拟阵与贪心算法-yxy

目

录

1.独立系统

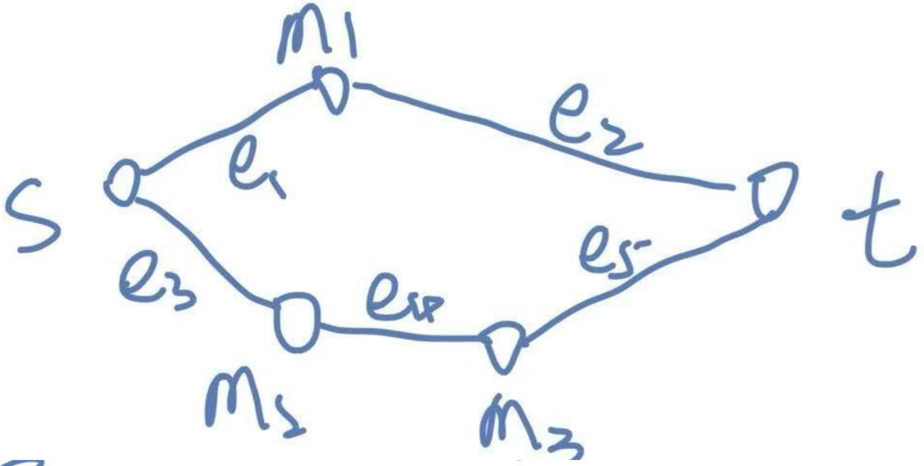
2.贪心算法

3.拟阵

4.拟阵交

独立系统

我们首先引入这一类问题：我们在执行某种算法时可以选择给定集合内的一个个元素，构成一个新的集合，这个集合存在某种约束，当我们选得足够多时就会打破这个约束，此时该算法就会停止，我们选择的集合就会停留在打破前的一刻。我们的目标一般是最大化（最小化）我们所选集合中所有元素的取值和。



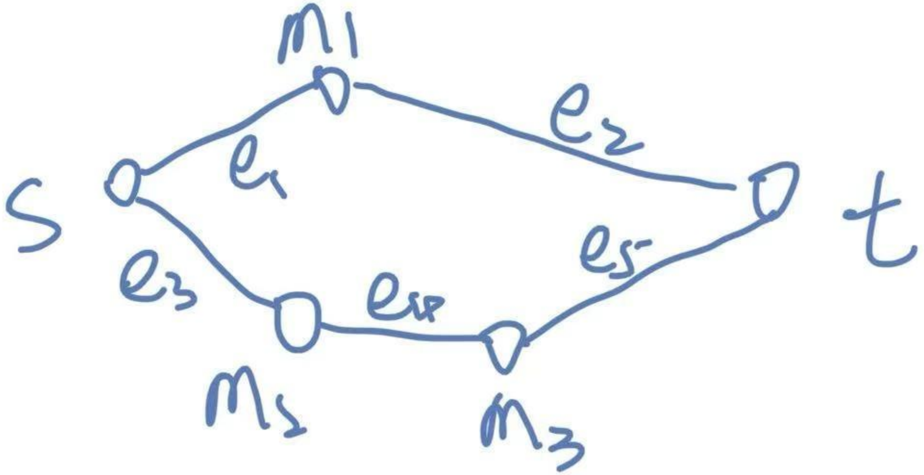
$$\mathcal{F}=\{\emptyset,\{e_1\},\{e_1,e_2\},\{e_2\},\{e_3\},\{e_4\},\{e_5\},\{e_3,e_4\},\{e_3,e_5\},\{e_4,e_5\},\{e_3,e_4,e_5\}\}$$

对于左图：

- 我们可选集合为 $E=\{e_1,e_2,e_3,e_4,e_5\}$。
- 我们所选集合 X (中途) 可以为 $X=\{e_1\}$。
- 约束：所选集合是某条路径的一部分。
  i.e. $X\in\mathcal{F}$:= 所有完整路径的一部分的集合
  上图中，我们的 $\mathcal{F}$ 其实就是上下两条路径的集合的幂集的并。
- 何时停止：约束“紧”时停止，i.e.
  我不能再加一个边，使得添加后构成的集合 $X'=X\cup e\in\mathcal{F}$

独立系统

我们首先引入这一类问题：我们在执行某种算法时可以选择给定集合内的一个个元素，构成一个新的集合，这个集合存在某种约束，当我们选得足够多时就会打破这个约束，此时该算法就会停止，我们选择的集合就会停留在打破前的一刻。我们的目标一般是最大化（最小化）我们所选集合中所有元素的取值和。



$B_1 = \{e_1, e_2\}$

$B_2 = \{e_3, e_4, e_5\}$

对于左图：

- 我们可选集合为 $E = \{e_1, e_2, e_3, e_4, e_5\}$。
- 我们所选集合 X (中途) 可以为 $X = \{e_1\}$。
- 约束：所选集合是某条路径的一部分。

  i.e. $X \in \mathcal{F} :=$ 所有完整路径的一部分的集合

  上图中，我们的 $\mathcal{F}$ 其实就是上下两条路径的集合的幂集的并。
- 何时停止：约束“紧”时停止，i.e.

  我不能再加一个边，使得添加后构成的集合 $X' = X \cup e \in \mathcal{F}$

独立系统

设 E 是一个有限集合，$\mathcal{F} \subseteq 2^E$ 为 E 的子集族。若满足以下两个条件，则称二元组 $(E, \mathcal{F})$ 为一个独立系统（*Independence System*）：

(M1) $\varnothing \in \mathcal{F}$;

(M2) 若 $Y \subseteq X \in \mathcal{F}$，则 $Y \in \mathcal{F}$

定义（基）. 设 $(E, \mathcal{F})$ 为一个独立系统。若 $B \in \mathcal{F}$ 且对任意 $e \in E \setminus B$，有 $B \cup \{e\} \notin \mathcal{F}$，则称 B 为一个基（*basis*）。换言之，基是极大的独立集。

定义（相关集）. 在独立系统 $(E, \mathcal{F})$ 中，若 $X \subseteq E$ 且 $X \notin \mathcal{F}$，则称 X 为一个相关集（*dependent set*）。即独立集族 $\mathcal{F}$ 的补集族。

定义（圈）. 在独立系统 $(E, \mathcal{F})$ 中，若 $C \subseteq E$ 满足：

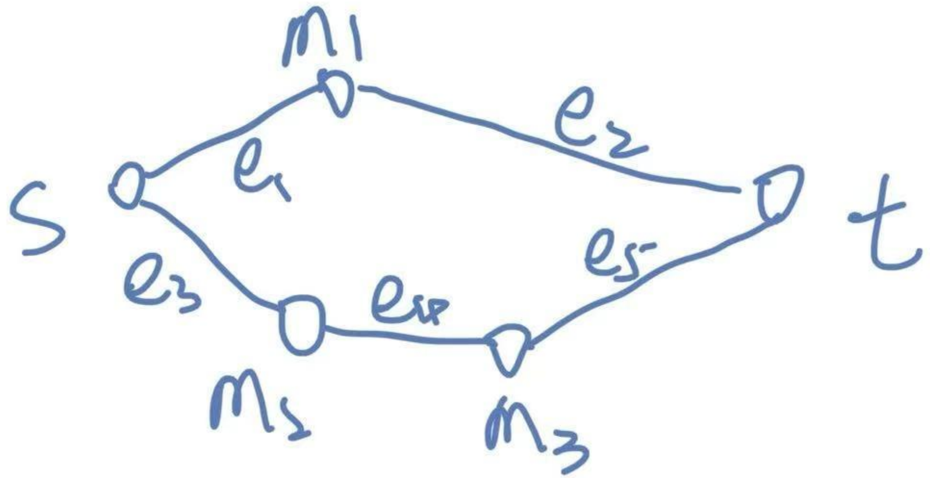
$$C \notin \mathcal{F}, \quad \text{且对任意真子集} C' \subsetneq C,\ C' \in \mathcal{F},$$

独立系统

设 E 是一个有限集合，$\mathcal{F} \subseteq 2^E$ 为 E 的子集族。若满足以下两个条件，则称二元组 $(E, \mathcal{F})$ 为一个 独立系统（*Independence System*）：

