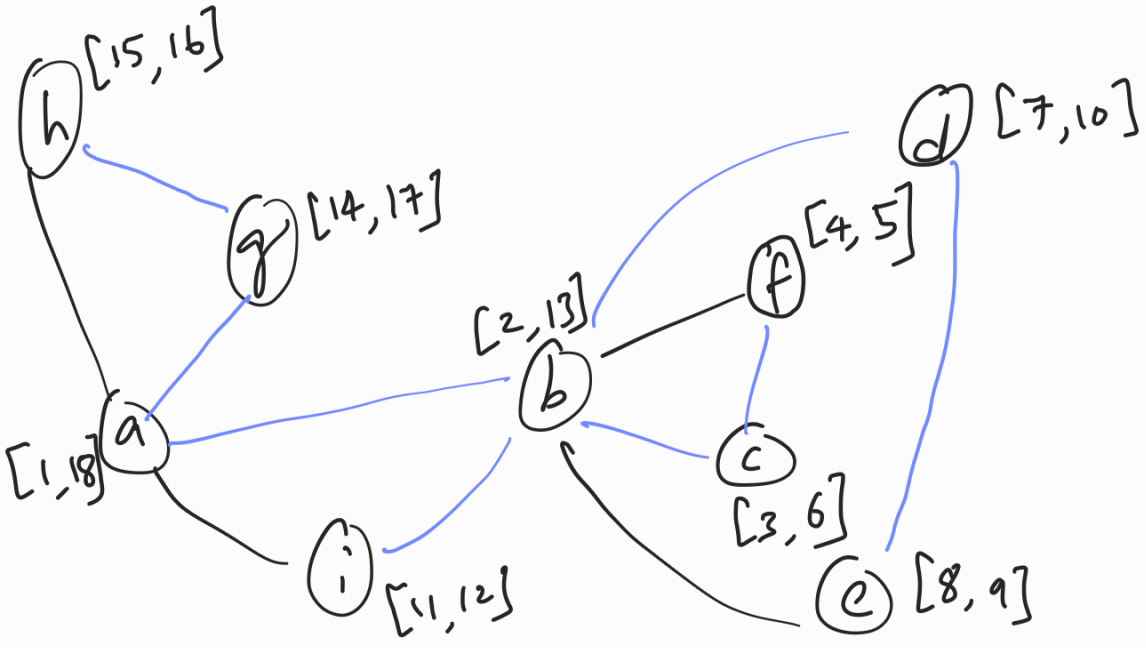
NAME :- VISWA NIHAR NUKALA

UBID :- 50414392

(2)



