第九讲 贪心算法

回顾前面提到的组合优化问题：

(1) 最小生成树（最大权森林）

(2) 最短路径问题

(3) 背包问题

(4) 顶点覆盖、集合覆盖

(5) 旅行商问题

(6) 装箱问题

$\Longrightarrow$

E：基础元素集

c：$E \to R^+$

$\mathcal{F} \subseteq 2^E$

在 $\mathcal{F}$ 中寻找一个元素 X

使其权重最大或最小

其中：$\boxed{c(X) = \sum_{e \in X} c(e)}$

独立系统

定义：一个集合系统 $(E, \mathcal{F})$ 称为独立系统，若

(M_1) $\emptyset \in \mathcal{F}$

(M_2) 若 $Y \subseteq X \in \mathcal{F}$，则 $Y \in \mathcal{F}$ （关于包含封闭）

独立集：$\mathcal{F}$ 中的元素

基：极大独立集

相关集：$2^E \setminus \mathcal{F}$ 中的元素

圈：极小相关集

对 $X \subseteq E$，包含在 X 中的极大独立集称为 X 的基

- $B \in \mathcal{F}$，$B \subseteq X$
- X 中无独立集 $\supset B$

更多的定义

设 $(E, \mathcal{F})$ 为一独立系统，对 $X \subseteq E$，定义 X 的秩

$$r(X) = \max\{|Y| : Y \subseteq X, Y \in \mathcal{F}\}$$

（X 中最大独立集所含元素个数）

X 的下秩

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

（X 中最小的极大独立集所含元素个数）

$(E, \mathcal{F})$ 的秩商定义为：

$$q(E, \mathcal{F}) = \min_{X \subseteq E} \frac{\rho(X)}{r(X)}$$

$$\boxed{q(E, \mathcal{F}) \le 1}$$

两个基本优化问题

极大化问题

Input: 独立系统 $(E, \mathcal{I})$

$c: E \to R^+$

Output: 独立集 $I \in \mathcal{I}$，使得 $c(I)$ 最大

（权重最大的独立集）

极小化问题

Input: 独立系统 $(E, \mathcal{I})$

$c: E \to R^+$

Output: 基 B，使得 $c(B)$ 最小

（权重最小的极大独立集）

若干组合优化问题(I)

<1> 最小生成树问题

Input：连通无向图 $G=(V,E)$

$c: E(G) \longrightarrow R^+$

Output：权重最小的生成树

独立系统：$E = E(G)$

$c: E \longrightarrow R^+$

$\mathcal{F} = \{F \subseteq E \mid F \text{是森林}\}$

<2> 最短路问题

Input：有向(或无向图) $G=(V,E)$

$c: E(G) \longrightarrow R^+$

$s, t \in V(G)$

Output：一条 s-t 最短路

独立系统：$E = E(G)$

$c: E \rightarrow R^+$

$\mathcal{F} = \{F \subseteq E \mid F \text{是一条} s\text{-}t \text{路的边子集}\}$

若干组合优化问题(Ⅲ)

(3) 顶点覆盖问题

Input: 无向图 $G=(V,E)$

$w: V \to \mathbb{R}^+$

Output: 权重最小的顶点覆盖

独立系统: $E = V$

$w: E \to \mathbb{R}^+$

$\mathcal{F} = \{ U \subseteq V \mid U$ 是 G 的某个极小顶点覆盖的子集 $\}$

(4) 最大权匹配问题

Input: 无向图 $G=(V,E)$

$c: E(G) \to \mathbb{R}^+$

Output: 权重最大的边集，其任意两边无公共顶点

独立系统: $E = E(G)$

$c: E \to \mathbb{R}^+$

$\mathcal{F} = \{ M \subseteq E \mid M$ 中的边无公共顶点 $\}$

若干组合优化问题(Ⅲ)

(5) 装箱问题

Input: $\{a_1, \cdots, a_n\}$, $0 < a_i \le 1$,

若干单位容量的箱子

Output: Packing 使用最少数目的箱子

换一种说法:

设单个箱子的可行"装填方案"

为 M 个, $\{\underbrace{t_1, \cdots, t_M}_{\text{pattern}}\}$

任一个装箱的解由若干 patterns 组成，使得每一个 item 一定出现在某个选中的 pattern 中，该解记作 configuration

独立系统: $E = \{t_1, \cdots, t_M\}$

$c(t_i) = 1$, $i = 1, \cdots, M$

$\mathcal{F} = \{ P \mid P$是某个 configuration 的子集$\}$

课堂练习

写出下列问题的独立系统：

(6) 背包问题

(7) 旅行商问题

若干组合优化问题（IV）

(6) 背包问题

Input: 背包容量 C，n 个物品

$\{(S_i, V_i), i=1,\cdots,n\}$

Output: 物品子集 $I \subseteq \{1,2,\cdots,n\}$

使得 $\sum_{i\in I} S_i \le C$，$\sum_{i\in I} V_i$ 最大

独立系统：$E = \{1,2,\cdots,n\}$

$C: i\in E \longrightarrow V_i$

$\mathcal{F} = \{ I \subseteq \{1,2,\cdots,n\} \mid \sum_{i\in I} S_i \le C \}$

(7) TSP

Input: 无向图 $G=(V,E)$

$c: E(G) \to R^+$

Output: 权重最小的哈密顿圈

独立系统：$E = E(G)$

$c: E \to R^+$

$\mathcal{F} = \{ F \subseteq E \mid F$ 是某个 H-圈的子集 $\}$

贪心算法

极大化问题

Best-In（优胜）法

(1) $c(e_1) \geqslant c(e_2) \geqslant \cdots \geqslant c(e_n)$

(2) Set $F = \phi$

(3) For $i = 1$ to n do

If $\boxed{F \cup \{e_i\} \in \mathcal{F}}$ set

$F = F \cup \{e_i\}$

oracle

极小化问题

Worst-out（劣汰法）

(1) $c(e_1) \geqslant c(e_2) \geqslant \cdots \geqslant c(e_n)$

(2) Set $F = E$

(3) For $i = 1$ to n do

If $F \setminus \{e_i\}$ $\boxed{\text{含“基”}}$ 则

$F = F \setminus \{e_i\}$

oracle

极大化问题的Best In算法

定理1：（Jenkyns, 1976，Korte and Hausmann, 1978）

设 $(E, \mathcal{F})$ 为一个独立系统，$c: E \to R^+$，记

$G(E, \mathcal{F}, c)$ 为 Best-In 的目标函数值

$OPT(E, \mathcal{F}, c)$ 为最优目标函数值，则

$$\boxed{q(E, \mathcal{F}) \le \frac{G(E, \mathcal{F}, c)}{OPT(E, \mathcal{F}, c)} \le 1}$$，对 $\forall c$ 成立

且 $\exists c$ 使得下界可以达到（tight）.

