Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1186 – 1190

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

The Improved Dijkstra's Shortest Path Algorithm and Its Application

Wang Shu-Xi*

School of Information Technology, The University of International Business and Economics, Bei Jing, China, 100029

Abstract

The shortest path problem exists in variety of areas. A well known shortest path algorithm is Dijkstra's, also called "label algorithm". Experiment results have shown that the "label algorithm" has the following issues: I.. Its exiting mechanism is effective to undigraph but ineffective to digraph, or even gets into an infinite loop; II. It hasn't addressed the problem of adjacent vertices in shortest path; III.. It hasn't considered the possibility that many vertices may obtain the "p-label" simultaneously. By addressing these issues, we have improved the algorithm significantly. Our experiment results indicate that the three issues have been effectively resolved.

Keywords: Shortest Path; Label Algorithm; Dijkstra; p-label; t-label

1. Introduction

In the process of production, organization and management, we need to solve many shortest path problems. For example, in the process of production, to complete production tasks quickly and with efficacy, we should find the shortest path to complete each production task; in the process of management, to make large gains with minimal cost, we should develop rational plans; in the existing transport network, to transport large quantity of goods with minimal costs, we should arrange for reasonable transport path. All these questions can be summed up as the "shortest path problem".

* * Corresponding author. Tel.: +86-013651056109.
E-mail address: wangshuxi2006@sina.com.

doi:10.1016/j.proeng.2012.01.110

The problem to find the shortest path between starting vertex and terminal vertex (which exist in a given network diagram) is widely used in various fields, such as: computer network routing algorithm, the robot Pathfinder, route navigation, game design, and so on.

The structure of the paper is as follows: first, we introduce Dijkstra's "label algorithm"; second, we point out that the algorithm needs to be improved through experiments; thirdly, we propose an improved algorithm and verified the algorithm through experiments. Experimental results show: the improved algorithm is more effective than "label algorithm", and can solve inadequacies of "label algorithm" effectively.

2. Dijkstra's "Label Algorithm"

Dijkstra's "label algorithm" which was proposed in 1959 is one of the best shortest path algorithms.

"Label algorithm" has a very wide range of applications, such as: multi-point routing[2], surveying and mapping science [5], the shortest path of logistics and transport [8], the intelligent transportation system [9], the expressway network toll collection [4], and so on.There are many related research about the shortest path algorithm and Dijkstra's "label algorithm" [3] [6][7].

2.1. The content of Dijkstra's "label algorithm" [1]

Suppose G = <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_i obtain the "p-label" $l_i^{(r)*}$ (permanent label) in step r(r≥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 $P_r = \{v \mid v$ 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's "label algorithm" is the following:

Initial: r←0, $l_1^{(0)*}=0$, $P_0=\{v_1\}$, T_0 =V-$\{v_1\}$, $l_j^{(0)}=w_{ij}$ (j≠1). (v_1 obtain "p-label", v_j obtain "t-label")

①. Find next "p-label" vertex.

Set $l_i^{(r)*}$ =min$\{l_j^{(r-1)}\}$(r≥1). v_i obtain "p-label" $l_i^{(r)*}$. Update the "pass vertex set" and the "not pass vertex set": $P_r=P_{r-1}\cup\{v_i\}$, $T_r=T_{r-1}$-$\{v_i\}$.

check T_r : If $T_r=\varnothing$ then the algorithm end else jump ②.

②. Update each vertex's "t-label" in T_r :

$l_j^{(r)}$ =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.

Dijkstra's label algorithm can effectively solve the shortest path problem of simple weighted undigraph. But Dijkstra's label algorithm is inadequate and need to be improved. The following will analyze each of them and propose the improvement.

2.2. The Exit Mechanism of Dijkstra's Label Algorithm

The exit mechanism of Dijkstra's label algorithm is: check T_r ("not pass vertex set" in step r, r≥0), if $T_r=\varnothing$ then the algorithm end.

Such an exit mechanism is effective to undigraph but ineffective to digraph. For example, if two vertices are disconnected in a digraph, the system will not shut down and fall into an infinite loop according to the algorithm.

To solve the problem, this paper propose the following exit mechanism: check T_r ("not pass vertex set" in step r, r≥0), If $T_r = \varnothing$ then the algorithm end; calculate "p-label" $l_i^{(r)*} = \min\{ l_j^{(r-1)} \}$(r≥1), if $l_i^{(r)*} = \infty$ then the algorithm end.

2.3. The adjacent vertices in the shortest path

Dijkstra's label algorithm did not specify how to get adjacent vertices (specific to the previous vertices) in the shortest path.

While in practice, it is often needed to find the adjacent vertices in the shortest path. So Dijkstra's label algorithm needs to be improved.

