

1. 二分图匹配讲义

匹配问题是组合优化中的基本问题之一。

在这组讲义中，我们重点讨论底层图是二分图的情况。

我们首先介绍一些基本的图论术语。一个图 $G = (V, E)$ 由一组顶点 V 和一组称为边的顶点对 E 组成。对于一条边 $e = (u, v)$ ，我们说边的端点是 u 和 v ；我们也说边 e 与 u 和 v 相交。如果图 $G = (V, E)$ 是二分图，那么顶点集 V 可以被划分为两个集合 A 和 B （称为二分划分），使得边集 E 中的任何一条边的两个端点不在同一个划分集合中。一个匹配 $M \subseteq E$ 是一组边，使得 V 的每个顶点至多与一条 M 中的边相交。如果一个顶点 v 没有与 M 相交的边，则称 v 为未匹配的。如果没有顶点是未匹配的，则匹配是完美的；换句话说，如果匹配的基数等于 $|A| = |B|$ ，则匹配是完美的。

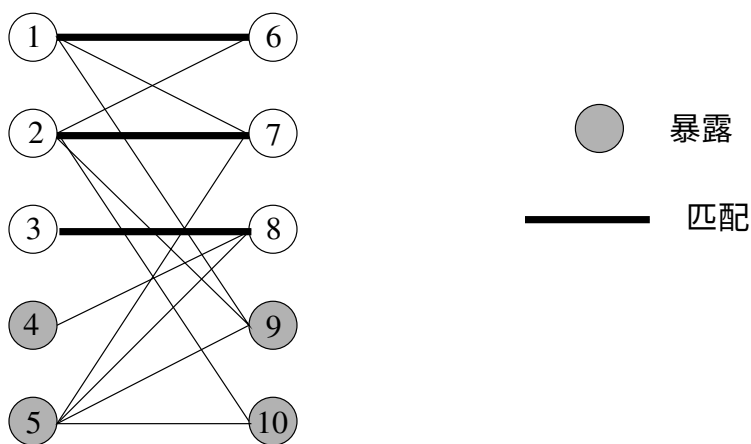


图1.1：示例。边(1,6)、(2,7)和(3,8)形成一个匹配。顶点4、5、9和10是暴露的。

我们对以下两个问题感兴趣：

最大基数匹配问题：找到一个最大大小的匹配 M 。

最小权重完美匹配问题：给定所有 $(i, j) \in E$ 的成本 c_{ij} ，找到最小成本的完美匹配，其中匹配 M 的成本由 $c(M) = \sum$ 这个问题也被称为分配问题。

类似的问题（但更复杂）可以在非二分图上定义。

1.1 最大基数匹配问题

在描述解决最大基数匹配问题的算法之前，人们希望能够证明匹配的最优性（不参考任何算法）。

为了这个目的，人们希望找到匹配的大小的上界，并希望最小的这些上界等于最大匹配的大小。这是一个在这个主题中无处不在的对偶概念。在这种情况下，对偶问题本身将成为一个组合优化问题。

顶点覆盖是一个顶点集合 C ，使得所有边 e 属于 E 都与 C 中至少一个顶点相邻。换句话说，没有边完全包含在 $V - C$ 中（我们使用 $-$ 和 \setminus 来表示两个集合的差异）。显然，任何匹配的大小都不超过任何顶点覆盖的大小。这是因为，给定任何匹配 M ，顶点覆盖 C 必须包含 M 中每条边的至少一个端点。我们刚刚证明了弱对偶性：最大匹配的大小至多等于顶点覆盖的最小大小。正如我们将在这些讲义中后面证明的那样，实际上等号成立：

定理 1.1 (König 1931) 对于任何二分图，最大匹配的大小等于最小顶点覆盖的大小。

我们将通过描述一种同时给出最大匹配和最小顶点覆盖的高效算法，来算法地证明这个最小最大关系。

König 定理给出了这个问题的一个很好的特征，即最优性的简单证明。在上面的例子中，可以证明匹配 $(1,9)$, $(2,6)$, $(3,8)$ 和 $(5,7)$ 是最大的，因为存在一个大小为 4 的顶点覆盖。只需取集合 $\{1, 2, 5, 8\}$ 即可。

解决这个基数匹配问题的自然方法是尝试一种贪心算法：从任意匹配开始（例如空匹配），然后重复添加不相交的边，直到不能再添加为止。然而，这种方法不能保证给出最大匹配（请自行验证）。我们现在将介绍一种确实有效的算法，它基于交替路径和增广路径的概念。路径就是一组边 $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$ ，其中 v_i 是不同的顶点。路径可以简单地表示为 $v_0 - v_1 - \dots - v_k$ 。

定义 1.1 对于匹配 M 来说，交替路径是在 M 的边和 $E - M$ 的边之间交替的路径。

定义 1.2 对于匹配 M 来说，增广路径是一条交替路径，其中第一个和最后一个顶点是暴露的。

在上面的例子中，路径 $4-8-3$ 、 $6-1-7-2$ 或 $5-7-2-6-1-9$ 是交替的，但只有最后一条是增广的。注意，对于匹配 M 来说，包含 k 条 M 的增广路径也必须包含 $k+1$ 条不在 M 中的边。此外，增广路径的两个端点必须在二分图的不同侧。增广路径 P 相对于匹配 M 的最有趣的性质是，如果我们设置 $M' = M \triangle P \equiv (M - P) \cup (P - M)$ ，那么我们得到一个匹配 M' ，并且，此外，

M' 比 M 的大小大一单位。也就是说，我们可以从中形成一个更大的匹配 M' 通过将不在 M 中的边添加到 M' 中，同时从 M' 中删除在路径 P 中也在 M 中的边。我们称这样做为沿着路径 P 增广 M 。增广路径的有用性由以下定理给出。

定理1.2如果没有相对于 M 的增广路径，则匹配 M 是最大的。

证明：（反证法）

(\Rightarrow) 假设 P 是相对于 M 的某个增广路径。令 $M' = M \triangle P$ 。那么 M' 是一个比 M 更大的匹配。这与 M 的最大性相矛盾。（ \Leftarrow) 如果 M 不是最大的，让 M^* 是一个最大匹配（使得 $|M^*| > |M|$ ）。令 $Q = M \triangle M^*$ 。那么：

- Q 从 M^* 比从 M 有更多的边（因为 $|M^*| > |M|$ 意味着 $|M^* - M| > |M - M^*|$ ）。
- 每个顶点最多与 $M \cap Q$ 中的一条边和 $M^* \cap Q$ 中的一条边相邻。
- 因此 Q 由从 M 和 M^* 中的边交替组成的循环和路径组成。
- 因此，必定存在一条路径，其中从 M^* 中的边比从 M 中的边更多（所有的循环长度都是偶数，并且从 M^* 和 M 中有相同数量的边）。这条路径是相对于 M 的增广路径。

因此，必定存在一条相对于 M 的增广路径 P ，这是一个矛盾。

△

这个定理激发了以下算法。从任意匹配 M 开始，比如空匹配。重复寻找相对于 M 的增广路径 P ，沿着 P 增广 M 并用结果替换 M 。当不存在更多的增广路径时停止。根据上述定理，我们保证找到了最优匹配。该算法在 μ 次增广中终止，其中 μ 是最大匹配的大小。

显然， $\mu \leq \frac{n}{2}$ 其中 $n = |V|$ 。

在这个例子中，我们可以沿着增广路径（例如5-7-2-6-1-9）来增广匹配 M ，得到匹配对 (1,9), (2,6), (3,8) 和 (5,7)，然后意识到没有更多的增广路径可以找到。

现在的问题是如何判断增广路径的存在以及如何找到一个增广路径（如果存在）。可以通过以下方式完成这些任务。根据匹配 M 在 G 中直接构建有向边，规则如下：如果一条边不属于匹配 M ，则从 A 到 B 有一条边，如果一条边属于匹配 M ，则从 B 到 A 有一条边。将这个有向图称为 D 。

命题1.3在 G 中存在一个增广路径，关于匹配 M ，当且仅当在 D 中存在一条从 A 中的未匹配顶点到 B 中的未匹配顶点的有向路径。

练习 1-1. 证明命题 1.3.

这给出了一个 $O(m)$ 算法（其中 $m = |E|$ ），用于在 G 中找到一条增广路径。设 A^* 和 B^* 分别是相对于 M 在 A 和 B 中的暴露顶点的集合。我们可以简单地将一个顶点 s 连接到 A^* 中的所有顶点，并从 s 开始进行深度优先搜索，直到我们遇到 B^* 中的一个顶点，然后回溯路径。

因此，找到最大基数匹配的总复杂度为 $O(nm)$ 。通过同时沿着多条增广路径进行增广，这可以改进为 $O(\sqrt{nm})$ 。

如果相对于 M 不存在增广路径，则我们还可以使用我们的搜索过程来构造一个最优顶点覆盖。考虑可以通过从 A 中的一个暴露顶点沿着有向路径到达的顶点的集合 L （用于标记）。

定理1.4当算法终止时， $C^* = (A - L) \cup (B \cap L)$ 是一个顶点覆盖。

此外， $|C^*| = |M^*|$ ，其中 M^* 是算法返回的匹配。

这个定理立即证明了König定理。

证明：我们首先证明 C^* 是一个顶点覆盖。假设不是。那么必定存在一条边 $e = (a, b) \in E$ ，其中 $a \in A \cap L$ 且 $b \in (B - L)$ 。边 e 不能属于匹配。如果是的话，那么 b 应该在 L 中，否则 a 就不会在 L 中。因此， e 必须属于 $E - M$ ，并且从 A 指向 B 。这就意味着 b 可以从 A 中的一个暴露顶点到达（即通过边 (a, b) ），与 $b \in L$ 的事实相矛盾。

为了证明证明的第二部分，我们证明 $|C^*| \leq |M^*|$ ，因为对于任何匹配和任何顶点覆盖，反向不等式都成立。证明从以下观察得出。

1. 根据 L 的定义， $A - L$ 中的任何顶点都不会被暴露。
2. 由于这将意味着存在一条增广路径，因此 $B \cap L$ 中的任何顶点都不会被暴露，从而算法不会终止。
3. 匹配中不存在连接一个顶点 $a \in (A - L)$ 和一个顶点 $b \in (B \cap L)$ 的边。否则， a 将在 L 中。

这些备注意味着 C^* 中的每个顶点都被匹配，并且匹配的相应边是不同的。因此， $|C^*| \leq |M^*|$ 。 △

尽管最大匹配和最小顶点覆盖的概念也可以应用于一般（即非二分图）的图，但我们应该注意到König定理并不适用于所有图。事实上，虽然最大匹配的大小总是小于等于最小顶点覆盖的大小，但它们并不一定相等。

考虑一下三个顶点的环 C_3 （最小的非二分图）。最大匹配的大小为1，但最小顶点覆盖的大小为2。我们将推导出一个涉及一般图的最大匹配的极小极大关系，但它会比König定理更复杂。

练习

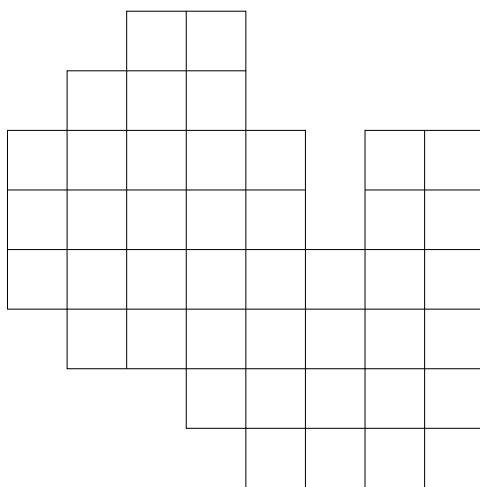
练习1-2. 图 $G = (V, E)$ 的边覆盖 R 是 E 的一个子集, 使得 V 的每个顶点至少与 R 中的一条边关联. 假设 G 是一个没有孤立顶点的二分图. 证明最小边覆盖 R^* of G 的基数等于 $|V|$ 减去最大匹配 M^* of G 的基数. 给出一个高效的算法来找到 G 的最小边覆盖. 这对于非二分图也成立吗?

练习题 1-3. 证明在任意图 $G = (V, E)$ (不一定是二分图) 中, 任意最大匹配 M (即不能再添加边使其仍然是匹配) 的大小至少是最大匹配 M^* 大小的一半.

练习题 1-4. 考虑将棋盘的一个子集 (即由单位方格组成, 见下面的示例) 完美铺砌成多米诺骨牌 (一块多米诺骨牌由两个相邻方格组成) 的问题.

1. 证明这个问题可以转化为判断一个二分图是否有完美匹配的问题.

2. 以下图形能够被多米诺骨牌铺砌吗? 给出一种铺砌方式或者证明无法铺砌.



练习题 1-5. 考虑一个二分图 $G = (V, E)$, 其顶点集分为两个部分 A 和 B : $V = A \cup B$. 假设存在某些顶点集 $A_1 \subseteq A$ 和 $B_1 \subseteq B$, 使得存在一个匹配 M_A 覆盖 A_1 中的所有顶点, 以及一个匹配 M_B 覆盖 B_1 中的所有顶点. 证明总存在一个匹配覆盖 $A_1 \cup B_1$ 中的所有顶点.

练习题 1-6. 考虑在一个 (不一定是二分图的) 图 $G = (V, E)$ 上的以下二人游戏. 玩家 1 和玩家 2 轮流选择图中的一条 (尚未选择的) 边 e , 使得 e 和之前选择的边构成一条简单路径. 第一个无法选择这样一条边的玩家失败. 证明如果图 G 有一个完美匹配, 则玩家 1 有一个获胜策略.

