二分图与网络流模型设计

GOOGLE 李煜东 2019/1/25

Review: 网络流基本定义

- 有向图G(V,E)，每条边(u,v)∈E有一个非负实数的容量c(u,v)
- 源点S和汇点T
- 实际流量f(u,v)，剩余容量c(u,v)-f(u,v)，净流f(u,v)-f(v,u)
- 容量限制：f(u,v)<=c(u,v)
- 斜对称：u到v的净流是v到u的净流的相反数
- 流守恒：除了源点和汇点外，其余各点流入和流出的流量相等
- 所有点和剩余容量大于0的边构成的图被称为残量网络。
- 残量网络中从源点S到汇点T的路径被称为增广路。

Review: 网络流基础算法

- Edmonds-Karp增广路算法
  - 求解最大流的基础算法
  - 求解最小费用最大流/最大费用最大流的常用算法
- Dinic算法 / SAP算法
  - 求解最大流的高效算法
- NOI中一般不需要应用其它算法

Review: 二分图

- 二分图
  - 无奇环无向图（点可以分成左右两部，每一部内没有边）
  - 判定：DFS 0/1染色
- 最大匹配
  - 增广路算法（匈牙利算法）、最大流
- 经典模型
  - 最小点覆盖、最大独立集、最小路径点覆盖
- 带权匹配
  - KM算法、最小/最大费用最大流

REVIEW: 二分图常见问题

- Points
  - 其一是如何选择点、边、部，其二是映射为何种经典模型，其三是建图和算法
- 矩形网格中的各类问题
  - 棋盘覆盖、放置車、放置马、带障碍放置(ZOJ1654)、带障碍覆盖(POJ2226)
- 多重匹配与拆点
  - 直接进行多重匹配：匈牙利算法的两种修改写法 or 直接用网络流
  - 拆点再做

例题

- HEOI2012 朋友圈
- http://www.lydsy.com/JudgeOnline/problem.php?id=2744

例题

- HEOI2012 朋友圈
- http://www.lydsy.com/JudgeOnline/problem.php?id=2744
- 观察原图：A国中友善度奇偶性不同的点之间有边，B国中所有奇数之间有边、所有偶数之间有边、奇偶不同的数之间一部分有边。
- 观察朋友圈的定义：朋友圈是一个团。
- 一般图的最大团问题是NPC的，然而这个图的补图很特殊。

例题

- 观察补图：A国的奇数值点构成完全图，偶数值点构成完全图；
- B国的奇数点和偶数点构成二分图。
- 补图上朋友圈的定义：朋友圈是一个独立集（最大团=补图最大独立集）
- A国最多取两个点，可以枚举这两个点x,y；
- 数据给定了A和B国之间有边的情况，删除与x,y之间无边的B国点；
- 然后在B国剩余的点上求二分图的最大独立集。

二分图的可行边与必须边

- 如果存在一组完备匹配？
  - 对于匹配边(u,v)，v到u连边；非匹配边u到v连边；
  - 求强连通分支，若(u,v)是匹配边或者u,v在同一个分支中——可行边；
  - 若(u,v)是匹配边且u,v不在同一个分支中——必须边。

二分图的可行边与必须边

- 如果不一定存在完备匹配？
  - 先用Dinic求出任意一组最大匹配。建一张新图：
  - 对于匹配边(u,v)，v到u连边；非匹配边u到v连边；
  - 对于匹配的左部点u，u到S连边；未匹配的左部点u，S到u连边；
  - 对于匹配的右部点v，T到v连边；未匹配的右部点v，v到T连边。
  - 求强连通分支，若(u,v)是匹配边或者u,v在同一个分支中——可行边；
  - 若(u,v)是匹配边且u,v不在同一个分支中——必须边。

最小割

- 删去之后使网络中源点S到汇点T不存在路径的边的集合称为网络的割。
- 最小割，就是使这个边集中所有边的容量之和最小。
- 最大流最小割定理：任何一个网络的最大流量等于该网络的最小割的容量。
- 如果最小割<最大流，那么根据最大流算法，割去这些边之后，仍然可以找到一条从S到T的增广路。所以最小割不小于最大流。因此如果我们能给出一个等于最大流的割集构造方案，就可以证明最小割=最大流。
- 求出最大流后，从源点开始沿残量网络BFS，标记能够到达的点。连接已标记的点和未标记的点的正向边就是该网络的一个最小割集。

最小割的可行边与必须边

- 最小割的必须边
  - 一定在最小割中的边、扩大容量后能增大最大流的边，Poj3204：
  - ① 满流；② 残余网络中S能到入点、出点能到T。
  - 从S开始DFS、T开始反向DFS，标记到达的点，然后枚举满流边即可。
- 最小割的可行边
  - 被某一种最小割的方案包含的边，AHOI2009：
  - ① 满流；② 删掉之后在残余网络中找不到u到v的路径。
  - 在残余网络中tarjan求SCC，(u,v)两点在同一SCC中说明残余网络中存在u到v路径。

