Travel planning using the best possible route and maximizing the visits to the sites using Orienteering Problem

M Pavan Naik
2021AIM1011
dept. of cse, IIT Ropar
2021aim1011@iitrpr.ac.in

Mohd Sadique Raheem Shaikh
2021AIM1012
dept. of cse, IIT Ropar
2021aim1012@iitrpr.ac.in

***Abstract*—While travelling the orienteering done by smart devices shows the shortest route from source to destination. If we are travelling for holidays then we plan to have a visit to the local sites on the way and to have the best possible scenic view of the road. The solutions for this have been proposed, but mostly they are based on the static dataset. We propose a solution by considering time-dependent data. As the problem is considered in the NP-hard category so the solution may be suboptimal but near to the optimal solution. This Orienteering Problem(OP) where the aim is to find the path from source to destination with maximising the heuristic function value (in this case value will depend on the location) within the time limit. The static OP approaches take these values constant throughout the time but in reality, they vary according to the time of day(eg: lake view at sunset and at night), therefore the static OP approach results are majorly far from optimal. We propose to use the time-dependent spatial data for the road networks as well as the time-dependent heuristic for the locations and then calculate the most optimal path for travelling. The solution will contain the route with maximum possible visits of different sites without exceeding the time window.**

I. Introduction

When we visit a place such as a tourist location and are willing to explore the most scenic places, it's very difficult to decide which place to visit and at what time to visit. Deciding the place and time to visit a place results in enjoying the place.

There are many websites that give such information, but those are static data that have been uploaded based on data collected many days before. Following those information leads to dissatisfaction and waste of cost and time.

So, the above problem can be considered a time-dependent orienteering problem.

II. Related published works

The following Paper were used for the literature review where each have unique approaches and the different methods were learned from them:

[2] from traditional route planners who focus on the shortest path the author focuses on finding the route which has the most scenic view using AOP categorized as NP-hard problem. They use memetic algorithm with heuristic as time and budget constraints. For path improvement chromosome encoding and decoding are used and the experiment tests are performed.

[1] The paper uses a novel approach called twofold time Twofold Time-Dependent Arc Orienteering Problem (2TD-AOP) which is a variant of AOP. Here Instead of taking the edge values as constant they are time-dependent. Then find solutions using spatial pruning techniques on a large scale

[3] The Paper is the special case of Arc Orienteering Problem which mainly focuses on cycle trip planning so the path is upper bounded and finds the best route using branch and cut approach.

III. Preliminaries

Let a directed graph or network G be given by a pair (V, E) with V the set of nodes and E the set of edges. A path is a sequence of nodes without duplicate elements, such that two consecutive nodes are connected by an edge. We assume that two particular nodes are given, an origin node and a destination node. The shortest path from the origin to the destination is looked for. The weight or length of an edge (u,v) is denoted by d(u,v), whereas $d*(u,v)$ denotes the length of the shortest path from u to v. As aforementioned, A* assumes a heuristic estimate h, defined as a function from V into R. An estimate h is called consistent if h obeys the inequality $h(u)- h(v) \leq d*(u,v)$ for any two nodes $u,v \in V$.In some textbooks a different definition is found: $h(u)-h(v) \leq d(u,v)$ for any edge $(u,v) \in V$.The two definitions are equivalent, as can readily be shown. In this paper, h is always assumed to be consistent.

Orienteering: It's a group of sports that needs navigational skills using a map and compass to navigate from point to point in very different from each other and usually unknown terrain at that time moving at speed.

Scenic view: It means a beautiful view that includes a river, lake, agriculture, mountain, or other visually impressive scenic views which attract visitors.

Bidirectional A*: Bidirectional search is a graph search algorithm that finds the shortest path from an initial vertex to a goal vertex in a directed graph. It runs two simultaneous searches: one forward from the initial state, and one backward from the goal, stopping when the two meet. As in A* search, the bi-directional search can be guided by a heuristic estimate of the remaining distance to the goal (in the forward tree) or from the start (in the backward tree). It runs two simultaneous searches-

- Forward search from source/initial vertex towards goal vertex.
- Backward search from goal/target vertex towards source vertex.

Methods: Heuristic: Its used when it's not possible(impractical) to solve a particular problem with a step-by-step algorithm. Because a heuristic approach emphasizes speed over accuracy, it is often combined with optimization algorithms to improve results. Distance from node to destination(Euclidean distance) Scenic view value between 0 to 100 .