1.1.1 Hall 定理

Hall 定理给出了一个二分图是否有一个匹配能够饱和（或匹配）所有 A 顶点（即大小为 $|A|$ 的匹配）的必要和充分条件。

定理 1.5 (Hall 1935) 给定一个二分图 $G = (V, E)$ ，其顶点集分为 A 和 B ($V = A \cup B$)，当且仅当对于每个 $S \subseteq A$ ，我们有 $|N(S)| \geq |S|$ ，其中 $N(S) = \{b \in B : \exists a \in S \text{ 使得 } (a, b) \in E\}$ 。

显然，霍尔定理中给出的条件是必要的；它的充分性可以从科尼希定理推导出来。

练习1-7. 从科尼希定理推导出霍尔定理。

练习1-8. 考虑一个二分图 $G = (V, E)$ ，其二分割为 (A, B) 。对于 $X \subseteq A$ ，定义 $\text{def}(X) = |X| - |N(X)|$ 其中 $N(X) = \{b \in B : \exists a \in X \text{ 使得 } (a, b) \in E\}$ 。令

$$\text{def}_{\max} = \max_{X \subseteq A} \text{def}(X)。$$

由于 $\text{def}(\emptyset) = 0$ ，我们有 $\text{def}_{\max} \geq 0$ 。

1. 通过展示在二分图 G 中匹配的最大大小等于 $|A| - \text{def}_{\max}$ 来推广霍尔定理。

2. 对于任意的两个子集 $X, Y \subseteq A$ ，证明

$$\text{def}(X \cup Y) + \text{def}(X \cap Y) \geq \text{def}(X) + \text{def}(Y)。$$

1.2 最小权重完美匹配

通过将不存在的边赋予无限成本，可以假设二分图是完全的。最小成本（权重）完美匹配问题通常通过以下故事来描述：有 n 个作业需要在 n 台机器或计算机上处理，希望每台机器只处理一个作业，使得处理作业的总成本最小化。形式上，我们给出了每个 $i \in A, j \in B$ 的成本 $c_{ij} \in \mathbb{R} \cup \{\infty\}$ ，并且目标是找到一个完美匹配 M 使得 $\sum_{(i,j) \in M} c_{ij}$ 最小化。

在这些讲义中，我们介绍了一个基于线性规划的算法来解决这个问题，并且我们将借此机会阐述线性规划中的几个重要概念。这些概念将在后续章节中被形式化和推广。

分配问题的第一个算法是由 Kuhn [1955] 提出的，但他只证明了算法的有限性。Jim Munkres [1957] 提出了一个改进版本，并证明了它的多项式运行时间。如果一个算法的运行时间（运行它所需的基本操作数）被一个多项式上界限制，那么它就是多项式时间算法。

即它的运行时间与输入的大小（即表示输入所需的位数）无关。Munkres的分析甚至表明该算法是强多项式的，这意味着运行时间是与所涉及的数字数量（即不依赖于成本的大小）的多项式。在这个算法中，操作的次数上界是 $O(n^3)$ ，其中 $n = |V|$ 。该算法通常被称为匈牙利方法，因为它依赖于匈牙利人的思想，特别是König和Egerváry的思想。2006

年，人们发现这种方法实际上是19世纪由Jacobi发现的，这篇论文于1890年以拉丁文发表，详见<http://www.lix.polytechnique.fr/~ollivier/JACOBI/jacobiEngl.htm>。我们首先给出了将问题作为整数规划的形式，即变量受限于整数值，约束条件和目标函数是这些变量的线性函数的优化问题。我们首先需要将每个匹配关联到一个点。为此，给定一个匹配 M ，让它的关联向量为 x ，其中 $x_{ij} = 1$ ，如果 $(i, j) \in M$ ，否则为0。可以将最小权重完美匹配问题表述如下：

$$\begin{aligned} & \text{最小} \quad \sum_{i,j} c_{ij} x_{ij} \\ & \text{满足以下条件:} \\ & \quad \sum_j x_{ij} = 1 \quad i \in A \\ & \quad \sum_i x_{ij} = 1 \quad j \in B \\ & \quad x_{ij} \geq 0 \quad i \in A, j \in B \\ & \quad x_{ij} \text{ 为整数} \quad i \in A, j \in B. \end{aligned}$$

这不是一个线性规划问题，而是一个所谓的整数规划问题。注意，任何对这个整数规划问题的解都对应一个匹配，因此这是二分图中最小权重完美匹配问题的一个有效表述。

现在考虑通过去除整数约束条件得到的线性规划问题 (P)：

$$\begin{aligned} & \text{最小} \quad \sum_{i,j} c_{ij} x_{ij} \\ & \text{满足以下条件:} \\ (P) \quad & \sum_j x_{ij} = 1 \quad i \in A \\ & \sum_i x_{ij} = 1 \quad j \in B \\ & x_{ij} \geq 0 \quad i \in A, j \in B. \end{aligned}$$

这是上述整数规划问题的线性规划松弛问题。在线性规划问题中，变量可以取分数值，因此存在许多满足上述约束条件的可行解，但并不对应匹配。但我们只关心最优解。

关于最优解。满足 (P) 中约束条件的可行解构成一个有界多面体或多胞体，当我们在多胞体上优化线性约束时，最优解将出现在多胞体的一个“角点”或极点上。集合 Q 的一个极点 x 是一个元素 $x \in Q$ ，不能表示为 $\lambda y + (1 - \lambda) z$ ，其中 $0 < \lambda < 1$ ， $y, z \in Q$ 且 $y \neq z$ 。（这将在多面体组合学章节中进行形式化和深入讨论。）

一般来说，即使线性规划的约束矩阵的所有系数都是0或1，线性规划的极端点也不能保证所有坐标都是整数（这并不令人惊讶，因为一般整数规划问题是NP难问题，而线性规划是多项式可解的）。因此，一般来说，整数规划的值 Z_{IP} 不一定等于其线性规划松弛问题的值 Z_{LP} 。然而，由于整数规划比松弛问题更受限制，我们总是有 $Z_{IP} \geq Z_{LP}$ ，这意味着 Z_{LP} 是最小化问题 Z_{IP} 的下界。此外，如果线性规划松弛问题的最优解是整数的（在我们的情况下，这意味着它是完美匹配的关联向量），那么它也必须是整数规划的最优解。

练习题 1-9. 证明这个最后的结论。

练习题 1-10. 给出一个整数规划的例子，其中 $Z_{IP} = Z_{LP}$ 。

然而，在完美匹配问题的情况下，约束矩阵具有非常特殊的形式，可以证明以下非常重要的结果。

定理 1.6 (P) 的任何极端点都是一个 0-1 向量，因此是一个完美匹配的关联向量。

由于上述定理，这个多面体

$$P = \left\{ x : \begin{array}{ll} \sum_j x_{ij} = 1 & i \in A \\ \sum_i x_{ij} = 1 & j \in B \\ x_{ij} \geq 0 & i \in A, j \in B \end{array} \right\}$$

被称为二分图完美匹配多面体。

为了展示匹配的美妙之处，我们将在这里给出两个完全不同的证明，一个是纯粹的算法证明，另一个是在多面体理论章节中的纯代数证明。代数证明与完全单调性的概念有关。

为了算法证明，我们描述了解决最小权重完美匹配问题的算法。该算法是“原始-对偶”算法。为了解释这意味着什么，我们需要介绍线性规划的对偶性概念，并在我们的二分图匹配问题的具体情况下进行。假设我们有值 u_i 对于 $i \in A$ 和 v_j 对于 $j \in B$ 。

使得对于所有 $i \in A$ 和 $j \in B$, 有 $u_i + v_j \leq c_{ij}$ 。那么对于任何完美匹配 M , 我们有

$$\sum_{(i,j) \in M} c_{ij} \geq \sum_{i \in A} u_i + \sum_{j \in B} v_j. \quad (1)$$

因此, $\sum_{i \in A} u_i + \sum_{j \in B} v_j$ 是最小成本完美匹配的下界 (对于二分图)。为了获得最佳下界, 我们希望最大化这个数量, 因此我们得到另一个线性规划。

$$\text{Max} \quad \sum_{i \in A} u_i + \sum_{j \in B} v_j$$

受限于:

$$(D) \quad u_i + v_j \leq c_{ij} \quad i \in A, j \in B.$$

这些约束可以解释为 $w_{ij} \geq 0$, 其中 $w_{ij} = c_{ij} - u_i - v_j$ 。这是一个线性规划问题, 称之为 (D)。到目前为止, 我们已经证明了这个线性规划问题 (D) 为任何完美匹配的成本提供了一个下界, 但我们甚至可以证明它为线性规划问题 (P) 的任何解的值提供了一个下界。事实上, 考虑任意 $x \in P$ 。我们有

$$\begin{aligned} \sum_{i \in A} \sum_{j \in B} c_{ij} x_{ij} &\geq \sum_{i \in A} \sum_{j \in B} (u_i + v_j) x_{ij} \\ &= \left(\sum_{i \in A} u_i \sum_{j \in B} x_{ij} \right) + \left(\sum_{j \in B} v_j \sum_{i \in A} x_{ij} \right) \\ &= \sum_{i \in A} u_i + \sum_{j \in B} v_j, \end{aligned}$$

由于 x 满足的约束条件。(D) 是线性规划对偶意义上的对偶线性规划问题。

总结一下, 到目前为止, 我们知道

$$\left[\min_{\text{完美匹配 } M} \sum_{(i,j) \in M} c_{ij} \right] \geq \left[\min_{x \in P} \sum_{(i,j)} c_{ij} x_{ij} \right] \geq \left[\max_{(u,v) \in D} \sum_{i \in A} u_i + \sum_{j \in B} v_j \right].$$

如果对于任何实例, 我们总是能找到一个可行解 u, v 使得 (D) 和一个完美匹配 M 满足 (1) 中的等式 (即完美匹配的成本等于对偶解的值), 那么我们就知道在整个过程中都有等式成立, 找到的匹配是最优的, 而且此外, 匹配 M 的关联向量对于线性规划 (P) 也是最优的。给定对偶解 u, v , 如果完美匹配 M 只包含 $w_{ij} = c_{ij} - u_i - v_j = 0$ 的边, 则满足等式。

这就是所谓的互补松弛。然而, 对于给定的 u, v , 我们可能无法在 $w_{ij} = 0$ 的边中找到一个完美匹配。

该算法执行一系列迭代，以获得适当的 u 和 v 。它始终保持一个对偶可行解，并尝试找到一个“几乎”满足互补松弛条件的原始可行解 x 。互补松弛条件的强制实施在任何原始对偶算法中都是至关重要的。事实上，在组合优化中，最重要（也最优雅）的算法是原始对偶算法。这是设计组合优化问题的高效算法的最重要工具之一（当然，前提是该问题具有这样的高效解决方案）。

更准确地说，该算法的工作原理如下。首先，它从任何对偶可行解开始，例如对于所有 i ， $u_i = 0$ ，并且对于所有 j ， $v_j = \min_i c_{ij}$ 。在给定的迭代中，算法具有对偶可行解 (u, v) 或者说 (u, v, w) 。强制实施互补松弛条件意味着我们对于 $E = \{(i, j) : w_{ij} = 0\}$ 的子图匹配感兴趣。如果 E 有一个完美匹配，则该匹配的关联向量是 (P) 中的一个可行解，并且与当前对偶解满足互补松弛条件，因此必须是最优的。要检查 E 是否有完美匹配，可以使用前面在这些讲义中开发的基数匹配算法。如果最大匹配输出不完美，则算法将使用来自最优顶点覆盖 C^* 的信息来更新对偶解，以使对偶解的值增加（我们在对偶中进行最大化）。

特别地，如果 L 如前一节所述，则 E 中不存在边连接 $A \cap L$ 和 $B - L$ 。换句话说，对于每个 $i \in (A \cap L)$ 和每个 $j \in (B - L)$ ，我们有 $w_{ij} > 0$ 。令

$$\delta = \min_{i \in (A \cap L), j \in (B - L)} w_{ij}.$$

根据上述论证， $\delta > 0$ 。对偶解的更新如下：

$$u_i = \begin{cases} u_i & i \in A - L \\ u_i + \delta & i \in A \cap L \end{cases}$$

和

$$v_j = \begin{cases} v_j & j \in B - L \\ v_j - \delta & j \in B \cap L \end{cases}$$

很容易验证这个对偶解是可行的，即相应的向量 w 满足 $w_{ij} \geq 0$ 对于所有的 i 和 j 。新对偶解的值是多少？新对偶解和旧对偶解之间的差值等于：

$$\delta(|A \cap L| - |B \cap L|) = \delta(|A \cap L| + |A - L| - |A - L| - |B \cap L|) = \delta\left(\frac{n}{2} - |C^*|\right),$$

其中 A 的大小为 $n/2$ ， C^* 是具有边集 E 的二分图的最优顶点覆盖。但根据假设 $|C^*| < \frac{n}{2}$ ，这意味着对偶解的值严格增加。

重复此过程直到算法终止。此时，我们有一个完美匹配的关联向量，同时还有满足互补松弛条件的对偶可行解。因此，它们必须是最优的，从而证明了存在一个整数最优解

对于问题 (P) 的存在。由于通过精心选择成本函数，可以使任何极端点成为线性规划的唯一最优解（这将在多面体章节中正式证明），这表明任何极端点都是整数的，从而证明了定理1.6。

当然，正如一些读者可能已经注意到的那样，证明还不完整，因为需要证明算法确实终止。这可以通过注意到至少还有一个更多的 B 的顶点可以从一个暴露的 A 的顶点到达（并且没有 B 的顶点变得不可达），因为一条边 $e = (i, j)$ 其中 $i \in (A \cap L)$ 且 $j \in B - L$ 现在根据我们选择的 δ 有 $w_{ij} = 0$ 来证明。这也给出了迭代次数的估计。

在最多 $n/2$ 次迭代中，所有 B 的顶点都是可达的，或者找到的匹配增加了至少一个单位。因此，在 $O(n^2)$ 次迭代后，找到的匹配是完美的。算法的总运行时间因此为 $O(n^4)$ ，因为每次迭代计算集合 L 需要 $O(n^2)$ 的时间。通过更仔细地观察在匹配大小增加两次之间顶点的标记方式，可以将运行时间分析降低到 $O(n^3)$ 。

练习题 1-11. 检查算法的运行时间确实为 $O(n^3)$ 。

例子：考虑以下在二分图上定义的成本矩阵给出的实例，每个边分割有5个顶点：

0	2	7	2	3
1	3	9	3	3
1	3	3	1	2
4	0	1	0	2
0	0	3	0	0

边集合 E 中的边 $w_{ij} = 0$ 精确对应于图1.1中的边集合。最大基数匹配算法找到匹配 $(1,9)$, $(2,6)$, $(3,8)$ 和 $(5,7)$ ，标记的顶点集合为 $\{3, 4, 8\}$ 。我们计算 δ 为

$$\delta = \min_{i \in \{3,4\}, j \in \{6,7,9,10\}} w_{ij} = 1$$

对应于边 $(3,9)$ 。新的向量 u 和 v 分别为 $u^T = (2, 3, 1, -1, 0)$ 和 $v^T = (-2, 0, 2, 0, 0)$ 。对偶解的值从4增加到5个单位。相应的集合 E 现在有一个完美匹配，即 $(1,6)$, $(2,7)$, $(3,9)$, $(4,8)$ 和 $(5,10)$ ，成本为5。匹配和对偶解都是最优的。

练习

练习题 1-12. 考虑一个二分图 $G = (V, E)$ ，其中每个顶点的度数为 k （即所谓的 k -正则二分图）。证明这样的图总是有两种不同的完美匹配方式：

1. 使用König定理,
2. 使用我们在本节中推导出的线性规划公式。

(可选: 这个结果对非二分图也成立吗?)

练习题 1-13. 利用前面的练习题 12, 证明 k -正则二分图 G 的边可以划分为 k 个匹配 (即着色二分图的边所需的颜色数, 使得任意两条有共同端点的边颜色不同 — 边的色数 — 恰好为 k)。 (可选: 这个结果对非二分图也成立吗?)

