International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Path Finder : An Artificial Intelligence Based Shortest Path

Anupam Singh, Vivek Lokchand Shahare, Nitin Arora, Ahatsham

Abstract*: *Artificial Intelligence driven assistant system is a supporting system that is used to find path within limited defined area as well as it can be used for the automation purpose. The path finder feature helps to find the path between two places within limited defined area. This approach is very useful for the people who are new to the organization, as they do not know the path inside the organization. There exist a lot of artificial intelligence assistants that help people to solve their problem. So, here is the implemented system that solves the path related problems within the organization and also helps to automate the system.

Keywords* : *Artificial Intelligence, Dijkstra's Algorithm, Shortest Path, Connected Graph.

I. INTRODUCTION

In a new organization, finding the shortest path for moving from one place to another place is a challenging problem. These places can be represented using nodes in a graph. Connected graph can be utilized to solve many practical applications like chip designing, reliability of network , transport planning and cluster analysis [1][2]. There are many applications of real world that are based on the concept of binary trees. A tree is said to be binary if each node has at most two children as left and right [3]. Some application like optimum path can be derived through a named algorithm called travelling salesman problem [4].

As the name of implemented system suggests, the vision was to develop a virtual assistant for people who are new to the organization to solve path based queries and provide path from the source to destination within campus. Virtual system may be considered as a system that is not physically avilable but can be used as a virtual assistant . The system uses Artificial Intelligence in software to guide people [5]. The implemented system can assist the user by solving user queries related to path within the campus. Artificial intelligence is an intelligence demonstrated by the machines in contrast to the intelligence displayed by the humans. Using this technology the machines behaves as humans. For example the query answered by the machine looks as if the answer is coming from humans. There are various applications of AI that is in health care, automotive, military, audit, art, advertising, government, video games, finance and economics. Similar is the behavior of this system as it provides the path to the user between two locations in such a way that a human is answering to the user query related to path.

To study an intelligent agents is a field of computer science. Any device that acts from its surroundings and perform actions accordingly can be considered as intelligent agents [6]. Learning and problem solving is the basic approach of artificial intelligence [7]. At times we saw that when parents and students come to the organization who are new to the organization face problems like they do not know the path within the campus so they need to stop someone and ask them to tell the path to a particular place within the campus so we got an idea to develop an assistant system which can help the people entering the organization by telling them the path within the campus.

This would be very helpful because they do not need to ask the path to anyone else. In other words, it can find the path in easy way as it is voice activated.

The path related information can be very helpful with respect to the new comers. Hence, there should be an assistant system that can guide them within the campus and provide them path related information.

II. RELATED WORK

Single source shortest path problem can be solved through Dijkstra's algorithm [8] or using Bellman Ford method. Dijkstra's algorithm can be used for non negative weights while Bellman Ford is used for positive as well as negative weights. Floyd Warshall algorithm is used to find all pair shortest path in a connected graph. The identified problem is to find out path in an organization through shortest path methods. We got an idea to make an application that can provide path to the people who do not know the campus. Secondly providing output in voice form as well as in display makes it quite interactive. In form of voice that system helps for some physically challenged people.

Manuscript published on November 30, 2019.
* Correspondence Author
Anupam Singh*, Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Email: anupam.singh@ddn.upes.ac.in
Vivek Lokchand Shahare, Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Email: vshahare@ddn.upes.ac.in
Nitin Arora, Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Email: narora@ddn.upes.ac.in
Ahatsham, Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Email:ahatsham@ddn.upes.ac.in

III. SYSTEM IMPLEMENTATION

It has been carried out in consideration that artificial intelligent system can be used with in the campus to find path between two ends. This is an assistant system that is capable of answering campus related queries of a new comer. The earlier existing solutions covers the following queries through multiple applications as Class time-table, Event timings, Examination time-table, Teacher assigned, Examination venue , Holidays, Lecture topics.

The system is designed and implemented on Python platform using Pycharm editor. For database connectivity it uses oracle and fetching the data related to path between source and destination. The user interface is developed using Tkinter which is a standard Python interface to GUI toolkit. The initial step is to take a voice input from the user which contains the source and destination for which the user wants to know the path. This voice input is processed to convert it into a text form so that we get the source and destination.

