Technical Documentation

NAVIGATE THE MARS ROVER

Problem Statement:

- Help the Mars Curiosity Rover find the shortest path between points.
- Take a step closer to the goal of establishing a permanent human settlement on Mars.

Name of Web Application: PERSEVERANCE

Technologies used: HTML, CSS, JAVASCRIPT mainly uses p5.js library functions, Github and Github pages.

Aim:
The application aims to find the shortest path that visits all the given points on the grid marked by the user using a genetic algorithm.
The problem this application aims to solve is a variant of the famous Travelling Salesman Problem.

Theme Relevance:

The application helps in navigation of the rover to visit all the “must visit” locations on the map following the shortest path and thus saving on the valuable fuel.

For eg:
The rover is designed to collect the soil samples from the particular destinations after it lands. Having fuel constraints the rover must collect the samples by visiting all the locations marked using the shortest path possible.
The application finds the solution to the above-mentioned problem.

Overview of Algorithmic solution provided:

- **Travelling salesman Problem:**
    1. The travelling salesman problem asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"
    2. It is an NP-hard problem in combinatorial optimization, important in theoretical computer science and operations research.

The application includes the Genetic Algorithm to solve the above-mentioned problem.

- **Brute Force:**
  Generate all n! Combinations of paths and find the one with the shortest one.
  This method is defined in file brute.js

- **Genetic Algorithm**

Genetic algorithms are heuristic search algorithms inspired by the process that supports the evolution of life. The algorithm is designed to replicate the natural selection process to carry generation, i.e. survival of the fittest of beings. Standard genetic algorithms are divided into five phases which are:

1. Creating an initial population.
2. Calculating fitness.
3. Selecting the best genes.
4. Crossing over.
5. Mutating to introduce variations.

This algorithm can be implemented to find a solution to the optimization problems of various types. One such problem is the Traveling Salesman Problem.

Approach:

In the following implementation, locations are taken as genes, string generated using these characters is called a chromosome, while a fitness score which is equal to the path length of all the locations mentioned, is used to target a population.

Fitness Score is defined as the length of the path described by the gene. Lesser the path length fitter is the gene. The fittest of all the genes in the gene pool survive the population test and move to the next iteration.

Algorithm:

1. Initialize the population randomly.

2. Determine the fitness of the chromosome.

3. Until done repeat:

   3.1 Select parents.

3.2 Perform crossover and mutation.

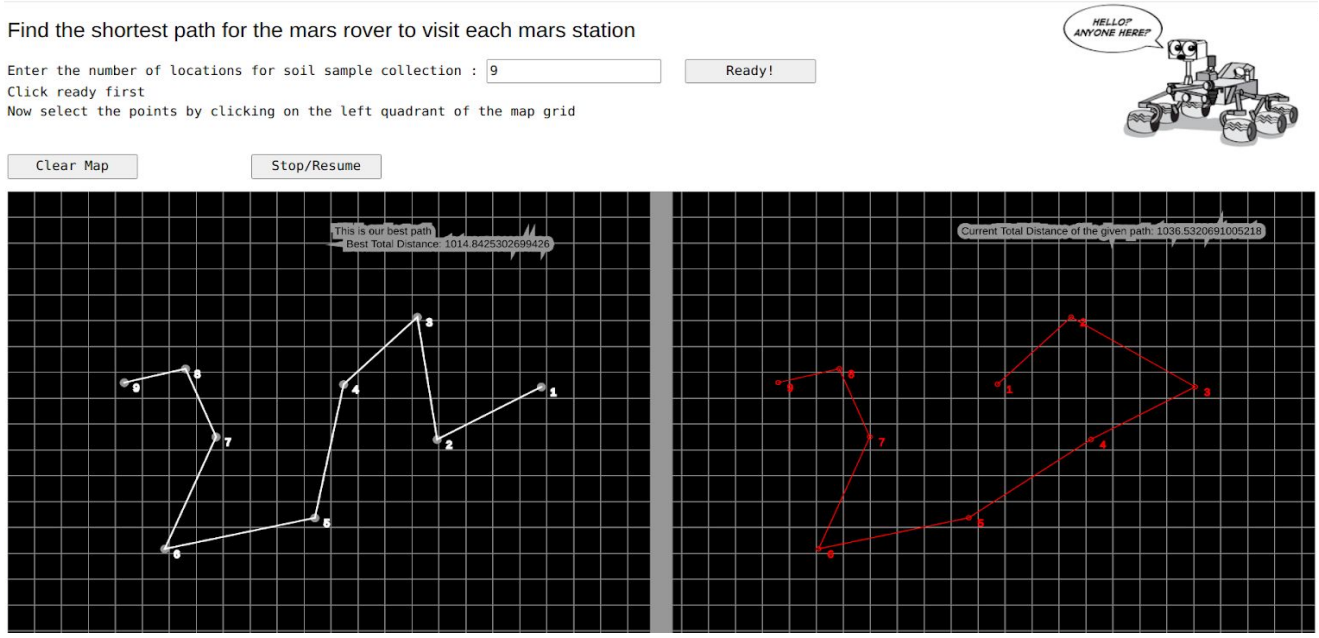
3.3 Calculate the fitness of the new population

4. Append it to the gene pool.

Instructions:

1. Enter the number of locations and then click ready
2. On the map grid of the left side of the screen click the locations needed to visit
3. The left screen grid displays the best path until the given time and the right part of the grid displays the possible paths.
4. Click stop to stop the path generation and the same button to resume the process.
5. Click ‘Clear the map’ button to clear the grid.

The path is labelled from 1-number of locations indicating the path to be followed. For a larger number of locations, the path gets updated with a better path than the previous one.



Shortest optimised path Possible path

References:

http://www.theprojectspot.com/tutorial-post/applying-a-genetic-algorithm-to-the-travelling-salesman-problem/5

https://www.hindawi.com/journals/tswj/2014/178621/