练习题 1-14. 给定一个图 $G = (V, E)$, 它的边着色数是为了使任意两条有共同端点的边具有不同颜色所需的最小颜色数。

1. 证明一个二分图 G 的边着色数总是等于它的最大度数 Δ (即所有顶点 v 的边数的最大值)。(使用上一个问题。)
2. 给出一个非二分图的例子, 其中边着色数严格大于 Δ 。

练习题 1-15. 我们已经证明了线性规划问题 (P) 总是存在一个解 x , 其中所有分量都是整数。以下是对这个结果的另一种证明方法。取一个 (可能非整数的) 最优解 x^* 。

如果存在多个最优解, 选择其中非整数值 x_{ij}^* 最少的一个。证明如果 x^* 不是整数, 存在一个循环 C , 其中所有边 $e = (i, j) \in C$ 的值 x_{ij}^* 都不是整数。现在展示如何得到另一个具有更少非整数值的最优解, 从而导致矛盾。

练习 1-16. 在这个练习中, 你将对 (最小成本) 分配问题进行一次小实验。取一个完全二分图, 每一边有 n 个顶点, 并假设所有的 c_{ij} (对于 $i, j \in \{1, \dots, n\}$) 都是独立均匀分布在 0 和 1 之间的随机变量。取 5 个不同的 n 值 (最大值为几百), 并计算每个 n 值下的 5 个实例的最小成本分配值。对于 n 趋向于无穷大, 你有什么猜测这个值会如何增加。它似乎收敛吗? 收敛到什么值? 感到惊讶吗? (是的, 你应该感到惊讶, 而且不理解为什么是正常的。) 要解决分配问题, 你可以使用 MATLAB (它在 athena 上可用)。要在 athena 上使用 MATLAB, 你需要输入:

```
>> 添加matlab
>> matlab
```

如果你以前从未使用过MATLAB，你可以在18.06网页上找到一些教程。

在<http://www-math.mit.edu/~goemans/bghungar.m>上有一个MATLAB m文件，它实现了匈牙利方法（只需将其放在运行MATLAB的目录中）。如果 C 是一个 $n \times n$ 矩阵，那么

```
>> a=bghungar(C)
```

给出一个向量 a ，使得 $(1, a(1)), (2, a(2)), \dots$ 是最大匹配。所以要得到最小分配的值，你只需要执行以下操作：

```
>> a=bghungar(-C);  
>> v=sum(diag(C(1:n,a(1:n))))
```

练习1-17。对于分配问题，贪婪算法（反复找到与所有先前选择的边不相交的最小成本边）可能导致解的成本除以最优成本可以任意大（即使对于两边各有2个顶点的图形也是如此）。

假设现在成本来自于一个度量，即使只是一条线度量。更准确地说，假设二分是 $A \cup B$ ，其中 $|A| = |B| = n$ ，并且 A 的第 i 个顶点（ B 的第 j 个顶点）与 $a_i \in \mathbb{R}$ （ $b_j \in B$ ）相关联。假设这些顶点之间的成本由 $c_{ij} = |a_i - b_j|$ 给出。

考虑贪心算法：选择最近的一对顶点，一个来自 A ，一个来自 B ，将它们匹配在一起，删除它们，然后重复，直到所有顶点都匹配。对于这些线度量实例，贪心解的成本是否总是由一个常数（与 n 无关）乘以分配的最优成本来上界限制？如果是，请证明；如果不是，请给出一个（以 n 为参数的）例子族，使得相应的比值变得任意大。

2. 非二分图匹配讲义

给定一个图 $G = (V, E)$ ，我们对于找到并描述最大匹配的大小感兴趣。由于我们不假设图是二分图，我们知道最大匹配的大小不一定等于最小顶点覆盖的大小，就像二分图的情况一样（König定理）。事实上，对于一个三角形，任何匹配最多只包含一条边，而我们需要两个顶点来覆盖所有的边。

为了得到任意匹配 M 的上界，考虑任意顶点集 U 。如果我们删除顶点 U （以及与之相邻的所有边），我们最多删除匹配 M 的 $|U|$ 条边。此外，在剩余的图 $G \setminus U$ 中，我们可以通过 $\sum_{i=1}^k \lfloor \frac{|K_i|}{2} \rfloor$ 来简单地上界剩余匹配的大小。其中 $K_i, i = 1, \dots, k$, 是 $G \setminus U$ 的连通分量的顶点集。因此，我们得到

$$|M| \leq |U| + \sum_{i=1}^k \left\lfloor \frac{|K_i|}{2} \right\rfloor. \quad (1)$$

如果我们将 $o(G \setminus U)$ 定义为 $G \setminus U$ 的奇数连通分量的数量，我们可以将 (1) 重写为：

$$|M| \leq |U| + \frac{|V| - |U|}{2} - \frac{o(G \setminus U)}{2},$$

或者

$$|M| \leq \frac{1}{2} (|V| + |U| - o(G \setminus U)). \quad (2)$$

我们将证明我们总是可以找到一个匹配 M 和一个集合 U ，使得我们有相等性；这给出了以下的最小最大关系，称为Tutte-Berge最小最大公式：定理2.1（Tutte-Berge公式）

对于任意图 $G = (V, E)$ ，我们有

$$\max_M |M| = \min_{U \subseteq V} \frac{1}{2} (|V| + |U| - o(G \setminus U)),$$

其中 $o(G \setminus U)$ 是 $G \setminus U$ 的奇数大小的连通分量的数量。

例子：在图2.1中的图中，可以轻松找到大小为8的匹配（找到它），并且从Tutte-Berge公式可以看出它的最优性。实际上，对于集合 $U = \{2, 15, 16\}$ ，我们有 $o(G \setminus U) = 5$ 和 $\frac{1}{2}(|V| + |U| - o(G \setminus U)) = \frac{1}{2}(18 + 3 - 5) = 8$ 。

为了证明定理2.1，我们首先将展示一种找到最大匹配的算法。

这个算法是由Edmonds [1965]提出的，是一颗纯粹的宝石。就像二分图匹配的情况一样（参见二分图匹配讲义），我们将使用增广路径。

事实上，二分图匹配讲义中的定理1.2在非二分图的情况下仍然成立；一个匹配 M 是最大的，当且仅当不存在相对于它的增广路径。困难之处在于找到增广路径或判断不存在这样的路径。我们可以尝试从匹配 M 的未匹配顶点集合 X 开始，每当我们

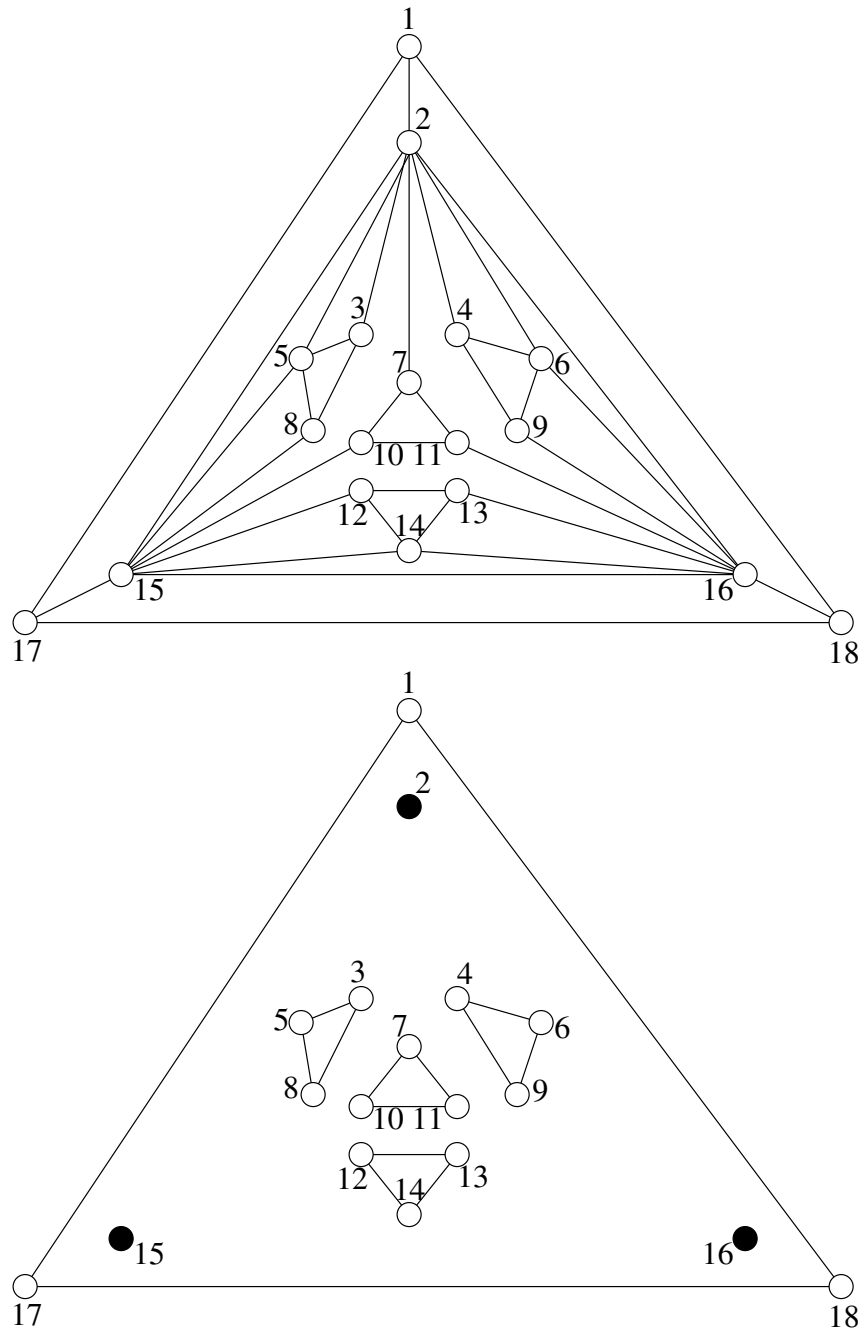


图2.1: 顶部: 图。 底部: 移除顶点2、15和16后得到5个奇数连通分量。

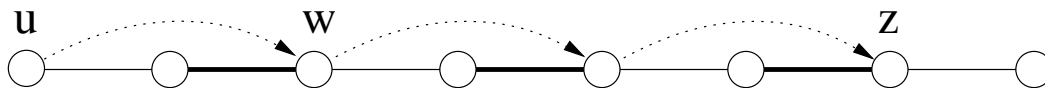


图2.2: 一条增广路径。粗边是匹配中的边。通过从未匹配顶点 u 开始，沿着虚线走直到找到一个与未匹配顶点相邻的顶点（这里是 z ）。找到这样的顶点后，路径就被找到了。

如果我们在一个顶点 u 处看到一条边 $(u, v) \notin M$ ，后面跟着一条边 (v, w) 在 M 中，我们可以从 u 到 w 添加一条有向边，并移动到 w 。如果我们到达一个与一个未匹配顶点相邻的顶点（即在 X 中），看起来我们找到了一条增广路径，见图2.2。但这并不一定是真的，因为这条‘路径’上的顶点可能不是不同的。我们找到了一个所谓的花，见图2.3。这个花不包含增广路径。更正式地说，一个花由一个从一个未匹配顶点 u 到一个顶点 v 的偶数交替路径 P 组成，称为茎，并且包含 v 的奇数环，其中边在匹配中交替进出，除了与 v 相邻的两条边；这个奇数环被称为花瓣。

该算法将找到增广路径或花朵，或者证明不存在这样的项目；在后一种情况下，匹配是最大的，算法停止。如果找到增广路径，则匹配被增广，算法继续使用这个新匹配。如果找到一个花朵，我们将创建一个新的图 G/B ，其中我们将 B 缩小为一个单独的顶点 b ；在 G 中，任何边 (u, v) 其中 $u \notin B$ 且 $v \in B$ 的边将被替换为 G/B 中的边 (u, b) ，所有 B 内的边消失，所有 $V \setminus B$ 内的边保留。请注意，在这个新图中我们也有一个匹配 M/B （通过简单地删除 M 在 B 内的所有边获得），而 M 和 M/B 的大小恰好相差 $|B|-1$

—₂— (因为我们删除了很多边

在匹配中 B) 我们使用以下关键定理。

定理2.2 设 B 是关于 M 的花朵。如果且仅如果 M 是 G 中的最大匹配
那么 M/B 是 G/B 中的最大匹配。

为了证明这个定理，我们可以假设带有花朵 B 的花有一个空的茎 P 。如果不是这种情况，我们可以考虑匹配 $M \triangle P = (M \setminus P) \cup (P \setminus M)$ 对于其中我们有一个带有花朵 P 和空茎的花。证明 $M \triangle P$ 的定理也证明了 M 的定理，因为 $(M \triangle P)/B = (M/B) \triangle P$ 并且通过偶数交替路径进行对称差不会改变匹配的基数。

证明: (\Rightarrow) 假设 N 是 G/B 中比 M/B 更大的匹配。将 N 拉回到 G 中的一组边，它最多与 B 中的一个顶点相邻。通过添加 1 将其扩展为 G 中的匹配 N^+ 。

$\frac{1}{2}(|B| - 1)$ 条边来匹配 B 中的每个其他顶点。那么 $|N^+|$ 将超过 $|M|$ 与 $|N|$ 超过 $|M/B|$ 的数量相同。

(\Leftarrow) 反证法。如果 M 在 G 中不是最大大小的，则它具有增广路径 P 在暴露的顶点 u 和 v 之间。由于 B 只有一个暴露的顶点，我们可以假设 $u \notin B$ 。让 w 是 P 中属于 B 的第一个顶点，让 Q 是 P 的一部分。

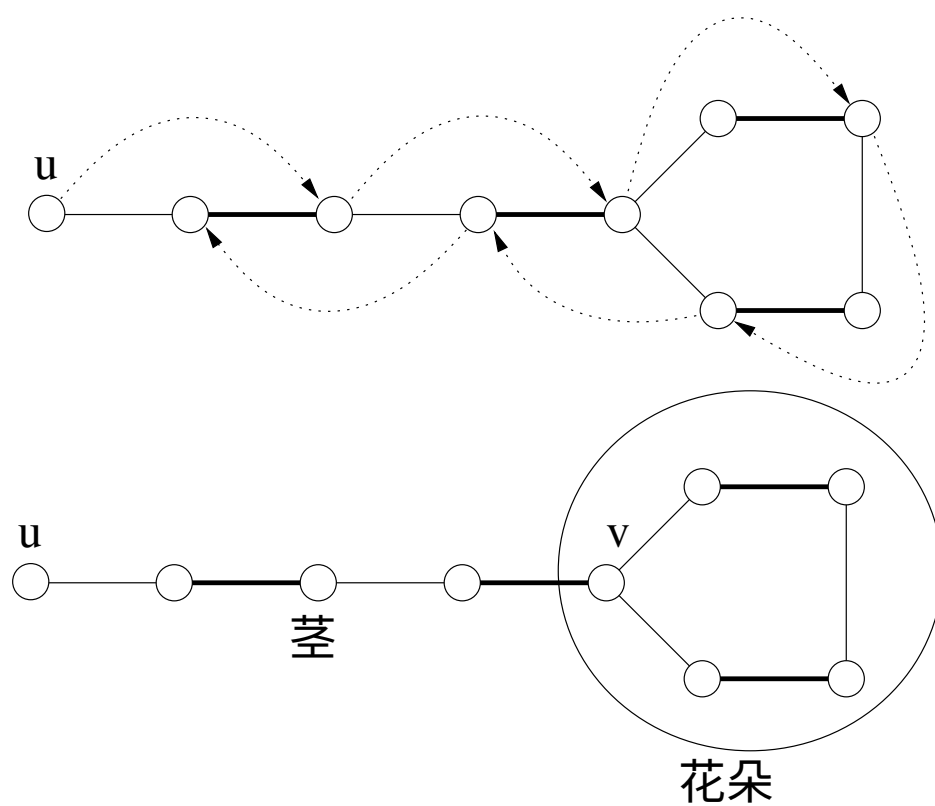


图2.3：一朵花。粗边是匹配的边。顶部：我们的虚线路径从一个暴露的顶点 u 开始，以一个暴露顶点的邻居结束，不对应于增广路径。

从 u 到 w 。注意，在缩小 B 后， Q 仍然是 M/B 的增广路径（因为在 G/B 中， b 是暴露的）。这意味着 M/B 也不是最大的，我们达到了矛盾。

△

观察到证明是算法性的：如果在 G/B 中有一个更大的匹配 M/B ，那么我们也可以在 G 中找到一个更大的匹配 M 。还要注意，定理2.2并不是说如果我们在 G/B 中找到一个最大匹配 M^* ，然后简单地添加 $\frac{|B|-1}{2}$ 条边从 B 内部到 M^* 得到 \hat{M} 将导致一个最大匹配在 G 中。事实上，这是不正确的。

