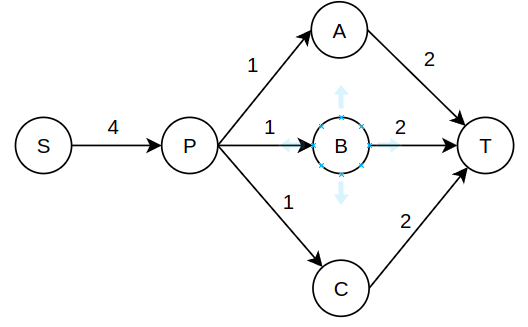
1.

(A,B) might be not still a minimum s-t cut.

It could be proved by this example:



Easily see the minimum s-t cut is (P,A), (P,B), (P,C): 1+1+1=3

But if all edges are increased by 1, then cut (P,A), (P,B), (P,C): 2+2+2=3

meanwhile (S,P): 5

The minimum s-t cut is (S,P)

2.Value of this flow: 18

It’s not the maximum (s,t) flow, the maximum flow value is 21  
minimum s-t cut is 21.

3.

Follow the Ford-Fulkerson algorithm, each iteration value of the flow will be increased by an integer multiple of 3. So, The output a.k.a the maximum s-t cut must be an integer multiple of 3.

4.

To solve this problem, First of all replace each vertex v in V by v1&v2, all input edges to v are also input edges v1, all output edges from v are also output edges v2, v1 connect to v2 via a directed edge.

Then add a node s(source) that connects (directed edge) to all dangerous nodes in D, also add a node t(sink) that is connected from (directed edge) all safe nodes in S. Assign capacity 1 to all current edges (paths), denote |T| is the capacity of all edges connect from s(source) to nodes in D group, assign capacities of the edges from vertexes in S group to t(sink) is |T|.

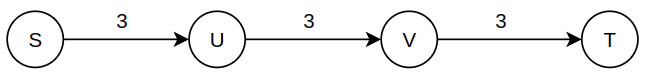
Finally use Ford-Fulkerson algorithm to find max flow in flow network. The set of evacuation routes exist if and only if value of max-flow = |T|

Since the Ford-Fulkerson algorithm runs in polynomial time, the whole algorithm runs in polynomial time as well.

5.

FALSE

Consider this simple graph G:



Easy to see there is no any edge that increasing the capacity of only 1 edge will increase the the maximum s-t flow of graph G.

6.

Step 1: Compute minimum s-t cut C (can use Ford-Fulkerson algorithm).

Step 2: iterate all edge ei (i from 1→k, k is the number of edge in cut C)

Step 3: Each ei, increase the capacity of ei by 1, then compute new minimum s-t cut Ci

Step 4: If there is a different minimum cut C’ in the graph G, there will be some ei ∈ C that is not in C’. so increasing the capacity of that ei will not change the value of C’.

Therefore If value of C = value of Ci for some i then Ci is also the minimum s-t cut. So, G has more than one distinct minimum s-t cut. In opposite, G has unique minimum s-t cut.

The Ford-Fulkerson algorithm runs in polynomial time, so the whole above algorithm also runs in polynomial time.