An Outlook For Traffic Congestion Using Tunneling Technology

P. Ashish[a], K. Gopi Krishna[b] ,Dr. V. Sireesha[c] , Dr. Nagaratna P Hegde[d]

[a,b,c,d] Vasavi College of Engineering(A), Hyderabad, India

Generally, we see a lot of traffic during these days. Traffic congestion is seen as the major issue that most civilians are facing despite measures being taken to reduce and control it. This is one of the most challenging situations for engineers, and planners. We have seen a rapid growth in vehicle ownership which has established congestion as an inescapable truth of urban life. Many attempts have been made to develop congestion reduction indices for heavily motorized countries. Even after the Metro Rail System has included in many metropolitan cities, we did not observe much change.Therefore, in our paper we have addressed this issue with the help of a system which transports the vehicles across various cities through tunnels which are laid underground. In this regard, our project aims to develop a prototype to alleviate traffic congestion and enable rapid transit across densely populated areas.

Proposed System

The main objective of this proposed work is to develop a technique which reduces traffic congestion in urban areas. Here the user is asked to select the tray to travel. These trays are implemented using queues and if the other user is selected the same tray, he is asked to wait for a specific time. This time is calculated on basis of the first user's travel time and once it is cleared the next user can board the tray or else, he is asked to choose the other tray if he/she does not want to wait. The user travels in the shortest path among multiple paths which is designed using Dijkstra's greedy algorithm where we are greedy with respect to path and time of travel. When the user enters his source and destination, this function chooses the shortest path among all the paths which is implemented using Min Heap whereas to store the graph's cities and distances between them we used adjacency list.

The shortest path is implemented in sample function in figure Fig. 1.

```
int getDistance(struct Graph* graph, int src,int dest){
        int V = graph->V;//n.o of vertices
        parent[src]=-1;
        int dist[V];
        struct MinHeap* minHeap = createMinHeap(V);
        for (int v = 0; v < V; ++v)
        {
                dist[v] = INT_MAX;
                minHeap->array[v] = newMinHeapNode(v,dist[v]);
                minHeap->pos[v] = v;
        }
        minHeap->array[src] =newMinHeapNode(src, dist[src]);
        minHeap->pos[src] = src;
        dist[src] = 0;
        decreaseKey(minHeap, src, dist[src]);
        minHeap->size = V;
        while (!isEmpty(minHeap))
        {
                struct MinHeapNode* minHeapNode =extractMin(minHeap);//minimum in dist array which is unvisited
                int u = minHeapNode->v;
                struct node* temp3 =graph->array[u].head;
                while (temp3 != NULL)
                {
                        int v = temp3->vertex;
                        if (isInMinHeap(minHeap, v) &&dist[u] != INT_MAX &&temp3->weight + dist[u] < dist[v])
                        {
                          parent[v]=u;
                                dist[v] = dist[u] + temp3->weight;
                                decreaseKey(minHeap, v, dist[v]);
                        }
                        temp3 = temp3->next;
                }
        }
        return dist[dest];
}
```

Fig. 1. Function which gives the shortest path for travel

The cost of travel is based on the time of travel. The function get_time() is implemented which includes the shortest distance algorithm and speed of the moving tray.

The function is shown in figure Fig. 2.

```
float get_time(struct Graph* graph,int src, int dest){//returns time of travel
  int d;
  float timeOfTravel=0.0;
  d=getDistance(graph,src,dest);
  timeOfTravel=((float)d)/3.0;
  return timeOfTravel;
}
```

Fig. 2. Function which calculates the time of travel.

ALGORITHM

There will be a unique file for each tray, we call it tray queue. Which has the following attributes. At any given time 't', all the users info whoever requested for that tray is displayed on it. And the user's info who is using that tray at 't' is present on the first line of that file. For instance, let's consider 'n' number of users are requesting for tray numbered as 14 (tray number 1 at location Hogwarts [4]) and users are u1, u2….un. The first user (u1) who requested for that tray gets approved and accessed to the tray. So, there are two options for users (u2, u3 …. un) i.e., either wait for that tray or select another tray. In first case, the waiting time for nth user becomes (t1 + t2 +….t n-1). Where t n is the time required for completion of journey for nth user. In second case, there is no waiting time since the user picks a tray which is currently not used by anyone.
Here, we are greedy with respect to the first requested user and approving his request. Since, the bill is calculated based on the time user spends in tunnels the overall profit remains same. So, if we keep on approving on this basis no tray is kept empty at any given time 't' and overall time is optimized. If we take a look on larger scale like a day's analysis the profit is also optimized since every second matters.

There will be trays which carries them through tunnels. The user who ever feels the road traffic is heavy can enter the tray which carries the user into the tunnels. These tunnels are paths connected from one city to other internally. So, the user chooses his/her source and destination and enter the tray. Within stipulated limit of time the user reaches his destination, and the tray would be empty so that next person who is waiting in the queue for the same tray can enter the tray and so his journey begins. At his destination he/she would be charged an amount for his travel which can be debited through his wallet. The user can also deliver packages. The user can also opt to travel through boring buses.

