k-Taxi Problem: A Comprehensive Overview

Qianxi (Ashley) Gao

Abstract—This report investigates the k-Taxi Problem, a relatively new and important topic in online algorithms. The author provides a comprehensive overview of the latest findings in the k-Taxi Problem, along with explanations of the terminology. Additionally, the author explores alternative proofs to establish the equivalency between the k-server problem and the easy version of the k-taxi problem. Lastly, several conjectures and open questions related to the k-Taxi Problem are presented.

Index Terms—Online Algorithm, k-Taxi Problem, Competitive Ratio

1 INTRODUCTION

Online problems are an important topic in computational theory, and have been discussed broadly for many years. One of the most important approaches to evaluating the performance of an algorithm on such problems is competitive analysis. The k-Taxi Problem is an interesting problem in this field which is related to many well-studied problems under an online problem context. The Metrical Task Systems(MTS) problem is one of the important online problems that is addressed by Borodin et al. in 1992 [1], and two special cases of MTS is the k-server problem and k-taxi problem. Furthermore, most of the results of the easy version [1] of the k-taxi problem follows by the results of the k-server problem, since the k-server problem could prove the equivalency with the easy version of the k-taxi problem. In this report, the author would like to give a comprehensive overview of the most recent results of the k-taxi problem, with sufficient terminology explanation and possible alternative proof of the equivalency of the k-server problem and the easy k-taxi problem. In the end, some simple conjectures and open questions related to k-taxi problem and online competitive analysis would be given.

1. Where it was called ”easy taxicab” and ”hard taxicab” in [2] and these two versions of problem firstly been addressed by [3]

2 PRELIMINARIES

Competitive Ratio: $\rho_{ALG} = \limsup_{OPT(x)\to\infty} \frac{ALG(x)}{OPT(x)}$

Metric Space: A $\mathcal{M} = (S, d)$ that defined by a set of points S and a distance function d, is called a metric space if it satisfies the listed 4 axioms.

- Positivity: $d(x, y) > 0, \forall x, y \in S, x \neq y$
- Reflexivity: $d(x, x) = 0, \forall x \in S$
- Symmetry: $d(x, y) = d(y, x), \forall x, y \in S$
- Triangle Inequality: $d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in S$

Memoryless: The algorithm is memoryless if and only if the algorithm makes the decision based on the current information (for example, in the case of the k-taxi problem, the current location of the servers), with no prior knowledge about past requests and decisions.

k-taxi problem: There exists k taxis placed on the metric space $\mathcal{M}$. We call the positions of these k taxis as *initial configuration*. Requests sequence, so-called $\sigma = [\sigma_1, \sigma_2, ...]$ is fed into the algorithm one by one. For each request $\sigma_i = (s, t)$, defined the pick-up location and the destination of the request i. Our algorithm needs to make a decision immediately to decide which taxi to serve for the request. After serving a request, the configuration is updated to indicate that the chosen server has moved to the destination of the request.

Easy version of k-taxi problem: Inherit the definition from the previous k-taxi problem (2), the objective for the easy version problem is to minimize the distance of the entire trip that the taxi traveled.

Hard version of k-taxi problem: Inherit the definition from the previous k-taxi problem (2), the objective for the hard version problem is to minimize the distance of the empty runs of a taxi, i.e. the distance between the original position of the taxi to s, when $\sigma_i = (s, t)$.

Layered Graph Traversal Problem: This problem is first introduced in [4]. [2] A layered graph is a graph such that all the nodes could be partitioned into layers, and edges only exist between the adjacent layers. The first and the last layer only contain 1 node [3]. For this problem, the next layer and connected edges are revealed each run by the adversary, and the player needs to make moves to the next layer. A backward move is allowed. The competitive ratio is defined as $\rho(ALG) = \frac{\text{cost of the player}}{OPT}$.