练习2-1. 给出一个图 G ，一个匹配 M 和一个花 B ，使得在 G/B 中的最大匹配 M^* 不会导致 G 中的最大匹配。解释为什么这不与定理2.2相矛盾。

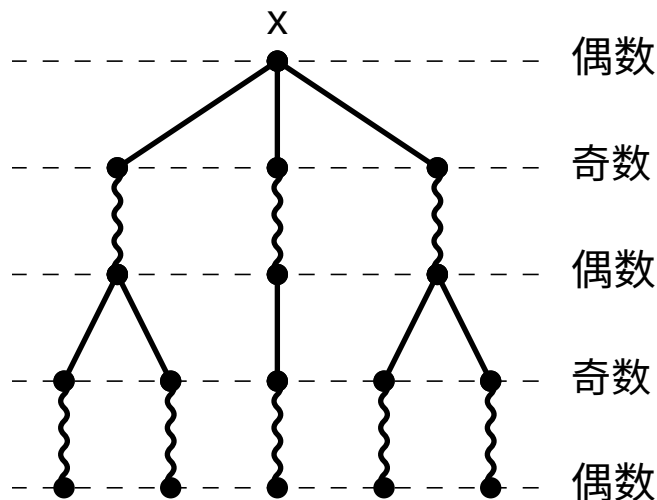


图2.4：一个交替树。波浪线边是匹配边。

要找到交替路径或花，我们按照以下步骤进行。我们将所有暴露的顶点标记为偶数，并在此时保持所有其他顶点未标记。随着我们的进行，我们将标记更多的顶点为偶数，并标记一些顶点为奇数。我们还维护一个交替森林——一个图，其中每个连通分量都是由在匹配中进出交替的边构成的树。我们逐个处理偶数顶点，假设我们当前正在处理 u ，并考虑与 u 相邻的边。有几种可能性：

1. 如果存在一条边 (u, v) ，且 v 未标记，我们将 v 标记为奇数。由于 v 不能是暴露的（否则它已经是偶数了），我们将其“伴侣” w （即 (v, w) 是匹配的一条边）标记为偶数。（ w 之前没有被标记，因为我们总是同时标记匹配边的两个端点。）我们扩展了正在构建的交替树（见图2.4）。

2. 如果存在一条边 (u, v) ，且 v 被标记为偶数，并且 v 属于与 u 不同的交替树，我们找到了一条增广路径（只需从 u 和 v 分别沿着交替树向上遍历到它们的根节点），并沿着该路径增广匹配，然后从这个新的、更大的匹配重新开始。从 u 和 v 到它们的根节点的两条子路径跨越不相交的顶点集合，因此它们的并集加上 (u, v) 确实构成一个有效的增广路径。
3. 如果存在一条边 (u, v) ，其中 v 被标记为偶数，并且 v 属于与 u 相同的交替树，则从 u 和 v 到它们的共同（暴露的）根节点 x 的两条子路径连同 (u, v) 形成一个花。我们将花 B 缩成一个顶点 b 。观察到我们可以保持标记不变，只要我们让新的顶点 b 被标记为偶数。我们递归地在这个图 G/B 中找到一个最大匹配（这可能导致进一步的缩小），当算法终止时，我们使用定理2.2将其扩展为原始图中的更大匹配。这个更大的匹配不一定是最优的（见定理2.2后的备注），我们重复这个过程，以找到当前匹配的增广路径或花。

正确性。现在假设这些可能性都不再适用于任何一个偶数顶点。那么我们声称我们在当前图 $G' = (V, E')$ （通过连续收缩花 B_1, B_2, \dots, B_k 从原始图 G 获得）中找到了一个最大匹配 M' 。为了证明这一点，考虑 $U = \text{奇数}$ 并考虑 G' 的上界 (2)。由于 EVEN 顶点之间没有边（否则会应用到2或3），也没有边连接 EVEN 顶点和未标记的顶点（否则会应用到1），我们得知每个 EVEN 顶点在 $G' \setminus \text{奇数}$ 中都是一个（奇数大小的）连通分量。此外，我们有 $|M'| = |\text{奇数}| + \frac{1}{2}(|V'| - |\text{奇数}| - |\text{偶数}|)$ ，第二项来自于所有未标记顶点都已匹配的事实。因此，

$$\frac{1}{2}(|V'| + |\text{奇数}| - o(G' \setminus \text{奇数})) = \frac{1}{2}(|V'| + |\text{奇数}| - |\text{偶数}|) = |M'|,$$

这表明我们的匹配 M' 对于 G' 是最大的。应用定理2.2的重复，我们得到算法在 G 中构建了一个最大匹配。

运行时间。算法最多执行 n 次增广（匹配）其中 $n = |V|$ 。在两次增广之间，它最多会收缩一个花朵 $n/2$ 次，因为每次收缩至少会减少2个顶点。构建交替树所需的时间最多为 $O(m)$ ，其中 $m = |E|$ ，因此总时间为 $O(n^2m)$ 。

Tutte-Berge公式的正确性。我们现在可以证明定理2.1。正如我们所讨论的，Tutte-Berge公式对于算法结束时得到的图成立。

假设我们已经进行了 k 个花托收缩，并且令 $G_i = (V_i, E_i)$ 为收缩花托 B_1, \dots, B_i 后得到的图， M_i 为相应的匹配；索引

$i=0$ 对应于原始图。对于最终图 $G_k = (V_k, E_k)$ ，我们已经看到 Tutte-Berge 公式成立，因为

$$|M_k| = \frac{1}{2} (|V_k| + |U| - o(G_k \setminus U)),$$

其中 $|U|$ 是奇数，并且每个偶数顶点对应于 $G_k \setminus U$ 的一个奇数连通分量。现在，让我们看看当我们逐个取消收缩花朵时会发生什么，让我们进行逆向归纳。假设我们取消收缩花朵 B_i ，从图 G_i 到 G_{i-1} 。首先注意到 $|V_{i-1}| = |V_i| + |B_i| - 1$ 和 $|M_{i-1}| = |M_i| + \frac{1}{2}$ 。此外，当我们还原花 B_i 时，我们会向 $G_i \setminus U$ 的一个连通分量中添加一个偶数个顶点（即 $|B_i| - 1$ ），因此我们不会改变奇数（或偶数）连通分量的数量。因此，当我们用 $i-1$ 替换 i 时，等式的右边和左边都会发生变化

$$|M_i| = \frac{1}{2} (|V_i| + |U| - o(G_i \setminus U))$$

增加精确地 $\frac{1}{2}$ 个。因此，通过逆向归纳，我们可以证明对于每个 $j=0, \dots, k$ ，我们有 $|M_j| =$

$$\frac{1}{2} (|V_j| + |U| - o(G_j \setminus U)),$$

并且 Tutte-Berge 公式对于原图（对于 $j=0$ ）成立。这证明了定理 2.1。

Tutte-Berge 公式意味着如果对于每个集合 U ， $G \setminus U$ 的奇数连通分量的数量至多为 $|U|$ ，则图有一个完美匹配。这被称为 Tutte 的匹配定理。

定理 2.3 (Tutte 的匹配定理) 如果对于所有 $U \subseteq V$ ，我们有 $o(G \setminus U) \leq |U|$ ，则 G 有一个完美匹配。

练习

练习 2-2. 设 $G = (V, E)$ 是任意图。给定一个集合 $S \subseteq V$ ，假设存在一个匹配 M 覆盖 S （即 S 是 M 中匹配顶点的子集）。证明存在一个覆盖 S 的最大匹配 M^* 。

练习 2-3. 设 U 为图中的任意最小化集合。设 K_1, \dots, K_k 为 $G \setminus U$ 的连通分量。证明对于任意最大匹配 M ，我们必须有

1. M 包含恰好 $\lfloor \frac{|K_i|}{2} \rfloor$ 条边来自 $G[K_i]$ （由 K_i 中的顶点诱导的子图 $G[K_i]$ ），即对于偶数分量 K_i ， $G[K_i]$ 是完美匹配的，对于奇数分量则是近似完美匹配的。

2. 每个顶点 $u \in U$ 都与 $G \setminus U$ 中的奇数分量的一个顶点 v 匹配。

3. 唯一未匹配的顶点必须位于 $G \setminus U$ 中的奇数分量 K_i 中。

练习2-4. 在图中可能存在多个最小化集合 U 吗? 要么给出一个具有多个达到最小值的集合 U 的例子, 要么证明集合 U 是唯一的。

练习2-5. 给定一个图 $G = (V, E)$, 一个非必要顶点是一个顶点 v , 满足存在一个 G 的最大匹配不包含 v 。令 B 为 G 中所有非必要顶点的集合 (例如, 如果 G 有一个完美匹配, 则 $B = \emptyset$)。令 C 表示不在 B 中但与 B 中至少一个顶点相邻的顶点的集合 (因此, 如果 $B = \emptyset$, 则 $C = \emptyset$)。令 $D = V \setminus (B \cup C)$ 。三元组 $\{B, C, D\}$ 被称为 G 的 Edmonds-Gallai 分割。证明 $U = C$ 是 Tutte-Berge 公式中的最小化器。(特别地, 在 Tutte-Berge 公式中, 我们可以假设 U 是这样的, 即 $G \setminus U$ 的奇数连通分量的并集正好是非必要顶点的集合。)

练习2-6. 证明任意3-正则2-边连通图 $G = (V, E)$ (不一定是二分图) 都有完美匹配。(一个2-边连通图在任意割集中至少有2条边; 割集是指 S 和 $V \setminus S$ 之间的边, 其中 S 是某个顶点集合。)

练习2-7. 图 $G = (V, E)$ 被称为因子临界图, 如果对于所有的 $v \in V$, 都有 $G \setminus \{v\}$ 包含一个完美匹配。在下面的 (a) 和 (b) 部分中, G 是一个因子临界图。

1. 假设 U 是 G 的 Tutte-Berge 公式的任意最小化集合。证明 $U = \emptyset$ 。(提示: 参见练习 2-3。)
2. 推断出当 Edmonds 算法终止时, 从 G 收缩花朵得到的最终图必定是一个单个顶点。
3. 给定一个图 $H = (V, E)$, 一条耳朵是一条路径 $v_0 - v_1 - v_2 - \dots - v_k$, 其端点 (v_0 和 v_k) 在 V 中, 而内部顶点 (v_i , 其中 $1 \leq i \leq k-1$) 不在 V 中。我们允许 v_0 等于 v_k , 这种情况下路径将变成一个循环。将这条耳朵添加到 H 中会在 $V \cup \{v_1, \dots, v_{k-1}\}$ 上创建一个新的图。当 $k=1$ 时的特殊情况 (一个 '平凡' 的耳朵) 意味着在 H 中添加一条边。如果 k 是奇数, 则称这条耳朵为奇耳朵, 否则称为偶耳朵; 例如, 一个平凡的耳朵是奇数。
 - (a) 让 G 是一个可以通过从一个奇数环开始并重复添加奇数耳朵构建的图。证明 G 是因子临界的。
 - (b) 证明任何因子临界图都可以通过从一个奇数环开始并重复添加奇数耳朵构建。

3. 线性规划和多面体组合学

总结线性规划和多面体组合学入门讲座中所见的内容。

定义3.1 半空间是 \mathbb{R}^n 中形式为 $\{x \in \mathbb{R}^n : a^T x \leq b\}$ 的集合，其中 $a \in \mathbb{R}^n$ 和 $b \in \mathbb{R}$ 是某个向量。

定义3.2 多面体是有限个半空间的交集： $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ 。

定义3.3 多面体是一个有界的多面体。

定义3.4 如果 P 是 \mathbb{R}^n 中的多面体，则 P 的投影 $P_k \subseteq \mathbb{R}^{n-1}$ 被定义为 $\{y = (x_1, x_2, \dots, x_{k-1}, x_{k+1}, \dots, x_n) : x \in P \text{ 对于某个 } x_k \in \mathbb{R}\}$ 。

这是一个线性空间上的投影的特殊情况（在这里，我们只考虑坐标投影）。通过反复投影，我们可以消除任何坐标的子集。我们声称 P_k 也是一个多面体，并且可以通过

给出 P_k 的显式描述来证明这一点，该描述基于线性不等式。为此，人们使用 *Fourier-Motzkin* 消元。设 $P = \{x : Ax \leq b\}$ ，让

- $S_+ = \{i : a_{ik} > 0\}$,
- $S_- = \{i : a_{ik} < 0\}$,
- $S_0 = \{i : a_{ik} = 0\}$.

显然， P_k 中的任何元素必须满足不等式 $a_i^T x \leq b_i$ 对于所有 $i \in S_0$ （这些不等式不涉及 x_k ）。同样，我们可以将 S_+ 中的一个不等式和 S_- 中的一个不等式线性组合，以消除 x_k 的系数。这表明以下不等式成立：

$$a_{ik} \left(\sum_j a_{lj} x_j \right) - a_{lk} \left(\sum_j a_{ij} x_j \right) \leq a_{ik} b_l - a_{lk} b_i \quad (1)$$

对于 $i \in S_+$ 和 $l \in S_-$ 的所有元素，上述不等式都成立。反过来，对于任何满足对于所有 $i \in S_+$ 和 $l \in S_-$ 以及式 (1) 的 $(x_1, x_2, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$

$$\text{对于所有的 } i \in S_0, \text{ 有 } a_i^T x \leq b_i \quad (2)$$

我们可以找到一个值 x_k 使得结果 x 属于 P （通过查看每个约束对 x_k 施加的界限，并证明最大下界小于最小上界）。这表明 P_k 由 (1) 和 (2) 描述，并且因此是一个多面体。

定义3.5 给定点 $a^{(1)}, a^{(2)}, \dots, a^{(k)} \in \mathbb{R}^n$,

- 线性组合是 $\sum_i \lambda_i a^{(i)}$ 其中 $\lambda_i \in \mathbb{R}$ 对于所有的 i ,
- 仿射组合是 $\sum_i \lambda_i a^{(i)}$ 其中 $\lambda_i \in \mathbb{R}$ 和 $\sum_i \lambda_i = 1$,
- 一个圆锥组合是 $\sum_i \lambda_i a^{(i)}$ 其中 λ_i 对于所有的 $i \geq 0$,
- 一个凸组合是 $\sum_i \lambda_i a^{(i)}$ 其中 λ_i 对于所有的 $i \geq 0$ 和 $\sum_i \lambda_i = 1$.

