Set $wgLogo to the URL path to your own logo image.

为防止广告，目前nocow只有登录用户能够创建新页面。如要创建页面请先登录/注册（新用户需要等待1个小时才能正常使用该功能）。

导航

- 首页
- 社区主页
- 当前事件
- 最近更改
- 随机页面
- 使用帮助
- NOCOW地图
- 新手试练场

搜索

进入 搜索

工具箱

- 链入页面
- 链出更改
- 特殊页面
- 可打印版
- 永久链接

网络流

(跳转自Ford-Fulkerson算法)

网络流（network flows）是图论中的一种理论与方法，研究网络上的一类最优化问题 。

- 1955年 ，T.E. 哈里斯在研究铁路最大通量时首先提出在一个给定的网络上寻求两点间最大运输量的问题。1956年，L.R. 福特和 D.R. 富尔克森等人给出了解决这类问题的算法，从而建立了网络流理论。所谓网络或容量网络指的是一个连通的赋权有向图 D＝ （V、E、C） ， 其中V 是该图的顶点集，E是有向边(即弧)集，C是弧上的容量。此外顶点集中包括一个起点和一个终点。网络上的流就是由起点流向终点的可行流，这是定义在网络上的非负函数，它一方面受到容量的限制，另一方面除去起点和终点以外，在所有中途点要求保持流入量和流出量是平衡的。
- 最大流理论是由福特和富尔克森于 1956 年创立的 ，他们指出最大流的流值等于最小割(截集)的容量这个重要的事实，并根据这一原理设计了用标号法求最大流的方法，后来又有人加以改进，使得求解最大流的方法更加丰富和完善 。最大流问题的研究密切了图论和运筹学，特别是与线性规划的联系，开辟了图论应用的新途径。
- 最大流问题仅注意网络流的流通能力，没有考虑流通的费用。实际上费用因素是很重要的。例如在交通运输问题中，往往要求在完成运输任务的前提下，寻求一个使总运输费用最省的运输方案，这就是最小费用流问题。如果只考虑单位货物的运输费用，那么这个问题就变成最短路问题。由此可见，最短路问题是最小费用流问题的基础。现已有一系列求最短路的成功方法。最小费用流(或最小费用最大流)问题，可以交替使用求解最大流和最短路两种方法，通过迭代得到解决。
- 目前网络流的理论和应用在不断发展，出现了具有增益的流、多终端流、多商品流以及网络流的分解与合成等新课题。网络流的应用已遍及通讯、运输、电力、工程规划、任务分派、设备更新以及计算机辅助设计等众多领域。

目录 [隐藏]

使用条件&范围 [编辑]

算法描述 [编辑]

最大流问题 最大流问题就是求总流量最大的可行流. 它是一个特殊的线性规划问题. 但是利用图的特点，解决这个问题较之线性规划的 一般解法要方便、快捷、直观得多.

相关定义

定义1. 设有一个有向连通图G(V, A), 在V中指定一点称为发点s, 和另一点称为收点t, 其余的称为中间点. 弧(arc)集A中每条弧 (i,j) 上有非负数cij 称为这弧的容量, 记容量集为c= {cij}, 称这样的图为一个网络. 记为G=(V,A,c). (注: 当(i,j)不是弧时 cij=0.)

定义2.在弧集A上的弧(i,j) 定义一非负数fij，称为弧(i,j)上的流量，流量的集合f={fij}称为网络的一个流.

定义3. 满足下列条件的流称为可行流:
1. 容量限制条件, 所有的弧的流量fij不大于弧的容量cij.
2. 平衡条件，对所有的中间点，流入的流量和等于流出的流量和; 发点流出的总流量F等于流进收点的总流量F.

定义4. 当一弧的流量等于弧的容量时，称该弧为饱和弧，当一弧的流量小于弧的容量时，称该弧为不饱和弧. 流量等于零的弧称为零流弧，流量大于零的弧称为非零流弧.

定义5. 若W是一条从发点到收点的有向道(walk)，规定从发点到收点的方向为道的方向. 道上与W的方向相同的弧叫前向弧，道W上的前向弧的全体记为A+, 道上与W的方向相反的叫后向弧. 后向弧的全体记为A-.

定义6. 设f 是一个可行流，W是发点到收点的一条有向道，如果W满足下列条件，称之为关于可行流f的一条可增广道(又称可扩道):
1. 每条前向弧是非饱和弧,
2. 每条后向弧是非零流弧.

可增广道的实际意义是: 沿着这条道可以调整道上弧的流量，使得道上的每条前向弧增加一流量d，每条后向弧减少同一流量d，使得所得的流仍是可行流，但使总流量F增加d.

定理 1. 可行流f*是最大流当且仅当不存在关于f*的可增广道.

