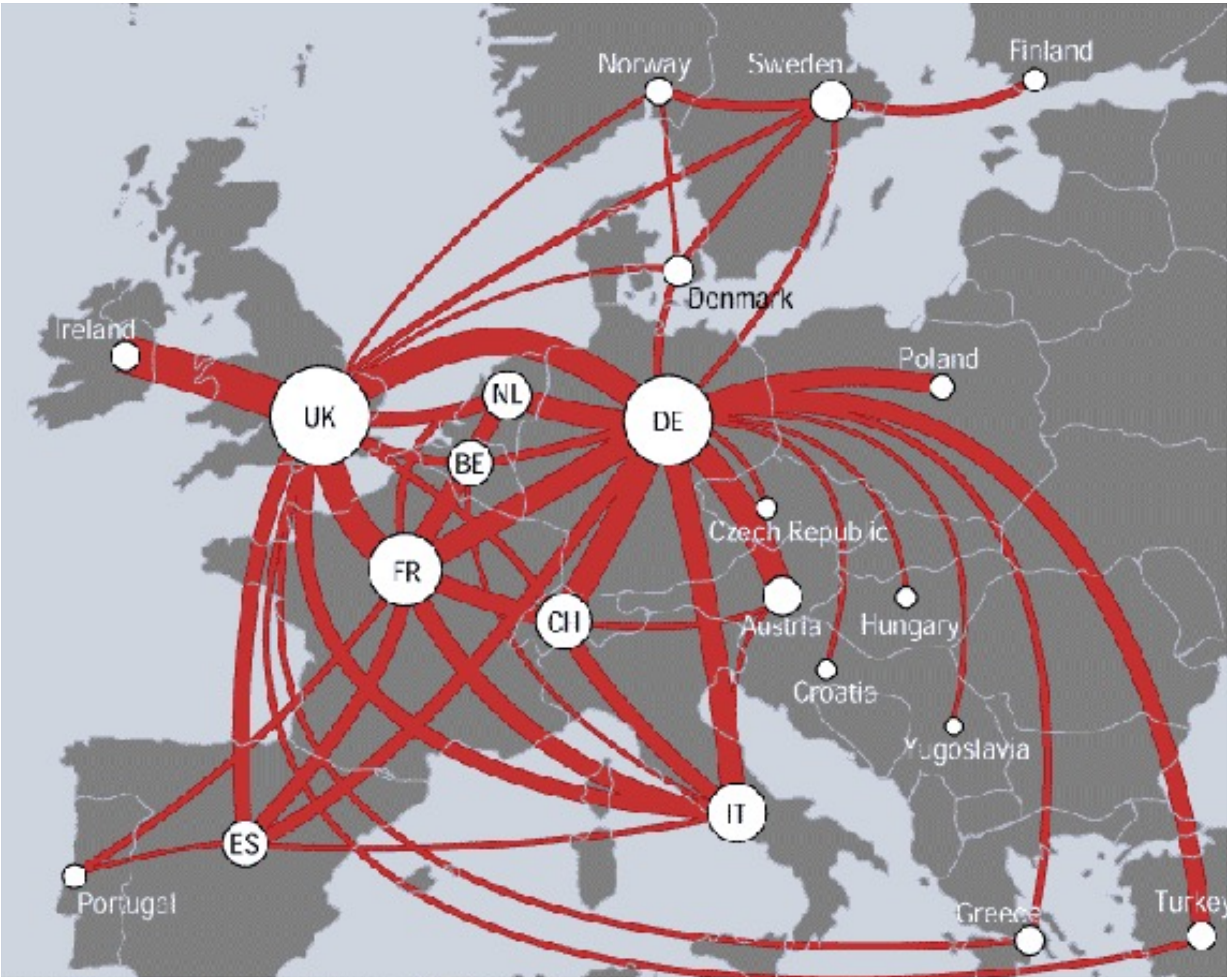
网络流

程龚 (gcheng@nju.edu.cn)



https://www.gislounge.com/wp-content/uploads/2014/04/network-flow-map.png

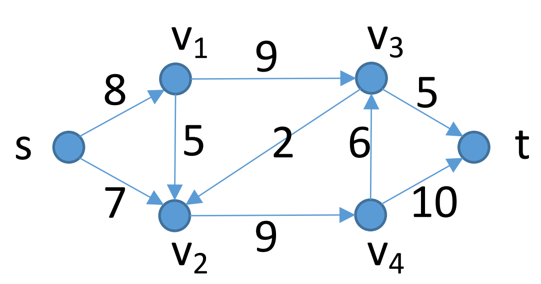
本节课的主要内容

9.1 网络与网络流的基本概念

9.2 最大流问题及其标号算法

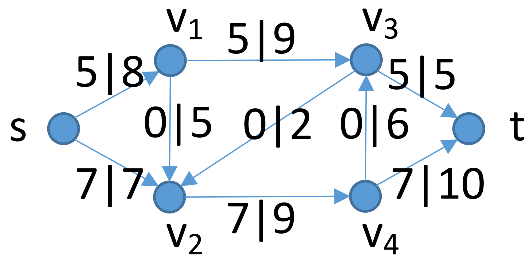
网络的基本概念

- 网络(network)：弧带权的有向图
  - 弧的权又称弧的容量(capacity)，记作$c(a) \geq 0$
  - 只讨论简单有向图（无环弧、并行弧）
  - 有一个特殊的源点(source vertex)，记作s
  - 有一个特殊的汇点(sink vertex)，记作t



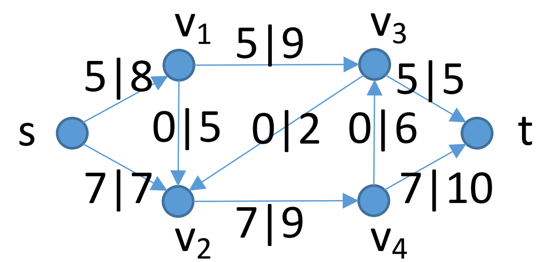
网络流的基本概念

- 流(flow)
  - f：定义在弧上的非负实值函数
  - $f^+(v)$：顶点v的所有出弧的流量和
  - $f^-(v)$：顶点v的所有入弧的流量和
- 可行流(feasible flow)
  - 容量约束：$\forall a \in A(G),\ 0 \le f(a) \le c(a)$
  - 守恒约束：$\forall v \in V(G) \setminus \{s,t\},\ f^+(v)=f^-(v)$
- 对于任意网络，可行流总是存在的（是什么？）
  - 零值流 (zero flow)



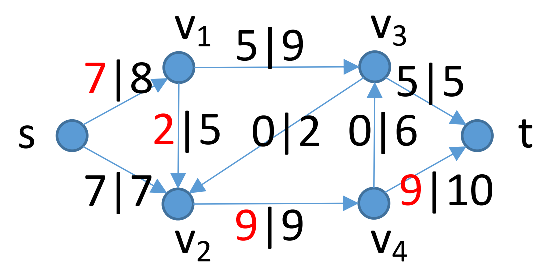
网络流的基本概念 (续)

