Route Planning based on Traffic Noise using Genetic and A* Algorithm

April 2023

***Abstract*—Sound is a decision variable not considered in the current route-finding systems. Studies have shown that noise directly affects the lives of millions of people. Stress-related illnesses, high blood pressure, hearing loss, sleep disruption, and lost productivity are some noise-related problems. So, the noise level is used as the path weighting factor, and routes with a minimum ratio of distance and noise are obtained using Genetic and A* algorithms.**

Keywords—Noise level, Genetic algorithm, A* algorithm

I. INTRODUCTION AND BACKGROUND

Noise pollution is one of the main stressors in urban environments, negatively impacting people's lives. Here, we attempt to find an optimal route based on noise and distance. This is done by considering noise level as the path weighting factor in various algorithms of optimal route finding. We focus on two algorithms of route finding-the genetic algorithm and the A* algorithm.

Firstly, data collection is an integral part of this project. Noise data can be collected via various means. Here the noise data is collected using a noise recorder device that records the noise data in multiple formats. Noise levels are then incorporated into the Genetic and A* star algorithm by transforming in the time-frequency domain. These noise levels are used as a path-weighting factor in these algorithms.

Genetic Algorithm- Genetic Algorithms (GAs) are a type of local search algorithm which is an abstraction of real biological evolution. The genetic algorithm starts with the initial population and is based on the idea of "survival of the fittest", which means that those species that adapt to the changes in the environment are likely to survive and reproduce the next generation. Here, each generation consists of a population of individuals, and each individual represents a point in search space and a possible solution. At each generation, the fitness value of an individual is calculated using an appropriate fitness function. The next step in this algorithm after fitness calculation is the selection of the fittest individual. Selection is directly proportional to the fitness value. Then crossover operation is performed on these individuals. The genes at the randomly chosen crossover sites are exchanged, thus creating a completely new individual. Then mutation is performed to insert random genes in the offspring to maintain the diversity in the population. These processes continue until a certain stopping criterion is satisfied. Genetic Algorithms provide optimisation over large state space.

A Search Algorithm*- A* Search algorithm is one of the best and most popular techniques used in path-finding and graph traversals. It is an extension of Dijkstra's Algorithm of the shortest path. A* uses a heuristic function that provides additional information regarding how far away from the goal node we are. It uses an evaluation function $f(n) = g(n) + h(n)$ where $g(n)$ is the actual cost to go from the current state n to the goal state, whereas $h(n)$ is the heuristic function which estimates the cost of reaching to goal state from the current state n. At each point, A* tries to find the next neighboring state with a minimum evaluation function value.

II. METHODOLOGY

A. Step1: Data Collection

Data collection is an important part of this project. Noise data from IISER Bhopal Campus is collected using a recorder device. Here, the time of collecting data might affect the result. However, due to low traffic noise on the campus, this won't affect our final output as much as it does when data is collected in traffic-prone areas of the cities.

B. Step 2: Building the data for implementation

After data collection, we started with pre-processing the data to transform it into an appropriate and suitable format. Firstly, we manually acquired the geographic coordinates using Google Maps of the important points on and around campus and recorded the distances among them in a *CSV* file. Then, we mapped the collected audio files via a dictionary to each of the pre-existing paths. This puts the data in the required format.

C. Step 3: Implementation of the Path Planning

3.1: Genetic Algorithm

Firstly, we load the pre-processed data, along with the audio files. While loading the audio files, we use the *Short Time Fourier Transform* from the *Librosa* library, to use the mean of the amplitude values, and the point-wise distances to use for the genetic algorithm implementation.

Now, we need to define our initial population. We define our starting and ending points, and then using function *next_dest*, we randomly pick a neighbour of the current point, and append it to our route. The neighbours of a point were systematically maintained using the function, *calculate_neighbors*, which keeps track of the viable neighbouring points of a source point. Our code keeps track of the current and the previous points in the route, while ensuring that the path length doesn't exceed 20 units. This marks the creation of the initial population.

Next, we calculate the fitness value of our population using the *calculate_fitness* funciton which adds the distance of the next point as we move from the start point to the endpoint. We select the best 10 routes based on their fitness value for crossover. We perform crossover iteratively by randomly choosing two routes from these 10 best routes and halving these routes and then merging them to generate new routes. Now for these new routes we calculate the fitness using the *mutuated_fitness* function, attempts to insert a valid intermediate node between two adjacent nodes in the route to explore alternate paths and evaluates the new route's fitness based on the combined cost. We then output the best route based on the fitness value.

3.2: A* Algorithm

As mentioned above, we first load the pre-processed data along with the collected audio files, and then use the Librosa Library to use the mean of the amplitude values as well as the point-wise distances to provide as input to the A* algorithm. Here we have used the average of the normalized noise levels and distance as our evaluation function for A* search.

III. Results and Discussion

Genetic Algorithm- Running the Genetic Algorithm we get the route which is the optimal route in the sense that the average noise level is minimum on this route while also maintaining the distance of the route. Note that, on running the genetic algorithm more than once we may get different routes as the output because the genetic algorithm, at each time randomly generates the population and then tries to find the optimal route based on the fitness values of the routes in the population.

For example, while choosing *H7* as our initial point and *Main Gate* as our endpoint, one run of the algorithm outputs the route:

H7 – H7 Jn - H1 Jn – IWD Jn – CCD – Library – Shopping Center – Director Bunglow – VH Jn – Resident Area Jn – Main Gate

Whereas another run of the algorithm may output:

H7 – Ideation Hut – H1 - H1 Jn – IWD Jn – CCD – Library – Shopping Center – Director Bunglow – VH Jn – Resident Area Jn – Main Gate

A Algorithm*- Running the A* search algorithm, we get the best route based on our evaluation function. Here, running it several times will not affect the final output as the evaluation function used is based on the average amplitude values and the distances which remains the same on each run.

Choosing *H7* as our initial point and *Main Gate* as our endpoint, it outputs the route:

H7 – Ideation Hut – H1 - H1 Jn – IWD Jn – CCD – Library – Shopping Center – Director Bunglow – VH Jn – Resident Area Jn – Main Gate

Thus, using the noise data we can perform the path planning with Genetic and A* star algorithm. Here, we haven't considered certain factors while collecting data such as collecting data on cycle or while walking and many more. These factors might affect our final output and so the above method can be made more accurate considering these factors.