Experimentation and Results

i)Data

For implementation, we utilized concept of graph data structure, shortest path algorithm for computing path and cost of travel. We have used heap to implement the Dijkstra's algorithm and used concept of threads for communication between trays. In this paper, we have shown the implementation of tray queue and travel history of the user along with his wallet balance.

ii)Algorithm description

There will be a unique file for each tray, we call it tray queue. Which has the following attributes. At any given time 't', all the users info whoever requested for that tray is displayed on it. And the user's info who is using that tray at 't' is present on the first line of that file. For instance, let us consider 'n' number of users are requesting for tray numbered as 14 (tray number 1 at location Hogwarts [4]) and users are u1, u2….un. The first user (u1) who requested for that tray gets approved and accessed to the tray. So, there are two options for users (u2, u3 …. un) i.e., either wait for that tray or select another tray. In first case, the waiting time for nth user becomes (t1 + t2 +….t n-1). Where t n is the time required for completion of journey for nth user. In second case, there is no waiting time since the user picks a tray which is currently not used by anyone. Here, we are greedy with respect to the first requested user and approving his request. Since, the bill is calculated based on the time user spends in tunnels the overall profit remains same. So, if we keep on approving on this basis no tray is kept empty at any given time ‘t’ and overall time is optimized. If we look on larger scale like a day’s analysis the profit is also optimized since every second matters

iii) Results:

Case I: Since the queue is empty, whoever books this tray will be asked to enter the tray and proceed to his journey. Once he books the tray, he will be added to the file which denotes he is travelling in that particular tray at a particular time ‘t’ and when his journey completes, he will be de-queued from the queue automatically according to time.

```
WHAT DO YOU LIKE TO DO
1.TO BOOK A BORING TRAY FOR YOUR CAR
2.TO BOOK A BORING BUS
3.TO DELIVER A PACKAGE
4.CHECK YOUR PREVIOUS HISTORY
5.CLEAR YOUR PREVIOUS HISTORY
6.TO CHECK YOUR WALLET BALENCE
7.LOGOUT
1

You can Enter and Exit Trays from regions listed here
0.HOGSMEADE STATION
1.BLACK LAKE
2.SHRIEKING SHACK
3.QUIDDITCH PITCH
4.HOGWARTS
5.HOGSMEADE VILLAGE
6.HAGRID'S HUT
7.FORBIDDEN FOREST
ENTER SOURCE AND DESTINATION INDICES
0 7
THERE ARE 4 TRAYS AVAILABLE AT SELECTED LOCATION
PICK ANY ONE OF THEM AND ENTER TRAY NUMBER
1
QUEUE IS EMPTY PLEASE PROCEED TO THE TRAY
PRESS 1 ONCE YOU ENTERED THE TRAY
1

WELCOME TO THE TUNNELS YOU WILL BE TRAVELLING AT A SPEED OF 180 KM/HR
YOU WILL BE TRAVELLING VIA Hogsmeade Station....Hogwarts....Hagrid's Hut....Forbidden Forest..
YOUR TOTAL TIME OF TRAVEL IS 7.000000 MINUTES
AND COST OF YOUR TICKET IS 210.000000
```

Fig.3. Case when the queue is empty.

```
14
1   gk FROM:4 TO:7
```

Fig 4. Case when the queue is not empty

Let us assume that a specific user ‘gk’ is travelling from 4(Hogwarts) to 7(Forbidden Forest). At this time if another user comes, he has to pick among two options either he wants to wait for the tray as shown in figure Fig. 3 or pick another tray as shown in figure Fig. 5.

Option-1: If the user wishes to wait

Fig. 3. If the user wishes to wait

When a specific user ‘gk’ completes his journey, the file will become empty and the waiting user will be added to the file, asked to enter the tray and his journey begins as previous case as shown in figure Fig. 4

```
ENTER SOURCE AND DESTINATION INDICES
4 7
THERE ARE 4 TRAYS AVAILABLE AT SELECTED LOCATION
PICK ANY ONE OF THEM AND ENTER TRAY NUMBER
1
THE REQUESTED TRAY IS BUSY PRESS 1 TO WAIT FOR TRAY PRESS 2 TO SELECT ANOTHER TRAY
1
YOU HAVE BEEN ADDED TO QUEUE PLEASE WAIT THE TRAY IS BUSY
ITS YOUR TURN NOW PLEASE ENTER THE TRAY
PRESS 1 ONCE YOU ENTERED THE TRAY
1

WELCOME TO THE TUNNELS YOU WILL BE TRAVELLING AT A SPEED OF 180 KM/HR
YOU WILL BE TRAVELLING VIA Hogwarts....Hagrid's Hut....Forbidden Forest..
YOUR TOTAL TIME OF TRAVEL IS 2.666667 MINUTES
AND COST OF YOUR TICKET IS 80.000002
YOUR JOURNEY IS COMPLETED
```

Fig. 4. Journey of the second user starts after first user completes his journey when belonged to same tray.