Hierarchically Well-separated Trees(HSTs): HST is a powerful idea that connects the optimization problem on general metric space to the same optimization problem on HST. Bartal states the result in [5] that any metric space could be transformed to HST by randomization approach with the distortion factor $O(\log(n))$. Bartal [6] states that: if there exists any (deterministic or randomized) algorithm for an optimization problem $\mathcal{P}$(with linear objective function) on HST with the competitive ratio ρ, and we can embedding the metric space $\mathcal{M}$ into HSTs with a distortion α, then there exists a randomized algorithm with the competitive ratio $\alpha\rho$ for the original optimization problem on $\mathcal{M}$.
There are also some restrictions for HST. First, it only guarantees a randomized algorithm for the optimization problem. Secondly, the objective function must be a linear combination of distance, otherwise, we cannot conclude anything from HST. Thirdly, even though the HST itself does not support for embedding a continuous infinite metric space, we could still embed the metric space into HST by discretization. The distortion factor used to have a gap between $\log(n)$ and $\log^2(n)$, but has been closed by Fakcharoenphol, Rao and Talwar in [7] in 2004.

k-Server Problem: The author would interpret this problem as a special case of the k-taxi problem. Inheriting the notation from k-taxi problem. If $\forall \sigma_i = (s, t)$: $s = t$, then this is a k-server problem.

2. Which is known equivalent to the Metrical Service System.

3. The author would argue this requirement can be waved, because if we consider the second layer in the original notion as the "first layer" in our new notation, we can just allow the player to choose the start point from the "first layer" – by the way, some of the vertexes may be unavailable to choose since there is no edge between the first and second layer of the original version of the graph. Similar reasoning could be done for deleting the last layer with one node.

3 Alternative Proof

3.1 Proof for the equivalence of easy-taxi problem and k-Server Problem

In [8], Coester and Koutsoupias prove that the easy k-taxi problem is equivalent to the k-server problem by extending the matric space to a generalized one. Therefore, we can adapt the competitive ratio from k-server problem to the easy k-taxi problem.
After considering this problem, I would like to show the reader a different approach to prove the equivalence of the easy k-taxi problem and k-server problem, from an online competitive ratio aspect.
Let's construct the input sequence from the adversaries' aspect. Note that, this proof works for all kinds of adversaries(oblivious adversary, adaptive online adversary, adaptive offline adversary).

Proof.
Assumption:

- The adversary wants the competitive ratio to be as worse as possible.
- $\mathcal{M}$ is the metric space for this problem.
- C is the initial configuration for taxis.

- ALG denotes an arbitrary algorithm, no matter whether it is a deterministic or randomized one.
- σ is the sequence generated by the adversary.

Claim 1: The adversary always generates a request (s, s), $s \in \mathcal{M}$, (i.e. for a request $\sigma_i = (s, t)$, $s = t$). Otherwise, our algorithm would get a better competitive ratio.

Proof of Claim 1.
Let ALG denote the cost for ALG, and OPT denotes the optimal solution for this request.
For any request $\sigma_i = (s, t)$, we would have to move a taxi to serve. Assume that g_i and g_{OPTi} is the location for the taxi serving this request, let us break this route into two parts: $ALG_{\sigma_i} = dist(g_i, s) + dist(s, t)$, $OPT_{\sigma_i} = dist(g_{OPTi}, s) + dist(s, t)$
Then the competitive ratio in this case is

$$\frac{ALG(\sigma)}{OPT(\sigma)} = \frac{\sum dist(g_i, s) + \sum dist(s, t)}{\sum dist(g_{OPTi}, s) + \sum dist(s, t)}$$

$$\frac{a+c}{b+c} \leq \frac{a}{b}, \forall a, b, c \in \mathbb{R}^+ \cup \{0\}$$

$$\text{holds iff } a \geq b$$

Since

$$ALG(\sigma) \geq OPT(\sigma)$$

Then

$$\sum dist(g_i, s) \geq \sum dist(g_{OPTi}, s)$$

With the fact that $dist \geq 0$ by the axiom of the metric space, $c = \sum dist(s, t) \geq 0$. Then we know that if the adversary wants the worst competitive ratio, the adversary would better to set $c = 0$ to get $\frac{a}{b}$, that is the upper bound of the competitive ratio. To be more specific, the adversary set $\sum dist(s, t) = 0$, in order to get

