贪心算法

贪心技术

- **A "greedy algorithm" is a "non-backtracking algorithm in which irrevocable decisions of global significance are made on the basis of local information".**

  ----G.B.McMahon

贪心技术

- **A “greedy algorithm” is a “non-backtracking algorithm in which irrevocable decisions of global significance are made on the basis of local information”.**

  ----G.B.McMahon

- 所做的每一步选择都必须满足：
  - 可行的：必须满足问题的约束
  - 局部最优：是当前步骤中所有可行性选择中最佳的局部选择
  - 不可取消：选择一旦做出，在算法的后面步骤中就无法改变了

贪心算法举例

- 部分背包问题
- 活动选择问题
- 赫夫曼编码问题（**Huffman** 算法）
- 最短路径问题（ **Dijkstra**算法）
- 最小生成树问题
  - **Kruskal**算法
  - **Prim**算法

主要内容

- 贪心算法的理论基础
  - 什么是拟阵（**Matroids**）？
  - 拟阵的通用贪心解法
  - 拟阵通用贪心解法的正确性
  - 基于拟阵的贪心算法举例

线性相关与线性无关

- 线性相关的两个性质
  - 如果X＝$\{x_1,x_2,\ldots,x_k\}$是一个线性无关向量组，则对X的任意子集X'也是线性无关的。

线性相关与线性无关

- 线性相关的两个性质
  - 如果$X=\{x_1,x_2,...,x_k\}$是一个线性无关向量组，则对X的任意子集X’也是线性无关的。
  - 如果 $X=\{x_1, x_2,..., x_r\}$和 $Y=\{y_1, y_2,..., y_m\}$是两个线性无关向量组且$m>r$，则必存在一个 $y_i \in Y$，使得$X \cup \{y_i\}$是一个线性无关向量组。

线性相关与线性无关

- 线性相关的两个性质

  - ✓ 如果$X=\{x_1,x_2,\ldots,x_k\}$是一个线性无关向量组，则对X的任意子集X’也是线性无关的。

  - ✓ 如果 $X=\{x_1, x_2,\ldots, x_r\}$和 $Y=\{y_1, y_2,\ldots, y_m\}$是两个线性无关向量组且$m>r$，则必存在一个 $y_i\in Y$，使得$X\cup\{y_i\}$是一个线性无关向量组。

1935年， 美国数学家哈斯勒·惠特尼（Hassler Whitney）把以上两条性质进行了抽象推广，提出了拟阵概念。

拟 阵

- 一个**拟阵**是一个满足如下条件的**有序对** $\mathbf{M = (S, I)}$

拟 阵

- 一个拟阵是一个满足如下条件的有序对 $M = (S, I)$

  (1) S是非空有限集;

拟 阵

- 一个拟阵是一个满足如下条件的有序对 M = (S, I)

(1) S是非空有限集；

(2) I 是 S 的子集的一个非空族，这些子集称为S的独立子集，即 I 满足遗传性质：若B∈I，且A⊆B，则A ∈I ；

拟 阵

- 一个**拟阵**是一个满足如下条件的有序对 $\mathbf{M = (S, I)}$

  (1) S是非空有限集；

  (2) I 是 S 的子集的一个非空族，这些子集称为S的独立子集，即 I 满足遗传性质：若$B \in I$，且$A \subseteq B$，则$A \in I$；

  （即：若B是S的独立子集，则B的任意子集均是S的独立子集）

拟 阵

- 一个拟阵是一个满足如下条件的有序对 $M = (S, I)$

  (1) S是非空有限集;

  (2) I 是 S 的子集的一个非空族，这些子集称为S的独立子集，即 I 满足遗传性质：若$B \in I$，且$A \subseteq B$，则$A \in I$；

  （即：若B是S的独立子集，则B的任意子集均是S的独立子集）

  (3) I 满足交换性质，即若$A \in I$, $B \in I$且 $|A| < |B|$，则存在某个元素 $x \in B - A$，使得$A \cup \{x\} \in I$。

拟 阵

- 一个拟阵是一个满足如下条件的有序对 $M = (S, I)$

(1) S是非空有限集；

(2) I 是 S 的子集的一个非空族，这些子集称为S的独立子集，即 I 满足遗传性质：若$B\in I$，且$A\subseteq B$，则$A \in I$；（即：若B是S的独立子集，则B的任意子集均是S的独立子集）

(3) I 满足交换性质，即若$A\in I$, $B\in I$且 $|A| < |B|$，则存在某个元素 $x \in B - A$，使得$A\cup\{x\}\in I$。

例（矩阵拟阵）：给定实数域上的矩阵 T，令

- S是T的列的集合，且
- $A\in I$ 当且仅当 A 中的列是线性无关的，

则 (S, I) 是一个拟阵。

例（图拟阵）：给定无向图$\mathbf{G = (V, E} \neq \emptyset\mathbf{)}$，定义

例（图拟阵）：给定无向图$G = (V, E \neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，

例（图拟阵）：给定无向图$G = (V, E \neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A = (V, A)$ 构成一个森林。

例（图拟阵）：给定无向图$G = (V, E \neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A = (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

例（图拟阵）：给定无向图$G = (V, E \neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A = (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

证明：$M_G = (S_G, I_G)$ 满足拟阵的3个条件。

例（图拟阵）：给定无向图$G = (V, E \neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A = (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

证明：$M_G = (S_G, I_G)$ 满足拟阵的3个条件。

(1) 因为 S_G 为图G的边集，显然非空；

例（图拟阵）：给定无向图$G=(V, E\neq\emptyset)$，定义

- S_G 为图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图$G_A=(V, A)$ 构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

证明：$M_G=(S_G, I_G)$ 满足拟阵的3个条件。

(1) 因为 S_G 为图G的边集，显然非空；

(2) I_G满足遗传性质：

例（图拟阵）：给定无向图$G=(V, E\neq\emptyset)$，定义

- S_G为图G的边集E，
- I_G是G的无循环边集的非空族：如果A是E的子集，则$A\in I_G$当且仅当A是无圈的，即图$G_A=(V, A)$构成一个森林。

则$M_G=(S_G, I_G)$是一个拟阵。

证明：$M_G=(S_G, I_G)$满足拟阵的3个条件。

(1) 因为S_G为图G的边集，显然非空；

(2) I_G满足遗传性质：由于从S_G的一个无循环边集中去掉若干条边不会产生循环，即森林的子集还是森林，因此S_G的无循环边集族I_G具有遗传性质。

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G为图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A=(V, A)$构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图$G_A= (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A = (V, A)$ 和 $G_B = (V, B)$ 是图G的两个森林，且$|B|>|A|$。