(M1) $\varnothing \in \mathcal{F}$；

(M2) 若 $Y \subseteq X \in \mathcal{F}$，则 $Y \in \mathcal{F}$



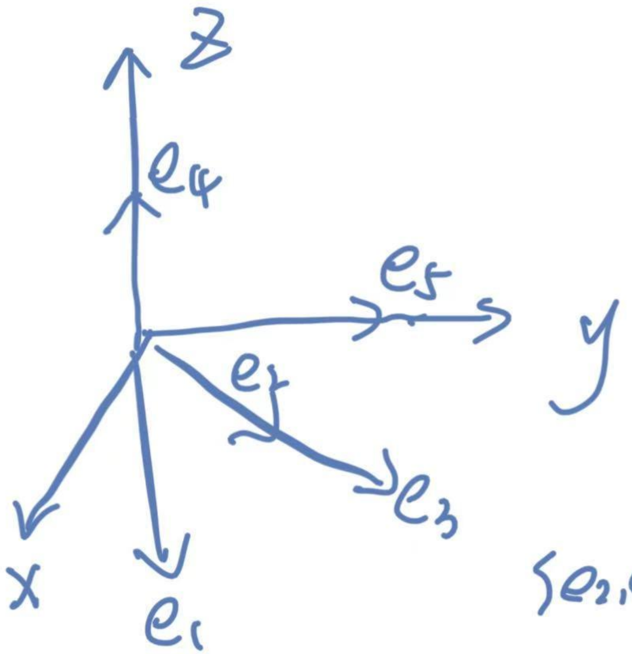
比如说左图{ e1,e2,e3}这个集合就是一个相关集，而{e1,e3}就是一个最小的相关集(圈)。

独立系统(更多例子)

设 E 是一个有限集合，$\mathcal{F} \subseteq 2^E$ 为 E 的子集族。若满足以下两个条件，则称二元组 $(E, \mathcal{F})$ 为一个 独立系统（*Independence System*）：

(M1) $\varnothing \in \mathcal{F}$；

(M2) 若 $Y \subseteq X \in \mathcal{F}$，则 $Y \in \mathcal{F}$



目标：找线性无关的一组向量，张成一个三维空间

$\mathcal{F} = \{\varnothing\} \cup$ 所有线性无关集合

$X = \{e_2\}$ 是一个独立集

$B = \{e_1, e_3, e_4\}$ 是基

$\{e_2, e_3\}$ 是圈，$\{e_1, e_3, e_5\}$ 不是

独立系统的优化问题

正如前面路径问题所示，独立系统一般有两种问题，分别是最大化问题和最小化问题。

两个基本优化问题

最大化问题	最小化问题
Input: 独立系统 $(E, \mathcal{F})$ $c : E \to \mathbb{R}$	Input: 独立系统 $(E, \mathcal{F})$ $c : E \to \mathbb{R}$
Output: 独立集 $X \in \mathcal{F}$，使得 $c(X)$ 最大	Output: 基 B，使得 $c(B)$ 最小
(也就是求权重最大的独立集)	(也就是求权重最小的极大独立集)

独立系统优化的经典例子

(1) 最大权稳定集问题（Maximum Weight Stable Set Problem）

给定图 $G=(V,E)$ 及权函数 $c:V(G)\to\mathbb{R}$，求 G 中权值最大的稳定集 $X\subseteq V(G)$。

$$E=V(G),\qquad \mathcal{F}=\{F\subseteq E \mid F\text{ 在}G\text{ 中是稳定的}\}.$$

(2) 旅行商问题（Traveling Salesman Problem, TSP）

给定完全无向图 G 及权函数 $c:E(G)\to\mathbb{R}_+$，求 G 中权值最小的哈密顿回路。

$$E=E(G),\qquad \mathcal{F}=\{F\subseteq E \mid F\text{ 是}G\text{ 中某个哈密顿回路的边集的子集}\}.$$

(3) 最短路径问题（Shortest Path Problem）

给定图 G（有向或无向）、权函数 $c:E(G)\to\mathbb{R}$ 以及顶点 $s,t\in V(G)$，其中 t 从 s 可达，求 G 中最短的 s-t 路径。

$$E=E(G),\qquad \mathcal{F}=\{F\subseteq E \mid F\text{ 是}G\text{ 中某条}s\text{-}t\text{ 路径的边集的子集}\}.$$

(4) 背包问题（Knapsack Problem）

给定 $n\in\mathbb{N}$、非负数 c_i,w_i $(1\le i\le n)$ 及容量 W，求集合 $S\subseteq\{1,2,\ldots,n\}$，使得 $\sum_{j\in S}w_j\le W$ 且 $\sum_{j\in S}c_j$ 最大。

$$E=\{1,2,\ldots,n\},\qquad \mathcal{F}=\Big\{F\subseteq E \;\Big|\; \sum_{j\in F}w_j\le W\Big\}.$$

(5) 最小生成树问题（Minimum Spanning Tree Problem）

给定连通无向图 G 及权函数 $c:E(G)\to\mathbb{R}$，求权值最小的生成树。

$$E=E(G),\qquad \mathcal{F}=\{F\subseteq E \mid F\text{ 是}G\text{ 中一片森林的边集}\}.$$

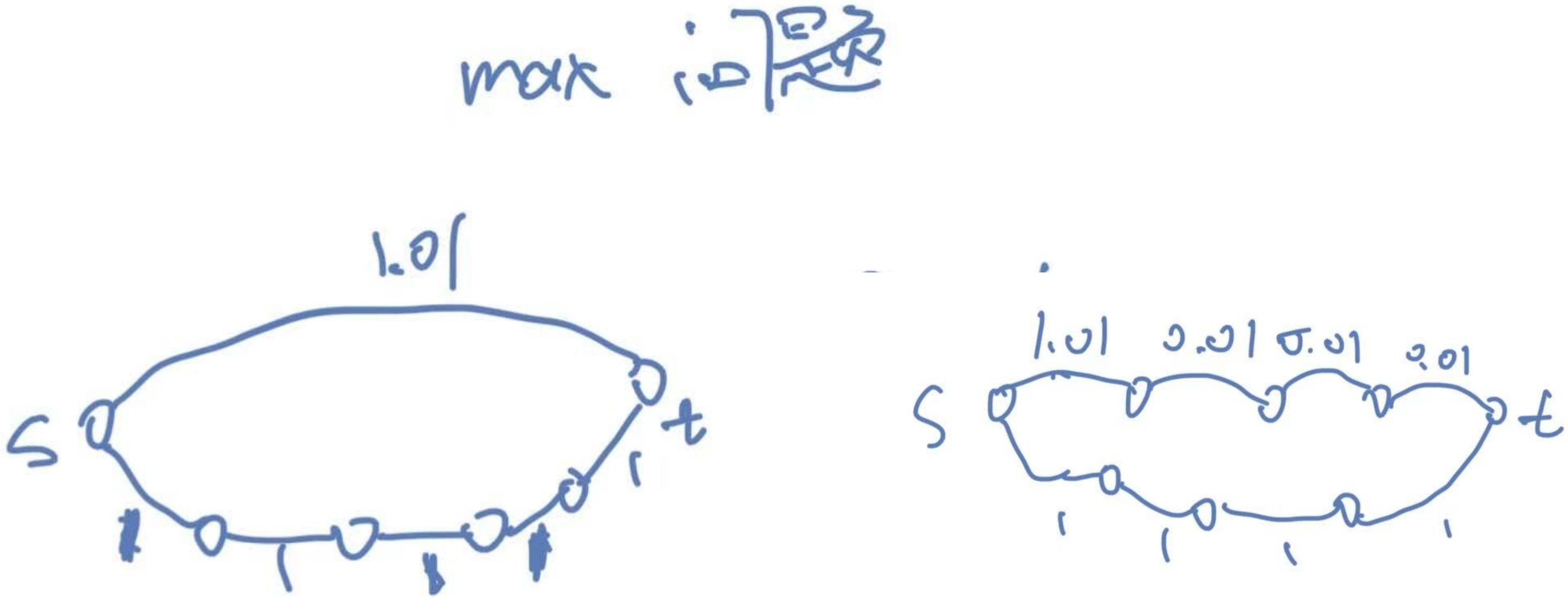
(6) 最大权森林问题（Maximum Weight Forest Problem）

给定无向图 G 及权函数 $c:E(G)\to\mathbb{R}$，求权值最大的森林。

$$E=E(G),\qquad \mathcal{F}=\{F\subseteq E \mid F\text{ 是}G\text{ 中一片森林的边集}\}.$$

直觉

独立系统似乎非常适用贪心算法，因为它的M2（遗传性）意味着我们可以从一个小的独立集慢慢长成基。但在某些独立系统当中，使用贪心算法可能是不好的，我们有以下两个直觉：



秩商

以上“不好”的直觉可以用一个概念来刻画：

秩与秩商（Rank and Rank Quotient）

设 $(E,\mathcal{F})$ 是一个独立系统，对任意 $X \subseteq E$，定义：

上秩（Rank / Upper Rank）

$$r(X) = \max\{\,|Y| : Y \subseteq X,\ Y \in \mathcal{F}\,\} = \max\{\,|Y| : Y \text{ 是} X \text{ 中的基}\,\}.$$