Dijkastra's algorithm [8] is used to find shortest path from defined source to all possible destinations in the graph. After fetching the shortest path, it is presented to the user in voice form as well as in written form in UI. Database will be maintained by inserting all information related with underlined organization. After updating all information related with organization, google speech API will be used to identify the shortest path on the basis of voice query. For any possible path from one place to another Dijkastra's algorithm is used. Graph can be plotted through multiple tools like MatLab. The system will be helpful for disabled people also. If a person is not capable to write query on screen then voice interactive system can be utilized.

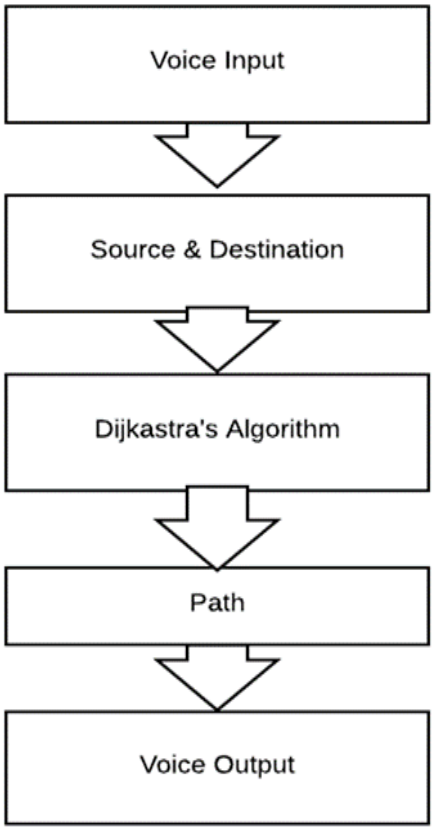

Fig. 1. Architecture of proposed system

For finding shortest path in a connected network there are two known algorithms Dijkstra algorithm and Bellman ford. Dijkastra works on positive weight values and gives the single source shortest path on the other hand Bellman ford can work upon positive as well as negative weights with consideration that there should not be any negative weight cycle[8]. A weight of one directed edge can be considered as the cost or distance between two nodes. In other words, moving from vertex a to b there is some weight assigned to that edge. Dijkstra's algorithm can be used to calculate and find the shortest path from source vertex to every other vertex within the same connected graph data structure, provided that the vertices are reachable from the source node.

Complexity of the algorithm depends upon number of vertex available in the graph. The implementation of Dijkastra is depend upon min priority queue and Fibonacci heap.

In Artificial intelligence, Dijkasra's procedure uses the concept of best first search [9].

The Bellman-Ford method is based on relaxation technique in which distance can be minimized by identifying the path in which total weight of the path is comparatively lower with second directed edge. This algorithm can identify negative weight cycle and if does not exist then shortest path will be evaluated.

Dijkastra's Algorithm

```
1.  Dijkstra(Graph, source):
2.  create vertex set Q
3.  for each vertex v in Graph:
4.  dist[v] ← ∞
5.  prev[v] ← UNDEFINED
6.  add v to Q
7.  dist[source] ← 0
8.  while Q is not empty
9.  u ← vertex in Q with min dist[u]
10. remove u from Q
11. for each neighbor v of u:
12. alt ← dist[u] + length(u, v)  if
        alt < dist[v]:
13. dist[v] ← alt
14. prev[v] ← u
15. return dist[], prev[]
16. exit
```

This algorithm is used to find shortest path from given source vertex. If distance is not known at beginning, keep it as infinity (not defined). For implementation, Queue data structure is used. Relax procedure is called to minimize the path distance from two identified nodes.

Retrieval Number: D7380118419/2019©BEIESP
DOI:10.35940/ijrte.D7380.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Bellman-Ford Algorithm

```
BELLMAN-FORD (G,w,s)
1   INITIALIZE (G,r)
2   for i ← 1 to |V|-1
3   do for each edge (a,b)∈E
4   do RELAX (a, b, c)
5   for each edge (a,b)∈E
6   do if d[b]>d[a]+w(a,b)
7   then return FALSE
8   return TRUE

Relax(a, b, c)
1 if d[b]> d[a]+w(a,b)
2 then d[b]:= d[a]+w(a,b)
3 exit
```

This algorithm may work upon negative weights also with no negative weight cycle[10]. A negative weight cycle can be explained by figure 2.

