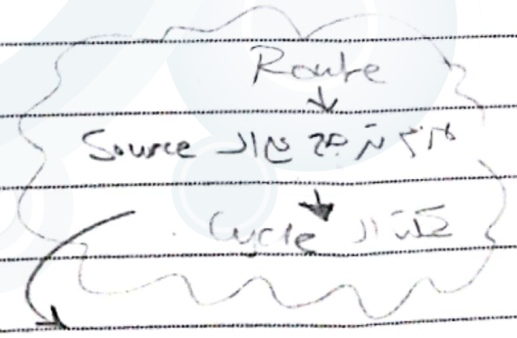
- * **Pros**:
- Fast for Sparse networks
 - Simple to implement.
- * **Cons**:
- Can not be truncated.
 - may be slow in dense Networks.



نوبت زي اول و كسور و لا انا *
 و لا انا و لا انا Optimal
 و لا انا
 و لا انا و لا انا *
 و لا انا و لا انا Optimal



max. $Z = X_1 + 2X_2$
 min. $-Z = -X_1 - 2X_2$

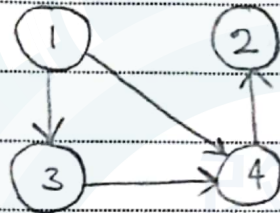


⇒ Dijkstra Algorithm:-

it is a labeling algorithm in which, at each iteration a different vertex p is given a permanent label $V(j)$ that indicates the cost of shortest path from s to p .

* Assumption: $c_{ij} \geq 0 \forall (i,j) \in E$.

↳ Can not be Negative.



$$E = \{(1,3), (1,4), (3,4), (4,2)\}$$

* Notation :-

- $V(j)$: Label for vertex j that denotes cost of path from s to j .
- $\text{pred}(j)$: predecessor of node j on shortest path from s to j .
- PERM: Set of permanently labeled nodes

* PERM not permanently labeled nodes

permanent set of nodes

Suppose (s) is input

Permanent

on Computer

*** Initialization:**

$$v(j) = \begin{cases} 0 & j = s \\ \infty & j \neq s \end{cases}$$

Costs Source

$$pred(j) = \begin{cases} 0 & j = s \\ -1 & j \neq s \end{cases}$$

bump to 1

$$PERM = \{ \}$$

let [empty]

$\overline{PERM} = V$ nodes of V

- Iterations:

• while $\overline{PERM} \neq \emptyset$ "Not empty".

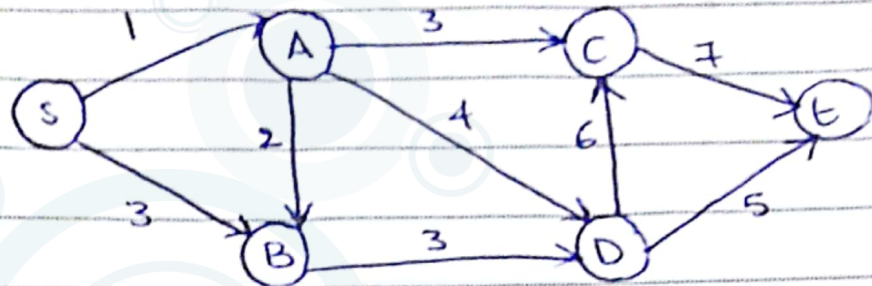
1. let $p \in \overline{PERM}$ be node for which
Cost of p is min $\leftarrow V(p) = \min \{V(j), j \in \overline{PERM}\}$

2. $\overline{PERM} \leftarrow \overline{PERM} + \{p\}$ $\rightarrow X = X + 1$

3. $\overline{PERM} \leftarrow \overline{PERM} \setminus \{p\}$ $\rightarrow$ exclude

4. For all $(p, j) \in E, j \in \overline{PERM}$ leading from (p) if $V(p) + C_{pj} < V(j)$ then
 $V(j) = V(p) + C_{pj}$
 $pred(j) = p$

Exampler



* $V(j)$: Cost of shortest path from $S \rightarrow j$.

* $\text{pred}(j)$: Node before j in path.

$\rightarrow V(j) [\text{pred}(j)]$

Cost

$V(j) + C_{ij}$

$C_{ij} + V(i) < V(j)$

ليست
أفضل

Iteration	S	A	B	C	D	E	P	PERM	PERM
Initialize	0[0]	$\infty[-1]$	$\infty[-1]$	$\infty[-1]$	$\infty[-1]$	$\infty[-1]$	—	{}	S, A, B, C, D, E
1		1[S]	3[S]				S	S	A, B, C, D, E
2				4[A] 5[A]			A	S, A	B, C, D, E
3							B	S, A, B	C, D, E
4						11[C]	C	S, A, B, C	D, E

$$V(A) + C_{AC} = 1 + 3 = 4 < V(C)$$

- Dijkstra's Algorithm - [Sip for Dijkstra]
 ↳ Labeled Algorithm → Cost 70.

* $V(k)$ → 0
 → ∞

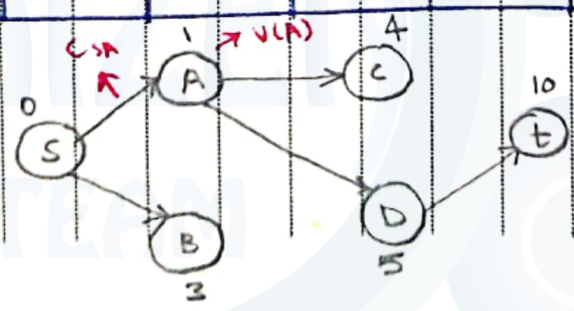
Iteration	S	A	B	C	D	E	P	PERM	PERM
Initial.	0[0]	∞[-1]	∞[-1]	∞[-1]	∞[-1]	∞[-1]	-	{3}	S, A, B, C, D, E
1	0[0] ↳ S is initial	1[5]	3[5]	* No direct edges from S.			S	S	A, B, C, D, E
2				4[A]	5[A]		A	S, A	B, C, D, E
3							B	S, A, B	C, D, E
4						11[E]	C	S, A, B, C	D, E
5						10[E]	D	S, A, B, C, D	E
6	0[0]	1[5]	3[5]	4[A]	5[A]	10[E]	E	S, A, B, C, D, E	{3}

Iteration 5 *
 the node is
 PERM - 1

Shows →
 all group with
 nodes & cost
 right side S will

Path 11 Cost
 11

Tree of
 Shortest
 path.



* Shortest path cost on D: $V(D) = 5$
 * Shortest path to reach D: S - A - D

* Iteration 1:-

* $P \rightarrow$ source node* $j \rightarrow$ node to be updated $\rightarrow S \rightarrow A$ $\rightarrow$ is $V(P) + C_{Pj} < V(j)$ is $V(S) + C_{SA} < V(A)$ $0 + 1 < \infty$ True $\Rightarrow \rightarrow V(A) = V(S) + C_{SA}$ * $C_{ij} \rightarrow$ Cost on Arc (i, j) .* $V(P) \rightarrow$ P will node of lowest cost of Source of it

* Iteration 2:-

 $\rightarrow A \rightarrow B$ $\rightarrow$ is $V(A) + C_{AB} < V(B)$ $1 + 2 < 3 \rightarrow$ No

[No Change] there is No

 $\rightarrow$ is $V(A) + C_{AC} < V(C)$ Benefit. $1 + 3 < \infty$ True $\Rightarrow \rightarrow V(C) = V(A) + C_{AC}$

* Iteration 5:-

 $\rightarrow V(D) + C_{DE} < V(E)$ $5 + 5 < 11$ $10 < 11$ True $\rightarrow V(E) = V(D) + C_{DE}$ $V(E) \leftarrow V(D) + C_{DE}$
assign $\rightarrow V(E) \leftarrow 10$

* Iteration 6:-

 $\Rightarrow$ No $P \in \text{PERM}$ to update $\Rightarrow$ final path is not updated $\rightarrow V(A) + C_{AD} < V(D)$ $1 + 4 < \infty$ True $\rightarrow V(D) = V(A) + C_{AD}$

* Iteration 3:-

Let $P \in \text{PERM}$ be node for which $V(P) = \min \{V(j) \mid j \in \text{PERM}\}$ all $j \in \text{PERM}$ is not updated will node of $\rightarrow$ all costs is in cost of it by its min* $V(B), V(C), V(D), V(E) \rightarrow$ then the min. $\rightarrow V(B) + C_{BD} < V(D)$ $3 + 3 < 5$ False $\rightarrow$ No Change to do in this iteration

* Iteration 4:-

 $\rightarrow V(C) + C_{CE} < V(E)$ $4 + 7 < \infty$ True $\rightarrow V(E) = V(C) + C_{CE}$

↳ This is sometimes referred to as a label
[Lab-Correcting] Algorithm.

- * There is no requirement for non-Negative edge costs. $< 0 \rightarrow$ Could be less than zero

$$v_j = \begin{cases} 0 & j = s \\ \infty & j \neq s \end{cases}$$

- $\text{pred}(j) = \begin{cases} 0 & j=s \\ -1 & j \neq s \end{cases}$

LIST = {5}

Iterations:-

← حكمة بالقيمة
• Coding

Case 1 • While $LIST \neq \emptyset$ [Not Empty] :-

II Remove node p from top of LIST

It will not be feasible
line arc

② For every (p, j) leading from node p

a. If $V(p) + C_{pi} < V(j)$, then

$$2. \quad v(j) = v(p) + c_{pj}$$
$$ii. \quad p_{\text{pred}}(j) = p$$

← true لكن القطر البسيط $\frac{1}{2}$

السؤال الثاني: ما هي أهمية التقييم الذاتي في تحسين الأداء؟

iii. If $j \notin \text{LIST}$, $\text{LIST} \leftarrow \text{LIST} + \overset{\uparrow}{e_j}$

* Pipe's Modification:

IF $j \notin \text{LIST}$, then

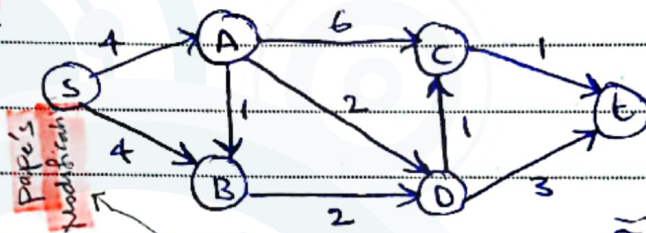
Sub LIST is as B ←
 line below is given

IF j has never been in LIST, put
 j at end

Else put j at beginning of LIST

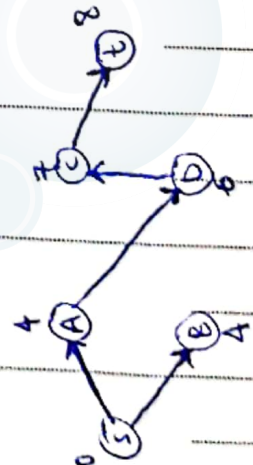
Up to this all j are in LIST and now
 LIST is as given, take, with some cost (like)

Example 1:



It	S	A	B	C	D	E	P	LIST
1	0	∞	∞	∞	∞	∞	-	S
2		4	∞	∞	∞	∞	S	A, B
3				10	6	∞	A	B, C, D
4						11	B	C, D
5				7		9	C	D, E
6						8	D	E
7						7	E	

⑤ If $S \rightarrow C$ cost 10
 If $S \rightarrow D$ cost 6
 Shortest path $S \rightarrow D$
 Cost = 6



$$\begin{aligned} \text{Cost} &= 1 + 4 + 4 + 4 \\ &= 13 \end{aligned}$$

No.

