Finding the Optimal Location of Hospitals using Hill Climbing and Genetic Algorithm in IISERB

Prajjwal Sanskar (20324)[a], **Manish Gayen (21161)**[b], **Sayandeep Pal (21244)**[b], **Srijan Pradhan (21273)**[c]

[a]Department of Electrical Engineering and Computer Sciences; [b]Department of Data Sciences and Engineering; [c]Department of Economic Sciences

Guided By: Pyare Lal, PhD, Department of Data Sciences and Engineering
Instructor: Dr. Vaibhav Kumar, Assistant Professor,Department of Data Sciences and Engineering

Abstract

Hospitals are critical infrastructure in any city or village, and their optimal location requires meticulous scientific planning rather than intuitive guesswork. In this project, we use the existing infrastructure at IISER Bhopal to evaluate whether the campus health center is optimally situated. We first apply the Hill Climbing algorithm to determine the best possible site for a new hospital within the campus. Additionally, we explore the potential of genetic algorithms to further refine our location analysis, ensuring a robust and comprehensive approach to facility placement solely within the IISER Bhopal campus.

1. Introduction

The objective of our project is to identify the optimal location for a hospital inside the IISER Bhopal campus. This task poses a computational challenge due to the consideration of the spatial layout of the existing buildings and roads. To address this problem we are using the Hill Climb Algorithm and in the later phase of the problem we intend to incorporate Genetic Algorithm as well.By using these techniques we aim to determine the most optimal allotment of a hospital to enhance accessibility and coverage within the given area. Our approach involves several key steps which are detailed in the following sections.

2. Data Extraction

We have used QGIS software to extract data as follows:

I. Hill Climbing

For finding an optimum hospital location in IISERB campus, we divided our total area into two sub-parts:

1. Exploring existing building centroids as potential location for a health centre.
2. Choosing a new location from the available empty land area.

- **Using Existing Buildings**

We took the centroids of buildings and considered them as nodes. Then, we used the roads as a network and generated the distance matrix for each node to all the others routing through the road network with the help of QNEAT3's *'OD Matrix from Layers as Lines'* algorithm.

- **Empty Land Area**

1. For the empty land area, we applied a square grid on the IISERB map area and clipped it to exactly overlap the shape of the IISERB map.
2. We used the Vector Overlay - Difference function to subtract the area of the buildings from this overlayed grid to work only with the leftover land area.
3. Then, we generated centroids for these grid boxes and generated the network distance matrix for each blank area centroid to all the buildings using the QNEAT3's algorithm.

In the end, we obtained two distance matrices containing the network distance of:

1. Each building to all other buildings.
2. Each blank area centroid to all the buildings.

These matrices are then supplied to the algorithm for further use.

Figure 1. *Extracted IISERB Map Represented by Nodes*

II. Genetic Algorithm

First we merge all building centroids and blank points into a single layer, named nodes layer. Then we define three zones over the IISERB map outline : student, academic and faculty. Next we find the overlap of each zone and the nodes layer in order to obtain three layers that contain the nodes for each zone. Then we use the QNEAT3 algorithm to calculate the road distance of each node of each zone to all other nodes in that zone. This information gets stored as distance matrices which are then used in the cost calculation of genes.

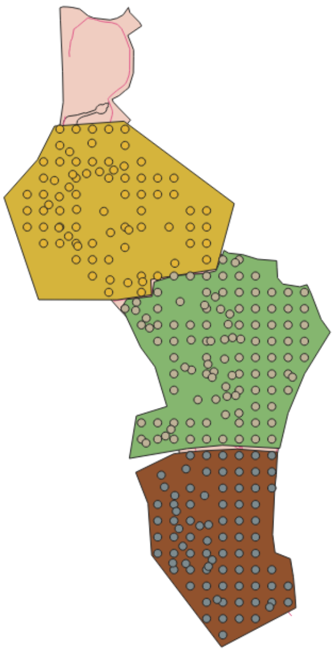


Figure 2. *Zone Map of IISERB for GA*

3. Methodology

3.1. Average Distance Calculation

Let $G = (V, E)$ represent a network graph where V is the set of all the nodes(Building centroids and blank points), and E is the set of edges connecting the nodes. Each node on the graph corresponds to a potential location, and each edge represents a real-world connection between two locations. The objective is to identify the optimal location v^* within V such that the average cost of the edges to the neighboring nodes is minimized. Formally the optimization problem can be formulated as follows:

$$v^* = \arg\min_{v \in V} \frac{1}{|N(v)|} \sum_{u \in N(v)} c(v, u)$$

where:

- v^* is the optimal location.
- $N(v)$ denotes the set of neighboring nodes of v in the graph G.
- $c(v, u)$ represents the cost associated with the edges between nodes v and u.

