International Journal of Engineering and Management Research e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-3 (June 2023)
https://ijemr.vandanapublications.com https://doi.org/10.31033/ijemr.13.3.34

Upgrading Shortest Path Problems

Shra Fatima[1] and Kamlesh Chandra Maurya[2]
[1]Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, INDIA
[2]Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, INDIA

[1]Corresponding Author: shraintegral251@gmail.com

Received: 01-06-2023 Revised: 16-06-2023 Accepted: 30-06-2023

ABSTRACT

There are several areas where the shortest path problem exists. One well-known algorithm is the Dijkstra Algorithm. The "Dijkstra Algorithm" has one flaw, according to the results of the experiments, namely that it doesn't deal with the issue of adjacent vertices in the shortest path. The algorithm has seen substantial improvement. The findings of our trial show that the problem has been successfully fixed.

Keywords-- Dijkstra Algorithm, Dijkstra's Label Algorithm, P-Lables, Shortest Path

I. INTRODUCTION

We must find solutions to several shortest path difficulties in the course of manufacturing, organisation, and management. For instance, in the production process, we should determine the shortest path to take in order to complete each production task quickly and effectively; in the management process, we should create rational plans in order to achieve significant gains at the lowest possible cost; and in the existing transportation network, we should make arrangements for a reasonable transport path in order to transport significant quantities of goods at the lowest possible cost. The "shortest path problem" can be used to describe all of these issues.

In many different domains, including computer network routing algorithms, the Pathfinder robot, route navigation, game design, and others, the challenge of determining the shortest path between starting vertex and terminal vertex (which exist in a given network architecture) is frequently encountered. The static path algorithm and the dynamic path algorithm are two different types of methods for finding the shortest path. In order to determine the shortest path, the static path algorithm assumes that the external environment is unchanging. The Dijkstra algorithm, the A* (A Star) algorithm, and others are just a few examples of the numerous static route algorithms.

The dynamic path method seeks to determine the shortest path in the event of unpredictability, indicating that the external environment is changing. The opponents and obstacles, for instance, are always moving in video games. Like the D* (D Star) algorithm, there are other common dynamic path algorithms. The D* (D Star) algorithm is reportedly the main algorithm employed by the American Mars Pathfinder spacecraft. The static path algorithm is the only one we cover in this article. The "Dijkstra's algorithm," which was first put forth by Dijkstra in 1959, is one of the best shortest path algorithms.

The study is organised as follows: first, we explain Dijkstra's Algorithm; second, we highlight the necessity to enhance the algorithm by experiments; third, we suggest an improved approach and validate the algorithm through tests. The enhanced method is more efficient than the "Dijkstra algorithm," according to experimental data, and it is capable of effectively addressing the shortcomings of the "Dijkstra algorithm."

