/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

User's Manual

@author: Andrew Haynes  
Project Name: Brute Force and Genetic Algorithm (Approximation Algorithm)

Files

Genetic.h  
BruteForce.h  
Genetic.c  
BruteForce.c  
main.c  
cityweights.txt

1. Installation (Make sure you have GCC installed!)

1. Create a folder and name it whatever you want.

2. Inside the folder, please include “genetic.c”, “genetic.h”, “bruteforce.h”, “bruteforce.c”, “main.c”, and “cityweights.txt”.

3. Go on the command line and change directories to where you created your folder. If you forgot how to navigate your directory:   
 On the command line: “cd <directoryname>” will take you into specified directory  
 “ls” will list all of the files in your current directory  
 “cd ..” will take you back one directory  
  
4. Once you reach the directory you created with the file included in it, you can compile the program by typing in “make”. The output of this command is an executable file called “TSP”. The reason for the name is what this program was made for, the Traveling Salesman Problem!

5. Type into the command line “ls” to make sure the file was made. You should see a highlighted word that states “TSP”. This is the object file! You have created the program. To run the program, type “./TSP”.

1. How It Works  
     
   1. The program is ran by prompts and it will explain to you where the numbers are going and what they are going to be, the variables per se. For the brute force algorithm, you will insert the number of cities, and the start city to start the tour from. For the genetic algorithm, you will need the number of cities, the start city the tour will start from, the number of individuals per generation, the number of generations to execute (this program terminates once it reaches this number!), and the mutation rate for each generation.

2. The output of the program depends on the choice you made in the beginning. If you selected one, only the output for Brute Force will show. It will show the optimal path and the fitness score of it. For choice two, it will show the genetic algorithm output which is the two optimal paths, and the fitness score for each path.

1. Experiment; The Comparison between Brute Force and Approximation

First Run:

BRUTE FORCE:

Number of Cities: 8

Start City: 3

OPTIMAL PATH: [3] [5] [6] [4] [2] [7] [1] [0] [3]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 8

Start City: 3

Number of individuals: 6

Number of generations: 200

Mutation Rate: 0.500000

OPTIMAL PATH [elite one]: [3] [5] [6] [2] [4] [7] [1] [0] [3]

FITNESS SCORE: [2047.904679]

OPTIMAL PATH [elite two]: [3] [5] [6] [2] [4] [0] [1] [7] [3]

FITNESS SCORE: [2146.561443]

Seconds Elapsed: 0 Microseconds elapsed: 2999

Second Run:

BRUTE FORCE:

Number of Cities: 8

Start City: 5

OPTIMAL PATH: [5] [3] [0] [1] [7] [2] [4] [6] [5]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 8

Start City: 5

Number of individuals: 5

Number of generations: 500

Mutation Rate: 0.800000

OPTIMAL PATH [elite one]: [5] [1] [0] [3] [7] [2] [4] [6] [5]

FITNESS SCORE: [2091.641689]

OPTIMAL PATH [elite two]: [5] [3] [0] [1] [7] [6] [2] [4] [5]

FITNESS SCORE: [2178.586719]

Seconds Elapsed: 0 Microseconds elapsed: 7275

Third Run:

BRUTE FORCE:

Number of Cities: 8

Start City: 3

OPTIMAL PATH: [3] [5] [6] [4] [2] [7] [1] [0] [3]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 8

Start City: 3

Number of individuals: 6

Number of generations: 1000

Mutation Rate: 0.900000

OPTIMAL PATH [elite one]: [3] [5] [6] [4] [2] [7] [1] [0] [3]

FITNESS SCORE: [1867.005669]

OPTIMAL PATH [elite two]: [3] [5] [1] [0] [6] [4] [2] [7] [3]

FITNESS SCORE: [2047.280633]

Seconds Elapsed: 0 Microseconds elapsed: 17498

Now we spike up the ante and see the brute force actually slowing down ::

Fourth Run:

BRUTE FORCE:

