网络流的应用

- 网络流的构图
- 网络流的应用
- 网络流的优化

网络流的构图

- 按照题意建图
- 抽象模型
  - 将限制条件当作容量限制
  - 将价值条件当作费用限制
- 拆点 同一个点拆成两个或者多个点
  - 不同的点代表不同的意义
  - 两点之间连边作为限制流量

网络流的解法

- 网络流
  - 网络流求解最大流 / 最小割
  - 网络流验证解的合法性
  - ……
- 平面图中网络流转成最短路
- 最大流 -> 最小割 -> 其他思路解决

网络流的优化

- 网络流构图优化
  - 动态拆点
  - 动态加边
  - 缩点 POJ1149

网络流的例题

K-联赛

- 现有N只球队，已知每只球队获胜和失败的场次分别为Wi和Li，以及两两球队之间还将举行的比赛场数Cij。球队获胜得一分，失败不得分，问最终有哪些球队可以获得冠军（允许并列第一的情况出现）

K-联赛

- 枚举每个队伍，判断是否能拿冠军
  - 这个队伍最后可以拿多少分?
    - 贪心，后面的场次全胜
  - 如果这个队伍拿冠军，那么别的队伍的分不能超过他的得分
    - 网络流Check一下
    - 怎么建图呢?

跳舞

- 一次舞会有N个男孩和N个女孩。每首曲子开始时，所有男孩和女孩恰好配成N对跳交谊舞。每个男孩都不会和同一个女孩跳两首（或更多）舞曲。
- 有一些男孩女孩相互喜欢，而其他相互不喜欢（不会“单向喜欢”）。每个男孩最多只愿意和K个不喜欢的女孩跳舞，而每个女孩也最多只愿意和K个不喜欢的男孩跳舞。
- 给出每对男孩女孩是否相互喜欢的信息，舞会最多能有几首舞曲?

跳舞

- 假设总共可以跳P轮，如何判断是否合法?
- 如何保证每个人只会和K个不喜欢的人跳舞?

跳舞

- 以舞蹈次数作为边权
  - 源点或汇点连向人的边权为P
  - 两两之间边权为1
- 拆点
  - 每个人拆成两个点，边权为K
  - 如果两两不喜欢，从拆出的点连边；否则正常连边

跳舞

- 如何确定P
    - 暴力枚举
    - O(n) * O(n^2*m)

跳舞

- 如何确定P
  - P满足单调性
  - 二分P
  - O(logn) * O(n^2*m)

跳舞

- 如何确定P
  - 仔细观察原图，每次改变P只是改变了S和T连出的边的容量
  - 每次增广路只会使结果+1
  - P+1会将P情况重新做一遍
  - 暴力增加P，但是在剩余图上不断修改P的影响

MAXIMUM FLOW

- 一个无向图有2n+2个点，分别为(0,0),(0,1)…(0,n),(1,0),(1,1)…(1,n)，图中有三种边
    - (0,i-1) -> (0,i) 容量为ai
    - (1,i-1) -> (1,i) 容量为bi
    - (0, [(i-1)/2]) -> (1,[i/2]) 容量为ci
- 求(0,0)到(1,n)的最大流

MAXIMUM FLOW

- N<=10^3
- 网络流

MAXIMUM FLOW

- N<=10^5
- 平面图最大流 = 对偶图最短路

MAXIMUM FLOW

- N<=10^7
- 最大流 = 最小割

MAXIMUM FLOW

- 如果将图中的无向边转化成有向边该如何处理?

CUTE PANDA

- 总共有N个熊猫，和N个篮子，每个熊猫有ai根竹子，每个篮子可以装最多bi根竹子，第i号熊猫职能将竹子放到i号和(i%N+1)号篮子中去，问最多可以放多少根竹子?
- 50% N<=3000
- 100% N<=10^6

CUTE PANDA

- N<=3000
- 网络流
  - S向熊猫连边 C=ai
  - 篮子向T连边 C=bi
  - 熊猫i向篮子i和i%N+1连边 C=oo

CUTE PANDA

- N<=10^6
- 最大流 = 最小割
- 只要将从S到T的所有可行流量全部切断

最小割

- 有N个点，编号0到N-1，点i和点j之间的有向边容量为i^j，方向从小编号到大编号，问0到N-1的最小割是多少?