3.2. Hill Climbing Algorithm

Hill climbing is a local search algorithm that is used for optimization problems. The algorithm starts with an initial point and iteratively moves to neighbours that optimize the objective function until a local optimum is reached. Here in this problem our objective function is the average cost of edge weights (road distances) to the neighbouring nodes. The algorithm hence iterates through these nodes and selects the neighbour which yields the lowest average cost.

3.2.1 Random Restarts

The hill climb algorithm usually gets stuck in a local optimum solution and hence it becomes important to escape such points in order to reach a global optimum point and explore a broader solution space. We thus incorporated random restarts into our hill climb algorithm, here on reaching a local optimum the algorithm restarts from a randomly chosen initial node and continues the search. This allows for greater exploration of the solution space and increases the likelihood of finding a global optimum.

3.3. Genetic Algorithm

Genetic algorithms are evolutionary techniques used for optimization problems. They mimic natural selection, where only the fittest solutions are chosen for reproduction to generate offspring. It begins with an initial population of potential locations, evaluates their fitness based on a specific criterion (like accessibility or cost), and uses operations like selection, crossover, and mutation to evolve the solutions and arrive at an optimal solution.

3.3.1 Chromosome Formation

In genetic algorithms initial step is chromosome formation where possible solutions to the problem are encoded into chromosomes. Each chromosome is a string of genes that stands for various parameters or decisions independent of the solution space. The particular structure of a chromosome depends upon the type of a problem, which generally consists of a sequence of values or traits specifying a full solution. This encoding enables crossing and mutating genetic operations to take place and thus facilitates the evolutionary search for the best solution. Initially, the population of chromosomes is generated either randomly or by exploiting some heuristic in order to get the cover of most of the space in the solution. .

3.3.2 Crossing Over

Crossing over is a process where genetic material is exchanged between parent chromosomes to create offspring. This involves selecting parents based on fitness, determining a crossover point, swapping genes at this point, and forming new chromosomes. These offspring are then evaluated for their fitness. The purpose of crossover is to combine beneficial traits from different solutions, encouraging the exploration of new solution spaces and improving problem-solving capabilities.

3.3.3 Mutation

The mutation process introduces variability into the population by randomly altering genes in the offspring chromosomes produced by crossover. This alteration typically involves changing a gene to a different value within its range, simulating random mutations in biological evolution. Mutation serves to maintain genetic diversity within the population, helping to prevent premature convergence on suboptimal solutions and enabling the exploration of new areas in the solution space. This step can lead to the discovery of new and potentially more effective solutions.

4. Experimental Setup

4.1. Hill Climbing Algorithm

We applied the proposed hill climbing algorithm with random restarts to the real-world dataset representing a network graph of IISER Bhopal. The dataset consists of nodes representing buildings and blank points, with edges indicating connectivity between them. The cost associated with each edge represents the road-distance or transportation cost between nodes.

