International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 01 | Jan 2022 www.irjet.net p-ISSN: 2395-0072

Dijkstra Shortest Path Visualization

Rohit Gurav[1], Sejal Hukare[2], Sakshi Suryawanshi[3], Parth Narkhede[4], Sankalp Patil[5], Vinay Thamke[6]

[1]Rohit Gurav, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Maharashtra , India
[2]Sejal Hukare , Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology , Maharashtra , India
[3]Sakshi Sanjay Suryawanshi, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Maharashtra, India
[4]Parth Satish Narkhede, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Maharashtra, India
[5]Sankalp Patil , Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Maharashtra, India
[6]Vinay Thamke, Department of Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Maharashtra, India

---***---

Abstract - This paper shows the implementation and visualization of Dijkstra Shortest Path Visualization algorithm using python turtle.

Key Words:**Dijkstra, Shortest Path, Vertices, Edges , Traverse , Matrix**

1.INTRODUCTION

Dijkstra's algorithm rule is incredibly just like Prim's algorithm rule for minimum spanning tree. Like Prim's mst, we have a tendency to generate a SPT (shortest path tree) with a given supply as a root. we have a tendency to maintain 2 sets, one set contains vertices enclosed within the shortest-path tree, alternative sets include vertices not nevertheless enclosed within the shortest-path tree. At each step of the algorithmic rule, we discover a vertex that's within the alternative set (set of not nevertheless included) and includes a minimum distance from the supply.

The code calculates the shortest distance however doesn't calculate the trail data. we will produce a parent array, update the parent array once distance is updated (like prim's implementation) and use it to indicate the shortest path from supply to completely different vertices.The code is for planless graphs, identical Dijkstra operate are often used for directed graphs additionally.The code finds the shortest distances from the supply to any or all vertices. If we have a tendency to have an interest solely within the shortest distance from the supply to one target, we will break the for loop once the picked minimum distance vertex is adequate to the target.

1467 Mohamed Hamada / Procedia applied science eighteen (2013) 1466 – 1474

In coming up with our simulators, we have a tendency to thought-about the active construction learning model [4, 12] that includes a range of basic style principles that embrace the following:

1. Lecturers act as facilitators not as information transmitters. This implies information should be actively constructed by learners, not passively transmitted by lecturers.

2. Learning ought to manifest itself in a very cooperative setting. To show the effectiveness of our simulators as a model of associate degree interactive learning tool, some experiments were carried out. The preliminary results of those experiments showed that exploitation our simulators not solely improved the learners' performance however additionally improved their motivation to actively participate within the learning method of the related subjects and get additional information on their own.

The paper is organized as follows. Following the introduction, section 2 introduces connected work. Section three offers an outline of the methodology of the dijkstra algorithm rule. we'll discuss the event of our simulators in section four. The performance analysis of the setting are conferred in section 5. Section six can conclude the paper and discuss future work.

2. LITERATURE REVIEW

While individuals like network designers and analysts ought to have an intensive understanding of Dijkstra's algorithm rule, a straightforward shut examination is spare for the remaining people. Rather than listing the algorithmic rule in stepwise type, let's merely practise a sample resolution. The goal of our example is to seek out, in Figure below, the least-cost routes from Node A to every of the opposite nodes.

2.1 APPLICATION

Before learning any algorithmic rule, we must always understand the basic purpose of exploiting associate degree algorithmic rule that might facilitate the United States of America in real-world applications. Such as, for Dijkstra's algorithm rule, we have a tendency to try to seek out the solutions to least path primarily based issues.

For example, if someone desires to travel from town A to town B wherever each city's area unit is connected with varied routes. That route usually he/ she ought to choose?

Undoubtedly, we might adopt the route because we have a tendency to reach the destination with the smallest amount of potential time, distance and even value.

Further, with the discussion, it's varied real-world use cases, a number of the applications area unit the following:

For map applications, it's vastly deployed in measurement of the smallest amount of potential distance and check direction amidst 2 nation-states like Google Maps, discovering map locations to inform the vertices of a graph, hard traffic and delay-timing, etc.

For phone networks, this can be additionally extensively enforced within the conducting of information in networking and telecommunication domains for decreasing the obstacle taken place for transmission.

Wherever addressing the necessity for shortest path explications either within the domain of artificial intelligence, transport, embedded systems, laboratory or production plants, etc, this algorithmic rule is applied.

Besides that, alternative applications area unit road conditions, road closures and construction, and science routing to sight Open Shortest Path 1st.

2.2 ADVANTAGES AND LIMITATIONS

Advantages:

1) One of the most blessings of its very little complexity is that it is sort of linear.

2) It may be used to calculate the shortest path between one node to all or any alternative nodes and one supply node to one destination node by stopping the algorithmic rule once the shortest distance is achieved for the destination node.

3) It solely works for directed-, weighted graphs and every one edge ought to have non-negative values.

Despite various applications and advantages, Dijkstra's algorithm has disadvantages also, such as;

1) It will associate obscured exploration that consumes loads of your time whereas process,

2) It is unable to handle negative edges,

3) As it heads to the acyclic graph, thus can't reach the correct shortest path, and

4) Also, there's a requirement to take care of trailing vertices, are visited

3. METHODOLOGY

Given N nearest vertices to the supply and their shortest distance from supply, a way to realize the N + one th nearest vertex to the supply and it's shortest distance? If we will figure that out, we will realize the shortest path to any vertex.

Dijkstra's algorithm rule will do precisely that. It finds the shortest distances of all vertices so as to their closeness to the supply. however will it do that? Given N nearest vertices and their distance, the N + one th nearest vertex should be one that may be reached by following only one edge from some vertex of N nearest ones. That statement is true if the graph contains only edges of non-negative weights, that's why Dijkstra's algorithm rule doesn't work for graphs with negative edges.

