CS 4820, Spring 2017 Homework 5, Problem 3

Name: Yuxiang Peng

NetID: yp344

Collaborators: jl3455, zl542

3. The algorithm is as follows:

Use DFS to find a simple positive s-t path and decrease the flow of this path by the minimum edge along this path. Add the path vector to the decomposition. The coefficient of the path is just the minimum edge we deduct. Continue until there is no longer a simple positive s-t path in the graph.

When we finish the previous step, we check, if there is no flow in the graph, the algorithm terminates. If there are some cycles in the graph, decrease the flow in the cycle by a value of the minimum edge along the cycle. Add the cycle path vector to the decomposition and the coefficient is the minimum edge. Continue until there is no cycle in the graph. Return the path vectors and coefficients.

Correctness

Nodes(except source and sink) in the graph still maintains conservation of flow in a minimum edge deduction each time, i.e. flow in is equal to flow out. At the same time, the capacity of each edge will not be exceeded in the process. So for the graph, it still keeps a flow after each deduction. In addition, each time, one edge in the graph is thrown away, so the algorithm will always terminate in at most m steps. After iterating all $s - t$ paths, because it preserves flow identity, we have a flow of total value zero, so there is either no flow at all in the graph or only cycles.

Running time:

The worst case for algorithm termination is m steps. DFS requires for O(m+n) each time. So the total cost is O((m+n)*m)=O(m^2)