所有由 S 元素的线性组合构成的集合称为 S 的线性包, 表示为 $\text{lin}(S)$ 。类似地, 通过将线性替换为仿射、圆锥或凸, 我们定义了仿射包, $\text{aff}(S)$, 圆锥包, $\text{cone}(S)$ 和凸包, $\text{conv}(S)$ 。我们可以给出一个多面体的等价定义。

定义3.6 一个多面体是有限点集的凸包。

定义3.6蕴含定义3.3的事实可以如下看出。取 P 为有限点集 $\{a^{(k)}\}_{k \in [m]}$ 的凸包。为了证明 P 可以描述为有限个超平面的交集, 我们可以反复应用Fourier-Motzkin消元在

$$\begin{aligned} x - \sum_k \lambda_k a^{(k)} &= 0 \\ \sum_k \lambda_k &= 1 \\ \lambda_k &\geq 0 \end{aligned}$$

消除所有变量 λ_k 并仅保留变量 x 。此外, P 是有界的, 因为对于任意 $x \in P$, 我们有

$$\|x\| = \left\| \sum_k \lambda_k a^{(k)} \right\| \leq \sum_k \lambda_k \|a^{(k)}\| \leq \max_k \|a^{(k)}\|.$$

逆命题将在本讲义的后面证明。

3.1 不等式系统的可解性

在线性代数中, 我们看到对于 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $Ax = b$ 没有解 $x \in \mathbb{R}^n$ 当且仅当存在一个 $y \in \mathbb{R}^m$ 使得 $A^T y = 0$ 和 $b^T y > 0$ (在18.06的符号/术语中, 这等价于说列空间 $C(A)$ 正交于左零空间 $N(A^T)$)。

对于线性不等式系统, 可以陈述一个类似的交替定理。

定理3.1 (交替定理) 如果 $Ax \leq b$ 没有解 $x \in \mathbb{R}^n$ 当且仅当存在 $y \in \mathbb{R}^m$ 使得 $y \geq 0$, $A^T y = 0$ 且 $b^T y < 0$ 。

可以很容易地证明这两个系统都无解，否则 $0 > b^T y = y^T b \geq y^T A x = 0^T x = 0$ 。对于另一个方向，可以取一个无解的系统 $Ax \leq b$ 并重复使用傅里叶-莫特金消元法来消除所有变量，从而得到一个形式为 $0^T x \leq c$ 的不等式，其中 $c < 0$ 。在这个过程中，已经得到了一个具有所需属性的向量 y （因为傅里叶-莫特金只执行线性不等式的非负组合）。

上述定理的另一个版本是Farkas引理：

引理3.2 $Ax = b, x \geq 0$ 没有解当且仅当存在 y 满足 $A^T y \geq 0$ 且 $b^T y < 0$ 。

练习3-1。从替代定理证明Farkas引理。

3.2 线性规划基础知识

线性规划（LP）是在一个多面体上最小化或最大化线性函数的问题：

$$\begin{aligned} & \text{Max } c^T x \\ & \text{subject to:} \\ (P) \quad & Ax \leq b, \end{aligned}$$

其中 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, 变量 x 在 \mathbb{R}^n 中。任何满足 $Ax \leq b$ 的 x 被称为可行解。如果没有满足 $Ax \leq b$ 的 x ，我们称线性规划是不可行的，其最优值为 $-\infty$ （因为我们在一个空集合上最大化）。如果线性规划的目标函数值可以任意大，我们称其为无界的，其最优值为 $+\infty$ ；否则它是有界的。如果既不是不可行的，也不是无界的，那么其最优值是有限的。

其他等价形式也涉及等式，或者非负约束 $x \geq 0$ 。在讨论线性规划算法（尤其是单纯形算法）时，经常考虑的一个版本是 $\min\{c^T x : Ax = b, x \geq 0\}$ 。

另一个线性规划问题，对偶问题（D）起着关键作用：

$$\begin{aligned} & \text{最小化 } b^T y \\ & \text{约束条件为:} \\ (D) \quad & \text{满足约束条件 } A^T y = c \\ & y \geq 0. \end{aligned}$$

（D）是对偶问题，（P）是原始问题。对偶问题的术语类似。如果（D）没有可行解，则称其为不可行问题，其最优值为 $+\infty$ （因为我们在一个空集合上进行最小化）。如果（D）无界（即其值可以任意小），则其最优值为 $-\infty$ 。

原始空间和对偶空间不应混淆。如果 A 是 $m \times n$ 的矩阵，那么我们有 n 个原始变量和 m 个对偶变量。

弱对偶性很明显：对于任意可行解 x 和 y 对于 (P) 和 (D)，我们有 $c^T x \leq b^T y$ 。对偶问题的目的是得到任意原始解的上界。例如，为了得到不等式 $y^T A x \leq b^T y$ ，我们需要 $y \geq 0$ ，因为我们知道 $A x \leq b$ 。特别地，弱对偶性意味着如果原始问题无界，则对偶问题必定无可行解。

强对偶性是线性规划中最重要的结果；它表明我们可以通过展示一个对偶问题的最优解 y 来证明一个原始解 x 的最优性。

定理3.3 (强对偶性) 假设 (P) 和 (D) 是可行的，令 z^* 为原始问题的最优值， w^* 为对偶问题的最优值。那么 $z^* = w^*$ 。

强对偶性的一个证明是通过将一大组不等式写成关于 x 和 y 的形式，其中 (i) x 是原始可行的，(ii) y 是对偶可行的，(iii) $c^T x \geq b^T y$ 。然后使用备择定理来证明这个不等式系统的不可行性将会与 (P) 或 (D) 的可行性相矛盾。

证明： 假设 x^* 是原始问题的可行解， y^* 是对偶问题的可行解。证明采用反证法。由于弱对偶性，这意味着不存在解 $x \in \mathbb{R}^n$ 和 $y \in \mathbb{R}^m$ 使得

$$\begin{cases} Ax & \leq b \\ & A^T y = c \\ & -Iy \leq 0 \\ -c^T x + b^T y & \leq 0 \end{cases}$$

通过替代定理或Farkas引理的一个变种（对于不等式和等式的组合情况），我们得出结论，必然存在 $s \in \mathbb{R}^m$, $t \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}$ such that:

$$\begin{aligned} s &\geq 0 \\ u &\geq 0 \\ v &\geq 0 \\ A^T s - v c &= 0 \\ A t - u + v b &= 0 \\ b^T s + c^T t &< 0. \end{aligned}$$

我们区分两种情况。

情况1: $v=0$. 那么 s 满足 $s \geq 0$ 和 $A^T s = 0$.. 这意味着，对于任何 $\alpha \geq 0$, $y^* + \alpha s$ 是对偶问题的可行解。同样地， $A t = u \geq 0$ ，因此，对于任何 $\alpha \geq 0$ ，我们有 $x^* - \alpha t$ 是原始问题的可行解。根据弱对偶性，这意味着，对于任何 $\alpha \geq 0$ ，我们有

$$c^T(x^* - \alpha t) \leq b^T(y^* + \alpha s)$$

或者

$$c^T x^* - b^T y^* \leq \alpha(b^T s + c^T t).$$

右侧趋于 $-\infty$ 当 α 趋于 ∞ ，这与左侧固定的情况矛盾。

情况2: $v > 0$ 。通过整除 v (并重新命名所有变量)，我们得到存在 $s \geq 0$, $u \geq 0$, 使得

$$\begin{aligned} A^T s &= c \\ At - u &= -b \\ b^T s + c^T t &< 0. \end{aligned}$$

这意味着 s 是对偶可行的， $-t$ 是原始可行的，因此根据弱对偶性有 $c^T(-t) \leq b^T s$ 与 $b^T s + c^T t < 0$ 矛盾。 △

练习3-2. 证明对偶的对偶是原始问题。

练习3-3. 证明只需要原始问题或对偶问题中的一个可行，就能保持强对偶性。更准确地说，如果原始问题可行但对偶问题不可行，证明原始问题将无界，意味着 $z^* = w^* = +\infty$ 。

观察 $c^T x = y^T Ax \leq b^T y$ ，我们发现要使 $c^T x$ 和 $b^T y$ 相等，我们需要互补松弛条件:

定理3.4 (互补松弛条件) 如果 x 在 (P) 中可行， y 在 (D) 中可行，则 x 在 (P) 中是最优解或 y 在 (D) 中是最优解，当且仅当对于所有的 i ，要么 $y_i = 0$ 或 $\sum_j a_{ij} x_j = b_i$ (或两者都是)

线性规划可以使用单纯形法来解决；这在这些讲义中不会被解释。目前尚无已知的单纯形法变体可以在多项式时间内运行，但是存在其他多项式时间算法用于线性规划，即椭圆算法和内点算法类。

3.3 多面体的面

定义 3.7 $\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ 线性无关，如果 $\sum \lambda_i = 0$ 对于所有 $i \in K$ 。

$\lambda_i a^{(i)} = 0$ 则意味着

定义 3.8 $\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ 仿射无关，如果 $\sum \lambda_i = 0$ 对于所有 $i \in K$ 。

$\lambda_i a^{(i)} = 0$ 并且 $\sum \lambda_i = 1$ 意味着 λ

观察到 $\{a^{(i)} \in \mathbb{R}^n : i \in K\}$ 是仿射独立的当且仅当

$$\left\{ \begin{bmatrix} a^{(i)} \\ 1 \end{bmatrix} \in \mathbb{R}^{n+1} : i \in K \right\}$$

是线性独立的.

定义3.9多面体 P 的维数, $\dim(P)$, 是 P 中仿射独立点的最大数量减1.

(这与仿射包络的维数相同 $\text{aff}(S)$.) 维数可以是-1 (如果 P 为空), 0 (当 P 只包含一个点), 1 (当 P 是线段), 和最多是 n 当 P 在 \mathbb{R}^n 中仿射张成时. 在后一种情况下, 我们说 P 是满维度的. 一个在 \mathbb{R}^3 中的立方体的维数是3, \mathbb{R}^3 本身的维数也是3 (它是一个平凡的多面体).

定义 3.10 $\alpha^T x \leq \beta$ 是对于 P 的一个有效不等式, 如果对于所有的 $x \in P$ 都有 $\alpha^T x \leq \beta$.

观察到对于一个不等式要对 $\text{conv}(S)$ 有效, 我们只需要确保它被 S 中的所有元素满足, 因为这将意味着这个不等式也被 $\text{conv}(S) \setminus S$ 中的点满足. 当处理组合优化中的匹配或生成树等组合对象的凸包时, 这个观察是很重要的.

定义 3.11 多面体 P 的一个面是 $\{x \in P : \alpha^T x = \beta\}$, 其中 $\alpha^T x \leq \beta$ 是 P 的一些有效不等式.

根据定义, 所有的面都是多面体. 空面 (维度为 -1) 是微不足道的, 多面体 P 整体 (对应于有效不等式 $0^T x \leq 0$) 也是如此. 非微不足道的是那些维度在 0 到 $\dim(P) - 1$ 之间的面. 维度为 0 的面被称为极点或顶点, 维度为 1 的面被称为边, 维度为 $\dim(P) - 1$ 的面被称为面. 有时, 人们使用维度为 $\dim(P) - 2$ 的棱来表示面.

练习3-4. 列出立方体的28个面 $P = \{x \in \mathbb{R}^3 : 0 \leq x_i \leq 1 \text{ for } i = 1, 2, 3\}$.

虽然有无限多个有效的不等式, 但只有有限多个面.

定理3.5 设 $A \in \mathbb{R}^{m \times n}$. 那么 $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ 的任何非空面对应于以下解集

$$\sum_j \text{对于所有 } i \in I, \text{ 有 } a_{ij}x_j = b_i$$

$$\sum_j \text{对于所有 } i \notin I, \text{ 有 } a_{ij}x_j \leq b_i,$$

对于某个集合 $I \subseteq \{1, \dots, m\}$. 因此, P 的非空面的数量最多为 2^m .

证明：考虑任意有效的不等式 $\alpha^T x \leq \beta$ 。假设相应的面 F 非空。因此 F 是所有最优解

$$\begin{aligned} & \text{最大化 } \alpha^T x \\ & \text{的条件是:} \\ (P) \quad & Ax \leq b. \end{aligned}$$

选择对偶线性规划的最优解 y^* 。根据互补松弛条件，面 F 由那些满足 $a_i^T x = b_i$ 的元素 x of P 定义，其中 $i \in I = \{i : y_i^* > 0\}$ 。因此 F 由以下方式定义

$$\begin{aligned} \sum_j a_{ij} x_j &= b_i \text{ 对于所有 } i \in I \\ \sum_j a_{ij} x_j &\leq b_i \text{ 对于所有 } i \notin I \end{aligned}$$

由于 I 有 2^m 种可能性，所以最多有 2^m 个非空面。 \triangle

定理3.5中给出的面的数量对于多面体是紧密的（见下面的练习），但对于所谓的上界定理中的多面体可以大大改进（这些笔记中没有给出）。

练习3-5. 证明 P 有 $2^n + 1$ 个面。
维度为 k 的面有多少个？

对于极点（维度为0的面），特征更加强烈（我们不需要不等式）：

定理3.6 设 x^* 是 $P = \{x : Ax \leq b\}$ 的一个极点。那么存在 I 使得 x^* 是唯一的解

$$\sum_j a_{ij} x_j = b_i \text{ 对于所有 } i \in I.$$

证明：给定一个极点 x^* ，定义 $I = \{i : \sum_j a_{ij} x_j^* = b_i\}$ 。这意味着对于不在 I 中的 i ，我们有 $\sum_j a_{ij} x_j^* < b_i$ 。

根据定理3.5，我们知道 x^* 是唯一确定的

$$\sum_j a_{ij} x_j = b_i \text{ 对于所有 } i \in I \tag{3}$$

$$\sum_j a_{ij} x_j \leq b_i \text{ 对于所有 } i \notin I. \tag{4}$$

现在假设存在另一个解 \hat{x} 当我们只考虑 $i \in I$ 的等式时。

然后由于 $\sum_j a_{ij} x_j^* < b_i$ ，我们得到 $(1 - \epsilon)x^* + \epsilon \hat{x}$ 也满足 ϵ 足够小的情况下的 (3) 和 (4)。矛盾（因为该面应该包含一个点）。 \triangle

如果 P 被给定为 $\{x : Ax = b, x \geq 0\}$ (通常是这种情况), 定理仍然适用 (因为我们仍然有一组不等式)。在这种情况下, 定理表明每个极端点 x^* 可以通过将某些变量设置为 0, 并求解得到结果的唯一解的等式系统来获得。不失一般性, 我们可以从 $Ax = b$ 中删除冗余的等式; 这意味着我们可以假设 A 具有完整的行秩 ($\text{rank}(A) = m$ 对于 $A \in \mathbb{R}^{m \times n}$)。令 N 表示非基本变量的索引我们将其设置为 0, 并将 B 表示剩余索引 (所谓的基本变量), 我们可以将 x^* 分成 x_B^* 和 x_N^* (对应于这两组变量) 并重写 $Ax = b$ 为 $A_B x_B + A_N x_N = b$, 其中 A_B 和 A_N 是 A 在 B 和 N 索引中的限制。定理表明 x^* 是 $A_B x_B + A_N x_N = 0$ 的唯一解和 $x_N = 0$, 这意味着 $x_N^* = 0$ 和 $A_B x_B^* = b$ 。后一个系统必须有一个唯一解, 这意味着 A_B 必须具有完整的列秩 ($\text{rank}(A_B) = |B|$)。由于 A 本身具有秩 m , 我们有 $|B| \leq m$ 并且我们可以扩展 B 以包括 N 的索引结果 B 满足 (i) $|B| = m$ 和 (ii) A_B 是一个 $m \times m$ 可逆矩阵 (因此存在仍然有 $A_B x_B = b$ 的唯一解)。在线性规划术语中, 基本可行解或 bfs $\{x : Ax = b, x \geq 0\}$ 通过选择一组 $|B| = m$ 的索引来获得 A_B 可逆, 并让 $x_B = A_B^{-1}b$ 和 $x_N = 0$, 其中 N 是不在 B 中的索引。因此, 我们已经证明了所有极端点都是 bfs, 反之亦然。注意, 两个不同的基 B 可能导致相同的极端点, 因为在上述讨论中可能有许多扩展 A_B 为一个 $m \times m$ 可逆矩阵的方式。

我们可以从定理 3.5 推导出一个结论:

推论 3.7 一个非空多面体

P 的最大 (包含关系) 的非平凡面是其面。

对于顶点, 需要满足一个额外的条件:

推论 3.8 如果 $\text{rank}(A) = n$ (满列秩), 则一个非空多面体 $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ 的最小 (包含关系) 的非平凡面是其顶点。

练习 3-7 表明满秩条件是必要的。

这意味着, 如果一个线性规划 $\max\{c^T x : x \in P\}$ with $P = \{x : Ax \leq b\}$ 是可行的, 有界的, 并且 $\text{rank}(A) = n$, 那么存在一个最优解是多面体 P 的一个顶点 (实际上, 所有最优解的集合定义了一个面——最优面——如果这个面本身不是多面体 P 的一个顶点, 它必须包含多面体 P 的顶点)。我们现在证明推论 3.8。

证明: 设 F 为 P 的最小 (包含关系) 的非平凡面。这意味着我们有一个集合 I 使得

$$F = \{x : \begin{array}{ll} a_i^T x = b_i & \forall i \in I \\ a_j^T x \leq b_j & \forall j \notin I \end{array}\}$$

并且将任何元素添加到 I 会使该集合为空。考虑两种情况。要么 $F = \{x \in \mathbb{R}^n : a_i^T x = b_i \text{ for } i \in I\}$, 要么不是。在第一种情况下, 这意味着对于每个 $j \notin I$, 我们有 $a_j \in \text{lin}(\{a_i : i \in I\})$ (否则就会存在一个解 x 满足 $a_i^T x = b_i$ 对于所有 $i \in I$ 和

$a_j^T x = b_j + 1$, 因此不在 F 中), 因此由于 $\text{rank}(A) = n$, 我们知道该系统 $a_i^T x = b_i$ 对于所有 $i \in I$ 有唯一解, 因此 F 是一个顶点。

另一方面, 如果 $F = \{x \in \mathbb{R}^n : a_i^T x = b_i \text{ for } i \in I\}$ 那么让 $j \notin I$ such that there exists \tilde{x} with

$$\begin{aligned} a_i^T \tilde{x} &= b_i \quad i \in I \\ a_j^T \tilde{x} &> b_j. \end{aligned}$$

由于 F 不是平凡的, 存在 $\hat{x} \in F$. 特别地, \hat{x} satisfies

$$\begin{aligned} a_i^T \hat{x} &= b_i \quad i \in I \\ a_j^T \hat{x} &\leq b_j. \end{aligned}$$

考虑一个凸组合 $x' = \lambda \tilde{x} + (1 - \lambda) \hat{x}$. 考虑最大的 λ such that x' is in P . 这是明确定义的, 因为 $\lambda = 0$ 给出了一个在 P 中的点, 而 $\lambda = 1$ 则不是。相应的 x' satisfies $a_i^T x' = b_i$ for $i \in I \cup \{k\}$ for some k (possibly j), 与 I 的最大性矛盾。

△

现在我们回到定义3.3和3.6之间的等价性, 并声称我们可以证明定义3.3蕴含定义3.6。

定理3.9 如果 $P = \{x : Ax \leq b\}$ 是有界的, 则 $P = \text{conv}(X)$ 其中 X 是 P 的极点集。

这是一个使用替代定理的好练习。

证明: 由于 $X \subseteq P$, 我们有 $\text{conv}(X) \subseteq P$. 假设, 通过反证法, 我们不满足相等关系。那么必然存在一个 $\tilde{x} \in P \setminus \text{conv}(X)$ 。 $\tilde{x} \notin \text{conv}(X)$ 的事实意味着没有解满足:

$$\begin{cases} \sum_{v \in X} \lambda_v v = \tilde{x} \\ \sum_{v \in X} \lambda_v = 1 \\ \lambda_v \geq 0 \end{cases} \quad v \in X.$$

根据替代定理, 这意味着 $\exists c \in \mathbb{R}^n, t \in \mathbb{R}$:

$$\begin{cases} t + \sum_{j=1}^n c_j v_j \geq 0 \quad \forall v \in X \\ t + \sum_{j=1}^n c_j \tilde{x}_j < 0. \end{cases}$$

由于 P 是有界的, $\min\{c^T x : x \in P\}$ 是有限的 (假设等于 z^*), 并且由 $c^T x \geq z^*$ 引起的面是非空的, 但不包含任何顶点 (因为所有顶点都被 \tilde{x} 通过上述不等式支配)。这与推论3.8矛盾。确实, 观察到推论3.8适用。如果 $\text{rank}(A) < n$, 则存在 $y \neq 0$ 使得 $Ay = 0$, 这将与 P 的有界性相矛盾 (因为我们可以无限地朝 y 的方向前进)。

△

当描述一个多面体 P 的线性不等式时, 只需要那些定义 P 的面的不等式。这在接下来的几个定理中说明。如果系统中的不等式 $Ax \leq b$ 对应的多面体去掉这个不等式后没有改变, 则称该不等式是多余的。对于 $P = \{x : Ax \leq b\}$, 我们用 $I_=\text{表示那些满足对于所有的 } x \in P, \text{ 有 } a_i^T x = b_i \text{ 的索引 } i, \text{ 用 } I_<\text{表示剩下的索引 (即存在 } x \in P \text{ 使得 } a_i^T x < b_i)$ 。

这个定理表明面是足够的:

定理3.10 如果与 $a_i^T x \leq b_i$ 对应的面不是一个面，则该不等式是多余的。

而这个则表明了面是必要的：

定理3.11 如果 F 是 P 的一个面，则必然存在 $i \in I_<$ 使得由 $a_i^T x \leq b_i$ 诱导的面恰好是 F 。

在 P 的最小描述中，我们必须有一组线性无关的等式，以及对于 P 的每个面恰好一个不等式。

练习

练习3-6. 证明推论3.7。

练习3-7. 证明如果 $\text{rank}(A) < n$ ，则 $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ 没有顶点。

练习3-8. 证明集合

$Q = \{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$ 的顶点是 P 的顶点的子集。

(特别地，这意味着如果路径 P 的所有顶点都属于 $\{0, 1\}^n$ ，那么路径 Q 的顶点也是如此。)

练习3-9. 给定一个多面体 P 的两个极点 a 和 b ，如果它们之间的线段形成多面体 P 的一个边（即维度为1的面），我们称它们是相邻的。这可以通过说 a 和 b 在多面体 P 上是相邻的来重新表述，当且仅当存在一个代价函数 c ，使得 a 和 b 是 P 上唯一的两个极点，使得 $c^T x$ 在 P 上最小化。

考虑一个（不一定是二分图的）图 G 中所有完美匹配的凸包（多面体） P 。给出两个匹配 M_1 和 M_2 在这个多面体上相邻的必要和充分条件（提示：考虑 $M_1 \triangle M_2 = (M_1 \setminus M_2) \cup (M_2 \setminus M_1)$ ），并证明你的条件是必要且充分的。

练习题3-10. 证明一个多面体 P 的两个顶点 u 和 v 是相邻的，当且仅当以唯一的方式将它们的中点 (¹ $\frac{1}{2}(u + v)$) 表示为 P 的顶点的凸组合时。

3.4 多面体组合学

简单来说，多面体组合学研究与组合优化问题（如匹配问题等）相关的离散集合所关联的多面体或多胞体。如果我们有一个离散集合 X （比如图中匹配的关联向量集合，或者图的生成树的关联向量集合，或者图中稳定集合的关联向量集合），我们可以考虑 $\text{conv}(X)$ 并试图用

¹图 $G = (V, E)$ 中的一个顶点集合 S 是稳定的，如果 S 中的任意两个顶点之间没有边。

线性不等式的优化问题。这在应用线性规划的机制时非常有用。

然而，在某些（大多数）情况下，很难准确描述所有不等式的集合 $\text{conv}(X)$ ；这在优化问题 X 上很困难，这个陈述可以在计算复杂性的背景下精确描述。对于匹配、生成树和其他一些结构（其对应的优化问题是多项式可解的），我们将能够用线性不等式来描述它们的凸包。给定一个集合 X 和一个提议的不等式系统 $P = \{x : Ax \leq b\}$ ，通常很容易检查 $\text{conv}(X) \subseteq P$ 。事实上，我们只需要检查 X 的每个成员是否满足 P 的每个不等式描述。反向包含更加困难。

这里有三种常用的技巧来证明 $P \subseteq \text{conv}(X)$ （如果是真的！）（一旦我们知道 $\text{conv}(X) \subseteq P$ ）。

1. 算法上的证明。这涉及到线性规划的对偶性。这就是我们在关于分配问题（二分图中的最小权重匹配）的笔记中所做的。一般来说，考虑任意的代价函数 c ，并考虑最大化 $c^T x$ 在 $x \in X$ 上的组合优化问题。我们知道：

$$\begin{aligned} \max\{c^T x : x \in X\} &= \max\{c^T x : x \in \text{conv}(X)\} \\ &\leq \max\{c^T x : Ax \leq b\} \\ &= \min\{b^T y : A^T y = c, y \geq 0\}, \end{aligned}$$

最后一个等式来自于强对偶性。如果我们能够展示一个解 $x \in X$ （比如说分配问题中的完美匹配的关联向量）和一个对偶可行解 y （分配问题中的值 u_i, v_j ）使得 $c^T x = b^T y$ ，那么我们就证明了我们在整个过程中都有等式成立，如果对于任意成本函数 c ，这个等式都成立，那么这意味着 $P = \text{conv}(X)$ 。

这通常是最复杂的方法，但也是最常用的方法。

2. 专注于极点。首先展示 $P = \{x : Ax \leq b\}$ 是有界的（因此是一个多面体），然后研究它的极点。如果我们能够证明 P 的每一个极点都在 X 中，那么我们就完成了，因为 $P = \text{conv}(\text{ext}(P)) \subseteq \text{conv}(X)$ ，其中 $\text{ext}(P)$ 表示 P 的极点（参见定理3.9）。假设 P 是有界的是为了展示确实 $P = \text{conv}(\text{ext}(P))$ （如果 P 是无界的，则不成立）。

对于二分图匹配的凸包，这可以很容易地完成，这导致了完全单调矩阵（TU）的概念，请参见下一节。

3. 专注于 $\text{conv}(X)$ 的面。这通常导致最短和最干净的证明。假设我们提出的 P 的形式为 $\{x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$ 。我们已经证明了 $\text{conv}(X) \subseteq P$ ，并且我们想要证明 $P \subseteq \text{conv}(X)$ 。

首先，我们需要证明我们没有遗漏任何等式。这可以通过证明 $\dim(\text{conv}(X)) = \dim(P)$ 来完成。我们已经知道 $\dim(\text{conv}(X)) \leq \dim(P)$ （因为 $\text{conv}(X) \subseteq P$ ），因此我们需要证明 $\dim(\text{conv}(X)) \geq \dim(P)$ 。

这意味着如果 C 中有 $m - n$ 个线性无关的行, 则我们可以在 X 中找到 $m + 1$ 个仿射无关的点。

然后我们需要证明我们没有漏掉一个能够引出 $\text{conv}(X)$ 的有效不等式。考虑任意一个对于 $\text{conv}(X)$ 有效的不等式 $\alpha^T x \leq \beta$, 其中 $\alpha \neq 0$ 。我们可以假设 α 是 $\mathbb{R}^n \setminus \{0\}$ 中的任意向量, 且 $\beta = \max\{\alpha^T x : x \in \text{conv}(X)\}$ 。这个不等式所定义的 $\text{conv}(X)$ 的面是 $F = \text{conv}(\{x \in X : \alpha^T x = \beta\})$ 。假设这是一个非平凡的面; 当且仅当 α 不在 C 的行空间中时, 这种情况会发生。我们需要确保如果 F 是一个面, 则我们在 P 的描述中有一个表示它的不等式。我们将证明如果 F 是非平凡的, 则我们可以在 P 的描述中找到一个不等式 $a^T_i x \leq b_i$, 使得 (i) $F \subseteq \{x : a^T_i x = b_i\}$, 并且 (ii) $a^T_i x \leq b_i$ 定义了 P 的一个非平凡的面 (如果 P 是满维的, 则不需要第二个条件), 或者简单地说, 最优解 $\{\alpha^T x : x \in X\}$ 满足 $a^T_i x = b_i$, 并且这个不等式不被 P 中的所有点满足。这意味着如果 F 是一个面, 根据极大性, 我们在我们的描述中有 F 的一个代表。

这是一个非常简单而强大的技术, 最好通过一个例子来说明。

例子。让 $X = \{(\sigma(1), \sigma(2), \dots, \sigma(n)) : \sigma \text{ 是 } \{1, 2, \dots, n\} \text{ 的一个排列}\}$ 。我们声称

$$\text{conv}(X) = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = \binom{n+1}{2}, \sum_{i \in S} x_i \geq \binom{|S|+1}{2} \text{ } S \subseteq \{1, \dots, n\}\}.$$

这被称为排列多面体。

这里的 $\text{conv}(X)$ 不是满维的; 我们只需要证明在 $\text{conv}(P)$ 的描述中没有遗漏任何面 and 任何等式。对于等式, 这很容易看出, 因为可以很容易地展示出 X 中的仿射独立的排列。对于面, 假设 $\alpha^T x \leq \beta$ 定义了一个非平凡的面 F of $\text{conv}(X)$ 。考虑最大化 $\alpha^T x$ 在所有排列 x 上。让 $S = \arg \min\{\alpha_i\}$; 根据我们的假设, F 是非平凡的, 我们有 $S = \{1, 2, \dots, n\}$ (否则, 我们将有等式 $\sum_{i=1}^n x_i =$

$\binom{n+1}{2}$) 此外, 很容易通过交换论证看出, 任何使得关联向量 x 最大化 $\alpha^T x$ 的排列 σ 都需要满足 $\sigma(i) \in \{1, 2, \dots, |S|\}$ 对于 $i \in S$, 换句话说, 它满足不等式 $\sum_{i \in S} x_i \geq \binom{|S|+1}{2}$ 在等式中取等 (这是一个非平凡的面, 因为存在不满足等式的排列)。因此 F 包含在我们描述中对应于不平凡的面中, 因此我们的描述包含了所有的不等式。这就是我们需要证明的。就是这样!

