Self Project Using Integer Programming **M a y - J u l y ' 2023**

Project Report : Longest Path for conducting race in IITB campus

Abstract

The aim of this project is to determine the longest possible path for conducting a race within the IIT Bombay campus. This project involves analyzing the campus layout, identifying potential routes and obstacles, and using mathematical algorithms(Integer Programming) to calculate the longest path that can be taken without repeating any section of the route. The findings of this project could be used to plan and organize future races and sporting events within the IIT Bombay campus.

1 Introduction

The problem of finding the longest route in the IITB campus can be related to the well-known Traveling Salesman Problem (TSP), which is a classic optimization problem. In the TSP, a salesman has to visit a set of cities and return to the starting city, minimizing the total distance traveled.

Similarly, in the problem of finding the longest route in the IITB campus, we are looking for the longest path that connects a set of locations while satisfying certain conditions. This can be viewed as a variation of the TSP, where the objective is to find the longest path instead of the shortest path.

However, there are some differences between the two problems. In the TSP, all cities must be visited exactly once, whereas in the IITB campus problem, we may not need to visit all locations. Additionally, the TSP usually involves finding the shortest path while the IITB campus problem involves finding the longest path, which changes the nature of the objective function and constraints.

Despite these differences, both problems can be formulated as integer programming problems and solved using similar techniques. The TSP is a well-studied problem with many practical applications, and the IITB campus problem is a specific instance of the more general TSP problem.

2 Problem Setup and data collection

In this project, we are solving the problem of finding the longest route for conducting race in IITB campus under some assumptions and those assumptions are:

- The race must start and end at the same point.
- In the race course same stretch of road can't be revisited.
- It is not required to visit all the nodes pointed in the figure 1.

To solve this problem we have considered graph-based approach.We have taken map of the campus, which includes all the buildings, roads, and other landmarks. The map is in the form of a graph, where each intersection is represented as a node, and the roads between the intersections are represented as edges connecting the nodes.We have considered total 37 number of nodes. The length of each edge is known. We

have taken graph of the IIT Bombay campus from the google which is shown in figure 1 and the distance of each edge is measured using google map which we have stored in an excel file.We have considered nodes N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, N_{14}, N_{15}, N_{16}, N_{17}, N_{18}, N_{19}, N_{20}, N_{21}, N_{22}, N_{23}, N_{24}, N_{25}, N_{26}, N_{27}, N_{28}, N_{29}, N_{30}, N_{31}, N_{32}, N_{33}, N_{34}, N_{35}, N_{36}, N_{37}.

We had used integer programming technique to find the optimal(longest) route for the race. We started by creating a graph representation of the campus, and then modelled an integer program and implemented the same using PYOMO and solved using CBC solver.

Figure 1: Map of IITB [1]

3 Model Formulation

Let us formulate the problem of finding the longest route for conducting race in IIT Bombay campus.

- Let c_{ij} represents distance from node i to node j.
- a_{ij} represents whether there is direct connection between node i and node j.

$$a_{ij} = \begin{cases} 1, & \text{if there is a direct connection between node i and j} \\ 0, & \text{Otherwise} \end{cases}$$

- Let x_{ij} be the decision variable ,

$$x_{ij} = \begin{cases} 1, & \text{if edge joining node i and j is used} \\ 0, & \text{Otherwise} \end{cases}$$

- Let u_i be a dummy variable which will represent the position of node i in the route,$1 \leq u_i \leq n$. Here, we have considered that the race will start from node 1, hence $u_1 = 1$.

Now let's start the formulation for the above mentioned problem using the above defined notations:

- **Objective:** We want to maximize the total distance covered in the route, which is :

$$\max \sum_{i=1}^{37} \sum_{j=1}^{37} c_{ij} \cdot xij$$

- **Constraint 1:** The path between node i and node j can be used if and only if there is a direct connection between node i and j,

$$x_{ij} \leq a_{ij} \qquad \forall i, j = 1, \ldots, 37$$

- **Constraint 2:** Since we have made an assumption that same stretch of road can't be revisited, that is, each edge can be visited at most once,

$$\sum_{i=1, i \neq j}^{37} x_{ij} \leq 1 \qquad \forall j = 1, \ldots, 37$$

- **Constraint 3:** To ensure the connectivity between each path included in the race, that is, number of incoming edges at node i must be equal to the number of outgoing edges from node i

$$\sum_{i=1, i \neq j}^{37} x_{ij} = \sum_{i=1, i \neq j}^{37} x_{ji} \qquad \forall j = 1, \ldots, 37$$