网络流基本技巧

- 点边转化
  - 一个点拆成两个，中间加一条边，把点的各种信息反映在这条边上
  - 一条边截成两半，中间插入一个点，把边的各种信息反映在这个点上
  - 求无向图点/边连通度（Poj1966、Poj1815、Poj2914）
  - K取方格数（POJ3422）
- $+\infty$ 容量边
  - 防割
  - 流量传递（Poj1149 Pigs）

动态加点

- NOI2012 美食节
  http://www.lydsy.com/JudgeOnline/problem.php?id=2879

动态加点

- NOI2012 美食节 http://www.lydsy.com/JudgeOnline/problem.php?id=2879
- 源点向每道菜连边，容量为 p[i]，费用为0。
- 每个厨师拆成n个点，向汇点连边，容量为1，费用为0。
- 第i到菜向第j个厨师拆成的第k个点连边，容量1，费用k*a[i][j]。
- 求最小费用最大流（超时）。
- 使用动态加点法：起初每个厨师只拆成一个点，每次无法增广时，把满流的厨师拆出一个新点。

混合图欧拉回路 - Poj1637

- 一张图中既有无向边，也有有向边，求经过每条边恰好一次的回路。

混合图欧拉回路 - Poj1637

- 一张图中既有无向边，也有有向边，求经过每条边恰好一次的回路。
- 把图中的无向边任意定向。
- 如果有某个点的出入度之差为奇数，则不存在欧拉回路。
- 设K[i] = | inDeg[i] - outDeg[i] | / 2
- 删除有向边，无向边容量设为1，新建源汇。
- 对于入度大于出度的点u，连边(u, t)、容量K[u]。
- 对于入度小于出度的点v，连边(s, v)、容量K[v]。
- 计算最大流，若满流则有解。

混合图欧拉回路 - Poj1637

- 总入度=总出度，即源点连出的边容量和等于汇点连入的边容量和，源点和汇点一定同时满流。
- 每个入度 > 出度的点v（与T相连），都有K[v]条边流出去到T；
- 对于出度 > 入度的点u（与S相连），都有K[u]条边从S流进来。
- 对于入度 = 出度的点（未与S、T相连），流量平衡。
- 将所有流量不为0的边反向，使得出入度相差2*K[i]的点i关联的K[i]条边改变方向，就得到了每个点入度 = 出度的欧拉图。

Poj1895 bring them there

- 题目描述：在公元3141年人类的足迹已经遍布银河系。为了穿越那巨大的距离，人类发明了一种名为超时空轨道的技术。超时空轨道是双向的，连接两个星系，穿越轨道需要一天的时间。然而这个轨道只能同时给一艘飞船使用，也就是说，每条轨道每天只能有一艘飞船穿越。现在IBM公司要把K（K≤50）台超级计算机从地球运到Eisiem星系去，由于这些超级计算机个头巨大，一台计算机就要用一艘飞船来运。现在人类能够到达N（N≤50）个星系，拥有M（M≤200）条超时空轨道，太阳系的编号为S，Eisiem星系的编号为T。你需要求出至少需要几天才能将这些超级计算机全部运到目的地。注意，IBM公司是非常NB的公司，所有的超时空轨道都会优先给IBM公司使用。

Poj1895 bring them there

- 从小到大枚举答案ans，构造(ans+1)层的图，每层都是原图的n个点。
- 源点连第一层的S，容量为k；
- 所有层的T连汇点，容量为inf；
- 每层的点i 都向下一层的点i 连边，容量为inf；
- 对于原图的无向边(u,v)，在相邻两层之间连边(u,v),(v,u)，容量1。
- 求最大流，如果最大流等于k说明可以满足题目要求，找到了答案。
- 每次不用清空以前的图，直接加一层继续增广即可。这样的时间复杂度基本上相当于只在最终的(ans+1)层的图上求了一个最大流。

Poj1895 bring them there

- 输出方案：从源点出发dfs k次，得到k条路径。每次dfs时寻找一条从当前点出发有流的边走过去，并且把这条边的流减一，直到到达汇点。注意以下几点：
- (1) 如果从当前层的i 走到了下一层的i ，实际上这一天仍在i 星系中没有移动，因此这条边不能记录到方案里。
- (2) 如果相邻两层之间的(u,v)和(v,u)都有流，根据题目要求，一条路上不能有两个不同方向的流同时流过。但是这样的方案可以等效成两个流各自停留在u和v没有移动，没有流过这两条边，而在后边的路程中两个流的路径进行了交换。因此此时的边也不能记录到方案中。

