Prim Algorithm Based on Heap for Finding Minimal Spanning Tree of Ventilation Network

Buchuan Wang & Qinglong Liu & Wenbin Wu

China coal technology engineering group Chongqing research institute, Chongqing, China

Keywords: ventilation network; minimum spanning tree; directed weighted graph; heap; Prim algorithm.

Abstract. Based on ventilation network graph is rooted directed weighted graph, its' nature of minimum spanning tree is proposed. According to the dense of ventilation network graph, introduces heap to improve the Prim algorithm, gives C++ implementation for the algorithm, and finally gains the Prim algorithm based on heap is both suitable for dense graph and sparse graph by analyzing its' time complexity, so the algorithm can meet the requirement to seek the minimum spanning tree of ventilation network graph.

1. Introduction

Mine ventilation system composed of crisscross roadway is a complex system. Describes ventilation system by the method of graph theory, ventilation network is a ventilation system which is composed of line, point and its properties. At present, calculation of ventilation network commonly used by Scott-Hinsley method, which needs determine the minimum spanning tree of network graph at first. The minimal spanning tree of an undirected weighted graph is a typical problem of graph theory, scholars of the world have conducted considerable research on it until now. The ventilation network graph is a weighted directed graph, there is a few research on the minimal spanning tree of a directed weighted graph. In addition, researches on dense degree of ventilation network graph isn't enough, scholars only qualitatively consider that ventilation network graph is dense graph.

For shorting the time of calculating ventilation network by minimal spanning tree, it needs improve the Prim algorithm that is based on ventilation network graph is a directed weighted graph and its dense degree.

2. Nature of minimal spanning tree of ventilation network

It supposes that *G(V,E)* is a nonnegative weighted directed graph, *w(e)* is a weight at every directed edge in graph *G*. Subgraph *T* is a tree in graph *G*, and it contains all the vertex of graph G, so *T* is a spanning tree of graph *G*. The weight of spanning tree *T* is $weight(T) = \sum_{e \in T} w(e)$, the minimum weight tree is the minimum spanning tree, So how to choose tree *T* is a difficult problem.

According to the connectivity of graph, directed weighted graph is divided into the following three types:

(1) Strongly connected directed graph: It is directed graph that can visit all vertices from any vertex.

(2) Rooted directed graph: It is a directed graph that can be iterated from its roots.

(3)None strongly connected directed graph: The graph is not strongly connected directed graph and the source points aren't roots of directed graph, it is a directed graph that only gets spanning forest when it is iterating.

The former two kinds of weighted directed graph have feasible solution of minimal spanning tree. the nature of minimal spanning tree as follows[2]: It suppose that graph *G(V,E)* is a nonnegative weighted directed graph, graph *T(U,D)* is one subgraph of graph *G*, and graph *T* is the minimum spanning tree of itself. If *(x, y)* is a directed edge that one vertex is in *U* and another isn't in *U*, and the weight of that edge is minimum, and the in-degree of head *y* in subgraph *T* is 0 (The in-degree of head *y* is 0 if head *y* isn't in subgraph *T*.), and then there must be a minimal spanning tree which contains directed edge *(x, y)*.

It supposes that point V_{in} is the intake air node, air flow starts from point V_{in}, and it can reach all other nodes in the ventilation network graph. So ventilation network graphs are rooted directed graph, and roots of graph are intake air nodes V_{in}.

Above all, the ventilation network graph is a rooted directed graph that there is feasible solution of the minimum spanning tree, its nature of minimal spanning tree is that of directed graph.

3. Prim algorithm for minimal spanning tree of a directed graph

Algorithm of finding minimum spanning tree of weighted undirected graph contains Breaking Loop (Rosenstiehl), Kruskal Algorithm, Prim Algorithm, Dijkstra Algorithm. Prim Algorithm and Kruskal Algorithm that are suitable for finding minimal spanning tree by computer are two of the most common algorithms. Time complexity of Prim Algorithm is $O(n^2)$,and it is suitable for dense graph. Time complexity of Kruskal Algorithm is $O(m\log_2 m)$ and it is suitable for dense graph.

Based on the minimal spanning tree nature of weighted directed graph, through minimum spanning tree Prim algorithm of undirected weighted graph is described again, that get the thought of minimum spanning tree Prim algorithm of directed weighted graph: Start from any vertex U_0 of graph *G*, choose the minimum weight directed edge (u_0,v) or edge (v,u_0) which is connect with vertex U_0, add the vertex *v* to the vertices in set *U* of spanning tree. Choose a directed edge *(x, y)*, that is in all directed edges that one vertex is in *U* and the other one isn't in *U*, whose weight is minimum and head *y* is in subgraph *T*, and the in-degree of edge (x, y) is 0. And then add all vertexes that is in *x* and *y* to set *U*. And so on, it get the minimum spanning tree of a directed graph until all vertexes in graph *G* are added to set *U* of vertexes in spanning tree.

