1 The K-server problem

Online algorithms for the K-server problem are considered. K servers need to be moved around to service requests appearing online at points of a metric space.The total distance travelled by the K servers must be minimised, where any request arising at a point of the metric space must be serviced on site by moving a server to that site.$d(a_1, a_2)$ is defined as the distance between a_1 and a_2. M represents the metric space where d is the metric which satisfies the triangle inequality. M^K represents the set of configurations of the K points of M. Given configurations C_1 and C_2, $d(C_1, C_2)$ is the minimum possible distance travelled by K servers that change configuration from C_1 to C_2. $C_0 \in M^K$ is the initial configuration. Let $r = (r_1, r_2, ...r_m)$ be the sequence of request points in M. The solution $C_1, C_2, ..., C_m \in M^K$ is such that $r_t \in C_t, \forall t = 1....m$. Serving $r_1, r_2, ...r_m$ by moving through $C_1, C_2, ..., C_m$ entails solution cost $\sum_{t=1}^{m} d(C_{t-1}, C_t)$.

The online algorithm uses only $r_1, r_2...r_t$ and $C_0, .., C_{t-1}$ to compute C_t. The offline algorithm uses also $r_{t+1}, r_{t+2}...r_m$. Given C_0, $r = (r_1, r_2, ...r_m)$, $cost_A(C_0, r)$ is the cost of the online algorithm A, and $opt(C_0, r)$ is the cost of the optimal algorithm. ρ is the competitive ratio. Competitive ratio is used as $cost_A(C_0, r) < \rho * opt(C_0, r) + \phi(C_0)$ for some ρ. $\phi(C_0)$ is independent of r. ρ_M may be used for metric space M. Conjecture: For every metric space with more than K distinct points the competetive ratio for the K-server problem is exactly K. $\rho = \inf_A \sup_r \frac{cost_A(C_0,r)}{opt(C_0,r)}$, modulo a constant term.

Theorem 1 *In every metric space with at least $K + 1$ points, no online algorithm for the K-server problem can have competitive ratio less than K.*

We wish to show there are request sequences of arbitrary high cost for A for which the online algorithm A has cost K times that of the optimal offline algorithm. We prove this lower bound result later below.

Now we consider the *double coverage* strategy, used for the online algorithm A to achieve the competitive ratio K. Let a_1, a_2, a_3 be ordered left to right on a horizontal line with a_2 closer to a_1 than a_2. Let servers s_1 and s_2 be at a_1 and a_3 respectively, initially, with no server at a_2.

Let serving requests come repeatedly alternating between a_2 and a_1. If we move only the (closest) server s_1 (which was intitally stationed at a_1) up and down between a_1 and a_2 for the sequence $a_2, a_1, a_2, a_1, \cdots$, we incur unbounded competive ratio for asympopically large strings of requests a_2, a_1. This is so because the offline algorithm would place servers s_1 and s_2 at a_1 and a_2 respectively, permanently, instead of fixing s_2 at a_3. In the *double coverage* strategy instead, we move both s_1 and s_2 towards a_2 by amount $d(a_1, a_2)$ on serving request a_2, and then move s_1 back to a_1 on serving request a_1. So we use travel cost at most 3 times of that used by the optimal offline algorithm, which moves s_2 only once to a_2 on the first serving request for a_2.

We continue to analyse the double coverage strategy whose suggested ratio is 3 for $K = 2$ servers. Note that consecutive configurations C_t, C_{t-1} differ only in r_t i.e., $C_t = C_{t-1} \cup \{r_t\}$.

The scenario where servers are moved only to service requests directly is called *lazy*. The double coverage algorithm in that sense is not lazy. A non-lazy algorithm can however be *memory-less* like the double coverage algorithm since decisions are based only on the current configuration. Let us use potential $\Phi(C_t, C'_t)$ where C stands for the online algorithm and C' for the offline algorithm. Let $cost(t)$ and $opt(t)$ be the costs to service r_t by online and offline methods at the instant t. We need to show that

$$cost(t) - K * opt(t) \leq \Phi(C_{t-1}, C'_{t-1}) - \Phi(C_t, C'_t) \tag{1}$$

Adding for m steps we have

$$\sum_{t=1}^{m} cost(t) - K * \sum_{t=1}^{m} opt(t) \leq \Phi(C_0, C'_0) - \Phi(C_m, C'_m) \tag{2}$$

We can drop $\Phi(C_m, C'_m)$ without disturbing upper bounding so that we have

$$\sum_{t=1}^{m} cost(t) - K * \sum_{t=1}^{m} opt(t) \leq \Phi(C_0, C'_0) \tag{3}$$

which gives the competitive ratio of at most K.

Let us use the offline algorithm to respond to r_t first and then the online algorithm.

1. $C'_{t-1} \leftarrow C'_t$, whereas C_{t-1} is unchanged.

2. $C_{t-1} \leftarrow C_t$ where C'_t has already reached a server to location r_t.

We define the potential function as

$$\Phi(C_t, C'_t) = K * d(C_t, C'_t) + \sum_{a_i, a_j \in C_t} d(a_i, a_j) \tag{4}$$

$d(C_t, C'_t) \leftarrow$ weight of the minimum weight bipartite matching in K_{C_t, C'_t}, the complete bipartite graph where servers of the offline and online algorithm form the two vertex sets C_t and C'_t.

