Principles of Distributed Computing

Exercise 5: Sample Solution

1 Greedy Dominating Set

Our worst-case graph $G_x = (V_x, E_x)$ for $x \in \mathbb{N}$ is defined as follows. The node set $V_x$ consists of a node $r$, $x$ nodes $u_1, \ldots, u_x$, $2^{x+1} - 2$ nodes $v_{k\ell}$ for all $k \in \{1, \ldots, x\}$ and $\ell \in \{1, \ldots, 2^k\}$, and the two nodes $w_1$ and $w_2$. There are edges between $r$ and the nodes $u_i$ for all $i \in \{1, \ldots, x\}$. Each node $u_i$ is further connected to all nodes $v_{k\ell}$ for which $i = k$. Moreover, $w_1$ is connected to all nodes $v_{k\ell}$ for which $\ell \leq 2^{k-1}$, and $w_2$ is connected to all nodes $v_{k\ell}$ for which $\ell > 2^{k-1}$. As an example, the graph $G_3$ is given in Figure 1.

![Figure 1: The graph G3.](image)

Note that the number of nodes of $G_3$ is $3 + x + (2^{x+1} - 2) = 2^{x+1} + x + 1 \leq 2^{x+2}$. The degree $\delta(r)$ of $r$ is exactly $x$. We further have that $\delta(u_i) = 2^i + 1$, $\delta(v_{k\ell}) = 2$, and $\delta(w_1) = \delta(w_2) = 2^x - 1$.

In the first round, only $u_x$ is chosen, because it has the largest degree. Let $\delta^{(i)}(v)$ denote the number of white (i.e., uncovered) nodes in round $i$. After the first round, we have that $\delta^{(2)}(u_i) = 2^i$ for all $i \in \{1, \ldots, x-1\}$ and $\delta^{(2)}(v_{11}) = \delta^{(2)}(v_{12}) = 2^{x-1} - 1$. This means that only node $u_{x-1}$ is chosen in round 2. Inductively, we get that only node $u_{x-i+1}$ is chosen in round $i$, as $\delta^{(i)}(u_{x-i+1}) = 2^{x-i+1} > \delta^{(i-1)}(u_{x-i+1}) = \delta^{(x-i+1)}(v_{11}) = \delta^{(x-i+1)}(v_{12}) = 2^{x-i+1} - 1$ for all $i \in \{2, \ldots, x-1\}$. In round $x$, we have that $\delta^{(x)}(u_1) = \delta^{(x)}(v_{11}) = \delta^{(x)}(v_{12}) = 2$, thus identifiers have to be used to decide which nodes join the DS. In the worst case, three nodes, e.g., $w_1$, $w_2$, and $u_1$, are chosen to complete the DS.

Overall, $x + 2$ nodes are chosen. The optimal DS consists only of the nodes $r$, $w_1$, and $w_2$, hence the approximation ratio is

$$\frac{x + 2}{3} \geq \frac{\log n}{3} \in \Omega(\log n).$$
2 Fast Dominating Set

We describe the messages a node $v$ executing Algorithm 21 sends and receives. Note that the local computation necessary to compute the messages is omitted.

Algorithm 1 Fast Distributed Dominating Set Algorithm (at node $v$):

1: send ID to neighbors; receive IDs from neighbors
2: no communication
3: no communication
4: send $\tilde{w}(v)$ to neighbors; receive $\tilde{w}(u)$ from neighbors; forward $\tilde{w}(u)$ to neighbors; receive $\tilde{w}(u)$ from 2-hop neighbors
5: no communication
6: send $v.active$ to neighbors; receive $u.active$ from neighbors
7: send $s(v)$ to neighbors; receive $s(u)$ from neighbors
8: no communication
9: no communication
10: no communication
11: no communication
12: send $v.candidate$ to neighbors; receive $u.candidate$ from neighbors
13: send $c(v)$ to neighbors; receive $c(u)$ from neighbors
14: no communication
15: no communication
16: send $v.joined$ to neighbors; receive $u.joined$ from neighbors; send $v.white$ to neighbors; receive $u.white$ from neighbors
17: no communication;

Eight communication rounds are necessary for each phase.

3 Dominating Set on Regular Graphs

a) The number of steps each node has to execute is constant, thus the time complexity of this algorithm is $O(1)$.

b) A node can join the DS either in Step 1 or in Step 5. The probability that a node joins the DS in Step 1 is $\frac{\ln(\delta + 1)}{\delta + 1}$. The probability that a node joins the DS in Step 5 equals the probability that it neither joined the DS in step 1, nor has any neighbors that joined the DS in Step 1. This probability is $(1 - \frac{\ln(\delta + 1)}{\delta + 1}) \leq \frac{1}{\delta + 1}$. The expected size of the DS is hence

$$n \cdot (Pr[\text{node joins in Step 1}] + Pr[\text{node joins in Step 5}]) \leq \frac{n(\ln(\delta + 1) + 1)}{\delta + 1}.$$

c) In an optimal dominating set $DS^*$ there is at least one node per $\delta + 1$ nodes, since no node can dominate more than $\delta + 1$ nodes, i.e., $|DS^*| \geq \lceil \frac{n}{\delta + 1} \rceil$. Consequently, the expected approximation ratio of this algorithm is

$$\mathbb{E} \left[ \frac{|DS|}{|DS^*|} \right] \leq \frac{n(\ln(\delta + 1) + 1)/(\delta + 1)}{n/(\delta + 1)} = \ln(\delta + 1) + 1.$$