4. Prim algorithm based on heap for minimal spanning tree of a directed graph

Dense of graph can be reflect through the gap between zero degree and rank of the graph. If the gap is positive number, the graph is dense graph; if the gap is negative number, it is sparse graph; if the gap is zero, it is medium dense graph. For a connected graph whose number of node is *n* and whose number of branch is *m*, its rank is *n-1* and its zero degree is *m-n+1*. There are 7 ventilation networks in table 1, and then the dense degree is analyzed.

After careful analysis, ventilation network contains sparse graph and dense graph, but Prim algorithm is only suitable for dense graph and Kruskal algorithm is only suitable for sparse graph. In order to make Prim algorithm suitable for dense graph and sparse graph, that is improved by thought of heap, and then that will meet the requirement of finding minimum spanning tree for ventilation network graph.

When it finds the minimum edge by common Prim algorithm which needs compare again every time, and it can't save the compared information before. It can make full use of original information to choose the next minimum edge after it adds thought of heap.

Tab.1 Analysis of ventilation network graph dense degree

Serial number	Number of nodes	Number of branch	Zero degree	Rank	Zero degree - Rank	Density
1	11	17	7	10	－3	sparse graph
2	15	25	11	14	－3	sparse graph
3	23	48	26	22	4	dense graph
4	27	61	35	26	9	dense graph
5	82	111	30	81	－51	sparse graph
6	104	161	58	103	－45	sparse graph
7	255	400	146	254	－108	sparse graph

4.1 Adjustment of heap

Method of heap adjustment is shown in figure 1.

(1) Adjust the numerical value of node whose subscript is i, but it is bigger than its parent node, at this time, it doesn't need to adjust and it is also a rootlet heap.

(2) Adjust the numerical value of node whose subscript is i, and it is smaller than its parent node, at this time, it needs adjust to keep a rootlet heap.

There are three that call function of heap adjustment in Prim algorithm based on heap:

(1) At preliminary establishment of heap, it needs call function of heap adjustment until it get a rootlet heap, function shift() is called.

(2) It can cause heap adjustment when it determines a minimum weight edge, function shift() is called.

After choose a minimum weight edge, one of the endpoints will go into the minimum spanning tree, if it turn down the shortest distance of the other nodes to the minimum spanning tree, it needs adjust heap that cause weight of node lower in heap, function Adjust() is called.

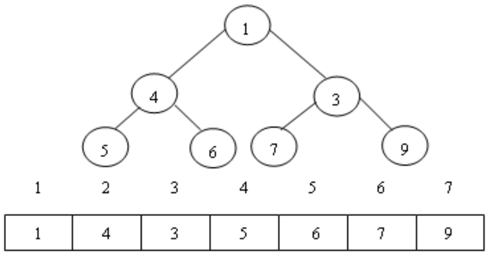

Fig.1 Rootlet heap and its Storage structure

4.2 Solution of C++

```
#include "iostream.h"
#include "math.h"
#define M 100.00
#define N 6
typedef struct
{int from, end; double weight;}Edge;
Edge MST[N];
void Sift(Edge* R, int m, int n)
{int i, j; Edge x; x=R[m]; i=m; j=2*i;
while(j<=n)
{if ((j<n)&&(fabs(R[j].weight)>fabs(R[j+1].weight))) j++;
if(fabs(x.weight)>fabs(R[j].weight))
{R[i]=R[j]; i=j;j=j*2;}
else break;
}
 R[i]=x;
}
void Adjust(int i)
{Edge t; int m, n; if(i==1) return;
t=MST[i];m=i;n=i/2;
while(n>=1)
{if(fabs(t.weight)<fabs(MST[n].weight))
 {MST[m]=MST[n];m=n; n=m/2;}
else break;}
MST[m]=t;
}
void Prim(double cost[N+1][N+1])
{ int v; double d; Edge e;
for(int i=1;i<=N-1;i++)
{MST[i].from=1;MST[i].end=i+1;MST[i].weight=cost[1][i+1];            }
for(i=(N-1)/2;i>=1;i--) Sift(MST, i, N-1);
for(int k=1; k<N; k++)
{e=MST[1];MST[1]=MST[N-k]; MST[N-k]=e;
Sift(MST,1,N-k-1); v=MST[N-k].end;
for ( int j=1; j<N-k; j++)
{d=cost[v][MST[j].end];
if(fabs(d)<fabs (MST[j].weight))
{MST[j].weight=d;MST[j].from=v; Adjust(j);}
}}}
void main()
{double weight[N+1][N+1];
for(int i=1;i<=N; i++)
{for(int j=1;j<=N; j++)
cin>>weight[i][j];}
Prim(weight);
for(i=N-1;i>=1;i--)
{cout<<"("<<MST[i].from<<","<<MST[i].end<<","<<MST[i].weight<<")"<<";";}}
```

4.3 Analysis of algorithmic time complexity

From the above procedures it can be seen: initialization time of edge *n-1* in planning tree is *O(n)*; time of preliminary establishment heap is *O(n)*. After determine the minimum weight edge, it calls *Sift()*, the time is $O(n\log_2 n)$. The number of calling *adjust ()* isn't more than *m* that is the number of edge of graph, the time is $O(m\log_2 n)$. For a graph, *m* is greater than or equal to *n*, so the total time complexity is $O(m\log_2 n)$. Time of finding the spanning tree of dense graph is less than that by common Prim algorithm, and time of finding the spanning tree of sparse graph is similar with that by Kruskal algorithm. So Prim algorithm based on heap is suitable for dense graph and sparse graph, and it can meet the requirements of minimum spanning tree of ventilation network graph.

