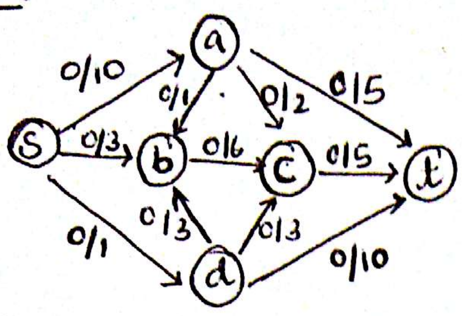
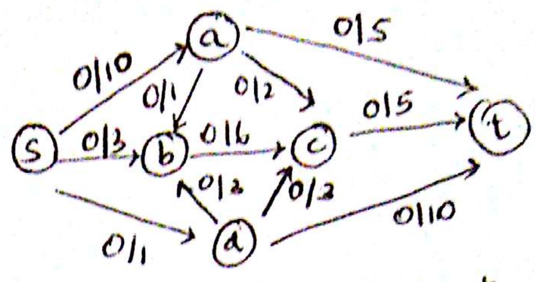
A1.



G_1

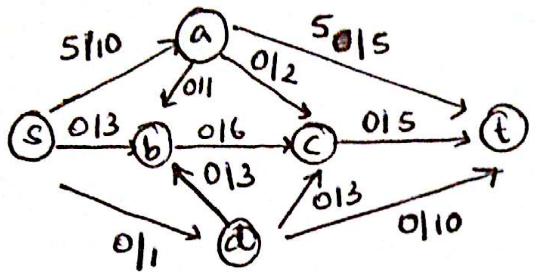
Reverse Graph GF_1



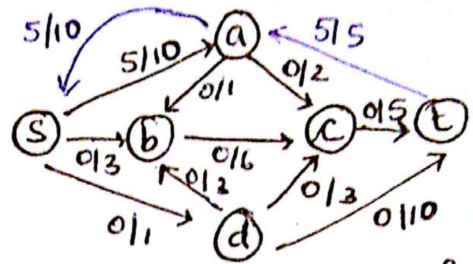
Simple s-t path = s-a-t

C(P) = 5

Augmented graph G_2 :



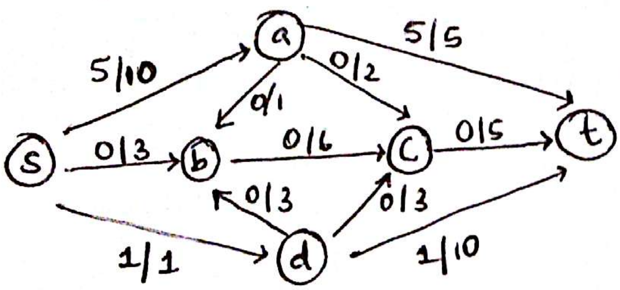
Reverse Graph GF_2:



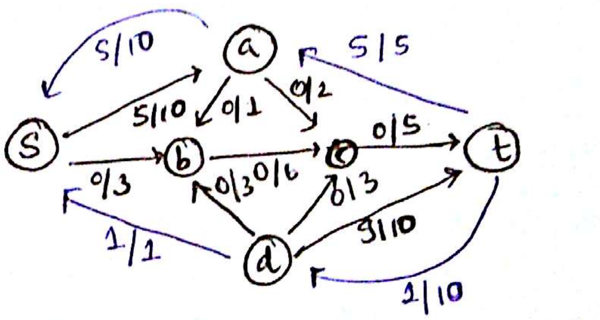
Simple s-t path = s-d-t

C(P) = 1

Augmented graph G_3 :



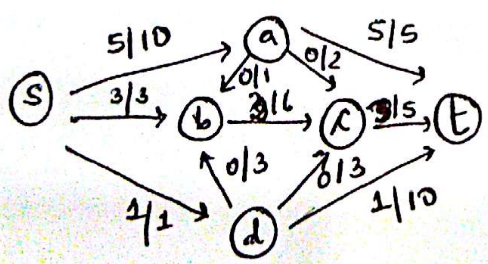
Reverse graph GF_3 :



