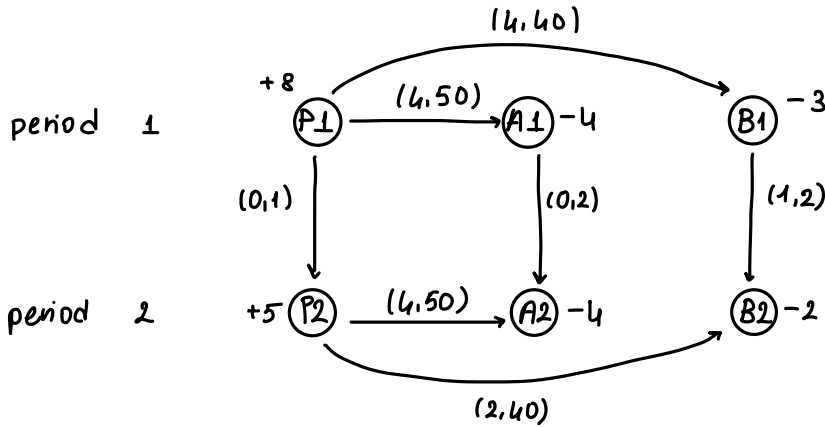


IE479 - Study Set 2 Solutions

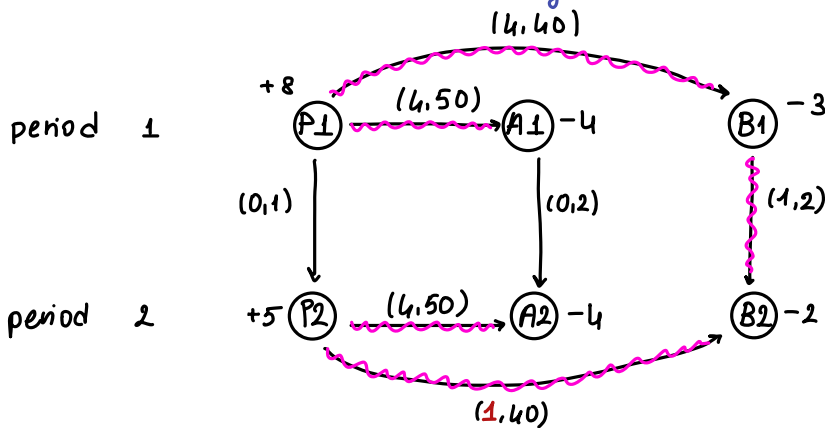
Question 1 $i \xrightarrow{(\text{flow, cost})} j$



sending 2 units to B from P at period 2 does not satisfy the flow balance.

$$\text{cost} = (8 \times 25) + (5 \times 24) + (4 \times 50) + (4 \times 40) + (4 \times 50) + (2 \times 40) + 2 = 962$$

fix feasibility



$$\text{cost} = (8 \times 25) + (5 \times 24) + (4 \times 40) + (4 \times 50) + 2 + (4 \times 50) + 40 = 922$$

arcs in the tree :

$$\begin{cases} (P1, A1) \rightarrow \pi_{P1} - \pi_{A1} = 50 \\ (P1, B1) \rightarrow \pi_{P1} - \pi_{B1} = 40 \\ (B1, B2) \rightarrow \pi_{B1} - \pi_{B2} = 2 \\ (P2, A2) \rightarrow \pi_{P2} - \pi_{A2} = 50 \\ (P2, B2) \rightarrow \pi_{P2} - \pi_{B2} = 40 \end{cases}$$

