Finding Shortest Path Between Source and Destination by Extended Dijkstra Algorithm

*

Soumyadip Guha
Computer Science and Engineering
Birla Institute of Technology and Science
Hyderabad, India
H20221030092H

Sadanala Pavan Sai Krishna
Computer Science and Engineering
Birla Institute of Technology and Science
Hyderabad, India
H20221030061H

Dantu Havishteja
Computer Science and Engineering
Birla Institute of Technology and Science
Hyderabad, India
H20221030083H

***Abstract*—The Dijkstra algorithm is a well-known method for finding the optimum result in shortest path search problems. However with that method, the time required to find the optimum path becomes really high when the search are is broad, so the Dijkstra algorithm is not suitable for real-time problems. In this paper, we propose a new method for obtaining, in a shorter time, a path that is near optimal and as close as possible to the path obtained by the Dijkstra method. The new method extends the conventional Dijkstra algorithm to obtain a solution to a problem given within short period of time, for example, path search in a car navigation system. The effectiveness of that extended method is described through use of experiments.**

***Index Terms*—Dijkstra, extended Dijkstra, shortest path**

I. Introduction

The finding shortest path problem involves finding optimum path between the present 1ocation and the destination under given conditions. Currently, these problems arise in different fields such as the highway system, railroads, and communication networks, and it covers a wide range of applications. In particular, currently car navigation systems have exhibited huge growth in popularity due to recent advances in science and technology. But the actual route found in car navigation is not really the optimum solution because of hardware restrictions and the calculation time required for the path search. A number of path search algorithms have been proposed for use in car navigation systems, but no exceptionally good method has yet been established. In this paper, we propose an algorithm of path search that is based on the conventional Dijkstra method, which can find the optimum solution, but extending that method we can obtain a path that is as close as possible to the optimum solution in a shorter time. This paper has the following sections. In section II, we explain the conventional algorithms that are employed for the shortest path search problems. In section III, we propose our new algorithm obtained by extending the Dijkstra method, that is capable of finding the near optimal shortest path, and show the results of comparative experiments to show the effectiveness of our proposed method. In section IV, we identify the issues to be addressed in future work to improve further our extended dijkstra algorithm.

Identify applicable funding agency here. If none, delete this.

II. Shortest Path Search

A. Features of Various Path Search Methods

Here, we describe respective features of mostly used algorithms for searching path like, Dijkstra algorithm, A* algorithm, and genetic algorithms.

1) The Dijkstra method: Dijkstra algorithm is used for finding the optimum path. As this algorithm searches for the minimum-cost path among all the possible paths in order, beginning from the current location, the search region expands concentrically. Thus this method has the disadvantages of poor search efficiency and long search time when the distance to the destination is long.

2) The A* *algorithm:* A* algorithm is also based on the method of the Dijkstra algorithm, but it eliminates useless searches by considering the distance to the destination.

3) Genetic algorithms: Genetics algorithms imitate number of processes those are seen in natural evolution. In case of path search, a path is selected and regarded as a gene, and solution path is obtained by performing calculations those emulate genetic crossing, spontaneous mutations, and so on. Search time of these algorithms is determined by the number of genes those are produced, so these methods have advantage of not being affected by network size. A problem with this approach is that the number of generations necessarily becomes large as the optimum solution is approached.

B. Dijkstra Algorithm

Here we discuss the Dijkstra algorithm. Here, we consider roads intersections as nodes and the roads those connect the nodes as routes and the length and ease of transportation of a path as cost, as shown in Fig. 1.

Algorithm is shown below:

- step1: Mark the starting point.
- step2: Calculate movement cost for movement from stating point to each node that are connected to the starting

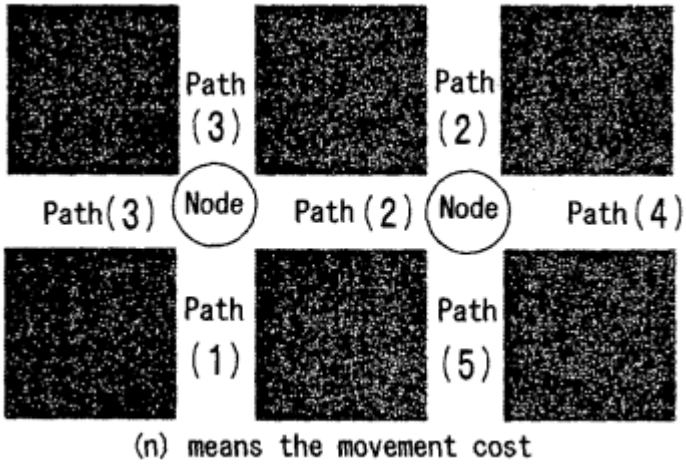

Fig. 1. Route diagram

node and mark the node which has smallest movement cost.

- step3: Calculate movement cost for movement between starting node and each node connected to the marked node and marked the node which has smallest cost.
- step4: Repeat the step3 until we mark the destination.