Adjacency list for the above graph

a → b → g → h → i

b → a → c → d → e → f → i

c → b → f

d → b → e

e → b → d

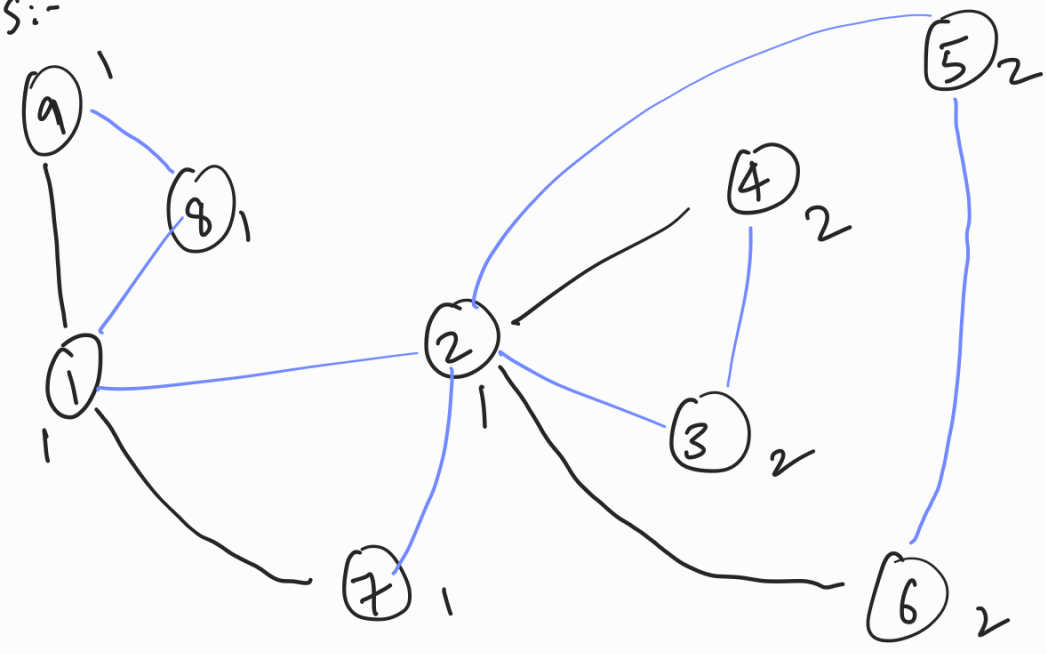
f → b → c

g → a → h

h → a → g

i → a → b

Renaming the vertices with and removing the edges not in DFS:-



Above is the graph and the respective low values computed.

Following are the cut vertices from the above graph:-

a, b

Q(3) Given: G is a strongly connected graph and for every vertex $deg_{in}(v) = deg_{out}(v)$. Hence all the conditions for existence of Eulerian circle are satisfied.

(i) Initialize two arrays to keep track of visited edges and vertices and set all the elements to false.

Note:- Any vertex can be visited multiple times but each edge can be visited only once.

(ii) Pick a starting vertex u randomly, using an untravelled edge $u \to v$ from the adjacency list and mark both the edge and vertex as visited

(iii) Do the same to v, i.e., $v \to w$ an untravelled edge from v and mark both the vertex and edge as visited.

(iv) Continue this until we reach a vertex x, where all edges to this vertex are visited and as $deg_{in}(v) = deg_{out}(v)$ for all vertices, we have $x = u$, the vertex we have started from and hence have completed tour P.

(v) If P has all edges, we are done.

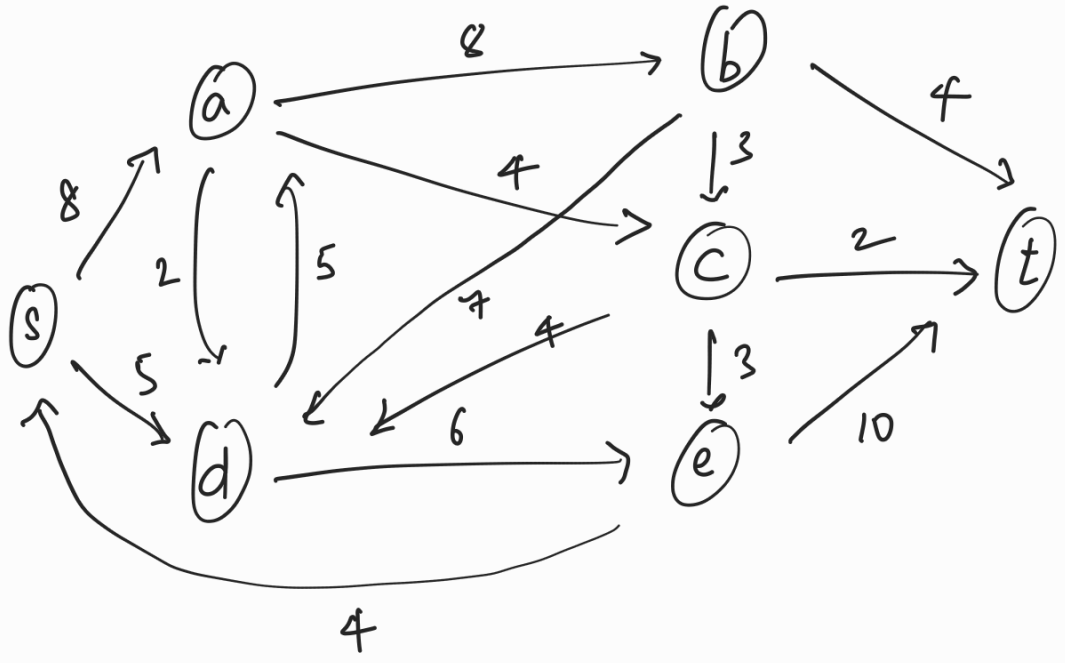
(vi) Else we have a vertex y with unvisited edges. We then start another tour from y, Q and then merge P and Q.

(vii) Continue this process until all the edges are visited.