Algorithm 1 Hill Climbing with Random Restarts

Require: $graph$, $iterations$
  $best\_solution \leftarrow$ None
  $best\_cost \leftarrow \infty$
  for $iteration$ **in** $range(1, iterations + 1)$ **do**
    $start\_node \leftarrow$ randomly choose a node from $graph.nodes$
    $current\_node \leftarrow start\_node$
    $initial\_cost \leftarrow calculate\_average\_cost(graph, current\_node)$
    $current\_cost \leftarrow initial\_cost$
    $improving \leftarrow$ True
    while $improving$ **do**
      $improving \leftarrow$ False
      $neighbors \leftarrow$ get neighbors of $current\_node$
      for $neighbor$ **in** $neighbors$ **do**
        $neighbor\_cost \leftarrow calculate\_average\_cost(graph, neighbor)$
        if $neighbor\_cost < current\_cost$ **then**
          $current\_node \leftarrow neighbor$
          $current\_cost \leftarrow neighbor\_cost$
          $improving \leftarrow$ True
          break {Move to the first better neighbor}
        end if
      end for
    end while
    if $current\_cost < best\_cost$ **then**
      $best\_solution \leftarrow current\_node$
      $best\_cost \leftarrow current\_cost$
    end if
  end for
  $node\_data \leftarrow$ get data of node $best\_solution$ from $graph$
  $optimal\_location \leftarrow$ { 'ID/Name': $best\_solution$, 'Average Cost': $best\_cost$, 'Coordinates': $(node\_data['x'], node\_data['y'])$ }
  if 'type' in $node\_data$ **and** $node\_data['type']$ == 'building' **then**
    $optimal\_location['Type'] \leftarrow$ 'Building'
  else
    $optimal\_location['Type'] \leftarrow$ 'Blank Point'
  end if
  return $optimal\_location$

We start with initiating the nodes of the graph with the given coordinates and the road distance as the edge weights. The objective of taking the average distance is our objective function and not the total distance because of:

- **Scale Sensitivity and Lack of Normalization:** The total distance measure is sensitive to the number of edges connected to every other node in the graph and hence favours nodes with fewer connections which are in remote locations on the map. This bias might not always align with finding the true optimal location in the real-world. Moreover, due to the lack of normalization, it becomes harder to compare nodes with different degrees directly.

We run the algorithm for 25 random restarts, and for every iteration, we print the optimum location, coordinates, final average cost, initial starting point, initial average cost, and the type of point (whether it is a blank point or a building centroid). The final optimum location is the optimal location among all the other optima with the least average cost. We then plot the optimal location on the graph with a marker for identification.

4.2. Genetic Algorithm

Similar to Hill Climbing, we implemented a Genetic Algorithm on the IISER Bhopal dataset to optimize hospital placement within the campus. The Genetic Algorithm uses an initial population of chromosomes, each consisting of nodes selected from predefined regions (Academic, Faculty, and Student areas) representing potential hospital locations.

- **Chromosome Representation:** The chromosomes represent potential solutions involving three key regions of IISER Bhopal: Academic, Faculty, and Student areas. To generate initial solutions, ten such chromosomes are created. Each chromosome comprises three genes, where each gene corresponds to a randomly selected node from each of these regions. This structure means that each chromosome encapsulates a specific trio of locations—one from each region.

- **Fitness Calculation:** The fitness of each chromosome is then assessed based on the sum of the distances between the nodes it contains. It helps guide the selection process within the genetic algorithm, favoring chromosomes that represent shorter total distances, thus optimizing the connectivity or accessibility among the chosen locations.

- **Genetic Operations:** The process of crossover is used to combine and enhance the genetic material of chromosomes. After sorting the chromosomes by fitness value, with the lower values indicating more optimal solutions, crossover occurs between successive pairs of chromosomes. This technique involves swapping genes between a chromosome and its neighbor to produce new offspring. These offspring, which inherit genes from both parent chromosomes, are then evaluated to determine their fitness based on the same criteria: the sum of the distances between the nodes they represent. This step helps to explore new potential solutions and refine the search towards the most efficient configurations of the three selected locations at IISER Bhopal.

After that, the mutation process is applied to each offspring resulting from the crossover. This process is performed for each of the three genes corresponding to the Academic, Faculty, and Student regions of IISER Bhopal. This step introduces variability into the population, potentially uncovering new and more effective solutions that were not previously considered.