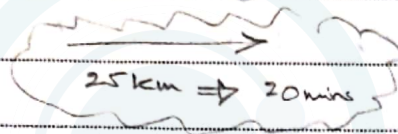
(49)

→ for shortest path.

Multi-Label Algorithm: → 2 Attributes at the
↳ GPS Same time.



map is not a straight line → Route find.
"Sole chas"



(49) - Multiple objectives: distance & time

- Vertices given labels with two Components
(d_k, t_k)

• d_k → First Cost Component

↳ total distance required to reach
this vertex along some path
from (S).

• t_k → Second Cost Component.

↳ total time.

- Vertices may have Multiple labeled
resulting from different paths.

• One path to a vertex may have
larger distance but shorter time
than the another path. (Vice versa).

• Now, Instead of a vertex having a
predecessors label, each Cost
Component Label (d_k, t_k) has
associated predecessor label $\boxed{\text{pred}(k)}$

- We only maintain non-dominated labels at each vertex.

• A label (d_q, t_q) is dominated by (d_k, t_k)

if both $d_q > d_k$ & $t_q > t_k$

↳ Better

- Maintain a LIST of Labels instead of vertices:-

• Remove Label from LIST and examine edges leaving from the vertex associated with the Label.

* Initialization:-

• Origin vertex (S) given the label $(0, 0)_1^S$,
Where Subscript indicates label # ($k=1$)
and Superscript indicated the vertex
where the label exists (S).

Successors $\leftarrow S$ // label // $\bar{r} \leftarrow 1$

① $\leftarrow (d_1, t_1)_1^S$

② $\leftarrow (d_2, t_2)_2^S$

The predecessor for this label is 0, so
 $\text{pred}(k) = \text{pred}(1) = 0$

• No labels are given to other vertices

• $\text{LIST} = \{(0, 0)_1^S\}$

• $r=2$ (keep track of # of next label generated).

- Iterations:

• While $LIST \neq \emptyset$:

1. Remove Label k from beginning of $LIST$.

Suppose Label k has Components such that it exists at Vertex p .

2. For every edge (p, j) leading from Vertex p :

a. Create a new Label r at vertex

j , where $(d_r, t_r) = (d_k + d_{p,j}, t_k + t_{p,j})$

b. define $pred(r) = k$

c. Add Label r to end of $LIST$.

d. Compare Label r to each existing Label (d_q, t_q) currently at Vertex j

i. If Label r is dominated by Label q , discard the new Label r and remove it from $LIST$

If $d_r \geq d_q$ and $t_r \geq t_q$.

ii. Else if Label q is dominated by r , discard q , and if necessary remove it from $LIST$

If $d_r \leq d_q$ and $t_r \leq t_q$

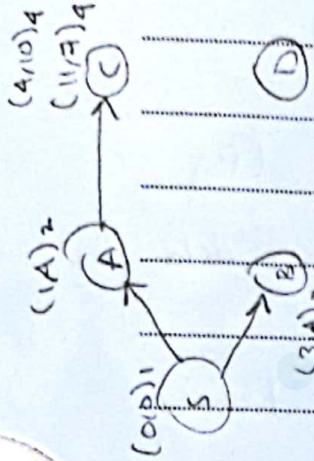
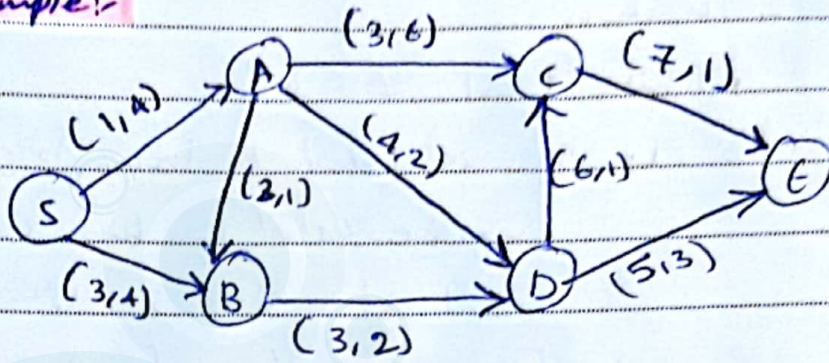
iii. Else retain both labels.

e. $r = r + 1$

• Stopping Criterion: When no Labels remain in the $LIST$, no remaining Label-Correction can be found.

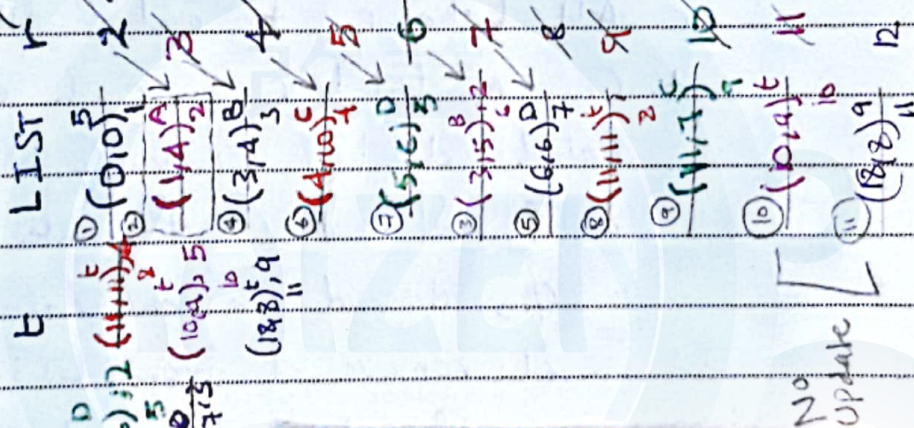
→ All non-dominated paths to all vertices have been identified.

Example:



visiting
is
label
(2,1)

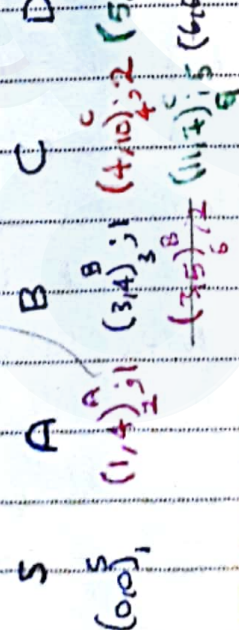
circles and arrows



No Update

Suppose visiting
ji Supos is

Label no sup
(1,1) 2



First Cost Component: distance
- Shortest distance path to all nodes from (S)



Second Cost Component: time
- Shortest path to all nodes from (S)



* to reach (E) = 8 min, 18 Units of length

- Minimum Spanning Tree Model:-

- Definition of a Tree:-

↳ Collection of $(n-1)$ edge such that all vertices in G are connected, No cycle exists.

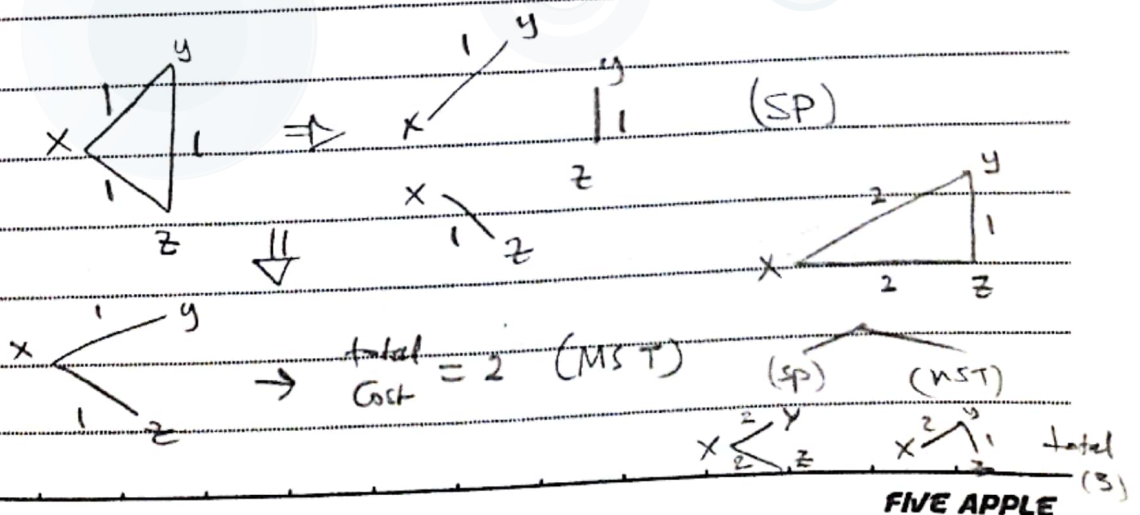
- A minimum Spanning tree (T^*) in $G = (V, E)$ is a tree such that total cost of all arcs in the tree is minimized.

$$T^* = \arg \min_{T \in G} \left\{ \sum_{(i,j) \in T} c_{ij} \right\}$$

T is a tree in Network G if it contains all vertices and $(n-1)$ arcs such that no cycle exists. (Sum for Min)

Q: Is MST the same as SP tree?

↳ No, Shortest path tree minimizes length of shortest path(s) from some origin node to all other nodes, while the MST minimizes the total cost of all arcs in the tree.



* Algorithm:- Steps, procedure to Solve a problem.

↳ Guarantee to Converge

[$J \rightarrow$ Optimum J deni]

* Heuristics:- No proof of Optimum Solution.

Algorithm J in J deni

* Kruskal's Algorithm:-

- Initialization:-

• $TREE = \{ \}$

- Iterations:-

• Sort arcs in Order of Increasing C_{ij} into LIST.

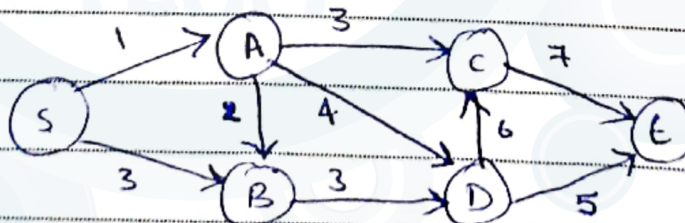
Break the loop while $|TREE| < n-1$

• Remove (i,j) from top of LIST.