下秩（Lower Rank）

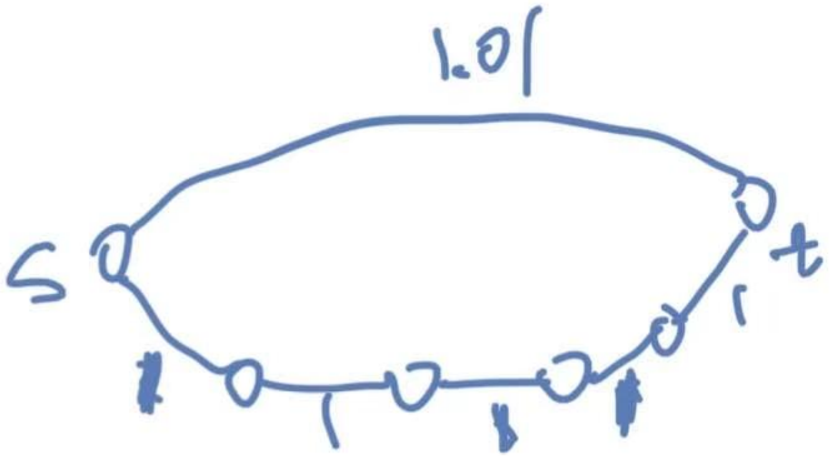
$$\rho(X) = \min\{\,|Y| : Y \text{ 是} X \text{ 中的基}\,\}.$$

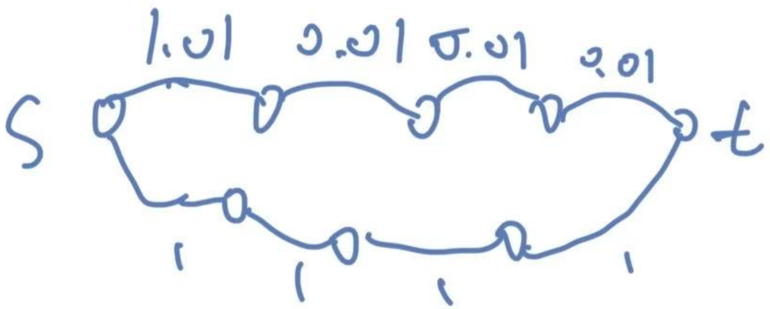
秩比（Rank Quotient）

$$q(E,\mathcal{F}) = \min_{X \subseteq E} \frac{\rho(X)}{r(X)}, \qquad \text{显然} q(E,\mathcal{F}) \le 1.$$

秩比（商）的刻画

max 问题





- 对于左图，假如我们取 $X =$ 所有边的集合 $= E$，我们就能得出该独立系统的秩商为 $1/4$.
- 对于右图，假如我们取 $X =$ 上面第一条边 $\cup$ 下面所有边，我们发现该独立系统的秩商也为 $1/4$.

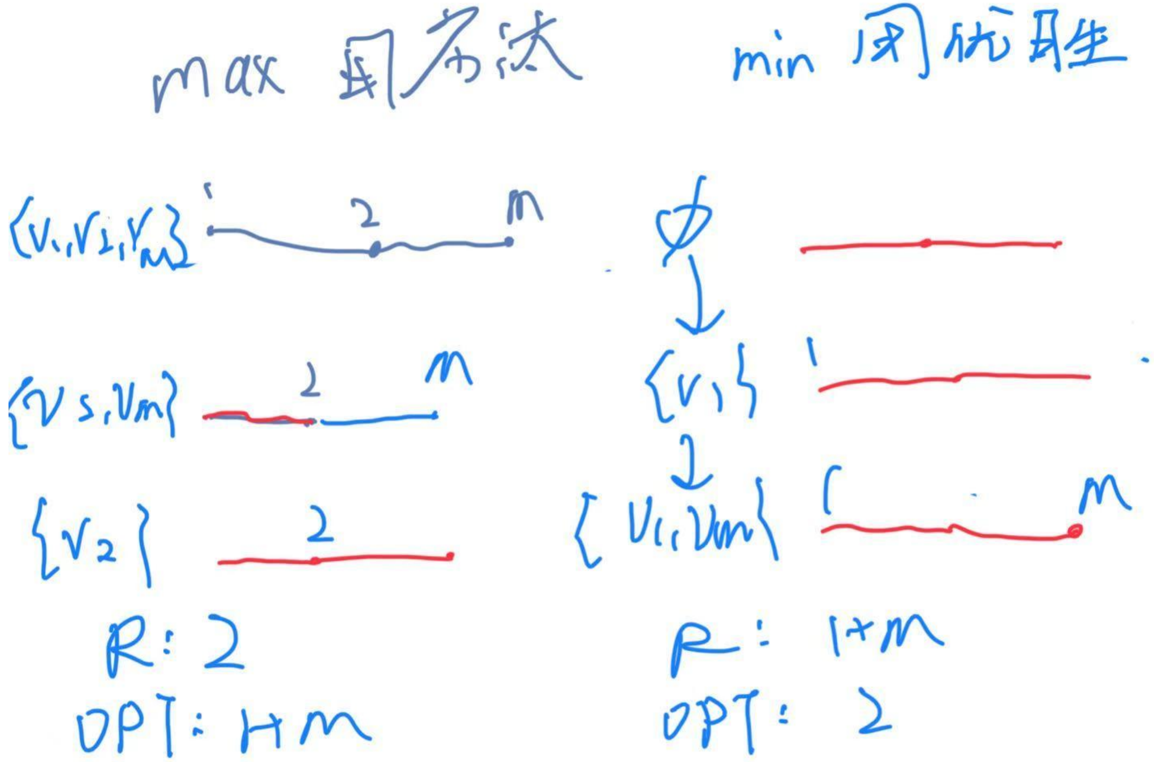
独立系统与贪心算法

虽然贪心算法有时不好，但有时也很好，所以我们希望更加定量地表示出它好友多好，差有多差。常见的贪心思路有两种：优胜法，劣汰法。分别与极大化问题和极小化问题一共有四种组合，我们将说明只有以下两种是有效率的。

极大化问题	极小化问题
优胜法 (**Best-In**)	劣汰法 (**Worst-Out**)
(1) 排序，设 $c(e_1) \geq c(e_2) \geq \cdots \geq c(e_n)$	(1) 排序，设 $c(e_1) \geq c(e_2) \geq \cdots \geq c(e_n)$
(2) 令 $F \leftarrow \emptyset$	(2) 令 $F \leftarrow E$
(3) $\mathbf{For}\ i = 1\ \mathbf{to}\ n\ \mathbf{do}:$	(3) $\mathbf{For}\ i = 1\ \mathbf{to}\ n\ \mathbf{do}:$
$\quad \mathbf{if}\ F \cup \{e_i\} \in \mathcal{F}:$	$\quad \mathbf{if}\ F/\{e_i\}$ 包含一个基：
$\qquad$ 令 $F \leftarrow F \cup \{e_i\}$	$\qquad$ 令 $F \leftarrow F/\{e_i\}$

独立系统与贪心算法

我们考虑一个极小顶点覆盖问题：



左图可以说明如果max问题用劣汰，而min问题用优胜的话可以达到任意差的界。

Best-In 在独立系统近似界定理

设 $(E, \mathcal{F})$ 是一个独立系统，$c: E \to \mathbb{R}^+$。记 $G(E, \mathcal{F}, c)$ 为 Best-In 算法的目标函数值，$\mathrm{OPT}(E, \mathcal{F}, c)$ 为最优目标函数值，则有：

$$q(E, \mathcal{F}) \leq \frac{G(E, \mathcal{F}, c)}{\mathrm{OPT}(E, \mathcal{F}, c)} \leq 1,$$

对任意的 c 成立；且存在 c 使得该不等式的下界可以达到。

右边成立是trivial的，关键是证明左边成立。

独立系统与贪心算法

证明

设 $E=\{e_1,e_2,\ldots,e_n\}$ 是基本元素集，其中

$$c(e_1)\geq c(e_2)\geq\cdots\geq c(e_n)>0.$$

记 G_n 为 Best-In 的解，O_n 为最优解。定义

$$E_j=\{e_1,\ldots,e_j\},\quad G_j=G_n\cap E_j,\quad O_j=O_n\cap E_j,\quad j=1,2,\ldots,n,$$