Time Complexity analysis :-

Here we are traversing over all edges to visit them as the termination condition and we know that the time complexity to traverse all edges is $O(|E|+|V|)$ i.e., $O(n+m)$.

(4)



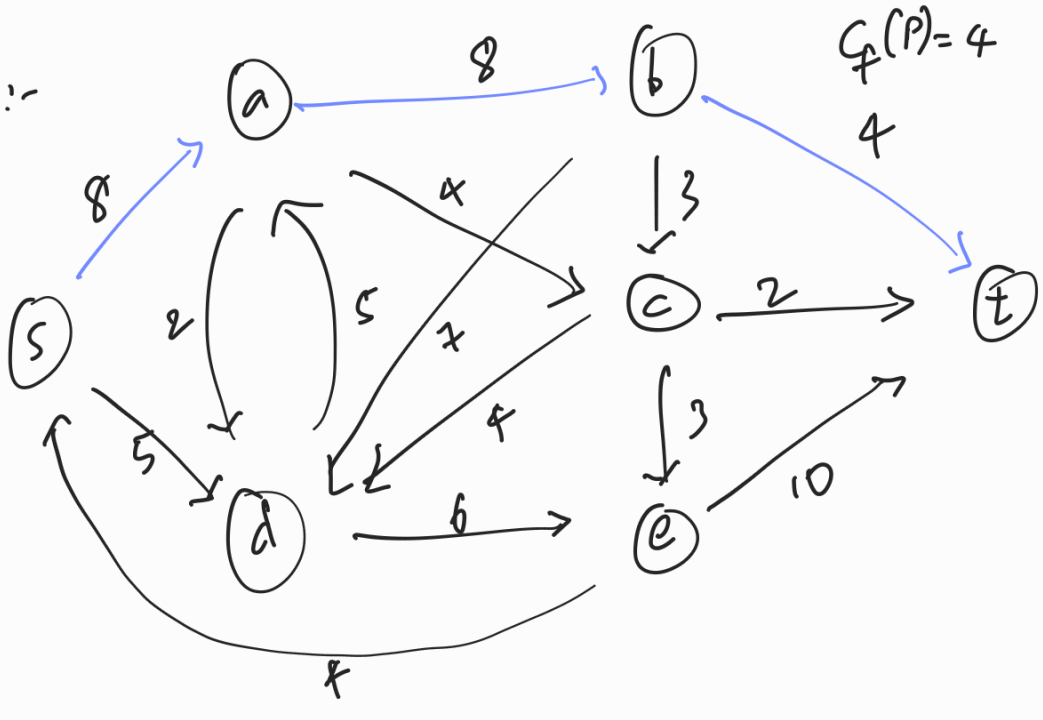
1. Edge set crossing the cut $(\{s, a, b\}, \{c, d, e, t\})$