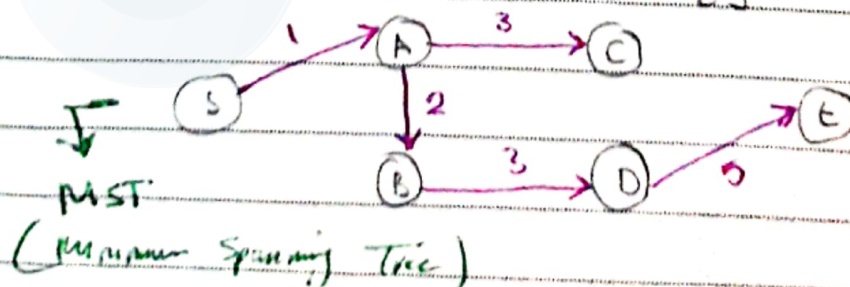
• Add (i,j) to TREE, if it does not create a cycle.

* At Completion of Algorithm $\rightarrow T^* = TREE$.

Example:- Find MST and its Cost.



I Initialization $\rightarrow TREE = \{ \}$



[2]

$$S-A \rightarrow C_{SA} = 1$$

$$A-B \rightarrow C_{AB} = 2$$

$$S-B \rightarrow C_{SB} = 3$$

$$A-C \rightarrow C_{AC} = 3$$

$$B-D \rightarrow C_{BD} = 3$$

$$A-D \rightarrow C_{AD} = 4$$

$$D-E \rightarrow C_{DE} = 5$$

$$D-C \rightarrow C_{DC} = 6$$

$$C-E \rightarrow C_{CE} = 7$$

While TREE =

5 arcs

 $< (n-1)$

Continue

amgle (to jump) *

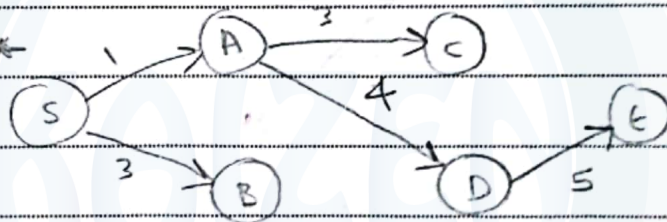
Loop jank 'ol

Killing 5 arcs $\leftarrow$

Nodes & its Connect

$$\Rightarrow \text{MST} \rightarrow C(\text{MST}) = \sum C_{ij} = 14.$$

is to jump *



→ * Shortest path tree rooted at (S).

$$\text{Total Cost} = 1 + 3 + 3 + 4 + 5 = 16$$

MST is not this

* Prim's Algorithm:-

We must first understand what a Cut is in a network.

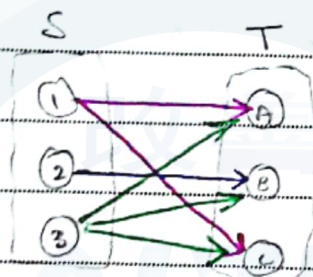


Suppose Vertices, in G partitioned into 2 Independent Subsets (S) and (T)

The Cut set (S, T) is Given by all arcs (i, j) Such that:

$i \in S$ and $j \in T$, or

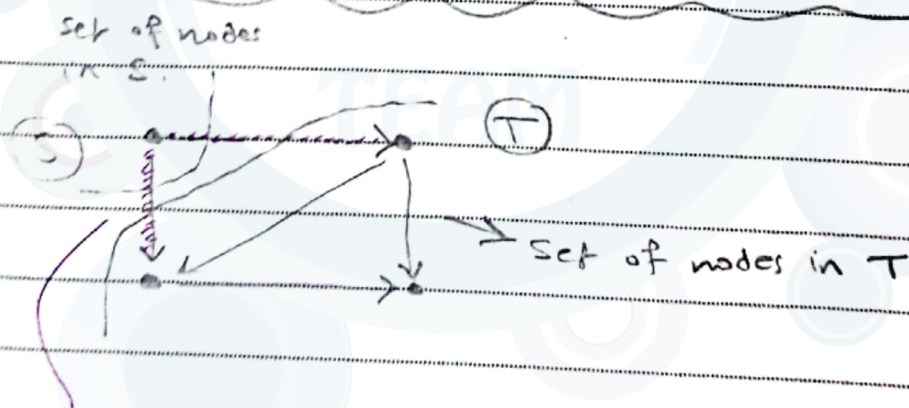
$j \in S$ and $i \in T$



$S = \{1, 2, 3\}$

$T = \{A, B, C\}$

$\Sigma (1, A), (1, C), (2, B), (3, A), (3, B), (3, C)$



arc $\rightarrow$ Set (1) $\Rightarrow$ Set (2) $\rightarrow$ Set (1)
Cut set will be

Continue Logistics

No.

57

- Initialization -

- $TREE = \{\}$
- $S = \{1\}, T = V \setminus S$

- Iterations -

→ No. of edges in tree.

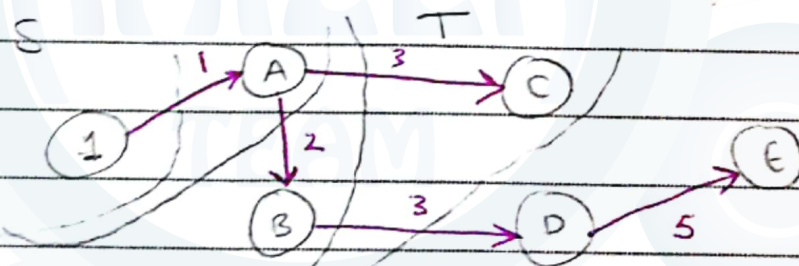
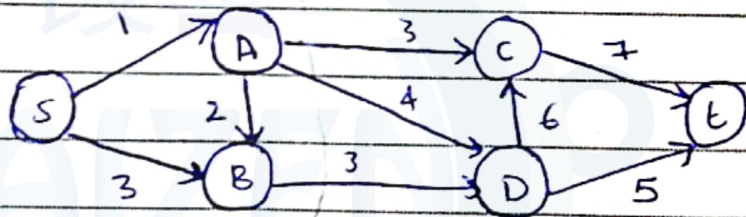
- While $|TREE| < n-1$

◦ Find arc $(i,j) \in (S,T)$ with min C_{ij} (Cost).

• Add (i,j) to TREE.

• Remove j from T and add to S .

Example:



Cost = 14

① $C_{SA} = 1$ ✓

$C_{SB} = 3$

$C_{AC} = 3$

② $C_{AB} = 2$ ✓

$C_{AD} = 4$

$C_{SB} = ?$

$S = \{1, A\}$

$T = \{B, C, D, E\}$

* Minimum Spanning Tree Applications:

→ * Standard minimum Cost paths (Shortest paths) are called **mini - Sum** paths, because among all paths P connecting S to t the minimum paths P_s^* has minimum total Cost.

$$P_s^* = \arg \min_{PEG} \left[\sum_{(i,j) \in P} C_{ij} \right] \rightarrow \text{Sum for all arcs of Costs for these paths}$$

↓
arguing which path is minimum
bt. all paths that belong to this graph G

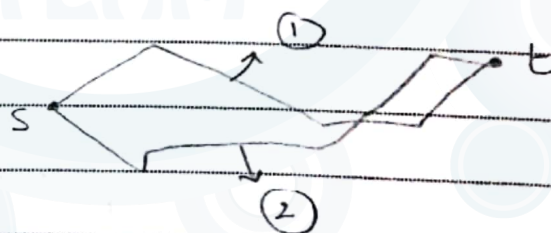
→ Satisfying $\sum$ for all arcs including in this path.

path that has the min. max.

* Minimax paths: the minimax path P_m^* encounters the smallest max. arc Cost.

$$P_m^* = \arg \min_{PEG} \max_{(i,j) \in P} C_{ij} \rightarrow \text{max. cost at path (1)}$$

5 ✓



→ max. arc at path (2)

↓
then choose min.



S-1-E
↑
PS

5

$$\frac{8-2-6}{\uparrow P_{\text{max}}}$$

7

4

Women: 1988-90

$\Rightarrow$ max. duration for Connection min

Abbildung von Jordan - Germany - Canada - USA → 16 hrs
4 8 4

2/20/2020

It is also used ← Jordan - Qatar - USA → 12 hrs
 Connection 2 10

total trip 51 miles
52 مایل و ۱ مایل

$\hookrightarrow$ min sum $\rightarrow 12 \rightarrow$ Second path $\rightarrow$ total path
 $\hookrightarrow$ min max $\rightarrow 8 \rightarrow$ First path $\rightarrow$ Shortest path problem

↓ Spanning tree

arcs $\gamma_1, \gamma_2, \dots$

Useful application of minimax path:-

- Elevation change:

Very steep road segments $\uparrow$, fuel $\uparrow$ and maintenance required $\uparrow$ (Costs) / So you want to avoid ^{no like connection} segments with maximum elevation change



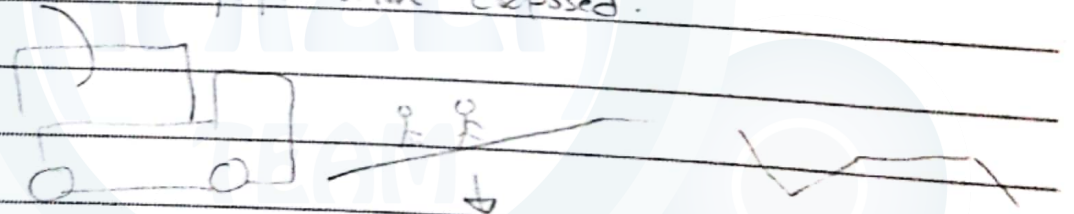
Max. Δ is no Segment Δ is no
MST $\Leftarrow$ is no Δ is no

- Hazardous Materials:-

If spilled, population harmed, avoid densely populated road ways

$\rightarrow$ Minimize maximum expected population exposed.

Spillage



Spillage $\rightarrow$ Density $\uparrow$

$\Rightarrow$ Minimize Max. # of people exposed for Material

Spillage $\rightarrow$ Risk $\rightarrow$ Minimize Risk

* Logistics Review:-

- Important Concepts:-

II] the minimum path problem is solved by solving shortest path problem.

III] the minimax problem is solved by solving a minimum spanning tree problem.

→ Solving minimum spanning tree problem solves the all-pairs minimax pathing problem.

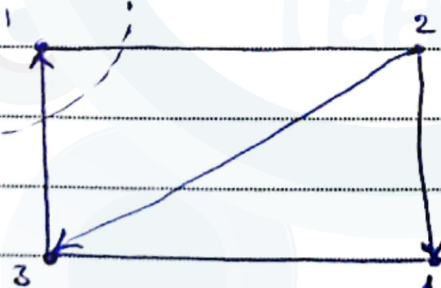
→ Solving shortest path problem solves the all-pairs minimum pathing problem.

all pairs min pathing problem ←

∴ (15) Simple *

* Review:-

→ Prims Algo:-



→ Cut set (S, T)

S = {1, 3}

T = {2, 4}

edges (i, j)

$i \in S + j \in T$ or
 $j \in S + i \in T$

$i \rightarrow j$

$j \rightarrow i$

* (q) is dominated by (r)

$$\hookrightarrow |r| < |q|$$

(q) is dominated by (r)
 (r) is dominating node.

$$(d_q, e_q) \leftarrow (d_r, e_r)$$

in which $e_r < d_r$ is $|d_r| < |d_q|$

dominated r since $e_q > d_q$

now d is $|d_r| < |d_q|$ -

Non dominated by

truncated $\rightarrow$ min Computation

* Vehicle Routing Problem:- [VRP]

In many transportation problems, a vehicle departs from a node (s) $\rightarrow$ like a ^{depot} depot, visiting a number of locations (e.g. 1, 2, ..., p) and returning to Node (s).

