A Tale of Two Cities

Prof. Yishay Mansour, Prof. Amos Fiat, Mr. Lior Shultz

July 19, 2016

Abstract

We consider the following problem to achieve a basic understanding of taxi driver on-line decisions: We have two cities and jobs arrive one after another and the taxi driver can decide for each job if he wants to pick up the passenger or forfeit the job. Accepting a job yields the driver a reward of 1. However, accepting a job in a different city means that the driver will have to drive to the other city and pay 1 for the relocation - effectively earning nothing for the fare.

1 Notation

We will denote the following:

1. T - The number of total jobs received.
2. L, R - The two cities.
3. $p_{t,L}$ - The probability of the taxi being at city L at time t of our algorithm.
4. $p_{t,R}$ - The probability of the taxi being at city R at time t of our algorithm. We will notice that always $p_{t,L} + p_{t,R} = 1$.
5. A turn is defined by first receiving a job request and then the decision as to how to act upon that request. We will denote the city the driver decided to go to in time t as d_t.
6. We will denote the job received at time t as $J_t \in \{L, R\}$

2 Deterministic Algorithm

Theorem 1. *For any deterministic algorithm there exists a sequence for which the algorithm receives a reward of* 0.

Proof. If we consider ALG at time t we will notice that we can compute d_t since it only depends on $J_1, J_2 \cdots J_{t-1}$ thus we can simply create a new job J_t so that $J_t \neq d_t$. This way at time t the reward will be 0 and this can be done for every t gaining a total of 0 for the entire algorithm. □

Theorem 2. *For every sequence* $J_1, J_2 \cdots J_t$ *we get* $OPT > \frac{T}{2}$.

Proof. Since $|\{t|J_t = L\}| + |\{t|J_t = R\}| = T$ then either $|\{t|J_t = L\}| \geq \frac{1}{2}$ or $|\{t|J_t = R\}| \geq \frac{1}{2}$ and so OPT can just pick the one that is greater and remain there receiving a reward of at least $\frac{T}{2}$

□

These two theorems prove together that no finite competitive ratio can be achieved for any deterministic algorithm.

3 Online simple strategies

We will discuss the following two simple strategies:

1. $STAY$ - At the beginning we will use a coin to decide on a city and simply stay there for the rest of the algorithm. It is easy to see that this algorithm is expected to earn $\frac{T}{2}$.

2. $RAND\frac{1}{2}$ - Every turn of the algorithm if the job is in a different city we will flip a coin and move to that city with probability $\frac{1}{2}$. If the job is in the city we are already in we will of course accept it along with our reward.

4 Rand Analysis

Let's look at a possible way of analyzing the profit of the $RAND\frac{1}{2}$ algorithm. At time t we have a probability of p of being in R at the beginning of the turn and probability of $1 - p$ of being in L at the beginning of the turn. This means that we can expect to earn p and update the probability of being at R at time $t + 1$ to $p + \frac{1}{2}(1 - p) = \frac{1}{2} + \frac{1}{2}p$ and the probability of being at L to $\frac{1}{2}(1-p)$. In the case of consistently alternating jobs (the job sequence consisting of $L, R, L, R, L, \ldots$) we can see that a balance is achieved when $p = \frac{1}{3}$. This can be easily demonstrated by the fact that through simple calculation we can see that the probabilities are swapped after the job arrives at R and thus they can continuously swap and remain balanced that way. This means that in every step we will earn exactly $\frac{1}{3}$ achieving a total expected revenue of $\frac{T}{3}$ compared to the $\frac{T}{2}$ of OPT.