Congestion and Minors

1 Connecting Terminals to Minimize Congestion

1.1 Setup

Recall from last time that we came up with the following LP:

$$\begin{aligned}
\min\ & c \\
\text{s.t.}\ & \sum_{e \text{ out of } s_i} f_i(e) = 1 && \forall i \\
& \sum_{e \text{ into } v} f_i(e) = \sum_{e \text{ out of } v} f_i(e) && \forall i \text{ and } \forall v \notin \{s_i, t_i\} \\
& f_i(e) \geq 0 && \forall i \text{ and } \forall e \\
& c \geq 1 \\
& c \geq \sum_i f_i(e) && \forall e
\end{aligned}$$

Recall that the path decomposition of a flow f from s to t is paths $P_1, P_2, \ldots, P_\ell$ from s to t with flow amounts $\alpha_1, \alpha_2, \ldots, \alpha_\ell \geq 0$ such that for each edge e,

$$\sum_{i: e \in P_i} \alpha_i = f_e.$$

That is, a flow is equivalent to a distribution over paths. There always exists a decomposition into at most m paths, and it can be found efficiently – in other words, $\ell \leq m$ is enough.

Specifically for our LP, focusing on one flow f_i, we have $P_{i1}, P_{i2}, \ldots, P_{i\ell_i}$ with $\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{i\ell_i} \geq 0$ such that

$$\sum_{j=1}^{\ell_i} \alpha_{ij} = 1.$$

Now we can interpret the α_{ij} as probabilities. For each terminal pair i, pick path P_{ij} with probability α_{ij}. Make the choices for different i independently.

1.2 Analysis

We want to bound the maximum congestion for any edge. First, focus on the congestion C_e of edge e. Let X_{ei} be an indicator variable representing whether (s_i, t_i) uses e.

$$\begin{aligned}
C_e &= \#\text{ of paths using } e \\
&= \sum_i X_{ei}, \\
\mathbb{E}[C_e] &= \sum_i \Pr[(s_i, t_i) \text{ uses } e] \\
&= \sum_i \sum_{j: e \in P_{ij}} \alpha_{ij} \\
&= \sum_i f_i(e).
\end{aligned}$$

We really care about $C = \max_e C_e$. To get a handle on C, we use tail bounds to get high-probability guarantees for all e. Using Chernoff bounds,

$$\Pr\left[C_e \geq (1+\delta)\mathbb{E}[C_e]\right] \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{\mathbb{E}[C_e]}.$$

We'd like this to be $\leq 1/n^3$ to do a union bound. Essentially, δ will be our approximation guarantee, so we'd like to make δ small. Which range of $\mathbb{E}[C_e]$ is most problematic? Since what's in the parentheses is always less than 1, large $\mathbb{E}[C_e]$ are good and small $\mathbb{E}[C_e]$ make it harder.

For now, focus on $\mathbb{E}[C_e] \geq 1$. Then the worst-case is $\mathbb{E}[C_e] = 1$ by our LP constraint.

$$\Pr\left[C_e \geq (1+\delta)\right] \leq \frac{e^\delta}{(1+\delta)^{(1+\delta)}}.$$

Using $\delta = \Theta(\log n / \log\log n)$, we obtain $1/n^3$ as desired (by balls-in-bins analysis).

What about when $\mathbb{E}[C_e] < 1$? Then, since $c \geq 1$, we have slack on the LHS: we want

$$\Pr\left[C_e \geq (1+\delta) \cdot 1\right]$$

to be small. So, we want a Chernoff bound expressed in terms of an upper bound on $\mathbb{E}[C_e]$. A useful upper-bound form of the Chernoff bound for $\mathbb{E}[X] \leq \hat{\mu}$ is:

$$\Pr\left[X > z\right] \leq \left(\frac{e\hat{\mu}}{z}\right)^z.$$

This follows because

$$\begin{aligned}
\Pr[X > z] &= \Pr\left[X > \left(1 + \left(\frac{z}{\mu} - 1\right)\right)\mu\right] \\
&< \left(\frac{e^{\frac{z}{\mu} - 1}}{\left(\frac{z}{\mu}\right)^{z/\mu}}\right)^\mu \\
&= \frac{e^{z-\mu}\mu^z}{z^z} \\
&\leq \left(\frac{e\mu}{z}\right)^z \\
&\leq \left(\frac{e\hat{\mu}}{z}\right)^z.
\end{aligned}$$