Algorithm

1) Produce a group sptSet (shortest path tree set) that keeps track of vertices enclosed within the shortest-path tree, i.e., whose minimum distance from the supply is calculated and finalized. Initially, this set is empty.

2) Assign a distance worth to any or all vertices within the input graph. Initialize all distance values as INFINITE. Assign distance worth as zero for the supply vertex so it's picked 1st.

3) Whereas sptSet doesn't embrace all vertices

....a) choose a vertex u that isn't there in sptSet and includes a minimum distance worth.

....b) embrace u to sptSet.

....c) Update distance worth of all adjacent vertices of u. To update the space values, reiterate through all adjacent vertices. for each adjacent vertex v, if the addition of distance worth of u (from source) and weight of edge u-v, is a smaller amount than the space worth of v, then update the space worth of v.

3.1 EQUATION

A. Recursive formulas

fn(U, v) = length of shortest path to v that

only goes via vertices in U,

for each vertex $v \in V \setminus U$.

$= \min_{u \in U} \{c(u, v) + d_u\}$

$U \leftarrow U \cup \operatorname{argmin}_{v \in V \setminus U} \{fn(U, v)\}$

B. Pseudocode

```
function Dijkstra(Graph, source):

    create vertex set Q

    for each vertex v in Graph:
        dist[v] ← INFINITY
        prev[v] ← UNDEFINED
        add v to Q
    dist[source] ← 0

    while Q is not empty:
        u ← vertex in Q with min dist[u]

        remove u from Q

        for each neighbor v of u still in Q:
            alt ← dist[u] + length(u, v)
            if alt < dist[v]:
                dist[v] ← alt
                prev[v] ← u

    return dist[], prev[]
```

Fig-1 : Pseudocode

3.2 FLOWCHART

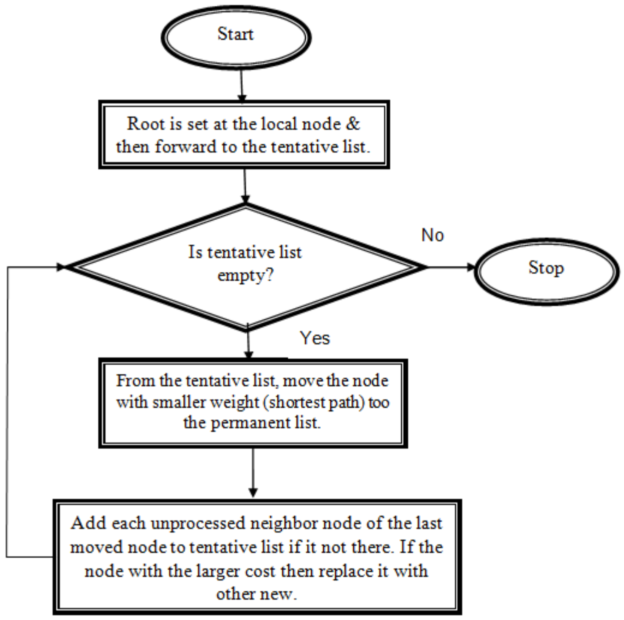

Fig-2: Flow Chart

4.Result

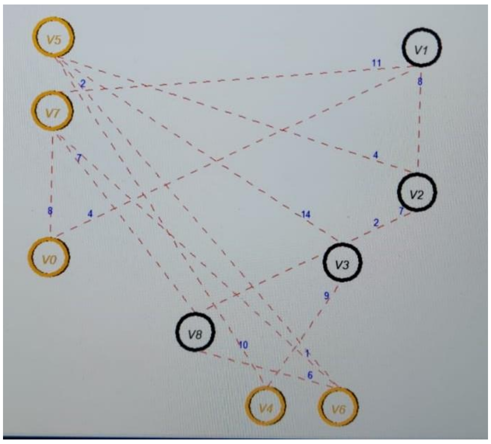

Fig -3:Visualization

Vertex	Distance from Source
0	0
1	4
2	12
3	19
4	21
5	11
6	9
7	8
8	14

Fig-4: Distance of Vertex

5. CONCLUSION

Among many, we've mentioned the Dijkstra algorithmic rule used for locating the shortest path, however, one among the obstacles whereas implementing the algorithmic rule on the net is to supply a full illustration of the graph to execute the algorithmic rule as a personal router contains a complete define for all the routers on the net. we've seen Graphs are accustomed to show connections between objects, entities or individuals, they need the most elements: Nodes and edges. Dijkstra's algorithm rule permits deciding the shortest path amid one designated node and every alternative node in a very graph.

6. FUTURE SCOPE

In this paper Dijkstra's algorithm rule is employed to seek out the shortest path in a graph. GPS is employed in Dijkstra's algorithm rule to urge this position of every node. Distance is additionally calculated from this position. an associate algorithmic rule is planned for this. however solely theoretical ideas are given, sensible implementation isn't given during this paper. So, in future you'll implement this algorithmic rule much. this idea of GPS is additionally used on others shortest path algorithms like A*, Warshals algorithmic rule & attendant Ford algorithmic rule etc.

7. REFERENCES

[1] https://www.researchgate.net/publication/273264449_Understanding_Dijkstra_Algorithm

[2] https://www.researchgate.net/publication/271617226_The_Improved_Dijkstra's_Shortest_Path_Algorithm_and_Its_Application

[3] http://www.ijstm.com/images/short_pdf/1430806082_453.pdf

[4] https://www.researchgate.net/publication/51988464_Dynamic_Scope-Based_Dijkstra's_Algorithm