Algorithm 2 Genetic Algorithm for Location Optimization

Require: $location\&distance\_matrices$, $num\_chromosomes$, $iterations$
  $chromosomes \leftarrow$ initialize $num\_chromosomes$ chromosomes
  $fitness\_scores \leftarrow$ calculate fitness scores for each chromosome
  $sorted\_chromosomes \leftarrow$ sort chromosomes based on fitness scores
  for $iteration$ **in** $range(1, iterations + 1)$ **do**
    $new\_generation \leftarrow$ empty list
    $zone\_dfs \leftarrow [academic\_zones\_df, faculty\_zone\_df, student\_zones\_df]$
    for i **in** $range(len(sorted\_chromosomes) - 1)$ **do**
      $offspring1, offspring2 \leftarrow$ crossover between adjacent chromosomes
      $gene\_to\_mutate \leftarrow$ randomly choose a zone to mutate
      $offspring1 \leftarrow$ mutate $offspring1$ in $gene\_to\_mutate$ zone
      $offspring2 \leftarrow$ mutate $offspring2$ in $gene\_to\_mutate$ zone
      append $offspring1$ and $offspring2$ to $new\_generation$
    end for
    $new\_fitness\_scores \leftarrow$ calculate fitness scores for new generation
    $best\_chromosome\_index \leftarrow$ index of chromosome with minimum fitness score in $new\_fitness\_scores$
    $best\_chromosome \leftarrow$ chromosome at $best\_chromosome\_index$
    $sorted\_chromosomes \leftarrow$ sort new generation based on fitness scores
  end for
  $best\_chromosome\_coordinates \leftarrow$ get coordinates from best chromosome
  $coordinates \leftarrow$ convert $best\_chromosome\_coordinates$ to numpy array
  Perform Delaunay triangulation on $coordinates$
  return $optimal\_location\_GA \leftarrow$ get coordinates of Triangle formed by Delaunay triangulation
  return $optimal\_location\_GA$

- **Algorithm Execution:** After mutation, the fitness of each chromosome is recalculated by assessing the total distance between the nodes it contains. The genetic algorithm runs over two generations: the parent generation and the offspring generation generated after crossover. This process continually improves the population. The best chromosome is identified based on the lowest fitness score among these two generations, representing the most efficient arrangement of nodes across the three regions. Finally, a triangular setup has been formed with these three nodes of each region, and we consider the centroid of that triangle as the best optimal location for our problem.
- **Result Visualization:** The optimal location determined by the genetic algorithm is visualized on the campus map, highlighting the most strategically beneficial site for a new hospital based on comprehensive genetic analysis.

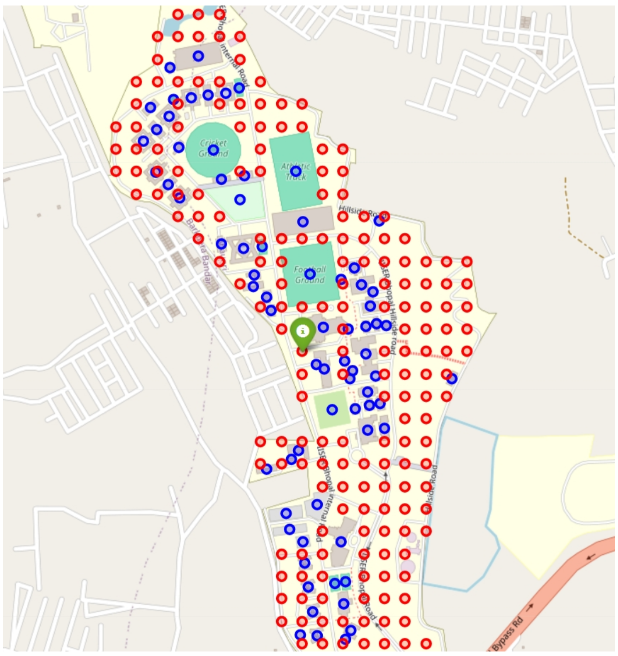
5. Result and Interpretation

I. Hill Climbing

The results of our iterations are summarized in Table 1. Each row in the table represents an iteration, with columns indicating the iteration number, the initial starting point, the initial cost associated with that point, the optimal location found after applying the hill climbing algorithm, and the average cost at the optimal location.

From the results, we observe that the algorithm converges to an optimal solution in most cases. However, there are instances where the algorithm fails to converge due to an infinite average cost, as seen in iteration 15. This suggests that there is no road or edges connected with this node.The optimal location (Figure 3 & 4) based on our analysis is a blank point with ID 882.0. It has an average cost of 572.83 and is located at coordinates (23.2876064, 77.2741175) between the Lecture Hall Complex (LHC) and Cafe Coffee Day (CCD) within the campus. This optimal location represents the most efficient placement for a hospital considering the existing infrastructure and spatial constraints.