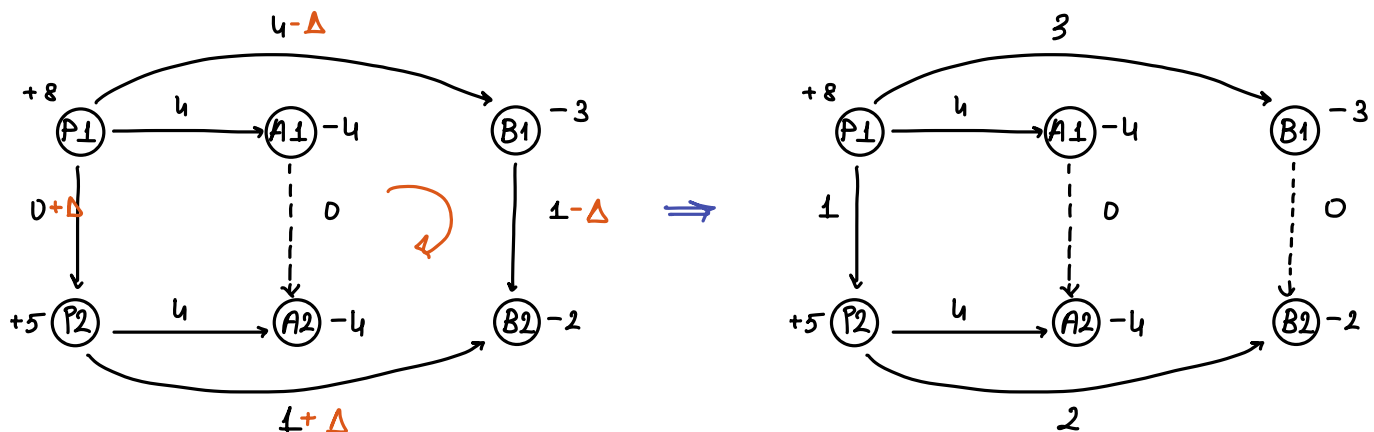
6 unknowns $\Rightarrow$ let $\pi_{A2} = 0 \Rightarrow$ 5 equations

$$\begin{aligned} \pi_{P1} &= 52 \\ \pi_{P2} &= 50 \\ \pi_{A1} &= 2 \\ \pi_{A2} &= 0 \\ \pi_{B1} &= 12 \\ \pi_{B2} &= 10 \end{aligned}$$

arcs not on the tree :

$$\begin{aligned} (P1, P2) &\rightarrow x_{P1P2} = 0 \rightarrow \text{check for } \pi_{P1} - \pi_{P2} \leq C_{P1P2} \Rightarrow 52 - 50 \leq 1 \quad \times \\ (A1, A2) &\rightarrow x_{A1A2} = 0 \rightarrow \text{" " } \pi_{A1} - \pi_{A2} \leq C_{A1A2} \Rightarrow 2 - 0 \leq 2 \quad \checkmark \end{aligned}$$

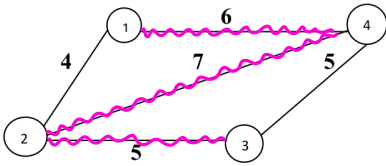
enter the arc (P1, P2) :



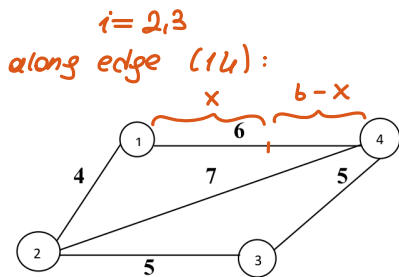
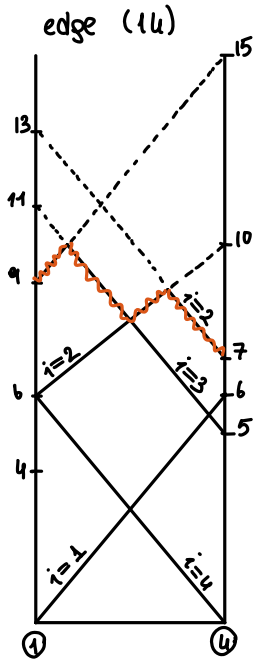
$$\text{new cost} : (8 \times 25) + (5 \times 24) + (3 \times 40) + (4 \times 50) + 1 + (4 \times 50) + (2 \times 40) = 921$$

