Oblig 1 The Traveling Salesman Problem

INF4490

Joseph Knutson
`github.com/mathhat`

September 10, 2017

Contents

1 Introduction **2**

2 Exhaustive search **2**

2.1 Creating the Code 2

2.2 timetable 2

1 Introduction

2 Exhaustive search

2.1 Creating the Code

Making my Exhaustive Search code began with using the example.py file on the course site which imports the city grid. From there I followed the advice of the assignment regarding the itertools module's permutations function. Looping over every sequence and summing the distances for each sequence, the final Exhaustive search function looked something like this:

```python
for sequence in Permutations:   #exhaustive search begins
        dist = 0
        for index in range(cities-1):
            i +=1
            dist += distances[sequence[index]][sequence[index+1]]
        dist += distances[sequence[cities -1]][sequence[0]]
        i+=1
        if dist < best:                 #save shortest distance yet
            best=dist
            best_sequence = sequence

    end = time.time()                  #end clock
    Time = (end-start)                 #sum time
    return(best,best_sequence, Time)
```

2.2 timetable