Article

Finding maximum flow in the network: A Matlab program and application

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 9 August 2016; *Accepted 6 October 2017*; *Published online 1 June 2018*

Abstract
Maximum flow problems are expected occurring in some biological networks. As early as in 1950s, Ford and Fulkcerson proposed an algorithm to find maximum flow in a network. In this study I presented the full codes of Ford-Fulkcerson algorithm and given an application example.

Keywords network; Ford-Fulkerson algorithm; maximum flow; Matlab.

Computational Ecology and Software
ISSN 2220-721X
URL: http://www.iaees.org/publications/journals/ces/online-version.asp
RSS: http://www.iaees.org/publications/journals/ ces/rss.xml
E-mail: ces@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Suppose there are n nodes in a network. The node v_0 is the source that has not any terminal link, and the node v_n is the sink node that has not any initial link. The weight c_{ij} of directed link e_{ij} is the flow capacity of the link. f_{ij} is the flow of the link, $0 \le f_{ij} \le c_{ij}$, and the in-flow sum and out-flow sum of node v_j are the same (Ford and Fulkcerson, 1956, 1957; Zhang, 2012, 2018)

$$\sum_{i=0}^{n-1} f_{ij} = \sum_{k=1}^{n} f_{jk}, \quad j=1,2,..., n-1$$

which means that the out-flow sum of source node is equal to the in-flow sum of sink node. The maximum flow problem is

$$\max \sum_{j=1}^{n} f_{0j}$$

Maximum flow problems are expected occurring in ecological networks and other types of biological networks. Ford and Fulkcerson (1957) proposed an algorithm to find maximum flow in a network. In this study I present the full Matlab codes of Ford-Fulkcerson algorithm and give an application example.

2 Algorithm

Ford-Fulkerson (1957) algorithm for maximum flow is as follows (Chan et al, 1982; Zhang, 2012, 2018):

(1) Labeling.

(a) Label the source node v_s with (+, +∞), $d_s = +\infty$.

(b) Choose a labeled node x. For all unlabeled adjacent nodes y of x, handle them by the following rules

If $yx\in E$ and $f_{yx}>0$, let $d_y=\min\{f_{yx}, d_x\}$, and label y with (x, d_y).

If $xy\in E$ and $f_{xy}<C_{xy}$, let $d_y=\min\{C_{xy}-f_{xy}, d_x\}$, and label y with (x+, d_y).

(c) Repeat the step (b) until the sink node v_t has been labeled or no more nodes can be labeled. If v_t is labeled, then there exists an augmenting chain, and return the procedure (2) to adjust the process; if v_t is not labeled, labeling process is not able to be conducted, and f is thus the maximum flow.

(2) Adjusting.

(a) Determine adjusting magnitude $d=d_{vt}$, and let $u=v_t$.

(b) If the node u is labeled by (v+, d_u), then replace f_{vu} with $f_{vu}+d$, and if the node u is labeled by (v, d_u), then replace f_{vu} with $f_{vu}+d$.

(c) If $v=v_s$, then remove all labels and return the procedure (1) for labeling again; otherwise let $u=v$, return (b).

If the computation terminates, let the set of labeled nodes be S, and cutset (S, S_c) is the minimum cut, and the maximum flow is $M_f=C(S, S_c)$.

The following are Matlab codes, maxiFlow.m, for Ford-Fulkerson algorithm. The Matlab algorithm needs the user to load an excel file that stores the Two Array Listing data of the form (d_{1i}, d_{2i}, c_i), where d_{1i}, d_{2i}, and c_i are start node and end node of the link i, and the flow capacity of the link i, respectively, i=1,2,…,e.

```
%Ford--Fulkerson algorithm for maximum flow in the network (graph).
%v: the number of nodes; Data are stored in Two Array Listing.
%d1(1-e), d2(1-e): start and end nodes of links; d(1-e): flow capacity of links; Nodes are sequentially numbered in the network.
d=input('Input the excel file name of Two Array Listing data (e.g., adj.xls, etc. Data is d=(d1i, d2i, ci),where d1i, d2i, and ci are start node and end node of the link i, and the flow capacity of the link i, respectively, i=1,2,…,e): ','s');
d=xlsread(d);
e=size(d,1);
d1=d(:,1); d2=d(:,2); dd=d(:,3);
v=max(max([d1 d2]));
c=zeros(v); f=zeros(v);
no=zeros(1,v); d=zeros(1,v);
for i=1:v
for j=1:v
for k=1:e
if ((d1(k)==i) & (d2(k)==j))
c(i,j)=dd(k);
break;
end; end; end; end
for i=1:v
for j=1:v f(i,j)=0;
```

```
end; end
for i=1:v
no(i)=0;
d(i)=0;
end
pr=1;
while(v>0)
no(1)=v+1;
d(1)=1e+30;
while(v>0)
pr=1;
for i=1:v
if (no(i)~=0)
for j=1:v
if ((no(j)==0) & (f(i,j)<c(i,j)))
no(j)=i;
d(j)=c(i,j)-f(i,j);
pr=0;
if (d(j)>d(i)) d(j)=d(i); end
end
if ((no(j)==0) & (f(j,i)>0))
no(j)=-i;
d(j)=f(j,i);
pr=0;
if (d(j)>d(i)) d(j)=d(i); end
end
end
end; end
if ((no(v)~=0) | (pr~=0)) break; end
end
if (pr~=0) break; end
dv=d(v);
s=v;
while (v>0)
if (no(s)>0) f(no(s),s)=f(no(s),s)+dv; end
if (no(s)<0) f(no(s),s)=f(no(s),s)-dv; end
if (no(s)==1)
for i=1:v
no(i)=0;
d(i)=0;
end
break;
end
s=no(s);
end; end
```

```
mf=0;
for j=1:v
mf=mf+f(1,j);
end
fprintf(['Maximum flow matrix:' '\n']);
f
fprintf(['Maximum flow=:' num2str(mf) '\n']);
fprintf(['Labels for minimum cut:' '\n']);
no
```

3 Application Example

As an example, suppose there are 8 nodes and 11 links in a network. The data are as follows

Node	Node	Flow capacity
1	2	5
1	3	1
1	4	3
2	4	2
2	6	3
3	5	1
3	8	3
4	7	2
5	8	7
6	7	3
7	8	4

Using the algorithm above, the maximum flow matrix is achieved as the following

0	2	1	2	0	0	0	0
0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	4
0	0	0	0	0	0	0	0

The maximum flow is 5. And the labels for minimum cut are, 9, 1, 0, 1, 0, 2, 6, 0.

References

Chan SB, et al. 1982. Graph Theory and Its Applications. Science Press, Beijing, China

Ford LR Jr, Fulkerson DR. l956. Maxima1 flow through a network. Canadian Journal of Mathematics, 8: 399-404

Ford LR Jr, Fulkerson DR. l957. A simple algorithm for finding maximal network flow and application to the Hitchcock problem. Canadian Journal of Mathematics, 9: 210-218

Zhang WJ. 2012. Computational Ecology: Graphs, Networks and Agent-based Modeling. World Scientific, Singapore

Zhang WJ. 2018. Fundamentals of Network Biology. World Scientific, Singapore