$$\Rightarrow \text{cost reduction} : 922 - 921 = 1 \$$$

Question 2



Hakimi p-center algorithm



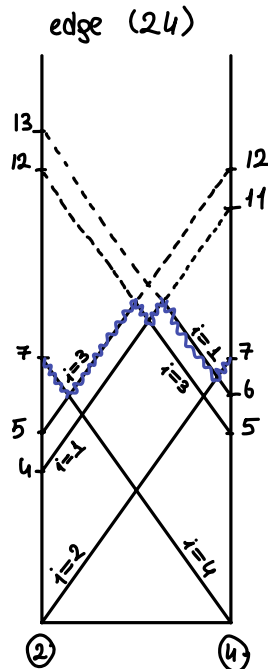
distance from ② : $4+x$
" " ③ : $11-x$

$$4+x = 11-x$$

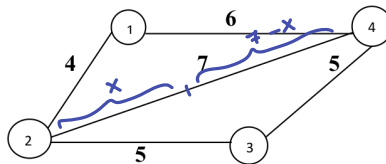
$$2x = 7$$

$$x = 3,5$$

$$\Rightarrow \text{worst dist.} = 7,5$$



$i=1,2,3,4$
along edge (24):



distance from ① : $4+x$ OR $13-x$
" " ② : x
" " ③ : $5+x$ OR $12-x$
" " ④ : $7-x$

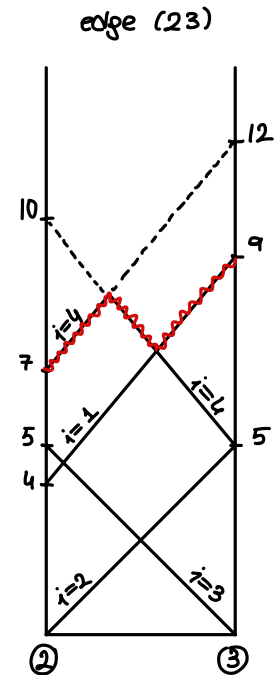
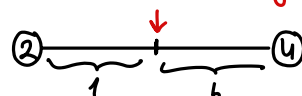
$$5+x = 7-x$$

$$2x = 2$$

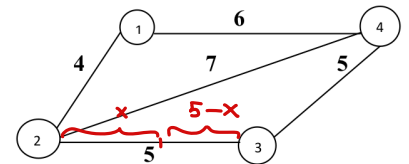
$$x = 1$$

$$\Rightarrow \text{worst dist.} = 6 \star$$

$$\Rightarrow \text{locate on the edge (24)}$$



$i=1,4$
along edge (23):



distance from ① : $4+x$
" " ④ : $10-x$

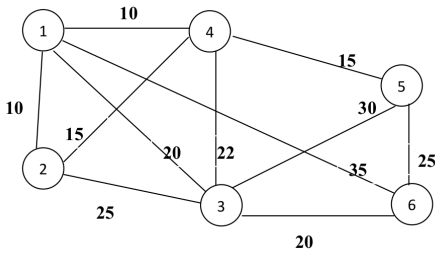
$$4+x = 10-x$$

$$2x = 6$$

$$x = 3$$

$$\Rightarrow \text{worst dist.} = 7$$

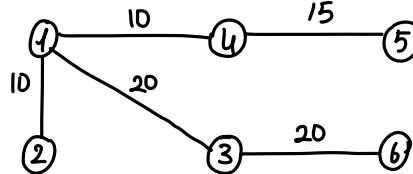
Question 3



	1	2	3	4	5	6
1	0	10	20	10	25	35
2		0	25	15	55	45
3			0	22	30	20
4				0	15	40
5					0	25
6						0