Number of Cities: 9

Start City: 6

OPTIMAL PATH: [6] [5] [3] [0] [1] [7] [8] [4] [2] [6]

FITNESS SCORE: [2045.139029]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 9

Start City: 6

Number of individuals: 4

Number of generations: 450

Mutation Rate: 0.800000

OPTIMAL PATH [elite one]: [6] [5] [3] [0] [1] [4] [8] [7] [2] [6]

FITNESS SCORE: [2580.137549]

OPTIMAL PATH [elite two]: [6] [2] [4] [8] [3] [5] [1] [7] [0] [6]

FITNESS SCORE: [2769.960180]

Seconds Elapsed: 0 Microseconds elapsed: 2871

Fifth Run:

BRUTE FORCE:

Number of Cities: 10

Start City: 2

OPTIMAL PATH: [2] [9] [4] [8] [7] [1] [0] [3] [5] [6] [2]

FITNESS SCORE: [1955.221546]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 10

Start City: 2

Number of individuals: 8

Number of generations: 600

Mutation Rate: 0.300000

OPTIMAL PATH [elite one]: [2] [5] [3] [0] [1] [7] [8] [4] [9] [6] [2]

FITNESS SCORE: [2557.292576]

OPTIMAL PATH [elite two]: [2] [9] [4] [6] [0] [1] [7] [3] [8] [5] [2]

FITNESS SCORE: [2873.870760]

Seconds Elapsed: 0 Microseconds elapsed: 6722

Sixth Run: (Now we see brute force search taking its toll!)

BRUTE FORCE:

Number of Cities: 11

Start City: 4

OPTIMAL PATH: [4] [9] [2] [6] [5] [3] [0] [1] [7] [10] [8] [4]

FITNESS SCORE: [1999.317716]

Seconds Elapsed: 1

GENETIC ALGORITHM:

Number of Cities: 11

Start City: 4

Number of individuals: 7

Number of generations: 800

Mutation Rate: 0.700000

OPTIMAL PATH [elite one]: [4] [2] [6] [10] [8] [3] [5] [1] [0] [7] [9] [4]

FITNESS SCORE: [3036.522530]

OPTIMAL PATH [elite two]: [4] [9] [1] [7] [8] [10] [0] [3] [5] [6] [2] [4]

FITNESS SCORE: [3076.413273]

Seconds Elapsed: 0 Microseconds elapsed: 11075

Seventh Run: (Now brute force is starting to spike up "exponentially")

BRUTE FORCE:

Number of Cities: 12

Start City: 4

OPTIMAL PATH: [4] [9] [2] [6] [5] [3] [0] [1] [7] [10] [11] [8] [4]

FITNESS SCORE: [1925.813024]

Seconds Elapsed: 19

GENETIC ALGORITHM:

Number of Cities: 12

Start City: 4

Number of individuals: 9

Number of generations: 1100

Mutation Rate: 0.700000

OPTIMAL PATH [elite one]: [4] [8] [11] [6] [0] [1] [10] [7] [3] [5] [2] [9] [4]

FITNESS SCORE: [3144.944179]

OPTIMAL PATH [elite two]: [4] [9] [6] [2] [1] [0] [3] [5] [7] [11] [8] [10] [4]

FITNESS SCORE: [3244.063654]

Seconds Elapsed: 0 Microseconds elapsed: 21696

Eigth Run: (Now brute force search is really taking a hurting!)

BRUTE FORCE:

Number of Cities: 13

Start City: 2

OPTIMAL PATH: [2] [9] [4] [12] [7] [10] [11] [8] [3] [5] [1] [0] [6] [2]

FITNESS SCORE: [1954.109068]

Seconds Elapsed: 237

GENETIC ALGORITHM:

Number of Cities: 13

Start City: 2

Number of individuals: 10

Number of generations: 1500

Mutation Rate: 0.900000

OPTIMAL PATH [elite one]: [2] [12] [1] [0] [5] [3] [8] [11] [10] [7] [6] [9] [4] [2]

FITNESS SCORE: [3049.856299]

OPTIMAL PATH [elite two]: [2] [6] [10] [1] [7] [8] [11] [9] [4] [12] [0] [3] [5] [2]

FITNESS SCORE: [3396.203511]

Seconds Elapsed: 0 Microseconds elapsed: 36315

Ninth Run: (It went past the five minute marker here for the brute force)

For the brute force, I put in 14 cities and I decided to kill the process because it went past

25+ minutes.