- 流量 (value)
  - $f^{-}(t)-f^{+}(t)$
- 必有$f^{+}(s)-f^{-}(s)= f^{-}(t)-f^{+}(t)$，为什么？



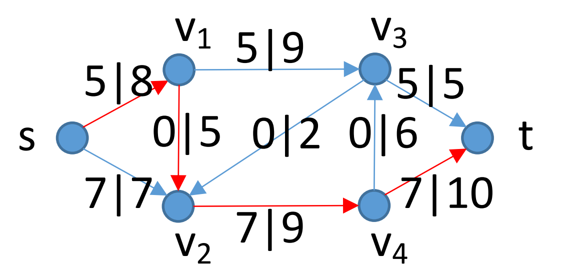
- 最大流 (maximum flow)
  - 流量最大的可行流

（右图是最大流吗？你是如何判断的？）

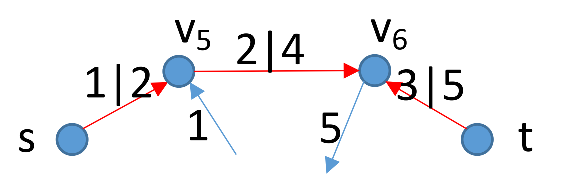


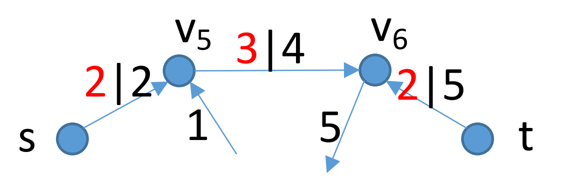
网络流的基本概念 (续)

- f增广路(f-augmenting path)
  - 底图中的一条s-t路
  - 经过的每条正向弧a∈A(G)：f(a)<c(a)
  - 经过的每条反向弧a∈A(G)：f(a)>0
- 增广路的“可增量”有多少？
  - 弧的可增量
    - 正向弧：c(a)-f(a)
    - 反向弧：f(a)
  - 增广路的可增量(tolerance)
    - 经过的弧的可增量的最小值



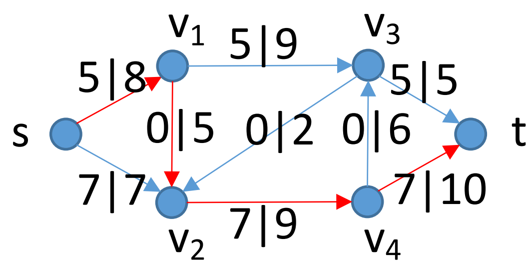
还有其它种类的增广路吗？

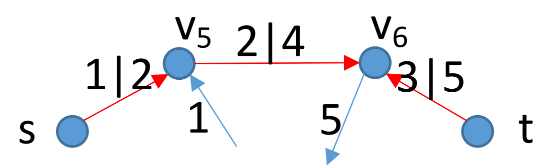




网络流的基本概念 (续)

- 对于可增量为z的f增广路，对f做如下调整，结果f’仍是一个可行流，且流量增加z：
  - 正向弧a：$f'(a)=f(a)+z$
  - 反向弧a：$f'(a)=f(a)-z$
  - 其它弧a：$f'(a)=f(a)$





证明：你能自己证明吗？

1、容量约束：$\forall a \in A(G),\ 0 \le f'(a) \le c(a)$

2、守恒约束：

3、流量增量：

$(f'^-(t)-f'^+(t))-(f^-(t)-f^+(t))$

$=(f'^-(t)-f^-(t))-(f'^+(t)-f^+(t))$

$=z-0$或$0-(-z)$

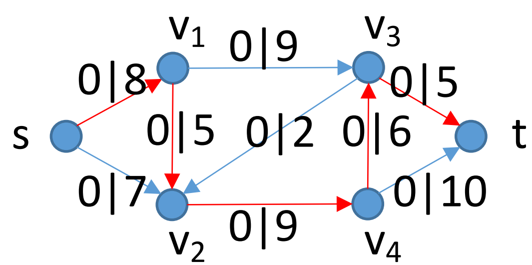
$=z$

Ford-Fulkerson标号算法

- 基本思路
  1. 从零值流开始
  2. 搜索一条增广路
  3. 如果找到了：调整得到流量更大的流，回到第2步
  4. 否则：结束

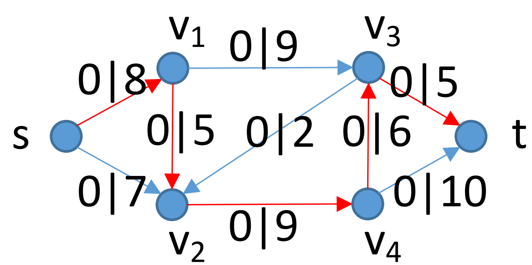
Ford-Fulkerson标号算法 (续)

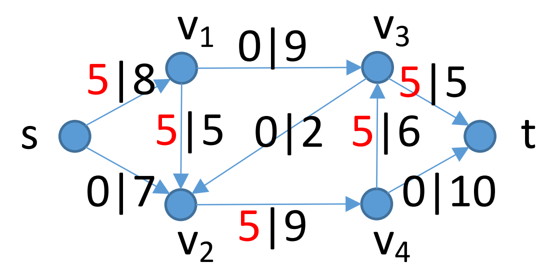
- 举例



Ford-Fulkerson标号算法 (续)

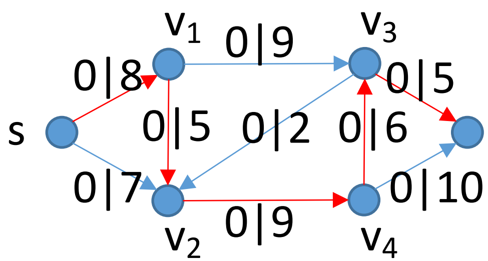
- 举例

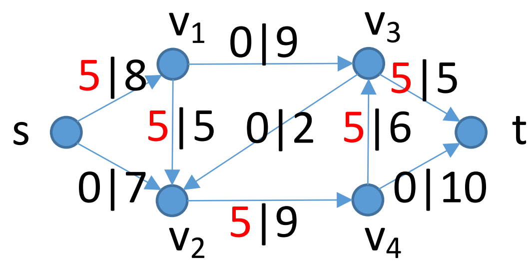


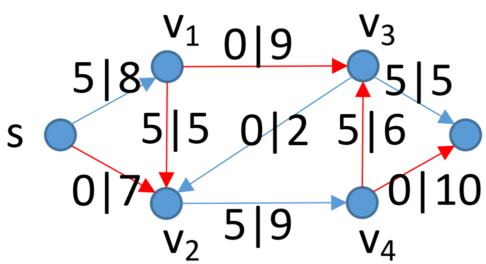


Ford-Fulkerson标号算法 (续)

