Min Weight Lower Bound

Max Bender

Connecticut College

August 12, 2015

The two instances we will consider are as follows:
Instance 1: Servers are located at $S = \{-c^k, -c^{k-1}, ..., -c^2, -c, c, c^2, ..., c^{k-1}, c^k\}$ where $3 < c \in \mathbb{R}$. Requests will be denoted $R = (q_1, q_2, ..., q_{2k})$. We will always set $q_1 = 0$, and define $\ell_1 = -c$ and $r_1 = c$. These will denote the next places for the requests to possibly land, with a uniform probability of choosing either at any time. Thus, if $q_{n+1} = \ell_n$, then $\ell_{n+1} = c\ell_n$ and $r_{n+1} = r_n$. Or, if $q_{n+1} = r_n$, then $\ell_{n+1} = \ell_n$ and $r_{n+1} = cr_n$.
Instance 2: Instance 2 will work in a similar way. Servers are located at $S = \{-k, -k+1, ..., -2, -1, 1, 2, ..., k-1, k\}$. Requests will be denoted $R = (q_1, q_2, ..., q_{2k})$. We will always set $q_1 = 0$, and define $\ell_1 = -1$ and $r_1 = 1$. Again, we will choose the left and the right options with uniform probability. Then if $q_{n+1} = \ell_n$, then $\ell_{n+1} = \ell_n - 1$ and $r_{n+1} = r_n$. Or, if $q_{n+1} = r_n$, then $\ell_{n+1} = \ell_n$ and $r_{n+1} = r_n + 1$.

Theorem 1. The expected cost of the optimal offline matching on Instance 1 is

$$\sum_{n=1}^{k-1} \sum_{i=n+1}^{2k-n-1} \binom{2k-1}{n} \frac{c^i}{2^{2k-2}}.$$

Proof. First, we will establish the probability corresponding to any layout of the requests. Note that at any time, there will never be a jump in the requests - it will always be a continuous string of integers from some point $-j$ to $2k - j - 1$ (the -1 comes from the request at 0). Further, the order of the requests does not matter in the offline solution, meaning the number of the requests on the positive or negative side does not change if you change the ordering of a sequence of requests. Thus, the probability that $n < k$ requests are positive is precisely $\binom{2k-1}{n}\frac{1}{2^{2k-1}}$, since there are $\binom{2k-1}{n}$ ways to arrange the $2k-1$ requests such that n of them are positive. Further, the expected cost of any optimal matching (which we determine by assigning the most positive request with the most positive server and repeating with the remaining sets of each) will be

$$
\begin{aligned}
&(c^k - c^n) + ... + (c^{k-n+1} - c) + (c^{k-n} + 0) + (c^{k-n-1} + c) + ... + (c + c^{k-n-1}) \\
&+ (-c + c^{k-n}) + ... + (-c^k + c^{2k-n-1}) = \sum_{i=1}^{k} c^i - \sum_{i=1}^{n} c^i + \sum_{i=1}^{2k-n-1} c^i - \sum_{i=1}^{k} c^i \\
&= \sum_{i=n+1}^{2k-n-1} c^i
\end{aligned}
$$

Now, the probability of each of these costs is actually doubled because n positive requests is equally as likely as n negative requests and by our assumption that $n < k$ we have assured no repeats. Thus the expected cost is precisely

$$\sum_{n=1}^{k-1} \sum_{i=n+1}^{2k-n-1} \binom{2k-1}{n} \frac{c^i}{2^{2k-2}}$$

□

Theorem 2. The expected cost of the optimal offline matching on Instance 2 is

$$\sum_{n=1}^{k-1} \binom{2k-1}{n} \frac{k(2k-2n-1)}{2^{2k-2}}.$$

Proof. Much of the first proof remains the same - we do not need to recalculate the probability of $n < k$ requests being positive, only their new costs. Now, the total cost of a matching with n positive requests is

$$
\begin{aligned}
&(k - n) + ... + (k - n + 1 - 1) + (k - n + 0) + (k - n - 1 + 1) + ... + (1 + k - n - 1) \\
&+ (-1 + k - n) + ... + (-k + 2k - n - 1) = \sum_{i=1}^{k} (k - n) + \sum_{i=1}^{k} (k - n - 1) \\
&= k(2k - 2n - 1)
\end{aligned}
$$

Thus the expected total cost of the matching is

$$\sum_{n=1}^{k-1} \binom{2k-1}{n} \frac{k(2k-2n-1)}{2^{2k-2}}$$

□