The objective is to choose the sequence of nodes $s, \dots, 1, 2, \dots, p$ which minimizes the cost of the circuit.



$\Rightarrow$ When routing only a single vehicle, we refer to this as the **Traveling Salesman problem [TSP]**.

The more general case, when any # of vehicles may be routed, is referred to as the vehicle routing problem [VRP] $\rightarrow$ general case

* The first step in solving routing problem is to develop a Complete Undirected Network $G(V, E)$ Satisfying the Triangle-inequality.

→ when V is limited to the nodes that will be visited [the start/end node(s), and all nodes $1, 2, \dots, p$].

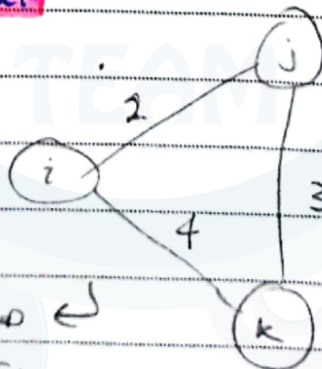
→ Triangle-inequality on a Complete Network:-

↳ A Complete Network $G = (V, E)$ Satisfies the Δ -ineq. iff $\forall i, j, k \in V$, the IP and only IP.

following relationship:

$$C_{ik} \leq C_{ij} + C_{jk}$$

Example:-



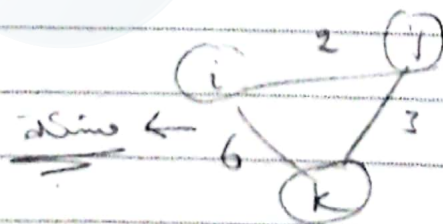
arc is the shortest path

$$2 < 3 + 4 \checkmark$$

$$4 < 2 + 3 \checkmark$$

$$3 < 2 + 4 \checkmark$$

Yes, it is Satisfying the Triangle-ineq.



$$2 < 6 + 3 \checkmark$$

$$6 < 2 + 3 \times$$

NO

* Generating a Complete Undirected Network Satisfying the Δ -Ineq.

- often not true.
- procedure:-

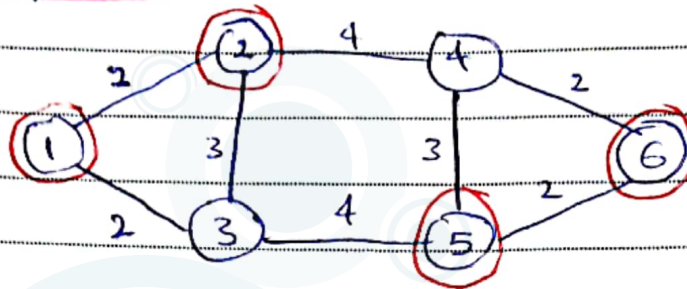
* Let $G' = (V', E')$ be a network that is ^{edges from nodes of G'} Not Complete. Let V be nodes of interest in routing problem, where $V \subseteq V'$. $\rightarrow$ Complete Network

* Create $G = (V, E)$ where:

- E includes edge from each node in V to every other node in V
- Cost on each $(i, j) \in E$ is Cost of minimum path from i to j in G' .
- Assumptions:-

$\hookrightarrow$ a path exists b/w all nodes in V' in G' , and Cost from the i to j does not differ from the Cost from j to i .

* Example:-



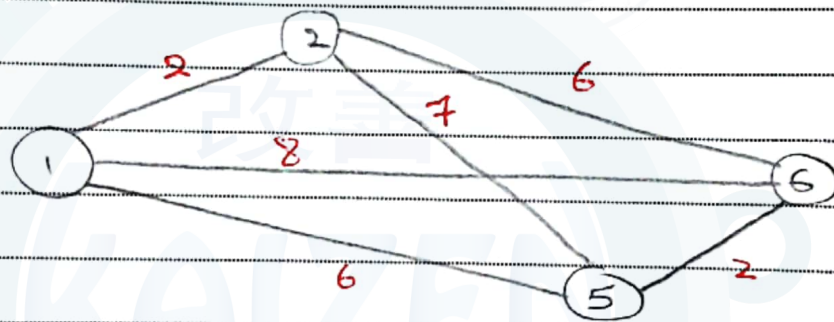
SPP

$$G' = (V', E')$$

Shortest path

$$V' = \{1, 2, 3, 4, 5, 6\}$$

$$V = \{1, 2, 5, 6\} \rightarrow \text{main path}$$



$\Rightarrow$ TSP tour on Complete undirected (Symmetric) Network $G = (V, E)$:

* Let n be the Number of Nodes in V .

* A TSP tour T on G is a

Sequence of all Nodes in V ,

$$T = \{z_1, z_2, z_3, \dots, z_n / z_1\}$$

Node.

Shortest path

Cycle

TSP.

where $z_k \in V$ & each z_k

Unique $\forall k = 1, \dots, n$

$$T = \{1, 2, 6, 5, 1\}$$

To avoid Subtour

لا يمكن ان يكون هناك

Subtour: a Sequence of nodes with Common start & end node, such that not all nodes in V are visited,

$\{z_1, z_2, z_3, \dots, z_q, z_1\}$ where:

$z_k \in V$ and each z_k Unique &

$q < n$ $\Rightarrow$ nodes start & end at same

* **Tour Cost:** Sum of the Costs of all edges traversed in the tour.

$$C(T) = \sum_{i=1}^{n-1} C_{v_i, v_{i+1}} + C_{v_n, v_1}$$

$\rightarrow$ Optimal TSP tour: Minimum
 $\hookrightarrow TSP = \arg \min_{T \in G} C(T)$

• One algorithm for TSP:-

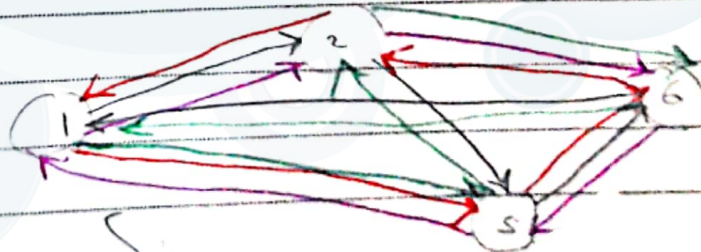
- Enumerate every tour in the Network
- Choose the one with lowest Cost

Inefficient
Algorithm



$n!$

Practical



* Based on Possibility

Heuristic Algorithms - [Heuristics]

→ * provide a feasible solution that may or may not be optimal.

* Good Heuristics:-

[1] provide solution close to Optimal solution.

[2] Efficient in Runtime

* Desirable property of Heuristic:

- Ability to Compare quality of solution produce by the Heuristic to the optimal solution.

Solutions can be guaranteed:-

- Worst Case quality of solutions:

(1) These "guarantees" (Maximum deviation from Optimality)

(2) If the Cost of Optimal solution is C^* , then heuristic finds a solution with cost no greater than αC^* .

→ Constant.

C^* is min cost

$$1.5 C^*$$

$$1.2 C^* \quad 1.5 C^*$$

$$1 C^* \quad \nabla$$

***Heuristics for TSP:-**

- 1] Christofides heuristic
- 2] Constructive heuristic: Nearest neighbor, Nearest insertion, farthest insertion, cheapest insertion.
- 3] Improvement heuristic: 2-opt
k-opt

⇒ Christofides heuristic:-

- Assumptions:- Complete Network Satisfying Δ -ineq.

→ Definitions:-

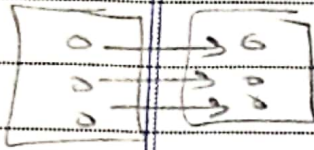
*Node degree: the degree of a node $i \in V$ with respect to an edge set E is actually the number of edges in E having i as an end point.



* perfect Node matching: Given a set of Node $\underline{S}$ such that the number of Nodes in $\underline{S}$ is even, a perfect Node matching $\underline{W}$ is a set of $|\underline{S}|/2$ node pairings such that each node in $\underline{S}$ is in exactly one pairing. In a Complete Network, each Node pairing (i, j) where $i, j \in \underline{S}$ represents an arc in the Network.

→ the Cost of Matching W is Given by

$$C(W) = \sum_{(i,j) \in W} C_{ij}$$



belongs for
all matches
that we did

* Christofides' Heuristic :-

• Iterations:-

- Given an Undirected Complete Network

$G = (V, E)$ Satisfying the Δ -ineq.,

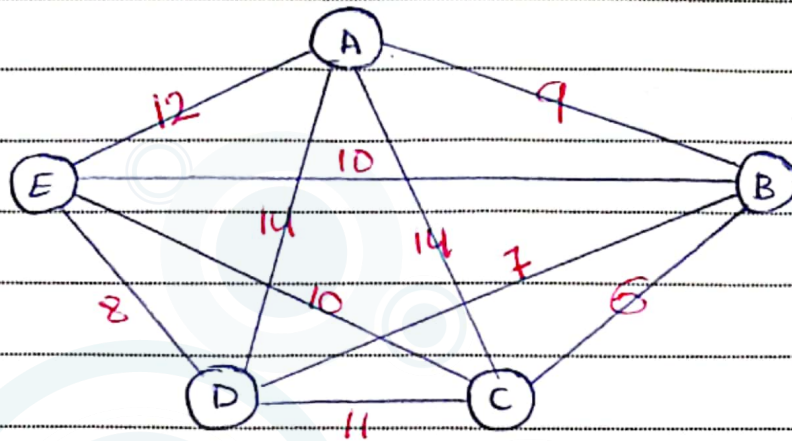
- Find a Minimum Spanning Tree MST^* on G .

- Let $\underline{S}$ be the nodes of G with odd degree with respect to MST^* .

- Find the minimum Cost of perfect Node Matching W^* of nodes in $\underline{S}$.

- Generate a tour T^c of the Nodes using each of the arcs in the set of $MST^* \cup W^* \rightarrow$ exactly once (improve T^c if necessary by introducing shortcuts to avoid revisiting nodes).