- 举例

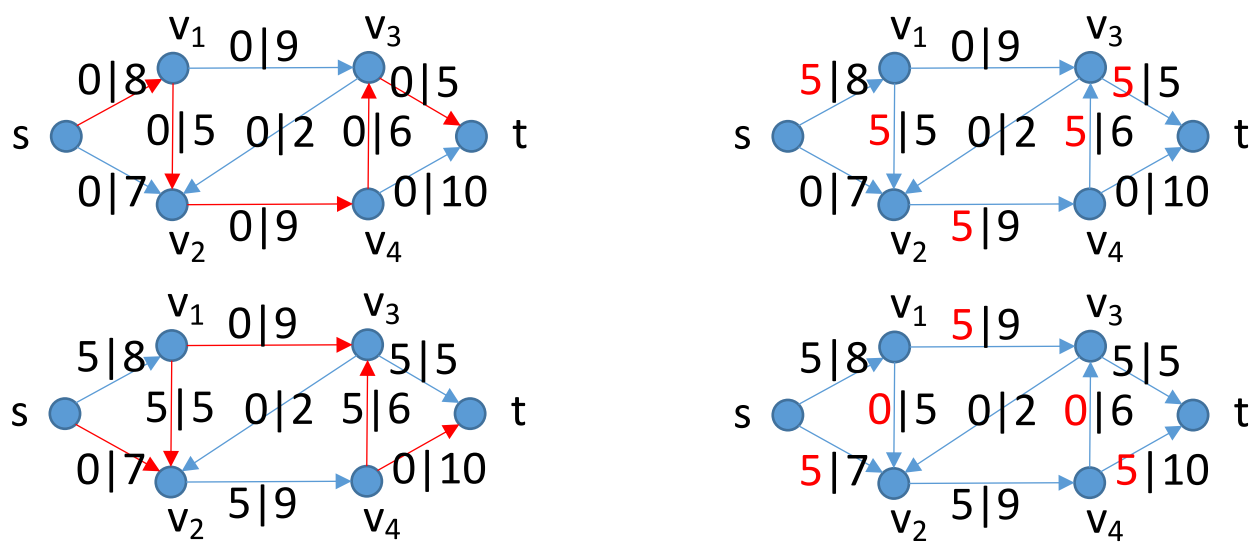






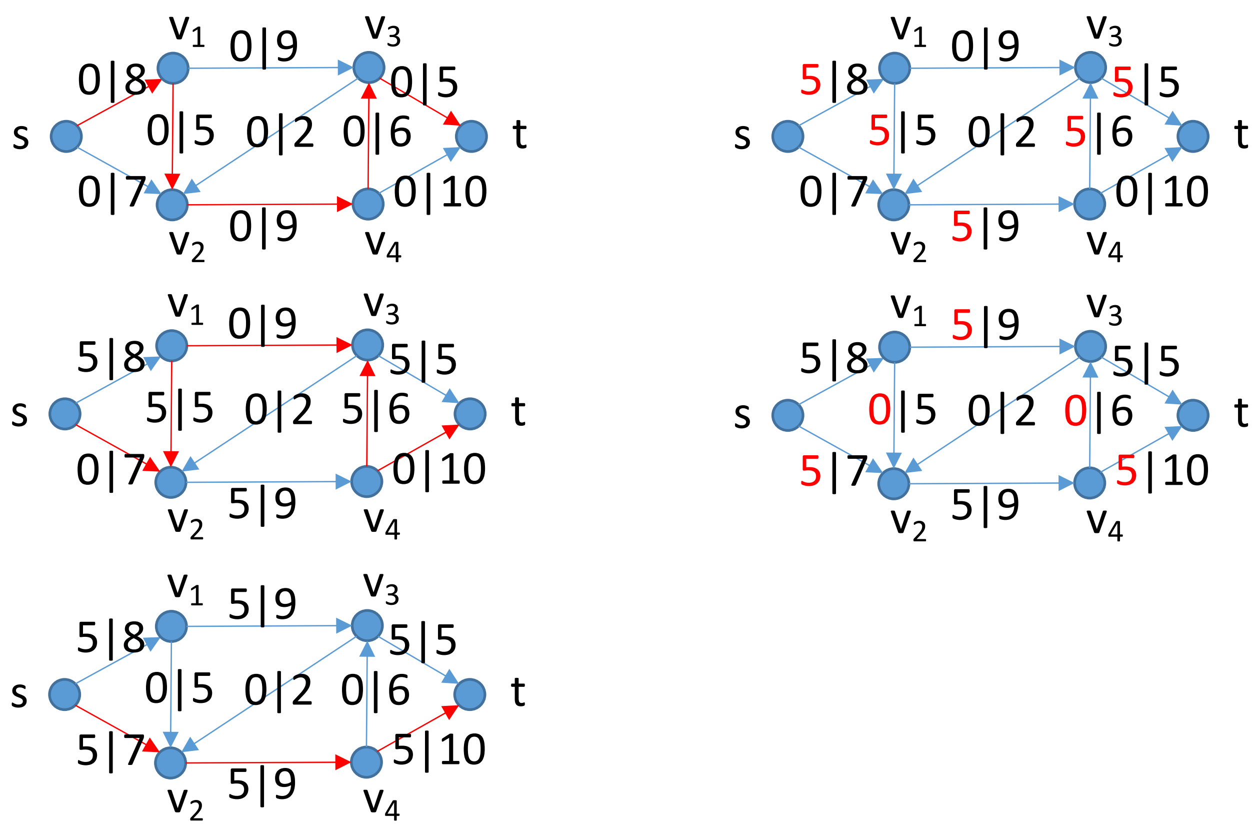
Ford-Fulkerson标号算法 (续)

- 举例



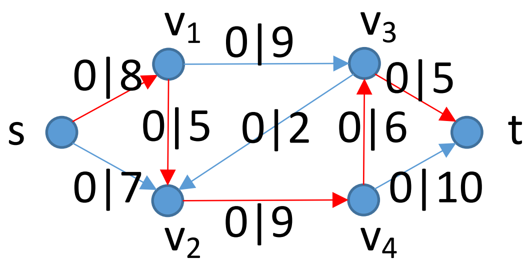
Ford-Fulkerson标号算法 (续)

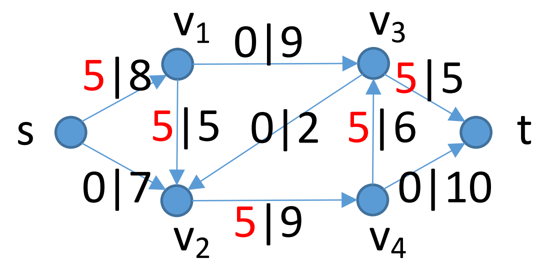
- 举例

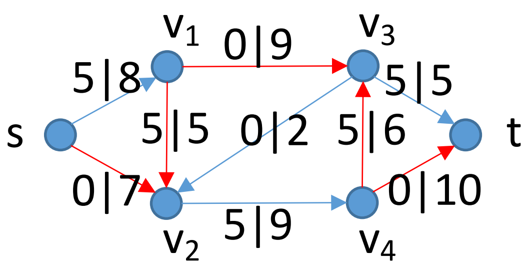


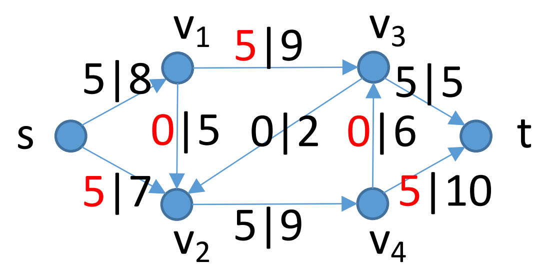
Ford-Fulkerson标号算法 (续)