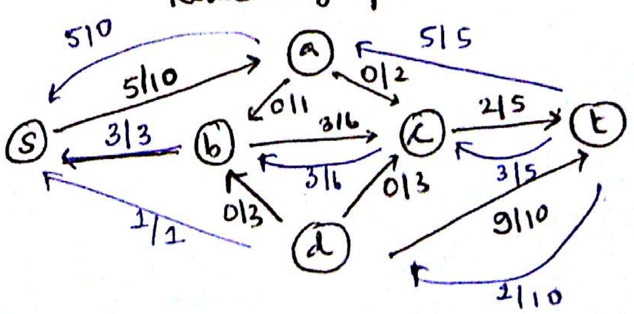
Simple s-t path = s-b-c-t

C(P) = 3

Augmented graph G_4



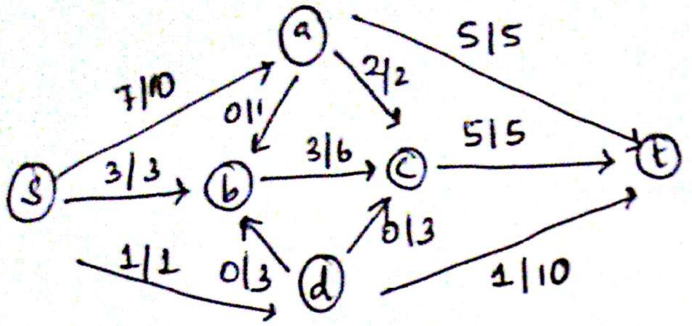
Reverse graph GF_4.



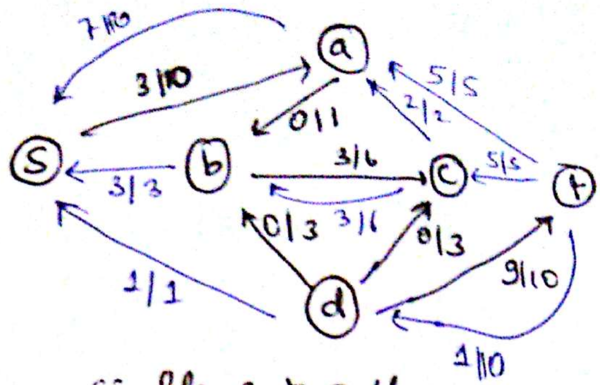
Simple s-t path = s-a-c-t

C(P) = 2

Augmented graph G_5



Reverse graph G_{F_5}:



Simple s-t path: Not possible

Hence, Maximum flow =

$$\sum_{e \text{ out of } s} f(e) = 7+3+1 = 11$$

Minimum cut:

$S = \{s, a, b, c, d\}$, $T = \{t\}$

$c(S,T) = 11$

CSOR:4246 Assignment 3

Vibhuti Mahajan (UNI: vm2486)

16 October 2016

1 Solution

Please check the scanned documents.

2 Solution

A) There are multiple sources and multiple sinks, and we wish to maximize the flow between all sources and sinks.

(a) Let's consider a reduction G'=R(G). We construct the flow network G' as follows:

- Let $S = s_1, s_2, ..., s_p$ where s_i is the source node for $1 \leq i \leq p$ and $T = t_1, t_2, ..., t_q$ where t_j is a sink node for $1 \leq j \leq q$
- Introduce a source node s and a sink node t.
- X=edge(u,v); where u=s and v$\in S$. Y=edge(u,v); where u$\in T$ and v=t.
- $c_e(s) = \Sigma_{eoutofS} c_e$; $c_e(t) = \Sigma_{einofT} c_e$
- Hence, the flow network G'=$([V \cup \{s,t\}], [E \cup \{X.Y\}], s, t)$

(b) Reduction R is efficient (polynomial in the size of G). We can prove this byu showing that the expansion of the reduced graph is polynomial in order: Finite number of nodes(2) are added to G. Thus $|V'| = |V| + 2 = \mathcal{O}(n)$ Number of edges added $= \Omega(n)$. Hence Total edges of the reduction $= |E| + \Omega(n) = \mathcal{O}(m+n)$

(c) Maximum flow in G is equivalent to maximum flow in G'

