Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn

June 13, 2013

Network Algorithms, Summer Term 2013
Problem Set 6

Exercise 1: Greedy Dominating Set

The distributed version of the greedy dominating set (DS) algorithm presented in the lecture computes a $\ln \Delta$-approximation in $O(n)$ rounds.

Construct a graph $G = (V, E)$ for which the approximation ratio is as large as possible, i.e., the size of the computed DS is a factor $\Omega(\log \Delta)$ larger than the optimal DS! Try to find a graph for which Δ is as large as possible!

Hint: It could make things easier if you try to find G by aiming for an optimal DS of constant size, e.g., 3 nodes.

Exercise 2: Fast Dominating Set

The second algorithm discussed in the lecture only needs $O(\log^2 \Delta \log n)$ rounds to compute an $O(\log \Delta)$-approximation. More precisely, the algorithm requires $O(\log^2 \Delta \log n)$ phases, where each phase (i.e., one iteration of the while loop) consists of a constant number of rounds. Write down the communication steps (node v sends/receives ... to/from ...) for a single phase in detail! How many rounds are required exactly in each phase?

Exercise 3: Vertex Cover Approximation

A related (but simpler) problem is the problem of computing a small vertex cover. A vertex cover of a graph $G = (V, E)$ is a subset $S \subseteq V$ of the nodes of G such that for every edge $\{u, v\} \in E$, at least one of the two nodes is in S, i.e., $\{u, v\} \cap S \neq \emptyset$. The objective of the minimum vertex cover problem is to compute a vertex cover of smallest possible size.

1. Show that in a d-regular graph G, taking $S = V$ gives a 2-approximation for the problem
2. Assume that in graph G, all nodes $v \in V$ have degree $\deg(v) \leq \Delta$. Consider the node set $A := \{u \in V : \deg(u) \geq \Delta/2\}$. How large can A be, compared to the smallest vertex cover of G?
3. Assuming that all nodes know Δ, can you use your result from b) to get a distributed approximation algorithm for the minimum vertex cover problem? What are approximation ratio and time complexity of your algorithm?
4. How would you have to adapt your algorithm if nodes do not know Δ?