Christofides Algorithm

1. min cost spanning tree $\Rightarrow$



$$E'_T = \{(1,2), (1,3), (1,4), (3,6), (4,5)\}$$

$\hookrightarrow$ arcs on the tree

$$V'_\Delta = \{1, 2, 5, 6\}$$

$\hookrightarrow$ odd edge vertices

2. least cost perfect match among V'_Δ

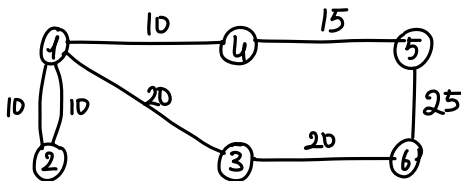
$$\begin{array}{ll} 1-2 & 5-6 \\ 1-5 & 2-6 \\ 1-6 & 2-5 \end{array} \quad \begin{array}{l} 10+25 = 35 \\ 25+45 = 70 \\ 35+30 = 65 \end{array}$$

$$\star \Rightarrow E'_\Delta = \{(1,2), (5,6)\} \rightarrow \text{edges in the match}$$

$$E'_H = E'_T \cup E'_\Delta = \{(1,2), (1,3), (1,4), (3,6), (4,5), (5,6)\}$$

$\hookrightarrow$ double

$$H = (V', E'_H)$$

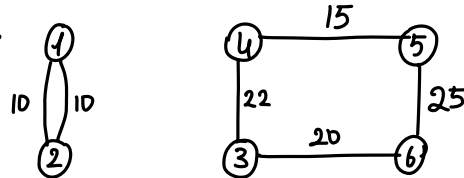


$$\text{cost} = 110$$

3. Vertices with degree $> 2 \Rightarrow$ ① $\Rightarrow$ eliminate 2 nodes, add 1

a) eliminate (1,4) and (1,3), add (3,4) $\Rightarrow$

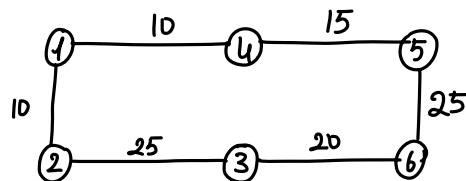
$$\Delta = -10 - 20 + 22 = -8$$



No tour, cannot be chosen!

b) eliminate (1,3) and (1,2), add (2,3) $\Rightarrow$

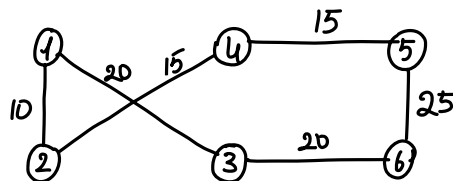
$$\Delta = -20 - 10 + 25 = -5$$



$$\text{cost} = 105$$

c) eliminate (1,2) and (1,4), add (2,4) $\Rightarrow$

$$\Delta = -10 - 10 + 15 = -5$$

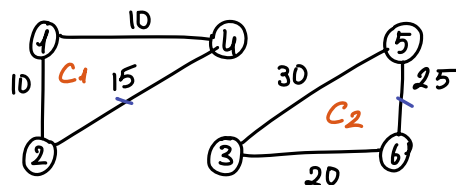


$$\text{cost} = 105$$

$\Rightarrow$ choose b) OR c)

Patching Heuristic

relaxed solution :



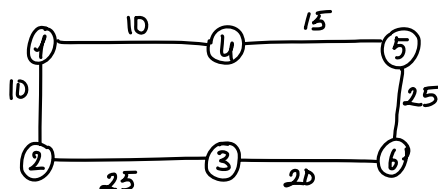
$$N = \{1, 2, 3, 4, 5, 6\}$$

$$C = \{C_1, C_2\}$$

$$\text{subtour number} \rightarrow P=2$$

⇒ merge C_1 and C_2 by removing arcs (24) and (35), adding (23) and (45) most costly ones

$$\text{min cost perfect match: } 2-3 \quad 4-5 \Rightarrow \text{cost} = 25 + 15 = 40 \quad \star$$



