Assignment 6

Kala Sagar,Vinod Kumar

14466,14826

1 Question 1

Part-1

Algorithm

Let the vertices $c_1, c_2, c_3, c_4.....c_n$ represent the n different conditions of the ozone layer and vertices $b_1, b_2, b_3, ...b_m$ represent the m different balloons.

Consider a network with:-

1. s as the source and t as the terminal
2. Each of the vertices $b_1, b_2, b_3, ...b_m$ is connected to source with an edge of capacity two and directed from sink to these vertices.
3. Each of the vertices $c_1, c_2, c_3, c_4.....c_n$ is connected to the terminal with an edge with minimum flow of k and is directed from the vertices to the terminal.
4. For $1 \leq i \leq m$ if S_i represents the set of vertices representing the conditions that can be measured by a balloon b_i,then c_j is connected to b_i with an edge of capacity 1 and directed from b_i to c_j $\forall j$ such that $c_j \in S_i$.

If a maximum flow for the above network exists,then there exists a solution satisfying all the conditions given in the part 1 of the question.

How does our Maximum Flow satisfy the conditions of the Original Problem?

1. The edges connecting b_i and various $c_j's$ represent the conditions measured by a balloon b_i.As discussed above only those conditions which can be measured by b_i are connected to it.If such an edge has a flow value of 1 then the balloon at one edge measures the condition at the other end of the edge.Also the condition that the balloon can measure atmost two conditions is fulfilled by limiting the flow into the balloon from the source as two(i.e. the capacity of that edge is 2).
2. And also the candition that single balloon should not measure the same condition twice, is satisfied by giving capacity to edges connecting connecting b_i and $c_j's$ as 1.
3. Also as a condition needs to be measured atleast k times, we make the lower bound of the flow from a condition c_j to the sink as k.This is done similar to the flow with lower bound problem discussed in class.

This is a typical example with three balloons and three conditions.Note that 'k' is a lower limit while others are upper limits.

Part-2

Algorithm

Let T_1, T_2, T_3 be the three possible categories of the balloons.Split each condition c_i in to three conditions c_{ij} where $j \in \{1, 2, 3\}$ such that only balloons in category T_j and belonging to S_i can satisfy the condition c_{ij}.

To solve this problem consider a network satisfying the first three properties described in the solution of part-1,and also satisfying the following conditions:-

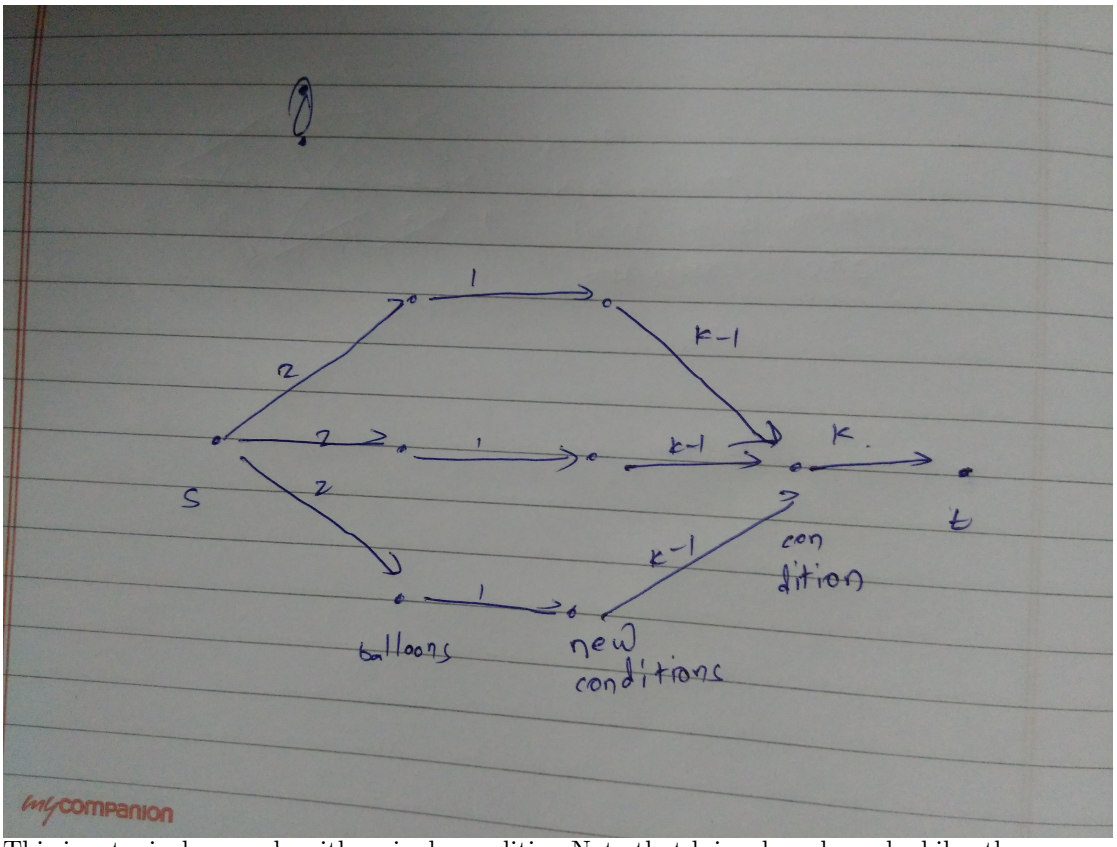
1. For all the vertices $\{b_1, b_2, ...b_m\}$ and the vertices $\{c_{11}, c_{12}, c_{13}, c_{21}, ...c_{n3}\}$ if any balloon b_i with $1 \leq i \leq m$ can satisfy a condition c_{jk} then they are connected with an edge with capacity 1 and directed from b_i to c_{jk} with $1 \leq j \leq n$ & $1 \leq k \leq 3$.
2. For any$i$ with $1 \leq i \leq n$ c_i is connected to c_{i1}, c_{i2}, c_{i3} with one edge from each of c_{i1}, c_{i2}, c_{i3} to c_i directed towards c_i with the capacity of each edge being $k - 1$.