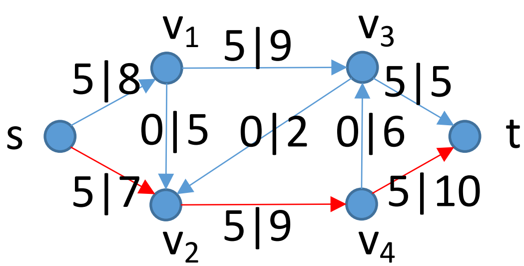
- 举例

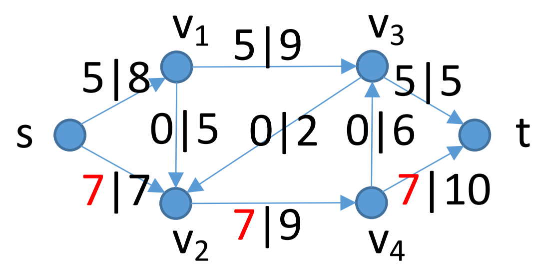






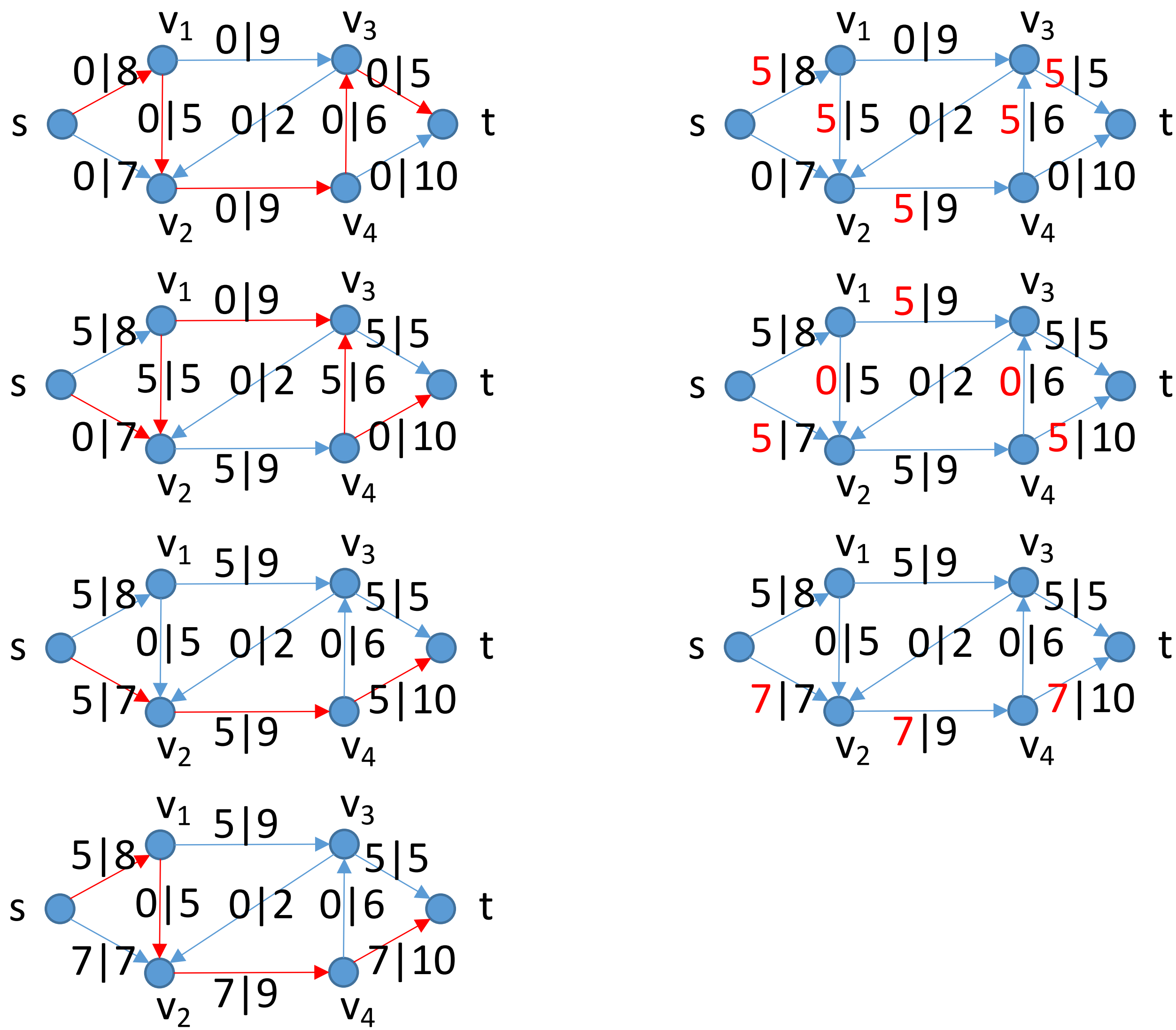






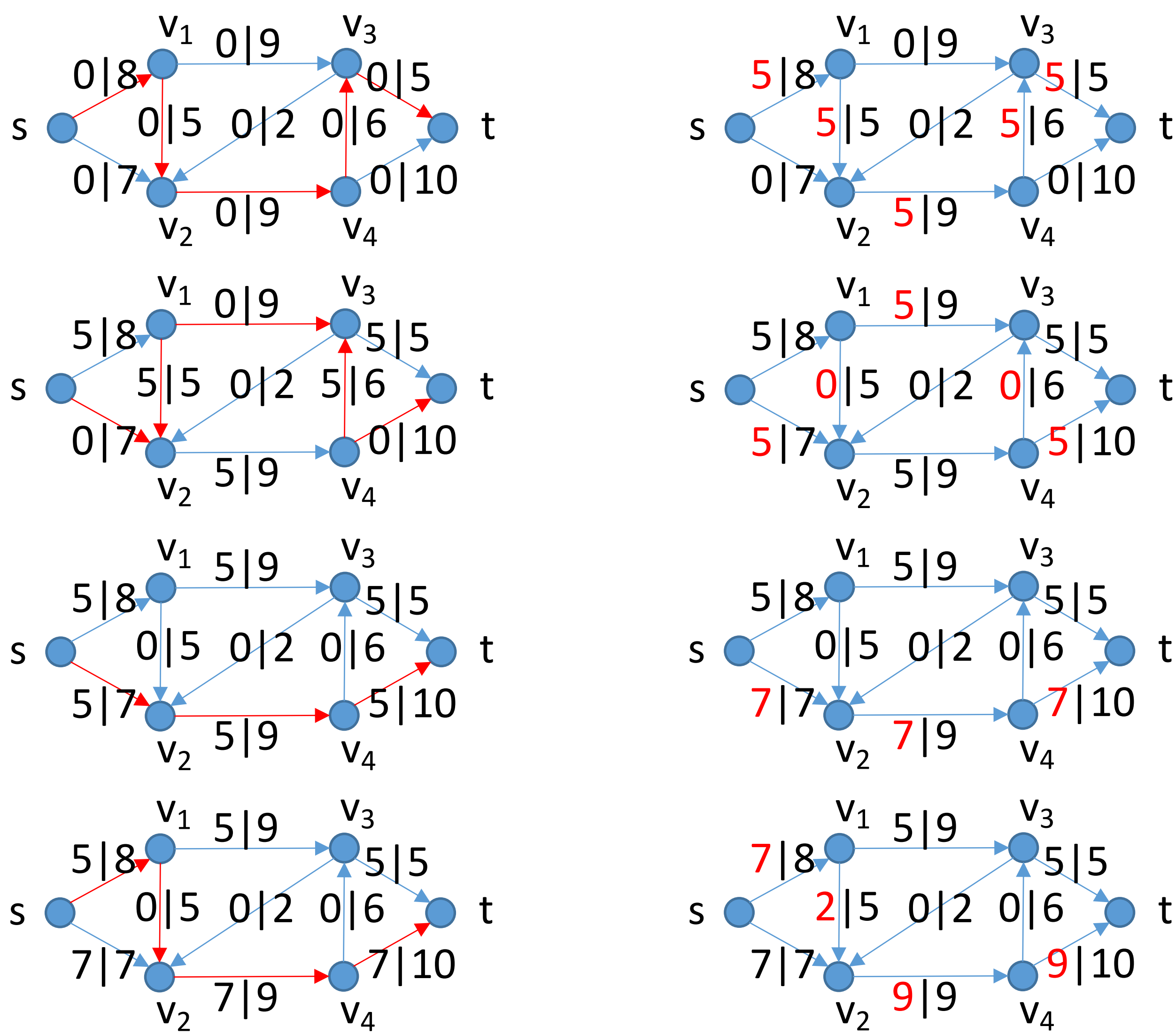
Ford-Fulkerson标号算法 (续)

- 举例



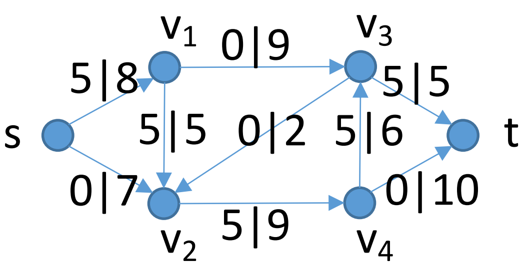
Ford-Fulkerson标号算法 (续)

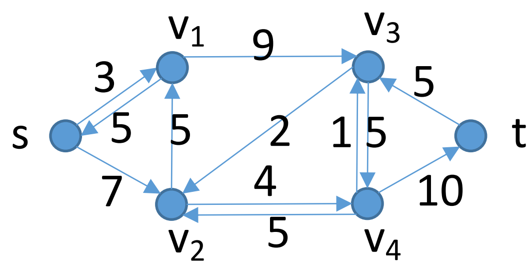
- 举例

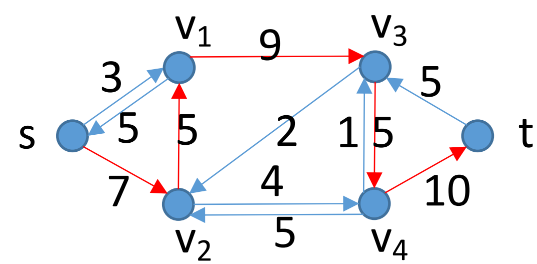


Ford-Fulkerson标号算法 (续)