定义7: 设容量网络G=(V,A,c)的顶点集V是两个不交的部分S, S′的併集,使得发点在S中，收点在S′中. 若A′是A的最小的子集，使得G中去掉A′后成为两个不相交的子图G1(S,A1), G2(S′,A2),分别以S, S′为顶点集，则称A′是关于(S,S′)的割集，记为A′=(S,S′). 割集A′中所有始点在S，终点在S′的弧的容量之和称为割集(S,S′)的割集容量，记为C(S,S′). 对于不同的S和S′就有不同的割集，其中割集容量最小的割集称为容量网络G的最小割集(简称最小割).

定理2. 容量网络的最大流不大于任一割集的割集容量.

(最大流--最小割)定理3. 容量网络的最大流等于最小割的割集容量.

求最大流的标号法(Ford-Fulkerson算法): 设已有一个可行流(如零流)

(1) 给发点标号(0,+, d(s)), d(s)=Inf, 这时发点是已标号但未检查的点，其余的点是未标号的点.
(2) 若被标号的点都已检查过，这时的可行流就是最大流. 最小割(S*,S*′)中的S*就是已标号(且已检查)的点集. 最大流的总流量等于从源出发的所有流量的和. 退出.
(3) 任取一个被标号但未检查的点i, 找所有与点i邻接但未标号的点j，使满足以下两条件之一:
(a)若弧(i,j)上, fij<cij, 则给j点标号(i, +, d(j)), 其中d(j)=min(d(i), cij-fij),
(b)若弧(j,i)上, fji>0, 则给j点标号(i, -, d(j)), 其中d(j)=min(d(i), fji),
(c) 若收点n没有被标号，对i点的标记中+, - 号加圈，表示i点已检查。转(2). 否则令j=n. 进入以下调整过程
(4) 若j点的标记为正, 即(i,+ ,d(j)), 令fij=fij+d(n), 否则fji=fji-d(n)
(5) 若j=1, 去掉全部标记, 转(1), 否则转(4).

时间复杂度 [编辑]

(V代表点,E代表边)

Ford-Fulkerson:O(VE^2);

Preflow-Push:O(V^2E);

Relable-to-front:O(V^3);

[编辑]

改进和优化

dinic [编辑]

Relable_to_front [编辑]

SAP [编辑]

实现 [编辑]

Edmonds-Karp 算法 [编辑]

建立在Ford-Fulkerson方法上的增广路算法,与一般的Ford-Fulkerson算法不同的是,它用广度搜索实现对增广路的寻找. 代码如下:

```
//Buffalo_NGJ
int n;
long int c[128][128];
long maxflow(int s, int t)
{
    int p, q, queue[128], u, v, pre[128];
    long int flow, aug;
    flow = 0;
    while(true)
    {
        memset(pre, -1, sizeof(pre));
        for(queue[p = q = 0] = s; p <= q; p++)
        {
            u = queue[p];
            for(v = 0; (v < n) && (pre[t]) < 0; v++)
            {
                if((c[u][v] > 0) && (pre[v] < 0))
                {
                    pre[v] = u;
                    queue[++q] = v;
                }
            }
            if(pre[t] >= 0)
            {
                break;
            }
        }
        if(pre[t] < 0)
        {
            break;
        }
        aug = 0x7fff;
        for(u = pre[v = t]; v != s; (v = u), (u=pre[u]))
        {
            if(c[u][v] < aug)
            {
                aug = c[u][v];
            }
        }
        for(u = pre[v = t]; v != s; (v = u), (u = pre[u]))
        {
            c[u][v] -= aug;
            c[v][u] += aug;
        }
        flow += aug;
    }
    return flow;
}
```

练习题目 [编辑]

1. NOI2006 Day2 profit
2. Translate:USACO/ditch
3. ruvtex.cn - CmYkRgB123 Online Grading System - 11 运输问题1

引用&参考 [编辑]

链接 [编辑]

图论及图论算法 [编辑]

图 - 有向图 - 无向图 - 连通图 - 强连通图 - 完全图 - 稀疏图 - 零图 - 树 - 网络

基本遍历算法：宽度优先搜索 - 深度优先搜索 - A* - 并查集求连通分支 - Flood Fill

最短路：Dijkstra - Bellman-Ford（SPFA） - Floyd-Warshall - Johnson算法

最小生成树：Prim - Kruskal

强连通分支：Kosaraju - Gabow - Tarjan

网络流：增广路法（Ford-Fulkerson，Edmonds-Karp，Dinic） - 预流推进 - Relabel-to-front

图匹配 - 二分图匹配：匈牙利算法 - Kuhn-Munkres - Edmonds' Blossom-Contraction

1个分类：图论

此页面已被浏览过22,876次。 本页面由cosechy@gmail.com于2012年3月3日 (星期六) 05:00做出最后修改。 在绝恋love枫和王轲、NOCOW用户Lanctina、NOCOW匿名用户124.93.203.49和其他的工作基础上。 本站全部文字内容使用GNU Free Documentation License 1.2授权。 隐私权政策 关于NOCOW 免责声明 陕ICP备09005692号