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A= (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A = (V, A)$ 和 $G_B = (V, B)$ 是图G的两个森林，且$|B|>|A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A= (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A = (V, A)$ 和 $G_B = (V, B)$ 是图G的两个森林，且$|B|>|A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

（假设森林 $F = (V_F, E_F)$包含了 t 棵树，其中第 i 棵树包含 v_i个顶点和 e_i 条边，则有

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图$G_A=(V, A)$ 构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A=(V, A)$ 和 $G_B=(V, B)$ 是图G的两个森林，且$|B|>|A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

（假设森林 $F=(V_F, E_F)$包含了 t 棵树，其中第 i 棵树包含 v_i个顶点和 e_i条边，则有

$$|E_F|=\sum_{i=1}^{t} e_i=\sum_{i=1}^{t}(v_i-1)=\sum_{i=1}^{t} v_i - t=|V_F| - t$$

得，$t=|V_F|-|E_F|$。）

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图$G_A=(V, A)$ 构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A=(V, A)$ 和 $G_B=(V, B)$ 是图G的两个森林，且$|B|>|A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

可得G_B中存在一棵树 T，它的顶点分别在森林 G_A的两棵树中。

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为 图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A \in I_G$ 当且仅当 A 是无圈的，即图$G_A= (V, A)$ 构成一个森林。

则 $M_G = (S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A = (V, A)$ 和 $G_B = (V, B)$ 是图G的两个森林，且$|B| > |A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

可得G_B中存在一棵树 T，它的顶点分别在森林 G_A的两棵树中。

由于T是连通的，故 T 中必有一条边 (u, v)满足u, v分别在G_A的两棵树中。因此将 (u, v) 加入 A 不会产生循环。

例（图拟阵）：给定无向图$G=(V, E\neq \emptyset)$，定义

- S_G 为图 G 的边集 E，
- I_G是G的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图$G_A=(V, A)$ 构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

(3) I_G 满足交换性质：设 $G_A=(V, A)$ 和 $G_B=(V, B)$ 是图G的两个森林，且$|B|>|A|$。

由于有 k 条边的森林恰由 n-k 棵树组成，其中n是该森林的节点数，因此 G_B中的树比G_A中的少。

可得G_B中存在一棵树 T，它的顶点分别在森林 G_A的两棵树中。

由于T是连通的，故 T 中必有一条边 (u, v)满足u, v分别在A的两棵树中。因此将 (u, v) 加入 A 不会产生循环。

所以 I_G 满足 交换性质。

综上所述， $M_G=(S_G, I_G)$ 是一个拟阵。

拟阵的重要概念和性质（1）

- 给定拟阵 $M = (S, I)$，对于 I 中的独立子集 $A \in I$，

拟阵的重要概念和性质（1）

- 给定拟阵 $M = (S, I)$，对于 I 中的独立子集 $A \in I$，若 S 有一元素 $x \notin A$，使得将 x 加入 A 后仍保持独立性，即 $A \cup \{x\} \in I$，则称 x 为 A 的一个扩展。

拟阵的重要概念和性质（1）

- 给定拟阵 $M = (S, I)$，对于 I 中的独立子集 $A \in I$，若 S 有一元素 $x \notin A$，使得将 x 加入 A后仍保持独立性，即 $A \cup \{x\} \in I$，则称 x为A的一个扩展。

例：对于图拟阵 M_G， 如果 A是一个边独立集(森林)，则边 e 是 A 的一个扩展当且仅当 e不在A中且将e加入A中不会形成圈。

拟阵的重要概念和性质（1）

- 给定拟阵 M = (S, I)，对于 I 中的独立子集 A∈ I，若 S 有一元素 x∉ A，使得将 x 加入 A后仍保持独立性，即 A∪{x} ∈ I，则称 x为A的一个扩展。

例：对于图拟阵 M_G， 如果 A是一个边独立集(森林)，则边 e 是 A 的一个扩展当且仅当 e不在A中且将e加入A中不会形成圈。

- 对拟阵 M中的一个独立子集 A，如果A不存在扩展，则称A是最大独立子集，或拟阵的基。

拟阵的重要概念和性质（1）

- 给定拟阵 M = (S, I)，对于 I 中的独立子集 A∈ I，若 S 有一元素 x∉ A，使得将 x 加入 A后仍保持独立性，即 A∪{x} ∈ I，则称 x为A的一个扩展。

例：对于图拟阵 M_G， 如果 A是一个边独立集(森林)，则边 e 是 A 的一个扩展当且仅当 e不在A中且将e加入A中不会形成圈。

- 对拟阵 M中的一个独立子集 A，如果A不存在扩展，则称A是最大独立子集，或拟阵的基。

定理 1 拟阵 M 中所有最大独立子集具有相同大小。

定理 1　拟阵 M 中所有最大独立子集具有相同大小。

定理 1 拟阵 M 中所有最大独立子集具有相同大小。

证明：(反证法) 假设 A，B是M的两个最大独立子集，且|A|<|B|。

定理 1 拟阵 M 中所有最大独立子集具有相同大小。

证明：(反证法) 假设 A，B是M的两个最大独立子集，且$|A|<|B|$。

则由交换性质可知，存在 $x \in B-A$，使得 $A \cup \{x\}$是一个独立子集，与A是最大独立子集矛盾。

定理 1 拟阵 M 中所有最大独立子集具有相同大小。

证明：(反证法) 假设 A，B是M的两个最大独立子集，且|A|<|B|。

则由交换性质可知，存在 $x \in B-A$，使得 $A \cup \{x\}$是一个独立子集，与A是最大独立子集矛盾。

例： 假设G是一个连通无向图。考虑G的图拟阵 M_G。

定理 1 拟阵 M 中所有最大独立子集具有相同大小。

证明：(反证法) 假设 A，B是M的两个最大独立子集，且|A|<|B|。

则由交换性质可知，存在 $x \in B-A$，使得 $A\cup\{x\}$是一个独立子集，与A是最大独立子集矛盾。

例： 假设G是一个连通无向图。考虑G的图拟阵 M_G。则 M_G 的最大独立子集必定是一棵边数为|V|-1，连接了G的所有顶点的树，即G的生成树

拟阵的重要概念和性质（2）

- **加权矩阵：** 如果一个拟阵 **M = (S, I)** 关联一个加权函数**w**，使得对于任意**x**∈ **S**，有**w(x)>0**，则称拟阵 **M** 为加权拟阵。

拟阵的重要概念和性质（2）

- **加权矩阵：** 如果一个拟阵 $\mathbf{M = (S, I)}$ 关联一个加权函数 $\mathbf{w}$，使得对于任意 $\mathbf{x \in S}$，有 $\mathbf{w(x)>0}$，则称拟阵 $\mathbf{M}$ 为加权拟阵。

