Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn June 20, 2013

Network Algorithms, Summer Term 2013
Problem Set 6 – Sample Solution

Exercise 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.

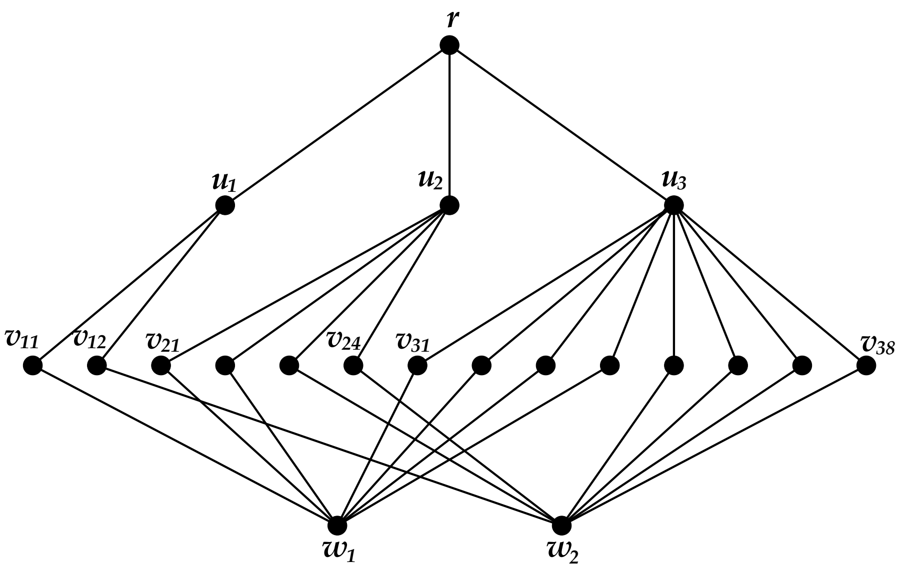

Figure 1: The graph G_3.

Note that the number of nodes of G_x 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 $\delta^{(2)}(u_i) = 2^i + 1$ for all $i \in \{1, \ldots, x-1\}$ and $\delta^{(2)}(w_1) = \delta^{(2)}(w_2) = 2^{x-1} - 1$. 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 $\delta^{(i)}(u_{x-i+1}) = 2^{x-i+1} + 1 > \delta^{(x-i+1)}(w_1) = \delta^{(x-i+1)}(w_2) = 2^{x-i+1} - 1$ 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

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

Exercise 2: Fast Dominating Set

We describe the messages a node v executing Algorithm 35 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):

```
send ID to neighbors; receive IDs from neighbors
no communication
no communication
send w̃(v) to neighbors; receive w̃(u) from neighbors; forward largest w̃(u) to neighbors; receive
    largest w̃(u) from 2-hop neighbors
no communication
send v.active to neighbors; receive u.active from neighbors
send s(v) to neighbors; receive s(u) from neighbors
no communication
no communication
no communication
no communication
send v.candidate to neighbors; receive u.candidate from neighbors
send c(v) to neighbors; receive c(u) from neighbors
no communication
no communication
send v joined MIS to neighbors, receive u joined MIS from neighbors; send v.white to neighbors,
    receive u.white from neighbors
no communication
```

There is one communication round at the beginning to exchange the IDs. Afterwards, 8 communication rounds are necessary for each phase (i.e., for each while loop iteration).

Exercise 3: Vertex Cover Approximation

1. Let $m = nd/2$ be the number of edges of G. Because a node with degree δ can cover at most its δ edges, a vertex cover of a d-regular graph needs to have at least $m/d = n/2$ nodes. The set V of all nodes is therefore a 2-approximation in d-regular graphs.

2. Using the same argument as before, no node can cover more than Δ edges. Because all the nodes in A have degree at least $\Delta/2$ and because each edge is incident to at most 2 nodes in A, G has at least $|A|\Delta/4$ edges. A dominating set of G therefore needs to have size at least $|A|/4$. The set A can therefore be at most 4 times as large as the smallest vertex cover of G.

3. Assume that the largest degree in G is Δ and for simplicity assume that the nodes know Δ. We get a distributed vertex cover algorithm as follows. The algorithm consists of $O(\log \Delta)$ phases. Initially the vertex cover is empty. In Phase i (the first phase is Phase 1), nodes with degree at least $\Delta/2^i$ join the vertex cover. After each phase, we remove all the edges that have been covered. Note that because in Phase i, all nodes with degree at least $\Delta/2^i$ join the vertex cover and all their incident edges are therefore removed at the end of Phase i, the maximum degree at the beginning of Phase $i+1$ is less than $\Delta/2^i$. Hence, by the observation in the b), in each phase, the number of nodes added to the vertex cover is at most 4 times as large as the size of the smallest vertex cover of G. Consequently, the algorithm computes an $O(\log \Delta)$-approximation in $O(\log \Delta)$ rounds.