Effective Searching Shortest Path in Graph Using Prim's Algorithm

D.Kalpanadevi[#1]
Assistant Professor & Department of Mathematics(CA), G.V.G Visalakshi College for Women Udumalpet, India

Abstract—Prims algorithm is studied the shortest path problem in the greedy method which is used to select a subset of the edge such that spanning tree is formed and the weight of the edges is minimal. There are various shortest path methods available. An algorithm is designed based on which, a new method in greedy method is proposed which is more effective and efficient than other well-known method. At each stage of the undirected graph it make a decision that appear to be best at the time, also made at one stage is not changed in a later stage, so each decision should assure feasibility.

Keywords— spanning tree, edge, greedy, shortest path.

I. INTRODUCTION

Prim Algorithm can be used in situation where the user preference includes a variety of criteria such as highway road shortest path distance, when we start and less time to achieve the destination. Further the logic of classifying a document as acceptable or not is too complex such as multi criteria selection of path find.

i) To find the shortest path distance when we choose a starting location, the prim's algorithm that find a minimum spanning tree for a undirected connected weighted graph G(V,E) this means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edge in the tree is minimized.

Using Euclidean distance

$dw=\sqrt{\sum_{i=1}^{n}(w_j-w_i)^2}$, to find the length of the weighted walk S = v0, e1, v1, e2, v2, ., en, vn in G is the sum of the distance G(S) = w(e1) + w(e2) + · · · + w(en

PRIM'S ALGORITHM

A greedy algorithm is a technique for solving problems with the following properties:

1. The problem is an optimization problem, to find the solution that minimizes or maximizes some value (weight/time).
2. The solution can be constructed in a sequence of steps/choices. The choice must be feasible, it looks as good or better than alternatives and the choice cannot be revoked.

A spanning tree of a connected graph is a tree containing all the vertices. A minimum spanning tree of a weighted graph is a spanning tree with the smallest weight and the weight of a spanning tree is the sum of the edge weights. Prims algorithm make it necessary to provide each vertex not in current tree with the information about the shortest path connecting the vertex to a tree vertex.

Prims algorithm constructs a minimum spanning tree through a sequence of expanding subtrees. The initial subtree in such a sequence consists of a single vertex selected arbitrarily from the set V of the graph's vertices. On each iteration, we expand the current tree in the greedy manner by simply attaching to it the nearest vertex not in that tree. The algorithm stops after all the graph's vertices have been included in the tree being constructed. Since the algorithm expands a tree by exactly one vertex on each its iteration s, the total number of such iteration is n-1, where n is the number of vertices in the graph.

Algorithm

// Returns the MST by Prim's Algorithm
// Input: A weighted connected graph G = (V, E)
// Output: Set of edges comprising a MST of G
$V_T \leftarrow$ {any vertex in G}
$E_T \leftarrow \phi$;
for i $\leftarrow$ 1 to |V | − 1 do

e $\leftarrow$ the minimum-weight edge (v, u)
with v $\in$ V_T and u $\in$ V − V_T

$V_T \leftarrow V_T \cup \{u\}$
$E_T \leftarrow E_T \cup \{e\}$

return E_T

This algorithm repeatedly chooses the smallest-weight edge from the tree so far to the other vertices. If a spanning tree has a weightier edge between V_T and V − V_T , it can be improved by replacing it with e.

We can put V_T edges into a priority queue, then dequeue edges until one goes between V_T and $V - V_T$. Prim's Algorithm using a binary heap to implement a priority queue is O(E log E) = O(E log V).

PROPOSED WORK

To provide users to visit the tour map and find the shortest distance what the visitor need to selecting the location from one place to another through given graph. Such that each vertex in a graph exactly visit once and finish at the vertex started from.
Suppose the user need to find the highway road path be search from the graph G(V,E), we give a preference (value) of such high path in graph as G(V,E,W,H) where V represent Vertices, E represent Edges, W represent Weight or Distance, H represent Highway path, under an algorithm H can handle the input value as 1.

Returns the MST by Prim's Algorithm
// Input: A weighted connected graph G = (V, E, W, 1 or 0)

if visited[i]==0,then
if d[i]<mincost, then
mincost=d[i];
repeat;

// Output: Set of edges comprising a MST of G high way road or shortest path
V ← {any vertex in G}
E ← ϕ;
for i ← 1 to |V | − 1 do
for i ← 1 to |E | − 1 do
weight[Vt_1][Vt_2][0]=weight[Vt_2][Vt_1][0]=w;
weight[Vt_1][Vt_2][1]=weight[Vt_2][Vt_1][1]=h;

While totalvisited!=v do
If weight[distance][i][0]!=0, then
If visited[i]= =0
If hway != 0
If d[i]>weight[distance][i][0] and weight[distance][i][1] != 0
d[i]=weight[distance][i][0];
p[i]= distance
else
If d[i]>weight[current][i][0] then
d[i]=weight[current][i][0];
p[i]=current;
mincost=32767;
for i ← 1 to |V | − 1 do
current=i;
visited[current]=1;
totalvisited++;

From the given graph, let us assume 0 represent to find a shortest path and 1 represent to find the high way road from the graph, this option is chosen from user input.

