Network Optimization by Randomization: Solution Distributed Algorithms and Social Networks

1 Vertex Coloring

a) Note that an "undecided" message can be realized by sending nothing at all; thus we do not count such messages. Therefore, each node sends exactly two messages to each neighbor, one in the first round and one after assigning a color. Hence, the total number of messages is $4|E|$, as 4 messages are sent over each edge.

b) Yes, the algorithm still works, it could be reformulated in the following way (we assume that each node knows its degree):

Algorithm 1 Asynchronous "$\Delta + 1$"-Coloring

c)

```
send node ID to all neighbors.
wait until all neighbor IDs have been received and all neighbors with a lower ID have chosen
  a color
choose smallest possible color
send chosen color to all neighbors
```

2 License to Match

a) We use an "Echo" algorithm. A node (i.e. an agent in the hierarchy) matches up all (except for at most one) of its children. If one participating child remains and the node itself also participates, it matches itself with that child. If either the node or a one of its children remain, then the node sends a request to "match" upwards in the hierarchy. Otherwise, it sends a "no match" and that subtree is done. We give an asynchronous, uniform matching algorithm below.

Algorithm 2 Edge-Disjoint Matching

```
wait until received message from all children
while at least 2 requests remain (including myself) do
  match any two requests
end while
if exists leftover request then
  send "match" to parent (= superior)
else
  send "no match" to parent
end if
```

When a node v sends a "match" request to its parent u, then the edge $\{u, v\}$ will be used only once since there will be only one request in the subtree rooted at v. Along with the messages of the algorithm, the required path information is sent; we left this out in the pseudocode to improve readability.

b) Let T be the tree with n nodes. Assuming each message takes at most 1 time unit, then the time complexity of Algorithm 2 is in $O(depth(T))$ since all the requests travel to the root (and back down if we inform the agents of their assigned partners). On each link, there are at most 2 messages: 1 that informs the parent whether a match is needed and optionally 1 more to be informed by the parent of the match partner. So there are a total of at most $2(n-1)$ messages.

3 Diameter of the Augmented Grid

a) Each link connects to the target set with probability $p \in \Omega((\log^2 n)/n)$. Thus, for sufficiently large[1] n, the probability that $cn/\log n$ many links miss the set is bounded by

$$(1-p)^{cn/\log n} \leq e^{-pcn/\log n} \in e^{-\Omega(c \log n)} = \frac{1}{n^{\Omega(c)}}.$$

Now we exploit the power of the Big-O notation. Choosing a sufficiently large multiplicative constant in front of the $(cn/\log n)$-term, this becomes a bound of $1/n^c$, and choosing a large additive constant, we make sure that the bound holds also for the values of n that are not "sufficiently large". Thus, the probability that at least one link enters the set of $\Omega(\log^2 n)$ nodes is at least $1 - 1/n^c$, i.e., this event occurs w.h.p.

In order to obtain the same result using Chernoff's bound, let X_i, $i \in \{1, \ldots, l\}$, where $l \in O(cn/\log n)$ is the number of considered links, be random variables that are 1 if the i^{th} link ends in the set (i.e., with the probability p from above) and 0 otherwise. Defining $X := \sum_{i=1}^{l} X_i$, we get that $E[X] = pl$. Picking a constant $C > 0$ and properly adapting the constants in the $O(cn/\log n)$-term, we get that $E[X] \geq C \log n$. Thus, Chernoff's bound states that for sufficiently large values of C and n (we need to cope with the fact that we do not know the constants in the O-term in Chernoff's bound), we have that $|X - E[X]| < C \log n$ w.h.p. Because $P[X > 0] \geq P[|X - E[X]| < E[X]]$, this is what we are looking for.[2]

b) Because $|S| \in o(n)$, also $O(|S|) \subset o(n)$, i.e., the union of the set S ($|S|$ nodes) with the destinations of the $|S|$ random links and all grid neighbors of such nodes (at most $4|S|$ many nodes) has $o(n)$ nodes. Thus, always $(1-o(1))n$ nodes can be found which neither have been visited themselves nor have any neighbors that have been visited so far. Hence, regardless of the choice of the set S and any random links leaving S we have (sequentially) examined up to now, any uniformly independent random choice will contribute 5 new nodes with some probability $p \in 1-o(1)$. Now we use Chernoff's bound on the number of such "good" choices, yielding that it will be in $(1-o(1))|S|$ w.h.p. (instead of just in expectation).[3] Thus, in total we reach $(5 - o(1))|S|$ many nodes.

c) Recall that we may choose the constant c in "w.h.p." by ourselves. Thus, we may rule that in Chernoff's bound, it is $c' := c + 1$. Hence, the probability that in a given step our set grows by a factor $(5 - o(1))$ (provided that $|S| \in o(n)$, as we use part **b)**) is always at least $1 - 1/n^{c'}$. This means in at most a fraction of $1/n^{c'}$ of the events, something goes wrong in a single step. We need less than $\log n$ steps to get to $O(n/\log n)$ nodes, as the number of nodes more than quadruples in each step. In total, in a fraction of less than $\log n/n^{c'} = \log n/n \cdot 1/n^c < 1/n^c$ of all cases something goes wrong.

d) Using the union bound again, we plug together the facts that (i) each node can reach $\Omega(\log^2 n)$ nodes following grid links only within $\log n$ steps, (ii) starting from these nodes, with high probability $O(n/\log n) \subset o(n)$ nodes can be reached within $O(\log n)$ more hops (part **b)**), (iii) from these nodes we reach with high probability the $(\log n)$-neighborhood (with respect to the grid) of any node (part **a**), and (iv) from there on we can reach the respective node with $\log n$ hops on the grid. Altogether, with high probability in total $O(\log n)$ hops are necessary to reach some node v starting at some other node u. Finally, observe that we have $n(n-1) < n^2$ possible (ordered) combinations of nodes; choosing

[1] This phrase means for some constant n_0, the statement will hold for all $n \geq n_0$.

[2] Small values of n are again dealt with by the additive constant in the O-notation. In general, it is always feasible to assume that n is "sufficiently large" when proving asymptotic statements.

[3] Since the expected value $E[X]$ of the respective random variable X is large compared to $\log n$ (here we use $|S| \in \Omega((\log n)^2)$), the deviation from the expected value is with high probability in $o(E[X])$.

$c' := c + 2$ and applying the union bound once more, we infer that we have with high probability a path of length $O(\log n)$ between any pair of nodes, i.e., the diameter of the graph is in $O(\log n)$ w.h.p.

References

[1] W. H. Gates, C. H. Papadimitriou, Bounds for sorting by prefix reversal, Discrete Math. 27, (1979), 47–57.