练习

练习3-11。考虑集合 $X = \{(\sigma(1), \sigma(2), \dots, \sigma(n)) : \sigma \text{ 是 } \{1, 2, \dots, n\} \text{ 的一个排列}\}$ 。证明 $\dim(\text{conv}(X)) = n - 1$ 。 (为了证明 $\dim(\text{conv}(X)) \geq n - 1$, 举个例子-

hibit *naffinely independent* permutations σ (并证明它们是仿射独立的)。

练习3-12。图中的一个稳定集 S (有时也称为独立集) 是一个顶点的集合, 使得 S 中的任意两个顶点之间没有边。如果我们用 P 表示 $G = (V, E)$ 的所有稳定集的凸包 (关于关联向量), 那么对于 P 来说, 对于任意边 $(i, j) \in E$, 不等式 $x_i + x_j \leq 1$ 是有效的。

1. 给出一个图 G , 使得 P 不等于

$$\{x \in \mathbb{R}^{|V|} : \text{对于所有 } (i, j) \in E, \quad x_i + x_j \leq 1 \\ x_i \geq 0 \quad \text{对于所有 } i \in V\}$$

2. 证明如果图 G 是二分图, 则 P 相等

$$\{x \in \mathbb{R}^{|V|} : \text{对于所有 } (i, j) \in E, \quad x_i + x_j \leq 1 \\ x_i \geq 0 \quad \text{对于所有 } i \in V\}.$$

练习3-13. 令 $e_k \in \mathbb{R}^n$ ($k = 0, \dots, n-1$) 是一个向量, 其中前 k 个元素为 1, 后面的 $(n-k)$ 个元素为 -1。令 $S = \{e_0, e_1, \dots, e_{n-1}, -e_0, -e_1, \dots, -e_{n-1}\}$, 即 S 包含所有由 +1 后跟 -1 或反之的向量。在这个问题集中, 你将研究 $\text{conv}(S)$ 。

1. 考虑任意向量 $a \in \{-1, 0, 1\}^n$, 满足条件 (i) $\sum_{i=1}^n a_i = 1$ 和 (ii) 对于所有 $j = 1, \dots, n-1$, 我们有 $0 \leq \sum_{i=1}^j a_i \leq 1$. (例如, 对于 $n=5$, 向量 $(1, 0, -1, 1, 0)$ 满足这些条件。) 证明 $\sum_{i=1}^n a_i x_i \leq 1$ 和 $\sum_{i=1}^n a_i x_i \geq -1$ 是凸包 $\text{conv}(S)$ 的有效不等式。
2. 有多少这样的不等式?
3. 证明任何这样的不等式都定义了 $\text{conv}(S)$ 的一个面。
(这可以用几种方法来完成。这里是一种方法, 但你也可以使用其他方法。首先证明对于任何 k , 要么 e_k 要么 $-e_k$ 满足这个不等式。然后证明在超平面上得到的向量集合是仿射独立的 (或唯一地标识它)。
4. 证明上述不等式定义了 S 的整个凸包。
(同样, 这可以用几种方法来完成。一种可能性是考虑上述第三种技术。)

3.5 全单调性

定义 3.12 矩阵 A 是完全单调的 (TU), 如果 A 的每个方阵子矩阵的行列式为 $-1, 0$ 或 $+1$ 。

全单调性的重要性源于以下定理。该定理给出了一类易于求解的整数规划问题。多面体 P 被称为整数的，如果它的所有顶点或极点都是整数（属于 \mathbb{Z}^n ）。

定理 3.12 设 A 是一个完全单调的矩阵。那么，对于任何整数右手边 b ，多面体

$$P = \{x : Ax \leq b, x \geq 0\}$$

是整数。

在我们证明这个结果之前，可以提出两点备注。首先，下面的证明实际上还可以证明对于多面体 $\{x : Ax \geq b, x \geq 0\}$ or $\{x : Ax = b, x \geq 0\}$ 也成立。然而，在后一种情况下，比完全单调性稍弱的条件就足以证明这个结果。其次，在上述定理中，我们还可以证明反过来的命题：如果对于所有整数 b ，多面体 $P = \{x : Ax \leq b, x \geq 0\}$ 都是整数的，那么 A 必须是完全单调的（然而，如果我们考虑 $\{x : Ax = b, x \geq 0\}$ 的话，这个结论就不成立了）。

证明：添加松弛变量后，我们得到多面体 $Q = \{(x, s) : Ax + Is = b, x \geq 0, s \geq 0\}$ 。可以很容易地证明（见下面的练习）如果 P 是整数，则 Q 也是整数。现在考虑 Q 的

任意 bfs。基 B 由 A 的某些列和单位矩阵 I 的某些列组成。由于 I 的列每列只有一个非零元素，即一个一，我们可以沿着这些元素展开 A_B 的行列式，并推导出，绝对值上， A_B 的行列式等于 A 的某个方阵的行列式。根据完全单调性的定义，这意味着 A_B 的行列式必须属于 $\{-1, 0, 1\}$ 。根据基的定义，它不能等于 0。因此，它必须等于 ± 1 。

我们现在证明 bfs 必须是整数。非基变量根据定义必须为零。另一方面，基变量的向量等于 $A_B^{-1}b$ 。根据线性代数， A_B^{-1} 可以表示为

$$\frac{1}{\det A_B} A_B^{adj}$$

其中 A_B^{adj} 是 A_B 的伴随矩阵，由 A_B 的子行列式组成。

因此， b 和 A_B^{adj} 都是整数，这意味着 $A_B^{-1}b$ 也是整数，因为 $|\det A_B| = 1$ 。这证明了 bfs 的整数性。 △

练习 3-14。证明当且仅当 $(x, b - Ax)$ 是 Q 的极点时， x 是 P 的极点。结论是，只要 A 和 b 的条目都是整数， P 是整数当且仅当 Q 是整数。

在二分图匹配问题的情况下，约束矩阵 A 具有非常特殊的结构，我们将在下面证明它是完全单调的。这与定理 3.12 一起证明了二分图匹配问题笔记中的定理 1.6。首先，让我们

重新陈述一下设置。假设我们的二分图的分法是 (U, V) (为了避免与矩阵 A 或基 B 产生混淆)。考虑

$$\begin{aligned} P &= \{x : \sum_j x_{ij} = 1 & i \in U \\ &\quad \sum_i x_{ij} = 1 & j \in V \\ &\quad x_{ij} \geq 0 & i \in U, j \in V\} \\ &= \{x : Ax=b, x \geq 0\}. \end{aligned}$$

定理3.13 矩阵 A 是完全单调的。

我们定义矩阵 A 的方式对应于一个完全二分图。如果我们考虑任何二分图，那么我们只需考虑 A 的一个子矩阵，根据定义它也是完全单调的。

证明： 考虑矩阵 A 的任意方阵子矩阵 T 。我们考虑三种情况。首先，如果 T 的某一列或某一行的所有元素都为零，则行列式为零。其次，如果 T 存在一列或一行只有一个+1，则通过沿着该+1展开行列式，我们可以考虑一个更小尺寸的矩阵 T 。最后一种情况是 T 的每列（和每行）至少有两个非零元素。鉴于 A 的特殊结构，每列实际上必须有恰好两个非零元素。通过将 U 的顶点对应的 T 的行相加，并将 V 的顶点对应的 T 的行相加，因此得到相同的向量，证明了 T 的行是线性相关的，从而其行列式为零。这证明了 A 的完全单调性。

△

我们以一个技术性的备注作为结论。在应用定理3.12之前，应该先删除矩阵 A 的一行，因为它不具有完全行秩，这一事实在定义bfs时隐含地使用了。然而，删除矩阵 A 的一行仍然保持其完全单调性。

练习3-15。 如果矩阵 A 是完全单调的，则矩阵 A 的转置 A^T 也是完全单调的。

练习3-16。 使用完全单调性证明König定理。

下面的定理给出了矩阵完全单调的必要和充分条件。

定理3.14 设 A 是一个 $m \times n$ 的矩阵，其元素属于 $\{-1, 0, 1\}$ 。那么，当且仅当对于所有的行子集 $R \subseteq \{1, 2, \dots, n\}$ ，存在一个将 R 划分为 R_1 和 R_2 的分割，使得对于所有的 $j \in \{1, 2, \dots, m\}$ 成立：

$$\sum_{i \in R_1} a_{ij} - \sum_{i \in R_2} a_{ij} \in \{0, 1, -1\}.$$

我们只证明 if 方向（但这是最重要的，因为这可以证明矩阵是完全单调的）。

证明： 假设对于每个 R ，存在所需的分区。我们需要证明 A 的任何 $k \times k$ 子矩阵的行列式在 $\{-1, 0, 1\}$ 中，并且对于任何 k 都必须成立。让我们通过对 k 进行归纳来证明它。对于 $k=1$ ，这是显然成立的。假设对于 $k-1$ 成立，我们将证明对于 k 也成立。

设 B 是 A 的一个 $k \times k$ 子矩阵，我们可以假设 B 是可逆的（否则行列式为0，没有什么可证明的）。逆矩阵 B^{-1} 可以写成 $\frac{1}{\det(B)} B^*$ ，其中 B^* 的所有元素对应于 A 的 $(k-1) \times (k-1)$ 子矩阵。根据我们的归纳假设， B^* 的所有元素都在 $\{-1, 0, 1\}$ 中。设 b_1^* 为 B^* 的第一行， e_1 为 k 维行向量 $[1 \ 0 \ 0 \ \cdots \ 0]$ ，则 $b_1^* = e_1 B^*$ 。根据 B 和 B^* 之间的关系，我们有

$$b_1^* B = e_1 B^* B = \det(B) e_1 B^{-1} B = \det(B) e_1. \quad \text{根}$$

据假设，我们知道存在一个将 R 分成 R_1 和 R_2 的划分，对于所有的 j ：

$$\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} \in \{-1, 0, 1\}. \quad (6)$$

从 (5) 中，我们得到

$$\sum_{i \in R} b_{1i}^* b_{ij} = \begin{cases} \det(B) & j = 1 \\ 0 & j = 1 \end{cases} \quad (7)$$

由于 $\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij}$ 和 $\sum_{i \in R} b_{1i}^* b_{ij}$ 对于每个 j （因为 $b_{1i}^* \in \{-1, 1\}$ ），这意味着

$$\sum_{i \in R_1} b_{ij} - \sum_{i \in R_2} b_{ij} = 0 \quad j = 1. \quad (8)$$

对于 $j=1$ ，我们不能得到0，否则 B 将是奇异的（通过添加和减去 B 的行，我们将得到全零向量）。因此，

$$\sum_{i \in R_1} b_{i1} - \sum_{i \in R_2} b_{i1} \in \{-1, 1\}.$$

如果我们定义 $y \in \mathbb{R}^k$ 为

$$y_i = \begin{cases} 1 & i \in R_1 \\ -1 & i \in R_2 \\ 0 & \text{否则，我} \end{cases}$$

们得到 $yB = \pm e_1$ 。因此，

$$y = \pm e_1 B^{-1} = \pm \frac{1}{\det B} e_1 B^* = \pm \frac{1}{\det B} b_1^*,$$

这意味着 $\det B$ 必须是1或-1。

△

练习3-17. 假设我们有 n 个活动可供选择。活动 i 从时间 t_i 开始，并在时间 u_i 结束（更准确地说是在 u_i 之前）；如果选择该活动，我们将获得 p_i 个单位的利润。我们的目标是选择一组不重叠的活动（尽管我们可以选择一个在相同时间 t 结束的活动和一个在相同时间 t 开始的活动），并且所选活动的总利润（即利润之和）最大。

1. 将变量 x_i 定义为表示活动 i 是否被选择（ $x_i = 1$ ）或未选择（ $x_i = 0$ ）的变量，编写形式为 $\max\{p^T x : Ax \leq b, x \in \{0, 1\}^n\}$ 的整数规划问题，用于解决此问题。
2. 证明矩阵 A 是完全单调的，这意味着可以通过解线性规划 $\max\{p^T x : Ax \leq b, 0 \leq x_i \leq 1 \text{ (对于每个 } i)\}$ 来解决此问题。

练习3-18. 给定一个二分图 G 和一个整数 k ，令 S_k 为至多有 k 条边的匹配的所有关联向量的集合。我们对于找到 $P_k = \text{conv}(S_k)$ 的线性不等式系统的描述感兴趣。更准确地说，你将展示 $\text{conv}(S_k)$ 由以下形式给出：

$$P_k = \{x : \begin{array}{ll} \sum_j x_{ij} \leq 1 & \text{对于每个 } i \in A \\ \sum_i x_{ij} \leq 1 & \text{对于每个 } j \in B \\ \sum_i \sum_j x_{ij} \leq k & \\ x_{ij} \geq 0 & \text{其中 } i \in A, j \in B \end{array}\}$$

如果没有最后一个约束条件，我们已经证明了结果矩阵是完全单调的。

1. 如果加上额外的约束条件，结果矩阵是否仍然是完全单调的？请证明或者反驳这个结论。
2. 证明 P_k 确实等于 $\text{conv}(S_k)$ 。
3. 现在假设我们的边不再有基数约束，而是被分成 E_1 和 E_2 ，我们只要求匹配中最多有 k 条边来自 E_1 （而可以来自 E_2 的边可以任意多）。在这种情况下，所有这样匹配的凸包是否仍然可以通过简单地替换 $\sum_i \sum_j x_{ij} \leq k$ by

$$\sum_i \sum_{j: (i,j) \in E_1} x_{ij} \leq k?$$

4. 流量和割集讲义

4.1 最大流

网络流处理的是在网络中物品（水、电、数据包、气体、汽车、火车、货币或任何抽象物体）的流动模型。网络中的链接具有容量限制，物品在网络中除了在指定位置注入或提取一定数量的物品外不会消失。主要问题是在这个网络中可以发送多少物品。

下面是最大流问题的更正式定义。我们有一个有向图 $G = (V, E)$ ，还有两个特殊顶点 S 和 T ； S 被称为源点， T 被称为汇点。我们有一个上界容量函数 $u : E \rightarrow \mathbb{R}$ ，还有一个下界容量函数 $l : E \rightarrow \mathbb{R}$ （有时选择在所有位置都为0）。一个流 x 将是对弧（有向边）的值的分配，使得：

1. 对于每个 $e \in E$: $l(e) \leq x_e \leq u(e)$,

2. 对于每个 $v \in V \setminus \{s, t\}$:

$$\sum_{e \in \delta^+(u)} x_e - \sum_{e \in \delta^-(u)} x_e = 0. \quad (1)$$

符号 $\delta^+(u)$ 表示离开节点 u 的弧集合，而符号 $\delta^-(u)$ 表示进入节点 u 的弧集合。

方程式 (1) 被称为流守恒约束。给定一个流 x ，它的流值 $|x|$ 是从节点 s 流出的净流量： $|x| :=$

$$\sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e. \quad (2)$$

一个重要的观察是， $|x|$ 也等于进入 t 的净流量，或者减去离开 t 的净流量。事实上，将 (1) 在 $u \in V \setminus \{s, t\}$ 上求和，再加上 (2)，我们得到：

$$\begin{aligned} |x| &= \sum_{v \in V \setminus \{t\}} \left(\sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e \right) \\ &= \sum_{e \in \delta^-(t)} x_e - \sum_{e \in \delta^+(t)} x_e \end{aligned}$$

通过观察第一个求和中每条弧的贡献来得到。

最大流问题是找到最大值为 $|x|$ 的流 x 的问题。
这是一个线性规划问题：

$$\begin{aligned} \text{Max} \quad & \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e \\ \text{满足条件:} \quad & \sum_{e \in \delta^+(u)} x_e - \sum_{e \in \delta^-(u)} x_e = 0 \quad u \in V \setminus \{s, t\} \\ & l(e) \leq x_e \leq u(e) \quad e \in E. \end{aligned}$$

因此，我们可以使用线性规划算法来找到最大流和对偶性来推导最优性属性，但我们将展示更多的组合优化算法可以被开发出来，而对偶性则转化为关于割的性质。

以矩阵形式，线性规划可以写成：

$$\max\{c^T x : \begin{array}{ll} Nx &= 0, \\ Ix &\leq u, \\ -Ix &\leq -l \end{array}\}$$

其中 N 是（顶点-弧关联¹）矩阵，行索引为 $u \in V \setminus \{s, t\}$ ，列索引为弧 $e = (i, j) \in E$ ； N_{ue} 的条目为：

$$N_{ue} = \begin{cases} 1 & u = i \\ -1 & u = j \\ 0 & u \notin \{i, j\}. \end{cases}$$

线性规划的约束条件为： $Ax \leq b$ 其中

$$A = \begin{pmatrix} N & & \\ - & - & - \\ I & & \\ - & - & - \\ -I & & \end{pmatrix},$$

其中一些约束是等式，一些是不等式。

引理 4.1 A 是完全单调的。

证明： 我们可以使用多面体章节的定理 3.14，但直接证明它很容易。考虑 A 的任意方阵子矩阵，我们希望计算其行列式的符号。如果存在一行只有一个 +1 或一个 -1（特别是来自单位子矩阵 I 或 $-I$ 的行），我们可以展开行列式并