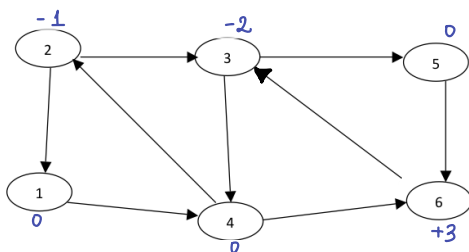
$$\Delta = -15 - 30 + 25 + 15 = -5$$

$$\text{cost} = 105 \quad P=1, \text{ stop!}$$

Question 4

out > in ⇒ deficit, in > out ⇒ surplus

a)



S = set of surplus vertices

D = " " deficit " "

C_{ij} = cost of shortest path from i to j

S_i = surplus of node $i \in S$

d_j = deficit " " $j \in D$

X_{ij} = # of chains added btw i and j

$$S(\text{surplus}) = \{6\}$$

$$D(\text{deficit}) = \{2\}$$

$$\rightarrow S_6 = 2 - 1 = 1 \quad (\text{surplus of node 6})$$

$$C_{62} = 3$$

$$\rightarrow d_2 = 2 - 1 = 1 \quad (\text{deficit of node 2})$$

$$(\text{deficit of node 3})$$

$$\text{solve } \min \sum_i \sum_j C_{ij} X_{ij}$$

$$\text{s.t. } \sum_{j \in D} X_{ij} = S_i \quad \forall i \in S$$

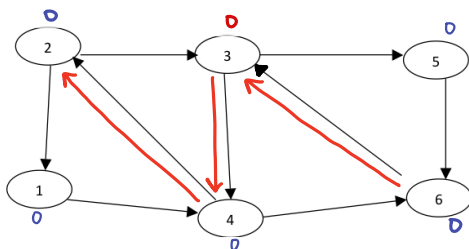
$$\sum_{i \in S} X_{ij} = d_j \quad \forall j \in D$$

$$X_{ij} \geq 0 \text{ and integer}$$

$$\left. \begin{array}{l} \text{for } i=6 \Rightarrow X_{62} = 1 \\ \text{for } j=2 \Rightarrow X_{62} = 1 \end{array} \right\}$$

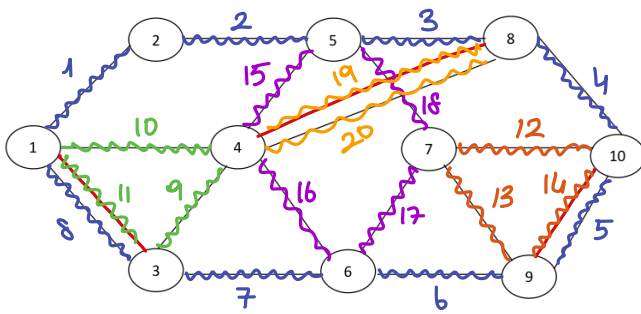
$$\text{solution: } X_{62} = 1$$

$$\text{OFV} = C_{62} \cdot X_{62} = 3$$



⇒ all nodes have even degrees!

b)



! Graph is connected ✓

! Not all vertices have even edges x

⇒ odd degree vertices = {1, 3, 4, 8, 9, 10}

⇒ add chains to make all vertices even
↳ new chains : (13), (48), (910)

Now apply End Pairing Algorithm with additional edges :

1st cycle : 1-2-3-4-5-6-7-8

2nd cycle : 9-10-11

↳ merge ⇒ 1-2-3-4-5-6-7-9-10-11-8

3rd cycle : 12-13-14

↳ merge ⇒ 1-2-3-4-12-13-14-5-6-7-9-10-11-8

4th cycle : 15-16-17-18

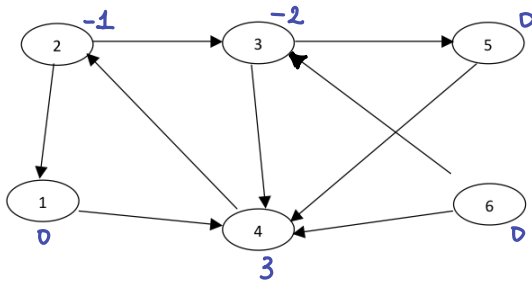
↳ merge ⇒ 1-2-15-16-17-18-3-4-12-13-14-5-6-7-9-10-11-8

