A graphical Approach for Personal Shopping Guide in Post Covid World

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR'S OF TECHNOLOGY

in

Computer Science and Engineering

by

JAGNANI REYAAN SUMIT (112015066)

Semester: IV

Computer Science and Engineeing

Indian Institute of Information Technology, Pune

Near Bopdev Ghat, Kondhwa Annexe, Yewalewadi, Pune, Maharashtra 411048

April 2022

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"A graphical Approach for Personal Shopping Guide in Post Covid World"** submitted by **JAGNANI REYAAN SUMIT** bearing the **MIS No : 112015066**, in completion of his project work under the guidance of **Dr. Bhupendra Singh** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering in the Department of Computer Science Engineering , Indian Institute of Information Technology, Pune, during the academic year 2021-22.

Dr. Bhupendra Singh

Project Guide
Assistant Professor
Department of CSE
IIIT Pune

Dr. Tanmoy Hazra
Head of the Department
Assistant Professor
Department of CSE
IIIT Pune

Project viva-voce held on 28/04/2022

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our beloved director, **Dr.Anupam Shukla**, for providing his kind support in various aspects.

I would like to express my gratitude to my project guide **Dr. Bhupendra Singh**, Professor, Department of CSE, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this B.Tech project.

I would like to express my gratitude to the head of department **Dr. Tanmoy Hazra**, Head of Department CSE, for providing his kind support in various aspects.

I would also like to thank all the faculty members in the Dept. of CSE and my classmates for their steadfast and strong support and engagement with this project.

Abstract

Shopping malls and shopping centres are among the busiest destinations in the world today. We evaluate ways for constructing an effective and efficient network of retail malls and demonstrate that a real-time solution for determining the best potential routes based on criteria such as weather, crowd density, budget, and so on is conceivable. We discovered that if there is no control over crowd density during a specific time period, people and employees are subject to spatial hazard. Graph theory overcomes the significant risk provided by planned shopper times and routes, and aids in the long-term creation of crowd-controlling models that can be utilised for crowd density estimation, prediction, and control. Using graph algorithms and pruning approaches, we created the Personal Shopping Guide system to accommodate tough eras such as Covid19 and the post-Covid world. Dijkstra, Kruskal (Minimum Spanning Tree), Disjoint Set Union, and Travelling Salesman Problem algorithm have no limitations and can be employed in complicated retail networks.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Project Intro

In today's society, stopping the spread of viruses and diseases is critical. One of the key reasons for this is the increased population density in public malls and retail centres. The global epidemic of Covid-19 necessitates great resilience in these networks. People and employees are exposed to spatial danger if there is no control over crowd density within a given time period, according to our findings. Graph theory helps in the long-term design of crowd-controlling models that may be used for crowd density estimation, prediction, and control by overcoming the major risk offered by planned shopping times and routes. In this paper, we argue that the concept of shopping recommendations contains a plethora of complex technical challenges that can be expressed as graph problems. To tackle diverse graph theoretic challenges, shopping guides can considerably benefit from the wealth of information and techniques gained in graph theory.

1.2 Project Outline

The project is organized as follows :

- Chapter 1 provides a general introduction to the project and a rough idea of the software development implementation.
- Chapter 2 introduces the necessary background, technical knowledge required to solve the task.
- Chapter 3 mainly gives a comprehensive summary of algorithms referred to and

used in our project.

- Chapter 4 discusses the basic design of the traffic optimization system. Here, we delve into the various algorithms used, methods implemented and algorithms discussion.
- Chapter 5 concludes our work and gives the future work which can be done to improve this scheme.

Chapter 2

Objectives

2.1 Developing a shopping assistance system that is effective, efficient, safer and user-friendly

Apps like Google Maps have revolutionised the way we travel and provided invaluable assistance to their users, but there is no such tool on the market that focuses on a more in-depth topic, such as a shopper guide. A user might have several characteristics that he or she must consider before heading to a shopping centre or mall, such as the list of items that he or she has to buy, their priorities, budget, travel distance, availability of a certain product at a given centre, and, especially in the post-covid world, the current crowd density at a particular centre and making sure that the weather is not extreme, which makes the process of determining the most optimum routes, and the order of visiting the shopping malls extremely difficult. The shopping aid service will be useful in this situation. The following are some of the places when using a shopping help service would be quite beneficial :

- **Limiting travel and disease spread** - By studying information provided by each shopping centre about its availability of items and prices, as well as information about the location of their centres, we can limit the extra and unnecessary travel that we may need to do in search of a specific product, and thus, we can reduce the spread of the virus and diseases. The store pathways and order can be determined ahead of time and given to the shopper.
- **Dependent Route Scheduling** - The paths and stores visited should be adapt-

able to changing conditions, such as inclement weather or store closing times. In the event of severe weather, a warning should be sent not to leave the house; nevertheless, if the shopper does decide to venture out, the best feasible approach should be to only visit establishments that are as close to the house as possible, allowing for fast assistance.

