**Design Doc: Initial Report**

**1 Initial Approaches & Algorithms**

**1.1 Dynamic Programming**

**2.3 Linear Programming**

Initially, we thought of using linear programming so as to minimize the cost function:

A close up of a logo

Description automatically generated

However, we soon realized that there would be too many constraints and variables to account for.

**2 Final Approach**

We are going to first try the implement the weaker solution of the Traveling Salesman Problem (TSP) that achieves a factor of 2 times the optimal. As TSP is usually solved on a complete graph, when parsing through the input file whenever we see an ‘x’ (where no edge exists between the two vertices) (i=j?) then we will assign it an arbitrarily large value to complete the graph so as to not affect the optimal cycle. We will be constructing a Minimum Spanning Tree (MST) that includes all the TA’s homes and the starting location. After building this weaker approach we will then look at implementing Christofide’s Algorithm so as to improve the factor to 1.5 times the optimal solution.

sys.maxsize