5th cycle : 19-20

↳ merge ⇒ 1-2-15-19-20-16-17-18-3-4-12-13-14-5-6-7-9-10-11-8

* all edges are visited, stop !

c)



$$S = \{4\} \Rightarrow s_4 = 3$$

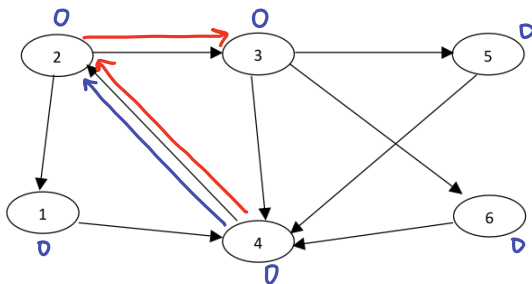
$$D = \{2, 3\} \Rightarrow d_2 = 1$$

$$d_3 = 2$$

$$C_{42} = 1$$

$$C_{43} = 2$$

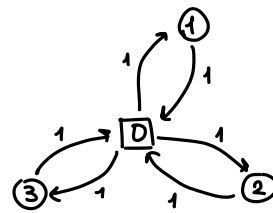
$$\begin{array}{lcl} \text{for } i=4 & \Rightarrow & x_{42} + x_{43} = 3 \\ \text{for } j=2 & \Rightarrow & x_{42} = 1 \\ \text{for } j=3 & \Rightarrow & x_{43} = 2 \end{array} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} \text{solution : } x_{42} = 1, x_{43} = 2 \\ \text{DFV} = 1 + 4 = 5 \end{array}$$



⇒ all nodes have even edges
and graph is strongly connected (Eulerian cycle)

Question 5

15 nodes (1 depot, 14 customers)
 $Q = 20 \rightarrow$ vehicle capacity
 $q_i = 5 \rightarrow$ customers' demands
 $m = 4 \rightarrow$ # of vehicles
all arc costs are 1
 $T = 5 \rightarrow$ max route length

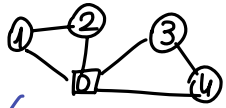


a) Savings Heuristic :

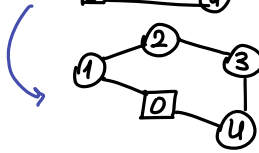
- Initially assign a route for each customer $\Rightarrow$ total cost = $14 \times 2 = 28$

- Merge a pair of routes (lets say ① and ②) $\Rightarrow$ saving : $S_{ij} = C_{i0} + C_{0j} - C_{ij}$ (0 is depot)
 $S_{12} = C_{10} + C_{02} - C_{12} = 1 + 1 - 1 = 1$

- Merge another pair (lets say ③ and ④) $\Rightarrow S_{34} = 1$



$\Rightarrow$ merge these two routes, saving is $3 + 3 - 5 = 1$



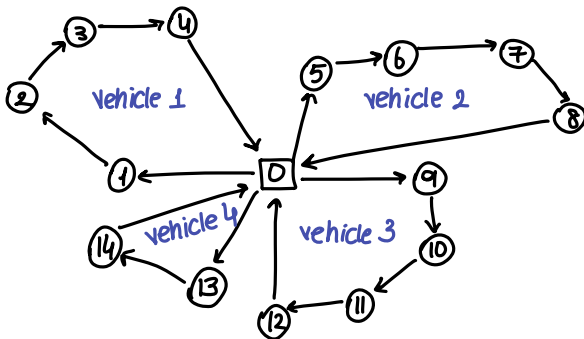
$\Rightarrow$ total traveled distance = $1 \times 5 = 5 = T \checkmark$
total satisfied demand = $4 \times 5 = 20 = Q \checkmark$ } no more expanding the route