并额外设定 $G_0=O_0=\varnothing$, $c(e_{n+1})=0$。则有：

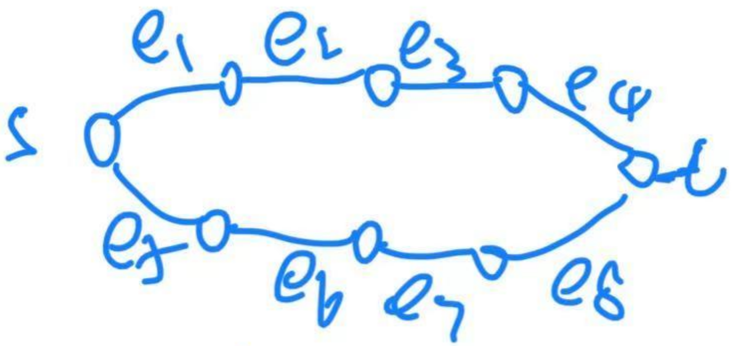
$$\begin{aligned}
c(G_n)&=\sum_{j=1}^{n}\left(|G_j|-|G_{j-1}|\right)c(e_j)\\
&=\sum_{j=1}^{n}|G_j|\left(c(e_j)-c(e_{j+1})\right)
\end{aligned}$$

$$\begin{aligned}
&\geq\sum_{j=1}^{n}\rho(E_j)\left(c(e_j)-c(e_{j+1})\right)\\
&\geq q(E,\mathcal{F})\sum_{j=1}^{n}r(E_j)\left(c(e_j)-c(e_{j+1})\right)\\
&\geq q(E,\mathcal{F})\sum_{j=1}^{n}|O_j|\left(c(e_j)-c(e_{j+1})\right)\\
&=q(E,\mathcal{F})\sum_{j=1}^{n}\left(|O_j|-|O_{j-1}|\right)c(e_j)\\
&=q(E,\mathcal{F})\cdot c(O_n).
\end{aligned}$$

那么，这个下界是紧的吗？

独立系统与贪心算法

我们可以利用那个取到秩商的集合来构造出来一个紧的例子，如下所示：



一个取到秩商的集合为 $X=\{e_1,e_5,e_6,e_7,e_8\}$

则令 $c(e_i)=1$, for all $e_i \in X$, $c(e_i)=0$. for other

显然，在 X 中我们能取到紧的界，那么将 $X \in F$ 扩展为 (E, F) 上的基也可以取到

独立系统与贪心算法

总之我们确定可以用秩商来紧紧地bound住best-in贪心算法的下界，不过独立系统的秩商似乎是很难确定的，因为我们需要考虑所有E的子集的上秩与下秩的比，幸而我们有以下的估计手段：

估计秩商的一个定理

设 $(E,\mathcal{F})$ 是一个独立系统。若对任意 $Z\in\mathcal{F}$、$e\in E$，集合 $Z\cup\{e\}$ 至多包含 P 个圈，则有：

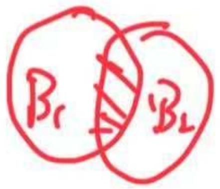
$$q(E,\mathcal{F})\geq\frac{1}{P}.$$

证明：我们任取任意 Z 当中的两个基 B_1,B_2，我们可以不失一般性地假设 $|B_1|>|B_2|$，现在我们考虑把 B_2 “改造”成 B_1：

- 我们设 $B_1\setminus B_2=\{e_1,e_2,...,e_n\}$
- 我们现在只需要把 n 个不在 B_2 的元素逐个加进去就好了。
- 如果加进去一个元素产生了 () 个圈，我们最多把原来属于 B_2 的元素移出去 () 个就好，其中 ()$\leq P$，现在我们就改造完成了。

独立系统与贪心算法

总结一下前三步



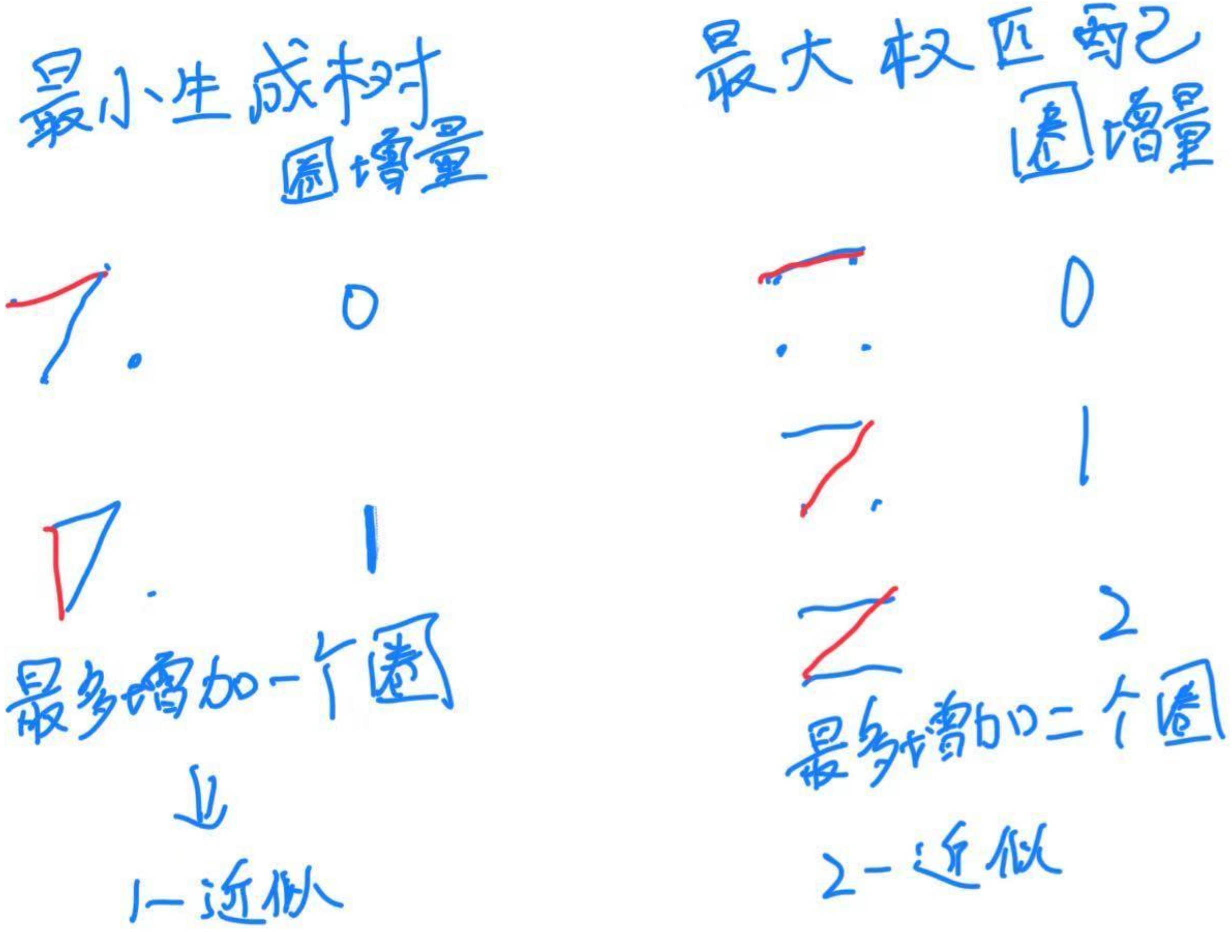
$$B_2 \xrightarrow{\text{移出最多 } P\times n \text{ 个，加入 } n \text{ 个}} B_1$$

可知 $|B_2\setminus B_1| \le P\times n = P\,|B_1\setminus B_2|$

$$\Rightarrow \frac{1}{P} \le \frac{|B_1\setminus B_2|}{|B_2\setminus B_1|} = \frac{|B_1|-|B_1\cap B_2|}{|B_2|-|B_1\cap B_2|} \le \frac{|B_1|}{|B_2|}$$