- 通过求和，可将权重函数 $\mathbf{w}$ 扩展到 $\mathbf{S}$ 的任意子集$\mathbf{A}$:

$$\mathbf{w(A) = \sum_{x \in A} w(x)}$$

拟阵的重要概念和性质（2）

- **加权矩阵**：如果一个拟阵 $M = (S, I)$ 关联一个加权函数w，使得对于任意$x \in S$，有$w(x)>0$，则称拟阵 M 为加权拟阵。

- 通过求和，可将权重函数 w 扩展到 S 的任意子集A:

$$w(A) = \sum_{x \in A} w(x)$$

例： 考虑G的图拟阵 M_G。如果以w(e)表示G中边e的长度，则w(A) 为边集A中所有边长的总长度。

主要内容

- 什么是拟阵(**Matroids**)?
- 拟阵的通用贪心解法
- 拟阵通用贪心解法的正确性
- 基于拟阵的贪心算法举例

加权拟阵上的贪心算法

- 许多用贪心算法求解的问题可以表示为加权拟阵的最大权独立子集问题：

加权拟阵上的贪心算法

- 许多用贪心算法求解的问题可以表示为加权拟阵的最大权独立子集问题：

给定加权拟阵 **M = (S, I)**，计算 **S** 的具有最大权值 **w(A)**的独立子集 **A∈I**，称为拟阵**M**的最优子集。

加权拟阵上的贪心算法

- 许多用贪心算法求解的问题可以表示为加权拟阵的最大权独立子集问题：

给定加权拟阵 $M = (S, I)$，计算 S 的具有最大权值 $w(A)$的独立子集 $A \in I$，称为拟阵M的最优子集。

- 由于S中任一元素x的权W(x)是正的，因此，最优子集也一定是最大独立子集 (不存在可扩展元素)。

加权拟阵上的贪心算法

- 许多用贪心算法求解的问题可以表示为加权拟阵的最大权独立子集问题：

给定加权拟阵 $M = (S, I)$，计算 S 的具有最大权值 $w(A)$的独立子集 $A \in I$，称为拟阵M的最优子集。

- 由于S中任一元素x的权W(x)是正的，因此，最优子集也一定是最大独立子集 (不存在可扩展元素)。

问题：如何计算最大独立子集？

加权拟阵上的贪心算法

- 许多用贪心算法求解的问题可以表示为加权拟阵的最大权独立子集问题：

给定加权拟阵 M = (S, I)，计算 S 的具有最大权值 w(A)的独立子集 $A\in I$，称为拟阵M的最优子集。

- 由于S中任一元素x的权W(x)是正的，因此，最优子集也一定是最大独立子集 (不存在可扩展元素)。

问题：如何计算最大独立子集？

- ✓ 独立子集的扩展

加权拟阵最优子集的贪心算法框架如下：

输入：具有正权函数 W 的加权拟阵M = (S, I)

输出：M的最优子集A。

```
GREEDY(M, w)
    A = ∅
    sort M.S into monotonically decreasing order by weight w
    for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
        if A ∪ {x} ∈ M.I
            A = A ∪ {x}
    return A
```

加权拟阵最优子集的贪心算法框架如下：

输入：具有正权函数 W 的加权拟阵M = (S, I)

输出：M的最优子集A。

nlog n

GREEDY(M, w)

```
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

加权拟阵最优子集的贪心算法框架如下：

输入：具有正权函数 W 的加权拟阵**M = (S, I)**

输出：M的最优子集**A**。

GREEDY(M, w)

```
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.𝓘
        A = A ∪ {x}
return A
```

nlog n

nf(n)

加权拟阵最优子集的贪心算法框架如下：

输入：具有正权函数 W 的加权拟阵$M = (S, I)$

输出：M的最优子集A。

```
GREEDY(M, w)
  A = ∅
  sort M.S into monotonically decreasing order by weight w
  for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
      if A ∪ {x} ∈ M.𝓘
          A = A ∪ {x}
  return A
```

- 复杂度：$O(n\log n + nf(n))$，其中
  - $n = |S|$
  - $O(f(n))$为每次检查 $A \cup \{x\}$ 是否为独立子集的时间

主要内容

- 什么是拟阵(**Matroids**)?
- 拟阵的通用贪心解法
- 拟阵通用贪心解法的正确性
- 基于拟阵的贪心算法举例

GREEDY算法的正确性

定理 2 如果 $M=(S, I)$ 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

GREEDY(M, w)

```
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

- 第一个选中的独立子集 **{x}**一定会包含于一个最优子集**A**中

GREEDY算法的正确性

定理 2　如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

GREEDY(M, w)

$A = \emptyset$
sort $M.S$ into monotonically decreasing order by weight w
for each $x \in M.S$, taken in monotonically decreasing order by weight $w(x)$
　　if $A \cup \{x\} \in M.\mathcal{I}$
　　　　$A = A \cup \{x\}$
return A

- 第一个选中的独立子集 {x}一定会包含于一个最优子集A中
- 第一个被舍弃的元素，永远不可能用于构造最优子集

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

- 第一个选中的独立子集 {x}一定会包含于一个最优子集A中
- 第一个被舍弃的元素，永远不可能用于构造最优子集
- 归纳利用以上两个结论

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 x∈ S是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 x∈ A。

引理 **1** （拟阵的贪心选择性质）设**M=(S, I)** 是具有正权函数**w**的带权拟阵，且**S**中元素依权值从大到小排列。又设 $x \in S$是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x \in A$。

证明: (1) 若不存在$x \in S$，使得$\{x\}$是独立子集，则引理1是平凡的。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

证明: (1) 若不存在$x\in S$，使得$\{x\}$是独立子集，则引理是平凡的。

(2) 设B是一个非空的最优子集。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in$ S是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x\in$ A。

证明: (1) 若不存在$x\in$S，使得{x}是独立子集，则引理是平凡的。

(2) 设B是一个非空的最优子集。

由于$B\in$I，且I具有遗传性质，故B中所有单个元素子集 {y} 均为独立子集。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

证明: (1) 若不存在$x\in S$，使得{x}是独立子集，则引理是平凡的。

(2) 设B是一个非空的最优子集。

由于$B\in I$，且I具有遗传性质，故B中所有单个元素子集 {y} 均为独立子集。

又由于 x 是S中的第一个单元素独立子集，故对任意的$y\in B$，均有：$W(x) \geq W(y)$。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in$ S是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x\in$ A。