IV. Proposed Model

Algorithm :

1: S = Ø;
2: R = Ø; // S and R are sets
3: M = V ; // M is a shared set
4: L = ∞; // L is a shared real value
5: for all v ∈V do
6: g(v) = ∞;
7: end for
8: g(s) = 0;
9: f = g(s) + h(s);
10: while any v ∈M has g(v) < ∞ do
11: u0 = arg min{g(v) + h(v) | v ∈M}; // u0 is selected
12: M = M − {u0};
13: if g(u0) + h(u0) − h(t) ≥ L or g(u0) + ˜f − h˜(u0) ≥ L then
14: R = R + {u0}; // u0 is rejected
15: else
16: S = S + {u0}; // u0 is stabilized
17: for all edges (u0, v) ∈E with v ∈M do
18: g(v) = min(g(v), g(u0) + d(u0, v));
19: L = min(L, g(v) + ˜g(v));
20: end for
21: end if
22: f = min{g(v) + h(v) | v ∈M};
23: end while

Equations:

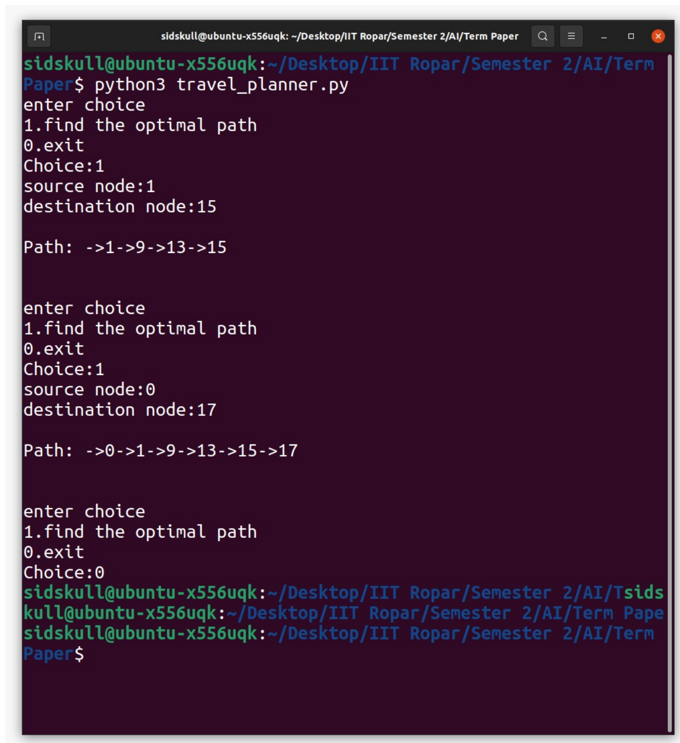
- Cost function = g(x) + h(x)
- g(x) = Dynamic travel time on an edge
- h(x) = Estimated time from next node to destination
- htime(x)=Manhattan distance to dest/ max speed limit in that region
- Will always be an under estimator and thereby admissible
- Edge weight distributions
- Individual distribution of travel time for each edge
- Euclidean distance:
  $d(p, q) = \sqrt{(q1 - p1)^2 + (q2 - p2)^2}$

V. Results and Discussions

In the result the algorithm two values to enter

- Source node
- Destination node

After entering the source and destination node,algorithm sets the source and destination and start searching most scenic and must visiting places between the mentioned source and destination. After searching the locations to visit algorithm provides a optimal route map, which shows directions to go from one place to another places as shown in the figure given.



In above figure when user gives his inputs as source=0 and destination=17, the algorithm provides a optimal path including locations such as 1,9,13,15,17, which means there are five location between the source and destination to visit apart from destination.

VI. Conclusions and Future work

Presently the model produced by us just provides the optimal path between the source and destination provided by the user. In future we will be adding other features such as

- User Preferences: It means the user can provide any other locations including source and destination so that the model add's those extra given locations as stops and provide a optimal path which starts at source joins all the locations given by users and end at the destination.
- Age constraints: In this we can take ages also as inputs and provide optimal route path based on the age groups. For example.

- input age$\leq$ 18: Model provides locations that gives knowledgeable places like Zoo's, Museums, Archaeological sites etc.
- input age$>$18 and input age$\leq$40: Model provides locations related to places where partying happens, adventurous, crazy nights etc.
- input age$>$40: Model provides locations where they can sit at a place,enjoy the scenic locations and relax.

REFERENCES

[1] Chao Chen, Liping Gao, Xuefeng Xie, and Zhu Wang. Enjoy the most beautiful scene now: a memetic algorithm to solve two-fold time-dependent arc orienteering problem. *Frontiers of Computer Science*, 14(2):364–377, 2020.

[2] Ying Lu, Gregor Jossé, Tobias Emrich, Ugur Demiryurek, Matthias Renz, Cyrus Shahabi, and Matthias Schubert. Scenic routes now: Efficiently solving the time-dependent arc orienteering problem. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 487–496, 2017.

[3] Cédric Verbeeck, Pieter Vansteenwegen, and E-H Aghezzaf. An extension of the arc orienteering problem and its application to cycle trip planning. *Transportation research part E: logistics and transportation review*, 68:64–78, 2014.