Lecture 10

1 Reminder: routing in general networks

On previous lectures we addressed the problem of finding optimal paths in arbitrary networks: given $G = (V, E)$, and pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k) \in V \times V$, find paths $p_1, \ldots, p_k$ in G, such that p_i connects s_i with t_i and the maximum number of paths that use any edge is minimized. Recall that we considered that to be a resource optimization problem, and therefore represented it as a *linear program* of the form

$$\begin{aligned} &\textbf{minimize } C \\ &\textbf{subject to} \quad \sum_{p \text{ uses } e} f(p) \le C \qquad \text{for each } e \in E, \\ &\qquad\qquad\qquad \sum_{\substack{p \text{ is} \\ s_i\text{–}t_i \text{ path}}} f(p) = 1 \qquad \text{for } 1 \le i \le k, \end{aligned}$$

where $f(p)$ associates a non-negative flow with each path. Note, that the above formulation implies an exponential number of variables (i.e., graph paths), but we argued that it can be replaced with an alternative tractable formulation that yields an equivalent solution set. Also note, that an exact solution to the path selection problem can immediately be obtained by adding the following constraint

$$f(p) \in \{0, 1\} \qquad \text{for each path } p,$$

thus switching to an *integer program*. However, as the latter is generally intractable, we avoid adding it and instead apply *random rounding* to the LP solution in order to obtain an approximate integer one: select a path p with probability $f(p)$, as returned by the LP procedure. In this case, we have shown that

$$C_{\max} \le C_{\text{lp}} + \theta\left(\sqrt{C_{\text{lp}} \log n} + \log n\right) .$$

Later, after examining the *dual form* of the above LP, we stated and proved the following claim.

LEMMA 1
Given some non-negative distance function $d : E \to \mathbb{R}^+$, the optimal (minimal) congestion C_{opt} is bounded by

$$C_{\text{opt}} \ge \frac{\sum d(s_i, t_i)}{\sum d(e)} ,$$

where $d(s_i, t_i)$ is the length of the shortest s_i–t_i path.

Finally, we presented an algorithm for directly solving the path selection problem in arbitrary graphs, which iteratively revises congested paths by penalizing for highly congested edges.

ALGORITHM 1 (PATH SELECTION)
Require: $\alpha, \beta > 1$
choose arbitrary paths p_i, for all i
$d(e) \leftarrow \alpha^{f(e)}$
while exists i such that $d(p_i) \geq \alpha\beta \cdot d(s_i, t_i)$ **do**
$p_i \leftarrow$ shortest s_i–t_i path
update $d(e)$
end while

In the following, we will show an upper bound to the approximation ratio of Algorithm 1, compared to the optimal solution, as well as prove its termination.

2 Bounding the approximation ratio

The question in this case is: how good is Algorithm 1, compared to the optimal solution? We can see that—upon termination (which will be proved later on)—the termination condition guarantees

$$\alpha\beta \sum d(s_i, t_i) \geq \sum_{p_i} d(p_i) \ .$$

Thus, by Lemma 1 we have

$$C_{\text{opt}} \geq \frac{\sum d(p_i)/\alpha\beta}{\sum d(e)} \ , \tag{1}$$

yet further refinement is needed... Let n_l denote the number of edges in E with congestion l, then

$$\sum d(e) = \sum_{l=0}^{C_{\max}} \alpha^l n_l \ , \tag{2}$$

and by summing the total cost over all paths we get

$$\sum d(p_i) = \sum_{l=0}^{C_{\max}} \alpha^l n_l l \ . \tag{3}$$

From (1), (2), and (3) it follows that

$$C_{\text{opt}} \geq \frac{\sum_{l=0}^{C_{\max}} \alpha^l n_l l/\alpha\beta}{\sum_{l=0}^{C_{\max}} \alpha^l n_l} \ . \tag{4}$$

