— MPRI —

Correction Examen "Graphs and Algorithms"

February 10, 2009 — 08h45-10h15

Exercise I. Multicast

Question 1. We have $b(G, D, s) \geq \log |D|$ because the 1-port model implies that the number of nodes informed after each round can at most double.

To prove that $b(G, D, s) \geq \Delta^+_{min}(G, D)$, we consider any optimal multicast protocol $\mathcal{M}$, from s to D in G. Let $\mathcal{P}$ be the digraph consisting of all communication paths used during the execution of $\mathcal{M}$. For every node $u \in D$, there is a path from s to u in $\mathcal{P}$. Hence, let T be a tree rooted at s, spanning D in $\mathcal{P}$, and let Δ^+ be the maximum out-degree of T. $\mathcal{M}$ performs in at least Δ^+ rounds. Indeed, for every node u in T, if the out-degree of u is $\delta^+(u)$ then $\mathcal{M}$ performs in at least $\delta^+(u)$ rounds because of the 1-port model. Now, by definition of $\Delta^+_{min}(G, D)$, we have $\Delta^+ \geq \Delta^+_{min}(G, D)$.

Question 2. We proceed as in the case of undirected graphs, excepted that calls can only proceed downward the tree. We prune T^* by keeping only the "core tree" of T^* consisting of all nodes such that the subtrees rooted at these nodes contain at least $|D|/m$ nodes in D.

Our first objective is to inform all the nodes of D that are in the core, as well as all the nodes in the core from which are pending subtrees containing nodes in D outside the core. The difficulty comes from the fact that T^* is directed, with all its edges directed towards the leaves. Nevertheless, we note that the core tree has at most m leaves. A tree with m leaves has at most $m - 1$ internal nodes with out-degree ≥ 2 (called branching nodes). Therefore, s can inform the root of the tree T^*, and then, in $2m - 1$ rounds, all leaves and all branching nodes can be informed from s. Then all the branching nodes inform their children in at most Δ^+ rounds.

From these children nodes, in parallel, all nodes in the core tree that are either in D or roots of subtrees outside the core and containing nodes in D are informed in $|D^*| \leq \lceil \log |D| \rceil$ additional rounds.

Finally, the roots of all subtrees outside the core are informed in Δ^+ additional rounds.

We are let with rooted subtrees of T^* containing at most $|D|/m$ destination nodes, to be informed from the root of each subtree.

Question 3. By applying $\mathcal{A}$ we get a tree T^* with out-degree $\Delta^+ \leq O(\Delta^+_{min}(G) + \log n)$. By applying recursively the technique of the previous question with $m = \log n$, we obtain a multicast protocol performing in $O(\log_m n)$ phases, that is $O(\log n / \log \log n)$ phases. Each phase takes time at most $2m + \lceil \log n \rceil + 2\Delta^+$ rounds, that is at most $O(\Delta^+_{min}(G) + \log n)$ rounds. Thus, by question 1, we get that each phase takes at most $O(b(G, s))$ rounds.

To prove the time complexity $t(n) + O(n)$, we observe that the decomposition in Question 2 can be obtained by a single DFS traversal of the tree T^* counting the number of nodes in the subtrees.

Question 4. By the same arguments as in the previous question with $m = \log |D|$, we get that if we have a polynomial time $(O(1), r)$-approximation algorithm for the general version of the MDST problem, then we get a multicast algorithm performing in

$$O((\log|D| + \Delta^+_{min}(G,D) + r)\frac{\log|D|}{\log\log|D|})$$

rounds. Hence if $r = \alpha \log |D|$, this algorithm performs in at most

$$O(\alpha\, b(G,D,s)\,\frac{\log|D|}{\log\log|D|})$$

rounds. We have seen during the lectures that the multicast problem cannot be approximated within less that $\Omega(\log|D|)$ rounds, unless P=NP. Hence $\alpha \geq \Omega(\log\log|D|)$. Therefore, $r = \Omega(\log|D|\log\log|D|)$.

Exercise II. Informative labeling

Question 1. $T \setminus \{c\}$ is a forest. Assume that it has at most k trees. Label these trees from 1 to k. Each node is labeled by its distance to the center c, and the label of its subtrees. Hence labels are on $O(\log k + \log n)$ bits, i.e., on $O(\log n)$ bits. The decoder simply add the distances whenever the two labels indicate that the corresponding nodes belong to two different subtrees, are return "don't know" whenever the two labels correspond to nodes in the same subtree.

Question 2. We use the center decomposition of trees presented during the lecture. To every node u is associated a sequence of centers $S_u = (c_0, c_1, \ldots, c_r)$ where $c_0 = c$ and $c_r = u$. The distance between u and u' is $\text{dist}_T(u, c_j) + \text{dist}_T(u', c'_j)$ where $S_u = (c_0, c_1, \ldots, c_r)$ and $S_{u'} = (c'_0, c'_1, \ldots, c'_{r'})$ with $c_i = c'_i$ for all $i = 0, \ldots, j$ and $c_{j+1} \neq c'_{j+1}$.

So the labeling is as follows. Every node u is given a distinct ID, $id(u) \in \{0, \ldots, n-1\}$. The label of node u with $S_u = (c_0, c_1, \ldots, c_r)$ is composed of the $r+1$ pairs $(id(c_i), \text{dist}_T(u, c_i))$ for $i = 0, \ldots, r$. Each pair uses $2\lceil \log n \rceil$ bits. Since (as seen during the lectures) the center decomposition splits the current tree in subtrees of size at most half, we have $r \leq O(\log n)$. Hence the labeling scheme uses labels of $O(\log^2 n)$ bits in n-node trees.

Question 3. This is straightforward : every nodes is given a distinct ID in $\{0, \ldots, n-1\}$ and every node stores in its label the distances to all the other nodes :

$$label(u) = (id(u), d_0, d_1, \ldots, d_{n-1})$$

where $d_i = \text{dist}_G(u, v)$ with $id(v) = i$.

Question 4. Let $(T, \{X_i\})$ be a tree decomposition of G of width $\leq k$ and containing at most $\leq n$ bags. Let C be a center of T. Each node u of G has its representative bag defined as the bag X containing u that is closest to C in T. We apply the center decomposition on T. Let X and X' be the two bags of nodes u, and u', respectively. Let $S_X = (C_0, C_1, \ldots, C_r)$ and $S_{X'} = (C'_0, C'_1, \ldots, C'_{r'})$ be the two sequences obtained by the center decomposition of T. Assume $C_i = C'_i$ for all $i = 0, \ldots, j$ and $C_{j+1} \neq C'_{j+1}$. Therefore C_j separates u from v in G. Hence,

$$\text{dist}_G(u, u') = \min_{v \in C_j}(\text{dist}_G(u, v) + \text{dist}_G(v, u')).$$

The labeling is thus like in Question 2, but every node u stores the distances in G to the at most k nodes of each of the bags in $S_X = (C_0, C_1, \ldots, C_r)$ where X is its representative bag. This results in labels of size $O(k \log^2 n)$ bits.