$\{s \to d, a \to d, a \to c, b \to d, b \to c, b \to t\}$

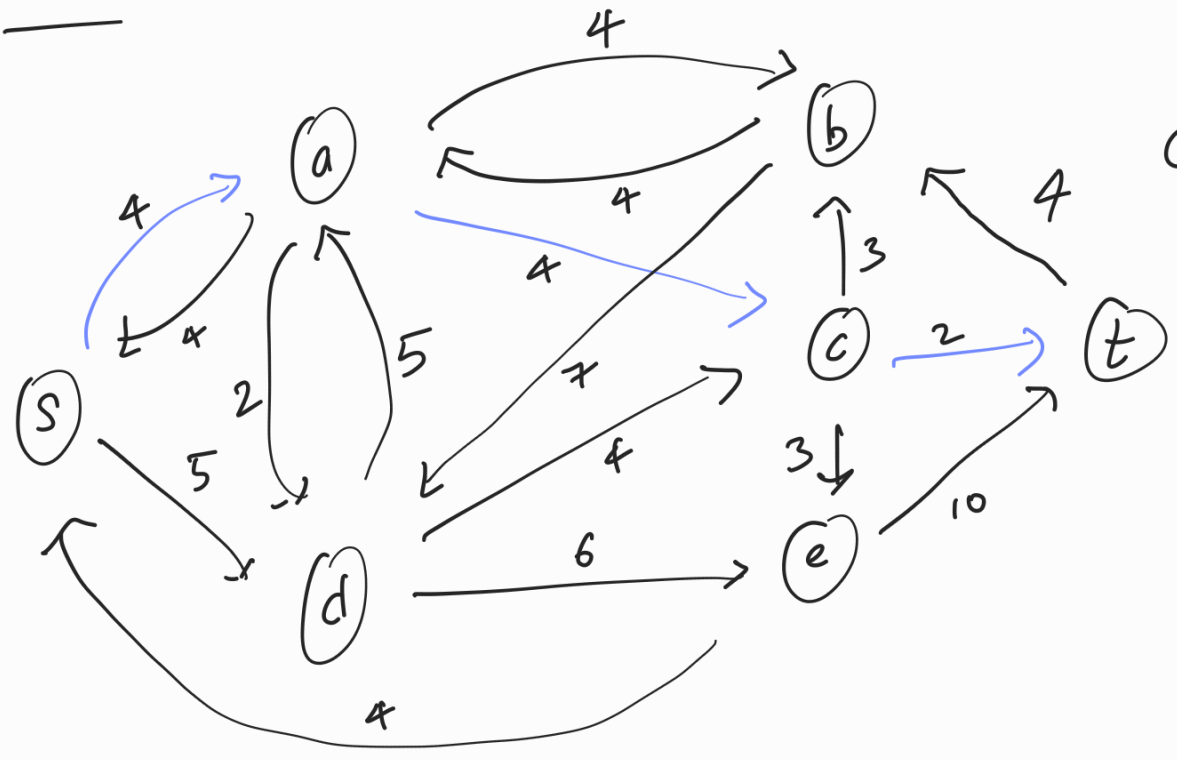
Capacity of the edge cut is :- $5 + 2 + 4 + 7 + 3 + 4$