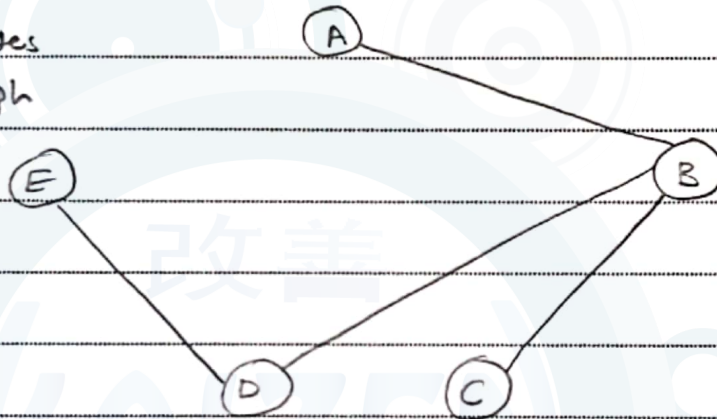
T^c if necessary by introducing shortcuts to avoid revisiting nodes.



* MST:-

↓
touch All Nodes
in the graph

↓
(stop)
n-1 ✓



* odd degree with respect to MST:-

$S = \{A, B, C, E\} \rightarrow |S| = \text{even} \checkmark$ $|S| = 4$

odd degree nodes $\leftarrow$ 2 nodes $|S|$ [3]

$AE = 12$	$AC = 14$	$AB = 9$
$BC = 6$	$BE = 10$	$CE = 10$
$\underline{18}$	$\underline{24}$	$\underline{19}$

↑
 w^*

$\min \sum_{(u,v) \in E} w_{uv} \rightarrow 18 \checkmark$

↑
 w

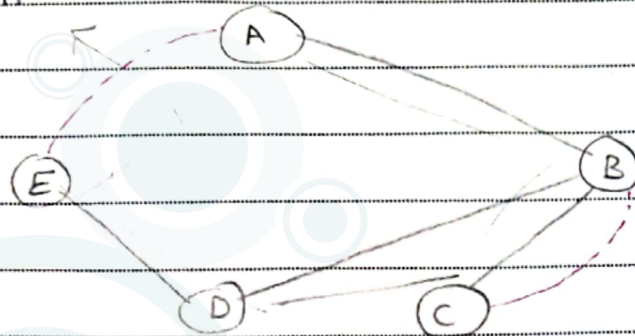
A-B-C-B-D-E-A

W^{*} U MST^{*}

A

A-B-C-D-E-A

(46)

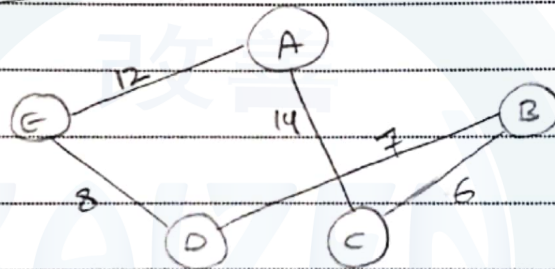


A-E-D-B-C-A

5

T = A-E-D-B-C-A

Shortcuts



C(T) = 47

Perfect Node matching Options:-

$\rightarrow n \text{ nodes} \rightarrow 4 \text{ nodes } 2 \text{ steps}$
 we need to select 2 nodes from 4,
 then 2 from 2.

Number of ways to divide 4 nodes into 2 groups

2 pairs

$$C_2^4 \times C_2^2 = \frac{4!}{2!} \times \frac{2!}{2!(2-2)!}$$

2 pairs

2 pairs

$$= \frac{4 \times 3 \times 2}{2 \times 1} \times \frac{1}{1} = \frac{4 \times 3}{2}$$

2

AE
 BC

↓

2 × 1 = 2

= 6V

= 3V

No.

(73)

6 nodes into groups of 2

A, B, C, D, E, F

$$C_2^6 * C_2^4 * C_2^2$$

$$\frac{6!}{2!(6-2)!} * \frac{4!}{2!(4-2)!} * \frac{2!}{2!(2-2)!}$$

$$\frac{6 \times 5 \times 4!}{2! (4)!} * \frac{4 \times 3}{2!} = (5 * 6)$$

$$\checkmark \frac{15 * 6}{3!} = (15) \leftarrow \text{pairs}$$

(A, B) (C, D) (E, F) $\rightarrow$ 3 pairs

Let T^c be tour produced by Christofides.

Theorem:- Worst Case performance bound for Christofides:

Chr. heuristic is 1.5 Optimal for every Complete Network G Satisfying Δ -ineq.

$$\Rightarrow C(T^c) \leq \frac{3}{2} C(TSP^*) \xrightarrow{10} (15) \xrightarrow{\text{min}} \text{Actual optimum of TSP.}$$

$$(14) \rightarrow \frac{14}{1.5} \Rightarrow \text{Best possible tour of TSP}$$

Construction heuristics of TSP:-

A. Nearest neighbor heuristic:

⇒ Assumption: Complete Undirected Network Satisfying Δ -ineq.

⇒ Initialization: 2 Option.

• Let $p = \{v_{ij}\}$ where $C_{ij} = \min_{(i,j) \in E} C_{ij}$

(or)

• Let i be arbitrary start node, find $j \in V \setminus \{i\}$ such that C_{ij} is minimized. Let $p = \{v_{ij}\}$.

⇒ Iterations:

while $|p| < n$:

• let s be the beginning node of the path & t be the end

• Choose the Node $k \in V \setminus p$ to enter the path:

* Let k be such that C_{tk} or C_{ks} is minimized



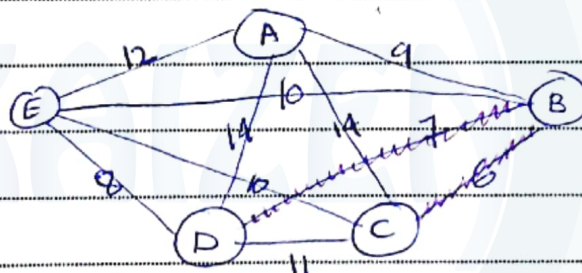
The New path is either $p + \{k\}$ or $\{k\} + p$, respectively.

- let $\text{begin}(p)$ be the beginning node of the path.

- let $T = p + \text{begin}(p)$

- if C_{tk} is minimized, $p = p + \{k\}$

- if C_{ks} is minimized, $p = \{k\} + p$



* Initialization $\rightarrow \min C_{ij} \rightarrow (B-C)$

* Iteration $\rightarrow$ Iter 1: $B \rightarrow$ Closest to D (Min 7)

$C \rightarrow$ Closest to (10) E (10)

$p = \{D-B-C\}$

Iter 2: $D \rightarrow$ Closest to E (8)

$C \rightarrow$ Closest to E (10)

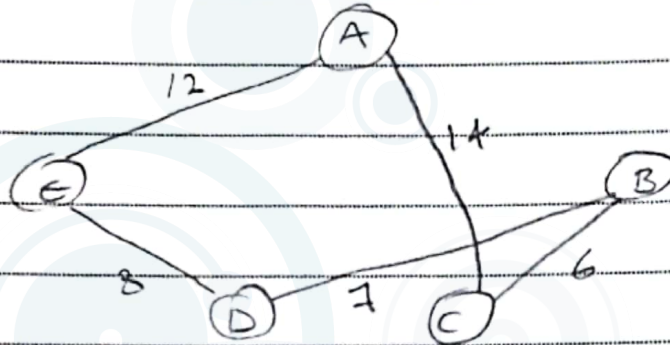
$p = \{E-D-B-C\}$

Iter 3: $E \rightarrow A$ 12

$C \rightarrow A$ 14

$p = \{A-E-D-B-C\}$

Tour $\Rightarrow T = \{A, E - D - B - C - A\}$
 Cost = 47



B. Nearest Insertion Heuristic:

$\rightarrow$ Insertion Cost: Inserting node j bt. nodes L & K where L & K are a pair of successive nodes in a tour, has Cost:

$$[C_{Lj} + C_{Kj} - C_{LK}]$$



Can never be Negative if Network Satisfying Δ -ineq.

$\Rightarrow$ Assumption: Undirected $G = (V, E)$ satisfy Δ -ineq.

$\Rightarrow$ Initialization:

- Find arc (i, j) with minimum Cost in E .

Let $T = \{i, j\}$



⇒ Iterations:-

While $|T| < n+1$:-

- Find the node not in the tour closest to a node in the tour

⇒ Let $C_j(T)$ be $\min C_{ij}$ for all Nodes $j \notin T$

$i \in T$.

Choose node j to Insert whose $C_j(T)$ is min.

- Find Insertion Location in tour with minimum insertion.

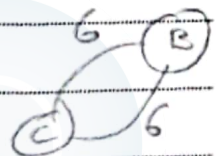
⇒ Find arc (L, k) in tour where $C_{Lj} + C_{jk} - C_{Lk}$ is Minimized.

- Insert j b/w L & k in T .

Example:- Same Network

Initialization → $T = \{B-C-B\}$

$C(T) = 12$



Iter 1 →

$\begin{matrix} i \in T \\ j \notin T \end{matrix}$	B	C
A	9	14
E	7	11
D	10	10

$\{B-C-B\}$

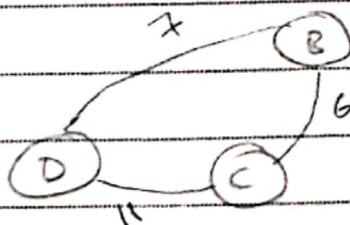
↑ ↑
or 21

Cost of Insertion:

$$= C_{BD} + C_{DC} - C_{BC}$$

$$= 7 + 11 - 6 = 12$$

$$C(T) = 12 + 12 = 24, T = \{B-D-C-B\}$$



No.

(78)

Iter2 :-

	24 18 T	B	C	D
A		9	14	14
E		10	10	8

$$T = \{B-D-C-B\} \rightarrow C(T) = 24$$

↑ ↑ ↑

$$(B, D) \Rightarrow C_{BE} + C_{ED} - C_{BD} = 10 + 8 - 7 = 11$$

$$\checkmark (D, C) \Rightarrow C_{DE} + C_{EC} - C_{DC} = 8 + 10 - 11 = 7$$

$$(C, B) \Rightarrow C_{CE} + C_{EB} - C_{CB} = 10 + 10 - 6 = 14$$

$$T = \{B-D-E-C-B\}$$

$$C(T) = 24 + 7 = 31$$

Her. 3 :-

Insert A: Since it is the last node.

$$T = \{B-D-E-C-B\}$$

Location $\rightarrow (B, D) \rightarrow C_{BA} + C_{DB} - C_{ED} = 9 + 14 - 7 = 16$

$$(D, E) \rightarrow C_{DA} + C_{AE} - C_{CB} = 18$$

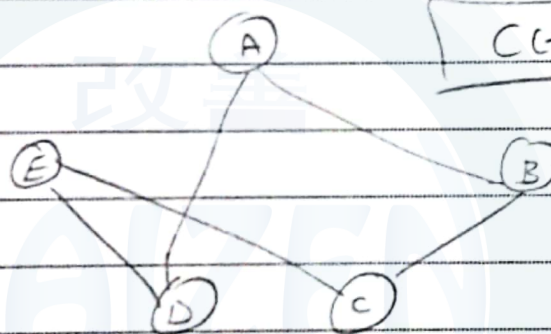
s_p, s_i, t_s

value

$$(E, C) \rightarrow 16$$

$$(C, B) \rightarrow 12$$

$$T = \{B-A-D-E-C-B\}$$



$$C(T) = 47$$

C. Farthest Insertion Heuristic :-

$\Rightarrow$ Assumption :- Undirected Network $G = (V, E)$

Satisfying Δ -ineq.