If a maximum flow for the above network exists,then there exists a solution satisfying all the conditions given in the part 2 of the question.

How does our Maximum Flow satisfy the conditions of the Original Problem?

Our network retains some of the structure from the network of the previous part which satisfy the initial conditions.The changes are made to satisfy the new condition given in this part of the problem.

1. The new constraint that each condition need to be measured by balloons from atleast two different categories is fulfilled by adding three new conditions for each condition one for each group.A new condition can be measured by a balloon only if it can measure the parent of the new condition and also belong to the group represented by the new condition.Hence the flow into a new condition is the number of balloons of the group represented by the condition that can measure the original condition.
2. The three new conditions are connected to their parent condition.hence the flow into the parent condition is the number of balloons that measure the parent condition.To ensure that the flow comes from atleast two edges we limit the flow from each of edges from new condition to the parent condition at k-1.Now as the flow out of the parent condition should be greater than k(as said in previous problem),a flow is possible only of there is a inflow from atleast two edges into the parent condition.



This is a typical example with a single condition.Note that k is a lowe bound while others are upper bound.

2.2 Social Networking Problem :

1 Idea :

We need to find if there exists a set of vertices X such that $|E(X)| \geq 10 \times |X|$. Let us assume $q(X)$ as

$$|E(X)| - 10 \times |X|$$

It implies that if there is no friend-group in G then for any subset A of V, q(A) is always negative.
$\implies$ If the maximum value of q(A) is non-negative then there exists a subset A of V for which q(A) is positive
$\implies$ there exists a friend-group in G.

2 Constructing a network flow :

Let edges present in E be represented as $e_1, e_2, ..., e_m$ and vertices of V be represented as $v_1, v_2, ..., v_n$. Also consider to a source(s) and terminal(t). Consider a Network $G^{'}$ with all of these as vertices. Now in network $G^{'}$

1. Connect each of e_i to source(s) with forward edge from source and capacity(or profit(p_i)) of each edge connecting e_i as 1.
2. Connect each of v_i with terminal(t) by a forward edge from v_i to with capacit(or cost(c_i)) =10 for every such edge.
3. Form each e_k draw forward edges to each of vertices v_i, v_j if $e_k = (v_i, v_j)$ *in* $G = (E, V)$, directed from e_k each with capacities as ∞.

Now this is similar to open-pit mining problem discussed in class.

3 Maximisation of Function $q(A)$:

Note: We use concept of valid set and profit, cost similar to open-pit mining directly from here-on.

For any valid subset A of network with $X = v_i : v_i \in A$

$$\begin{aligned} q(A) &= |E(X)| - 10 \times |X| \\ &= \sum_{i|e_i \in A} p_i - \sum_{i|v_i \in A} c_i \\ &= P_{total} - \{ \sum_{i|e_i \in \overline{A}} p_i + \sum_{i|v_i \in A} c_i \} \\ &= m - \{ \sum_{i|e_i \in \overline{A}} p_i + \sum_{i|v_i \in A} c_i \} \end{aligned} \tag{1}$$

We can consider another function $q^{'}(A)$ as

$$\begin{aligned} q^{'}(A) &= \sum_{i|e_i \in \overline{A}} p_i + \sum_{i|v_i \in A} c_i \\ \implies q(A) &= m - q^{'}(A) \end{aligned} \tag{2}$$

Hence maximisation of $q(A)$ is same as minimisation of $q^{'}(A)$.