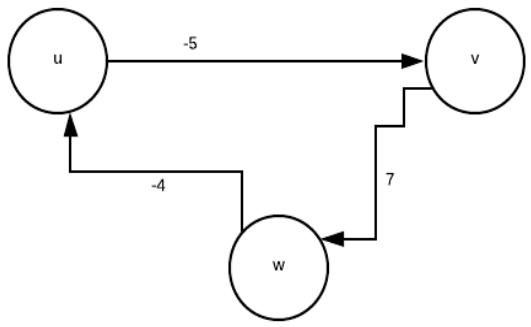

fig. 2. Negative weight cycle

In figure 2 negative weight cycle exist as total weight in a closed loop is negative. If graph contains such negative weight cycle then there is no possibility of shortest path from the source.

In such case, the Bellman–Ford algorithm can detect negative cycles and report their existence [11]. Negative edge weights may be available in different applications of graphs, hence the usefulness of this algorithm is quite high [12].

The time complexity of this algorithm is based on number of vertex as well as number of edges in the connected graph. On the basis of usage in graph applications, Bellman ford is comparatively better as it can work on negative weights and complexity for dense network should be depend upon vertex and edge both. If weights are non negative then Dijkastra's method is comparatively better.

Floyd Warshall algorithm is used to find all pair shortest path in a network. Here all pair means all permutations in the graph[13].

Execution of this algorithm will give shortest path from one vertex to all other vertex. Negative weights are allowed but there should not be any negative weight cycle.

This algorithm uses Fibonacci heap to store and process all nodes in the underlined graph. Path matrix and diagonal values play vital role in calculation of shortest path in a connected graph.

For a very large directed graph Dijkastra's approach is better as comparision to Floyd Warshall' method because of high constant factor in the running time of algorithm. Negative weight cycles can be identified through path matrix by inspecting the diagonal values and the presence of a negative number indicates that the graph contains at least one negative cycle [14].

Floyd Warshall's algorithm is applicable on directed graph that can be represented by adjacency matrix in terms of Boolean values. This method is useful when edges are undirected and weighted. Here two matrices are maintained one for distance minimization and other for calculation of predecessor node. In this application for finding shortest path, edges may directed means there is possibility of one path from a to b but not from b to a, hence this method should be avoided. Time required to execute this algorithm is depend upon distance calculation and finding parent node at same time.

The worst-case time complexity of Floyd Warshall's algorithm is cubic in nature. Therefore, this method is not meaningful for underlying problem.

The Floyd–Warshall Algorithm

```
1 let dist be a |V| × |V| array of minimum distances
initialized to ∞ (infinity)
2 for each edge (u,v)
3   dist[a][b] ← w(a,b)  // the weight of the
edge (a,b)
4 for each vertex b
5   it[v][v] ← 0
6     for k from 1 to |V|
7     for i from 1 to |V|
8     for j from 1 to |V|
9     if dist[i][j] > dist[i][k] + dist[k][j]
10    dist[i][j] ← dist[i][k] + dist[k][j]
11      end if
```

The reason behind choosing the Dijkstra algorithm over other algorithms is that Dijkstra's algorithm is asymptotically faster single-source shortest-path algorithm for any arbitrary directed graphs with nonnegative weights.

When the user comes in front of the system, the system says How can I help you and provides various options like say Path Finder to open path finder and other options to automate the system.

Then the user provides input by saying Path Finder. Then the system asks for the source where the user is present. So the user gives input by saying source place. Then system asks for the destination and the user provides the destination.

Now the system processes this information by applying the Dijkstra's algorithm in it and the graph is present within it. Now after fetching the shortest path it uses the database to get the individual path between the nodes. This information is provided to the user in form of voice and when the voice is over the output is also displayed on the user interface so that everything is clear to the user because at times it is possible that the user might not be able to understand everything.