$\Rightarrow$ Initialization:-

- Find arc (i, j) with maximum cost in E . let $T = \{i, j, i\}$.

$\Rightarrow$ Iterations:-

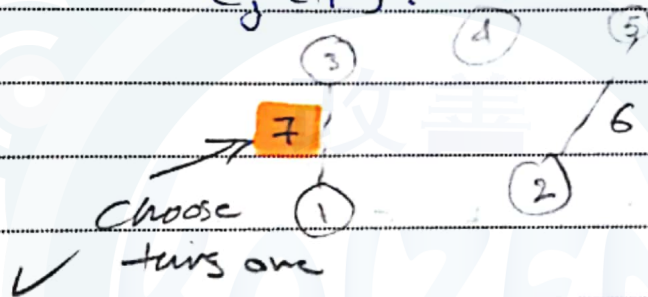
- Identical to nearest Insertion, except when choosing the node enter the tour in each iteration, use the following Criteria:-

- for every node j not in the tour, find minimum distance edge (i, j) connecting node j to a node in the tour that is:

$$C_j(T) = \min_{i \in T} C_{ij} \quad \text{for } j \notin T$$

* pick node j for which $C_j(T)$ is minimized.

- The node to insert is the one with max $C_j(T)$.



⇒ Initialization:-

$$\text{Init} \rightarrow C(T) = 28$$

$$\{A-C-A\}$$

⇒ Iterations:-

①	$j \notin T$	$i \in T$	
		A	C
	B	9	6
	D	14	11 ✓
	E	12	10

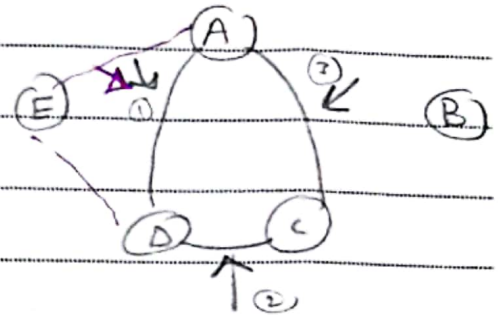
* Min on each Row then take Max.

$$T = \{A-D-C-A\}, \quad C_{AD} + C_{DC} - C_{AC} = 11$$

$$C(T) = 11 + 28 = 39$$

(2)

JET \ JET	A	C	D
B	9	<u>6</u>	7
E	12	10	<u>8</u>



$$C_{AD} + C_{ED} - C_{AE} = 12 + 8 - 14 = 6 \quad \checkmark \quad \text{Insert E bt. (A,D)}$$

$$C_{DB} + C_{EC} - C_{DC} = 8 + 10 - 11 = 7$$

$$C_{CE} + C_{EA} - C_{CA} = 10 + 12 - 14 = 8$$

$$T = \{A-E-D-C-A\}$$

$$CCT = 39 + 6 = 45$$

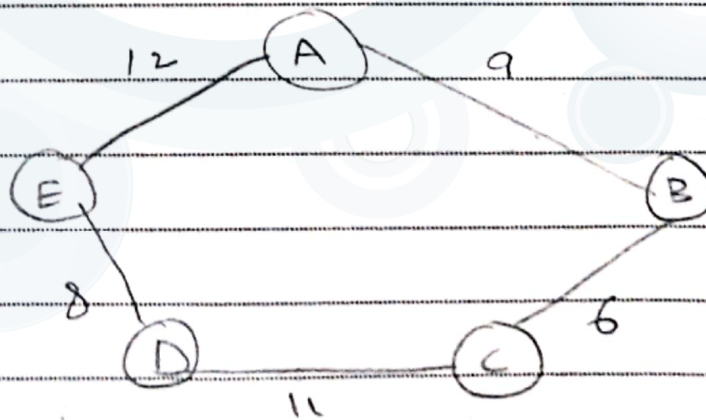
(3) Final Iteration:-

$$T = \{A-E-D-C-A\}$$

↑ ↑ ↑ ↑

Insert B bt. A & C → Insertion Cost = 1

$$\hookrightarrow CCT = 45 + 1 = 46$$



D. Cheapest Insertion Heuristic:-

⇒ Assumption :- Undirected Network

$G(V, E)$ Satisfying Δ -ineq

⇒ Initialization:

- Find arc (i, j) with minimum Cost in E , $T = \{i, j\}$

⇒ Iterations:-

While $|T| < n+1$

- For every node j not in the tour, Find the Insertion Cost for all possible Insertion locations in the tour

⇒ i.e. → Find (i, j) in the Current Subtour and k not in the Subtour, such that

$C_{ik} + C_{kj} - C_{ij}$ is minimized

→ Insert k bt. i & j

Insert location $k \notin T$	(B C)	(C D)	(D B)
A	$C_{BA} + C_{AC} - C_{BC}$	$C_{CA} + C_{AD} - C_{CD}$	—
E	$C_{BE} + C_{EC} - C_{BC}$	—	—

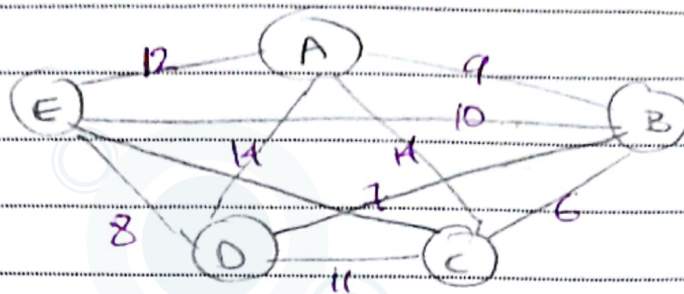
Choose Min.

(i, j)

add E in this location

assume the Current tour → B-C-E-D

↳ B-C-E-D-B



$$T = \{B-C-B\} \rightarrow \text{Initialization}$$

* Iterations:-

Insertion Location k & T	(B/C)
A	$9 + 14 - 6 = 17 = C_{BA} + C_{AC} - C_{BC}$
D	$7 + 11 - 6 = 12 \checkmark$
E	$10 + 10 - 6 = 14$

$$\Rightarrow T = \{B-D-C-B\}$$

$$C(T) = 12 + 12 = 24$$

$$= 7 + 11 + 6 \rightarrow$$

	(B/D)	(D/C)	(C/B)
A	$9 + 14 - 7$	$14 + 14 - 11$	$14 + 9 - 6$
$\rightarrow$	16	17	17
E	$10 + 8 - 7$	$8 + 10 - 11$	$10 + 10 - 6$
$\rightarrow$	11	7	14

$$T = \{B-D-E-C-B\}$$

$$C(T) = 24 + 7 = 31 \checkmark$$

then Add A in the last iteration.

No.

(84)

$$\textcircled{15} T = \{B-D-E-C-B\}$$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

$$C_{EA} + C_{AS} - C_{EC}$$

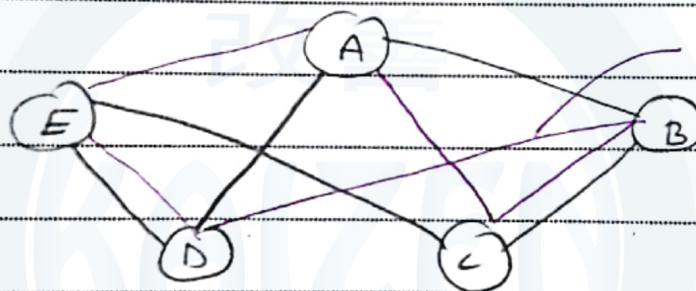
	(B D)	(D E)	(E C)	(C B)
A	9+14-7	12+14-8	12+14-10	14+9-6
	$\textcircled{16}$	18	$\textcircled{16}$	17



Choose one

$$T = \{B-A-D-E-C-B\}$$

$$= 16 + 31 = \textcircled{47} \checkmark$$



another tour with the same cost

↓
when we choose (E,C)



↓
and, is
objective J1

Mathematical formulation of TSP:-

→ Complete Undirected graph $G=(V,E)$, edge $(i,j) \in E$ has a cost C_{ij} and all costs satisfy Δ -ineq.

⇒ The Symmetric TSP is formulated as follows:-

note: that $i < j$ for each $(i,j) \in E$.

for $\{1,2,3,4,5\}$

$\{(1,2), (1,3), (1,4), (1,5)$

$(2,3), (2,4), (2,5)$

$(3,4), (3,5), (4,5)\}$

↓
 $j < i, i < j$

Let $X_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ is included in tour} \\ 0 & \text{otherwise (0.0).} \end{cases}$

↑
variable
if 0 or 1

Cost of arc

$$\Rightarrow \min. \sum_{(i,j) \in E} C_{ij} X_{ij} \Rightarrow \sum_i \sum_j C_{ij} X_{ij} \quad (1)$$

→ Coding.

s.t.

$$\sum_{i \in V: (i,j) \in E} X_{ij} + \sum_{j \in V: (i,j) \in E} X_{ji} = 2 \quad \forall j \in V \quad (2)$$

$$\sum_{(i,j) \in E} X_{ij} + \sum_{(j,i) \in E} X_{ji} \geq 2 \quad \forall S \subset V \rightarrow \text{subset of } V. \quad (3)$$

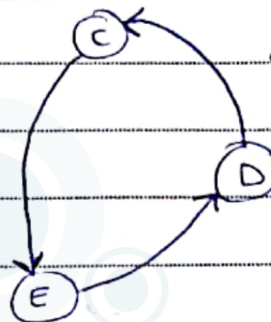
$$X_{ij} \in \{0,1\} \quad \forall (i,j) \in E \quad (4)$$

Scanned with CamScanner



Not Feasible

↓
 jēl 0.15 is
 jēl 0.15 is



initially unvisited nodes

1 Successor $b1 \leftarrow A$

1 predecessor

↓
visited once

Const (2) alone can result in Subtour trap

→ we need another Const. Set to eliminate

Subtours

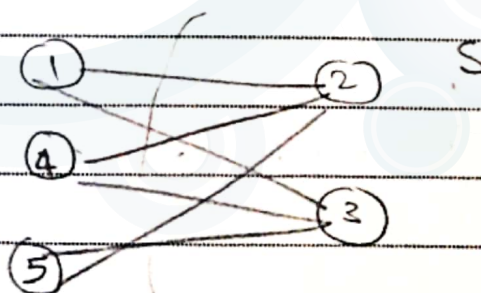
Const. Set (3): Subtour elimination

→ we should have one such Constraint

for every possible subset of $\boxed{\text{Size } \geq 1 \leq 3}$

The circuit is the same as the one in the set

5 Nodes

for $S = \{2, 3, 3\}$ ↑
 Subtour $V \setminus S \leftarrow$ Subtour of the set of all nodes $\rightarrow 1 \leftarrow$ → $V \setminus S = \{1, 4, 5\}$ Sub
 Complement S^c 

$$(x_{24} + x_{25} + x_{34} + x_{35}) + (x_{12} + x_{13}) \geq 2$$

hina 2
 move all sub
 set 1 at sub size

For $S = \{1, 2, 3\}$ $V \setminus S = \{4, 5\}$

$\rightarrow (x_{14}, x_{15} + x_{24} + x_{25} + x_{34} + x_{35}) \Rightarrow \text{SCV}$

$\Rightarrow |S| = 1, 4, 5 \rightarrow$ four subsets to look for

$\rightarrow$ Size 4 $\rightarrow$ Complement of Size (1)

Const (2)

takes care of
Sub set of size (1)

$\rightarrow$ Size 5 $\rightarrow$ This is what we are looking for

$$2 \leq |S| \leq \left\lceil \frac{|V|}{2} \right\rceil$$

$$2 \leq |S| \leq \left\lceil \frac{S}{2} \right\rceil$$

$$2 \leq |S| \leq 3$$

Summary:-



Const. Set (3):- it is only necessary to formulate a Constraint of type (3) for Subsets of size $2 \leq |S| \leq \left\lceil \frac{|V|}{2} \right\rceil$

- why not of size (1)? $\rightarrow$ Const. Set (2) takes care of it

- for subset of size $> \left\lceil \frac{|V|}{2} \right\rceil$, there subset are complements to some subset of size $< \left\lceil \frac{|V|}{2} \right\rceil$

- Subsets of size $|V| \Rightarrow$ it is the four we are looking for.