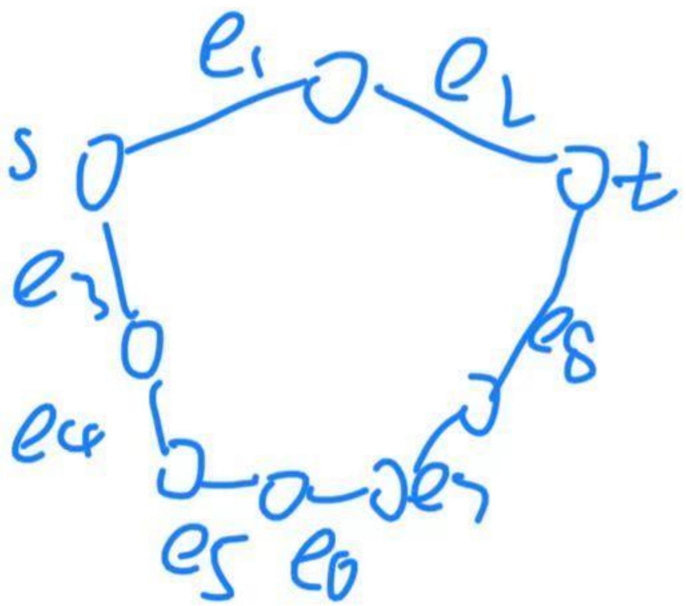
独立系统与贪心算法

总结一下，现在我们可以用“向独立系统中添加一个元素所增加的圈”来估计出best-in贪心算法的近似界。



独立系统与贪心算法

在Worst-out算法中，直觉上如果我们希望算法尽可能的坏，我们需要把那些比较好的解都给剔除掉，最后仅仅剩下来一个很坏的基。显然，原问题的不同基能够在剔除之后保留下来的能力不同。



$B_1 = \{e_1, e_2\}$. 能够承受住 $e_3 \sim e_8$ 共6次剔除

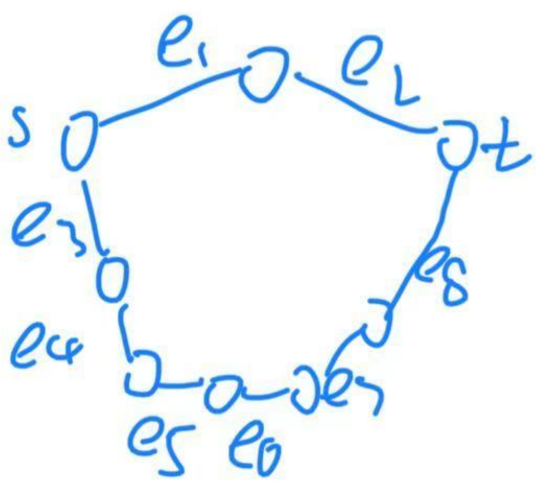
$B_2 = \{e_5 \sim e_8\}$. 能够承受住 e_1, e_2 2次剔除

独立系统与贪心算法

定义（对偶独立系统）： 独立系统 $(E,\mathcal{F})$ 的对偶为 $(E,\mathcal{F}^*)$，其中

$$\mathcal{F}^* = \{F \subseteq E \mid \text{存在}(E,\mathcal{F}) \text{ 的基} B \text{ 使得} F \cap B = \varnothing\}$$

显然，对偶独立系统的对偶就是它本身。



$B_1=\{e_1,e_2\}$. 能够承受住 $e_3\sim e_8$ 共6次删除

$B_2=\{e_3\sim e_8\}$. 能够承受住 e_1,e_2 2次删除

我们可以发现对偶独立系统的基与原系统的基一一对应，其实就可以反应出原系统的基的“抗删除”能力。

独立系统与贪心算法

定理： 设 $(E, \mathcal{F})$ 是一个独立系统，$c: E \to \mathbb{R}^+$，设 $G(E,\mathcal{F},c)$ 为 Worst-Out 算法的目标函数值，$\mathrm{OPT}(E,\mathcal{F},c)$ 是最优目标函数值，则

$$1 \le \frac{G(E,\mathcal{F},c)}{\mathrm{OPT}(E,\mathcal{F},c)} \le \max_{F\subseteq E} \frac{|F|-\rho^*(F)}{|F|-r^*(F)}$$

对任意 c 成立，其中 ρ^*、r^* 为对偶独立系统 $\star$（在集合 F 上）的下秩和秩，且存在 c 使得该上界可以取到。

我们再详细一点描述 Worst-out 的方法：

- 我们将所有元素从高到低排列为 $\{c(e_1), c(e_2)..., c(e_n)\}$
- 然后思考是否应该移除第一个元素，我们问 oracle，移除这个元素后剩下的集合包含一个基吗？
  - 如果包含，那我们直接移除。
  - 如果不包含，我们就保留这个元素。
- 重复执行

换言之，我们不可能再减少我们在前 j 个元素中留下的元素构成的集合中的元素了，否则我们就无法构成基。

记 G_n 为 Worst-Out 的解，O_n 为最优解。定义

$$E_j = \{e_1, \ldots, e_j\}, \quad G_j = G_n \cap E_j, \quad O_j = O_n \cap E_j, \quad j = 1, 2, \ldots, n,$$

我们首先证明 $E_j \setminus G_j \in \mathcal{F}^*$

- 这是显然的，因为我们可以把 $E_j \setminus G_j$ 扩张成在 $(E, \mathcal{F}^*)$ 上的基，只要避开 G_n 即可。

我们接下来证明 $E_j \setminus G_j$ 是 $(E, \mathcal{F}^*)$ 在 E_j 上的基。假如不是，那么存在 $e \in G_j$，使得 $(E_j \setminus G_j) \cup e$ 是对偶独立系统的基。意味着我们可以在前 j 个元素中多剔除一个，矛盾。

独立系统与贪心算法

既然 $E_j \setminus G_j$ 是基，其大小 $|E_j| - |G_j| \geq \rho^*(E_j)$

又因为 O_n 是 $(E, \mathcal{F})$ 的一个基，所以容易得出 $E_j/O_j \in \mathcal{F}^*$，从而 $|E_j| - |O_j| \leq r^*(E_j)$。故 $|G_j| \leq |E_j| - \rho^*(E_j)$，$|O_j| \geq |E_j| - r^*(E_j)$，我们令

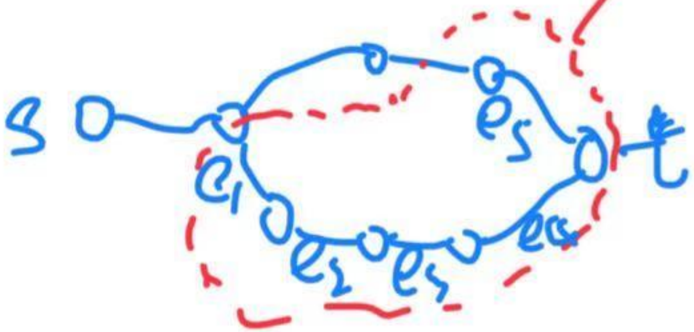
$$\lambda = \max_{F \subseteq E} \frac{|F| - \rho^*(F)}{|F| - r^*(F)},$$

就可以得到 $|G_j| \leq \lambda |O_j|$。

$$\begin{aligned}
c(G_n) &= \sum_{j=1}^{n} (|G_j| - |G_{j-1}|) c(e_j) \\
&= \sum_{j=1}^{n} |G_j| (c(e_j) - c(e_{j+1})) \\
&\leq \lambda \sum_{j=1}^{n} |O_j| (c(e_j) - c(e_{j+1})) \\
&= \lambda c(O_n)
\end{aligned}$$

独立系统与贪心算法

我们可以仿照Best-In的方法构造一个Worst-out的紧的界。



取到入的集合 $\{e_1, e_2, e_3, e_4, e_5\}$

令 $\begin{cases} e_i = 1, & i \in \text{该集合 } F^* \\ e_i = 0, & \text{other} \end{cases}$

$$\frac{\text{贪心}}{\text{最优}} = \frac{4}{1} = \frac{|F| - \rho^*}{|F| - g^*}$$

拟阵

我们观察独立系统在 Best-In 算法上的 Bound，不难发现它会被某个 $X \subseteq E$ 的上秩和下秩 bound 住，并且可以进一步发现，如果一个独立系统在任何 $X \subseteq E$ 的上秩和下秩都一样的话，那贪心就能得到最优解。

定义：满足下列等价条件的独立系统 $(E, \mathcal{F})$ 称为拟阵：

- (M_3) 若 $X, Y \in \mathcal{F}$，且 $|X| > |Y|$，则存在 $e \in X \setminus Y$ 使得 $Y \cup \{e\} \in \mathcal{F}$。
- (M_3') 若 $X, Y \in \mathcal{F}$，$|X| = |Y| + 1$，则存在 $e \in X \setminus Y$ 使得 $Y \cup \{e\} \in \mathcal{F}$。
- (M_3'') 对任意的 $F \subseteq E$，F 上的基有相同的元素个数。