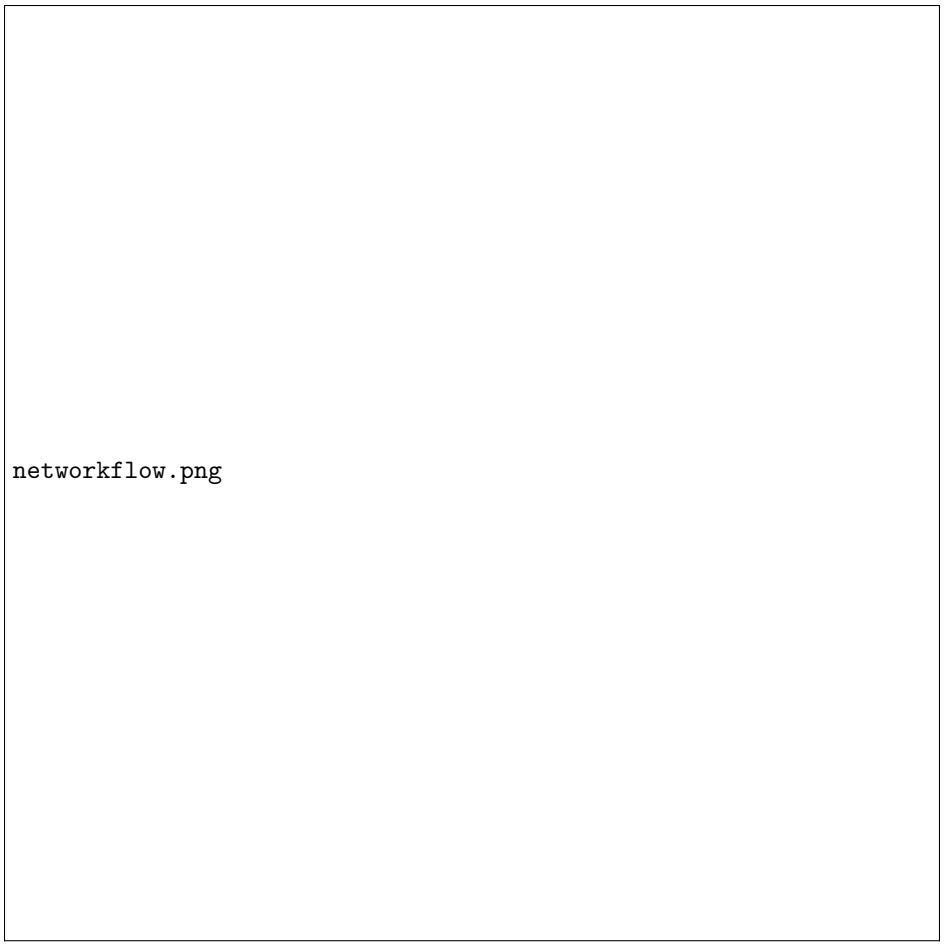


Figure 1: Flow Netwok - $G^{'}$

4 Minimisation of $q^{'}(A)$:

Concept of validity :

Similar to open-pit mining, a subset A is said to be valid only if there are no black edges leaving from it. This concept of validity makes sure that given a set A, set X the edges present inside it are only $E(X) = E \cap (X \times X)$.

Expressing $q^{'}(A)$ in-terms of $Cut(A, \overline{A})$:

$Cut(A, \overline{A}) = \{(x, y) \mid x \in A, y \in \overline{A}$ *or* $x \in \overline{A}, y \in A\}$
$\implies$ *capacity of* $Cut(A, \overline{A}) = \sum c(u, v)$

In capacity of a cut, only capacity edges directed from A to $\overline{A}$ are considered. And since for a valid cut there are no black edges coming out of A,

capacity of $Cut(A, \overline{A})$
$= \sum$ *capacity of blue edges coming out from*$A + \sum$*capacity of red edges coming out from* A

$\implies$ For any valid cut

$$\begin{aligned} Cut - capacity(A, \overline{A}) &= \sum_{i|e_i \in \overline{A}} p_i + \sum_{i|v_i \in A} c_i \\ &= q^{'}(A) \end{aligned} \tag{3}$$

Relationship of *min* of $q^{'}(A)$ and min-Cut of $G^{'}$:

Any cut in $G^{'}$ can be seen as either a valid cut or invalid cut according to our definition of validity. But for any invalid cut there exists at-least one black-edge. And since, capcity of any black edge in network is ∞.
$\implies$ *capacity of any invalid cut* $\rightarrow \infty$

And for every valid cut capacity is finite and is equal to $q^{'}(A)$

$$\implies minimum\ of\ Cut - Capacities\ =\ minimum\ of\ q'(A)$$

5 Concluding if ***friend-group*** is *present or not* :

And we know from theorem discussed in calss that

For any network with source S and sink t the maximum value of s-t flow is equal to capacity of min-Cut.

And we can find min-cut capacity in polynomial time using concepts of max-flow. And hence we will get min-value of $q^{'}(A)$ in polynomial-time.

From min of $q^{'}(A)$ we can get the maximum possible value of$q(A)$ by using the above relations.

If the maximum value of $q(A)$ is non-negative
$\implies$ there is a friend-group in G.

Else there is no such friend-group in G.