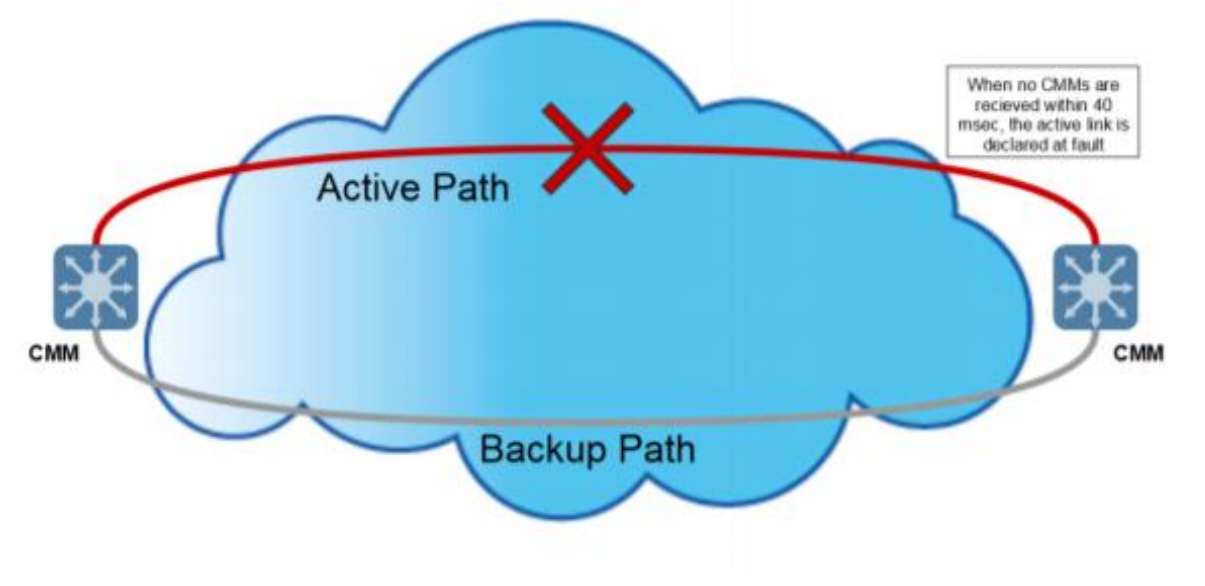
12. RESILIENT ROUTING OF CIRCUITS IN NETWORKS

Consider a connected network G(V,E) in which every edge $e \in E$ has an associated cost K(e). Given a set of source-destination pairs (s_i,d_i), for every pair (s_i,d_i), find a pair of edge-disjoint paths connecting s_i to d_i, such that the total path cost (the cost of a path is the sum of its edge costs) is minimized. Observe that one path (the primary) is used to route the connection, the other (secondary) is used as backup path. (used only in case of failure of the main path). Reconsider the problem by adding the following constraint: at most h primary paths can share the same link.



Questions:

- How much the parameters influence the solution?

Inputs:

n: the number of nodes	1*N array
e_num: the number of edges	1*N array
V: set of nodes	
E: set of edges	
$G(V,E)$: undirected network	N*N matrix
e: a edge belong in E	
$c(e)$: the cost of each edge	1*N array
$cons[e]$: the counter of using times of each link	N*N matrix

Outputs:

A graph showing the two shortest paths in different color.

Primary path(N,N) : the shortest path of nodes	N*N matrix
Secondary path(N,N):the backup path which is edge-disjoint from primary path	N*N matrix

sum_cost: the total cost of a path

ILP Model:

s: source node of the path

d: destination node of the path

i,j: variables referring to edges

x_{ij} and y_{ij}: key decision variables, when $x_{ij} = 1$, it means the link V_iV_j belongs to primary path. When $y_{ij} = 1$, it means the link V_iV_j belongs to secondary path. Otherwise they're 0.

$\hbar$: the maximum number of paths which can share the same link

$$min \quad \sum_{V_iV_j \in E} c_{ij} \times x_{ij} + \sum_{V_iV_j \in E} c_{ij} \times y_{ij}$$

$s.t.$

$$\sum_{V_iV_j \in E} x_{ij} - \sum_{V_iV_j \in E} x_{ji} = \begin{cases} 1, & i = s \\ -1, & i = d \\ 0, & i \neq s,d \end{cases}$$

$$\sum_{V_iV_j \in E} y_{ij} - \sum_{V_iV_j \in E} y_{ji} = \begin{cases} 1, & i = s \\ -1, & i = d \\ 0, & i \neq s,d \end{cases}$$

$i = s$:there must be a path connected to point s
$i = d$:there must be a path to point d
$i \neq s,d$: the path can continue through the path created in constraint 1 until point d

$$\sum_{V_iV_j \in E} {x_{ij}}^{sd} + \sum_{V_iV_j \in E} {y_{ij}}^{sd} \leq 1 \quad \forall s, d$$

$$\sum_{V_iV_j \in E} {x_{ij}}^{sd} \leq \hbar \quad \forall s, d$$

Constrain:

At most h primary paths can share the same link, every time we use this link as primary path, $cons[e]$ is increased by 1 until it goes to h, this link cannot be used again, we delete this edge from $G(V, E)$.

