PAPER • OPEN ACCESS

The Comparison of Three Algorithms in Shortest Path Issue

View the article online for updates and enhancements.

The Comparison of Three Algorithms in Shortest Path Issue

Xiao Zhu WANG

Northeastern University at Qinhuangdao, Computer science and technology, Hebei 066004, China

2656398251@qq.com

Abstract. The shortest path problem is one of the most classical algorithm issues in graph theory, aiming to find the shortest path between the two nodes in a network. It is widely used to solving the path search problems in the relevant application, including network routing algorithm, artificial intelligence, game design, and so on. This paper introduces three basic algorithms and compares them theoretically by analyzing the time and space complexity. Then, the paper discusses their performances and summarizes the best application range. At last it introduces applications of shortest path algorithm.

1. Introduction

Cross-section The original of the shortest path algorithm is to solve the problem of finding a path between two nodes in a graph such that the sum of the weights of its constituent edges is minimized. Actually, it is a kind of searing algorithm of calculator graphics, which means searching the lowest cost path between the starting point and object point [1]. It is not only used in finding the shortest distance in the general geographical problems, but also used in others fields, such as finding the most reliable path in communication networks and Critical path problem [2]. According to the final node and the characteristics of the path, the shortest path problem can be divided into five types: The shortest path between two nodes, the shortest path among all nodes, the K shortest path, the real-time shortest path and the shortest path of the specified path. This paper mainly introduces the second shortest path problem.

The shortest path is an important research topic in graph theory. Massive domestic and foreign scholars discussed it deeply and proposed diverse algorithms to solve the shortest path question. The shortest path algorithm can be divided into single source shortest path and multi-source shortest path. Single source shortest path is getting the shortest path from a given vertex to any other vertex. The classical single source shortest path algorithm Dijkstra's algorithm, which was conceived by computer scientist Edsger W. Dijkstra in 1956, is in the widespread used in road networks. Later, Richard Bellman and Lester Ford published the Bellman-Ford algorithm in 1958 and 1956, which has a significant difference with Dijkstra's algorithm, it can be used on graphs with negative edge weights, as long as the graph contains no negative cycle reachable from the source vertex [3]. In 1962, the Floyd-Warshall algorithm could find shortest paths in a weighted graph with positive or negative edge weight, which is a typical multi-source shortest path. The rest of this paper is organized as following: The first portion will introduce three algorithms which are mentioned above. Then the second section will compare the performances and the present the most suitable field of three algorithms, like in the time and space complexity. Next, there is an application to represents the differences of three algorithms. The last part of the paper is a summary.

2. The introduction of three algorithms

2.1. The Dijkstra algorithm

Finite element models of hot-rolled RHS/SHS stub columns were developed by using the non-linear finite element program ABAQUS [5], in which two steps including linear perturbation and non-linear analyses were performed in order to obtain the ultimate carrying capacity and failure modes of RHS/SHS stub columns. Material properties and cross-section dimensions measured from Joanna's test [6] were included in the finite element model. The typical nomenclature is defined in Figure 1.

The Dijkstra algorithm is used to find a single node as the source node and calculate the shortest paths from the source to all other nodes in the graph, increasing node by node to get a shortest path tree [4]. Here is the algorithm:

Step1: Initializing $D[i]$. $D[i]$ represents the distance from the starting point V to the point Vi. If there is an arc between the two vertices, then $D[i]$ is the weight of the arc; Otherwise, $D[i]=\infty$.

Step2: Finding a node Vj, which is abutting to node V and has the shortest distance from V.

Step3: Then finding a node Vk, which is abutting to node Vj and has the shortest distance from Vj. Making $D[j]=Min\{D[j], D[i] +$ the weight from Vj to Vk$\}$.

Step4: Repeating the step3 until reaching the destination node.

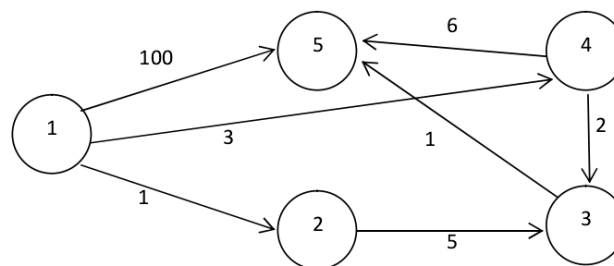Here is a simple example to show detailed steps of   the algorithm.[9]

Figure 1. A example of the Dijkstra algorithm

Table 1. A example of the Dijkstra algorithm

Circulation	V	Node2	Node3	Node4	Node5
Initial node	1	1	Max	3	10
1	1,2	1	6	3	10
2	1,2,4	1	5	3	9
3	1,2,4,3	1	5	3	6
4	1,2,4,3,5	1	5	3	6

2.2. The Bellman‐Ford algorithm

The Bellman‐Ford algorithm is an algorithm that computes the shortest path from a single source vertex to all of the other vertices. It is capable of solving graphs in which some of the edge weights are negative numbers. Here is the algorithm:

Step1: Initializing $D[i]$. $D[i]$ represents the distance from the starting point V to the object point Vi .