" logarithm@ubuntu:~$ ps 2685

PID TTY STAT TIME COMMAND

2685 pts/1 R+ 27:18 ./TSP "

GENETIC ALGORITHM:

Number of Cities: 14

Start City: 0

Number of individuals: 20

Number of generations: 50000

Mutation Rate: 0.900000

OPTIMAL PATH [elite one]: [0] [12] [7] [10] [6] [2] [9] [4] [13] [3] [8] [11] [5] [1] [0]

FITNESS SCORE: [3062.982309]

OPTIMAL PATH [elite two]: [0] [1] [11] [10] [7] [12] [9] [4] [8] [13] [2] [6] [5] [3] [0]

FITNESS SCORE: [3078.412446]

Seconds Elapsed: 3 Microseconds elapsed: -293500

Tenth Run: Proof the GA works and sometimes find an optimal route.

BRUTE FORCE:

Number of Cities: 8

Start City: 5

OPTIMAL PATH: [5] [3] [0] [1] [7] [2] [4] [6] [5]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0

GENETIC ALGORITHM:

Number of Cities: 8

Start City: 5

Number of individuals: 10

Number of generations: 10000

Mutation Rate: 0.800000

OPTIMAL PATH [elite one]: [5] [3] [0] [1] [7] [2] [4] [6] [5]

FITNESS SCORE: [1867.005669]

OPTIMAL PATH [elite two]: [5] [6] [4] [2] [7] [1] [0] [3] [5]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0 Microseconds elapsed: 157589

**Analysis:** The brute force search does indeed grow exponentially in time to do the permutation once we get pass 13 cities. It fairly evident when I ran this at 14 cities that my computer was going to take quite a long time to do 14 cities, let alone 15 cities. The genetic algorithm works beautifully if you create a lot of individuals per generation and allow it to generate thousands of times. Sometimes it was approximate if I used a lot of individuals, but small generation amount, and other times the opposite work so it is hard to tune this perfectly for each city, or any other application that this local search approximation algorithm can be applied to.

**Table of Time Results**(sample data, not all variables are in play in this table, it becomes apparent at city 13 when the difference comes in)

|  |  |  |
| --- | --- | --- |
| Cities | Brute Force | Genetic |
| 8 | 3899 ms | 2999 ms |
| 9 | 27552 ms | 2871 ms |
| 10 | 157931 ms | 6722 ms |
| 11 | 2 seconds | 11075 ms |
| 12 | 19 seconds | 21969 ms |
| 13 | 3 minutes and 55 seconds | 36315 ms |
| 14 | 27+ minutes | 3 seconds |

**More Results to show that Brute Force does indeed grow in time complexity exponentially…**  
  
Number of Cities: 8

Start City: 0

OPTIMAL PATH: [0] [3] [5] [6] [4] [2] [7] [1] [0]

FITNESS SCORE: [1867.005669]

Seconds Elapsed: 0

Microseconds Elapsed: 3889

BRUTE FORCE:

Number of Cities: 9

Start City: 0

OPTIMAL PATH: [0] [3] [5] [6] [2] [4] [8] [7] [1] [0]

FITNESS SCORE: [2045.139029]

Seconds Elapsed: 0

Microseconds Elapsed: 27552

How many cities are in the current run?

10

Pick your starting city (Must be less than number of cities)!

0

BRUTE FORCE:

Number of Cities: 10

Start City: 0

OPTIMAL PATH: [0] [3] [5] [6] [2] [9] [4] [8] [7] [1] [0]

FITNESS SCORE: [1955.221546]

Seconds Elapsed: 0

Microseconds Elapsed: 157931