5. Application example

There is a connecting branch between the two parallel branches in the diagonal ventilation network that make one branch connect to another, and then it forms a ventilation network. Ventilation network that only has one connecting branch is called simple diagonal network, ventilation network that has two connecting branches or more is called complex diagonal network.

Figure 2 is a mine ventilation network, and it is a complex diagonal network that contains fans, so that makes it as a common example. R1 to R9 are windage of 9 branches: $R_1=1.47$,$R_2=1.372$,$R_3=4.116$,$R_4=1.176$,$R_5=0.784$,$R_6=0.98$,$R_7=8.82$,$R_8=0.588$,$R_9=2.\ 93$(Ns^2/m^8).

Input the adjacency matrix of directed graph, the result is that: (1, 2, 1.372), (2, 6, 1.176), (6, 3, 0.784), (3, 4,0. 588), (6, 5, -0.98), the minimum spanning tree is shown in figure 3.

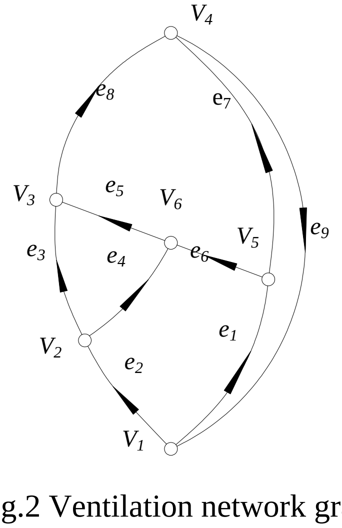

Fig.2 Ventilation network graph

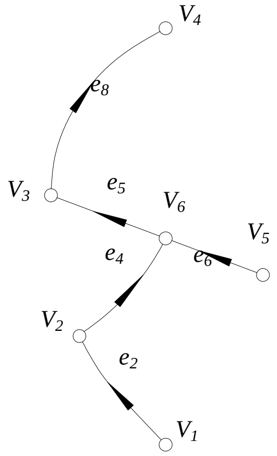

Fig.3 The minimum spanning tree of ventilation network graph

6. Conclusions

Through the above analysis, it obtains the following conclusions:

(1) Ventilation network graph is a rooted directed graph, and it has the nature of the minimum spanning tree of directed graph.

(2) Ventilation network graph contains dense graphs and sparse graphs, Prim algorithm is suitable for dense graphs, and Kruskal algorithm is suitable for sparse graphs.

(3) It introduces the Prim algorithm that is based on heap, and it also raises the solution of C++, through analyzing time complexity of algorithm is $O(mlog_2n)$; Prim algorithm which is based on heap is suitable for dense graph and sparse graph, it is suitable for finding the minimum spanning tree of ventilation network graph, so it is more suitable for calculating ventilation network of large mine.

Acknowledgement

This work was financially supported by National Science and Technology Major Project (201 1ZX 05041-003-03), the Chongqing Natural Science Foundation (cstc2012jjA9003) and the National Natural Science Foundation (51304237).

References

[1] Jiang JC, Chen Q. Aplication of computer in calculating optimization regulation of mine ventilation network [J]. Coal Technology of Northeast China, 1995.1:57-61.

[2] Lan JL, Liu J. Graph-theory of application and Algorithm [M]. Chengdu: University of Electronic Science and Technology China Press, China, 1995.

[3] Liu ZX, Xie XY. Improve the speed of ventilation network calculation by side adding method [J] Coal Science and Technology 2000; 28(5):30-31,45.

[4] Liu J, Jia JZ, Zheng D. Fluid Network Theory [M]. Beijing: China coal industry publishing house, China, 2002.

[5] Sun LY, Leng M, Tan YL, Yu SN. Minimum spanning tree algorithm of weighted directed Graph [J]. Computer Engineering, 2010; 36(2):61-66.

[6] Wang G, Bi YS, Wen YY, etc. Conception of greedy algorithm for mine ventilation network calculation [J]. Safety in Coal Mines, 2005; 36 (10):40-42.

[7] Wu C, Su Y, Yang HL, etc. Discussion about the two algorithms for finding the minimal spanning tree of an undirected weight graph [J]. Journal of East China Shipbuilding Institute, 2004; 18 (2):27-32.

[8] Xu RL. Ventilation Network Theory. Beijing: China coal industry publishing house, China, 1993:37.

[9] Yang YL. Evaluation of mine ventilation network structure complexity [J]. Safety in Coal Mines, 1998; 29(1):321-334.

[10] Zhang GS, Tan YZ, Chen KY, Liu ZG, Yang YL. Ventilation and safety [M]. Xuzhou: China University of Mining and Technology press. China, 2007.

[11] Zhang SY. Mine the research and implementation ventilation network real-time solution and safe early-warning system [D]. Xi’an: Northwestern Polytechnical University, China, 2005.