- **Cost Minimization** - By optimising the path that should be travelled and minimising any dangers, the shopper can also save a lot of money on transportation. The house and the centres can be thought of as nodes, and knowing the beginning node and the destination allows us to discover the most efficient approach to save time and money.
- **Controlling Crowd Density** - The one thing that everyone will want in the post-covid world is to keep their hygiene and social-distancing. No one wants to be in a crowded atmosphere, which is why, by analysing data on crowd density in shopping malls, we can recommend malls and shopping centres that are less crowded.

2.2 Eliminating needless travel to reduce the danger of virus spread.

In a graph, a bridge is defined as an edge that, when removed, causes the graph to become unconnected (or more precisely, increases the number of connected components in the graph). So, in times of crisis, we can remove particular channels or routes (nonbridges) while retaining connectivity across shopping centres to limit the chance of disease spread. This can also be accomplished by employing the Minimum Spanning Tree approach.

Chapter 3

Literature Review

Author	Title	Source	Findings
Pasi Fränti, Teemu Nenonen, Mingchuan Yuan	Converting MST to TSP Path by Branch Elimination	School of Computing, University of Eastern Finland, 80101 Joensuu, Finland 27 December 2020	Examined the approaches to find the best approximate travelling salesman problem solution my MST.
Ms.Ch,Sai Harika, Ms,D.Anusha	Use Of graph Theory In Transportation Problems and Different Networks	Journal of Applied Science and Computations Volume V, Issue XII, December/2018	Examined the traffic congestion problem and modelled a solution induced by graph theory
Jeevitha S, Rajeswari R	A Review of Crowd Counting Techniques	E ISSN 2348 –1269, PRINT ISSN 2349-5138	Methods which can be used in shopping malls to find the crowd density of that particular mall.
Muhammad Adeel Javaid	Understanding Dijkstra's Algorithm	Understanding Dijkstra Algorithm January 2013SSRN Electronic Journal	Gives a greedy algorithm to find the nearest paths during transportation from a fixed source

Chapter 4

Methodology

4.1 Pseudo Code

PSEUDO CODE

1. Input : Adjacency List
2. Input : Shopping Markets desired to Visit
3. If (weather is bad) :
   a. Warning and Recommendation of some of the malls to skip them.
   b. Input : If I want to skip or not.
      i. If (No) : Go to Step 3c
      ii. Else : skip the malls and go to step 3c
   c. dijkstra()
   d. Go to step 8
4. Input the preference of user
5. If(Money is preference)
   a. Sort by Money
   b. find_path()
   c. Go to step 8
6. If(Crowd Density)
   a. Sort by Crowd Density
   b. find_path()
   c. Go to step 8
7. Minimum_Spanning_Tree()
   a. Travelling_Salesman_Path()
8. END.

Figure 4.1: Pseudo Code

4.2 Pre-Processing :

Getting data from the database and feeding it into the code (network representing the shopping centres and malls). Regardless of the format of the input, it must be saved as a graph (adjacency list representation). The House of the Shopper Node is similarly pre-determined. The distance between this node and all other malls will be calculated and saved for future use. For this, we'll use Dijkstra's algorithm. To save time in the future, the shortest path between any two nodes will be determined in advance.

```cpp
void dijkstra(vector<vector<vector<ll>>> &adj, vector<vector<ll>> &dis, vector<vector<string>> &path, ll curr)
{
    priority_queue<vector<ll>, vector<vector<ll>>, greater<vector<ll>>> q1;
    q1.push({0,curr,-1});
    while(!q1.empty())
    {
        ll node = q1.top()[1];
        ll par = q1.top()[2];
        ll distance = q1.top()[0];
        q1.pop();
        if(dis[curr][node]!=LLONG_MAX) continue;
        if(par==-1) path[curr][curr] = to_string(curr);
        else path[curr][node] = path[curr][par] + "->" + to_string(node);
        dis[curr][node] = distance;
        for(ll i=0; i<adj[node].size(); i++)
        {
            ll child = adj[node][i][0];
            ll edge = adj[node][i][1];
            if(dis[curr][child]!=LLONG_MAX) continue;
            q1.push({distance + edge, child, node});
        }
    }
}
```

4.3 Processing :

The graph will go through multiple different algorithms depending on the user's requirement to optimise the path finding method.

4.3.1 Weather

If the weather is terrible, the user will be warned, and it will be suggested that some of the malls on the list that are quite far away from home be removed from the list. The user must next decide whether or not to reduce the overall number of malls to visit. The path-finding algorithm is then invoked, and the proposed path is displayed.

4.3.2 Preference - Less Price First

If the shopper's preference is cost-effective, the shops will be ordered according to how much money they spent there. The path finding algorithm will then be called, and the recommended path will be displayed.

4.3.3 Preference - Less Crowd Density

If the customer prefers a less crowded environment, the shops will be classified according to their respective crowd density. The path-finding algorithm is then invoked, and the proposed path is displayed.