- Given a maximum flow $|f|$ in G, we can construct a maximum flow $|f'|$ in G' , such that $|f| = |f'|$.
  - For all $s_i \in S$, send a flow from s, such that $f'^{in}(s_i) = f^{out}(s_i)$ and for all $t_j \in T$, send a flow from t_j to t, such that $f'^{out}(t_i) = f^{in}(t_i)$.
  - Consider a an s-t cut for G'= $(\{s \cup V\}\{t\})$. The amount of flow passing between the 2 sets is $\|f\|$ by construction. Hence, $|f| \leq |f'|$
  - Now if we consider that $|f| < |f'|$, then there exists atleast 1 simple s-t path, P, the capacities whose edges haven't been saturated yet. Such a path is of the form: $P = \{s, s_p\}, e_1, e_2, ..., e_n, \{t_p, t\}$, where $s_p \in S$, $t_p \in T$ and $e_1, ..., e_n \in E$ that carries the flow, say $f_p > 0$, along all its edges.
    $\rightarrow$ there exists a simple path from $s_p to t_p$ that hasn't been saturated yet. But this is not possible because G is already in its max-flow state with flow value $|f|$
  - Thus by contradiction, the only case that is possible is that $|f| = |f'|$
- If $|f'|$ is the maximum flow of the reduced graph then $|f'|$ is the max-flow for the original graph G.
  - Consider a min s-t cut for graph G', such that the s and set S belongs to set A and t and set T belongs set B.
  - The flow between A and B is $\|f'\|$ which is the max flow.

- We modify the span of the 2 sets as: $A \leftarrow A - \{s\}$ and $B \leftarrow B - \{t\}$. The flow value between A and B is still $\|f'\|$
- Also, all the edges between A and B are saturated and no more flow is possible from A to B.
- Thus, $\|f'\|$ is the max flow for graph G.

B) Both the edges and the vertices (except for s and t) have capacities. The flow into and out of a vertex cannot exceed the capacity of the vertex.

(a) Consider the reduction G'=R(G)

- I'll assume the symbolic representation of the graph: G(V,E,s,t). Capacity of edge $e \in E = c_e$ and capacity of node $u \in V = c_u$.
- For each node u other than s and t, replace with 2 nodes: u_{in} and u_{out}.
- There is an edge e going from u_{in} to u_{out} with $c_e = c_u$.
- For all edges $(v, u) \in G(V, E, s, t)$ replace with edges (v, u_{in}). For all edges $(u, w) \in G(V, E, s, t)$ replace with edges (u_{out}, w)
- We can now run the max-flow algorithm on the reduced graph G' which give us the maximum flow in the original graph G

(b) Reduction R is efficient. For the reduced graph, number of nodes= $|V| + \mathcal{O}(n) = \mathcal{O}(n)$ and number of edges added $n - 2 + \mathcal{O}(m) = \mathcal{O}(m + n)$.

(c) max-flow for graph G is equivalent to that of graph G'.

- Let G have the maximum flow $|f|$ then G' also has a max-flow $|f|$
  - If we assume that max-flow for the reduced graph is not the same as for G, then there has to exist a disjoint s-t path in G', that hasn't been saturated yet.
  - This possible only if the new edges introduced form an s-t path because rest of the edges have already been saturated.
  - This is not possible because there is no s-t path in G' exclusively made of new introduced edges.
  - Hence the max-flow for G' is also $|f|$
- If $|f'|$ is the maximum flow for G', then there exists a feasible maximum-flow for G whose value is $|f'|$.
  - For the reduced graph G', the flow passing through a node satisfy the flow conservation and capacity constraints.
  - This implies that flow constraints and flow conservation is observed for the original graph G as well because of the construction.
  - No extra flow can pass between s and t in the original graph G, because then it will falsify the capacity constraints of the nodes. Also, if extra flow can pass then flow conservation cannot hold for atleast 1 node.

3 Solution

1. Given constraint $= f^{in}(v) - f^{out}(v) = d(v)$
   Taking the sum over all nodes v of the network: $\Sigma_{v \in V}(f^{in}(v) - f^{out}(v) = d(v))$
   LHS: $\Sigma_v(f^{in}(v) - f^{out}(v)) = 0$; for a feasible flow there cannot be an imbalance for the net flow over the entire network
   $\rightarrow \Sigma_v d(v) = \Sigma_{d(v) \leq 0} d(v) + \Sigma_{d(v) \geq 0} d(v) = 0$

2.
   - We can reduce the problem by introducing a source 's' and a sink 't'
   - We join an edge from 's' to all nodes having negative demand with capacity of each of this edges is infinitely high. Also, we join an edge from all nodes having positive demand to the sink with each of these edges having infinite capacty.