最小路径边覆盖

- 题意：求有向无环图可重叠的最小路径边覆盖（二分图中解决的是点覆盖）。

最小路径边覆盖

- 题意：求有向无环图可重叠的最小路径边覆盖（二分图中解决的是点覆盖）。
- 如果不能重复覆盖，那么记一个点的入度为in[i]，出度为out[i]，那么答案就是$\sum$max(in[i]-out[i],0)。方案直接搜索就行了。
- 重复走一条边，可以看做加入了一条重边。若加入边(u,v)，则out[u]++, in[v]++。添加之前如果in[u]>out[u]，in[v]<out[v]，那么这次加边会使答案变优1。
- 进一步扩展，如果把连续的若干条边加上重边，那么相当于这条路径的起点s，out[s]++，终点t，in[t]++。
- 问题变为：添加一些从入度>出度的点到入度<出度的点的路径，使答案最优。

最小路径边覆盖

- 我们根据以往的经验知道，网络流寻找路径的问题，应当把每个点拆点，然后把路径拆成若干条边、以及拆点之后两个同点之间的边。
- 把每个点拆成左、右两个，右点向左点连容量为+∞的边。
- 从源点S向所有in[i]>out[i]的左点连边，容量为in[i]-out[i]。
- 从所有in[i]<out[i]的右点向汇点T连边，容量为out[i]-in[i]。
- 对于原图中的边(u,v)，从u的左点向v的右点连容量为+∞的边。
- 求最大流，那么答案就是满流减去最大流。
- 一条边上有多少流量，就要添加多少条重边，然后dfs输出方案。

星际竞速 - Bzoj1927

- http://www.lydsy.com/JudgeOnline/problem.php?id=1927

星际竞速 - Bzoj1927

- http://www.lydsy.com/JudgeOnline/problem.php?id=1927
- 每个点只访问一次，就是说回路中每个点的入度出度都是1。这有点类似于匹配，启发我们用二部图来处理。
- 如果能够选出一些边，使得每个点仅包含在一条入边和一条出边里，那么最后把这些边组合一下就可以得到答案。
- 每个点拆成一个入点（右边一排）和一个出点（左边一排）。

星际竞速 - Bzoj1927

- 源点S向出点连容量1费用0的边，入点向汇点连容量1费用0的边。
- 高速航道(x,y)，x<y，从x的出点到y的入点连容量1费用为边权的边。
- 从S向所有入点连容量1费用为定位费用的边。
- 求最小费用最大流。
- 到汇点的边的容量限制、以及最大流保证了每个入点有且仅有一条有流量的入边。这条边要么从某个出点来，要么从源点定位瞬移过来。

平面图最小割 = 对偶图最短路

- 在原平面图中添加一条从起点S到终点T的边，会增加一个区域S’。
- 无限大的区域设为T’。
- 对加边后的平面图的每条边，在对偶图上从顺时针到逆时针方向连边。
- 删去边T’-S’。
- 此时S-T的最小割大小等于S’到T’的最短路长度。
- Bzoj1001 狼抓兔子

海拔 – NOI2010

- NOI2010 海拔
- 分析题目得到如下关键性质：存在一个最优解，仅包含海拔为0或1的点，并且海拔为0、1的点分别构成一个连通块。
- 因此题目实际上就是求西北角与东南角之间的最小割。
- 可以求最大流，也可以在对偶图上求最短路得到答案。

无源汇上下界可行流

- Sgu194 给定N个点，M条边的网络，求一个可行解，使得边(u,v)的流量介于[B(u,v), C(u,v)]之间，并且整个网络满足流量守恒。

无源汇上下界可行流

- Sgu194 给定N个点，M条边的网络，求一个可行解，使得边(u,v)的流量介于[B(u,v), C(u,v)]之间，并且整个网络满足流量守恒。
- 如果把C-B作为容量上界，0作为容量下界，就是一般的网络流模型。
- 然然而求出的实际流量为f(u,v)+B(u,v)，不一定满足流量守恒，需要调整。
- 设inB[u]=∑B(i,u)，outB[u]=∑B(u,i)，d[u]=inB[u]-outB[u]。
- 新建源汇，S向d>0的点连边，d<0的点向汇点连边，容量为相应的d。
- 在该网络上求最大流，则每条边的流量+下界就是原网络的一个可行流。

有源汇上下界可行流

- Poj2396 Budget http://poj.org/problem?id=2396

有源汇上下界可行流

- Poj2396 Budget http://poj.org/problem?id=2396
- 建立源s和汇t，每行每列都看作一个点。源连所有行，上下界均为此行的和，所有列连汇，上下界均为此列的和。
- 对于每个点，可能读入多个限制，取其上界的最小值，下界的最大值，连接对应的行点和列点。
- 从t到s连一条(t,s,0,+∞)的边，把汇流入的流量转移给源流出的流量，转化为无源汇的网络，然后求解 <无源汇上下界可行流> 问题。