4.3.4 Preference - Less Travelling

```
class Union
{
public:
    vector<ll> _rank,par;
    void initialize(ll n)
    {
        _rank.assign(n+1,1);
        par.assign(n+1,0);
        for(ll i=1; i<=n; i++)
        {
            par[i]=i;
        }
    }
    ll find(ll a)
    {
        if(par[a]==a) return a;
        else return par[a] = find(par[a]);
    }
    void merge(ll a, ll b)
    {
        a = find(a);
        b = find(b);
        if(a==b) return;
        if(_rank[a]<_rank[b]) swap(a,b);
        par[b] = a;
        _rank[a]+=_rank[b];
    }
};
```

- Nodes that do not need to be visited will be temporarily deleted from the adjacency list.
- We'll use Kruskal's Minimum Spanning Tree Algorithm to turn this graph into a

tree. Then, to identify the desired path, we'll use the Traveling Salesman Problem's Algorithm.

- Because we're using an approximate approach, these stages may not always lead to the most ideal path. However, the approximate approaches only function if the issue instance is Triangle-Inequality compliant.

  Triangle-Inequality : The least distant path to reach a vertex j from i is always to reach j directly from i, rather than through some other vertex k (or vertices).

- The cost of best possible Travelling Salesman tour is never less than the cost of MST. (The definition of MST says, it is a minimum cost tree that connects all vertices).

  The total cost of full walk is at most twice the cost of MST (Every edge of MST is visited at-most twice)

```cpp
#include"dsu.cpp"
void mst(vector<vector<vector<ll>>> &adj, unordered_map<ll,ll> &mall,
    vector<vector<ll>> &distance, ll home)
{
  ll n = adj.size();
  n--;
  vector<vector<ll>> edges;
  vector<vector<vector<ll>>> temp(n+1);
  for(ll i=1; i<=n; i++)
  {
    for(ll j=1; j<=n; j++)
    {
      if(i==j) continue;
      if((mall.find(i)!=mall.end() || home==i) && (mall.find(j)!=mall.
  end() || home==j))
        edges.pb({distance[i][j],i,j});
    }
  }
  sort(all(edges));
  Union u1;
  u1.initialize(n);
  for(ll i=0; i<edges.size(); i++)
  {
    ll a1 = u1.find(edges[i][1]), a2 = u1.find(edges[i][2]);
    if(a1==a2) continue;
    u1.merge(a1,a2);
    temp[edges[i][1]].pb({edges[i][2], edges[i][0]});
    temp[edges[i][2]].pb({edges[i][1], edges[i][0]});
  }
  swap(adj, temp);
}
```

4.4 Expected Outcome :

After the algorithm has completed, we will print the desired path result to the consumer based on his or her preferences.

```
void get_path(vector<ll> &vect, vector<vector<string>> &path, ll &home)
{
  vect.pb(home);
  cout<<path[home][vect[0]];
  ll prev = vect[0];
  for(ll i=1; i<vect.size(); i++)
  {
    string s = path[vect[i-1]][vect[i]];
    ll j = 0;
    while(j<s.length() && s[j]!='-') j++;
    while(j<s.length())
      cout<<s[j++];
  }
}
```

Chapter 5

Conclusion and Future Work

While the current effort prioritises the COVID scenario, algorithms such as dijkstra, kruskal, and kosaraju have no bounds and can be employed in complicated mall networks with nearly any level of load.

The Kosaraju algorithm, which has been shown to be highly beneficial in maintaining seamless connection, can be used to airports, railroad networks, and road transport utilising the method utilised in this research.

In the future, if this project is implemented on a large scale, we can use some famous machine learning models to predict the crowd density according to a specific time and we can simulate it according to the current time.

The project can be fine-tuned to place a greater emphasis on strategically and geographically significant regions.

Chapter 6

References

1. An efficient distributed algorithm for finding articulation points, bridges, and biconnected components in asynchronous networks. Mohan Raju, Yahui zhu International Conference on Foundations of Software Technology and Theoretical Computer Science. Link

2. Use Of Graph Theory in transportation problems and different networks. Ms. Ch Sai Harikam, Ms. D Anusha. Jorunal of Applied Science and Computations Volume V, Issue XII, December/2018. Link

3. A Review of Crowd Counting Techniques. Jeevitha S, Rajeswari R. E ISSN 2348 1269 PRINT ISSN 2349-5138 Link

4. Research Paper on Travelling Salesman Problem And it's Solution Using Genetic Algorithm. Kartik Rai, Lokesh Madan, Kislay Anand. 2014 IJIRT — Volume 1 Issue 11 — ISSN: 2349-6002 Link

5. Analysis of Strongly Connected Components (SCC) Using Dynamic Graph Representation. Saleh Alshomrani, Gulraiz Iqbal IJCSI International Journal of Computer Science Issues, July 2012. Link

6. Converting MST to TSP Path by Branch Elimination. Pasi Fränti, Teemu Nenonen, Mingchuan Yuan. School of Computing, University of Eastern Finland, 80101 Joensuu, Finland. Published: 27 December 2020. Link

7. Approximation Algorithm for Travelling Salesman Problem. Abhijit Tripathy. Metric Travelling Salesman Problem and an approximation algorithm named 2 ap-

proximation algorithm, that uses Minimum Spanning Tree in order to obtain an approximate path. Link

8. A Persistent Union-Find Data Structure. Sylvain Conchon Jean-Christophe Filliˆatre LRI, Univ Paris-Sud, CNRS, Orsay F-91405. Link