* With this logic, we can end up with 3 routes with 4 nodes each and a route with 2 nodes, as follows :

$\hookrightarrow$ merge 5, 6, 7, and 8

$\hookrightarrow$ " 9, 10, 11, and 12

$\hookrightarrow$ " 13, and 14 (we have capacity, but no other customer is left)

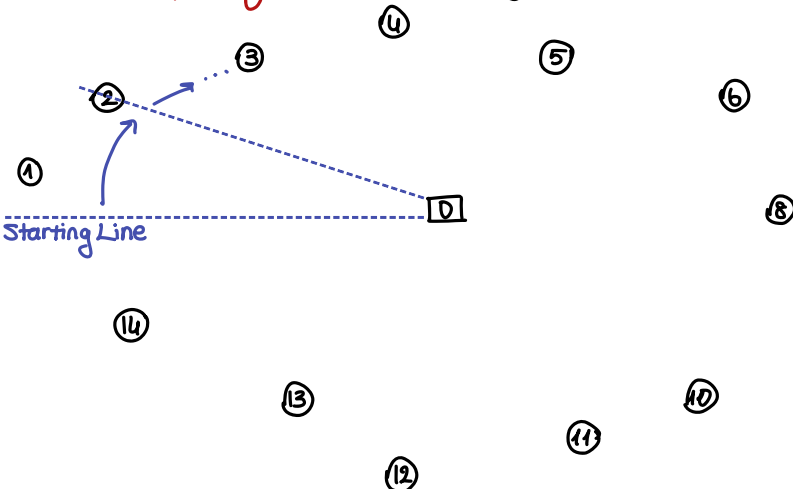


4 routes :

total cost = $5 + 5 + 5 + 3 = 18$

total saving = $28 - 18 = 10$

b) Sweep Algorithm : Imagine that the layout of nodes are as follows :



$\Rightarrow$ Sorted customer list according to their angular position around the depot :

⑦ 1-2-3-4-5-6-7-8-9-10-11-12-13-14

$\Rightarrow$ Add customers in order while satisfying capacity constraints :

⑨ Route 1 : $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$

Route 2 : $0 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 0$

Route 3 : $0 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 0$

Route 4 : $0 \rightarrow 13 \rightarrow 14 \rightarrow 0$

Total cost = 18

c) Cluster First - Route Second :

Cluster 1 = {1, 2, 3, 4}
 Cluster 2 = {5, 6, 7, 8}
 Cluster 3 = {9, 10, 11, 12}
 Cluster 4 = {13, 14}

} total demand of each cluster $\leq Q$
 route length of each cluster $\leq T$
 total cost = 18

Solving TSP for each cluster $\Rightarrow$ same routes as before (since all arc costs are equal)

Which one is better? Since the problem is symmetric (all arc costs are equal), we don't see a difference in terms of cost and route assignments.

d) Inter route improvement? $\rightarrow$ since all routes gives the less costly option, there is no improvement

Intra route improvement? $\rightarrow$ 3 routes has achieved full capacity, adding customers to these routes is infeasible.

$\rightarrow$ customer exchange between routes makes no improvement, since arc costs are equal.

Question 6 $\rightarrow$ Hungarian Method

	1	2	3	4	5	ROW MIN
1	8	3	5	2	4	$\rightarrow$ 2
2	4	5	3	6	7	$\rightarrow$ 3
3	6	2	1	3	5	$\rightarrow$ 1
4	3	7	2	4	3	$\rightarrow$ 2
5	5	2	6	3	4	$\rightarrow$ 2

subtract ROW MIN from each row :

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

COL MIN $\Rightarrow$ 1 0 0 0 1

subtract COL MIN from each column :

$C' =$

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

cover zero's
with min #
of lines
 $\Rightarrow$

$C' =$

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

5 Lines,
 $n=5$
STOP!
Optimum
sol. is achieved

$\Rightarrow$

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

Optimal
assignment :

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

cost

2
4
1
3
2

} total
cost = 12