This paper proposes the following improvements: 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 v_j in the shortest path. Each vertex v_j may has more than one adjacent vertices.

2.4. More than one vertices obtain the p-label simultaneously

Dijkstra's label algorithm ignores the problem that many vertices may obtain the "p-label" simultaneously. Thus Dijkstra's label algorithm should be improved.

This paper proposes the following improvements: set $l_i^{(r)*} = \min\{ l_j^{(r-1)} \}$(r≥1) and v_i obtain the "p-label" $l_i^{(r)*}$. Update P_r and T_r : $P_r = P_{r-1} \cup \{ v_i \}$, $T_r = T_{r-1}$-$\{ v_i \}$. If many vertices have the same "t-label" then these vertices obtain the "p-label" simultaneously.

3. 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.

3.1. Basic definitions

(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≥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 $P_r = \{ v \mid v$ 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.

3.2. Improved algorithm

The following is the improved algorithm:

Initial: r←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 $A_j = \varnothing$ (j≠1).

①. Find next "p-label" vertex.

set $\min l^{(r-1)} = \min\limits_{v_j \in T_{r-1}} \{ l_j^{(r-1)} \}$, r≥1. $N_r = \varnothing$.
if $\min l^{(r-1)} = \infty$ then the algorithm end.
check $v_i \in T_{r-1}$: if $l_i^{(r-1)} = \min l^{(r-1)}$ then v_i obtain the "p-label": $l_i^{(r)*} = \min l^{(r-1)}$,
update: $P_r = P_{r-1} \cup \{ v_i \}$, $T_r = T_{r-1} - \{ v_i \}$, $N_r = N_r \cup \{ v_i \}$.
check T_r :if $T_r = \varnothing$ then the algorithm end else jump ②。
②.Update each vertex's "t-label" in T_r according to N_r
for $v_j \in T_r$, $l_j^{(r)} = l_j^{(r-1)}$,
for $v_i \in N_r$,
if $(l_i^{(r)*} + w_{ij}) < l_j^{(r)}$ 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←r+1, jump ①.

4. Experiment of the Improved Algorithm

According to the improved Dijkstra's "label algorithm", this paper conducted an experiment.

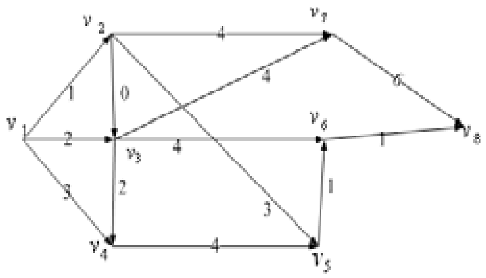

Fig. 1. Directed Graph

Using the improved "label algorithm" to calculate the shortest path between v_1 and other vertices in the above weighted digraph. The process is the following table:

Table 1. The Processing Process

R \ V_i	V_1	V_2	V_3	V_4
1	$\boxed{0}$	$1/v_1$	$2/v_1$	$3/v_1$
2		$\boxed{1}/v_1$	$1/v_2$	$3/v_1$
3			$\boxed{1}/v_2$	$3/v_1, v_3$
4				$\boxed{3}/v_1, v_3$
5				
6				
7				

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)

5. Conclusion and Future Work

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: a. The improved algorithm's exit mechanism is improved so that the algorithm will avoid falling into an infinite loop. b. The improved algorithm can get adjacent vertices (specific to the previous vertices) in the shortest path. c. The improved algorithm solved the problem of more than one vertices obtain "p-label" at the same time.

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, "Discrete Mathematics"(4th edition), Tsinghua University Press, 2008.3.

[2] Zhang Yi, Application of Improved Dijkstra Algorithm in Multicast Routing Problem, Journal of Computer Science, 2009.8.

[3] Zhang Jin-Yang, Optimization of the Dijkstra shortest route algorithm, Journal of Science of Surveying and Mapping, 2009.9.

[4] Ku Xiang-Yang, The Shortest Path Algorithm With Constraints, Journal of ComputerApp lications and Software, 2009.5

[5] Wang Hai-Xiao, Dijkstra Algorithm in the Shortest Path of Logistics Transporting, Journal of Value Engineering, 2009.5.

[6] Liu Ping, An algorithm of shortest paths based on dijkstra algorithm, Journal of Value Engineering, 2008.1.

[7] Wang Lin, Analysis of the Shortest Path Algorithm in ITS, Technology & Economy in Areas of Communications,2009.4.

[8] Li Jing-Jing, Analysis of Shortest Path Algorithms in Intelligent Traffic System. Journal of Computer, 2009.4.

[9] Gu Yun-Yun, Optimization For The Shortest Path Search Algorithm, Journal of ComputerApplications, 2008.4.