有源汇上下界最大流

- 解法一：
- 二分答案ans，从T到S连一条(T,S,ans,+∞)的边，转化为无源汇网络。
- 找到最大的ans，使得该上下界无源汇网络有可行流。
- 解法二：
- 从T到S连一条(T,S,0,+∞)的边，转化为无源汇网络。
- 按照 <无源汇上下界可行流> 的做法求一次超级源到超级汇的最大流。
- 回到原网络，在上一步的残量网络基础上，求一次S到T的最大流。

有源汇上下界最大流

- Zoj3229 Shoot the Bullet 有M个女孩拍N天照片，第i天拍照的总张数不超过Di，并且只能给指定的Ci个人拍照。第i个女孩N天内拍照总数不超过Gi，每天拍照的数量介于[Li, Ri]之间。求M个女孩N天拍照的总数的最大值及方案。

有源汇上下界最大流

- Zoj3229 Shoot the Bullet 有M个女孩拍N天照片，第i天拍照的总张数不超过Di，并且只能给指定的Ci个人拍照。第i个女孩N天内拍照总数不超过Gi，每天拍照的数量介于[Li, Ri]之间。求M个女孩N天拍照的总数的最大值及方案。
- 建立源S、汇T，天数作为二分图左部，女孩作为二部图右部。
- S向第i天连边(S, Day[i], 0, Di)，第i个女孩向T连边(Girl[i], T, 0, Gi)。
- 若第i天可以给第j个女孩拍照，连边(Day[i], Girl[j], Lj, Rj)。

有源汇上下界最小流

- Sgu176 Flow Construction
- 给定源(点1)汇(点n)和管道(边)，管道有容量限制，并且其中一些管道必须满流（其它的不必），求满足要求的最小流。
- 解法一：
- 二分答案ans，从T到S连一条(T,S,0,ans)的边，转化为无源汇网络。
- 找到最小的ans，使得该上下界无源汇网络有可行流。

有源汇上下界最小流

- 解法二：
- 类似 <有源汇上下界可行流> 的构图方法，但是不添加T到S的边，求一次超级源到超级汇的最大流。
- 加边(T,S,0,+∞)，在上一步残量网络基础上再求一次超级源到超级汇的最大流。
- 流经T到S的边的流量就是最小流的值。
- 该算法的思想是在第一步中尽可能填充循环流，以减小最小流的代价。

有源汇上下界最小费用可行流

- 解法类似于<有源汇上下界可行流>，求最大流改为求最小费用最大流。
- Bzoj2055 80人环游世界
- 每个国家拆成两个点（入点i 和出点 i'），建立源S汇T附加源S'。
- 连边(S, S', 0, m, 0), (S', i, 0, +∞, 0), (i', T, 0, +∞, 0), (i, i', V[i], V[i], 0)
- 若i,j两个国家通航，连边(i', j, 0, +∞, Cost[i,j])
- 对网络S-T求<有源汇上下界最小费用可行流>

例题

- NOI2008 志愿者招募
- http://www.lydsy.com/JudgeOnline/problem.php?id=1061
- 有一个持续N天的项目，第i天需要Ai个志愿者。
- 可供招募的志愿者有M类，第i类可以从第Si天工作到第Ti天，费用是每人Ci元。
- 求招募足够的志愿者所需的最少花费。

例题

- 把每一天看做一个点，第i天到第i+1天连边(i, i+1, Ai, +∞, 0)
- 对于第i类志愿者，从Ti+1到Si连边(Ti+1, Si, 0, +∞, Ci)
- 在这个无源汇网络中，招募1个第i类志愿者，就产生一个Ti+1→Si→Si+1→Si+2→……→Ti→Ti+1的圈，花费Ci，使Si~Ti天的流+1
- 这与题目实际意义是对应的，求<无源汇上下界最小费用可行流>即可

最小割问题的二元关系新解

PART 2

引入

- POJ3469
- 有N个任务，每个任务可以在机器A或机器B上完成。
- 在机器A上完成花费Ai，在机器B上完成花费Bi。
- 有M对关系，每对关系是一个二元组(x,y)。
- 如果第x个任务和第y个任务在不同的机器上完成，额外花费C。
- 求一种安排任务的方式，使得总花费最小。

引入

- 解法：最小割。
- 源点S向每个任务连边，容量Ai。
- 每个任务向汇点T连边，容量Bi。
- 对于二元组关系(x,y)，在x,y之间连双向边，容量C。
- 最小割就是最优的方案。
- 每个任务与S割开表示在A上完成，与T割开表示在B上完成。