$$\frac{ALG(\sigma)}{OPT(\sigma)} = \frac{\sum dist(g_i, s)}{\sum dist(g_{OPTi}, s)}$$

which is the worst competitive ratio that possibly achieves. That is, an adversary would like to set $s = t$ for any request of the request sequence σ, which match the definition of the k-server problem. Therefore, the easy k-taxi problem would share the same result as the k-server problem. $\square$

Side notes: I found that my proof is much more easier than the proof in [8], I suppose there are some unaware gaps in my proof. I would really appreciate any suggestion that comes from the readers.

4 Related Work

For the easy k-taxi problem, [8] shows the equivalence of the k-taxi problem and k-server problem. Therefore, the result of the deterministic competitive ratio of k-server problem from [9] [10], which is $2k - 1$, could be inherited to the easy k-taxi problem[4]. It would be k if and only if the k-server conjecture holds.
In terms of the lower bound related to the easy k-taxi problem, [9] gives a proof and result for the deterministic case, which is $\Omega(k)$. For the randomized case, recent result in [11] updates the negative result of randomized k-server problem of any metric with $k + 1$ points from $\Omega(\log(n))$ to $\Omega(\log^2(n))$, refute the previous conjecture that the lower bound for any metric is $\Omega(\log(n))$. By the equivalence of k-server and easy k-taxi problem proved in [8], this lower bound naturally carry over to the easy k-taxi problem. Moreover, the theorem in [12] states, for online problems against the adaptive online adversary, the randomized competitive ratio at most square root is better than the deterministic competitive ratio, establish the relationship between the randomized competitive ratio and deterministic competitive ratio for the same problem, which may be useful on proving further lower bound.
In [13], Dehghani et al. name the easy k-taxi problem as Uber problem, and propose some results for the stochastic input. This is more related to the real-life uber/scheduling scenario. Because at most of the time, an adversary is unavailable. They present that for any k-server(uber problem) on a line, there exists a 3-approximation of the online algorithm. Also, in a stochastic manner, [13] also give a result

4. Specifically, work function algorithm in [9] is one of the well-known algorithms that can achieve this bound

for general metric space with n points, which is $O(\log n)$.
However, in an adversarial manner, we have a competitive ratio $2k-1$, which is good enough due to n could be really large and do not depend on k. This $O(\log n)$ result is useful only if n is less than k^2.

For the hard k-taxi problem, the results state here would be grouped by the category of the algorithm(i.e. Deterministic or Randomized).
For the deterministic algorithm on the hard k-taxi problem, [8] have proved a lower bound on a deterministic algorithm, which is $\Omega(2^k)$ by the reduction of the layered graph traversal problem.
For the randomized algorithm, [8] shows a tight bound on HST for a randomized algorithm against the adaptive online adversary. [14] comes up with the probabilistic approximation of n-points metrics by HST with distortion $O(\log n)$. Therefore, [8] states that there exists a $O(2^k \log n)$ randomized algorithm for hard k-taxi problem. (By the theorem states in Section 2, Hierarchically Well-separated Tree).
Coester and Koutsoupias also provide some special results for this hard k-taxi problem in [8]. They propose an algorithm called `BIASEDDC`, which can solve the hard 2-taxi problem in general metric space with a competitive ratio of 9. Meanwhile, an algorithm for 3-taxi problem on a line, called `REGIONTRACKER` achieves a constant competitive ratio. This hard 3-taxi problem can be transformed to other scheduling problems, e.g. three elevator scheduling, so the result and algorithm they give are useful in terms of real-life applications.