可以推出拟阵的相关结论：

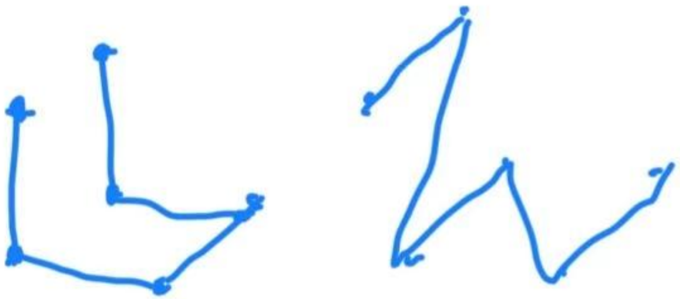
- 拟阵的秩商为 1，所以 Best-In 是最优的。
- 对任意 $X \subseteq E, X$ 上的基包含的元素个数都是 $r(X)$；又由于对任意的 $F \in \mathcal{F}$ 都有 $|F| = r(F)$，故而 $\mathcal{F} = \{F \mid |F| = r(F), F \subseteq E\}$。
- （基交换性质）如果拟阵 $(E, \mathcal{F})$ 有两个不同的基 A, B 且 $a \in A \setminus B$，那么一定存在 $b \in B \setminus A$ 使得 $(A \setminus \{a\}) \cup \{b\}$ 是基；

常见拟阵

2. 图拟阵（Graphical Matroid）

定义：给定一个无向图 $G=(V,E)$，图拟阵 $M(G)$ 的基础集是边集 E，其独立集族是所有不包含环的边集，即所有的森林。

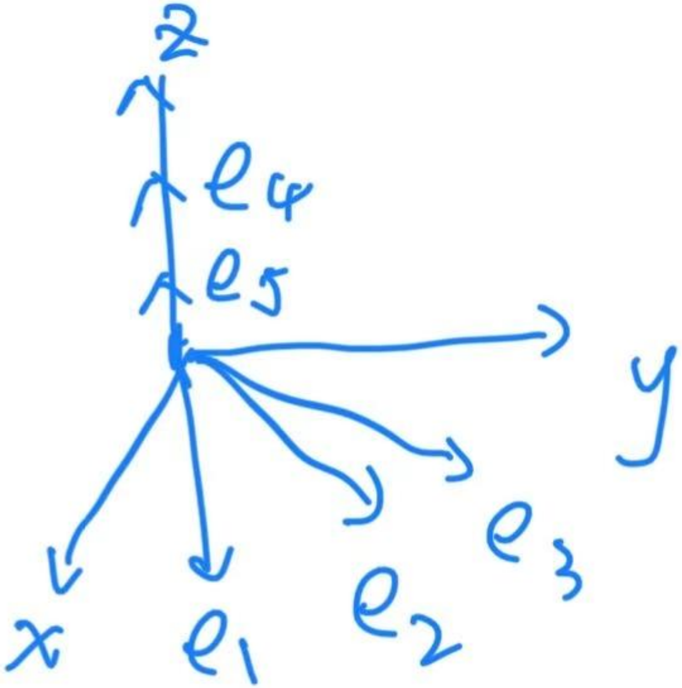
- **基**：图中的生成树（在连通图的情况下）。生成树是极大的独立集，无法再增加边而不形成环。
- **圈**：图中的简单环，去掉环中的任意一条边，剩余部分都为独立集。
- **秩**：$r(E)=|V|-c$，其中 c 是图的连通分支数。对于一个连通的无向图，其秩等于顶点数减一，即 $|V|-1$。



3. 线性拟阵（Linear Matroid）

定义：线性拟阵基于向量空间。给定向量空间 V，基础集 E 是 V 中的一组有限向量，其独立集族是 E 中所有线性无关的向量子集。

- **基**：极大的线性无关向量集，其大小等于向量空间的维数。
- **圈**：最小的线性相关向量集合，其任意真子集都是独立的，而自身是线性相关的。
- **秩**：线性拟阵的秩 $r(E)=\dim(V)$，即向量空间的维数。独立集的大小不能超过向量空间的维数。



常见拟阵

1. 均匀拟阵（Uniform Matroid）

定义：给定基础集 E 和非负整数 k，均匀拟阵 $U_{k,E}$ 的独立集族是所有大小不超过 k 的子集，表示为：

$$\mathcal{I} = \{I \subseteq E \mid |I| \leq k\}。$$

- **基（Bases）**：所有大小为 k 的子集。
- **圈（Circuits）**：所有大小为 $k+1$ 的子集。
- **秩（Rank）**：$r(E) = \min(k, |E|)$，即独立集中最多能有 k 个元素。

eg:你去超市买东西，超市一共有100个东西，你可以任选最多10个带走。

4. 划分拟阵（Partition Matroid）

定义：将基础集 E 划分为不相交的子集 $E_1, E_2, \ldots, E_m$，并为每个子集 E_i 指定一个非负整数 k_i。划分拟阵的独立集族由满足每个部分选取元素数量不超过 k_i 的子集组成，表示为：

$$\mathcal{I} = \{I \subseteq E \mid \forall i, |I \cap E_i| \leq k_i\}。$$

- **基**：满足 $|I \cap E_i| = k_i$ 的独立集是划分拟阵的基。每个基在每个子集中选取了恰好 k_i 个元素。
- **圈**：划分拟阵的圈是最小的依赖集，即包含至少一个元素数量超过 k_i 的子集。
- **秩**：划分拟阵的秩为 $r(E) = \sum_{i=1}^{m} k_i$，即最大独立集的大小等于每个子集中允许选取的最大元素数的总和。

eg:你去超市买东西，超市一共有50个苹果，20个西瓜，1个香蕉。你可最多可以选10个苹果，10个西瓜，1个香蕉。

拟阵的秩函数

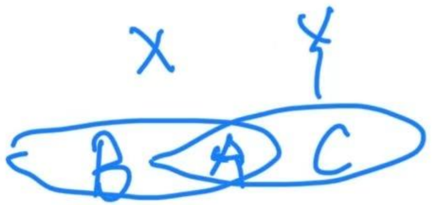
事实上，Worst-Out最优必须要满足对偶独立系统E的任何一个子集上下秩相等，也就是说此对偶独立系统是拟阵。这里我们将证明拟阵的对偶独立系统是拟阵，这样在拟阵上使用Worst-out也是最优的。证明过程需要用到拟阵的秩函数。

定理： 拟阵的秩是一个次模函数。

证明： 如果 r 是独立系统 $(E,\mathcal{F})$ 的秩函数，则 (R1) 和 (R2) 显然成立。若 $(E,\mathcal{F})$ 是一个拟阵，我们还可以证明 (R3)：

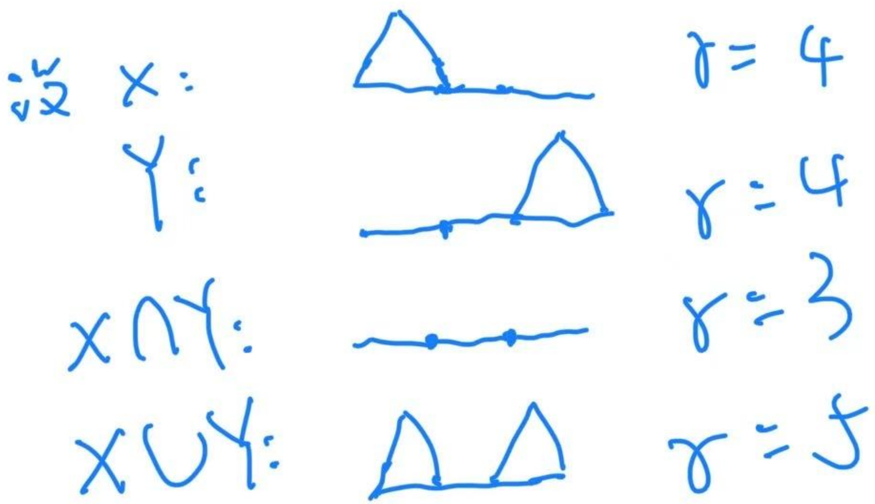
设 $X,Y\subseteq E$，令 A 为 $X\cap Y$ 的一个基。由性质 (M3)，A 可以扩展为 X 的一个基 $A\cup B$，也可以扩展为 $X\cup Y$ 的一个基 $(A\cup B)\cup C$。于是 $A\cup C$ 是 Y 的一个独立子集，因此有

