# 线性规划松弛算法

## 1. Vertex Cover

$\min \sum_{i=1}^{n} w_i x_i$ $w_i$：顶点权重 $x_i$：是否在顶点覆盖中

$x_i + x_j \geq 1 \quad (i,j) \in E$ 每条边至少有一个顶点在顶点覆盖中

$x_i \in \{0,1\} \xrightarrow{IP \to LP} x_i \geq 0 \quad i=1,2,\cdots,n$

$C_{appro} \geq C_{IP} \geq C_{LP}$ ~ 不是可行解

$$\frac{C_{appro}}{C_{IP}} \leq \frac{C_{appro}}{C_{LP}} \leq \alpha$$

C：cost 目标函数值
IP：integer programming
LP：linear programming
appro：近似算法

### A. 一个从LP→IP的简单算法

设 $x^*$ 为 LP 最优解

$$x^* = \begin{bmatrix} x_1^* \\ \vdots \\ x_n^* \end{bmatrix} \xrightarrow{\text{rounding}} \bar{x} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad \text{其中 } \bar{x_i} = \begin{cases} 1 & x_i^* \geq \frac{1}{2} \\ 0 & x_i^* < \frac{1}{2} \end{cases}$$

$C_{LP} = \sum_{i=1}^{n} w_i x_i^*$

$C_{appro} = \sum_{i=1}^{n} w_i \bar{x_i} \leq 2C_{LP}$ （最多全部从 $\frac{1}{2}$ 放大到1）

**线性规划可以给出一个下界，但又不想解线性规划**

$$\Downarrow$$

**原始对偶算法**

(Dual) $\max \sum_{e\in E} y_e$ 边

$$\text{s.t. } \sum_{i\in e} y_e \le w_i \quad \forall i\in V$$
$$y_e \ge 0$$

互补松弛条件

$$\begin{cases} x_i\left(w_i - \sum_{i\in e} y_e\right) = 0 \\ y_e(x_i + x_j - 1) = 0 \end{cases}$$

PCS : primer complementary slackness

DCS: Dual complementary slackness

**若满足以下条件，我们可以证明近似比为2**

若 $x, y$ 可行且为整数，并满足

PCS 成立

$x_i = 0$ OR $x_i \ne 0 \Rightarrow w_i = \sum_{i\in e} y_e$

DCS 不太差

$y_e = 0$ OR $y_e \ne 0 \Rightarrow 1 \le x_i + x_j \le 2$

则 $\sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n}\left(\sum_{i\in e} y_e\right) x_i$ $= \sum_{i=1}^{n}$ (i这个点连的所有边的权值) $\cdot x_i$

$= \sum_{e\in E}$ 边的权值 $y_e \cdot (x_i + x_j)$

∵ 一条边 $y_e$ 对应一对点 $(i, j)$

$$\le 2 \sum_{e\in E} y_e$$

$$\le 2 \sum_{i=1}^{n} w_i x_i^*$$ 弱对偶

$$\le 2 C_{IP}^*$$

**下面我们给出算法来满足红色框中的条件**

B. 算法

① $x=0$，$y=0$，所有边未标记

这种情况下 $\begin{cases} x \text{ 不可行（不满足 } x_i+x_j \geq 1\text{）} \\ y \text{ 可行} \end{cases}$

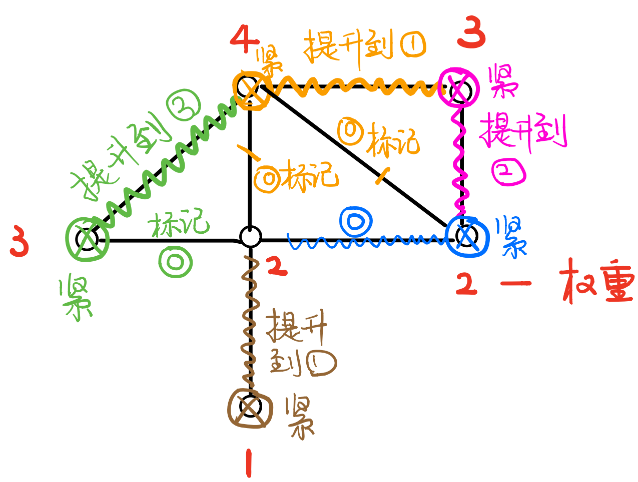
② 任选一条未标记的边 e,提升 e,直至其某一端的约束变紧

即 $x_i(w_i - \sum_{i\in e} y_e)=0$，满足 PCS

③ $x_i=1$，将 i 加入顶点覆盖 已满足PCS 又满足DCS

与顶点i相连的边全部标记

④ 重复直到所有的边都被标记




step 1
step 2
step 3
step 4
step 5

顶点：$3+4+3+2+1=13$

边：$3+1+2+0+1=7$

$$\sum_{i=1}^{n} w_i x_i \leq 2 y_e$$

# Unrelated 平行机调度（负载均衡 load balancing）

$\min T$

$\sum_{j=1}^{n} P_{ij} x_{ij} \le T \quad i=1,\cdots,m$ 安排在第i个机器上的负载之和

$\sum_{i=1}^{m} x_{ij} = 1 \quad j=1,\cdots,n$ 一个工件只能被安排在一个机器上

$x_{ij} = \{0,1\} \Rightarrow x_{ij} \ge 0$

m：机器数量
n：工件数量
$x_{ij}$：第j个工件是否安排到机器 i 上
$P_{ij}$：第j个工件在机器i上的负载

这IP对应的 LP 结果很糟糕

如： m 个机器， 一个 job,其长度为 m
则IP， 这个 job 放到一个机器上， T=m
而LP, 拆分 job 到m个机器， T=1
LP给出的下界太松了