- 50% N <= 500
- 100% N <= 10^6

最小割

- 性质:
  - $i > j$，如果$i < i\hat{}(N-1)$ 则一定有$j < j\hat{}(N-1)$

网络流的经典问题

分数规划

$$\text{Minimize} \quad \lambda = f(\mathbf{x}) = \frac{a(\mathbf{x})}{b(\mathbf{x})} \quad (\mathbf{x} \in S)$$

$$s.t. \quad \forall \mathbf{x} \in S, \quad b(\mathbf{x}) > 0$$

- S是解空间，x是解空间中的一个向量

分数规划

- 分数规划一般处理成其非分数形式：

$$g(\lambda) = \min_{\mathbf{x} \in S} \{(a(\mathbf{x}) - \lambda \cdot b(\mathbf{x})\}$$

- 该函数满足:
  - 严格单调递减
  - 如果λ1是原函数最优解，g(λ)=0当且仅当λ=λ1

0-1分数规划

$$\text{Minimize} \quad \lambda = f(\mathbf{x}) = \frac{\sum_i a_i x_i}{\sum_i b_i x_i} = \frac{\mathbf{a}\cdot\mathbf{x}}{\mathbf{b}\cdot\mathbf{x}} \qquad (\mathbf{x} \in \{0,1\}^n)$$

$$s.t. \qquad \mathbf{b}\cdot\mathbf{x} > 0$$

- 将x的取值限定为01向量

网络战争

- 给出一个带权无向图，每条边e有一个权。求将点s和点t分开的一个边割集，使得该割集的平均边权最小，也就是最小化：

$$\frac{\sum_{e\in C} w_e}{|C|}$$

网络战争

$$\text{Minimize} \quad \lambda = f(\mathbf{x}) = \frac{\sum_{e \in E} w_e x_e}{\sum_{e \in E} 1 \cdot x_e} = \frac{\mathbf{w} \cdot \mathbf{x}}{\mathbf{c} \cdot \mathbf{x}}$$

- 原图边权We’ = We - λ
- 二分λ，g(λ)的最小值用最小割模型计算

最大权闭合子图

- 描述：每个点之间有一些依赖（取了某些点才可以取别的点），点上有权值（可正可负），怎么选择点使得获得的权值之和最大。
- 建图： S向所有正权值的点建立权值的边，点和点之间的依赖建立权值oo的边，负权值点向T建立权值绝对值的边。
- 结果 = 正权值之和 - 最小割

最大密度子图

- 给定一个无向图，找到它的一个子图使得改图的边数和顶点数的比值（定义为该图的密度）最大。

- 也就是G=(V,E)，找到子图G’=(V’,E’)，maximize { |E’| / |V’| }

最大密度子图

- h(g) = max { E*Xe - g * V * Xv }

- 二分g的范围 m ~ 1/n， 最小精度 1 / n^2
- 二分次数 log(n^2 * m) = 4log(n)

最大密度子图

- h(g) = max { E*Xe - g * V * Xv }
- 怎么找到Xe和Xv的关系，找到一个X可以完全的代表Xe和Xv?
- Xe = 1 当 e = (u, v), Xu = Xv = 1