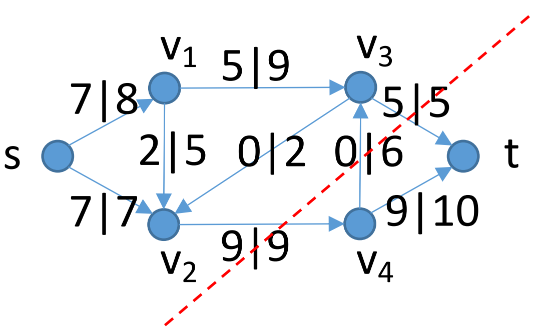
- 如何搜索到一条增广路？
  - 构建余量网络(residual network)

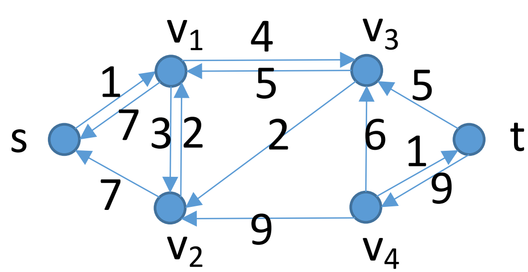


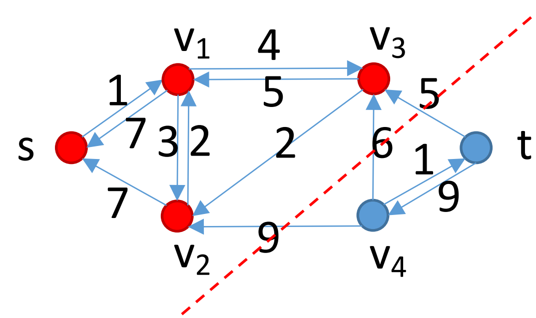




  - 找不到增广路时，余量网络呈现什么特征？

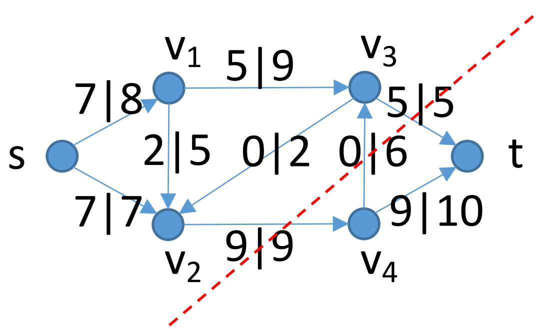






最大流与最小割

- 将顶点集任意划分为S和T，使s∈S, t∈T=V(G)\S
- s-t割(s-t cut)
  - [S,T]={<x,y>∈A(G): x∈S, y∈T}
- 割的容量
  - 弧的容量和：$\sum_{a\in[S,T]}c(a)$
- 最小割(minimum cut)
  - 容量最小的割