引入

- 对于每个二元组(x,y)：
- 若同时在A完成，割开S与x,y之间的边，花费Ax+Ay。
- 若同时在B完成，割开x,y与T之间的边，花费Bx+By。
- 若x在A完成，y在B完成，还需要割开x,y之间的边，花费Ax+By+C。
- 若x在B完成，y在A完成，也需要割开x,y之间的边，花费Bx+Ay+C。
- 可以看出，每一个割集恰好对应一种方案。

问题一般化

- N个任务，M对二元组关系(x,y)，求最小总花费，条件如下：
- 每个任务可以在机器A或机器B上完成，分别花费Ai、Bi。
- 若x,y同时在A上完成，花费C1(x,y)。
- 若x,y同时在B上完成，花费C2(x,y)。
- 若x在A、y在B上完成，花费C3(x,y)。
- 若x在B，y在A上完成，花费C4(x,y)。

模型构建

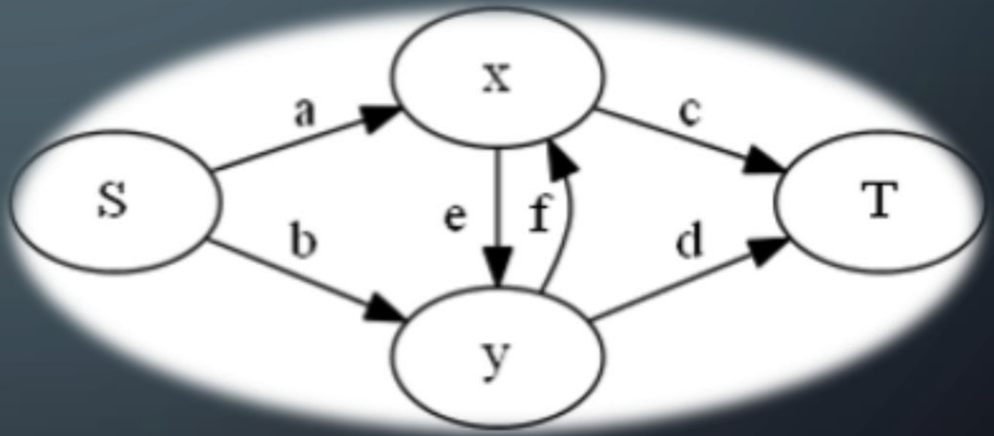
- 每个任务自身的花费Ai、Bi暂不考虑，最后再加入割边。
- 边的容量具有可加性，可以分开考虑每个二元组关系，最后再合并。
- 对于每个二元组关系(x,y)，加入以下边：
- $S \xrightarrow{a} x,\ S \xrightarrow{b} \mathrm{y},\ x \xrightarrow{c} T,\ y \xrightarrow{d} T,\ x \xrightarrow{e} y,\ y \xrightarrow{f} x$
- 割开与S、T相连的边分别表示在A、B完成。
- 给容量a,b,c,d,e,f 赋适当的值，使其意义与问题中的花费吻合。

模型构建

- 由题意得，它们需要满足：

$$\begin{cases} a+b=C1 \\ c+d=C2 \\ a+d+f=C3 \\ b+c+e=C4 \end{cases}$$

- 不会存在类似$a+b+d$的方程，它显然不如$a+b$更优。



性质探究

- 思考以下问题：
- a,b,c,d,e,f 是网络流中边的容量，因此该方程的解一定要求a,b,c,d,e,f 均≥0吗？
- 观察方程发现，a,c不会同时出现在最小割集中，b,d也不会同时出现在最小割集中。
- 如果a,c<0，可以令其同时加一个值，求出最小割后再减去。
- 因此只需要e,f ≥0，而a,b,c,d 没有限制。

性质探究

- 6个未知数，4个方程，多解。
- 方程能否再简化?
- 它的解有意义（使用该模型建图有意义）的条件是什么?
- 如何求出一组合适的解?
- 有何不足?
- 后两个方程相加再减去前两个方程，得到 e+f=C3+C4-C1-C2.
- 令K=C3+C4-C1-C2.
- 由于方程多解，不妨令e=f=K/2.
- 方程的解有意义的条件是K≥0.
- 给a任意一个方便的值，解出b,c,d.