证明: (1) 若不存在$x\in$S，使得{x}是独立子集，则引理是平凡的。

(2) 设B是一个非空的最优子集。

由于B$\in$I，且I具有遗传性质，故B中所有单个元素子集 {y} 均为独立子集。

又由于 x 是S中的第一个单元素独立子集，故对任意的y$\in$B，均有：$W(x) \geq W(y)$。

(a)若$x\in$B，则只要令A=B，定理得证；

引理 1 （<u>拟阵的贪心选择性质</u>）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x \in S$是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x \in A$。

(b) 若$x \notin B$，按如下方法构造包含元素x的最优子集A。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in$ S是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x\in$ A。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设A={x}，此时A是一个独立子集。

若|B|=|A|=1，则定理得证。

引理 1 （拟阵的贪心选择性质）设$M=(S, I)$是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设$A=\{x\}$，此时A是一个独立子集。

若$|B|=|A|=1$，则定理得证。

(ii) 若$|B|>|A|$，反复利用拟阵M的交换性质，从B中选择一个新元素加入A中并保持A的独立性，直到$|A|=|B|$。

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in$ S是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x\in$ A。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设$A=\{x\}$，此时A是一个独立子集。

若$|B|=|A|=1$，则定理得证。

(ii) 若$|B|>|A|$，反复利用拟阵M的交换性质，从B中选择一个新元素加入A中并保持A的独立性，直到$|A|=|B|$。

此时必有一元素 $y\in B$且$y\notin A$,使得$A=B-\{y\}\cup\{x\}$且$W(x)\geq W(y)$，

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设$A=\{x\}$，此时A是一个独立子集。

若$|B|=|A|=1$，则定理得证。

(ii) 若$|B|>|A|$，反复利用拟阵M的交换性质，从B中选择一个新元素加入A中并保持A的独立性，直到$|A|=|B|$。

此时必有一元素 $y\in B$且$y\notin A$,使得$A=B-\{y\}\cup\{x\}$且$W(x)\geq W(y)$，且满足：$W(A) = W(B)-W(y)+W(x) \geq W(B)$

引理 1 （拟阵的贪心选择性质）设M=(S, I) 是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 {x} 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设A={x}，此时A是一个独立子集。

若|B|=|A|=1，则定理得证。

(ii) 若|B|>|A|，反复利用拟阵M的交换性质，从B中选择一个新元素加入A中并保持A的独立性，直到|A|=|B|。

此时必有一元素 $y\in B$且$y\notin A$,使得$A=B-\{y\}\cup\{x\}$且$W(x)\geq W(y)$，

且满足：$W(A) = W(B)-W(y)+W(x) \geq W(B)$

由于B是一个最优子集，所以$W(B) \geq W(A)$。

引理 1 （拟阵的贪心选择性质）设$M=(S, I)$是具有正权函数w的带权拟阵，且S中元素依权值从大到小排列。又设 $x\in S$是 S 中第一个使得 $\{x\}$ 是独立子集的元素，则存在S的最优子集 A 使得 $x\in A$。

(b) 若$x\notin B$，按如下方法构造包含元素x的最优子集A。

(i) 首先，设$A=\{x\}$，此时A是一个独立子集。

若$|B|=|A|=1$，则定理得证。

(ii) 若$|B|>|A|$，反复利用拟阵M的交换性质，从B中选择一个新元素加入A中并保持A的独立性，直到$|A|=|B|$。

此时必有一元素 $y\in B$且$y\notin A$,使得$A=B-\{y\}\cup\{x\}$且$W(x)\geq W(y)$，

且满足：$W(A) = W(B)-W(y)+W(x) \geq W(B)$

由于B是一个最优子集，所以$W(B) \geq W(A)$。

因此$W(A)= W(B)$，即A也是一个最优子集，且$x\in A$。

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

- 第一个选中的独立子集 {x}一定会包含于一个最优子集A中
- 第一个被舍弃的元素，以后也永远不可能用于构造最优子集
- 归纳利用以上两个结论

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

- 第一个选中的独立子集 {x}一定会包含于一个最优子集A中
- 第一个被舍弃的元素，以后也永远不可能用于构造最优子集
- 归纳利用以上两个结论

引理2：设M = (S, I) 是拟阵。若S中元素 x 不是空集Φ的可扩展元素，则 x 也不可能是 S中任一独立子集A的可扩展元素。

引理2：设M = (S, I) 是拟阵。若S中元素 x 不是空集Φ的可扩展元素，则 x 也不可能是 S中任一独立子集A的可扩展元素。

证明：(反证法) 设 x∈S不是Φ的一个可扩展元素，但它是 S 的某独立子集A的一个可扩展元素，即

$$A \cup \{x\} \in I。$$

引理2：设$M = (S, I)$ 是拟阵。若S中元素 x 不是空集Φ的可扩展元素，则 x 也不可能是 S中任一独立子集A的可扩展元素。

证明：(反证法) 设 $x \in S$不是Φ的一个可扩展元素，但它是 S 的某独立子集A的一个可扩展元素，即

$$A \cup \{x\} \in I。$$

由 I 的遗传性质可知 $\{x\}$ 是独立的。

引理2：设$M = (S, I)$是拟阵。若S中元素 x 不是空集Φ的可扩展元素，则 x 也不可能是 S中任一独立子集A的可扩展元素。

证明：(反证法) 设 $x \in S$不是Φ的一个可扩展元素，但它是 S 的某独立子集A的一个可扩展元素，即

$$A \cup \{x\} \in I。$$

由 I 的遗传性质可知 {x} 是独立的。

这与 x 不是空集 Φ 的一个可扩展元素相矛盾。

引理2：设M = (S, I) 是拟阵。若S中元素 x 不是空集Φ的可扩展元素，则 x 也不可能是 S中任一独立子集A的可扩展元素。

证明：(反证法) 设 x∈S不是Φ的一个可扩展元素，但它是 S 的某独立子集A的一个可扩展元素，即

$$A \cup \{x\} \in I。$$

由 I 的遗传性质可知 {x} 是独立的。这与 x 不是空集 Φ 的一个可扩展元素相矛盾。

- 算法Greedy在初始化独立子集A时舍弃的元素可以永远舍弃。

GREEDY算法的正确性

定理 2 如果 M＝(S, I) 是具有正权函数 w 的加权拟阵，则调用GREEDY(M, w) 返回一个最优子集。

- 第一个选中的独立子集 {x}一定会包含于一个最优子集A中
- 第一个被舍弃的元素，以后也永远不可能用于构造最优子集
- 归纳利用以上两个结论

引理3 （拟阵的最优子结构性质）

引理3 （拟阵的最优子结构性质）设 **x** 是求带权拟阵**M=(S, I)**最优子集的贪心算法**GREEDY**所选择的**S**中的第一个元素。

引理3（拟阵的最优子结构性质） 设 **x** 是求带权拟阵**M=(S, I)**最优子集的贪心算法**GREEDY**所选择的**S**中的第一个元素。那么，原问题可简化为**求带权拟阵M'= (S', I')的最优子集问题**，