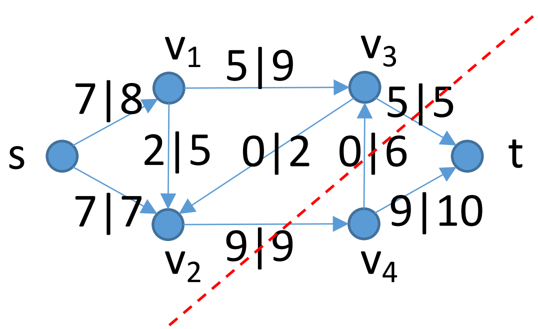
最大流与最小割 (续)

- 引理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量$f^+(s)-f^-(s)=f^+(S)-f^-(S)$，其中$f^+(S)$表示离开S的弧的流和、$f^-(S)$表示进入S的弧的流和。

证明：你能自己证明吗？

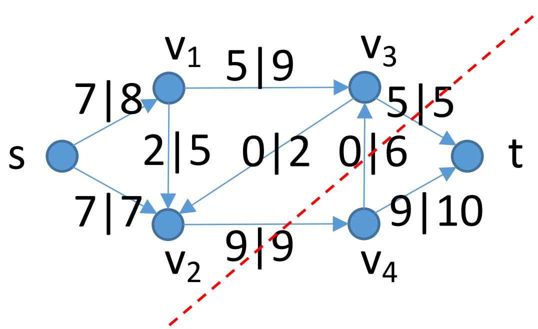
$\forall v \in S\setminus\{s\},\ f^+(v)=f^-(v)$

$\Rightarrow f^+(s)-f^-(s) = f^+(s)-f^-(s)+\sum_{v\in S\setminus\{s\}}(f^+(v)-f^-(v)) = \sum_{v\in S}(f^+(v)-f^-(v)) = f^+(S)-f^-(S)$



最大流与最小割 (续)

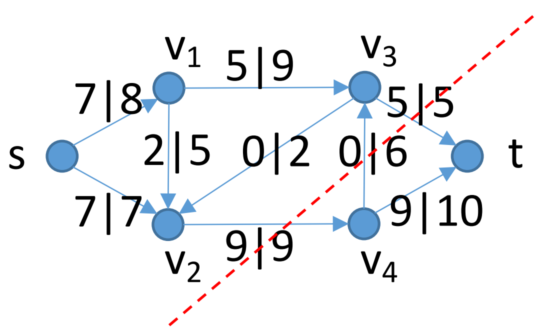
- 引理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量$f^+(s)-f^-(s)=f^+(S)-f^-(S)$，其中$f^+(S)$表示离开S的弧的流和、$f^-(S)$表示进入S的弧的流和。
- 定理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量不超过[S,T]的容量。（什么时候两者相等？）



S到T的弧：f=c
T到S的弧：f=0

最大流与最小割 (续)

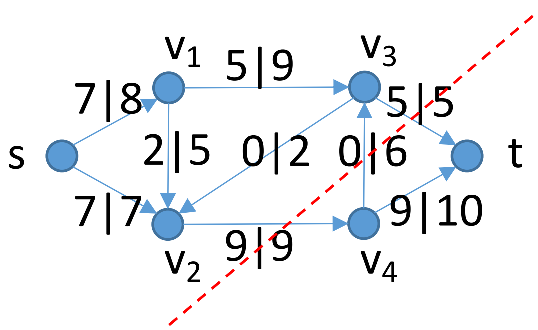
- 引理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量$f^+(s)-f^-(s)=f^+(S)-f^-(S)$，其中$f^+(S)$表示离开S的弧的流和、$f^-(S)$表示进入S的弧的流和。
- 定理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量不超过[S,T]的容量。（什么时候两者相等？）
- 推论9.1.1 设f是网络中的一个可行流、[S,T]是一个割，若f的流量与[S,T]的容量相等，那么f是一个最大流而[S,T]是一个最小割。



S到T的弧：f=c
T到S的弧：f=0

最大流与最小割 (续)

- 引理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量$f^+(s)-f^-(s)=f^+(S)-f^-(S)$，其中$f^+(S)$表示离开S的弧的流和、$f^-(S)$表示进入S的弧的流和。
- 定理9.1.1 对网络中任一可行流f和任一割[S,T]，均有f的流量不超过[S,T]的容量。（什么时候两者相等？）
- 推论9.1.1 设f是网络中的一个可行流、[S,T]是一个割，若f的流量与[S,T]的容量相等，那么f是一个最大流而[S,T]是一个最小割。



S到T的弧：f=c
T到S的弧：f=0

Ford-Fulkerson标号算法运行结束时，
恰是这种状态（为什么？），因此算法是正确的。

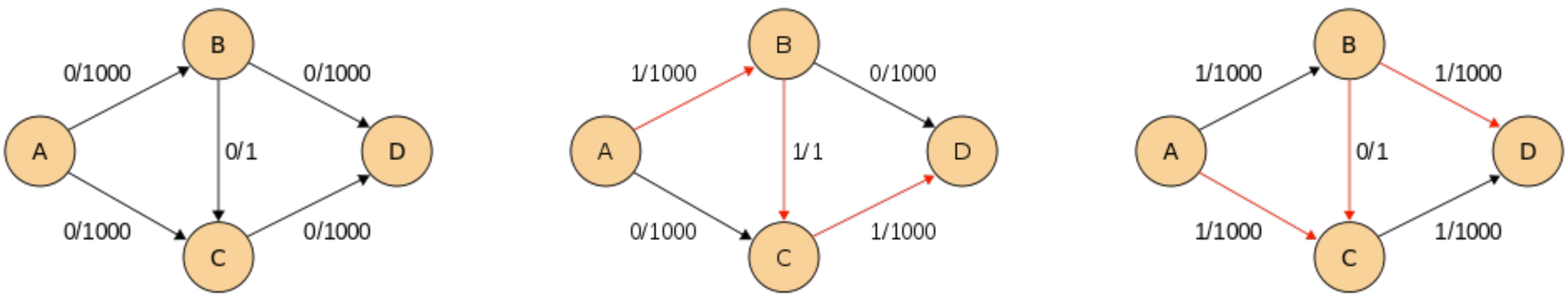
因为在余量网络中，S到T没有弧

最大流与最小割 (续)

- 定理9.1.2 最大流的流量=最小割的容量。
- 推论9.1.2 若所有弧的容量都是整数，则最大流的流量也必为整数。

Ford-Fulkerson标号算法 (续)

- 算法的运行时间
  - 迭代的轮数？
    - $O(f_{max})$，f_{max}为最大流的流量
  - 每轮迭代的时间：O(E)