模型改进

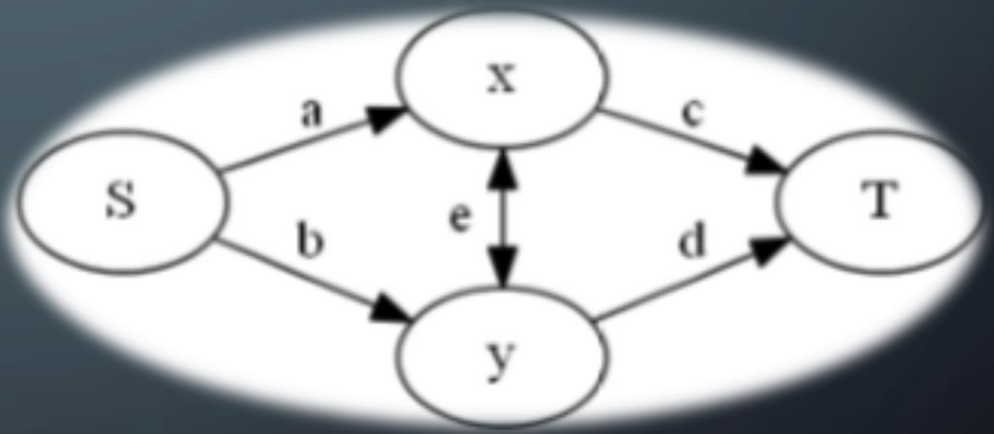
- 当K≥0时，上面的模型已经可以解决问题了。
- 当K<0时，把割集的含义做一些改变（对x的含义取反，y的不变）：
- 割开S与x相连的边，表示x在机器B上完成。
- 割开x与T相连的边，表示x在机器A上完成。
- 割开S与y相连的边，表示y在机器A上完成。
- 割开y与T相连的边，表示y在机器B上完成。

模型改进

- 按照同样的方式列出方程：

$$\begin{cases} a+b=C4 \\ c+d=C3 \\ a+d+e=C2 \\ b+c+e=C1 \end{cases}$$

- 这时 2e=C1+C2-C4-C3=-K>0.



归纳总结

- 对于此类模型，首先计算每个二元组的K值：$K=C3+C4-C1-C2$.
- 如果$K\geq 0$，令$e=f=K/2$，a为任意值，解出b,c,d，a,b,c,d<0时加以调整.
- 如果$K<0$，按照上述方式改变方程组，此时$K'=C1+C2-C4-C3=-K$.
- 不过在$K<0$的建图方式中，x,y与S,T连边的含义是相反的，这就要求原问题中使得$K<0$的(x,y)不构成奇环（$K<0$的二元关系是二分图）。
- 最后把各个任务本身的花费加到对应边的容量上，求出最小割。

例题

- 2010年国家集训队测试题，happiness(吴确)
- 某班的座位表是n*m的矩阵，每个同学和前后左右相邻的同学是好朋友。
- 现在要文理分科了，每个同学对于选择文科和理科有自己的喜悦值。
- 一对好朋友如果同时选择文科or理科，又可以收获一些喜悦值。
- 问如何分配文理科可以使得全班的喜悦值总和最大?
- （喜悦值是>0的整数）

例题

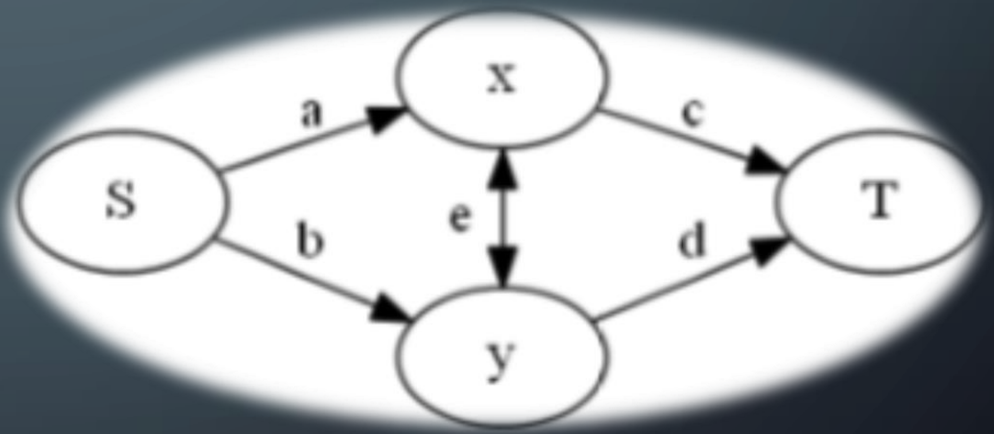
- 题目要求最大，可以把给定的喜悦值取相反数，使问题变为总和最小。
- 每个同学自身对文理科的喜悦值最后加上，先考虑二元关系：
- 一对好朋友同时选择文科，获得 -v1的喜悦值。
- 一对好朋友同时选择理科，获得 –v2的喜悦值。
- 尝试最小割模型，割开与S相连的边表示选文科，割开与T相连的边表示选理科。