$= 25$

2. Iteration (1) :-

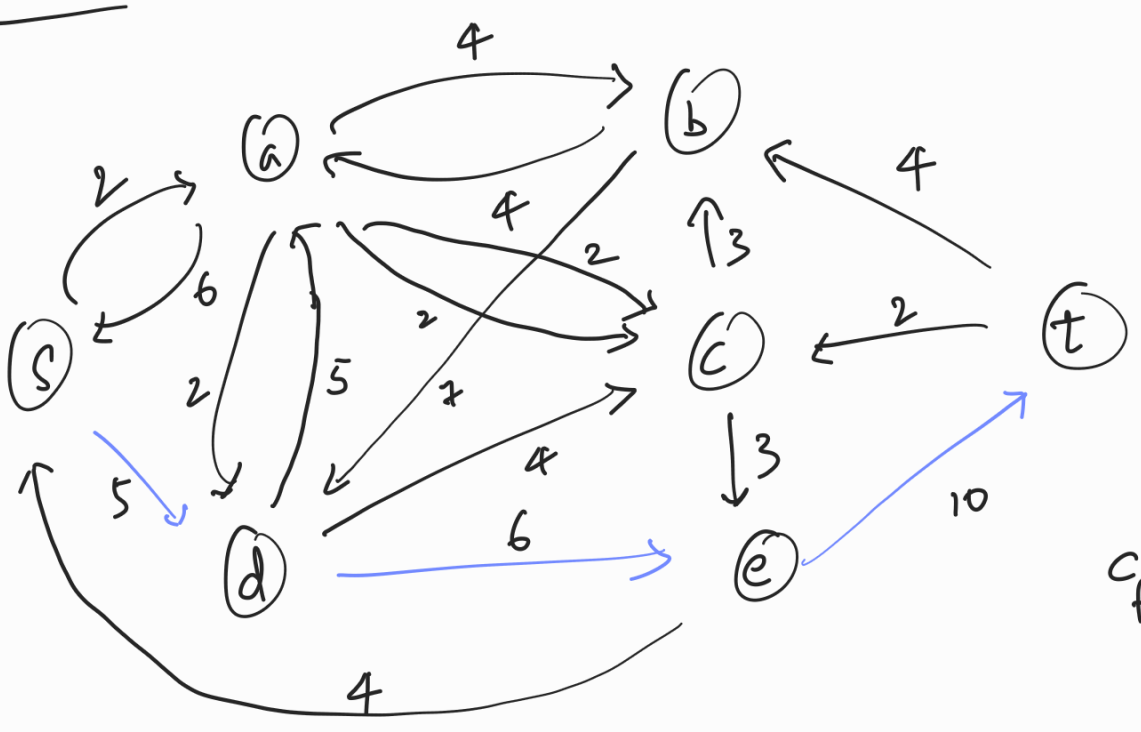


Iteration (2) :-



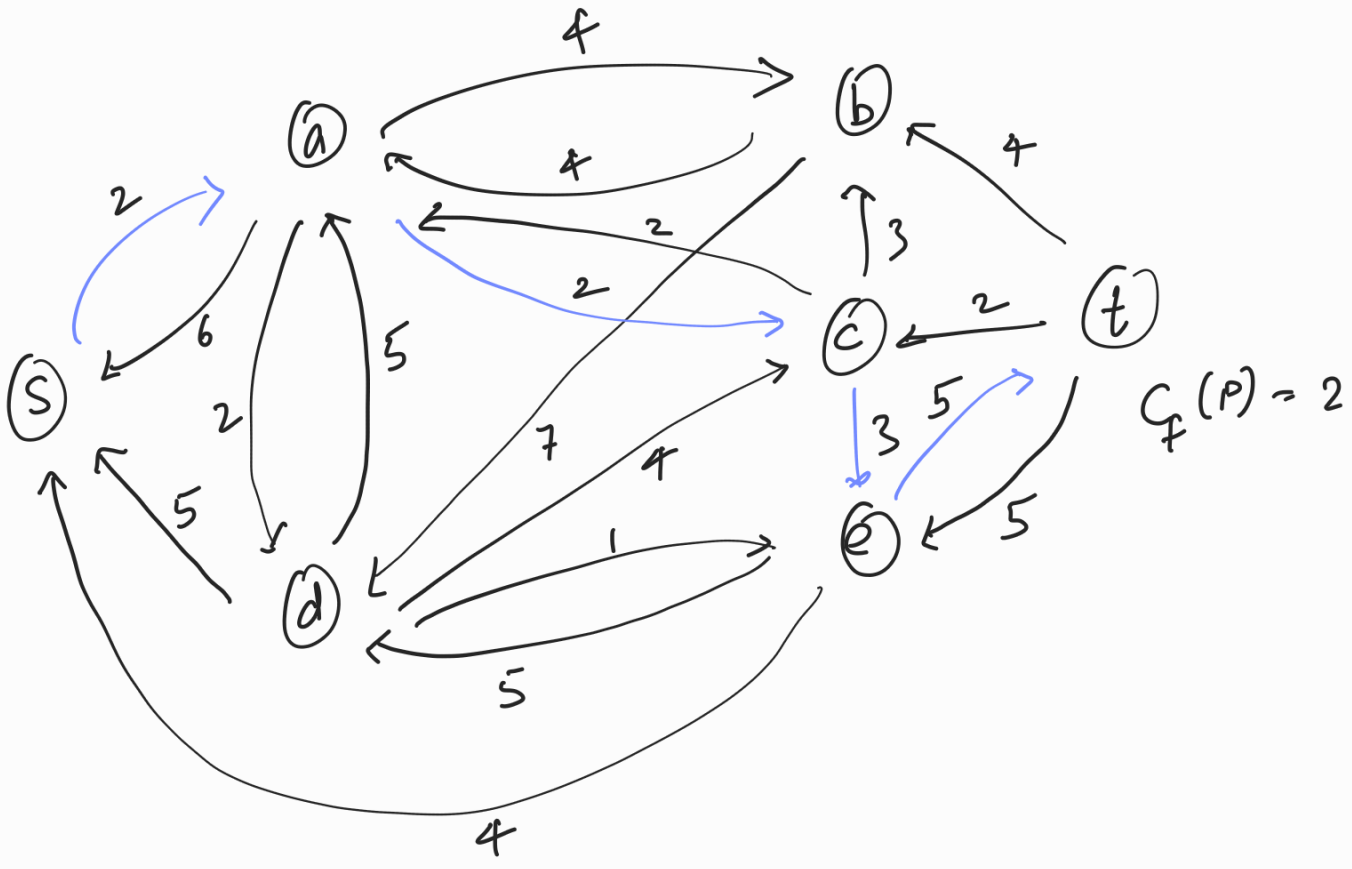
$C_f(P) = 2$

Iteration (3) :-

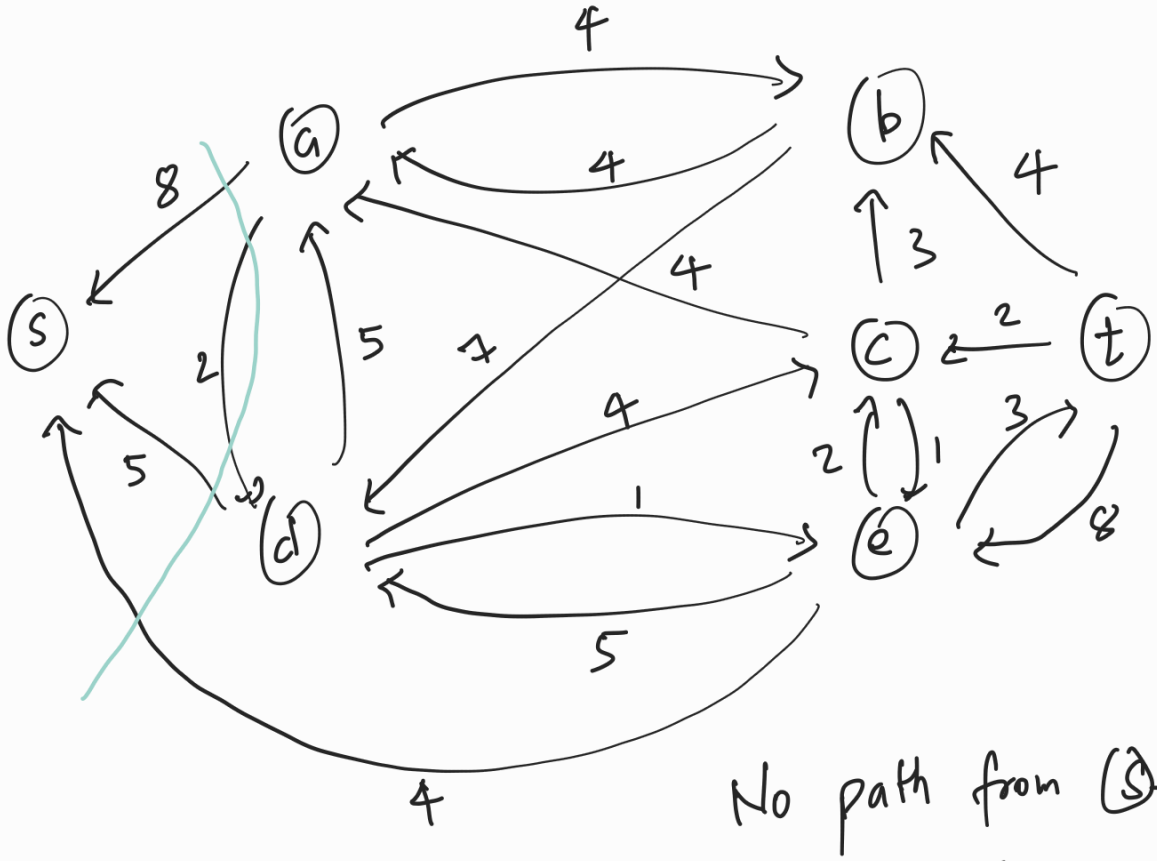


$C_f(P) = 5$

Iteration (4) :-



Iteration (5) :-



No path from (s) → (t)

Since all the edges are in-coming to s

(5) Create a network (V, E)

$V = \{ S, C_1, C_2, C_3 \ldots C_n, bs_1, bs_2, bs_3 \ldots bs_n, t \}$

with s → source
c → clients
bs → base stations
t → sink

Edges are from source to clients with capacity 1, then from clients to base station with many-1 mapping and satisfying the condition of distance less than or equal to r with capacity 1 and also edges from base stations to the sink with capacities L_1, L_2, L_3 and so on.

Find max of integer flow (f) using Ford-Fulkerson algorithm.

We produce a valid flow (f) by setting

if client connected to source:

$f(s, c_i) = 1$

if client connected to base station:

$f(c_i, bs_j) = 1$

$f(bs_j, t)$ = no. of clients connected to base_station

For all base stations and clients, capacity constraints and conservations are obtained easily.

We should only produce connection by connecting clients to base stations if and only if $f(c_i, bs_j) = 1$ in any flow.

By considering the constraints, we can say that one client cannot be connected to more than one base station and each base station cannot have connections more than L_j, it's load.

Hence,

f:- maximum number of clients that can be connected.