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.34

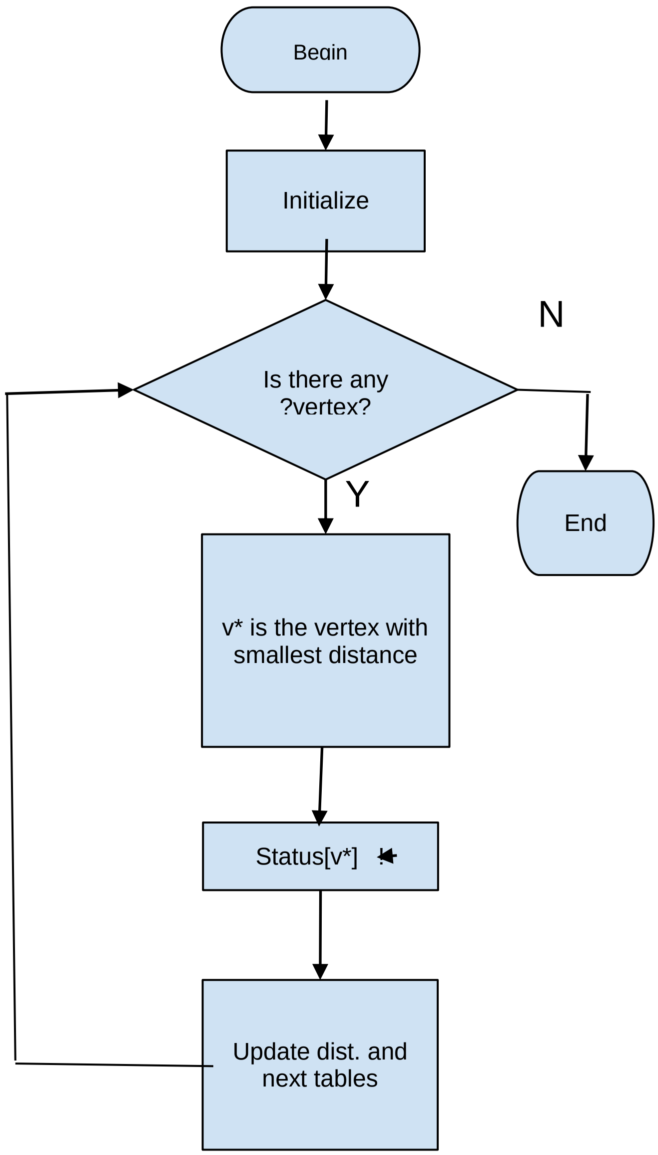


Figure 1: Flowchart of Dijkstra Algorithm

II. DIJKSTRA ALGORITHM

One of the best shortest path algorithms is Dijkstra's algorithm, which was put forth in 1959.

The "Dijkstra algorithm" has a very broad variety of applications, including multi-point routing, surveying and mapping science, the shortest path for logistics and transportation, the intelligent transportation system, the motorway network toll collecting, and more. The shortest path algorithm and the "Dijkstra algorithm" have been the subject of extensive investigation.

A. The Content of Dijkstra Algorithm

SupposeG=<V,E,W>is a n-order simple weighted graph ($w_{ij} \geq 0$).
If vertex v_i is not adjacent to vertex v_j
then set $w_{ij}=\infty$.
To calculate the shortest path between vertex v_1
and other vertices in graph G.
Following are related definitions:

1. Suppose $l_i^{(r)*}$ is the weight of the shortest path from v_1 to v_i . If v_i obtain the label $l_i^{(r)*}$ then v_1 obtain the "p label" $l_i^{(r)*}$ (permanent label) in step r(r≥0).

2. Suppose $l_i^{(r)}$ is the upper of the shortest path from v_1 to v_j .If v_j obtain $l_j^{(r)}$ then v_j obtain "t-label"$l_j^{(r)}$ (temporary label) in step r.
3. Set P_r= { v | v has obtained "p-label"} to be "pass vertex set" in step r.
4. Set T_r =V- P_r to be "not pass vertex set" in step r. Dijkstra algorithm is the following:

Initial: r←0, $l_1^{(0)*}$ =0, P_0 ={v_i}, T_0=V-{ v_1}, ($l_j^{(0)}$ = w_{ij}(j≠1).v_1(obtain "p-label",v_j obtain "t-label")
1. Find the next "p-label" vertex.
Set $l_i^{(r)*}$ = 1 $\min_{j\varepsilon Tr-1}${ $l_j^{(r-1)}$ }(r≥1). v_i obtain "p-label"$l_i^{(r)*}$.
Update the "pass vertex set" and the "not pass vertex set":
P_r =
P_{r-1} ∪{ v_i}, T_r =T_{r-1}-{ v_i}. check T_r :
If Tr =∅ then the algorithm end else jump2..
2.Update each vertex's "t-label" in Tr : $l_j^{(r)}$=min{ $l_j^{(r-1)}$, $l_i^{(r)*}$+ w_{ij}}. $l_i^{(r)*}$ is the latest "p-label".set r←r+1 and jump 1.

B. The Algorithm Experiment

The simple weighted undigraph shortest path problem can be successfully solved using the Dijkstra method. The shortest path between v_0 and v_5 can be determined using the Dijkstra algorithm, which is illustrated in Figure 2.

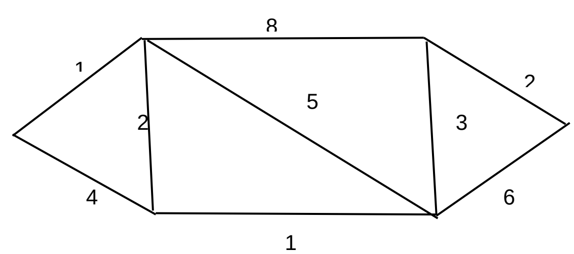


Figure 2: Simple weighted Undiagraph

The process of calculating is the following

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.34

Table I

V_1 \ R	V_0	V_1	V_2	V_3	V_4	V_5
0	0	1	4	∞	∞	∞
1		$1/v_0$	3	8	6	∞
2			$3/v_1$	8	4	∞
3				7	$4/v_2$	10
4				$7/v_4$		9
5						$9/v_3$
W	0	1	3	7	4	9