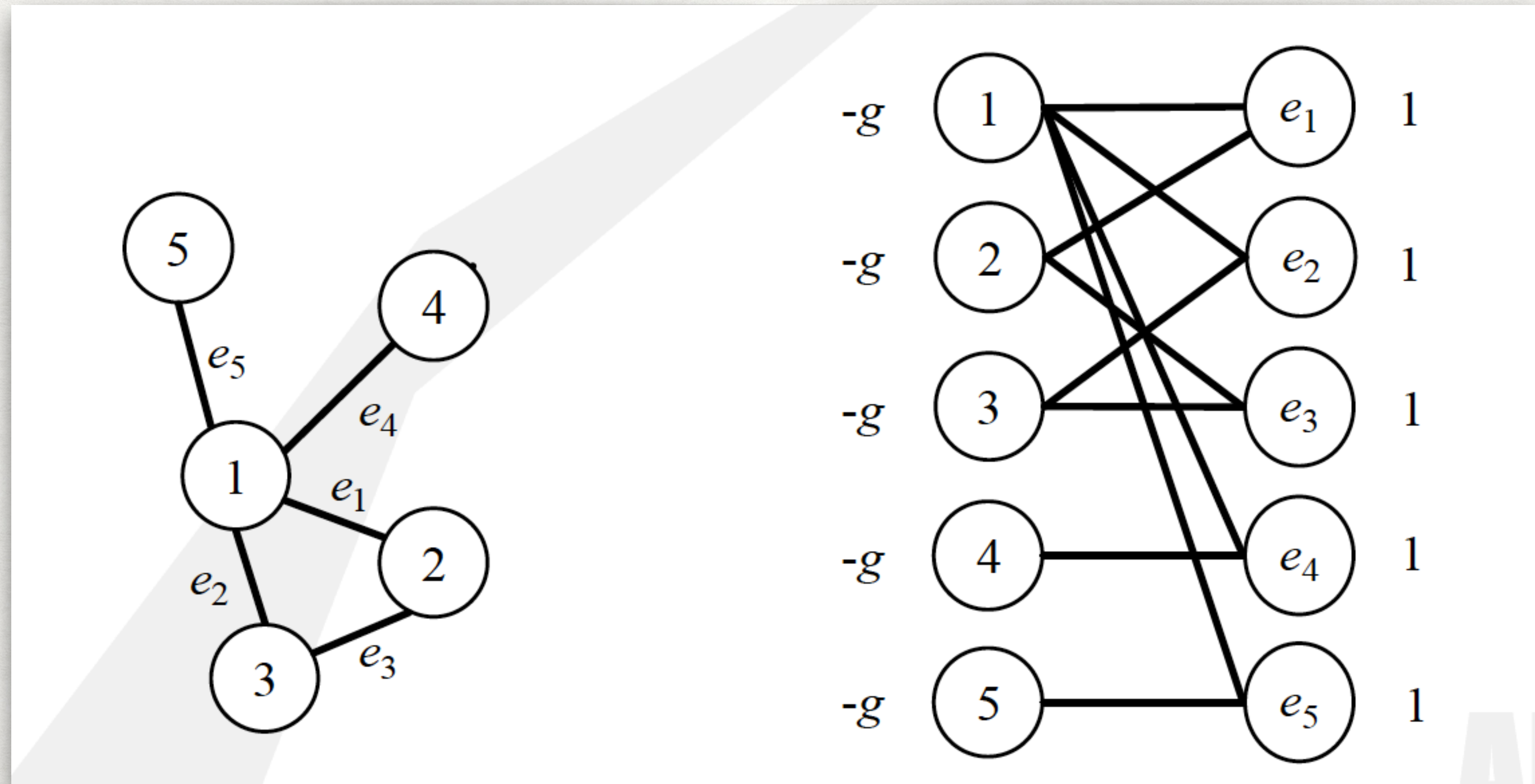
最大密度子图

- h(g) = max { 1*Xe - g * Xv }
- 怎么找到Xe和Xv的关系，找到一个X可以完全的代表Xe和Xv?
- Xe = 1 必有 e = (u, v), Xu = Xv = 1
- 也就是说：当Xu和Xv都选的时候可以选Xe，反之不成立（必要条件）

最大密度子图

- 最大权闭合图： G’ = (V’, E’)
- V’ = V + Ve
- 将所有的边看成新的点加入点集，带上点权为1，点的点权为-g
- 转化为：最大权闭合图的模型

最大密度子图



最大密度子图

- maximize { |E’| / |V’| }
- 如果Xu = Xv = 1， 那么e=(u, v)，Xe =1显然比Xe = 0优，说明上述问题仍然有改进的空间
- 给定V’，得到V’的导出子图一定是最优的E’
- 2 * E’ = deg(V’) - cut[V’, V-V’]

最大密度子图

- max {1*Xe - g*Xv} = min {g*Xv - 1*Xe}
- g*Xv - 1*Xe
- = g * Xv - (deg(Xv) / 2 - cut[v, V-v])
- = (g-deg(v) / 2) * Xv + cut[v, V-v]

最大密度子图

$$V_N = V \bigcup \{s, t\}$$

$$E_N = \{\langle u, v\rangle \mid (u, v) \in E\} \bigcup \{\langle s, v\rangle \mid v \in V\} \bigcup \{\langle v, t\rangle \mid v \in V\}$$

$$\begin{cases} c(u,v) = 1 & (u,v) \in E \\ c(s,v) = U & v \in V \\ c(v,t) = U + 2g - d_v & v \in V \end{cases}$$

- U是一个大常数，防止负数出现
- ans = (U*n - cut[S, T]) / 2

最大密度子图

- 边权：可以看成是有了多条重边

$$V_N = V \bigcup \{s,t\}$$

$$E_N = \{\langle u,v\rangle \mid (u,v) \in E\} \bigcup \{\langle s,v\rangle \mid v \in V\} \bigcup \{\langle v,t\rangle \mid v \in V\}$$

$$\begin{cases} c(u,v) = w_e & (u,v) \in E \\ c(s,v) = U & v \in V \\ c(v,t) = U + 2g - d_v & v \in V \end{cases}$$

$$U = \sum_{e \in E} w_e$$

二分图的最小点权覆盖和最大点权独立

- 点覆盖集：图中每条边至少有一个点在集合中
- 点独立集：集合中任意两个点在图中都不相邻
- 最小点权覆盖
- 最大点权独立
- NP-Complete -> 限定在二分图中可做

二分图最小点权覆盖

- 二分图求匹配的建图
- 最小点权 联想 最小割
- (u,v)不能割 权值oo
- S->u Wu   u in X
- v->T Wv   v in Y
- 得到的最小割就是最小点权之和

二分图的最大点权独立

- 二分图中最小点权覆盖集和最大点权独立集互补

$$\begin{aligned} &\neg(u \in V' \text{ and } v \in V') \\ \Leftrightarrow\quad &\neg u \in V' \text{ or } \neg v \in V' \\ \Leftrightarrow\quad &u \in \overline{V'} \text{ or } v \in \overline{V'} \end{aligned}$$

- 求出最小点权覆盖之和求补集就是最大点权独立集