Option-2: If the user wants to select another tray.

```
ENTER SOURCE AND DESTINATION INDICES
4 7
THERE ARE 4 TRAYS AVAILABLE AT SELECTED LOCATION
PICK ANY ONE OF THEM AND ENTER TRAY NUMBER
1
THE REQUESTED TRAY IS BUSY PRESS 1 TO WAIT FOR TRAY PRESS 2 TO SELECT ANOTHER TRAY
2
SELECT ANOTHER TRAY
3
QUEUE IS EMPTY PLEASE PROCEED TO THE TRAY
PRESS 1 ONCE YOU ENTERED THE TRAY
1

WELCOME TO THE TUNNELS YOU WILL BE TRAVELLING AT A SPEED OF 180 KM/HR
YOU WILL BE TRAVELLING VIA Hogwarts....Hagrid's Hut....Forbidden Forest..
YOUR TOTAL TIME OF TRAVEL IS 2.666667 MINUTES
AND COST OF YOUR TICKET IS 80.000002

YOUR JOURNEY IS COMPLETED
```

Fig. 5. If the user does not want to wait instead board another available tray.

The specific user ‘gk’ history will be saved to a file automatically in the background along with his wallet balance “w”.

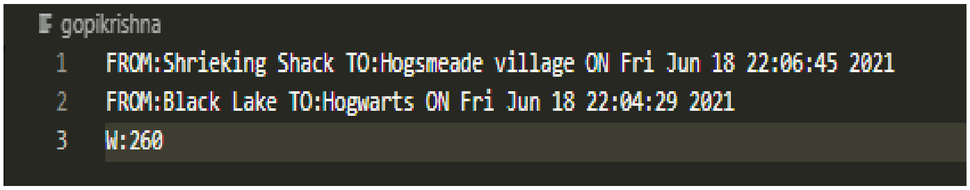

Fig.6. Travel history of the user will be saved in the respective file denoted by the name of the user.

Acknowledgments

We whole heartedly thank Mr. Elon Musk for contributing so much to this world and make our life comfortable with his new ideologies.
We would also like to express our special gratitude to our mam Dr.V.Sireesha who supported and guided us in making this project successful.

References

1. A Review of The Literature on The Evaluation and Roles of Tunnels Technology in Managing Environmental Disaster by Nuhu Isah and Maimunah Ali, Department of Technology, Management Faculty of Technology Management and Business , Universiti Tun Hussein Onn, Malaysia. Issue:2017
2. Tunnel Systems as a Critical Alternative in Solving Transport Problems by Vladimir V.Makarov, Far Eastern Federal University, Vladivostok 690090, Russia. Issue published : 2017.
3. A Survey of Road Traffic Congestion Measures towards a Sustainable and Resilient Transportation System by Tanzina Afrin and Nita Yodo, Department of Industri-al and Manufacturing Engineering, North Dakota State University,1410 14th Avenue North, Fargo, ND 58102, USA. Issue published:2019.
4. An Empirical Study of the Effects of Road Tunnel on Driving Performance by Alessandro Calvi, Maria Rosaria De Blasiis, Claudia Guattari, Department of Scienc-es of Civil Engineering, University Roma Tre, Via Vito Volterra, Rome, Italy. Issue published :2018.
5. Basic Traffic Engineering Concepts, Zainab Ahmed Alkaissi, Al-Mustansiriya University, Baghdad, Iraq. Issue published:2020.
6. In-vehicle information systems to improve traffic safety in road tunnels, Geva Greenfield, Imperial College London, David Shinar , Ben-Gurion University of the Negev, Issue published : 2018.
7. Video based image analysis for tunnel safety :A tunnel video surveillance and traffic control system, Helmut Schwabach, Horst Bischof, Graz University of Tech-nology, Issue: 2019.
8. A case study on Tunnel traffic analysis by simulation at the peak traffic time of vehicles, Amir Abbas Shojaie, Issue: 2016.
9. Solutions for Urban Traffic Congestion, Chao Dong, School of Economics and Business, Beijing Jiaotong University, Beijing, China, Issue: 2016.
10. Gao S, Frejinger E, BenAkiva M., Adaptive route choice in risky traffic networks: A prospect theory approach [J]. Transportation Research C.2010,18(5):727-740.

Conclusion And Future Scope

People fore see a great increase in underground construction, numerical estimates are crude at best. Key factors affecting the actual increase are technological improvements reducing costs and an increasing awareness on the part of society and public-works planners of the many potential applications for better use of the underground.

In conclusion, our program is a prototype which provides user in accomplishing a user-friendly journey in reducing time and confusion.