DATS-SHU 240 Project Proposal

jn2294 lg3490

March 2023

1 Introduction

As NYU Shanghai has built its beautiful New Bund Campus, there are many visitors coming to the new campus, and having a campus tour. Since most visitors have a limited amount of time, the campus tour should be efficient. This project aims to optimize visitors' route of campus tours in order to maximize their campus tour experience within a limited amount of time. Therefore, we seek to construct a mathematical model to choose an optimal visiting route that maximizes the total weights of each part of the campus they have visited, with a time limit as a general constraint. Each part of the campus is represented by different weights, respectively.

2 Methodology

Routing problems, also known as path-finding problems, refer to the task of finding the optimal path between two points in a graph or network. The objective is typically to find the path that minimizes a given cost or distance metric while adhering to any constraints or limitations that may exist in the network. There are many algorithms to find the optimal path, some uses the classical algorithm like A^* algorithm and genetic algorithms [CLY22]. But they all took some heuristic method in settling the path and in certain fields like robotics these algorithms work very well. But in our project, we don't want to include randomness to complicate our model. Thus, we won't use these algorithms. Instead, the backbone theory behind our mathematical optimization model is the traditional routing finding algorithm. The modeling process basically follows the optimization setting under the framework of capacitated vehicle routing problems(CVRP in short, see Appendix A).

Since it is almost impossible for visitors to go to the same floor several times, we assume that visitors take the campus tour floor by floor. Then the mathematical model becomes a combination of n subproblems, where n is the number of floors of the campus. Each subproblem is an optimization of the route of a floor, although in order to maximize the total weights, those subproblems are interconnected since the general limitation is the total amount of time.

Each subproblem would be to maximize the importance/value of the covered places on campus. The importance of a place is modeled as a node with a constant weight. And the traveling time between two places is modeled as an edge with constant time. Thus, each floor of our campus can be modeled as an undirected cyclic graph G(N,E) where $N = C \cup \{0, n+1\}$, where C is the places of interest that we pick on a specific floor, and 0 and n+1 is the departing point and ending point on each floor.

The constant parameter for our optimization problem includes:

- A symmetric adjacency matrix $\boldsymbol{A}$ to represent the connectivity between node i and j (denoted as A_{ij}) in the graph (correspondingly, whether two places are directly connected) where each entry is a binary constant.
- A symmetric distance matrix $\boldsymbol{T}$ to represent the traveling time between node i and j (denoted in T_{ij}).

The variable matrix would be a route selection matrix $\boldsymbol{X}$ to represent whether the visitor travels from node i to node j (denoted in $\boldsymbol{X}_{ij}$) where each entry is a binary constant.

The constraint would be to limit the time that a person can explore the campus, which can be expressed as a weighted sum. Moreover, we want the edges that are selected only when the nodes at both ends are connected ($\boldsymbol{X}_{ij} \leq \boldsymbol{T}_{ij}$). To guarantee the validity of the path to generate, we will use the method from [MDS17] to ensure that the edges that the algorithm selected actually form a valid path, i.e., whenever a person enters a node, he must leave that node before traveling to the next node.

When connecting each subproblem, we assume that the time that is spent between ending path-finding on floor i and starting path-finding on floor j is constant. (By taking a specific elevator).

On top of that, we assume that either we explore the floor i or we skip the floor j, which can be modeled as a binary variable defined for each floor.

Finally, the objective function would be to maximize the total weight of the path that is chosen across all floors.

3 Data Source and Description

We measure the total time consumption as the summation of time of exploring each site, the walking time, and the time for taking the life. The time of exploring each site is constant in order to let visitors fully experience the site. In addition, the walking time is measured by the walking distance between each site, which can be obtained from the official map of each floor provided by NYU Shanghai Student Life. In addition, we measure the visitors' experience of the campus tour by the summation of the weights of each site that they have visited. The weights of each site are constant, which can be obtained from our personal experience and the area of a site. Also, the time for taking the lift is also constant. We can obtain it from our personal experience.

[CE14]

Appendix A CVRP

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{n+1}\sum_{j=0}^{n+1} c_{ij}x_{ij} && && (2.1)\\
\text{s.t.} \quad & \sum_{\substack{j=1\\ j\neq i}}^{n+1} x_{ij} = 1, && i = 1,\ldots,n, && (2.2)\\
& \sum_{\substack{i=0\\ i\neq h}}^{n} x_{ih} - \sum_{\substack{j=1\\ j\neq h}}^{n+1} x_{hj} = 0, && h = 1,\ldots,n, && (2.3)\\
& \sum_{j=1}^{n} x_{0j} \leq K, && && (2.4)\\
& y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}), && i,j = 0,\ldots,n+1, && (2.5)\\
& d_i \leq y_i \leq Q, && i = 0,\ldots,n+1, && (2.6)\\
& x_{ij} \in \{0,1\}, && i,j = 0,\ldots,n+1. && (2.7)
\end{aligned}
$$

Constraints (2.2) ensure that all customers are visited exactly once. Constraints (2.3) guarantee the correct flow of vehicles through the arcs, by stating that if a vehicle arrives to a node $h \in \mathcal{N}$, then it must depart from this node. Constraint (2.4) limits the maximum number of routes to K, the number of vehicles. Constraints (2.5) and (2.6) ensure together that the vehicle capacity is not exceeded. The objective function is defined by (2.1) and imposes that the total travel cost of the routes is minimized.

Constraints (2.5) also avoid subtours in the solution, *i.e.* cycling routes that do not pass through the depot. Different types of constraints are proposed in the literature to impose vehicle capacities and/or avoid subtours [17]. The advantage of using (2.5) and (2.6) is that the model has a polynomial number of constraints in terms of the number of customers. However, the lower bound provided by the linear relaxation of this model is known to be weak in relation to other models. Hence, many authors recur to capacity constraints that results in better lower bounds, even though the number of constraints becomes exponential in terms of the number of customers, requiring the use of a branch-and-cut strategy [28].

Figure 1: CVPR formulation

References

[CE14] G.C. Calafiore and L. El Ghaoui. *Optimization Models*. Control systems and optimization series. Cambridge University Press, October 2014.

[CLY22] R Cheng, X Lu, and X Yu. A Mathematical Model for the Routing Optimization Problem with Time Window. *Journal of Physics: Conference Series*, 2219(1):012038, April 2022.

[MDS17] Pedro Munari, Twan Dollevoet, and Remy Spliet. A generalized formulation for vehicle routing problems, 2017.