引理3（拟阵的最优子结构性质）设 x 是求带权拟阵$M=(S, I)$最优子集的贪心算法GREEDY所选择的S中的第一个元素。那么，原问题可简化为求带权拟阵$M'= (S', I')$的最优子集问题，其中：

- $S'=\{y \mid y\in S$且$\{x, y\} \in I\}$，即y是x的可扩展元素}
- $I' = \{B \mid B \subseteq S-\{x\}$ 且 $B\cup\{x\} \in I \}$
- M'的权函数是M的权函数在S'上的限制（称M'为M关于元素x的收缩）。

引理3（拟阵的最优子结构性质）设 x 是求带权拟阵 $M=(S, I)$ 最优子集的贪心算法GREEDY所选择的S中的第一个元素。那么，原问题可简化为求带权拟阵 $M'=(S', I')$ 的最优子集问题，其中：

- $S'=\{y \mid y\in S$ 且 $\{x, y\} \in I\}$，即y是x的可扩展元素}
- $I' = \{B \mid B \subseteq S-\{x\}$ 且 $B\cup\{x\} \in I\}$
- M'的权函数是M的权函数在S'上的限制（称M'为M关于元素x的收缩）。

证明：首先，由M'的定义可得：若A是M的包含元素 x的最大独立子集，则 $A'=A-\{x\}$ 是M'的一个独立子集。

引理3（拟阵的最优子结构性质）设 x 是求带权拟阵M=(S, I)最优子集的贪心算法GREEDY所选择的S中的第一个元素。那么，原问题可简化为求带权拟阵M’= (S’, I’)的最优子集问题，其中：

- S’={y | y∈ S且{x, y} ∈ I，即y是x的可扩展元素}
- I’ = {B | B ⊆ S-{x} 且 B∪{x} ∈ I }
- M’的权函数是M的权函数在S’上的限制（称M’为M关于元素x的收缩）。

证明：首先，由M’的定义可得：若A是M的包含元素 x的最大独立子集，则A’=A-{x}是M’的一个独立子集。反之，M’的任一独立子集A’产生M的一个独立子集A=A’∪{x} 。

引理3（拟阵的最优子结构性质）设 x 是求带权拟阵 $M=(S, I)$ 最优子集的贪心算法GREEDY所选择的S中的第一个元素。那么，原问题可简化为求带权拟阵 $M'=(S', I')$ 的最优子集问题，其中：

- $S'=\{y \mid y\in S$ 且 $\{x, y\} \in I\}$，即y是x的可扩展元素}
- $I' = \{B \mid B \subseteq S-\{x\}$ 且 $B\cup\{x\} \in I \}$
- M'的权函数是M的权函数在S'上的限制（称M'为M关于元素x的收缩）。

证明：首先，由M'的定义可得：若A是M的包含元素 x的最大独立子集，则 $A'=A-\{x\}$ 是M'的一个独立子集。反之，M'的任一独立子集A'产生M的一个独立子集 $A=A'\cup\{x\}$。

(2)在这两种情形下均有：$W(A)=W(A')+W(x)$。

引理3（拟阵的最优子结构性质）设 x 是求带权拟阵$M=(S, I)$最优子集的贪心算法GREEDY所选择的S中的第一个元素。那么，原问题可简化为求带权拟阵$M'=(S', I')$的最优子集问题，其中：

- $S'=\{y \mid y\in S$且$\{x, y\} \in I$，即y是x的可扩展元素$\}$
- $I' = \{B \mid B \subseteq S-\{x\}$ 且 $B\cup\{x\} \in I \}$
- M'的权函数是M的权函数在S'上的限制（称M'为M关于元素x的收缩）。

证明：首先，由M'的定义可得：若A是M的包含元素 x的最大独立子集，则$A'=A-\{x\}$是M'的一个独立子集。反之，M'的任一独立子集A'产生M的一个独立子集$A=A'\cup\{x\}$。

(2)在这两种情形下均有：$W(A)=W(A')+W(x)$。

因此M的包含元素x的最优子集包含M'的一个最优子集，反之亦然。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：(1)由引理1可知，如第一次加入A的元素是x，则必存在包含元素x的一个最优子集。因此，Greedy第一次选择是正确的。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：**(1)**由引理**1**可知，如第一次加入**A**的元素是**x**，则必存在包含元素**x**的一个最优子集。因此，**Greedy**第一次选择是正确的。

(2)由引理**2**可知，选择 **x** 时被舍弃的元素不可能被再选中，即它们不可能是任一最优子集中的元素。因此，这些元素可以永远舍弃。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：(1)由引理1可知，如第一次加入A的元素是x，则必存在包含元素x的一个最优子集。因此，Greedy第一次选择是正确的。

(2)由引理2可知，选择 x 时被舍弃的元素不可能被再选中，即它们不可能是任一最优子集中的元素。因此，这些元素可以永远舍弃。

(3) 由引理3可知，Greedy选择了元素x后，原问题简化为求拟阵M’的最优子集问题。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：(1)由引理1可知，如第一次加入A的元素是x，则必存在包含元素x的一个最优子集。因此，Greedy第一次选择是正确的。

(2)由引理2可知，选择 x 时被舍弃的元素不可能被再选中，即它们不可能是任一最优子集中的元素。因此，这些元素可以永远舍弃。

(3) 由引理3可知，Greedy选择了元素x后，原问题简化为求拟阵M’的最优子集问题。

由于对 M’=(S’,I’)中的任一独立子集B∈I’，均有B∪{x}在M中是独立的（由M’的定义可知）。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：(1)由引理1可知，如第一次加入A的元素是x，则必存在包含元素x的一个最优子集。因此，Greedy第一次选择是正确的。

(2)由引理2可知，选择 x 时被舍弃的元素不可能被再选中，即它们不可能是任一最优子集中的元素。因此，这些元素可以永远舍弃。

(3) 由引理3可知，Greedy选择了元素x后，原问题简化为求拟阵M’的最优子集问题。

由于对 M’=(S’,I’)中的任一独立子集B∈I’，均有B∪{x}在M中是独立的（由M’的定义可知）。因此，Greedy选择了元素x后，后续求解将演变为求拟阵M’=(S’,I’)的最优子集问题。