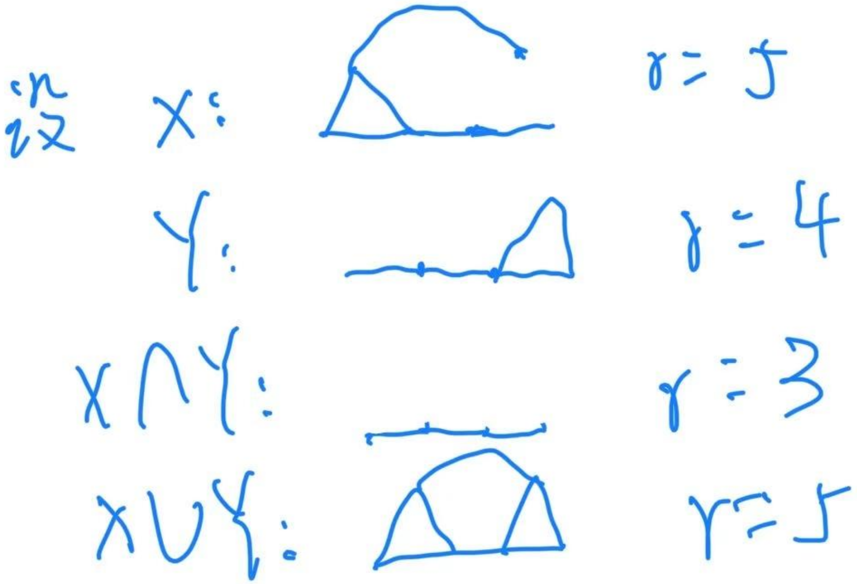
$$
\begin{aligned}
r(X)+r(Y) &\geq |A\cup B|+|A\cup C| \\
&= 2|A|+|B|+|C| \\
&= |A\cup B\cup C|+|A| \\
&= r(X\cup Y)+r(X\cap Y).
\end{aligned}
$$



拟阵的秩函数

直觉上甚至可能会觉得似乎得取等，但是则不然，对于一个最小生成树问题：





拟阵的秩函数

定义： 集合函数 $r: 2^E \to \mathbb{Z}^+ \cup \{0\}$ 是拟阵的秩函数（Rank Function），如果 r 满足：

- （有界性）对于任意的 $X \subseteq E$，$0 \le r(X) \le |X|$；
- （单调性）如果 $Y \subseteq X$，那么 $r(Y) \le r(X)$；
- （次模性）对任意的 $A, B \subseteq E$，有 $r(A) + r(B) \ge r(A \cup B) + r(A \cap B)$。

定理： 对任意一个满足上述条件的 r，我们都可以构造一个拟阵 $(E, \mathcal{F})$。换言之，拟阵的秩函数和拟阵是一一对应的。

证明：只需令

$$\mathcal{F} = \{F \subseteq E : |F| = r(F)\}$$

即可。这是因为：

- $r(\varnothing) = 0 = |\varnothing|$，故 M_1 成立；
- 若 $Y \subseteq X \subseteq E$ 且 $r(X) = |X|$，那么根据次模性和有界性，有

$$r(Y) \ge r(X) - r(X/Y) \ge |X| - |X/Y| = |Y|$$

  又由有界性 $r(Y) \le |Y|$，故 $r(Y) = |Y|$，即 $Y \in \mathcal{F}$，因此 M_2 成立；
- 用反证法，设 M_3 不成立，则存在 $X, Y \subseteq E$，使得 $r(X) = |X|, r(Y) = |Y|$，且 $|X| > |Y|$，并且对任意 $e \in X \setminus Y$ 有 $r(Y \cup \{e\}) = |Y|$。不难由归纳法得出

$$r(X \cup Y) = r(Y) = |Y| < |X| = r(X)$$

  这与单调性矛盾。

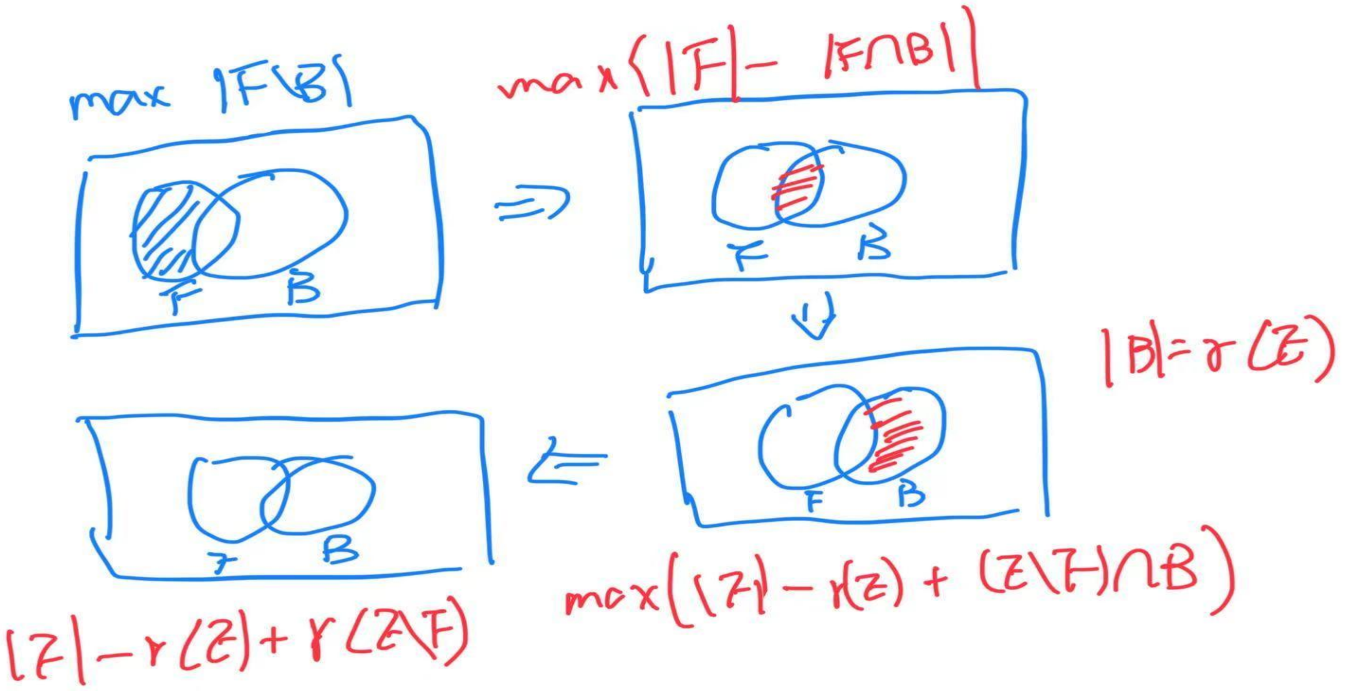
拟阵的秩函数

我们的下一步就是写出拟阵的对偶独立系统的秩函数，如果这个秩函数满足拟阵的秩函数的三个性质，那么拟阵的对偶独立系统就是拟阵，Worst-out就是最优的。

定理： 拟阵的对偶独立系统仍然是拟阵，且对任意的 $F \subseteq E$ 有

$$r^*(F) = |F| + r(E/F) - r(E),$$

其中 r, r^* 分别是拟阵 $(E, \mathcal{F})$ 和其对偶 $(E, \mathcal{F}^*)$ 的秩函数。



拟阵的秩函数

我们的下一步就是写出拟阵的对偶独立系统的秩函数，如果这个秩函数满足拟阵的秩函数的三个性质，那么拟阵的对偶独立系统就是拟阵，Worst-out就是最优的。

定理： 拟阵的对偶独立系统仍然是拟阵，且对任意的 $F \subseteq E$ 有

$$r^*(F) = |F| + r(E/F) - r(E),$$

其中 r, r^* 分别是拟阵 $(E, \mathcal{F})$ 和其对偶 $(E, \mathcal{F}^*)$ 的秩函数。

我们可以分别验证 $r*$ 分组满足拟阵秩函数的有界性、单调性、次模型即可。证明过程略去。

总之，现在我们证明了对于拟阵，Best-In 和 Worst-out 都可以得到最优结果。

拟阵的交

定理： 任何一个独立系统 $(E,\mathcal{F})$ 都是有限多个拟阵的交。

证明： 设 C_i 是 $(E,\mathcal{F})$ 的任一个圈（注意，C_i 是有限多个的），令

$$\mathcal{F}_i = \{F \subseteq E : C_i \not\subseteq F\},$$

即 $\mathcal{F}_i$ 中的元素都不包含 C_i。注意到这些 $\mathcal{F}_i$ 的交不包含原系统的任意一个圈，因此

$$\mathcal{F} = \bigcap_i \mathcal{F}_i,$$

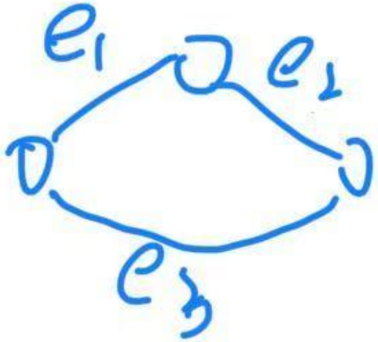
如果 $(E,\mathcal{F}_i)$ 都是拟阵，那么定理 7 就得到了证明。

