

RaoTA algorithm for Drive the TAs Home (DTH)

Kevin An, Matthew Tran, Joe Zou

December 2019

1 Introduction

This article discusses approximation algorithms we developed for a NP-complete problem ("Drive the TAs Home", or DTH) given to us in CS170 the fall of 2019 at UC Berkeley taught by professors Satish Rao and Prasad Raghavendra. The problem statement, restated below, can be found at <https://cs170.org/assets/project/spec.pdf>. Our code itself can be found at <https://github.com/dragonlock2/RaoTA>.

2 Problem Statement

Intuitively, say Prof. Rao is at Soda Hall, and has a bunch of TAs with him. All the TAs have to walk home, and costs them each energy equal to the distance they walk if they have to walk some distance d . However, Prof. Rao has a car, so he takes energy $\epsilon \in (0, 1)$ times the amount of energy it takes for a TA to travel the same distance. Rao can take an arbitrary number of TAs in his car, and can drop off TAs at some location and have each TA walk the rest. Note that if $\epsilon \leq \frac{1}{2}$, it is always optimal to simply drop everyone off at their respective homes, and thus just solving the Travelling Salesman problem (TSP) on the homes would be optimal, so Rao would visit each home with the least energy (and the TAs spend none).

Suppose we are given a connected, undirected graph $G(V, E)$ with edges weighted by a distance function $d : |E| \rightarrow \mathbb{R}_{\geq 0}$. If G is not complete, we can extend d to $|V| \times |V|$ such that $d(u, v) = sp(u, v)$ where $sp(u, v)$ with $u, v \in V$ is the weight of the shortest path in G between u and v . We additionally require that d satisfies the triangle inequality. We have a starting point $s \in V$, and then some $H \subset V$ of homes, each associated to a TA. We assume the homes are in bijection with the set of TAs (call them I), and index each house by $h_i \in H$ for each $i \in I$ to mean h_i is TA i 's house. Let $\epsilon \in (\frac{1}{2}, 1)$. We wish to find a sequence (a_1, a_2, \dots, a_n) with $a_j \in V$ and $a_n = s$ of (not necessarily distinct) *dropoff points* and a bijective function $f : \{1, \dots, n\} \rightarrow I$ (which represents which TA is dropped off at which dropoff point) such that

$$\sum_{j=1}^n (\epsilon * d(a_j, a_{j+1}) + d(a_j, h_{f(j)}))$$

or the total energy cost of everyone's travelling, is minimized.

Theorem 2.1 (NP-Completeness). *Drive the TAs Home is NP-complete.*

Proof. Clearly, DTH is in NP. We now give a reduction from TSP to DTH. Given a graph $G(V, E)$, the reduction of TSP to DTH can be done by making every vertex a home, and then setting $0 < \epsilon < \frac{1}{2}$. Alternatively, set $\epsilon \in (\frac{1}{2}, 1)$ for every vertex v add two homes h_v and h'_v each with only an edge to v . Then, each vertex must be a dropoff point and so Rao

must visit every vertex. Since Rao's energy cost is also minimized, Rao's path here gives a solution to TSP, and hence DTH is NP-complete. \square

3 Algorithm: Optimized TSP, or Travelling Rao's Men

Theorem 3.1 (TSP Approximation). *Solving the Travelling Salesman Problem (TSP) starting on the starting point s through the homes and then having Rao follow this path and drop off each TA at their respective homes gives a 2ϵ approximation of DTH.*

Proof. Suppose TA i is dropped off at $v_i \in V$, and $v_i \neq h_i$ in the optimal solution. Since i has to now walk home from v_i , that walking gives a cost of $d(v_i, h_i)$. However, if instead Rao just drove i and dropped them off at home starting from v_i , it would cost him $2\epsilon * d(v_i, h_i)$ to drive to h_i and then come back to v_i . Repeating this argument for all i , we get a 2ϵ approximation, and the cost of TSPing over the homes would be less equal this cost, since the original dropoff points in the optimal need not be visited. \square

Based on the above theorem, we developed the following approximation algorithm for DTH. Suppose we are given an algorithm $TSP(G, V, E)$ that solves the TSP on G . We then have:

Algorithm 1: Optimized TSP (optiTSP)

Result: Approximation for DTH
initialization;