定理2（加权拟阵贪心算法的正确性）设M＝(S, I)是具有正权函数w的加权拟阵，算法greedy返回M的最优子集

证明：(1)由引理1可知，如第一次加入A的元素是x，则必存在包含元素x的一个最优子集。因此，Greedy第一次选择是正确的。

(2)由引理2可知，选择 x 时被舍弃的元素不可能被再选中，即它们不可能是任一最优子集中的元素。因此，这些元素可以永远舍弃。

(3) 由引理3可知，Greedy选择了元素x后，原问题简化为求拟阵M’的最优子集问题。

由于对 M’=(S’,I’)中的任一独立子集B∈I’，均有B∪{x}在M中是独立的（由M’的定义可知）。因此，Greedy选择了元素x后，后续求解将演变为求拟阵M’=(S’,I’)的最优子集问题。

由归纳法可知：其后继步骤求出M’的一个最优子集，从而算法Greedy最终求出的是M的一个最优子集。

加权拟阵最优子集的贪心算法框架如下：

输入： 具有正权函数 **W** 的加权拟阵**M = (S, I)**

输出： **M**的最优子集**A**。

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

- 复杂度：**O(nlog n + nf(n))**，其中
  - **n = |S|**，
  - **O(f(n))**为每次检查 A∪{x} 是否为独立子集的时间。

主要内容

- 什么是拟阵(**Matroids**)?
- 拟阵的通用贪心解法
- 拟阵通用贪心解法的正确性
- 基于拟阵的贪心算法举例
  - 最小生成树问题
  - 单位时间任务调度问题

最小生成树问题

输入：无向连通图 $G = (V, E)$ 及边权函数 w，使得G中的每一条边 $(u,v)\in E$ 有权值 $w(u,v)$。

输出：G的最小生成树T，即边权和最小的生成树。

最小生成树问题

输入：无向连通图 **G = (V, E)** 及边权函数 **w**，使得**G**中的每一条边 **(u,v)∈E** 有权值 **w(u,v)**。

输出：**G**的最小生成树**T**，即边权和最小的生成树。

- 可转化为一个拟阵中找出最大独立子集的问题

最小生成树问题

输入：无向连通图 $G=(V, E)$ 及边权函数 w，使得G中的每一条边 $(u,v)\in E$ 有权值 $w(u,v)$。

输出：G的最小生成树T，即边权和最小的生成树。

- 可转化为一个拟阵中找出最大独立子集的问题

例（图拟阵）：给定无向图 $G=(V, E)$，定义

- S_G 为 图 G 的边集 E，
- I_G 是 G 的无循环边集的非空族：如果 A是 E 的子集，则 $A\in I_G$ 当且仅当 A 是无圈的，即图 $G_A=(V, A)$ 构成一个森林。

则 $M_G=(S_G, I_G)$ 是一个拟阵。

Kruskal 算法

输入：无向连通图 G = (V, E) 及边权函数 w

（构造出与最小生成树对应的w，转化为最大独立集问题）

输出：最小生成树T

1. A← ∅;
2. 对E中所有边按照边权值大小递减排序;
3. 按边权值大小递减顺序，对于E中每条边 (u,v),
4. 若A∪{(u, v)}不构成回路
5. A ← A∪{(u, v)};
6. return A

Kruskal 算法

输入：无向连通图 G = (V, E) 及边权函数 w

（构造出与最小生成树对应的w，转化为最大独立集问题）

输出：最小生成树T

1. A← ∅;
2. 对E中所有边按照边权值大小递减排序；
3. 按边权值大小递减顺序，对于E中每条边 (u,v),
4. 若A∪{(u, v)}不构成回路
5. A ← A∪{(u, v)};
6. return A

可以证明：Prim算法不满足拟阵结构

主要内容

- 什么是拟阵(**Matroids**)?
- 拟阵的通用贪心解法
- 拟阵通用贪心解法的正确性
- 基于拟阵的贪心算法举例
  - 最小生成树问题
  - 单位时间任务调度问题

单位时间任务调度问题

- 单位时间任务：需要一个单位的时间来运行的任务

单位时间任务调度问题

- 单位时间任务：需要一个单位的时间来运行的任务
- 调度：给定一个单位时间任务的有限集合 S，

单位时间任务调度问题

- 单位时间任务：需要一个单位的时间来运行的任务
- 调度：给定一个单位时间任务的有限集合 S，

对 S 的一个调度即为S的一个排列，它规定了各任务的执行顺序。

  - 第一个任务开始于时间0，结束于时间1
  - 第二个任务开始于时间1，结束于时间2，… …

单位时间任务调度问题

- 单位时间任务：需要一个单位的时间来运行的任务
- 调度：给定一个单位时间任务的有限集合 S，

对 S 的一个调度即为S的一个排列，它规定了各任务的执行顺序。

  - 第一个任务开始于时间0，结束于时间1
  - 第二个任务开始于时间1，结束于时间2，… …

例： 任务集合 $S=\{a_1, a_2, \ldots, a_n\}$

S的一个调度 $a_{i1}\ a_{i2}\ \ldots\ a_{in}$, $a_{ij} \in S, j \in [1, n]$

具有截止时间和误时惩罚的单位时间任务的调问题

输入:

(1) n个单位时间任务的集合 S={ $a_1, a_2, \ldots, a_n$ };

具有截止时间和误时惩罚的单位时间任务的调问题

输入：

(1) n个单位时间任务的集合 S={ $a_1, a_2, \ldots, a_n$ }；

(2) 任务a_i 的截止时间 d_i, $1 \le i \le n$, $1 \le d_i \le n$，即要求任务 i 在时间 d_i之前结束；

具有截止时间和误时惩罚的单位时间任务的调问题

<u>输入</u>:

(1) n个单位时间任务的集合 S={ $a_1, a_2, \ldots, a_n$ };

(2) 任务a_i 的截止时间 d_i, $1\le i \le n$, $1\le d_i \le n$， 即要求任务 i 在时间 d_i 之前结束;

(3) 任务 a_i 的误时惩罚 w_i, $1\le i \le$ n, 即任务 i 未在规定时间 d_i 之前结束将招致 w_i 惩罚；若按时完成则无惩罚。

具有截止时间和误时惩罚的单位时间任务的调问题

输入:

(1) n个单位时间任务的集合 S={ $a_1, a_2, \ldots, a_n$ };

(2) 任务a_i 的截止时间 d_i, $1 \le i \le n$, $1 \le d_i \le n$, 即要求任务 i 在时间 d_i 之前结束;

(3) 任务 a_i 的误时惩罚 w_i, $1 \le i \le$ n, 即任务 i 未在规定时间d_i之前结束将招致 w_i 惩罚; 若按时完成则无惩罚。

输出:

S的最优调度, 即总误时惩罚最小的调度。

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成
  - 早任务优先形式：早的任务总是在迟的任务之前

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成
  - 早任务优先形式：早的任务总是在迟的任务之前

例. 早任务优先调度：

早任务 a_{i1} a_{i2} … a_{ik} $a_{i,k+1}$ … a_{ip} … a_{in} 迟任务

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成
  - 早任务优先形式：早的任务总是在迟的任务之前

例. 早任务优先调度：

早任务 a_{i1} a_{i2} … a_{ik} $a_{i,k+1}$ … a_{ip} … a_{in} 迟任务

截止时间： $d_{ij} \geq j$, $j=1,\dots,k$; $d_{ij} < j$, $j=k+1,\dots,n$

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成
  - 早任务优先形式：早的任务总是在迟的任务之前

例. 早任务优先调度：

早任务 a_{i1} a_{i2} … a_{ik} $a_{i,k+1}$ … a_{ip} … a_{in} 迟任务

截止时间： $d_{ij} \geq j$, $j = 1, \ldots, k$; $d_{ij} < j$, $j = k+1, \ldots, n$

- 任意一个调度总可以置换成早任务优先的形式

a_{i1} a_{i2} … $\boxed{a_{ip}}$ $a_{i,k+1}$ … $\boxed{a_{ik}}$ … a_{in}

基本概念

- 给定一个调度，定义：
  - 迟(late)任务：一个任务在截止时间后完成
  - 早(early)任务：一个任务在截止时间前完成
  - 早任务优先形式：早的任务总是在迟的任务之前

例. 早任务优先调度：

早任务 a_{i1} a_{i2} … a_{ik} $a_{i,k+1}$ … a_{ip} … a_{in} 迟任务

截止时间： $d_{ij} \geq j,\ j=1,\ldots,k;\ d_{ij} < j,\ j=k+1,\ldots,n$

- 任意一个调度总可以置换成早任务优先的形式

a_{i1} a_{i2} … $\boxed{a_{ip}}$ $a_{i,k+1}$ … $\boxed{a_{ik}}$ … a_{in}

交换 a_{ip}与 a_{ik} 的位置，a_{ip} 仍是迟任务， a_{ik} 仍是早任务

$d_{ip}<p<k$ $d_{ik} \geq k>p$

问题：如何转化为拟阵？

问题：如何转化为拟阵？

- 独立任务集：称一个任务的集合A是独立的，如果存在A的一个调度，使得没有一个任务是迟的。

问题：如何转化为拟阵？

- 独立任务集：称一个任务的集合A是独立的，如果存在A的一个调度，使得没有一个任务是迟的。
- 设：$N_t(A)$是任务子集A中所有截止时间小于等于 t 的任务数，t=1,2,…, n。

问题：如何转化为拟阵？

- 独立任务集：称一个任务的集合A是独立的，如果存在A的一个调度，使得没有一个任务是迟的。
- 设：$N_t(A)$是任务子集A中所有截止时间小于等于 t 的任务数，t=1,2,…, n。

引理4 对于 S 的任一任务子集 A，下面各命题是等价的。

(1) 任务子集 A 是独立子集。

(2) 对于 t = 1, 2, …, n，$N_t(A) \le t$。

(3) 若A中任务依其截止时间非减序排列，则A中所有任务都是早的。

问题：如何转化为拟阵？

- 独立任务集：称一个任务的集合A是独立的，如果存在A的一个调度，使得没有一个任务是迟的。
- 设：$N_t(A)$是任务子集A中所有截止时间小于等于 t 的任务数，t=1,2,…, n。

引理4 对于 S 的任一任务子集 A，下面各命题是等价的。

(1) 任务子集 A 是独立子集。

(2) 对于 t = 1, 2, …, n，$N_t(A) \le t$。

(3) 若A中任务依其截止时间非减序排列，则A中所有任务都是早的。

- 最小化迟任务的惩罚之和等价于最大化早任务的惩罚之和

引理4 对于 S 的任一任务子集 A，下面各命题是等价的。

(1) 任务子集 A 是独立子集。

(2) 对于 $t = 1, 2, \ldots, n$， $N_t(A) \le t$。

(3) 若A中任务依其截止时间非减序排列，则A中所有任务都是早的。

证明：((1)→(2)) (反证法) 假设任务子集A是独立的，且存在某个 t 使得 $N_t(A) > t$, 则A中有多于 t 个任务要在时间 t 之前完成，这显然是不可能的。故A中必有迟任务，这与A是独立任务子集相矛盾。因此，对所有$t=1,2,\ldots,n$，$N_t(A) \le t$。

引理4 对于 S 的任一任务子集 A，下面各命题是等价的。

(1) 任务子集 A 是独立子集。

(2) 对于 $t = 1, 2, \ldots, n$， $N_t(A) \le t$。

(3) 若A中任务依其截止时间非减序排列，则A中所有任务都是早的。

证明：((1)→(2)) (反证法) 假设任务子集A是独立的，且存在某个 t 使得 $N_t(A) > t$, 则A中有多于 t 个任务要在时间 t 之前完成，这显然是不可能的。故A中必有迟任务，这与A是独立任务子集相矛盾。因此，对所有$t=1,2,\ldots,n$，$N_t(A) \le t$;

(2)→(3)：将A中任务按截止时间的非减序排列，则(2)中不等式意味着排序后A中截止时间为 t 的任务前面，需要调度的任务数少于 t。故排序后A中所有任务都是早任务的;

引理4 对于 S 的任一任务子集 A，下面各命题是等价的。

(1) 任务子集 A 是独立子集。

(2) 对于 t = 1, 2, …, n， $N_t(A) \le t$。

(3) 若A中任务依其截止时间非减序排列，则A中所有任务都是早的。

证明：((1)→(2)) (反证法) 假设任务子集A是独立的，且存在某个 t 使得 $N_t(A) > t$, 则A中有多于 t 个任务要在时间 t 之前完成，这显然是不可能的。故A中必有迟任务，这与A是独立任务子集相矛盾。因此，对所有t=1,2,…,n，$N_t(A) \le t$;

(2)→(3)：将A中任务按截止时间的非减序排列，则(2)中不等式意味着排序后A中截止时间为 t 的任务前面，需要调度的任务数少于 t。故排序后A中所有任务都是及时的；

(3) →(1)：显然。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \le N_t(A)$ 的 t 值为 k,

