

The Computational Complexity of the Traveling Salesman Problem

Craig Alan Feinstein

Received: 9 November 2011 Accepted: 6 December 2011 Published: 17 December 2011

Abstract

In this note, we show that the Traveling Salesman Problem cannot be solved in polynomial-time on a classical computer.

Index terms— Traveling, Problem, polynomial, classical.

Traveling Salesman Problem -A traveling salesman starts at city 1, travels to cities 2, . . . , n in any order that the salesman chooses, and then ends his trip in city n. Let us denote (i, j) to be the distance from city i to city j . The goal of the Traveling Salesman Problem is to find the minimum total distance possible for the traveling salesman to travel. There are no restrictions on the possible distances (i, j) between each of the cities other than the requirement that each (i, j) is a positive integer and $(i, j) = (j, i)$.

We give a simple proof that no deterministic Problem in $O(2^n)$ time:

For any nonempty subset $S \subseteq \{2, \dots, n\}$ and for any city $i \in S$, let us define (S, i) to be the length of the shortest path that starts at city 1, visits all cities in the set S , and ends at city i . The Traveling Salesman Problem is equivalent to the problem of computing $(\{2, \dots, n\}, n)$. Clearly, $(\{i\}, i) = (1, i)$ and when S is a singleton, this recursive formula cannot be simplified, so the fastest way to compute $(\{2, \dots, n\}, n)$ is to apply this recursive formula to $(\{2, \dots, n\}, n)$. Since this subsets $S \subseteq \{2, \dots, n\}$ and each $i \in S$, we obtain a lower bound on the Traveling Salesman Problem.

This lower bound is confirmed by the fact that the fastest known deterministic and exact algorithm which solves the Traveling Salesman Problem was first time of $O(2^n)$.