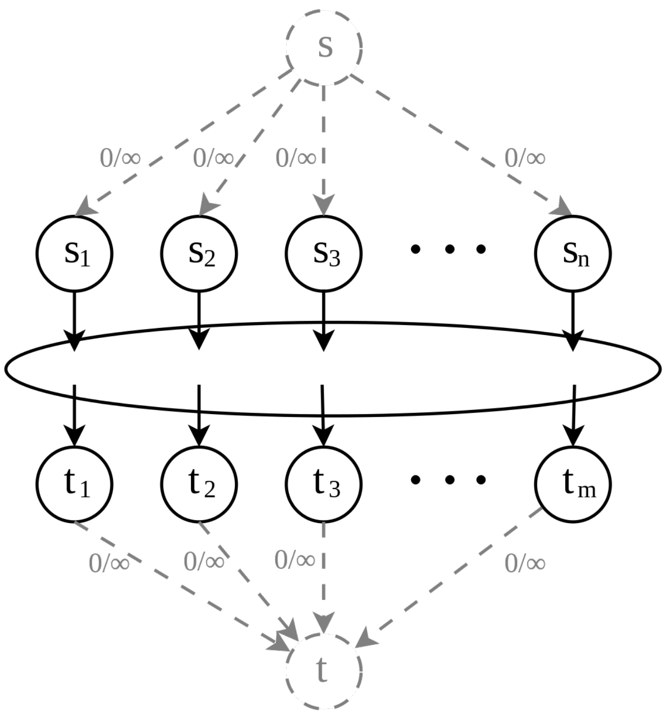


甚至，当容量包含无理数时，算法有可能永不终止。

https://upload.wikimedia.org/wikipedia/commons/thumb/f/f1/Ford-Fulkerson_example_0.svg/300px-Ford-Fulkerson_example_0.svg.png

最大流的应用与扩展

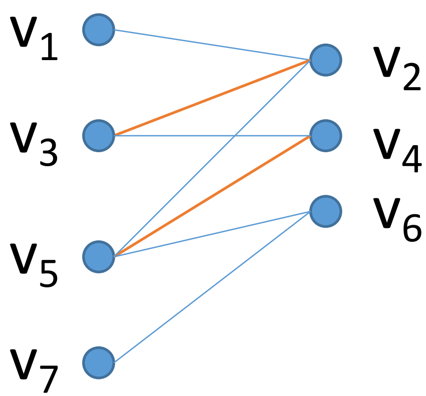
- 多源多汇网络



https://en.wikipedia.org/wiki/File:Multi-source_multi-sink_flow_problem.svg

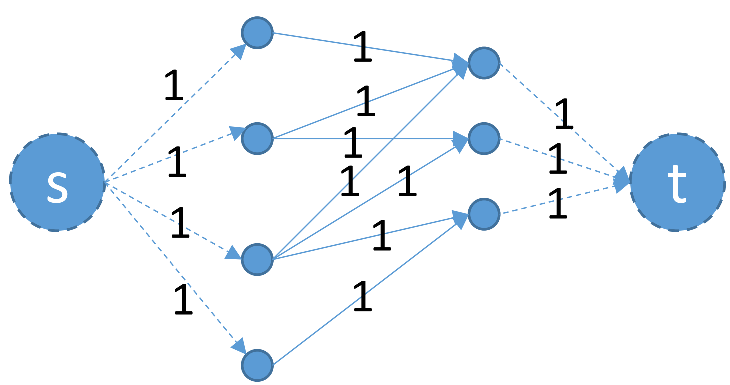
最大流的应用与扩展 (续)

- 还记得二部图中求最大匹配的增广路算法吗？与Ford-Fulkerson标号算法是不是有些神似？
- 你能将该问题转化为最大流问题吗？



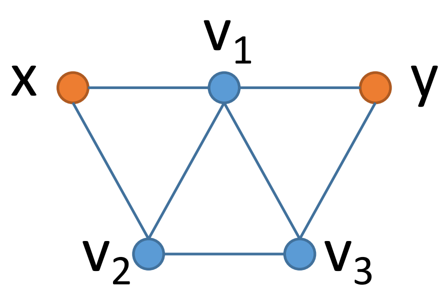
最大流的应用与扩展 (续)

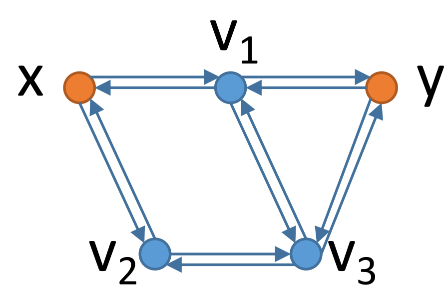
- 还记得二部图中求最大匹配的增广路算法吗？与Ford-Fulkerson标号算法是不是有些神似？
- 你能将该问题转化为最大流问题吗？



最大流的应用与扩展 (续)

- 定理2.4.2 (边形式Menger定理) 设x,y是图G中两个不相邻顶点，则G中分离x,y所需的最少边数s’(x,y)等于G中两两无公共边的x-y路的最大条数r’(x,y)。
- 如果能求出s’(x,y)，就能求出边连通度κ’。（怎么做？）
- 如何计算s’(x,y)？你能将该问题转化为最大流问题吗？



