Choosing Paths to Minimize Congestion using Randomized Rounding

Lecture Notes for CSCI 570 by David Kempe

December 2, 2007

To see a more complex application of Chernoff and Union Bounds, we will consider a randomized approximation algorithm for a routing problem trying to minimize congestion. We are given a (directed) graph $G = (V, E)$, with source-sink pairs (s_i, t_i). Each pair should be connected with a single path P_i. The *congestion* (or *load*) L_e of an edge e is the number of paths P_i using e, and our goal is to minimize the maximum load of any edge $\max_e L_e$. This problem is NP-complete, since even deciding if it can be solved with maximum load 1 is the edge-disjoint paths problem.

We will derive an approximation algorithm based on LP rounding. To start, we phrase the problem as an ILP with exponentially many variables. For each pair (s_i, t_i), and each s_i-t_i path P, we have a variable $x_{i,P}$: if $x_{i,P} = 1$, this means that the pair (s_i, t_i) uses the path P to connect; otherwise, it does not. We also have one more variable, L, the maximum load of any edge. Thus, we get the following ILP:

$$\begin{array}{ll}
\text{Minimize} & L \\
\text{subject to} & \sum_P x_{i,P} = 1 \qquad \text{for all } i \\
& L \geq \sum_i \sum_{P:e \in P} x_{i,P} \qquad \text{for all } e \\
& L \geq 1 \\
& x_{i,P} \geq 0 \qquad \text{for all } i, P \\
& x_{i,P} \in \{0, 1\} \qquad \text{for all } i, P.
\end{array}$$

The first constraint states that each pair must select exactly one path. The second constraint says that L is at least the maximum load of any edge. Because the objective is to minimize L, it will not be any larger than necessary, i.e., equal to the maximum load. The final two constraints are just the standard non-negativity and integrality constraints.

The third constraint is redundant for the integer LP, since any integer solution will have $L \geq 1$. However, we use it to strengthen the LP. Otherwise, the integrality gap will be very large, as we can see with an example with just one s_i-t_i pair, but with m parallel edges from s_i to t_i (or two-edge paths). A fractional solution to this instance could assign $x_{i,P} = 1/m$ to each of these parallel edges, for a load of $1/m$. The integral solution must pick one edge, and thus the integrality gap is at least m. The $L \geq 1$ constraint rules out this fractional solution.

As usual, we drop the integrality constraint. However, it is not clear how to solve an LP with exponentially many *variables*. We saw before that exponentially many constraints are not a problem so long as we have membership and separation oracles. But exponentially many variables are: among others, even writing down or reading the solution would take exponential time. However, we notice that the fractional LP really just describes sending one unit of flow from each s_i to the corresponding t_i, and minimizing the maximum flow through any edge. Thus, we can rewrite the fractional LP as follows:

$$\begin{array}{ll}
\text{Minimize} & L \\
\text{subject to} & \sum_{e \text{ out of } s_i} f_{i,e} = 1 \qquad \text{for all } i \\
& \sum_{e \text{ out of } v} f_{i,e} = \sum_{e \text{ into } v} f_{i,e} \qquad \text{for all } i, v \neq s_i, t_i \\
& L \geq \sum_i f_{i,e} \qquad \text{for all } e \\
& L \geq 1 \\
& f_{i,e} \geq 0 \qquad \text{for all } i, e
\end{array}$$

The fractional multi-commodity LP can now be solved in polynomial time, and the variables $x_{i,P}$ we really are interested in can be found from the flows $f_{i,e}$ using path decomposition of each f_i in polynomial time. Notice that this only gives polynomially many non-zero values $x_{i,P}$.

To decide on one path for each s_i-t_i pair, we observe that paths with larger $x_{i,P}$ values are better candidates for the path P_i, but we shouldn't simply commit to the single largest $x_{i,P}$ value, as that might overload one edge with many slightly larger values. Instead, we interpret the $x_{i,P}$ values as probabilities. For each s_i-t_i pair, we independently choose one path P with probability $x_{i,P}$. That is, we divide the $[0, 1]$ interval into disjoint intervals of length $x_{i,P}$, and label them with the corresponding path P. Then, we choose a uniformly random number from $[0, 1]$, and pick the path P corresponding to the chosen point. This defines a polynomial time algorithm picking exactly one path for each pair.

To analyze the approximation guarantee, we focus on one edge e at a time. Let X_e be the load on edge e, and write $X_{i,e}$ for the indicator random variable which is 1 if pair i connects via a path using edge e, and 0 otherwise. Thus, $X_e = \sum_i X_{i,e}$ is a sum of indicator random variables. Notice that $\mathrm{E}[X_{i,e}] = f_{i,e}$, so $\mathrm{E}[X_e] = \sum_i f_{i,e} \leq L$.

To show that X_e does not deviate much from its expectation, we use the Chernoff Bound. Notice that the $X_{i,e}$ are indeed independent indicator variables. Thus, $\mathrm{Prob}[X_e \geq (1+\delta)L] < (\frac{e^\delta}{(1+\delta)^{1+\delta}})^L$ for any δ. Because $L \geq 1$ by the added (strengthening) LP constraint, using the same analysis as in Chapter 13.10 of the textbook, we see that $\delta = \Theta(\frac{\log m}{\log \log m})$ is sufficient to guarantee that $\mathrm{Prob}[X_e \geq (1+\delta)L] < \frac{1}{m^2}$. We can then take a Union Bound over all m edges, and obtain that with probability at least $1 - 1/m$, the randomized rounding will give a set of paths with maximum load at most $L \cdot O(\frac{\log m}{\log \log m})$. That is, the algorithm is an $O(\frac{\log m}{\log \log m})$ approximation. Notice also that by the same type of analysis as in Section 13.10, whenever the fractional solution L is large (say, at least $16 \log m$), the algorithm actually gives a constant-factor approximation.