Figure 3. *Optimal Allocation (In IISERB Map)*

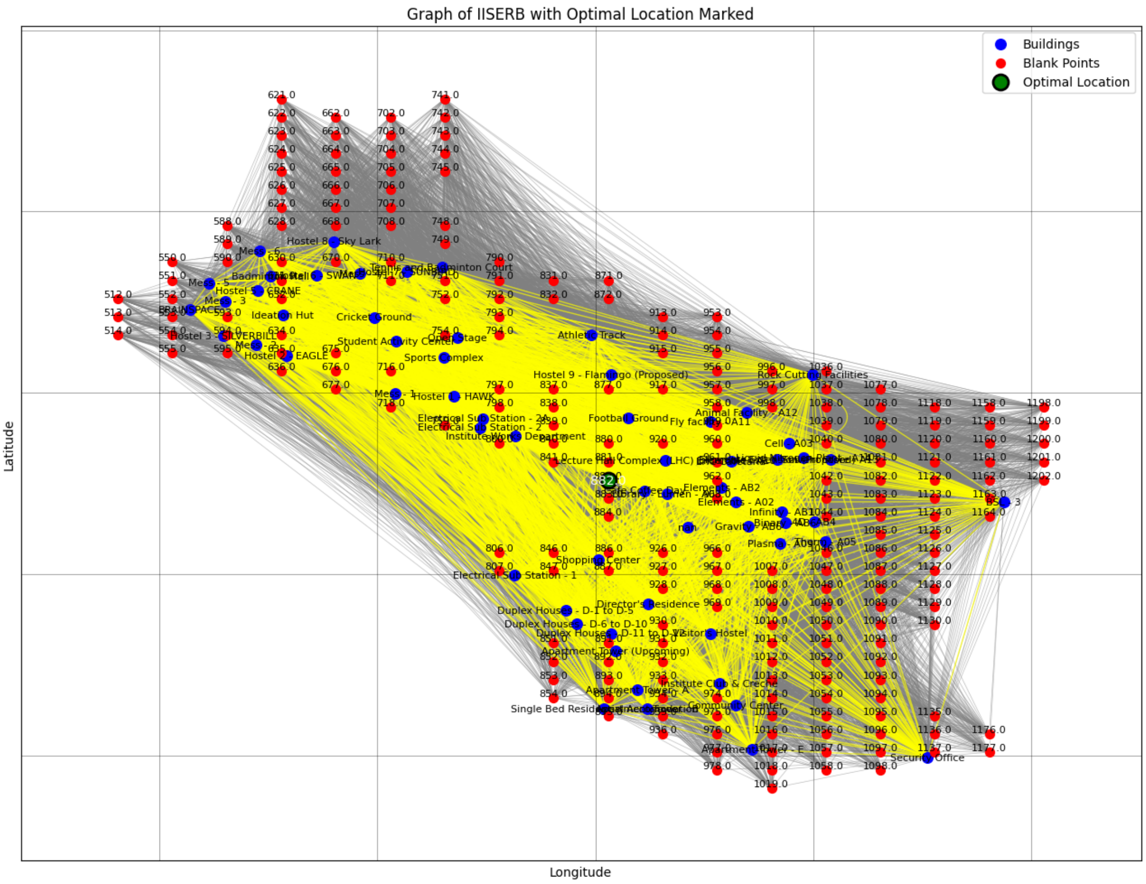


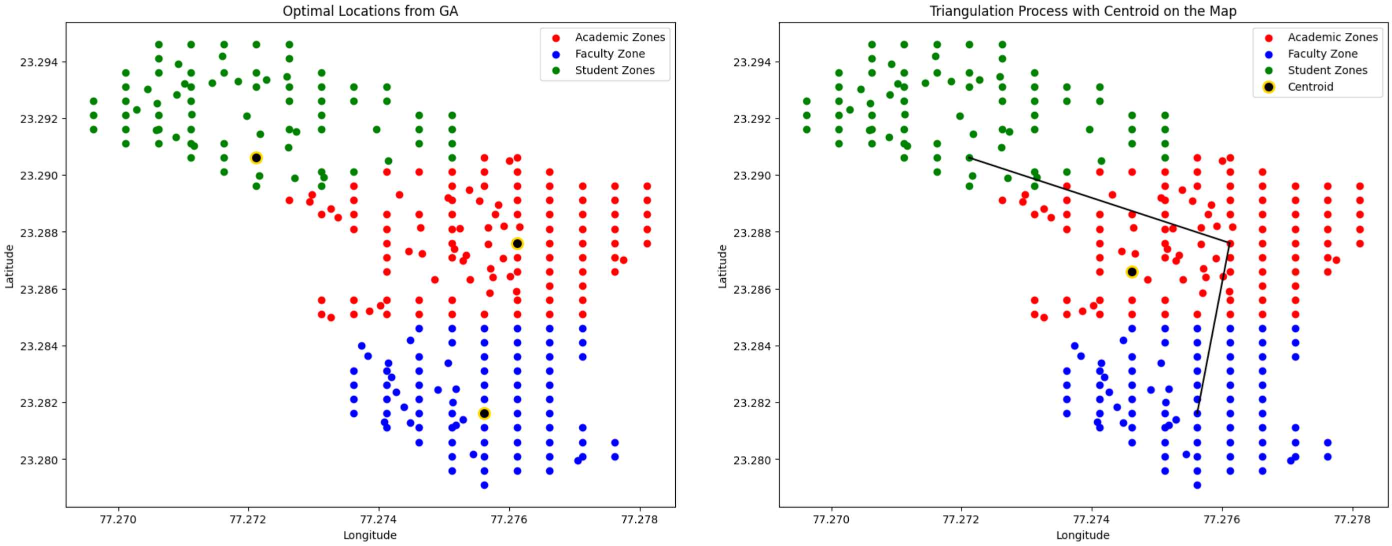
Overall, these results provide valuable insights into the effectiveness of the hill climbing algorithm in finding optimal hospital locations within the IISER Bhopal campus. Further

Table 1. *Iteration Results*

Iteration	Initial Point	Initial Cost	Optimal Location	Average Cost
1	838.0	580.687031	838.0	580.687031
2	1078.0	711.081965	716.0	682.221454
3	Football Ground	799.554864	716.0	682.221454
4	975.0	989.733192	716.0	682.221454
5	882.0	572.826198	882.0	572.826198
6	955.0	785.416222	716.0	682.221454
7	1198.0	974.171541	716.0	682.221454
8	1121.0	744.322672	716.0	682.221454
9	Animal Facility - A12	750.885367	716.0	682.221454
10	626.0	1241.979424	716.0	682.221454
11	1096.0	1098.037392	716.0	682.221454
12	1089.0	1061.181995	716.0	682.221454
13	1036.0	737.202192	716.0	682.221454
14	708.0	906.705285	716.0	682.221454
15	1092.0	inf	1092.0	inf
16	Director's Residence	812.589926	716.0	682.221454
17	1082.0	714.561843	716.0	682.221454
18	Electrical Sub Station - 2	828.889651	716.0	682.221454
19	996.0	733.414183	716.0	682.221454
20	590.0	1030.056097	716.0	682.221454
21	668.0	962.474136	716.0	682.221454
22	853.0	972.183423	716.0	682.221454
23	Animal Facility - A12	750.885367	716.0	682.221454
24	Hostel 2 - EAGLE	963.121686	716.0	682.221454
25	1013.0	957.588794	716.0	682.221454

Figure 4. *Optimal Allocation (Nodes & Edges)*





(a) *Best Chromosome (GA)* **(b)** *Optimal Allocation (GA)*

Figure 5. *Genetic Algorithm*

analysis and refinement of the algorithm may be necessary to improve its performance and reliability.