Step2: w (m,n) is the weight of the edge e (m,n), and e is the shortest path between (m,n). For each edge e (m,n). If $D[m] + w(m, n) < D[n]$,

$D[n] = D[m] + w(u, v)$.

Step3: The loop performs up to i - 1 times, and i is the number of the vertices. If the operation above does not update the D[i], the shortest path has been searched, or some of the points cannot be reached. Otherwise, executing to the next cycle.

Step4: Testing the diagram to find whether it has a negative loop (the sum of weight is less than 0). If D [u] + w (u, v) < Distant [v], there is a negative loop, which means the shortest path cannot be founded in the graph. Otherwise D[i] records the shortest path. For example, if there is a negative loop, the values of each point will decrease, after one traversal. It is shown in Figure 2.
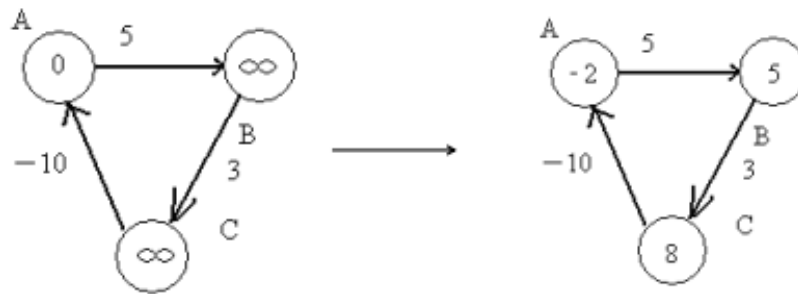

Figure 2. A negative loop

2.3. The Floyd－Warshall algorithm

The Floyd－Warshall algorithm compares all possible paths through the graph between each pair of vertices, namely, it calculates the shortest   path between all nodes.

The basic idea of Floyd algorithm has four steps [5]:

Step1: Find two vertices from the network, put each vertex in the network into these two points as an intermediate.

Step2: Compare the original distance with the new distance between these two points, treat the smaller distance as the new shortest distance.

Step3: Sequentially construct n matrix S(1), S(2), ⋯, S(n) by looping iteration, the each element in the last matrix S(n) represents the shortest distance between the two points. Figure 3 and Table 2 show the detailed process.

Step4: Get the minimum distance of one point to the other points by summing the elements in each lines of S(n), one or more best locations can be found by comparing these summations.
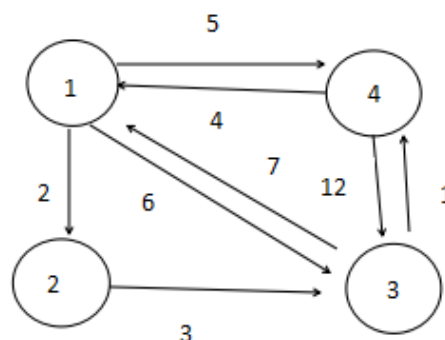

Figure 3. An example of The Floyd–Warshall algorithm

Table 2. An example of The Floyd–Warshall algorithm

0	2	6	4
∞	0	3	∞
7	∞	0	1
5	∞	12	0

3. Discussion

The current study focused on two areas: single-source shortest path and the shortest path among all vertices [6]. The advantages and disadvantages of an algorithm are mainly measured from two aspects: the execution time of the algorithm and the storage space. By summarizing the papers from others [2] [3] [7], obtaining Table 1.

Table 3   Comparison Between Numerical and Experimental Results

	The Dijkstra algorithm	The Bellman-Ford algorithm	The Floyd-Warshall algorithm
space complexity	$O(M)$	$O(M)$	$O(N^2)$
time complexity	$O(N^2)$	$O(MN)$	$O(N^3)$
The edge weights are negative	×	√	√
M is the number of edges N is the number of notes			

The Dijkstra algorithm and the Bellman-Ford algorithm are similar in some place. Both of them use the method of relaxation calculation, which is to find the shortest path by modifying the values of D[i] during the process of traversing the vertices and edges of the graph. The Dijkstra algorithm is mainly aimed at the graph with non-negative weight nodes, while the bellman-ford algorithm can deal with the shortest path problem with negative weights. They are used to draw the optimal solution of the shortest path, but the Bellman-Ford algorithm has huge redundancy and lower efficiency, which is shown in Table 1.

The Dijkstra algorithm can only be used in single-source shortest path problem. But the Floyd-Warshall algorithm is available to find a shortest path between any two points [7]. It is suitable for finding the shortest path among all vertices or in a small data scope.