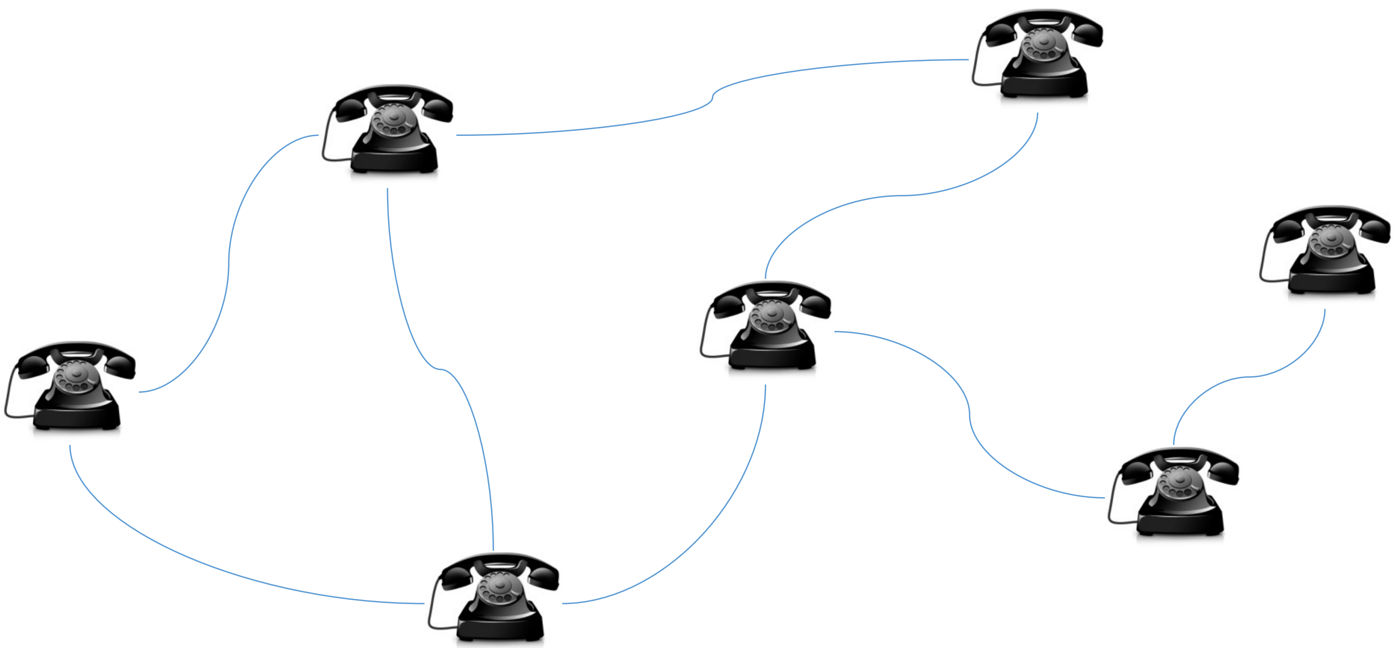


讨论课：图论的应用

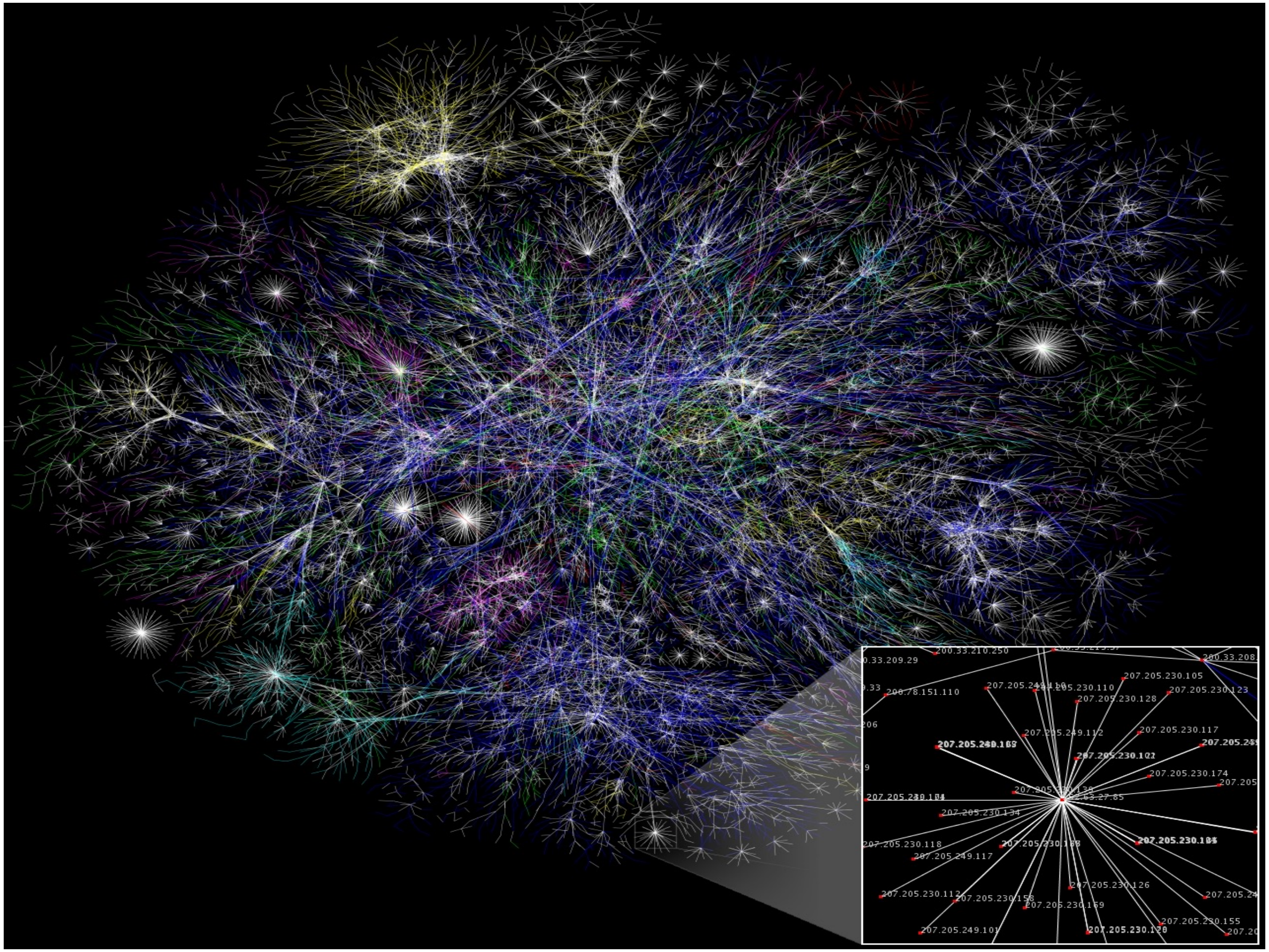
- 本科生：从指定的若干开放性问题中任选一个
  - 利用图论知识，对问题进行建模并给出解决方案
    1. 建模（细化问题/作出假设、实际问题 → 图论问题）
    2. 求解（关键步骤、核心算法……）
    3. 讨论（优缺点、扩展延伸……）
  - 制作一个约10页的PPT，发到**xxwang@smail.nju.edu.cn**
    - 文件命名：题号-学号-姓名
  - 截止时间：**6月2日18:00**
- 研究生：结合本人研究方向，自拟问题，要求同上

- 6月3日讨论课请部分同学讲解PPT（5-10分钟）

1. How will you lay cable to connect telephones?



2. Is your network vulnerable to attacks?



3. How will you fill jobs with people having different specialties?

4. How will you distribute images (or textures) on a game disk?

5. At which crossroads will you install 360-degree cameras to monitor the traffic?

6. Which accounts will you hack to access the posts in an online social network?

7. How will you make a timetable?

课程 星期 节次	一	二	三	四	五
1--2节	操作系统 (一) 仙 1-115 (二) 仙 1-104 / 计算机组织与系统结构 逸 A-212	体育			模式识别 逸 B-101
3-4节	数理逻辑 仙 1-107 / 数字逻辑电路实验 基础实验楼乙 125	数据通信 (一) 仙 1-115 (二) 仙 1-103 / 数据结构 逸 A212 / 随机算法 仙 1-101 / 软件产业概论 逸 B101	数字信号处理 逸 B-212 / 计算机组织与系统结构 逸 A-212	操作系统 (一) 仙1-115 (二) 仙1-103 / 数据结构 逸A212	图论 逸 B101
5--6节	计算机组成与系统结构实验（一） 基础实验楼乙 125	计算方法 (一) 仙 1-115 / (二) 仙 2-212	计算机组成与系统结构实验（二） 基础实验楼乙 125	数据通信（双） (一) 仙 1-115 (二) 仙 1-103 / 数理逻辑（单）仙 1-107	微机原理与接口技术 逸 B-101
7--8节	博弈论及其应用 逸 A117				

8. How will you assign seats to improve communication between classes?

9. How will you rank teams based on their one-round game results?

- 最后三次课的安排
  - 6月3日：讨论课（交回发言卡片并写上姓名学号）
  - 6月10日：习题讲解
  - 6月17日：期末考试