To prove inequality (1) we do the two transitions of [1], the offline algorithm, and then [2], the online algorithm.

$cost(t) - K * opt(t) \leq \Phi(C_{t-1}, C'_{t-1}) - \Phi(C_t, C'_t)$

Wherever $cost(t)$ is more than $K * opt(t)$, there is a *balancing payment* from fall in the potential function. Observe that $d(C_t, C'_t)$ is simply $\sum_{i=1}^{K} d(s_i, a_i)$ for the scenario of straightline geometry. **Offline algorithm movement** of servers for the request r_t. We have the following equations for potential functions for transition [1].

$$\Phi(C_{t-1}, C'_t) = K * d(C_{t-1}, C'_t) + \sum_{a_i, a_j \in C_{t-1}} d(a_i, a_j) \tag{5}$$

$$\Phi(C_{t-1}, C'_{t-1}) = K * d(C_{t-1}, C'_{t-1}) + \sum_{a_i, a_j \in C_{t-1}} d(a_i, a_j) \tag{6}$$

By the definition of Φ in Equation 4 and from Equations 5 and 6 we deduce

$$\Phi(C_{t-1}, C'_t) - \Phi(C_{t-1}, C'_{t-1}) = K * [d(C_{t-1}, C'_t) - d(C_{t-1}, C'_{t-1})] \tag{7}$$

Now by the triangle inequality

$d(C_{t-1}, C'_t) \leq d(C_{t-1}, C'_{t-1}) + d(C'_{t-1}, C'_t)$

and inequality 7 we have

$$\Phi(C_{t-1}, C'_t) \leq \Phi(C_{t-1}, C'_{t-1}) + K * d(C'_{t-1}, C'_t) \tag{8}$$

We will remember equation 8 for future use to prove inequality (1).

Suppose we show for the online movement that

$$\Phi(C_t, C'_t) \leq \Phi(C_{t-1}, C'_t) - d(C_{t-1}, C_t) \tag{9}$$

Combining Equation 8 and 9 we get

$\Phi(C_t, C'_t) + d(C_{t-1}, C_t) \leq \Phi(C_{t-1}, C'_t) \leq \Phi(C_{t-1}, C'_t)$ or

$d(C_{t-1}, C_t) - K * d(C'_{t-1}, C'_t) \leq \Phi(C_{t-1}, C'_{t-1}) - \Phi(C_t, C'_t)$

But $d(C_{t-1}, C_t) = cost(t)$ and $d(C'_{t-1}, C'_t) = opt(t)$. So we have established inequality (1) Therefore, inequality (2) follows and we are done. Finally, to show Equation 9 for movement of online steps, we do as follows.

Let us account the cost of the online algorithm for moving survers at the request r_t. Again, by the definition of Φ, we have $\Phi(C_t, C'_t) = K \times d(C_t, C'_t) + \sum_{a_i, a_j \in C_t} d(a_i, a_j)$ and $\Phi(C_{t-1}, C'_t) = k \times d(C_{t-1}, C'_t) + \sum_{a_i, a_j \in C_{t-1}} d(a_i, a_j)$. Observe that if r_t is a point between two online servers s_i and s_{i+1}, then one of them moves towards its matching point of the offline configuration and the other server may move away from its matching offine server an equal distance. So, their total contribution does not increase the matching, i.e., $d(C_t, C'_t) - d(C_{t-1}, C'_t) \leq 0$. Without loss of generality assume that $d(s_i, r_t) \leq d(s_{i+1}, r_t)$. Since s_i and s_{i+1} move towards r_t by the same distance, $\sum_{a_i, a_j \in C_t} d(a_i, a_j) - \sum_{a_i, a_j \in C_{t-1}} d(a_i, a_j)$ is reduced by $2d(s_i, r_t)$. So, $\Phi(C_t, C'_t) - \Phi(C_{t-1}, C'_t) \leq 2d(s_i, r_t)$, or $\Phi(C_t, C'_t) \leq \Phi(C_{t-1}, C'_t) - 2d(s_i, r_t)$, or $\Phi(C_t, C'_t) \leq \Phi(C_{t-1}, C'_t) - d(C_{t-1}, C_t)$, where $d(C_{t-1}, C_t)$ represents the change in the distance between online servers. This is the very Inequality 9 for this case.

Now consider the other case where r_t lies outside the interval of the K servers, and only one server (say, s_1) moves to r_t. Here, the first term of the potential decreases by $K \times d(s_1, r_t)$, because s_1 moves closer to its matching point a_1. The second term of the potential increases by $(K-1) \times d(s_1, r_t)$ as the distance to s_1 from s_2, s_3, ... , s_k increases by $d(s_1, r_t)$. The difference of these two terms is $d(s_1, r_t)$, which is equal to $d(C_{t-1}, C_t)$. So, in this second case too, $\Phi(C_t, C'_t) \leq \Phi(C_{t-1}, C'_t) - d(C_{t-1}, C_t)$, that is, Inequality 9. This completes the proof.