- **Constraint 4:** If node i is visited before node j then node j can't be visited before node i,

$$x_{ij} + x_{ji} \leq 1 \qquad \forall i, j = 1, \ldots, 37, i \neq j$$

- **Constraint 5:** There should not be any sub-tour in the route.

$$u_i - u_j + 1 \leq n(1 - x_{ij}) \qquad \forall i, j = 1, \ldots, 37, i \neq j$$

So the final model that we have considered is as follows:

$$\text{Objective:} \quad \max \sum_{i=1}^{37} \sum_{j=1}^{37} c_{ij} \cdot xij$$

Subject to

1. $x_{ij} \leq a_{ij} \qquad \forall i, j = 1, \ldots, 37$
2. $\sum_{i=1, i \neq j}^{37} x_{ij} \leq 1 \qquad \forall j = 1, \ldots, 37$
3. $\sum_{i=1, i \neq j}^{37} x_{ij} = \sum_{i=1, i \neq j}^{37} x_{ji} \qquad \forall j = 1, \ldots, 37$
4. $x_{ij} + x_{ji} \leq 1 \qquad \forall i, j = 1, \ldots, 37, i \neq j$
5. $u_i - u_j + 1 \leq n(1 - x_{ij}) \qquad \forall i, j = 1, \ldots, 37, i \neq j$
6. $1 \leq u_i \leq n \qquad \forall i = 1, \ldots, 37, u_1 = 1$

4 Experiments and Results

We have implemented the above formulated model in python using pyomo and solved the same using CBC solver and we got the following results:

- Optimal distance is 7672 meter.
- Time taken to solve the model is around 1.22 seconds.
- Obtained optimal path is shown in figure 2 indicated by black line.

Figure 2: Map with longest route:In this map black drawn path is the longest route.

Obtained optimal path is : $N_1 \longrightarrow N_{29} \longrightarrow N_{30} \longrightarrow N_{34} \longrightarrow N_{36} \longrightarrow N_{35} \longrightarrow N_{32} \longrightarrow N31 \longrightarrow N_{13} \longrightarrow N_{33} \longrightarrow N_{18} \longrightarrow N_{23} \longrightarrow N_{19} \longrightarrow N_{21} \longrightarrow N_{22} \longrightarrow N_{24} \longrightarrow N_{16} \longrightarrow N_{17} \longrightarrow N_{14} \longrightarrow N_{20} \longrightarrow N_{12} \longrightarrow N_{25} \longrightarrow N_{26} \longrightarrow N_{27} \longrightarrow N_{28} \longrightarrow N_2 \longrightarrow N_3 \longrightarrow N_4 \longrightarrow N_6 \longrightarrow N_{10} \longrightarrow N_{11} \longrightarrow N_9 \longrightarrow N_7 \longrightarrow N_8 \longrightarrow N_5 \longrightarrow N_1$

5 Conclusion

In the process of finding the longest route for conducting a longest race in the IIT Bombay campus involved several stages of planning, mapping, and measuring distances. Through the use of Google Maps and integer programming technique we were able to identify a route that spans a distance of approximately 7.67 kilometers.

The identified route which starts from H-5(N_1) passes through different parts of the campus, including roads, walkways, and green spaces, providing a challenging and diverse terrain for the race participants. Moreover, the route is well-suited for a variety of race types, including long-distance running, cycling, and walking events.

Overall, this project has been successful in achieving its goal of identifying the longest route for conducting a race in the IIT Bombay campus.

6 Details of additional files

we have uploaded 3 additional files with this report and those are:

1. **Distance Matrix :** It is CSV file which contains the distances between each nodes.If there is not any direct connection between node i and node j then we have considered it to be zero. All the distances are measured in meter.
2. **Adjacency Matrix :** It gives us information whether there is a direct connection between node i and node j. If there is a direct connection then entry will be 1 otherwise 0.
3. **Python File :** It is the implementation of our formulated model in python and to model it in python we used PYOMO and we used CBC solver to solve the model

References

[1] IIT Bombay Map - Mood Indigo 2015 [Online : accessed 08-06-2023]. URL:https://www.behance.net/gallery/36946033/IIT-Bombay-Map-Mood-Indigo-2015

[2] Prof. A Mahajan, Lecture Notes of Integer Programming : Theory and Computations, IIT Bombay,

[3] https://www.google.com/maps