FIVE APPLE

Const. Set (3A), Introduced here, is an alternative for Constraint set (3): Would replace Constraint set (3).

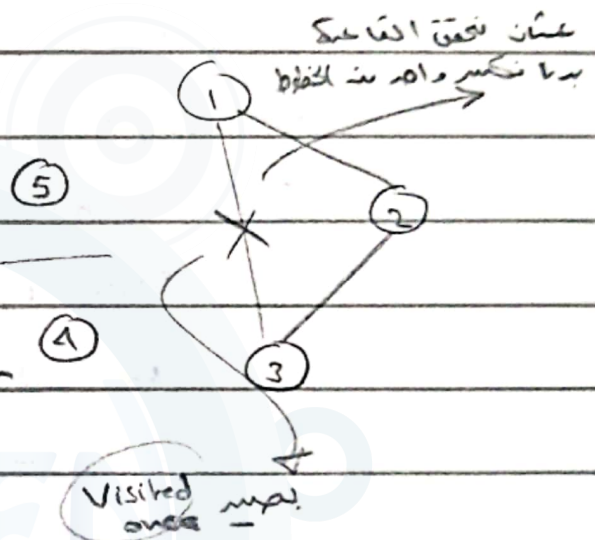
$$\sum_{(i,j) \in E: i \in S, j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset V, 2 \leq |S| \leq \left\lceil \frac{|V|}{2} \right\rceil$$

(i,j) ∈ E: i ∈ S, j ∈ S → (i,j) ∈ S

For $S = \{1, 2, 3\}$

$$(x_{12} + x_{13} + x_{23}) \leq 3 - 1$$

↳ Subtour elimination Constraint



⇒ Vehicle routing problem:-

* TSP: all Customers requests are served by a single vehicle (tour).

* In the VRP → m vehicles are based in depot and the problem is to find the set of m least-cost tours, based at the depot, serving all (or a subset) of Customers requests.

Subject: _____

(90)

الموضوع:

Date: _____

/

/

/

/

التاريخ:

* Common Operational Constraints in VRP:

[1] Number of vehicles n

↳ Can be fixed, or a decision variable.

[2] Capacitated vehicles → Capacity Q

↳ # of weight, volume

that can fit in a truck.

[3] Work shift duration → Capacity Constraint

[4] Time windows → "for Customer"

Show up no earlier than 8:00 am &

no later than 10:00 pm.

Subject:

(91)

الموضوع: (23)

الموضوع:

Date:

/ /

/ /

التاريخ:

* The Classic VRP Includes the Capacitated vehicles Constraint :-

→ Suppose a homogeneous fleet of vehicles, each with Capacity Q , is based at a depot, and each of n Customers demands that a quantity q_i , $i = 1, 2, \dots, n$.

be picked up or delivered, and demand cannot be split between vehicles. Travel between locations i & j incurs Cost c_{ij} .

→ The VRP is to find a set of vehicles tours, each beginning and ending at the depot, such that each Customer's demand is served, No vehicle violates Capacity Constraint, and total travel Cost of all vehicles is minimized.

⇒ Which problem that we have studied is a special Case of the VRP?

↳ TSP with $m = 1$

$Q = \infty$ [No Constraint on Capacity]

→ $Q = \infty \equiv q_i = 0$

Another problem with important application in transportation and logistics problem is a special

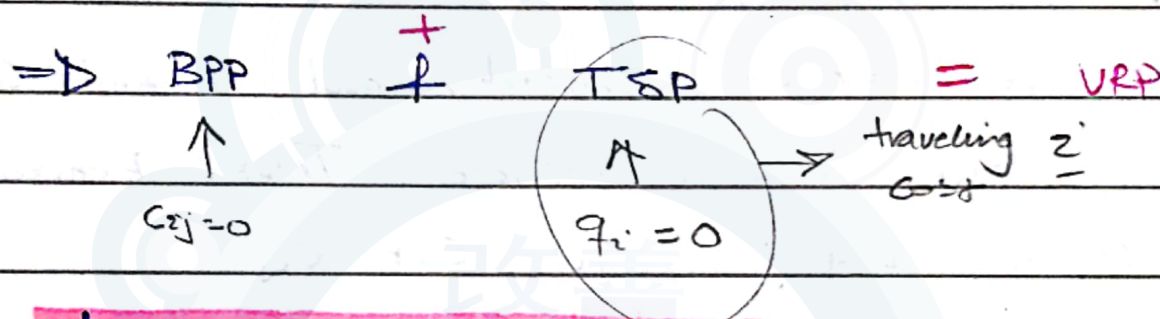
Case of VRP ⇒ Bin Packing Problem

ABC

⇒ Bin packing problem:-

- Assuming $c_{ij} = 0$ for all arcs (i, j)
 ↳ this means that the travel cost is Negligible.

- The Capacity Constraints remains



⇒ Bin packing problem:-

↳ Let m be the number of items to be loaded, and n be the number of bins.

* Let p_i , $i=1, 2, \dots, m$ be the weight of item i .

* let q_j , $j=1, 2, \dots, n$ be the Capacity of bin j .

* Items Cannot be split to bins. The problem is to assign each item to a bin such that:

1) total weight of items in bin $j \leq q_j$ ↳ Capacity of the truck

2) Smallest # of bins is used

- Variants of Packing Problems :-

• offline $\rightarrow$ All item sizes known in advance.
When you solve the problem.

• Online $\rightarrow$ Input data is revealed over time

$\Rightarrow$ Mathematical Formulation of the BPP:-

Let $y_j = \begin{cases} 1 & \text{if bin } j \text{ is used} \\ 0 & \text{o.w.} \end{cases}$

Let $x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is assigned to bin } j \\ 0 & \text{o.w.} \end{cases}$

$x_{11} = 1$ $x_{21} = 1 \rightarrow$ bin 1 contains items 1 and 2
Item (1) $\uparrow$ bin (1) Item (2) $\uparrow$ bin (1) (1) 5, 10, 15

$x_{22} = x_{23} = 0 \rightarrow$ as (2) is not item 2
(1) 5, 10, 15

$$\min \sum_{j=1}^n y_j$$

--- (1)

assign for 1 bin $\rightarrow$ 1 = 1
 $\rightarrow x_{11} + x_{12} + x_{13} + \dots + x_{1n} = 1$

$$\text{s.t.} \sum_{j=1}^n x_{ij} = 1$$

$\forall i = 1, \dots, m$ --- (2)

$$\sum_{i=1}^m p_i x_{ij} \leq q_j \quad \forall j = 1, \dots, n \quad \text{--- (3)}$$

$$x_{ij} \leq y_j \quad \forall i = 1, \dots, m, j = 1, \dots, n \quad \text{--- (4)}$$

$$x_{ij}, y_j \in \{0, 1\}$$

* Objective (1): minimize # of bins Used.

* Const. set (2):- Each Item is assign to exactly one bin.

* Const. set (3):- bin Capacity is not violated

* Const. set (4):- we cannot assign an item to a bin unless the bin is opened

⇒ Lower bounds of BPP:-

• Assume $q_j = Q$ for all $j = 1, \dots, n$

then what is a valid lower bound of the # of bins required for item $i = 1, \dots, m$?

↳ $q_j = Q \forall j$

⇒ LB is $\left\lceil \frac{\sum p_i}{Q} \right\rceil$ round up

↳ Best Solution [Optimal]

⇒ Suppose that bins have different size.

$q_j \neq Q$.

∴ take the largest bin ⇒ let $r = \max q_j$.

$$LB = \left\lceil \frac{\sum p_i}{r} \right\rceil$$

max. Capacity

70	20
20	15
35	10
4	

← إلى هنا نبدأ bins

* BPP has no efficient algorithm for its solution
 $\Rightarrow$ Heuristics. $\downarrow$ Optimal Solution.

$\Rightarrow$ Heuristic Solution Approach:-

* Online problem:-

- $\rightarrow$ Next fit (NF), 2-optimal
- $\rightarrow$ First fit (FF), 1.7-optimal
- $\rightarrow$ Best fit (BF), 1.7-optimal

* offline problem:-

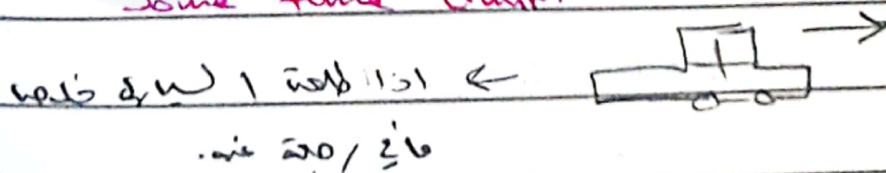
- (NP1)
- $\rightarrow$ Next fit decreasing (NFD)
 - $\rightarrow$ First fit decreasing (FFD), 1.5-optimal
 - $\rightarrow$ Best fit decreasing (BFD), 1.5-optimal

* Variants of packing problem:-

$\rightarrow$ offline: all items sizes are known in advance.

$\rightarrow$ online: Input data is revealed overtime.

$\hookrightarrow$ Irrevocable decisions made within
 Some time limit.



The difference in Online & offline heuristics is that in the offline versions items are sorted in decreasing order when the heuristic begins.

* Example $\rightarrow$ Items with weights/sizes 4, 9, 5, 7, 10, 3, 2, 11 becomes 11, 10, 9, 7, 5, 4, 3, 2

$\rightarrow$ In each of the heuristics below, let L be a list of item weights - $L = \{P_1, P_2, \dots, P_m\}$

The Example, we will use for each heuristic has

$L \rightarrow q_j = Q = 20$

orders

and $L = \{16, 10, 14, 12, 4, 8, 3, 12, 7, 14, 9, 6, 3\}$

a quick LB = $\left\lceil \frac{\sum P_i}{Q} \right\rceil = \left\lceil \frac{112}{20} \right\rceil = 6$ bins.