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \le N_t(A)$ 的 t 值为 k, 即

$$k=\max\{t \mid N_t(B) \le N_t(A), 1\le t \le n\}，其中n=|S|。$$

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \le N_t(A)$ 的 t 值为 k, 即

$$k=\max\{t\ |N_t(B) \le N_t(A),\ 1\le t \le n\}，其中n=|S|。$$

由于 $N_n(B)=|B|$, $N_n(A)=|A|$，而$|B| > |A|$，即 $N_n(B)>N_n(A)$。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \le N_t(A)$ 的 t 值为 k, 即

$$k=\max\{t \mid N_t(B) \le N_t(A), 1\le t \le n\}，其中 n=|S|。$$

由于 $N_n(B)=|B|$, $N_n(A)=|A|$，而$|B| > |A|$，即 $N_n(B)>N_n(A)$。

因此必有这样的k<n，对于满足$k+1\le j\le n$的j有

$$N_j(B) > N_j(A)。$$

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \leq N_t(A)$ 的 t 值为 k, 即

$$k=\max\{t \mid N_t(B) \leq N_t(A), 1 \leq t \leq n\}，其中 n=|S|。$$

由于 $N_n(B)=|B|, N_n(A)=|A|$，而|B| > |A|，即 $N_n(B)>N_n(A)$。

因此必有这样的k<n，对于满足k+1≤j≤n的j有

$$N_j(B) > N_j(A)。$$

取x∈ B-A，且x的截止时间为k+1。令 A'=A∪{x}。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：(1)首先，独立任务集的子集显然也是独立子集。故 I 满足遗传性质。

(2)设A、B为独立任务子集且|B|>|A|，下面证明(S, I)满足交换性质。

设从时刻 1开始，最后一次出现 $N_t(B) \le N_t(A)$ 的 t 值为 k, 即

$$k=\max\{t \mid N_t(B) \le N_t(A),\ 1 \le t \le n\}，其中n=|S|。$$

由于 $N_n(B)=|B|$, $N_n(A)=|A|$，而|B| > |A|，即 $N_n(B)>N_n(A)$。

因此必有这样的k<n，对于满足k+1≤j≤n的j有

$$N_j(B) > N_j(A)。$$

取$x \in B-A$，且x的截止时间为k+1。令 $A'=A \cup \{x\}$。

下面证明 A’是独立的。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：首先，由于A是独立的，故对于$1 \le t \le k$有：

$$N_t(A') = N_t(A) \le t。$$

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：首先，由于A是独立的，故对于$1 \le t \le k$有：

$$N_t(A') = N_t(A) \le t。$$

又由于B是独立的，故对$k+1 \le t \le n$有

$$N_t(A') = N_t(A)+1 \le N_t(B) \le t。$$

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：首先，由于A是独立的，故对于$1 \le t \le k$有：

$$N_t(A') = N_t(A) \le t。$$

又由于B是独立的，故对$k+1 \le t \le n$有

$$N_t(A') = N_t(A)+1 \le N_t(B) \le t。$$

由引理4 的条件(2)可知：A’是独立的。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

证明：首先，由于A是独立的，故对于$1 \le t \le k$有：

$$N_t(A') = N_t(A) \le t。$$

又由于B是独立的，故对$k+1 \le t \le n$有

$$N_t(A') = N_t(A)+1 \le N_t(B) \le t。$$

由引理4 的条件(2)可知：A'是独立的。

综上所述，可得 (S, I) 是一个拟阵。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

- 独立任务集A：如果存在关于中任务的一个调度，使得没有一个任务是迟的。

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

- 独立任务集A：如果存在关于中任务的一个调度，使得没有一个任务是迟的。
- 最小化迟任务的惩罚之和等价于最大化早任务的惩罚之和

定理3 设S是带有截止时间的单位时间任务集，I 是 S的所有独立任务子集构成的集合。则有序对(S, I)是拟阵。

- 独立任务集A：如果存在关于中任务的一个调度，使得没有一个任务是迟的。
- 最小化迟任务的惩罚之和等价于最大化早任务的惩罚之和

加权拟阵最优子集的贪心算法框架如下：

输入：具有正权函数W的加权拟阵M = (S, I)

输出：M的最优子集A。

w：误时惩罚

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

$N_t(A\cup\{x\})\leq t$

算法分析

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

O(nlog n)

$N_t(A) \le t$， O(n)时间

- 确定**A**后，按照截止时间单调递增顺序列出**A**中所有元素，然后按任意顺序列出迟的任务（即**S-A**）

算法分析

```
GREEDY(M, w)
A = ∅
sort M.S into monotonically decreasing order by weight w
for each x ∈ M.S, taken in monotonically decreasing order by weight w(x)
    if A ∪ {x} ∈ M.I
        A = A ∪ {x}
return A
```

O(nlog n)

$N_t(A) \le t$， O(n)时间

- 确定A后，按照截止时间<u>单调递增</u>顺序列出A中所有元素，然后按任意顺序列出迟的任务（即S-A）
- 计算时间复杂性是O(nlog n+nf(n))，其中**f(n)是用于检测任务子集A的独立性**所需的时间
- **Greedy-Job**时间复杂度为 **$O(n^2)$**

例：

a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

例：

a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

(1) $A=\{1\}$, $N_i(A)=0\le i$, $i=1, 2, 3$; $N_j(A)=1$, $j=4, 5, 6, 7$.

(2) $A=\{1, 2\}$, $N_1(A)=0\le 1$; $N_i(A)=1\le i$, $i=2, 3$; $N_j(A)=2\le i$, $j=4, 5, 6, 7$.

(3) $A=\{1,2,3\}$, $N_1(A)=0\le 1$; $N_i(A)=1\le i$, $i=2, 3$; $N_j(A)=3\le i$, $j=4, 5, 6, 7$.

(4) $A=\{1,2,3,4\}$, $N_1(A)=0\le 1$; $N_2(A)=1\le 2$, $N_3(A)=2\le 3$, $N_j(A)=4\le i$, $j=4,5,6,7$.

(5) $A=\{1,2,3,4,5\}$, $N_1(A)=1\le 1$; $N_2(A)=2\le 2$, $N_3(A)=3\le 3$, $N_4(A)=5>4$.

(6) $A=\{1,2,3,4,6\}$, $N_1(A)=0\le 1$; $N_2(A)=1\le 2$, $N_3(A)=2\le 3$, $N_4(A)=5>4$.

(7) $A=\{1,2,3,4,7\}$, $N_1(A)=0\le 1$; $N_2(A)=1\le 2$, $N_3(A)=2\le 3$, $N_i(A)=4\le i$, $j=4,5$; $N_j(A)=5$, $j=6, 7$.

最优调度为：2, 4, 1, 3, 7, 5, 6

例：

a_i	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

A={1, 2, 3, 4, 7}

问题：为什么算法没有选择 5和6？

✓ 违反了 $N_4(A) \le 4$

最优调度为：2, 4, 1, 3, 7, 5, 6

总结

- 贪心算法的理论基础
  - 什么是拟阵（**Matroids**）？
  - 拟阵的通用贪心解法
  - 拟阵通用贪心解法的正确性
  - 基于拟阵的贪心算法举例