例题

- 设未知数，列方程：

$$\begin{cases} a+b=-v1 \\ c+d=-v2 \\ a+d+e=0 \\ b+c+e=0 \end{cases}$$

- K=0+0-(-v1)-(-v2) > 0. 满足模型要求。



例题

- 解方程：e=K/2=(v1+v2)/2.
- 令a= -v1/2，解出b= -v1/2, c= -v2/2, d= -v2/2.
- a,b,c,d<0，令它们同时加上(v1+v2)/2，得到a=b=v2/2, c=d=v1/2.
- 把每个同学本身选择文、理科的喜悦值分别加入它与S、T之间的边。
- 为了避免分数，令所有边的容量乘2，求最小割。
- 求出的结果除以2，再减去M*(v1+v2)，M为二元关系个数。

最大权闭合子图

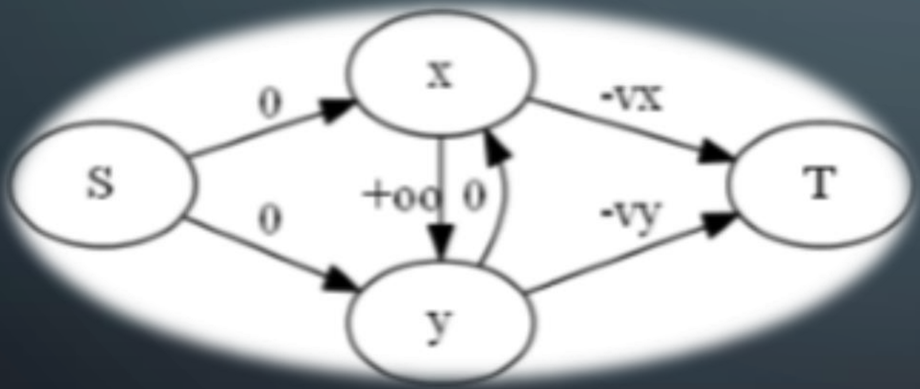
- 若有向图G的子图V满足：V中顶点的所有出边均指向V内部的顶点，则称V是G的一个闭合子图。
- 若G中的点有点权，则点权和最大的闭合子图称为有向图G的最大权闭合子图。

最大权闭合子图

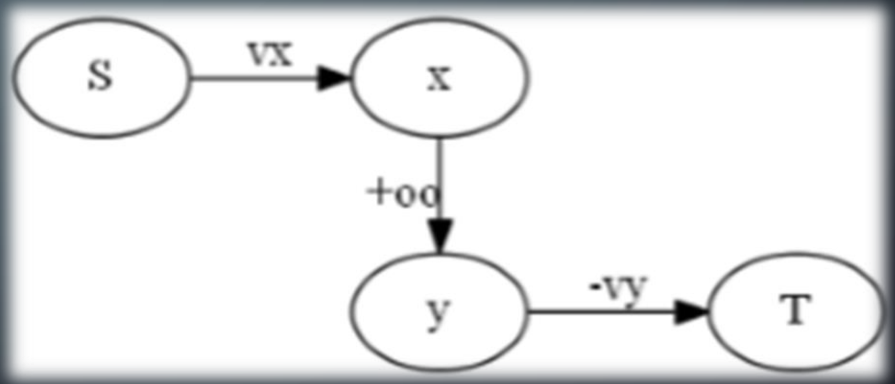
- 尝试对该问题应用前面提到的二元关系模型。
- ① 权值取相反数，变为求最小权闭合子图。
- ② “对于一条有向边(x,y)，若x选，y必须选” 可以转化为：若x选、y不选，就需要付出+∞的代价。
- ③ 设x与T之间的边表示选，S与x之间的边表示不选。

最大权闭合子图

解方程后得到的图，假设vx>0，令S→x、x→T的容量同时加vx。



按照二元关系模型，算出的答案就是 正点权和减去最小割！



最大权闭合子图

- 求法：
- 建立源点S和汇点T，源点S连所有的正权点，容量为该点点权；
- 所有负权点连汇点T，容量为该点点权的绝对值；
- 对于原图中的边<u,v>，连边<u,v>，容量+∞。

最大权闭合子图

- 定理1：最大权闭合子图的点权和 = 所有正权点权值和 – 最小割。
- 定理2：上述图的最小割包含两类边：

  （1）S到“不在最大权闭合子图内的正权节点”的边

  （2）“在最大权闭合子图内的负权节点”到T的边
- 定理2的推论：在残余网络中由源点S能够访问到的点，就构成一个点数最少的最大权闭合子图。

最大密度子图

- 一个无向图G=(V,E)的边数|E|与点数|V|的比值 D=|E|/|V| 称为它的密度。
- 求G的一个子图G'=(V',E')，使得 D'=|E'|/|V'| 最大。
- 这是一个01分数规划问题的模型，应该二分答案d。
- 构造新函数F(d)=Max{|E'|-d*|V'|}，设d$为最优解。
- F(d)>0 ⇨ d<d$， F(d)=0 ⇨ d=d$， F(d)<0 ⇨ d>d$。
- 二分下界：1/n，上界：m/1，精度：1/n^2。