接下来我们证明 $(E,\mathcal{F}_i)$ 是拟阵。设 $F_1, F_2 \in \mathcal{F}_i$，且 $|F_1| = |F_2| + 1$，我们只需证明其满足 (M_3')，即：我们总可以找到一个 $e \in F_1 \setminus F_2$，使得 $F_2 \cup \{e\} \in \mathcal{F}_i$，如下：

- 若存在 $e \in F_1 \setminus F_2$ 且 $e \notin C_i$，那么 $F_2 \cup \{e\} \in \mathcal{F}_i$；
- 若 $F_1 \setminus F_2 \subseteq C_i$：
  - 若 $|F_1 \setminus F_2| \geq 2$，那么 C_i 中至少有两个元素不在 F_2 中，加入任何一个都不会让 F_2 包含 C_i，即：对 $\forall e \in F_1 \setminus F_2$ 都有 $F_2 \cup \{e\} \in \mathcal{F}_i$；
  - 若 $|F_1 \setminus F_2| = 1$，那么 $F_1 \supset F_2$，也就是说 F_1 与 F_2 仅差一个元素，设 $F_1 \setminus F_2 = \{e\}$，则 $F_2 \cup \{e\} = F_1 \in \mathcal{F}_i$。

拟阵的交

可以想象任何一个独立系统都是若干“禁圈拟阵”交出来的，所以，某种程度上来说，“圈”其实表达了独立系统的特殊性。利用禁圈拟阵交实际上就是一刀刀地把这个独立系统雕刻出来。



$2^E: \{\phi, \{e_1\}, \{e_2\}, \{e_3\}, \{e_1,e_2\}, \cancel{\{e_2,e_3\}}, \cancel{\{e_1,e_3\}}, \cancel{\{e_1,e_2,e_3\}}\}$

禁: $\{e_1,e_3\} \Rightarrow$ —

禁: $\{e_2,e_3\} \Rightarrow$ —

最后得到 $\mathcal{F} = \{\phi, \{e_1\}, \{e_2\}, \{e_3\}, \{e_1,e_2\}\}$

拟阵的交

定理：若独立系统 $(E,\mathcal{F})$ 是 P 个拟阵的交，则 $q(E,\mathcal{F})\geq 1/P$。**证明**：

我们首先证明这样一个引理：一个拟阵的任何独立集 X 中加入 E/X 中的一个元素 e 后，至多含有一个圈；用反证法，设存在两个圈 $C_1\cup C_2\subseteq X\cup\{e\}$，那么

$$(C_1\cup C_2)/\{e\}\subseteq X.$$

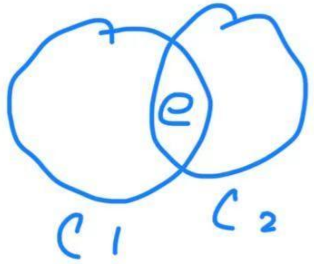
- 如果 $e\in C_1/C_2$，则 $C_2\subseteq X$，与 X 是独立集矛盾；$e\in C_2/C_1$ 同理。
- 如果 $e\in C_1\cap C_2$，由于圈（极小相关集）的极小性，一定不会出现一个圈包含另一个圈，因此 C_1/C_2 非空，设存在 $e'\in C_1/C_2$。由于 $(C_1\cup C_2)/\{e\}\subseteq X$，则 $(C_1\cup C_2)/\{e\}$ 是独立集；由于 C_1 是圈，则 $C_1/\{e'\}$ 是独立集，我们将它扩展为 $C_1\cup C_2$ 上的极大独立集 Z。由于 C_2 是圈，一定有 $Z\not\supseteq C_2$，因此

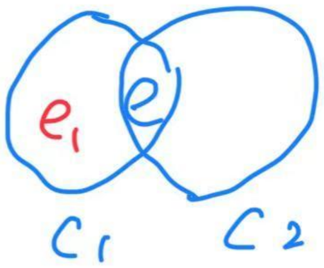
$$|Z|<|(C_1\cup C_2)/\{e\}|,$$

与 Z 的极大性矛盾。

$e=C_1\cap C_2$

$\exists\, e_1\in C_1/C_2$, s.t. C_1/e是独立集





构造的Z比$C_1\cup C_2$少至少2个元素

拟阵交与贪心算法

备用引理 1：我们首先证明这样一个引理：一个拟阵的任何独立集 X 中加入 E/X 中的一个元素 e 后，至多含有一个圈；用反证法，设存在两个圈 $C_1 \cup C_2 \subseteq X \cup \{e\}$，那么

$$(C_1 \cup C_2)/\{e\} \subseteq X.$$

备用引理 2：若对任意 $Z \in \mathcal{F}$、$e \in E$，集合 $Z \cup \{e\}$ 至多包含 P 个圈，则有：

$$q(E,\mathcal{F}) \geq \frac{1}{P}.$$

定理：若独立系统 $(E,\mathcal{F})$ 是 P 个拟阵的交，则 $q(E,\mathcal{F}) \geq 1/P$。

总的证明：如果一个独立系统是 P 的拟阵的交，那么往这个独立系统的任何独立集 X 添加 e 最多产生 P 个圈，用备用引理 2 的结论即可证明。

有了该定理意味着，如果我们能够把一个独立系统巧妙地表示成几个拟阵的交，那么我们就能够给出Best-In贪心算法的界。

拟阵交与贪心算法(eg1)

有了该定理意味着，如果我们能够把一个独立系统巧妙地表示成几个拟阵的交，那么我们就能够给出Best-In贪心算法的界。

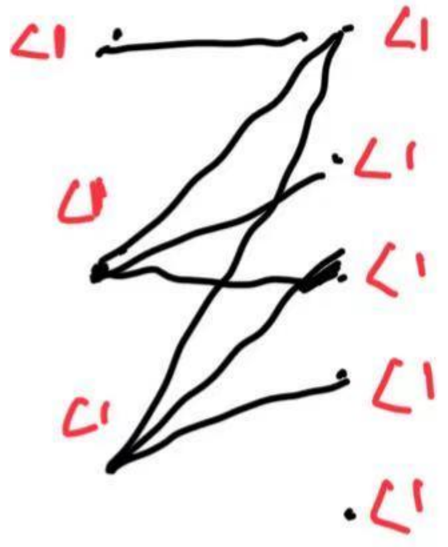
以二部图的匹配为例，设二部图 $G=(A\cup B,E)$，

$$\mathcal{F}=\left\{M\subseteq E: M\ \text{是}G\ \text{中的匹配}\right\};$$

则 $(E,\mathcal{F})$ 是两个拟阵 $(E,\mathcal{F}_1)$ 和 $(E,\mathcal{F}_2)$ 的交，其中，

$$\mathcal{F}_1=\{F\subseteq E: |\delta_F(u)|\le 1,\ u\in A\},$$

$$\mathcal{F}_2=\{F\subseteq E: |\delta_F(v)|\le 1,\ v\in B\}.$$



因此，（最大权）二部图匹配问题的贪心算法，其近似比最差为 2。

拟阵交与贪心算法(eg2)

有了该定理意味着，如果我们能够把一个独立系统巧妙地表示成几个拟阵的交，那么我们就能够给出Best-In贪心算法的界。

给定有向赋权图 $G=(V,E)$，其中 $|V|=n$，指定顶点 $s,t\in V$ 作为起点和终点，哈密顿路径（Hamiltonian path）问题是寻找一条从 s 开始到 t 结束、长度为 $n-1$ 的简单路径（途中经过所有其他节点且只经过一次），且其权重之和最大。

考虑三个拟阵 $(E,\mathcal{F}_1)$、$(E,\mathcal{F}_2)$、$(E,\mathcal{F}_3)$ 的交，其中：

$$\mathcal{F}_1=\left\{X\subseteq E: \text{对}\forall v\in V\setminus\{s\}\ \text{有}|\delta_X^+(v)|\le 1,\ \text{且}\delta_X^+(s)=0\right\},$$

$$\mathcal{F}_2=\left\{X\subseteq E: \text{对}\forall v\in V\setminus\{t\}\ \text{有}|\delta_X^-(v)|\le 1,\ \text{且}\delta_X^-(t)=0\right\},$$

$$\mathcal{F}_3=\left\{X\subseteq E: \text{无向图}G'=(V,X)\ \text{是森林}\right\}.$$

可见近似比为最差为 3。

参考

90%：数学规划与运筹学（7）贪心算法的统一视角-金鱼马
7%：拟阵-OI Wiki
2%：Combinatorial Optimization: Theory And Algorithms-Bernhard Korte
1%：Matroids（视频）-Federico Ardila