证明: $E=\{e_1,e_2,\cdots,e_n\}$

$c(e_1)\geqslant c(e_2)\geqslant\cdots\geqslant c(e_n)\geqslant 0$

G_n: Best-In 的解

O_n: 最优解

$E_j=\{e_1,e_2,\cdots,e_j\}$

$G_j=G_n\cap E_j$

$O_j=O_n\cap E_j$

$j=1,2,\cdots,n$

则: $C(G_n)=\sum_{j=1}^{n}(|G_j|-|G_{j-1}|)c(e_j)$

$c(e_{n+1})=0$

$G_0=\phi$

$=\sum_{j=1}^{n}|G_j|(c(e_j)-c(e_{j+1}))$

G_j: E_j上的极大独立集

$\geqslant\sum_{j=1}^{n}\rho(E_j)(c(e_j)-c(e_{j+1}))$

$\geqslant g(E,\mathcal{F})\sum_{j=1}^{n}r(E_j)(c(e_j)-c(e_{j+1}))$

$\geqslant g(E,\mathcal{F})\sum_{j=1}^{n}|O_j|(c(e_j)-c(e_{j+1}))$

$=g(E,\mathcal{F})\sum_{j=1}^{n}(|O_j|-|O_{j-1}|)c(e_j)$

$=g(E,\mathcal{F})\cdot C(O_n)$

实例: 设$\mathcal{F}\subseteq E$上的两个基B_1,B_2

达到 $|B_1|/|B_2|=g(E,\mathcal{F})$

令 $c(e)=1,\ e\in F$; $c(e)=0,\ e\notin F$

* B_1的元素排在前面…

极大化问题的 Worst-Out 算法

定理2：（Korte and Monma, 1979）

设 (E, F) 是一个独立系统

$c: E \to R^+$

设 $G(E,F,c)$ 为 Worst-out 算法的目标函数值，$OPT(E,F,c)$ 为最优值

则

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

对 $\forall c$ 成立，其中 ρ^*, r^* 为对偶独立系统的下秩和秩函数. 且 $\exists c$ 使上界是紧的.

? 若将 Best-In 用于极小化问题

Worst-Out 用于极大化问题

考虑下面两个问题

(1) 极小顶点覆盖的最大权问题

权重最大的顶点覆盖 $\bar{V}$，$\bar{V}$ 去掉任何顶点将不再是顶点覆盖

(2) 极大"顶点独立集"的最小权问题

权重最小的"顶点独立集"，增加任何顶点都不再是独立集

独立系统的对偶

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

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

$(E,\mathcal{F}^*)$ 显然也是一个独立系统

结论：$(E,\mathcal{F}^{**}) = (E,\mathcal{F})$

证明：

$F \in \mathcal{F}^{**} \iff \exists\ (E,\mathcal{F}^*)$ 的基 B^*，$F \cap B^* = \phi$

$\iff \exists\ (E,\mathcal{F})$ 的基 B，

$$F \cap (E \setminus B) = \phi$$

$\iff \exists\ (E,\mathcal{F})$ 的基 B，$F \subseteq B$

$\iff F \in \mathcal{F}$

课堂讨论

- 背包问题的 Best-In 算法
  - 当所有物品的价值相同
  - 当所有物品的尺寸相同
- 最大权匹配问题的 Best-In 算法

- 最小生成树问题的 Worst-Out 算法

秩商的估计式 (I)

定理: (Hausmann, Jenkyns, Korte, 1980)

设 $(E, \mathcal{F})$ 为一独立系统。若 $\forall A \in \mathcal{F}$ and $e \in E$, $A \cup \{e\}$ 至多含 p 个圈, 则 $q(E, \mathcal{F}) \geq 1/p$

证明: $\forall F \subseteq E$, J, K 为 F 的两个基, 只需证 $\boxed{|J|/|K| \geq 1/p}$

设 $J \setminus K = \{e_1, \cdots, e_t\}$

构造 $\underbrace{\boxed{K_0 = K}, K_1, K_2, \cdots, K_t}_{J\cup K \text{ 的独立集}}$

对 $i = 0, 1, \cdots, t-1$

$K_i \cup \{e_{i+1}\}$ 至多含 $\underline{p \text{ 个圈}}$

含 $K_i \setminus J$ 中的元

故 $\exists X \subseteq K_i \setminus J$, $|X| \leq p$ 且

$$\underbrace{(K_i \setminus X) \cup \{e_{i+1}\}}_{\text{记为 } K_{i+1}} \in \mathcal{F}$$

可见 $J \subseteq K_t$, J 是 $F \supseteq J \cup K$ 的基

所以 $\boxed{J = K_t}$

$$|K \setminus J| = \sum_{i=1}^{t} |K_{i-1} \setminus K_i| \leq p \cdot t$$

$$= p|J \setminus K|$$

得证

利用前面的联merged估计式

再次分析如下问题

---

(1) 最小生成树 (最大权森林)

(2) 最大权匹配问题