Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn July 24, 2014

Network Algorithms, Summer Term 2014
Problem Set 6

1 Distributed Network Partitioning

In this exercise, we will derive an asynchronous distributed version of the cluster construction algorithm presented in the lecture.

Assume that (in $O(n)$ time and using $O(m + n \log n)$ messages) a spanning tree has already been computed. You can further assume that the constant ρ in the algorithm is 2. Moreover, you can ignore the intercluster edges in this exercise, i.e., the number of intercluster edges does not have to be reduced once the clusters are built.

Just like the centralized algorithm, the distributed algorithm repeatedly applies the following two steps to construct a partitioning:

1. Find a (cluster) leader.
2. Construct a cluster C, remove all the nodes $v \in C$ and remove all the edges $\{u, v\}$ for which $u \in C$ or $v \in C$ (or both).

Let us now construct the clusters:

a) We need a first leader to start the algorithm. Describe how a leader can be determined in $O(n)$ time on the spanning tree using $O(n)$ messages!

b) Given a leader, we need to build the cluster by adding more and more nodes to the cluster. Describe how the leader constructs the cluster! Let C denote the constructed cluster. Show that the time complexity to construct the cluster is bounded by $O(|C|)$!

   Hint: If the radius of C is r, show that the time complexity is $O(r^2) \subseteq O(\log^2 |C|) \subset O(|C|)$!

c) Let $E' \subset E$ be the set of edges that can be removed in Step 2. Show that the construction of cluster C requires $O(|E'|)$ messages!

   Hint: Use the observation that edges closer to the leader have to be traversed more often, but there are more edges with a greater distance to the leader!

d) Once a cluster has been constructed, we need to find the next leader in the remaining graph. Show how each node in the graph can be visited in $O(n)$ time using $O(n)$ messages! How can this tree-traversal scheme be used to find the cluster leaders every time a new cluster has been built?

e) Putting everything together, show that the entire partitioning requires $O(n)$ time and uses $O(m)$ messages![1]

[1]The construction phase of the spanning tree is (clearly) not considered in these bounds.

And here are some interesting bonus questions:

f*) Using the centralized partitioning algorithm, compute a clustering and simultaneously a coloring of the cluster graph! (The cluster graph is the graph $G_C = (V_C, E_C)$ such that the clusters are the nodes V_C and there is an edge between two nodes $C, C' \in V_C$ iff there exists an edge between the two clusters in the original graph G.)
What running time does your algorithm have?

Hint: In the cluster construction, clusters evolve in a BFS-manner, shell by shell is added until the cluster stops growing. What can be said about the clusters before the last 'shell' is added? For this problem compute your clustering in a slightly different way.

g*) Use the above cluster graph coloring to calculate a $(\Delta + 1)$-coloring in G in $O(\log^2 n)$ rounds!

Hint: Assume arbitrary message sizes and think of the chapter *Lower Bounds*!

h*) Use $g*$) to construct an MIS in $O(\log^2 n)$ rounds!