We now want to show that *most* of the congestion cost induced by the distance metric is due to edges with high congestion values (recall that penalty is exponential by the congestion). We therefore split the sum into two parts, and bound the cost for low-congested edges, and that of highly-congested edges, separately. Let $l' = C_{\max} - 2\log_\alpha |E|$, then

$$\begin{aligned}
\sum_{l=0}^{l'} \alpha^l n_l \leq \alpha^{l'} \sum_{l=0}^{l'} n_l & \\
= \frac{\alpha^{C_{\max}}}{|E|^2} \sum_{l=0}^{l'} n_l & \qquad \text{by definition of } l' \\
\leq \frac{\alpha^{C_{\max}}}{|E|} & \qquad \text{since } \sum_{l=0}^{l'} n_l \leq \sum_{l=0}^{C_{\max}} n_l = |E| \\
\leq \frac{1}{|E|} \sum_{l>l'}^{C_{\max}} \alpha^l n_l & \qquad \text{since at least one edge has congestion } C_{\max}.
\end{aligned}$$

Therefore, we have

$$\sum_{l=0}^{C_{\max}} \alpha^l n_l \leq \left(1 + \frac{1}{|E|}\right) \sum_{l>l'}^{C_{\max}} \alpha^l n_l \ . \tag{5}$$

It also holds that

$$\sum_{l=0}^{C_{\max}} \alpha^l n_l l \geq \sum_{l>l'}^{C_{\max}} \alpha^l n_l l \geq l' \sum_{l>l'}^{C_{\max}} \alpha^l n_l \ . \tag{6}$$

Then, using (5) and (6), we can further refine (4) to

$$C_{\text{opt}} \geq \frac{l' \sum_{l>l'}^{C_{\max}} \alpha^l n_l / \alpha\beta}{\left(1 + \frac{1}{|E|}\right) \sum_{l>l'}^{C_{\max}} \alpha^l n_l} = \frac{(C_{\max} - 2\log_\alpha |E|)/\alpha\beta}{\left(1 + \frac{1}{|E|}\right)} \ ,$$

by that leading to the following upper bound

$$C_{\max} \leq \alpha\beta \left(1 + \frac{1}{|E|}\right) C_{\text{opt}} + 2\log_\alpha |E| \ .$$

Keeping α, β close to 1 would guarantee a good practical bound. Also note, that $\left(1 + \frac{1}{|E|}\right)$ collapses to 1, but we're still left with the logarithmic added factor, which is hard to reduce.

3 Proving termination

The following observation establishes the proof of the convergence of Algorithm 1.

Claim 2
At any iteration of Algorithm 1, $\sum d(e)$ strictly decreases.

Proof: Let p_i be the path to be replaced by the shortest s_i–t_i path p_i^* at some iteration, and let f' and d' denote the resulting flow and distance functions, respectively. Then

$$f'(e) = \begin{cases} f(e) - 1 & \text{if } e \in p_i \text{ and } e \notin p_i^*, \\ f(e) + 1 & \text{if } e \notin p_i \text{ and } e \in p_i^*, \\ f(e) & \text{otherwise,} \end{cases}$$

and

$$d'(e) = \begin{cases} d(e)/\alpha & \text{if } e \in p_i \text{ and } e \notin p_i^*, \\ d(e) \cdot \alpha & \text{if } e \notin p_i \text{ and } e \in p_i^*, \\ d(e) & \text{otherwise.} \end{cases} \tag{7}$$

Therefore,

$$\begin{aligned}
\sum d'(e) - \sum d(e) &= \sum \big(d'(e) - d(e)\big) \\
&= \sum_{e \in p_i^* \setminus p_i} d'(e) - d(e) + \sum_{e \in p_i \setminus p_i^*} d'(e) - d(e) \\
&= (\alpha - 1) \sum_{e \in p_i^* \setminus p_i} d(e) - \left(1 - \frac{1}{\alpha}\right) \sum_{e \in p_i \setminus p_i^*} d(e) \\
&\leq (\alpha - 1) \sum_{e \in p_i^*} d(e) - \left(1 - \frac{1}{\alpha}\right) \sum_{e \in p_i} d(e) && \text{since } \alpha - 1 > 1 - \frac{1}{\alpha} \\
&\leq \frac{\alpha - 1}{\alpha\beta} \sum_{e \in p_i} d(e) - \left(1 - \frac{1}{\alpha}\right) \sum_{e \in p_i} d(e) && \text{by the iteration condition} \\
&= \left(\frac{1}{\beta} - 1\right) \left(1 - \frac{1}{\alpha}\right) \sum_{e \in p_i} d(e) \\
&< 0 && \text{as } a, b > 1 \text{ and } d(e) > 0.
\end{aligned}$$

□

Note, that through a more delicate analysis, the algorithm can be shown to converge within a polynomial number of iterations.