Table 1: Calculating Process

All of the vertices have their "p-labels" after the calculation is complete. The shortest routes connecting vertex v_0 to other vertices (v_1, v_2, v_3, v_4, v_5) are:
$v_0 \to v_1$, $v_0 \to v_1 \to v_2$, $v_0 \to v_1$
$\to v_2 \to v_4 \to v_3$, $v_0 \to v_1 \to v_2 \to v_4$.
The length of each shortest path is: 1, 3, 7, 4

III. THE INADEQUATE OF DIJKSTRA ALGORITHM

Dijkstra's label algorithm has to be improved because it is insufficient. The analysis and improvement suggestion are as follows:

The Adjacent Vertices in the Shortest Path

The shortest path did not include instructions on how to obtain nearby vertices (namely those to the preceding vertices) in Dijkstra's label algorithm. While in actuality, locating the nearby vertices along the shortest path is frequently required. Dijkstra's label algorithm must therefore be enhanced.
This paper proposes the following improvement:
while updating the "t-label" of each vertex in
T_r (v_j).
According to v_i, if v_j's "t-label" is updated then v_i is the adjacent vertex of vj in the shortest path. Each vertex vj may has more than one adjacent vertices.

IV. THE IMPROVED DIJKSTRA'S LABEL ALGORITHM

According to the inadequate of Dijkstra's label algorithm and the corresponding improvement, this paper proposes an improved algorithm of Dijkstra's label algorithm.

1) Basic Definition

(1). Suppose $l_i^{(r)*}$ is the weight of the shortest path between v_1 and v_i. If v_i obtain the label $l_i^{(r)*}$ then v_i obtain the "p-label" $l_i^{(r)*}$ (permanent label) in step r ($r \geq 0$).
(2). Suppose $l_j^{(r)}$ is the upper of the shortest path from v_1 to v_j .If v_j obtain $l_j^{(r)}$ then v_j obtain "t-label" $l_j^{(r)}$(temporary label) in step r.
(3). Set P_r= { v | v has obtained "p-label"} to be "pass vertex set" in step r.
(4). Set T_r =V- P_r to be "not pass vertex set" in step r.
(5). Set A_i to be " v_i 's adjacent vertices set".
(6).Set N_r to be "vertices which obtain p-label" in step r.

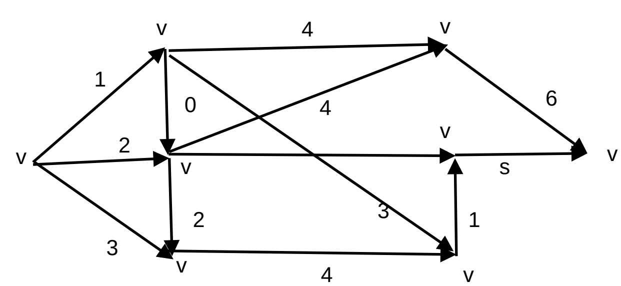
2) Improved Algorithm

Initial: $r \leftarrow 0$, v_1 obtain "p-label": $l_1^{(0)*}=0$,
$P_0=\{v_1\}$, $T_0=V-\{v_1\}$, v_j 's "t-label" :
$l_j^{(0)}= w_{1j}$, if $l_j^{(0)} \neq \infty$ then $A_j=\{v_1\}$
else $A_j=\emptyset$ ($j \neq 1$).
①.Find next "p-label" vertex.
set min $l^{(r-l)}=\min_{v_j \in T_{r-v}} \{ l_j^{(r-1)} \}$,
$r \geq 1$.$N_r=\emptyset$. if min $l^{(r-l)}=\infty$ then the algorithm end.
check $v_i \in T_{r-1}$: if $l_i^{(r-1)}=\min l^{(r-l)}$
then v_1 obtain the "p-label": $l_i^{(r)*}=\min l^{(r-l)}$,
update: $P_r=P_{r-1} \cup \{v_i\}$, $T_r=T_{0r-1}-\{v_i\}$,$N_r=N_r \cup \{v_i\}$.check T_r : if $T_r=\emptyset$ then the algorithm end
else jump ②。
②.Update each vertex's "t-label" in T_r according to N_r for $v_{jj} \in T_r$,$l_j^{(r)}=l_j^{(r-1)}$,for $v_i \in N_r$,
if $(l_j^{(r)*}+w_{ij})< l_j^{(r)}$ then $l_j^{(r)}=(l_i^{(r)*}+w_{ij})$,$A_j=\{v_i\}$
if $(l_i^{(r)*}+w_{ij})=l_j^{(r)}$then $A_j=A_j \cup \{v_i\}$ set
$r \leftarrow r+1$, jump ①.