最大密度子图

- 在F(d)=Max{|E'|-d*|V'|}中，(V',E')是G=(V,E)的子图。
- 这隐含着一个条件：若边e=(u,v)∈E'，则必须有u,v∈V'。
- 因此可以通过最大权闭合子图模型求解，点数边数均为O(|V|+|E|)：
- 把边e看作点Ve，点权为1；
- 本图中的点依然保留，点权-d；
- 对于每条边e=(u,v)，建立两条有向边(Ve,u)，(Ve,v)。

最大密度子图

- 用二元关系模型来解最大密度子图！
- ① 二分答案d，问题转化为 |E|-d*|V| 最大，即 d*|V|-|E| 最小。
- ② 割开点与S之间的边表示不选、与T之间的边表示选，带权值 0 或 d。
- ③ 二元关系：若一条边的两个端点同时被选，则获得该边的权值 -1。边权≤0，意味着二元关系的K值≥0。
- ④ 解方程，建图，求最小割，问题解决！

最大密度子图

- 避免实数→让式子乘2，避免负容量→每条边的容量加U=|E|。
- 源点S向每个点v连边，容量为U。
- 每个点v向汇点T连边，容量为$U+2d-degree(v)$。
- 原图的每个边(u,v)拆为两条有向边(u,v),(v,u)，容量为1。
- 当$Cut(S,T)$取最小值时，F(d)有最大值$\frac{U|V|-MinCut(S,T)}{2}$。

最大密度子图(推广)

- 把最大密度子图推广到带非负边权的无向图：边权和/点数 定义为密度。
- 此时分数规划的函数变为F(d)=$\sum_{e\in E'} weight(e) - \mathrm{d} * |\mathrm{V}'|$。
- 只需改变$degree(v)$为与v相连的所有边的边权和;
- 原图中的边e在网络中的对应容量从1变为$weight(e)$;
- U值从|E|变为总边权和。
- 其余求解方法相同，只不过二分的精度有所改变。

最大密度子图(推广)

- 推广到带点权和非负边权的无向图：(边权和+点权和)/点数定义为密度。
- 此时分数规划的函数变为F(d)=$\sum_{e\in E'} w(e)-\sum_{v\in V'}(d-p(v))$。
- 在上一页的基础上，再令v到T的容量变为U+2(d-p(v))-degree(v)。
- 再令U变为 2*总点权绝对值和+总边权和；
- 其余求解方法与上一页相同。
- 模板题：POJ3155 Hard Life

例题

- NOI2009 Plants VS Zombies
- http://www.lydsy.com:808/JudgeOnline/problem.php?id=1565
- （1）建立图论模型。
- 把每个植物当做一个顶点，植物携带的资源数目为顶点的权值。
- 如果植物b在植物a的攻击范围内，连接一条有向边<a,b>，表示a可以保护b。
- 由于僵尸从右向左进攻，可以认为每个植物都被它右边相邻的植物保护，对于每个植物a（除最左边一列），向其左边的相邻植物b，连接一条有向边<a,b>。

例题

- （2）可能有一些植物是互相保护的，都不能被吃掉。
- 因此可以用拓扑排序去掉图中的环，使图得到简化。
- （3）吃掉一个植物时，必须把它的前驱（保护它的植物）先吃掉。
- 根据最大权闭合图的定义，把上面构的图的转置后，可以吃掉的植物构成一个闭合子图。
- 最优解就是转置后的图的最大权闭合图。
- 按照最大权闭合图的算法建图。

例题

- 建立源S和汇T。
- 图中原有的转置后的边容量设为∞。
- 从S向每个权值为正的点连接一条容量为该点权值的有向边。
- 从每个权值非正的点向T连接一条容量为该点权值绝对值的有向边。

- 求S到T的最大流（最小割），最大子权闭合图的权值就是所有正权点权值之和 – 最大流流量。

例题

- NOI2006 最大获利
- 解法一：最大权闭合子图
- 建立源点S和汇点T。
- 从源点出发连向用户群，容量为公司的收益。
- 从中转站出发连向汇点，容量为投入的成本。
- 若用户群i需要中转站x和y，连边(i,x)(i,y)，表示选i后必须选x和y。

例题

- 解法二：最大密度子图
- 该问题中用户群与中转站之间的关系，与最大密度子图的隐含条件“若边e=(u,v)∈E’，则必须有u,v∈V’ ”相同。
- 在此前提下，该问题实际上是最大化$\sum_{e\in E'} profit(e) - \sum_{v\in V'} cost(v)$。
- 令$w(e) = profit(e)$，$d - p(v) = cost(v)$，直接套用点边均带权的最大密度子图模型求解。

谢谢大家

lyd@pku.edu.cn