Problem 1:

How to get the primary and secondary paths which is disjoint?

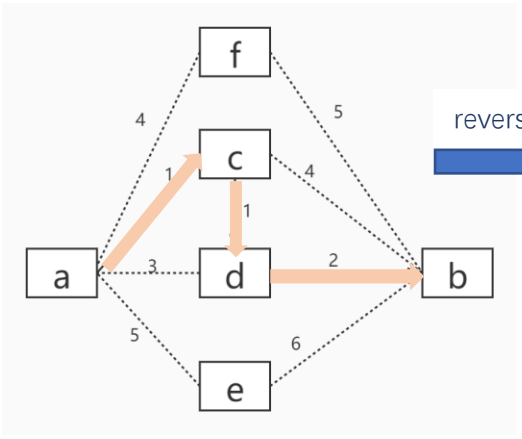
(1) Finding the shortest path $P1(= a \to c \to d \to b$ in our example) of the graph $G(V, E)$, using the Dijkstra algorithm.

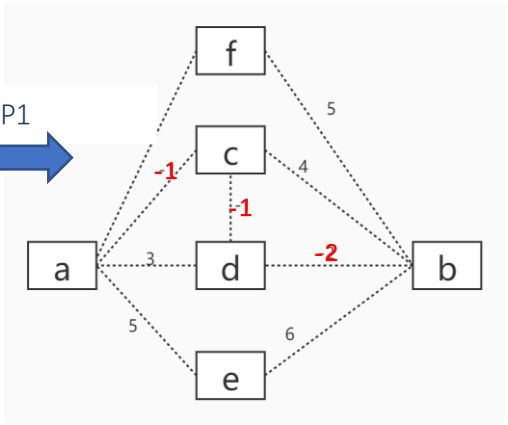
(2) Modifying the original graph $G(V, E)$ by reversing the direction of $P1$, and the costs are changed to the opposite value, this new graph is called modified graph $G'(V, E')$.

(3) Finding the shortest path $P2(a \to d \to c \to b)$ in the modified graph $G'(V, E')$ by a modification of the Dijkstra algorithm that allows its convergence in a graph with negative costs. If $P2$ doesn't exist, then stop. Otherwise, deleting the union of $P1$ and $P2$. $P1 \cap P2$ could be none, which represents they are already disjoint.

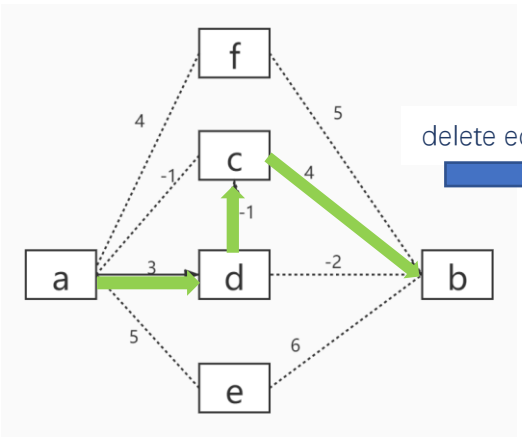
(4) Finally, the shortest pair of edge-disjoint paths is either $P1$ and $P2$, or another pair of paths made up from the combination of the nodes of $P1$ and $P2$, in our example, after deleting $c \leftrightarrow d$, $P1' = a \to c \to b, p2' = a \to d \to b$, which satisfies $P1'$ and $P2'$ are disjoint and have minimum total cost.

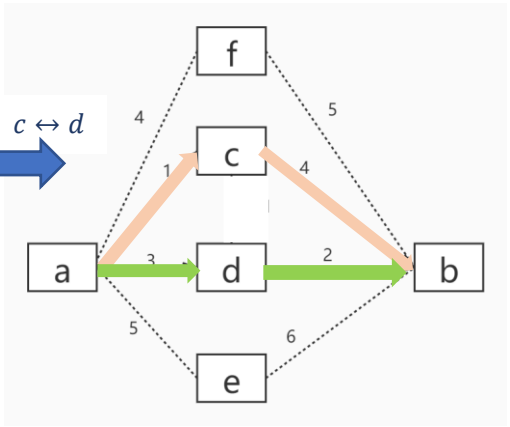
Example:





step(1) step(2)





step(3) step(4)

Problem 2: How to enforce the constraint about the maximum number of paths that can share a link?

(1)Setting integer h as the maximum number of paths that can share a link, matrix $cons[e]$ as the counter once a link was used.
(2)After finding the shortest path pair from s_i to d_i, we add the less cost one as primary path, if two costs equal, choose a random one. Then saving another as secondary path.
(3)Once a link is used as primary path, for example $a \to c$, increase the value of $cons[a,c]$ by 1 until it reaches h, deleting this link in $dis[a,c]$ by setting its cost to ∞.