5 Open Questions and Conjectures

For hard k-taxi problem, there could be a study on improving the lower bound from $\Omega(2^k)$ in [8] to a tighten lower bound $\Omega(2^k \log n)$. Since intuitively, a deterministic algorithm can not do better than a randomized algorithm. Therefore, the lower bound may be able to bump up to $\Omega(2^k \log n)$ for deterministic algorithms in this scenario.
In terms of the easy k-taxi problem, currently, the lower bound for the deterministic algorithm is $\Omega(k)$([9]), while for the randomized algorithm is $\Omega(\log^2(n))$. The author would argue that further work on improving the bound for the deterministic algorithm is possible, based on the lower bound result of the randomized algorithm for the easy k-taxi problem.
There could also be some study on this problem if we slightly change some conditions. For example, the metric space with a relaxed triangle inequality, or discuss results on an infinite metric space (i.e. $n \to \infty$), etc.
The research based on the stochastic input or input with a specific distribution/feature would also be an interesting question to discuss. Even though it might be slightly off-topic of computational theory, in the case of the taxi problem, besides the algorithms discussed above, we can always learn the past data and improve our algorithm performance over time through reinforcement learning. Even though memory probably is not always helpful in a competitive analysis context, it helps in other aspects.

Acknowledgments

I would like to show my appreciation and thanks to Professor Allan Borodin for his support and expert supervision throughout this term.

References

[1] A. Borodin, N. Linial, and M. E. Saks, "An optimal on-line algorithm for metrical task system," *J. ACM*, vol. 39, no. 4, p. 745–763, oct 1992. [Online]. Available: https://doi.org/10.1145/146585.146588

[2] A. P. Kosoresow, "Design and analysis of online algorithms for mobile server applications," 1996.

[3] A. Fiat, Y. Rabani, and Y. Ravid, "Competitive k-server algorithms," in *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, 1990, pp. 454–463 vol.2.

[4] C. H. Papadimitriou and M. Yannakakis, "Shortest paths without a map," *Theoretical Computer Science*, vol. 84, no. 1, pp. 127–150, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304397591902632

[5] Y. Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications," in *Proceedings of 37th Conference on Foundations of Computer Science*, 1996, pp. 184–193.

[6] ——, "Distributed paging," in *Online Algorithms, The State of the Art (the book grow out of a Dagstuhl Seminar, June 1996)*, ser. Lecture Notes in Computer Science, A. Fiat and G. J. Woeginger, Eds., vol. 1442. Springer, 1996, pp. 97–117. [Online]. Available: https://doi.org/10.1007/BFb0029566

[7] J. Fakcharoenphol, S. Rao, and K. Talwar, "A tight bound on approximating arbitrary metrics by tree metrics," *Journal of Computer and System Sciences*, vol. 69, no. 3, pp. 485–497, 2004, special Issue on STOC 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000004000637

[8] C. Coester and E. Koutsoupias, "The online k-taxi problem," *CoRR*, vol. abs/1807.06645, 2018. [Online]. Available: http://arxiv.org/abs/1807.06645

[9] A. Borodin and D. Pankratov, *Online and Other Myopic Algorithms(Draft)*, 2023.

[10] E. Koutsoupias, "The k-server problem," *Computer Science Review*, vol. 3, no. 2, pp. 105–118, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013709000197

[11] S. Bubeck, C. Coester, and Y. Rabani, "The randomized k-server conjecture is false!" 2022.

[12] S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Wigderson, "On the power of randomization in on-line algorithms." *Algorithmica*, vol. 11, pp. 2–14, 01 1994.

[13] S. Dehghani, S. Ehsani, M. Hajiaghayi, V. Liaghat, and S. Seddighin, "Stochastic k-server: How should uber work?" *CoRR*, vol. abs/1705.05755, 2017. [Online]. Available: http://arxiv.org/abs/1705.05755

[14] J. Fakcharoenphol, S. Rao, and K. Talwar, "A tight bound on approximating arbitrary metrics by tree metrics," in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 448–455. [Online]. Available: https://doi.org/10.1145/780542.780608