This form is useful when you want to apply Chernoff bounds, but only have an upper bound on $\mathbb{E}[X]$. Applying this to our case with $\mu \leq 1$ and $c \geq 1$,

$$\begin{aligned}
\Pr\left[C_e > \delta\right] &\leq (\frac{e}{\delta})^{\delta} \\
&= \frac{e^{\delta}}{\delta^{\delta}} \\
&\approx \frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}
\end{aligned}$$

because $\delta >> 1$. So $\delta = \mathcal{O}(\log n / \log\log n)$ still gives a bound of $1/n^3$ of one specific e being overcongested. By the union bound over all edges, the congestion is at most $\mathcal{O}(\log n / \log\log n)$ with probability at least $1 - 1/n$.

1.3 Tightness

How tight is the analysis? Does the algorithm actually do better than $\Theta(\log n / \log\log n)$? Consider the case where we have m different sources in the same node, and m different sinks in the same node, with m edges to choose from. The solution is to pick a different edge for every terminal pair, but if the LP splits each flow uniformly over the m edges, randomized rounding becomes balls-in-bins. That is, with high probability, at least one bin has $\Omega(\log m / \log\log m)$ balls.

2 A Brief Overview of Graph Minors and the Robertson-Seymour Theorems

These theorems were built up by Robertson and Seymour over 20+ papers and a 600 page proof. They are useful for proving the existence of a polynomial time algorithm. Define the **contraction** of an edge $e = (u, v)$ in an undirected graph as the removal of u and v, replacing them with a new vertex w that inherits all their edges (but no self-loop).

Graph H is a **minor** of graph G iff H can be obtained from G via edge or vertex deletions and edge contractions. H is a **topological minor** of G if there is a bijective mapping $\varphi : V(H) \to V(G)$ such that for each $e = (u, v) \in E(H)$, we have a path P_e in G from $\varphi(u)$ to $\varphi(v)$ and the P_e are edge-disjoint. That is, edges in H correspond to edge-disjoint paths in G.

Theorem 1. *[Robertson and Seymour]. Given a graph H of size k and G of size n, it can be tested in time $\mathcal{O}(f(k) \cdot n^3)$ if H is a minor (or topological minor) of G. Note: the growth of $f(k)$ is horrendous, but if k is fixed, then this is a polynomial time algorithm.*

A **graph property** is just something that is true or false for each undirected graph. For example: planarity, connectedness, colorability, etc. A graph property P is closed under minors if whenever G has property P and H is a minor of G, H also has P. (It does not apply in the other direction). For example: planarity (and embeddability more generally), *dis*connectedness, and (indirectly) having EDP between k terminal pairs (with k and the terminals fixed).

Theorem 2. *[Robertson and Seymour]. If P is closed under taking minors, then there exists a finite set $S_P = \{H_1, H_2, \ldots, H_{m(P)}\}$ of graphs such that any graph G has property P iff G contains none of the H_j as minors.*

Example 3. For planarity, we have $S_P = \{K_5, K_{3,3}\}$.

Corollary 4. *If P is closed under taking minors, then there is an $\mathcal{O}(n^3)$ algorithm to decide if G has property P. Of course, $m(P)$ can be extremely large, the graphs in S_P can be large, and $f(|H_i|)$ grows fast.*

3 Permutation Routing on a Hypercube

We have a d-dimensional hypercube with $n = 2^d$ nodes. Each node u wants to send a packet to some node $\sigma(u)$. The σ form a permutation. The edges of the hypercube are used for routing, and can have queues for storing packets if more than one packet is ready to cross an edge at the same time. In each timestep, at most one packet can cross each edge.

An algorithm is **oblivious** if the route of a packet i depends only on i and $\sigma(i)$, but not $\sigma(j)$ for any $j \neq i$. This is similar to a parallel computer in that each processor makes its own decisions independent of the other processors.

A simple algorithm is to consider $i \otimes \sigma(i)$ and use hypercube edges to "fix" bits from i to $\sigma(i)$ one at a time, in some fixed order (*e.g.*, left to right). Each edge has a FIFO queue.

Next time, we'll see that this algorithm can create (under some permutation σ) a lot of congestion at some edges, and hence a lot of delay.