V. EXPERIMENT OF THE IMPROVED ALGORITHM

According to the improved Dijkstra algorithm", this paper conducted experiment.

Experiment



Using the improved "Dijkstra algorithm" to calculate the shortest path between v_1 and other vertices.

e-ISSN: 2250-0758 | p-ISSN: 2394-6962
Volume-13, Issue-3 (June 2023)
https://doi.org/10.31033/ijemr.13.3.34

The process is Table II

v_1 R	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
1	0	$1/v_1$	$2/v_1$	$3/v_1$	∞	∞	∞	∞
2		$1/v_1$	$1/v_2$	$3/v_1$	$4/v_2$	∞	$5/v_2$	∞
3			$1/v_2$	$3/v_{1,}v_3$	$4/v_2$	$5/v_3$	$5/v_{2,}v_3$	∞
4				$3/v_{1,}v_3$	$4/v_2$	$5/v_3$	$5/v_{2,}v_3$	∞
5					$4/v_2$	$5/v_{3,}v_5$	$5/v_{2,}v_3$	∞
6						$5/v_{3,}v_5$	$5/v_{2,}v_3$	$6/v_6$
7								$6/v_6$

Experimental Analysis

A vertex may has many adjacent vertices. For example, v_4 has 2 adjacent vertices: v_1 and v_3. All the shorted paths:

$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_6 \rightarrow v_8$

$v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6 \rightarrow v_8$(weight 6)

$v_1 \rightarrow v_2 \rightarrow v_7$ $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_7$(weight 5)

$v_1 \rightarrow v_4$ $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$(weight 3)

VI. CONCLUSION

In this paper, we analyzed Dijkstra's "label algorithm", pointed out the inadequacies of the algorithm and proposed the improved methods.On this subject, this paper proposed the improved algorithm and conducted a series of targeted experiments. Experiment results indicate that the improved algorithm can not only solve the shortest path problem of undigraph but also can solve the shortest path problem of digraph. The improved algorithm is better than the original algorithm: The improved algorithm can get adjacent vertices(specific to the previous vertices) in the shortest path.

The efficiency of Dijkstra's "label algorithm" is low. Next step we will continue improve Dijkstra's "label algorithm" and to improve its efficiency. Dijkstra's "label algorithm" is widely used. Next step we will research the application of the algorithm.

REFERENCES

[1] Geng Su-Yun. (2008). *Discrete mathematics*. (4^{th} ed.), Tsinghua University Press.
[2] Zhang Yi. (2009). Application of an improved dijkstra algorithm in multicast routing problem. *Journal of Computer Science*.
[3] Zhang Jin-Yang. (2009). Optimization strategies of the Dijkstra's shortest route algorithm. *Journal of Science of Surveying and Mapping*.
[4] Ku Xiang-Yang. (2009). The shortest path algorithm with constraints. *Journal of Computer App locations and Software*.
[5] Wang Hai-Xiao. (2009). To apply on dijkstra algorithm in solving the shortest path of logistics transporting. *Journal of Value Engineering*.
[6] Liu Ping. (2008). An improved algorithm of shortest paths based on dijkstra algorithm. *Journal of Qinghai Normal University(Natural Science)*.
[7] Wen Xiao-Lei. (2009). Shortest path optimization based on mixed algorithm. *Journal of Tianjin University of Technology*.
[8] Li Jing-Jing. (2009). Analysis of shortest path algorithms in intelligent traffic system. *Journal of Computer & Digital Engineering*.
[9] Wang Lin. (2009). Analysis of the shortest path algorithm in ITS. *Technology & Economy in Areas of Communications*.
[10] Gu Yun-Yun. (2008). SEVERAL optimization for the shortest path search algorithm. *Journal of Computer Applications and Software*.
[11] Chen Ling-Min. (2009). The research and application of shortest path algorithm in highway networked toll collection. *Journal of Guizhou University (Natural Sciences)*.