Homework 10

Randomized Algorithms

Due Wednesday, November 29

1. In class we showed that network coding works well on a static graph. The key property was that, if vertex $v$ is “aware” of a vector $u$ in one round, then each neighbor becomes aware of it in the next round independently with probability at least $1 - 1/q$. We showed that this implies that after $R$ rounds, the destination $t$ becomes aware of each $u$ with probability $1 - q^{-C_{s,t} R (1 - \epsilon)}$, where $C_{s,t}$ is the $(s, t)$ min cut and suitably large parameters ($q > 2^{O(1/\epsilon)}$ and $R > O(n/\epsilon)$).

In this problem we extend this to dynamic graphs. We instead suppose that the graph changes arbitrarily in every round, with the condition that the $(s, t)$ min cut is at least $C$ in each round.

(a) For any $u$ that $s$ is aware of, at the beginning of round $i$ let $S_i$ be the set of vertices that are aware of $u$. Show that, if $t \notin S_i$, then over the randomness in round $i$ we have

$$\Pr[S_{i+1} = S_i] \leq q^{-C}.$$ 

That is to say, almost always at least one new vertex will become aware of $u$.

(b) Show that, after $R \geq O(n/\epsilon)$ rounds and with $q \geq 2^{O(1/\epsilon)}$,

$$\Pr[t \notin S_R] \leq q^{-CR(1 - \epsilon)}.$$ 

(c) Suppose that $s$ starts with a $k$-dimensional subspace. Show that if $R > O(n/\epsilon)$ and $R > (k/C)(1 + O(\epsilon))$, there is a large probability that $t$ learns $s$’s subspace in its entirety.

(d) (Optional) Show that, in general, no algorithm can transmit a dimension-$k$ subspace from $s$ to $t$ in $R < (1 - \epsilon)k/C$ rounds.