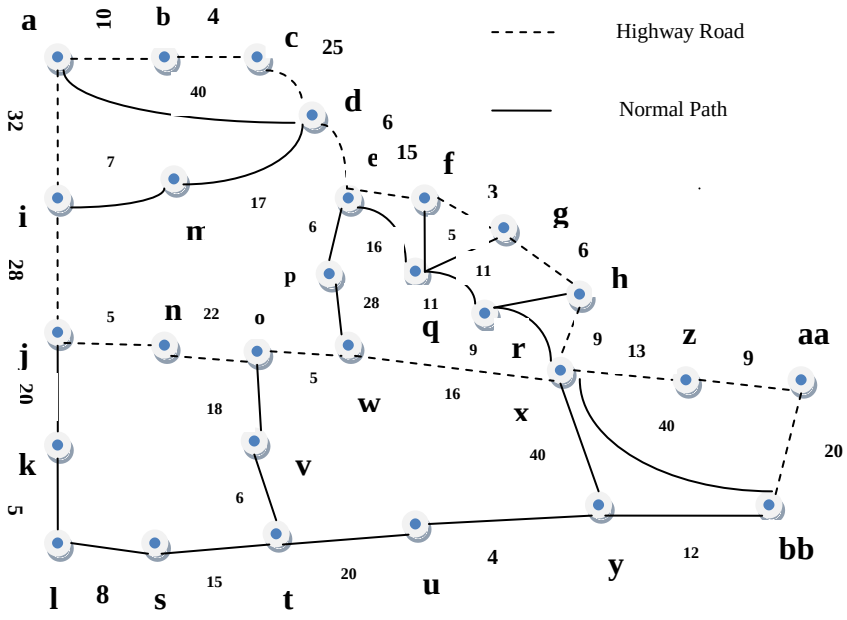

Fig1: Graph for finding shortest distance

EXPLANATION:

Step 1: Input the number of vertices and edges enter the specify graph.
Step 2: Get the option whether to find highway road or minimum shortest path.ie 1 or 0.
Step 3: Check the choice 1 or 0 and input the several vertices1, vertices2, distance and 1 or 0.
Step 4: If the choice is 1, the vertices analysis the highway road of shortest path in the given graph .
Step 5: If the choice is 0, the vertices analysis the normal path given in the graph fig1.
Step 6: Finally find the path of vertices, calculate the minimal path of the total distance and connected graph of vertices respectively.

IMPLEMENTATION:

Let us assume a->i = i->j = j->n = n->o = o->w = a->b = b->c = c->d = d->e = e->f = f->g = g->h = h->x = w->x = r->z = z->aa = aa->bb are highway road and other roads are normal path.
Suppose the visitor wants to know the minimal path between e (kumbakonam) to x(mayiladuthurai), if they choose 1 in option for highway road the path show the shortest distance as

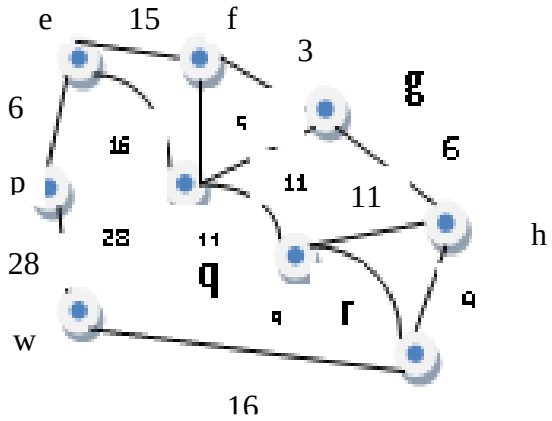

Fig 2: Selected graph for searching the shortest path

The result will be e->f->g->h->x->w due to all vertices visit atleast one time and minimum path distance weight is 49.
Suppose the visitor need to know the minimal path between the same locations without highway road path, they choose a choice as 0, then display the path as e->p, f->g->h->x->w, x->r->q and minimum path distance weight is 64.

CONCLUSION:

In this paper, prim's algorithm is used to find the minimum shortest path distance between the two location, User has to decide, visit the place and reach the destination as early of their planning with in time duration. From the implementation, the two results are taken for the minimum path distance and the highway path to follow the route of the vertices to reach the destination in efficient manner. Under the result, high way road is the best way to drive for reach the destination. The node selection actually depends on the graph itself. Finally, the Prim's algorithm run efficient not only on optimal path found in graph and also it run effective to find the optimal highway road rather than shortest path in normal road way and may also to find the shortest path of traffic clear road and achieve the destination with in the low time period.

REFERENCE:

[1]. G P Raja Sekhar, Design & Analysis of Algorithms Department of Mathematics, II T Kharagpur.

[2]. V.Sundaresan,K.S Subramanian, and K.Ganesan, Discrete Mathematics, A.R Publication 2002.

[3]. Abuja'farMhammadibnMusaal-khwarizmi, Design and analysis of Algorithm, http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Greedy/prim.htm

[4]. Sangam Jain Professor*:* Partha Bhowmik, Prim's Algorithm ,Department of Computer Science and Engineering IIT Kharagpur, August 21, 2008.