ACN 915: Algorithms for networked information
Homework 1

December 5, 2016

Exercise 1

We consider the Ford and Fulkerson's augmenting path algorithm studied in class for flow networks with integer capacities. As explained in class, the poor behavior of the algorithm can be blamed on poor choices for the augmenting path. In this exercise, we consider the following rule: choose the augmenting path with the smallest number of edges.

1. How much time is required to find such a path?

Let f_i be the current flow after i augmentation steps, let G_i be the corresponding residual graph. In particular, f_0 is zero and $G_0 = G$. For each vertex v, let $level_i(v)$ denote the unweighted shortest path distance from s to v in G_i. We want to prove that $level_{i+1}(v) \geq level_i(v)$ for all vertices v and integers i.

2. Prove the claim for $v = s$ or if $level_{i+1}(v) = \infty$.

Choose now a vertex $v \neq s$ and let $s \to \cdots \to u \to v$ be a shortest path in G_{i+1} so that $level_{i+1}(v) = level_{i+1}(u)+1$.

3. In the case where $u \to v$ is an edge of G_i, show that $level_i(v) \leq level_i(u) + 1$.
4. If $u \to v$ is not an edge in G_i, show that $level_i(v) = level_i(u) - 1$.
5. Conclude that $level_{i+1}(v) \geq level_i(v)$ for all vertices v and integers i.

Whenever we augment the flow, the bottelneck edge in the augmenting path disappears from the residual graph, and some other edge in the reversal of the augmenting path may (re-)appear (see example seen in class). We will now prove that during the execution of the Edmonds-Karp algorithm where the augmenting path with the smallest number of edges is selected, any edge $u \to v$ disappears from the residual graph G_f at most $V/2$ times.

Suppose $u \to v$ is in two resildual graphs G_i and G_{j+1}, but not in any of the intermediate residual graphs $G_{i+1}, \ldots, G_j$, for some $i < j$.

6. Show that $level_i(v) = level_i(u) + 1$ and $level_j(v) = level_j(u) - 1$.
7. Show that $level_j(u) - level_i(u) \geq 2$ and conclude for the number of disappearances of a given edge.
8. Give an upper bound on the number of iterations and the overall time complexity of the algorithm.

Exercise 2

We consider the following algorithm for finding dense subgraphs:

Data: a graph G with n vertices and a subset of vertices $X \subset V$
Result: a dense subgraph containing the vertices in X
Let $G_n \leftarrow G$;
for $k = n$ *downto* $|X| + 1$ **do**
    Let $v \notin X$ be the lowest degree node in $G_k \setminus X$;
    Let $G_{k-1} \leftarrow G_k \setminus \{v\}$.
end
Output the densest subgraph among $G_n, \ldots, G_{|X|}$.

We now prove that this algorithm is a 1/2-approximation. Let S be the densest subgraph containing X. If our algorithm outputs S, then it is clearly optimal. We now assume that at some point our algorithm has deleted a node $v \in S$. Let G_k be the graph right before the first $v \in S$ was removed.

1. Show that
$$\frac{|e(S)|}{|S|} \geq \frac{|e(S)| - d_S(v)}{|S| - 1}.$$
2. Show that $d_{G_k}(v) \geq d_S(v) \geq \frac{|e(S)|}{|S|}$.
3. Show that
$$\frac{|e(G_k)|}{|G_k|} \geq \frac{\sum_{u \in S} d_S(u) + \sum_{u \in G_k \setminus S} \frac{|e(S)|}{|S|}}{2|G_k|}.$$
4. Deduce that
$$\frac{|e(G_k)|}{|G_k|} \geq \frac{|e(S)|}{2|S|},$$
and conclude.