* Online :-

I. Next Fit heuristic :- (NF) :-

- Initialization :- place item 1 in bin 1 and remove from L .

let $j=1, i=2$.

- Iterations:-

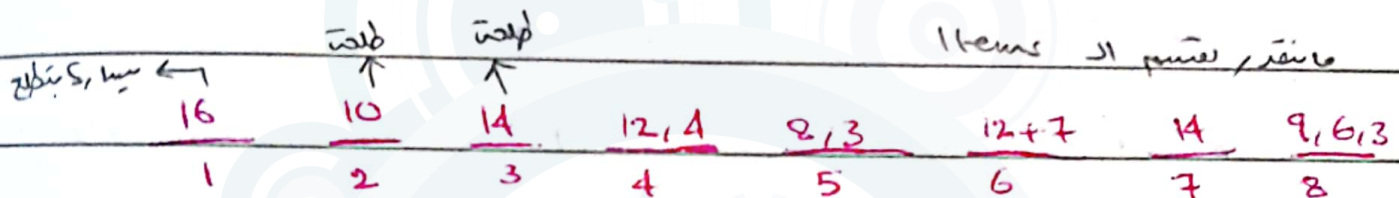
$\rightarrow$ If item i fits in bin j , place i in j .
other wise open a new bin $j+1$ and place i in $j+1$.

* let $j = j + 1$.

→ Remove i from list L , let $i = i + 1$

→ While Items remain in L , repeat

** Advantages of NF:- allows immediate dispatch of bin j (Vehicle j) once bin $j+1$ is opened.



8 bins ✓

II: First Fit heuristic (FF):-

- Initialization :- place Item 1 in bin 1 and remove from bin 2.

* let $i = 2, j = 1$.

- Iterations:-

→ place i in the lowest numbered bin for which it fits. If Item i does not fit in any open bin, open a new bin $j+1$ and place i in it.

→ Remove i from L , let $i = i + 1$.

→ While Items remain in L , repeat.

$$L = \{16, 10, 14, 12, 4, 8, 3, 12, 7, 14, 9, 6, 3\}$$

16, 4	10, 8	14, 3, 3	12, 7	12, 6	4	9
1	2	3	4	5	6	7

→ 7 bins

TU Best fit heuristic: (BF)

- Initialization:- place Item 1 in bin 1 and remove from L.

* Let $j=1$, $i=2$.

- Iterations:-

→ find the bin j whose contents (total weights of items) are maximum but greater than

bin j and P_i → i fits in this bin.

* If Item i does not fit in any open bin, open a new bin $j+1$ and place i in it.

→ Remove i from L. let $i = i+1$.

→ While Item remain in L, repeat.

$$L = \{16, 10, 14, 12, 4, 8, 3, 12, 7, 14, 9, 6, 3\}$$

16, 4	10, 8	14, 3, 3	12, 7	12, 6	4	9
1	2	3	4	5	6	7

→ 6 bins ✓

Practical Issues: With FF & BF.

→ If any open bin is Completely full. at some iterations, it can be closed / dispatched, but not otherwise. So, in practice, it is necessary to add some rules. → Similar to:-

* dispatch $j \rightarrow$ once j reaches full percentage from its full capacity.

$$Q=20$$

$$\rightarrow 80\% \rightarrow = 16$$

or once bin j has been open for y time units.

or a Combination

time + %

* | Practice |

$$L = \{ 3, \dots, 1 \}$$

→ Create the solution to this $\uparrow$ instance of BPP.

Using NFD, FFD, BFD.

the only difference is that you will initialize each heuristic with L decreasing order

Sorted in a

$$\rightarrow L = \{ 16, 14, 14, 12, 12, 10, 9, 8, 7, 6, 4, 3, 3 \}$$

Result of $j \leftarrow$	NFD = 8 bins	FFD = 6 bins	BFD = 6 bins	] Offline
	↓ 8	↓ 7	↓ 6	

ABC

Recall, in the VPP:-

- Fleet of Homogeneous vehicles, each with Capacity Q , based at depot.
- n Customers with demand q_i .
- Find least Cost Set of vehicles tours (based at depot) serving all customers.

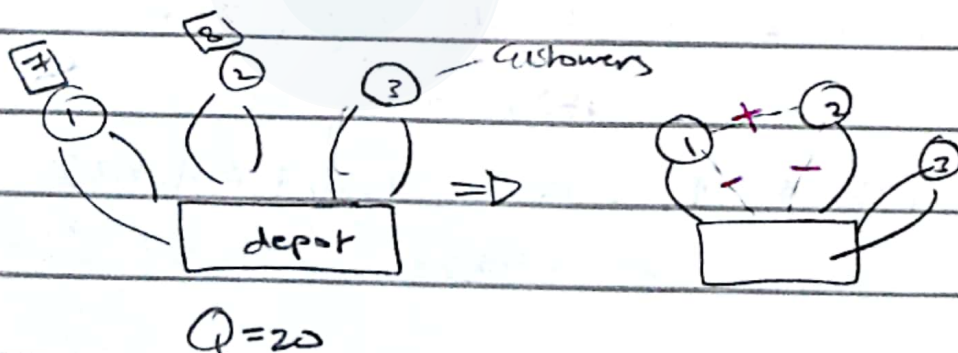
⇒ Heuristics for VPP:-

1. Clarke-Wright Saving Heuristic
2. Route - first cluster-second. (RFS) Heuristics.
3. cluster - first Route - Second (CFRS) Heuristics

Multiple Heuristics.

II Clarke-Wright Saving Heuristic :-

↳ Begin with a separate tour for each customer then combines tours, if feasible with respect to demand. and if saving result.



* Let T_i and T_j be two tours, and let $\text{end}(i)$ denotes the end of tour i , and $\text{begin}(j)$ be the beginning of tour j

* Savings: $S_{ij} = C_{\text{end}(i), 0} + C_{0, \text{begin}(j)} - C_{\text{end}(i), \text{begin}(j)}$

→ Clarke - Wright Savings Heuristic:

1) For each node $i \in N$, Construct a tour T_i beginning at the depot, traveling to node i , and returning to the depot.

Let S be the set of tours.

2) Calculate the Savings S_{ij} for each pair of Nodes i, j $S_{ij} = C_{0i} + C_{0j} - C_{ij}$

3) Sort the Savings in non-increasing [decreasing] order.

4) Walk down the Saving Combination list, and find next feasible pair (i, j) such that:-

- i and j are on different tours, and i is the last node on its tour T_k and j is the first node on its tour T_l

- $S_{ij} \geq 0$ Savings that added ≥ 0

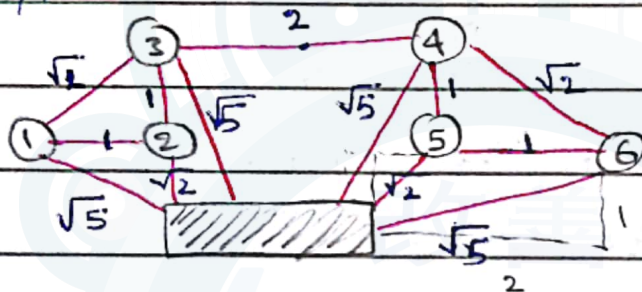
- $\sum_{T \in T_i \cup T_j} q_k \leq Q$ demands at a nodes in tours $<$ capacity

5 Add the Combined tour $T_i + T_j$ to S and remove T_i and T_j from S

$$T_i + T_j = T_g$$

6 Repeat this step as long as feasible combinations remain in the list.

Example:-



* Vehicle Capacity is 5

* demand at Customers 1, 2, 3 is 3

* demand at Customers 4, 5, 6 is 2

enter

1

$$T_1 = \{0-1-0\}$$

$$T_2 = \{0-2-0\}$$

$$T_3 = \{0-3-0\}$$

$$T_4 = \{0-4-0\}$$

$$T_5 = \{0-5-0\}$$

$$T_6 = \{0-6-0\}$$

$$LB \text{ on Vehicles} = \left\lceil \frac{\sum d_i}{Q} \right\rceil$$

$$= \left\lceil \frac{9+6}{5} \right\rceil = 3$$

lower = most efficient
tours are less
impr. in cost
disposal

Best scenario

$$S_{12} = \sqrt{5} + \sqrt{2} - 1 = 2.65$$

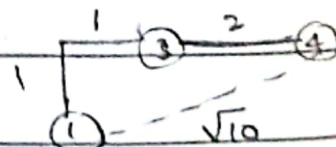
$$S_{13} = \sqrt{5} + \sqrt{5} - \sqrt{2} = 3.06$$

$$S_{14} = \sqrt{5} + \sqrt{5} - \sqrt{10} = 1.31$$

$$S_{15} = \sqrt{5} + \sqrt{2} - 3 = 0.65$$

$$S_{16} = \sqrt{5} + \sqrt{5} - 4 = -0.35 \rightarrow \text{No Savings}$$

$$S_{23} = \sqrt{2} + \sqrt{5} - 1 = 2.65$$



Subject: (103) الموضوع:

Date: / / التاريخ:

$$S_{24} = \sqrt{2} + \sqrt{5} - \sqrt{5} = 1.41$$

$$S_{25} = \sqrt{2} + \sqrt{2} - 2 = 0.82$$

$$S_{26} = \sqrt{2} + \sqrt{5} - 3 = 0.65$$

$$S_{34} = \sqrt{5} + \sqrt{5} - 2 = 2.47$$

$$\rightarrow S_{35} = \sqrt{5} + \sqrt{2} - \sqrt{5} = 1.41$$

$$S_{36} = \sqrt{5} + \sqrt{5} - \sqrt{10} = 1.31$$

$$S_{45} = \sqrt{5} + \sqrt{2} - 1 = 2.65$$

$$S_{46} = \sqrt{5} + \sqrt{5} - \sqrt{2} = 3.06 \checkmark$$

$$S_{56} = \sqrt{2} + \sqrt{5} - 1 = 2.65$$

$$\begin{matrix} C_1 & C_3 \\ \Rightarrow S_{13} \rightarrow 3+3 > 5 & \text{infeasible } X \end{matrix}$$

$$\Rightarrow S_{46} \rightarrow 2+2 < 5 \quad \text{feasible. } \checkmark$$

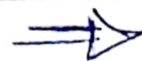
$$* T_1, T_2, T_3, T_{46}, T_5$$

$$\Rightarrow S_{12} \rightarrow 3+3 > 5 \quad X$$

$$S_{23} \rightarrow 3+3 > 5 \quad X$$

$$S_{35} \rightarrow 3+2 = 5 \quad \checkmark$$

$$* T_1, T_2, T_{35}, T_{46}$$



Subject: 101

الموضوع:

Date: / /

التاريخ:

