第七讲

基于LP松弛的近似算法

LP rounding based 近似算法

- 许多组合优化问题都可以用线性整数规划表示
- 但仅有极少数可由 LP 直接求解
- 大多数问题是 NP-难的，近似算法是一类重要的求解方法

I：问题的实例（极小化问题）

A：近似算法（多项式时间）

A(I)：算法 A 的目标函数值

OPT(I)：最优目标函数值

- 若 $\exists\, r \geq 1$，对 $\forall I$，有：

$$A(I) \leq r \cdot OPT(I) \quad \cdots\cdots \quad ①$$

则 A 称为该问题的 r-近似算法

而 $\rho = \inf\{r \mid r \text{ 满足①式}\}$ 为近似比

近似比的等价定义：

$$\rho = \sup_{I} \left\{ \frac{A(I)}{OPT(I)} \right\}$$

对极大化问题：

r-近似：$\forall I$, $OPT(I) \leq r \cdot A(I)$

$$\rho = \sup_{I} \left\{ \frac{OPT(I)}{A(I)} \right\}$$

一般地，

$$\rho = \sup_{I} \left\{ \frac{OPT(I)}{A(I)}, \frac{A(I)}{OPT(I)} \right\}$$

如何证明给定算法 A 的近似比呢？

以极小化问题为例：

1> $\forall I$, $A(I) \leq r \cdot \underline{OPT(I)}$ 如何估计？

$$\begin{cases} OPT(I) \geq LB(I) \\ A(I) \leq r \cdot LB(I) \end{cases}$$

2> $\forall \varepsilon > 0$, $\exists I_\varepsilon$, 使得：

$$A(I_\varepsilon) \geq (r-\varepsilon) \cdot OPT(I_\varepsilon)$$

构造 I_ε

我们希望设计近似比尽量小的近似算法。那么，在 $P \neq NP$ 等假设下，什么样的近似比是最好可能的？

PTAS (Polynomial Time Approximation Scheme)

$\{A_\varepsilon\}$: $\forall \varepsilon > 0$, $\forall I$,

$$A_\varepsilon(I) \le (1+\varepsilon)\cdot OPT(I)$$

其运行时间为 $|I|^{O(f(\frac{1}{\varepsilon}))}$

EPTAS: $|I|^{O(1)} \cdot f(\frac{1}{\varepsilon})$

FPTAS: $|I|^{O(1)} \cdot (\frac{1}{\varepsilon})^{O(1)}$

负面结果 (To sharpen the results)

基于复杂性假设:
$$\begin{cases} P \neq NP \\ NP \neq ZPP \\ \vdots \\ UGC \end{cases}$$

$$\Longrightarrow \begin{cases} \text{NO FPTAS} \\ \text{NO EPTAS} \\ \text{NO PTAS} \\ \text{NO } (r-\varepsilon)\text{-近似} \\ \cdots \end{cases}$$

回到 LP-rounding based 算法

$$(IP)\quad \min c^T x$$
$$Ax \geq b$$
$$x \geq 0 \quad x \in Z^+$$

$$\Downarrow \text{松弛}$$

$$(LP)\quad \min c^T x$$
$$Ax \geq b$$
$$x \geq 0$$

得到 x^*_{LP}

$$\begin{array}{ccc} x^*_{LP} & \xrightarrow{\text{rounding}} & x_{IP} \\ \downarrow & & \downarrow \\ c^*_{LP} & \text{目标函数值} & c_{IP} \end{array}$$

近似比 $\cancel{\rho \leq} \dfrac{c_{IP}}{c^*_{IP}} \leq \boxed{\dfrac{c_{IP}}{c^*_{LP}}}$

注意 $\dfrac{c_{IP}}{c^*_{LP}} \geq \boxed{\dfrac{c^*_{IP}}{c^*_{LP}}}$

"Integrality Gap" 整数间隙，不能太大

背包问题

$$\max\ w=\sum_{i=1}^{n} p_i x_i$$

s.t. $$\sum_{i=1}^{n} s_i x_i \le C,\quad i=1,2,\cdots,n$$

$$\cancel{x_i \in \{0,1\}}$$

$$x_i \ge 0,\quad x_i \le 1$$

不妨设 $$\frac{p_1}{s_1} \ge \frac{p_2}{s_2} \ge \cdots \ge \frac{p_n}{s_n}$$

LP的一个最优解为 $(1,\cdots,1,\alpha,0,\cdots,0)$

其中 $0<\alpha\le 1$, $$\sum_{i=1}^{k} s_i + \alpha\cdot s_{k+1} = C$$

Integrality Gap = 2

① $$\frac{w^*_{LP}}{w^*_{IP}} \le 2$$

IP的两个可行解:

1) 前k个物品

2) 第k+1个物品

② 考虑例子 $\left\{\left(\frac{C}{2}, C\right), \left(\frac{C+1}{2}, C\right)\right\}$

$$w^*_{LP} = C + \frac{C}{C+1}\cdot C$$

$$w^*_{IP} = C$$

二者之比为 $\frac{2C+1}{C+1} \to 2$

顶点覆盖问题

$$\min \ C = \sum_{i=1}^{n} w_i x_i$$

s.t. $x_i + x_j \geq 1, \quad (i,j) \in E$

$\cancel{x_i \in \{0,1\}}$ $i = 1, 2, \cdots, n$

$x_i \geq 0$

结论：上述LP的基本可行解是半整的，即 $x_i = \{0, \frac{1}{2}, 1\}$

证明：若∃基本可行解 X，其不是半整的。

$V_+ = \{ i \mid \frac{1}{2} < x_i < 1 \}$

$V_- = \{ i \mid 0 < x_i < \frac{1}{2} \}$

$\boxed{V_+ \cup V_- \text{ 非空}}$

对 $\varepsilon > 0$，构造两个解.

$$y_i = \begin{cases} x_i + \varepsilon, & i \in V_+ \\ x_i - \varepsilon, & i \in V_- \\ x_i, & \text{else} \end{cases} \qquad z_i = \begin{cases} x_i - \varepsilon, & i \in V_+ \\ x_i + \varepsilon, & i \in V_- \\ x_i, & \text{else} \end{cases}$$

$x = \frac{1}{2}(y + z)$, $\varepsilon > 0$ 足够小

保证 y, z 可行

Integrality Gap = ?

Ref.

$$\boxed{IG = 2 - \frac{2}{\chi^f(G)}}$$ ← 分数色数

Integrality gap of the vertex cover linear programming relaxation

Mohit Singh

Operations Research Letters 47, 288-290, 2019

Unrelated Machine Scheduling

问题描述：

给定
$$\begin{cases} J: \text{工件集合}, 1, \dots, n \\ M: \text{机器集合}, 1, \dots, m \\ P_{ij}: \text{工件 } j \text{ 在机器 } i \text{ 上的运行时间} \end{cases}$$

将J中的工件分配到M的机器上，使得机器的最大负载（完工时间）最小。

$$\boxed{C_{max} = \max_{i \in M} \sum_{j \to i} P_{ij}}$$

IP 模型

$$x_{ij} = \begin{cases} 1, & j \to i \\ 0, & \text{else} \end{cases}$$

$$\min \quad t$$

$$\text{s.t.} \quad \sum_{i \in M} x_{ij} = 1, \quad j \in J$$

$$\sum_{j \in J} x_{ij} P_{ij} \le t, \quad i \in M$$

$$x_{ij} \in \{0,1\}, \quad i \in M, j \in J$$

"IG" = ?　考虑仅有一个工件的例子

$$P_{i1} = m, \text{ 则 } \begin{cases} t^*_{IP} = m \\ t^*_{LP} = 1 \end{cases}$$

$$IG \ge m$$

猜一个目标函数值 T，令

$$\boxed{x_{ij}=0,\ 若\ p_{ij}>T,\ i\in M,\ j\in J}$$

是否可行？

$$\begin{cases} \sum x_{ij}=1, & j\in J \\ \sum x_{ij}p_{ij}\le T, & i\in M \\ x_{ij}\ge 0 \end{cases}$$

记为 $LP(T)$
$$\begin{cases} 可行, & T\downarrow \\ 不可行, & T\uparrow \end{cases}$$

$T\in[\hat{T}/m, \hat{T}]$

$\hat{T}$：将工件 j 分配到 p_{ij} 最小的机器让得到的最大负载

当 input 均为整数时，需要猜 $O(\log \hat{T})$ 步数

$LP(T)$ 可在多项式时间求得最优的 T^*.

那么，如何 rounding 呢？

先分析一下有多少个分数工件

$$\begin{cases} \sum_i x_{ij} = 1 & j = 1, \cdots, n \\ \sum_j x_{ij} p_{ij} \le T & i = 1, \cdots, m \end{cases}$$

$LP(T)$ 的基本可行解（极点）X

中正分量个数 $\le n+m$

约束方程个数

设 X 对应的整数工件数为 n_1

分数工件数为 n_2

$$\begin{cases} n_1 + n_2 = n \\ n_1 + 2n_2 \le n+m \end{cases} \Longrightarrow n_2 \le m$$

分数对应两个分数变量

最多只有 m 个工件是拆分的！

注意：每个工件 j，只要 $x_{ij} > 0$

那么 $p_{ij} \le T^*$

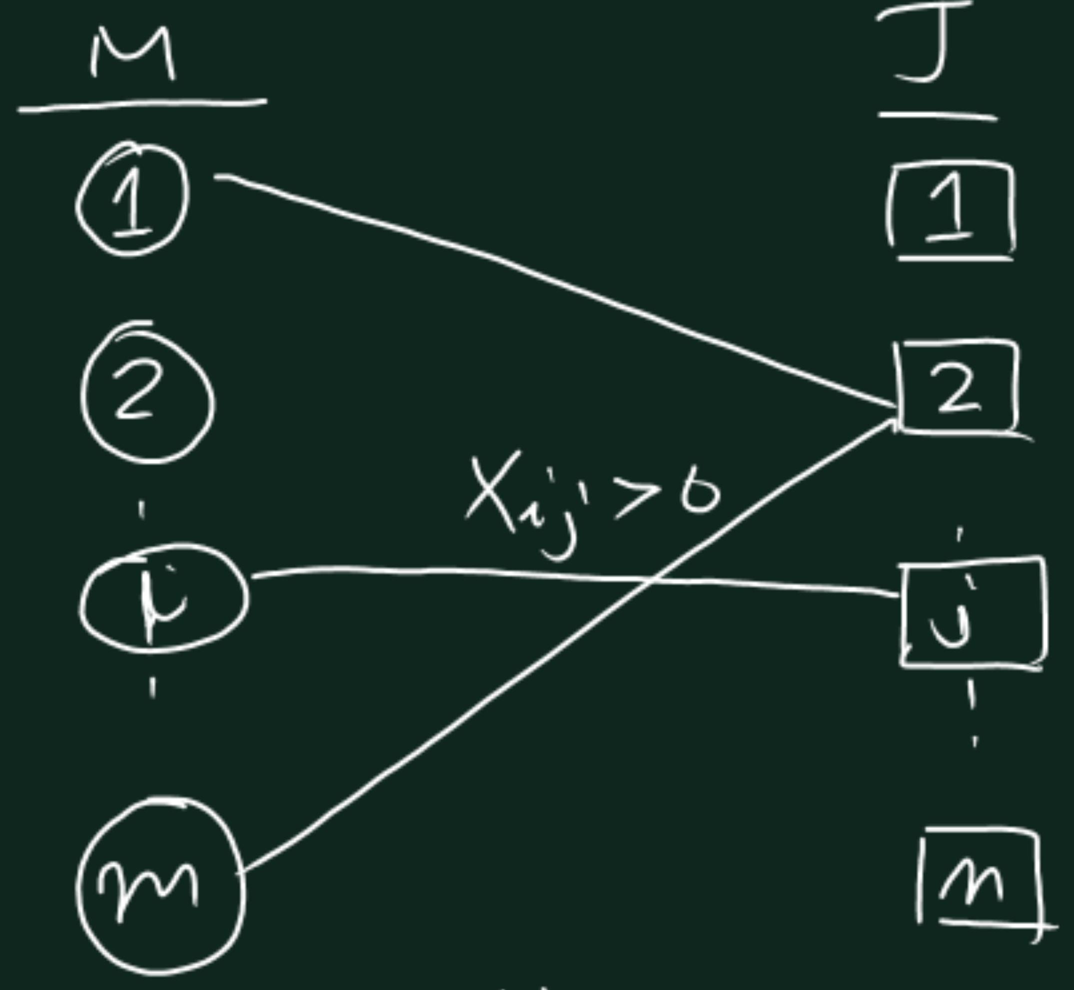
如果能够将这（至多）m 个拆分工件

匹配到 $x_{ij} > 0$ 的机器上，则可得到

$C_{max} \le 2T^*$

这样的匹配存在吗？如何构造？

考察 (J, M, E) 二部图 G



$(i,j)\in E \iff x_{ij}>0$

H：把 G 中**整工件**的顶点及其唯一关联边去掉而得的子图

G 有 $n+m$ 个顶点，至多 $n+m$ 条边

G 若连通，则为一棵**伪树**

G 若不连通，则其任一连通分支为一棵伪树。G 为**伪森林**

H 是伪森林

匹配方案

注意 H 是伪森林，其每一个连通分支为一棵伪树

若伪树不是一个圈，其叶子点一定是"机器"顶点 i，分配与其邻接的工件 j 到该机器，删去 i, j 及 j 关联边。

得到的仍是伪森林

如此操作，或者所有工件均已分配，或者剩下的伪树都是圈。后者很容易得到完美匹配

所有拆分的工件重新得到了完整分配。因为每个机器至多被分配一个这样的工件。

$C_{max} \leq T^* + T^* = 2T^*$