The system will not only provide the single path but also gives the node to node path because the user is new to the place and the user will not be knowing how to go to the next place so to remove these type of ambiguities, node to node path is also provided in terms of voice form as well as displayed on the screen.

There are various possible paths from source to the destination but the system provides the shortest path between the two points using the Dijkstra's algorithm so that the user needs to move less and saves the time.

Besides this there are other functionalities too which are used to automate the system like play song, in this the user needs to input the song name in voice form then the assistant will search the song within the system and if the song is present then the assistant will play it for the user. So the process of searching the song then playing the song is automated and will be done by the assistant, the user only needs to input the song name and his work is done.

Open application, in this the user needs to input the application name in voice form then the assistant will open the application for the user.

Search file, in this the user needs to input the file name in voice form then the assistant will search the file within the system and if the file is found then the assistant asks if the user wants to open the file or not. If the user says yes then file is opened and if the file is not found then an appropriate message is said file not found by the system.

Web crawler, if the user wants to search any information on web browser then he needs to input the query in voice form and the assistant will open the web browser along with the searched information.

So these were the functionalities which are automated else the user needs to do them on his own. For example to play a song normally the user first need to search in all directories then click on the song to play but here all these things will be done by the assistant itself. The user only needs to input the song name and wait for the output.

Shortest Path Identifier Algorithm

- The user enters the source and destination in voice form for this purpose the Google speech recognition API is used.
- After this voice is converted into text form and the source and destination is stored in the variables as they need to be passed in the function.
- These source and destination are passed to the function where Dijkstra's algorithm is implemented.
- With the help of the nodes and the distances between them it forms the graph and in the parameter we have the source and destination so with the help of it, the algorithm returns the shortest path between the nodes.
- This returned information is processed so that it can be properly presented on the screen.

The screen in the fig. 3 depicts the first screen which is displayed in front of the user when the system starts. Here the system says How can I help you and provides various options for the voice input which can be for the path finder or for automating the functionalities of the system.The system uses google API to read voice input.

Artificial Intelligence solves the problem by maintain the database from where result will be processed. If source is not available as per query then system will return a message that requested information can not be processed.

The system returns all possible cross points of the path (intermediate node) that will automate the system.

Fig.3. Implemented system Input Screen

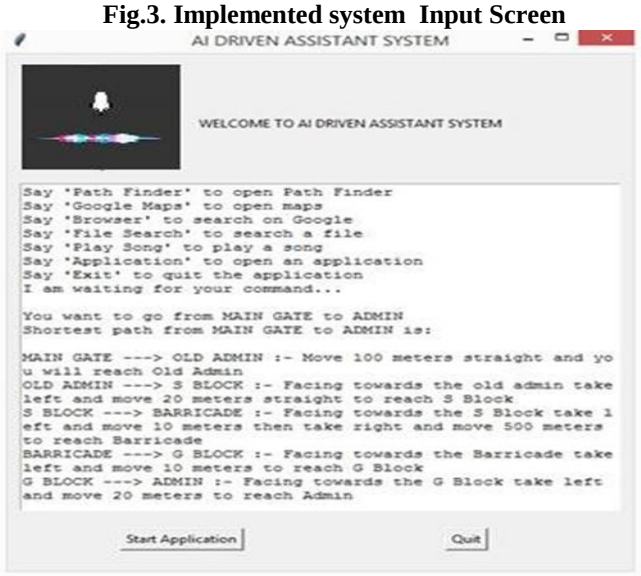

Fig. 4. Implemented system Output Screen

The screen in the fig. 4 is displayed along with the output of shortest path in voice form as well as in graphical form.

IV. CONCLUSION

In this paper we have presented the working of assistant system using artificial intelligence which is very useful to the people who are new to the organization and do not know much about the locations within the organization. In this application, we have presented the two main functionalities that are finding path within the organization using Dijkstra's algorithm and others are used to automate the system. So this system provides the shortest path within two locations in the campus in voice form and then it is displayed on the screen too. In future this system can be extended for other requirements and can be utilized as an assistant at home for day to day activities. .