II. Genetic Algorithm

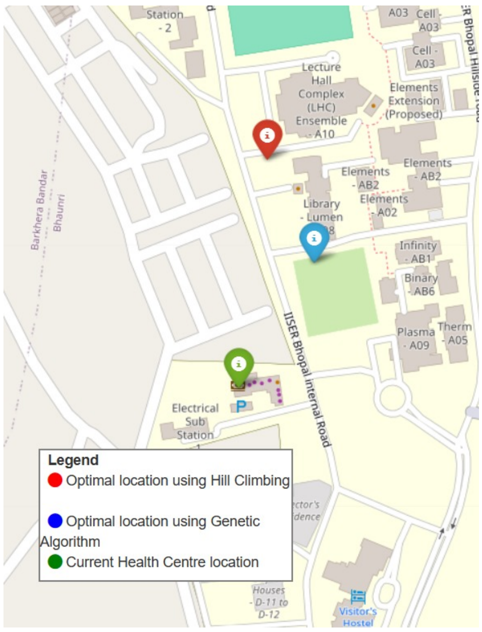
After employing the Genetic Algorithm (including Crossing Over, Mutation, and all other components) on IISER Bhopal with a seed value of 42 for the random function, we have found our best chromosome (Figure 5a) based on the minimum fitness value, which contains the coordinates (23.2876064, 77.2761175) for the academic region, (23.2816064, 77.2756175) for the faculty region, and (23.2906064, 77.2721175) for the student region. We then utilized these coordinates to perform Delaunay triangulation, forming a triangle with the academic, faculty, and student region coordinates present in the best chromosome as its vertices. The centroid of this triangle was considered as the optimal location (Figure 5b) for the hospital, which was determined to be (23.2866064, 77.2746175). This optimal location is situated in front of the Library.

6. Comparison between HC & GA

In a Genetic Algorithm, the initial population is randomly generated and mutation is also performed randomly, thereby creating an inherent randomness in the overall search space, which implies that it may not always result in the optimal location. On the other hand, in Hill Climbing, we move to the next neighbor if the cost value decreases, and the process continues until it reaches a local optima. Subsequently, the algorithm restarts when it gets stuck at a local optima or plateau points. Hence, with Hill Climbing we get to explore a wider search space thereby increasing our chances of reaching the global optima. So Hill Climbing seems to be more promising for the task when there are no limitations on the number of iterations. However Genetic Algorithm proves to have an upper hand when it comes to finding optimal location within limited iterations. So both algorithms may give us the optimal location, but it all boils down to available compute power and time.

Now, looking at the obtained locations in figure 6 on the map of IISERB, we can clearly see that the current location of the health centre is not optimal as it neither coincides with the location obtained by the hill climbing algorithm nor the location obtained by the genetic algorithm. However, its fairly close to both of them. Plus we haven't considered other logistical issues associated with the location of a health centre. ***So, our final verdict is that the location current location of the health centre is not optimal. However, given the fact that some logistical issues have been ignored for the sake of simplicity of the problem, the current location lies in an acceptable proximity of the obtained optimal locations.***

Figure 6. *Comparison between HC and GA*



7. Limitations

1. Calculating network distance matrix for large number of nodes using the QNEAT3 algorithm is computationally expensive.

2. For selecting the optimum location, we are only considering road distance as the sole factor. however other factors like population density, lodging facilities, railways and other non-road means of transportation,etc should also be considered. These factors were ignored due to unavailability of accurate data on these factors.

8. Future Direction of Work

To expand our project to vast urban areas like the size of Bhopal or other cities, the application of hill climbing and genetic algorithm can be suggested to optimize the placement of the requirement facilities such as hospitals among others. This strategy successfully handles such complex variables as population density and attending infrastructure costs. Therefore, the distribution of health services is carefully organized to be equally available to all though the extensive regions. This framework is presented with the aim of evaluating the findings of subsequent works and can be adjusted for similar planning projects. .

9. Individual Contribution

- **Manish Gayen:** QGIS, Data extraction, Genetic Algorithm implementation, Documentation
- **Srijan Pradhan:** Data Preprocessing, Implementation Idea, Code Modification (HC & GA), and Debugging, Documentation
- **Sayandeep Pal:** Coding(Hill Climb and Genetic Algorithm), Algorithm Design, Data Preprocessing,Documentation
- **Prajjwal Sanskar:** Literature Review, Genetic Algorithm Implementation, Code Review

10. References

- CS50's Introduction to Artificial Intelligence with Python, Brian Yu & David J. Malan, Harvard
- Week 3 Optimization (Video and Code), Brian Yu, Harvard
- Jingkuang Liu, Yuqing Li, Ying Li, Chen Zibo, Xiaotong Lian, Yingyi Zhang, Location optimization of emergency medical facilities for public health emergencies in megacities based on genetic algorithm, ISSN: 0969-9988 (2023)
- QNEAT3 OD Matrix from Layers as Lines

11. Supplementary

- Link to the Github Repository