下面我们会给出一个二近似算法

## 改进

① 先猜一个T

将所有job放到一个机器上或者用greedy算法

将每个job放到$P_{ij}$最小的机器

这样猜得一个 $T_0 \le \sum_{i,j} P_{ij} \le mnP_{max}$

② 然后对T从$T_0$开始进行二分

如果 $P_{ij} > T$，则加入 $x_{ij}=0$ 约束

这就不会出现上面糟糕的例子中 $P_{ij}=m, T=1$ 的情况

再解线性规划判断T是否有可行解

若有：二分变小

若没有：二分变大

二分查找 run $\lceil \log T^0 \rceil$ 轮　$\log T^0 = \log m + \log n + \log P_{max}$ 多项式

这样二分得到的$T^*$构成了IP的下界

怎么求LP最优解中得到IP的解呢？

我们再来分析一下问题

$$\boxed{\begin{aligned} &\sum_{j=1}^{n} P_{ij} x_{ij} \le T \quad i=1,\cdots,m \\ &\sum_{i=1}^{m} x_{ij} = 1 \quad j=1,\cdots,n \end{aligned}} \Rightarrow \begin{aligned} &\text{一共有 } m+n \text{ 个约束} \\ &m \times n \text{ 个变量} \end{aligned}$$

$\Downarrow$

则非0变量个数 $\le m+n$

**n个job 可以拆成2部分**

$n = n_1 + n_2$　$\begin{cases} n_1 \text{ 完整的job} \\ n_2 \text{ 拆分到不同机器上的job} \end{cases}$

| $n_1$ | $n_2$ |
|---|---|
| 每个job对应1个$x_{ij}$变量 | 每个job对应2个及以上$x_{ij}$变量 |
| $n_1$ | $\geq 2n_2$ |

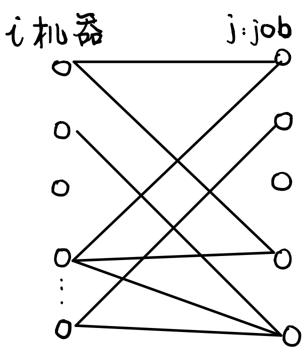
$\therefore m+n \geq$ 非0变量数 $\geq n_1 + 2n_2$

$\begin{cases} m+n \geq n_1 + 2n_2 \\ n = n_1 + n_2 \end{cases} \Rightarrow n_2 \leq m$

即被拆job数量≤机器数量

**用以下给每台机器分配job,将被拆分的job合起来，每个分别分配到不同机器上，可以证明这个算法的近似比为2**

构造二分图：左边m台机器，右边n个job



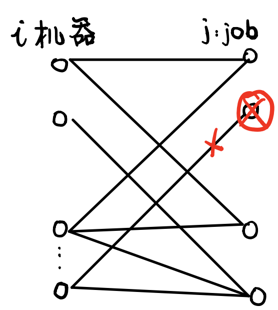

$x_{ij} > 0$ 则在机器i和job j之间有一条边

边数 $\leq m+n$
顶点数 $= m+n$

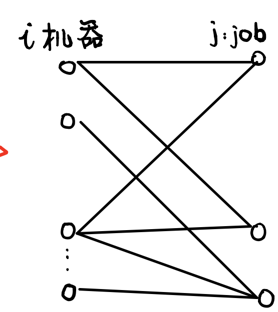
伪树：边数≤顶点数

① **如果二分图连通**

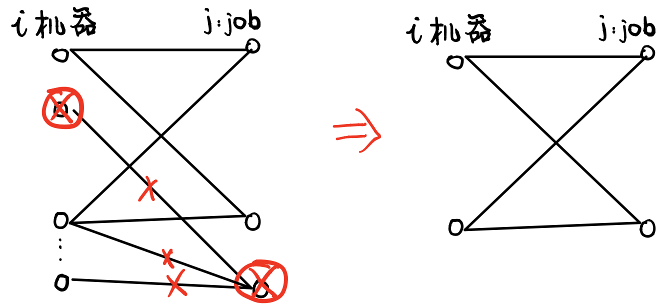
A. 先考虑只分配到一个机器上的job, $x_{ij} = 1$
那就将job i 分配给机器 i,并且删掉了这个点和(i,j) 这条边
连通性不改变



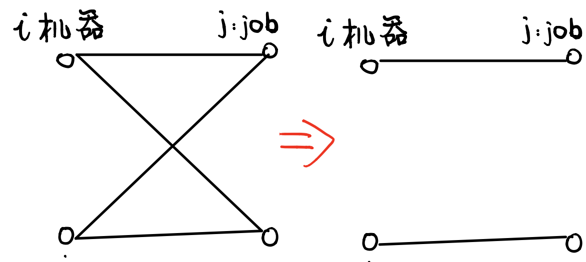

$\Rightarrow$



B. 图中只剩下在LP中分配到多个机器上的job
考虑度为1的机器 i,假设唯一连的边为(i,j),
那么就将整个 job j分配给机器 i
然后去掉机器 i, job j 以及 j 连的所有边
连通性不改变




C. 剩下的图中只能有偶环，可以得到一个完美匹配
一个 job 匹配到一台机器




这样，每台机器除了分配到LP分配到的完整job外，最多分配到一个LP中拆分到多个机器的job。每台机器的总负载最多位2T

$$\text{LP完整job} \leq \sum_{j=1}^{n} p_{ij} x_{ij} \leq T \quad i=1,\cdots,m$$

因为前面二分时，如果 $p_{ij} > T$，则加入 $x_{ij} = 0$ 约束
所以保证没有 $>T$ 的 job，因此 LP 拆分 job 的原本负载也 $<T$

② 如果原图不是连通的，各个连通分量之间不会相互影响，因此，每个连通分量可以分别用上述算法分配job