The value obtained by the above method is the minimum cost of movement or the shortest path to the destination. And also by storing the previous node in memory when marking a node we can obtain the shortest path to destination.

III. Extended Dijkstra Algorithm

A. Basic Algorithm

In normal Dijkstra method, search region generally expands concentrically, as a result many nodes, which are not useful, are checked for each search, so the search time is long. Because of that this method cannot. obtain solution in real time if the distance from the starting point to the destination is long. Therefore, we propose a new algorithm in which Dijkstra method is applied from both directions source and destination, and this is how conventional Dijkstra algorithm is extended t0 make it possible to obtain solution to the given problem within specified time. In this way, the concentric expansion of search region is restricted and the number of nodes to be searched can be reduced. (See Fig. 2.)

Algorithm is shown below:

- step1: Calculate movement cost for movement from starting point to each node connected to starting point and for movement from destination point to each node connected to destination point, and mark the node which has the smallest movement cost.
- step2: Mark the node that has the smallest movement cost from the unmarked nodes those are connected to the marked nodes to either starting point or destination point.
- step3: Repeat Step 2 until the search from starting point and the search from destination point overlap and then end.

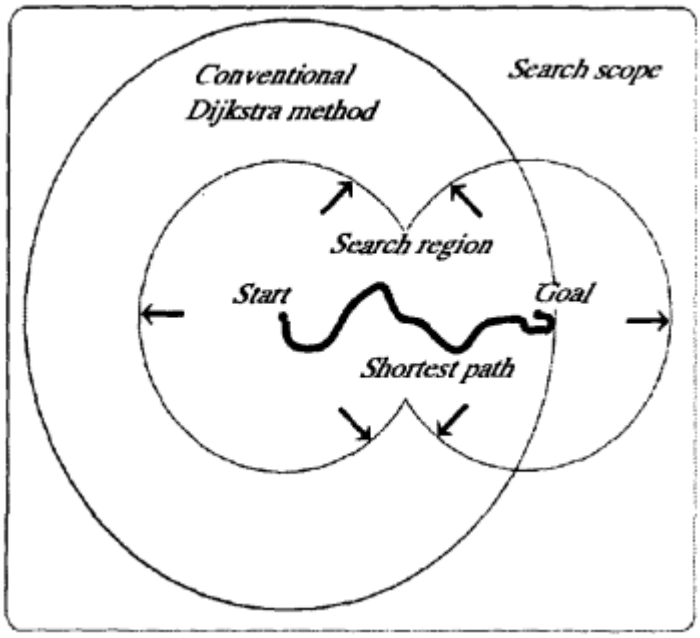

Fig. 2. Comparison between conventional Dijkstra method and the extended Dijkstra method

I the search scope is too large the above mentioned method will take long time as this method based on Dijkstra method. So, if the search time or search scope exceeds a certain value, the search is ended at that point and a genetic algorithm will be employed at that point to approximate the remaining search region.

B. Experiments

We performed our experiments on the Intel i5 7th Gen processor and 8GB RAM computer, compiled with gcc. We have created 100X100 node grid pattern of paths to serve as search region. Path movement costs are assigned randomly in the range of 0 to 9, with the lower movement cost represent easier movement. We created a smaller square and put starting point and destination point diametrically opposite corner of the square. The distance between source and destination can be change by changing the size of the square. Under this conditions we ran many searches and among those some experiments result

From the above table we can see that for all the distance number of nodes counted by the extended Dijkstra algorithm is almost half of that by the conventional Dijkstra algorithm. And, time taken by the extended Dijkstra algorithm is almost 25% of the time taken by conventional Dijkstra algorithm.

IV. Conclusion

Extended Dijkstra method give near optimal result as close as possible to the solution given by conventional Dijkstra method. And, moreover time taken by Extended Dijkstra method is very less compared to the conventional Dijkstra

TABLE I
SIMULATION RESULT

Distance	Conventional Dijkstra Method			Extended Dijkstra Method		
	Search Time(s)	*Nodes Counted*	*Mean cost*	*Search Time(s)*	*Nodes Counted*	*Mean Cost*
20	0.03	64	38.1	0.01	30	38.1
40	0.20	167	74.5	0.05	84	74.8
60	0.57	313	110.0	0.18	181	110.3
80	2.34	570	147.0	0.44	257	147.3

method. So, as a result we can use this methid for finding path in real-time system like car navigation system. But, here to mention that even the above mentioned method giving solution faster, its based on the Dijkstra method, so if the distance between source and destination is really long, then this method also take make more compared to expected time to give the shortest path. Genetics algorithm is one of the popular method for giving optimal solution faster. So, in future extending the proposed method further with the genetic algorithm we may get better method for finding shortest path in real time.

REFERENCES

[1] M. Noto and H. Sato, "A method for the shortest path search by extended Dijkstra algorithm," Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0, 2000, pp. 2316-2320 vol.3, doi: 10.1109/ICSMC.2000.886462.