CMPS 102 — Quarter Spring 2017 – Homework 4

VLADOI MARIAN

June 3, 2017

I have read and agree to the collaboration policy.Vladoi Marian
I want to choose homework heavy option.
Name of students I worked with: Victor Shahbazian and Mitchell Etzel

Solution to Problem 1 Node-capacitated networks :

Assume we have a directed graph G = (V, E) with a source (s) and a destination (t). The node capacity for each node node $\in V$, $c_v \geq 0$. The flow f in G through a node v is defined as $f^{in}(v) \leq c_v$.

a) The algorithm for the maximum flows in node-capacitated networks:

1. We create another Graph G' = (V', E').
2. The s $\in$ G becomes $s^{ou}t$ in G'. (Is the same node as s. It has all the edges that used to come out of s)
3. We link the node $s^{ou}t$ with all the nodes v^{in}.
4. The t $\in G$ becomes t^{in} in G'. (Is the same node as t. It has all the edges that used to come into t)
5. We link all the nodes v^{out} with the node t^{in}.
6. For all the other nodes $v \in G$, except s and t, we create two nodes $v^{in}\ and\ v^{out}$ in G'. (v^{in} is the node $\in$ G' that has all the edges that used to come into node v $\in$ G. v^{out} is the node $\in$ G' that has all the edges that used to come out of v $\in$ G).
7. For all the new edges in G' that we have created $c_e = \infty$.
8. The edge between the nodes $v^{in}\ and\ v^{out}$ has capacity $c_e = c_v\ (v \in G)$.
9. We run Ford-Fulkerson described in class with capacity scaling on G' graph.
10. At the end of the algorithm we have a max flow from $s^{out}\ to\ t^{in}$ in the G' . This is also the maximum flows in node-capacitated graph G.

To compute the max flow in node-capacitated graph G we have to compute the standard Ford - Fulkerson flow in the newly created graph G'.

b,c) Assume there is a flow of value x in the original graph G. We can use this fact to construct a flow of the same value (x) in the new graph G'. The flow that passed through node v in G, in the new created graph G' will pass through the edge between ($v^{in}\ and\ v^{out}$). We know that this edge has capacity == $c_v, v \in G$. In the same time , if there is a flow of value (x) in G' it must be a flow of the same value (x) in G that respects all the constraints stated at the begining of the problem. The flow in G' was obtained by using the edges of the form (v^{in}, v^{out}).

We consider the Max-Flow Min-Cut theorem for node capacitated Graph G. We apply the same theorem on the modified graph G'. Assume we have any cut (A,B) ($s^{out} \in A\ and\ t^{in}B$). The theorem states that the cut (A,B) has to be a minimum cut. The theorem also states that the max flow f = minimum capacity of the cut. This theorem was proved in class. Because we assigned capacity to be infinity to all the edges that come into v^{in}, and all the edges that come out v^{out}, we know that a min cut (A,B) will have only the edges between ($v^{in}\ and\ v^{out}$). Chosing any other edge that is not of the form (v^{in}, v^{out}) will not create a min cut. Cutting all this edges will separate the source node and the sink node . In the original graph G , this cut can be interpreted as creating a set T of all this nodes v that belong to the min cut (A,B) $\in$ G' , and deleting all the nodes in T. The capacity of the set T is the sum of all nodes capacities $\in$ T. We are asked to define our "capacity" object . All the previous statements implies that the value of maximal flow in a node capacitated network is equal to the minimum capacity of a set T that disconect the source node and the sink node. The correcteness of the Ford-Fulkerson was proved in class.

Because we use the Ford-Fulkerson with capacity scaling, the running time of our algorithm is O ($m^2 logC$) , where m is the number of edges in G' ,and C is the sum of all capacities edges that come out of the source node.