In conclusion, the Dijkstra algorithm applies spares graph. In actual application, the algorithm is always optimized, like heap optimization. The time complexity cuts to n*logn [8]. The Bellman-Ford algorithm is inefficient but it is easy to implement. When dealing with the problem with a large number of points and edges, the Floyd-Warshall algorithm is the slowest and wastes redundancy space.

4. Applications

Routing algorithm is a part of network layer software, which determines the outgoing route of the received packet. When studying routing algorithms, there is a widely used the shortest path algorithm.

The main idea is to create a subnet figure. Each node in the diagram represents a router and each arc is a communication line. In order to select the routing among a pair of routers, the algorithm should find the shortest path in the diagram. Dijkstra routing algorithm is suitable for calculating the shortest path of a router to other routers. But in a computer network, if there are n routers, the Dijkstra routing algorithm will be reused N times. Thus, the Floyd-Warshall routing algorithm is used more in practice, because it can calculate the shortest distance between any two routers.

Modern computer networks usually use dynamic routing algorithms, namely link state routing algorithm and distance vector routing algorithm.

Link state routing protocol collects all kinds of information of the whole network, which constitute a topological database of routers. Open Shortest Path First (OSPF) is a typical protocol which is an internal gateway protocol which used to make routing decision within a single autonomous system. Besides, it is a specific implementation of Dijkstra algorithm. It mainly uses the algorithm to generates a tree without loops. Then starting from a router and passing the information to all the routers in the tree. Thus, all the routers is shared by all the state of the link. Each router is calculated in the local routing and avoids updating the routing table blindly.

The distance vector routing algorithm is that each router maintains a table. The table has the best path and route for each destination through exchanging the information with neighboring routers to update the table information. RIP protocol is a dynamic routing protocol and uses the Bellman-Ford algorithm. The process of routing announcement is the process of the Bellman-Ford algorithm's implementation. The routers collect all different paths to the destination and save the number of sites about information of each destination path. Any other information will be discarded, except the best route to the destination. The algorithm is distributed execution. All the routers are in the execution of the algorithm and the results are calculated together by all machines. In OSPF, the algorithm only executes on one machine which is not distributed.

5. Conclusion

The shortest path issue is still one of the hottest topics in the research field. This paper introduces the basic principle of three shortest path algorithms and compares them by analyzing the time and space complexity. Besides, it summaries the application range of different algorithms. The Dijkstra algorithm is the classical single-source algorithm. And the Bellman-Ford algorithm can be a supplementary when there is a negative loop. And the Floyd-Warshall algorithm is a dynamic programming algorithm. It can solve the shortest path problem between any two vertices. However, in practical applications, these three algorithms are not applied directly. They are always optimized to increase the efficiency, like the heap optimization and SPFA. As the increase of the number of data that computer processing, the load of traditional serial computers is also gradually increasing. The shortest path algorithm runs on the server side to develop the parallel algorithm based on graph decomposition to meet the demand of a large number of real-time shortest path queries.There are two types of shortest path parallel algorithms, one is the parallel computer model that does not limit the number of processors. Another is to study the performance analysis of the actual parallel computing system and the actual system. Parallel computing system is more valuable in practice and is the trend of shortest path problem algorithm.[8][9]

References

[1] Kong, D., Liang, Y., Ma, X., & Zhang, L. (2011, July). Improvement and Realization of Dijkstra Algorithm in GIS of Depot. In Control, Automation and Systems Engineering (CASE), 2011 International Conference on (pp. 1-4). IEEE. (APA)

[2] Hougardy, S. (2010). The floyd-warshall algorithm on graphs with negative cycles. Information Processing Letters, 110(8), 279-281.

[3] Li, T., Qi, L., & Ruan, D. (2008, November). An efficient algorithm for the single-source shortest path problem in graph theory. In Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on (Vol. 1, pp. 152-157). IEEE.

[4] Yue, Y. (1999). An efficient implementation of shortest path algorithm based on dijkstra algorithm. Journal of Wuhan Technical University of Surveying & Mapping.

[5] Tao, L., Tong, X., He, X., Xu, H., & Tan, Z. F. (2013). Location analysis of electric vehicle charging station based on the Floyd shortest path algorithm. In Applied Mechanics and Materials (Vol. 389, pp. 1014-1018). Trans Tech Publications.

[6] Sun, H. (2013). Path-Based Floyd Algorithm in Automatic Search System. Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012. Springer London.

[7] Wang, Z. H., Shi, S. S., Yu, L. C., & Chen, W. Z. (2012). An efficient constrained shortest path algorithm for traffic navigation. In Advanced Materials Research (Vol. 356, pp. 2880-2885). Trans Tech Publications.

[8] Klein, P., Rao, S., Rauch, M., & Subramanian, S. (1994). Faster shortest-path algorithms for planar graphs. Twenty-Sixth ACM Symposium on Theory of Computing (Vol.55, pp.27-37). ACM.

[9] Jian, L. I. (2009). Optimization studies based on shortest path algorithm of dijkstra. Journal of Weinan Teachers University.