- We can now run max-flow algorithm on this reduced graph to get a feasible circulation (if it exists) in the original network.
- This is an efficient reduction because number of nodes = $|V| + 2 = \mathcal{O}(n)$ and number of edges = $|E| + \Omega(n) = \mathcal{O}(m + n)$
- If there exists a feasible flow for the flow-network G then we can get a feasible flow of value $|f|$ in G': Just saturate the edges $(s, u), u \in V$ and $(v, t), v \in V$. This results in a feasible flow in the reduced graph G' with flow value $|f|$ (this can be attained from an s-t of the type $s \cup V, t$)
- A flow of value $|f'|$ on G' results in a feasible flow for G: Edges of the type $(s, u), u \in V$ and $(v, t), v \in V$ are saturated in G'. Remove these edges and then the original network follows the demands constraints resulting in a feasible flow.

4 Solution

1. A max-flow problem, G(V, E, s, t, c) can be formulated as min-cost problem as follows.
   - We assume the supply of all the nodes to be 0.
   - Thus to maintain the flow conservation constraints, we'll have to add an auxiliary edge from sink t to source s.
   - Let the maximum flow value of the original network be k. Then the flow passing from t to s must be of the value k to satisfy the constraints.
   - Our goal is to maximise the value of k in the augmented graph, or equivalently, minimise the value of -k.
   - We thus give the cost value to the edge (t,s) to be -1. Rest of the edges have cost value 0.
   - Hence, if min-cost algoriythm is run on this augmented graph it gives the max-flow value for the original graph G. Also, this augmentation is efficient since we have just added 1 extra edge and updated the cost and supply values of the nodes.
2. Formulating an LP for min-cost problem for the graph G(V,E,s,c,a)
   - The variables for this problem are values f_e: the value of flow on the edge $e \in E$.
   - The cost for flow over edge e = $f_e.s_e$
   - Hence, the objective is:

     $$\min \Sigma_{e \in E} f_e.a_e$$

     subject to

     $$f^{out}(v) - f^{in}(v) = s_v, \forall v \in V$$

     $$0 \leq f_e \leq c_e, \forall e \in E$$

A5

Variables:-

~~d_i~~ x_i = No of carpets produced in month i, $i = 1(1)12$
Here, for the sake of simplicity, $x_0 = 0$

v_i = change in the number of carpets produced from the last month. i.e $|x_i - x_{i-1}|$; $i = 1, \dots, 12$

s_i = No of carpet stored at the end of month i; $i = 1, \dots, 12$
Here also, for the sake of simplicity, $s_0 = 0$

Objective function:-

$$\min \quad 150 \sum_{i=1}^{12} v_i + 6 \sum_{i=1}^{12} s_i$$

s.t.

$$d_i \geq 0 \quad ; \quad i = 1, \dots 12$$
$$x_i \geq 0 \quad ; \quad i = 1, \dots, 12$$
$$x_0 = 0$$
$$s_0 = 0$$
$$s_i \geq 0 \quad ; \quad i = 1, \dots, 11$$
$$s_{12} = 0$$
$$s_i = s_{i-1} + x_i - d_i \quad ; \quad i = 1, \dots, 12$$
$$v_i = |x_i - x_{i-1}| \quad ; \quad i = 1, \dots, 12$$
$$v_i \geq 0 \quad \{\text{Redundant constraint}\}$$

This is an LP but we have to convert all constraints into linear functions of variables.

For the ~~variab~~ constraint:-

$$v_i = |x_i - x_{i-1}| \quad ; \text{ for a particular } i$$

Can be converted to 2 constraints (equivalent)

$$v_i \leq |x_i - x_{i-1}| \quad \& \quad v_i \geq |x_i - x_{i-1}|$$

$\Rightarrow -(x_i - x_{i-1}) \leq v_i$ and $v_i \leq x_i - x_{i-1}$

$\Rightarrow -(x_i - x_{i-1}) \geq v_i$ and $v_i \geq (x_i - x_{i-1})$

Thus, the LP is :-

$$\min_{\substack{d_i \geq 0,\, x_i \geq 0, \\ s_i \geq 0}} \quad 150 \sum_{i=1}^{12} v_i + 6 \sum_{i=1}^{12} s_i$$

Subject to :-

$$x_0 = 0$$

$$s_0 = 0$$

$$s_{12} = 0$$

$$\left.\begin{aligned} s_i &= s_{i-1} + x_i - d_i \\ v_i + x_i - x_{i-1} &\geq 0 \\ -v_i + x_i - x_{i-1} &\geq 0 \\ -v_i - x_i + x_{i-1} &\geq 0 \\ v_i - x_i + x_{i-1} &\geq 0 \end{aligned}\right\} \text{for } i \text{ ranging from 1 to 12}$$