REFERENCES

1. Kumar A., Singh SP and Arora N. "A New Technique forFinding Min-cutTree". International J. of Comp. Applications(IJCA) 69(20):1-7, May 2013.
2. Arora N., Kaushik P. K. and Singh S. P., "A Survey onMethods forfinding Min-Cut Tree". International J. of Comp. Applications(IJCA), vol. 66, 23, Jan. 2013.
3. Arora, N., Tamta, V. K., and Kumar, S. 2012. Modified Non-RecursiveAlgorithm for Reconstructing a BinaryTree. International J. of Computer Applications(IJCA). vol 43. No 10. 25-28
4. Arora N., Garg S., Sant V. and Goyal R., "Automated OptimumRoute Generator and DataAnalyzer", International J. of Comp. Applications(IJCA), vol. 181, 48, pp. 17-21, April 2019.
5. Shetty V.Y.,Polekar N., Das S., Pansambal Artificially Intelligent College oriented virtual assistant ISSN: 2320-8163, Volume 4, Issue 2 (March-April, 2016).
6. Definitionof AI as the studyof intelligent agents:Poole, Mackworth & Goebel 1998, p. 1, Russell et al. (2003)
7. Russell, Stuart J. et al., Artificial Intelligence: A Modern Approach (2009) (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 978-0-13-604259-4.
8. Szcześniak et al. "Generic Dijkstra for Optical Networks". October 2018
9. Position Paper: Dijkstra's Algorithm versus Uniform Cost Search or a Case Against Dijkstra's Algorithm
10. Bang-Jensen, Jørgen; Gutin, Gregory (2000). "Section 2.3.4: The Bellman-Ford-Moore algorithm". Digraphs: Theory, Algorithms and Applications (First ed.). ISBN 9781-84800-997-4
11. Schrijver, Alexander (2005). "On the history of combinatorial optimization (till 1960)"
12. Sedgewick, Robert (2002). "Section 21.7: Negative Edge Weights". Algorithms in Java (3rd ed.). ISBN 0-20136121-3.
13. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990). Introduction to Algorithms (1st ed.). MIT Press and McGraw-Hill. ISBN 0-262-03141-8.
14. Hochbaum, Dorit (2014). "Section 8.9: Floyd-Warshall algorithm for all pairs shortest paths"

AUTHORS PROFILE

Anupam Singh is Assistant Professor–Senior Scale in Department of Informatics, School of Computer Science, University of Petroleum & Energy Studies, Dehradun. He is pursuing Ph.D. from Dr. A P J Abdul Kalam Technical University Lucknow. Formerly UPTU Lucknow). He has done B. Tech. in 2004, M. Tech. in 2011 from Uttar Pradesh Technical University Lucknow. His area of interest are Formal Methods, Distributed System and Database System. He is Reviewer of various referred journals. He has evaluated many presentations in International Conference as Session Chair.

Vivek Lokchand Shahare is Assistant Professor in School of Computer Science and Engineering, UPES, Dehradun. He has completed his M.Tech from Visvesvaraya National Institute of Technology (VNIT), Nagpur in 2016 and B.Tech from Government College of Engineering, Amravati in 2012. He has 5 years of experience in Teaching, Research and Industry. His areas of interest are Bioinformatics and Theoretical Computer Science.

Nitin Arora received B. Tech. (Comp. Sci. Engg.) from NIT ALLAHABAD in 2008 and M. Tech. (Comp. Sci. & Engg.) in 2012 with GOLD MEDAL from Uttarakhand Technical University, Dehradun, the member of IANEG (USA), ISOC (USA) has published over Fifteen plus research papers in National and International Journals/Conferences and IEEE proceeding publication in field of Data Structures and Algorithms, Mobile Ad-hoc networks & Digital Image Processing. He is working as an Assistant Professor (Senior Scale) at University of Petroleum and Energy Studies (UPES), Dehradun.

Ahatsham was born in Roorkee, India in 1993. He received his B.tech degree from Uttrakhand Technical University in 2014 and M.tech degree from NIT Nagpur in 2016.He is currently working as Assistant Professor in University of Petroleum and Energy Studies, Dehradun, India. His research interests include data mining, machine learning and data streaming.