¹更准确地说，我们不考虑顶点 s 和 t 的部分

计算 A 的一个较小子矩阵的行列式（符号不变）。重复此过程，我们现在有一个 N 的方阵子矩阵。如果存在一列只有一个 $+1$ 或一个 -1 ，那么我们可以沿着该列展开行列式并得到一个较小的子矩阵。因此，我们要么得到一个空子矩阵，此时原矩阵的行列式为 $+1$ 或 -1 ，要么得到一个 N 的方阵子矩阵，其中每列恰好有一个 $+1$ 和一个 -1 。这个子矩阵的行是线性相关的，因为它们的和为零向量。因此，行列式为 0 。这证明了完全单调性。

△

作为一个推论，这意味着如果右侧（即上下容量）是整数值，则总是存在一个仅取整数值的最大流。

推论4.2 如果 $l: E \rightarrow \mathbb{Z}$ 和 $u: E \rightarrow \mathbb{Z}$ ，那么存在一个最大流 x ，使得对于所有的 $e \in E$ ，都有 $x_e \in \mathbb{Z}$ 。

4.1.1 特殊情况

弧不相交路径。如果对于所有的 $e \in E$ ，都有 $l(e) = 0$ ，并且对于所有的 $e \in E$ ，都有 $u(e) = 1$ ，则任何整数流 x 只能取值为 $\{0, 1\}$ 。我们声称对于一个整数流 x ，存在 $|x|$ 个弧不相交（即没有共同弧）的从 s 到 t 的路径。实际上，可以通过流分解来获得这样的路径。只要 $|x| > 0$ ，就从 s 中取出一条弧，使得 $x_e = 1$ 。

现在沿着这条弧线走，每当我们到达一个顶点 $u = t$ ，根据流量守恒定律，我们知道存在一条离开 u 的弧线，我们还没有遍历过（即使我们再次到达 s ）。当我们到达 t 时，这个过程停止，因此我们确定了一条从 s 到 t 的路径。去除这条路径后，我们得到一个新的流量 x' （事实上，顶点 $= s, t$ 的流量守恒得到保持），其中 $|x'| = |x| - 1$ 。重复这个过程，我们得到 $|x|$ 条从 s 到 t 的路径，并且根据构造，它们是弧线不相交的。我们得到的路径可能不是简单路径；然而，可以通过删除行走中重复出现的顶点之间的部分使它们变为简单路径。总结一下，如果对于所有的 $e \in E$ ， $l(e) = 0$ ，并且对于所有的 $e \in E$ ， $u(e) = 1$ ，那么从流量为 k 的最大流中，我们可以提取出 k 条弧线不相交的（简单）路径，从 s 到 t 。反过来，如果有 k 条弧线不相交的从 s 到 t 的路径，很容易构造出一个流量为 k 的流。这意味着从 s 到 t 的最大流值表示了 s 和 t 之间弧线不相交路径的最大数量。

二分图匹配。可以将二分图中的最大匹配问题转化为最大流问题。考虑一个二分图 $G = (V, E)$ ，其中顶点集 $V = A \cup B$ 。现在考虑一个有向图 D ，其顶点集为 $V \cup \{s, t\}$ 。在 D 中，从 s 到 A 中的每个顶点都有一条弧，其容量满足 $l(e) = 0$ 和 $u(e) = 1$ 。同时，从 B 中的每个顶点到 t 都有一条弧，其容量满足 $l(e) = 0$ 和 $u(e) = 1$ 。每条边 $(a, b) \in E$ 都是从 A 到 B 的有向边，其容量下界为 0 ，上界为 $+\infty$ （或者只是 1 ）。很容易看出，从任意大小为 k 的匹配可以构造出一个流量为 k 的流；类似地，对于任意流量为 k 的整数值流，都可以对应一个大小为 k 的匹配。由于容量是整数，根据推论4.2，这意味着在二分图中的最大流问题中，最大流量是整数。

²一个简单路径是指没有重复顶点的路径。

D 与 G 中任何匹配的最大大小相同。观察到 A 和 B 之间的弧的上限容量并不重要, 只要它们大于等于1。

方向。考虑将无向图 $G=(V, E)$ 的边定向的问题, 使得结果有向图中任何顶点 v 的入度最多为 $k(v)$ 。这可以被表述为一个最大流问题, 其中我们有 (i) 对于 G 的每个顶点一个顶点, (ii) 对于 G 的每条边一个顶点, 以及 (iii) 额外的两个顶点 S 和 T 。详细内容留作练习。

练习4-1. 假设给定一个 $m \times n$ 矩阵 $A \in \mathbb{R}^{m \times n}$, 其行和为 $R_1, \dots, R_m \in \mathbb{Z}$, 列和为 $C_1, \dots, C_n \in \mathbb{Z}$ 。一些条目可能不是整数, 但行和和列和是整数。证明存在一个满足以下条件的四舍五入矩阵 A' :

- 行和列的和 A 和 A' 是相同的,
- $a'_{ij} = \lceil a_{ij} \rceil$ 或者 $a'_{ij} = \lfloor a_{ij} \rfloor$ (即 a'_{ij} 要么向上取整, 要么向下取整)。

顺便说一下, 这种取整对人口普查局很有用, 因为他们不想发布会给出关于具体个人太多信息的统计数据。他们希望能够修改条目而不修改行和列的和。

4.2 割

在本节中, 我们为最大流问题推导出一个重要的对偶性结果, 通常情况下, 这将采取极小极大关系的形式。

在有向图 $G=(V, A)$ 中, 我们将 *cutset* 或者更简单地说 *cut* 定义为弧的集合 $\delta^+(S) = \{(u, v) \in A : u \in S, v \in V \setminus S\}$ 。观察到我们之前的符号 $\delta^+(v)$ 对于 $v \in V$ 而不是 $\delta^+(\{v\})$ 是一种轻微的滥用符号的情况。同样地, 我们将 $\delta^-(S)$ 定义为 $\delta^+(V \setminus S)$, 即进入顶点集 S 的弧。我们通常将割集 $\delta^+(S)$ 与相应的顶点集 S 等同起来。我们说割 $\delta^+(S)$ 是一个 $s-t$ 割 (其中 s 和 t 是顶点), 如果 $s \in S$ 且 $t \notin S$ 。

对于一个无向图 $G=(V, E)$, $\delta^+(S)$ 和 $\delta^-(S)$ 是相同的, 并且将被表示为 $\delta(S) = \{(u, v) \in E : |\{u, v\} \cap S| = 1\}$ 。观察到现在 $\delta(S) = \delta(V \setminus S)$ 。

对于一个有向图 $G=(V, E)$ 上的最大流问题和上界和下界容量函数 u 和 l , 我们定义由 S 引起的割的容量 $C(S)$ 为

$$C(S) = \sum_{e \in \delta^+(S)} u(e) - \sum_{e \in \delta^-(S)} l(e) = u(\delta^+(S)) - l(\delta^-(S)).$$

根据流 x 的定义, 我们有

$$C(S) \geq \sum_{e \in \delta^+(S)} x_e - \sum_{e \in \delta^-(S)} x_e.$$

我们之前已经证明了从 s 流出的净流量等于流入 t 的净流量。同样地，我们可以证明对于任何 $s \in S$ 和 $t \notin S$ （即诱导出的割是一个 $s-t$ 割），流值 $|x|$ 等于：

$$|x| = \sum_{e \in \delta^+(S)} x_e - \sum_{e \in \delta^-(S)} x_e.$$

这是通过对 $u \in S \setminus \{s\}$ (1) 和 (2) 求和得到的。因此，对于任何 $s \in S$ 和 $t \notin S$ 以及任何 $s-t$ 流 x ，我们有：

$$|x| \leq C(S).$$

因此，通过最大化 $s-t$ 流和最小化 $s-t$ 割，我们得到

$$\max |x| \leq \min_{S: s \in S, t \notin S} C(S).$$

这是弱对偶性，但实际上，根据以下定理，我们总是有相等性。

当然，我们需要假设最大流问题是可行的。例如，如果存在一条边 $l(e) > u(e)$ ，则不存在流量（我们将在后面证明流存在的必要和充分条件是 (i) 对于每个 E 中的 $l(e) \leq u(e)$ ，以及 (ii) 对于任意 $S \subset V$ 且 $|S \cap \{s, t\}| \neq 1$ ，我们有 $u(\delta^+(S)) \geq l(\delta^-(S))$ ）。

定理4.3（最大流-最小割定理） 对于任何存在可行流的最大流问题，我们有最大流值等于任何割的最小容量：

$$\max_{\text{流量 } x} |x| = \min_{S: s \in S, t \notin S} C(S).$$

证明这个定理的一种方法是利用线性规划的强对偶性，并且证明从任何最优对偶解可以导出一个具有相同容量的割。另一种方法，也是我们追求的方法，是开发一个算法来找到最大流，并且证明当算法终止时，我们也有一个容量等于我们构造的流的割，从而证明流的最优性和最小最大关系的相等性。

这里有一个寻找最大流的算法。让我们假设我们已经有一个可行流 x （如果对于所有的边 e ，有 $u(e) \geq 0$ 和 $l(e) \leq 0$ ，我们可以从 $x=0$ 开始）。给定一个流 x ，我们在相同的顶点集 V 上定义一个剩余图 G_x 。在 G_x 中，如果 (i) $(i, j) \in E$ 且 $x_{ij} < u(i, j)$ 或者 (ii) $(j, i) \in E$ 且 $x_{ji} > l(j, i)$ ，则有一条弧 (i, j) 。在情况 (i) 下，我们称 (i, j) 为正向弧，在情况 (ii) 下，它是反向弧。如果同时发生 (i) 和 (ii)，我们引入两条弧 (i, j) ，一条正向弧和一条反向弧；准确地说， G_x 因此是一个多重图。现在考虑剩余图中从 s 到 t 的任意有向路径 P ；这样的路径被称为增广路径。记 P^+ 为 P 中的正向弧， P^- 为反向弧。我们可以按以下方式修改流 x ：

$$x'_e = \begin{cases} x_e + \epsilon & e \in P^+ \\ x_e - \epsilon & e \in P^- \\ x_e & e \notin P \end{cases}$$

这被称为沿着 P 推动 ϵ 个流量单位，或者简单地沿着 P 增广。观察到在沿着路径推动流量时，任何顶点 u 上的流量守恒仍然成立。如果 u 不在路径上，这是显然的；如果 u 在路径上，与 u 相邻的两条弧在 P 上的贡献互相抵消。为了确保得到的结果 x' 是可行的（满足容量约束），我们选择

$$\epsilon = \min \left(\min_{e \in P^+} (u(e) - x_e), \min_{e \in P^-} x_e - l(e) \right).$$

通过构建剩余图，我们有 $\epsilon > 0$ 。因此，沿着增广路径推动 ϵ 单位的流量提供了一个新的流量 x' ，其值 $|x'|$ 满足 $|x'| = |x| + \epsilon$ 。因此，流量 x 不是最大的。

反之，假设剩余图 G_x 不包含从 s 到 t 的任何有向路径。令 $S = \{u \in V: \text{在 } G_x \text{ 中存在一条从 } s \text{ 到 } u \text{ 的有向路径}\}$ 。根据定义， $s \in S$ 且 $t \notin S$ （否则会存在增广路径）。另外，根据定义， G_x 中不存在从 S 到 $V \setminus S$ 的弧。这意味着，对于 $e \in E$ ，如果 $e \in \delta^+(S)$ ，则 $x_e = u(e)$ ，如果 $e \in \delta^-(S)$ ，则 $x_e = l(e)$ 。这意味着

$$C(S) = \sum_{e \in \delta^+(S)} u(e) - \sum_{e \in \delta^-(S)} l(e) = u(\delta^+(S)) - l(\delta^-(S)) = \sum_{e \in \delta^+(S)} x_e - \sum_{e \in \delta^-(S)} x_e = |x|.$$

这表明流量 x 是最大的，并且存在一个相同容量的 $s - t$ 割，就像 $|x|$ 一样。

这几乎证明了定理4.3。实际上，只要存在增广路径，我们就可以沿着它推动流量，更新剩余图并继续。无论何时算法停止，如果它停止，我们都有一个最大流和一个相应的最小割。但是，也许这个算法永远不会停止；这实际上可能发生，如果容量可能是无理数，并且在每次迭代中选择了“错误”的增广路径。为了完成最大流最小割定理的证明，我们可以简单地使用线性规划的形式化来解决最大流问题，这表明最大流存在（在线性规划中，最大值是一个真正的最大值（因为它是由一个顶点实现的），而不仅仅是一个可能无法达到的上确界）。从该流 x 开始并构造其剩余图 G_x ，我们得到一个相应的相同值的最小 $s - t$ 割。

4.2.1 最大流最小割的解释

最大流最小割定理与最大流的整数性质结合起来，可以推导出几个组合最小-最大关系。

二分图匹配。例如，考虑最大二分图匹配问题及其在第4.1.1节中给出的最大流问题的表述。我们说对于 A 和 B 之间的弧，我们在选择 $u(e)$ 时有灵活性；在这里，让我们假设我们将它们设置为等于 $+\infty$ （或任何足够大的整数）。考虑任意集合

对于 $C(S)$ 是有限的, 不能存在任何边 $(i, j) \in E$, 其中 $i \in A \cap S$ 且 $j \in B \setminus S$ 。换句话说, $N(A \cap S) \subseteq B \cap S$, 即如果我们设置 $C = (A \setminus S) \cup (B \cap S)$, 我们有 C 是一个顶点覆盖。对应割的容量 $C(S)$ 是多少? 它恰好是 $C(S) = |A \setminus S| + |B \cap S|$, 第一项对应于从 s 到 $A \setminus S$ 的弧, 第二项对应于 $B \cap S$ 和 t 之间的弧。因此, 最大流最小割定理暗示着存在一个顶点覆盖 C , 其基数等于最大匹配的大小。因此, 我们重新推导了 König 定理。我们还可以推导出 Hall 定理, 关于完美匹配的存在性。

弧不相交路径。对于最大弧不相交路径的问题, 可以将最大流最小割定理解释为 Menger 定理:

定理4.4 在有向图 $G = (V, A)$ 中, 如果在 s 和 t 之间存在 k 个弧不相交路径, 那么对于所有满足 $s \in S$ 的 $S \subseteq V \setminus \{t\}$, 我们有 $|\delta^+(S)| \geq k$ 。

练习4-2. 在棒球赛季的某个时刻, 美国联盟的每个 N 支球队都已经进行了几场比赛。假设第 i 支球队到目前为止赢得了 w_i 场比赛, 而 $g_{ij} = g_{ji}$ 表示球队 i 和 j 之间尚未进行的比赛次数。没有比赛以平局结束, 所以每场比赛给一支球队一分, 另一支球队得零分。你想要确定你最喜欢的球队, 比如第 N 支球队, 是否仍然有机会获胜。换句话说, 你想要确定是否存在一种比赛结果 (记住, 没有平局), 使得球队 N 至少获得其他所有球队的胜利次数 (我们允许球队 N 与其他球队并列第一名)。

证明这个问题可以被解决为一个最大流问题。给出关于 g_{ij} 的必要和充分条件, 使得团队 n 仍然能够获胜。

练习4-3. 考虑以下定向问题。我们给定一个无向图 $G = (V, E)$, 以及每个顶点 $v \in V$ 的整数值 $p(v)$ 。我们想知道是否可以将 G 的边定向, 使得得到的有向图中至多有 $p(v)$ 个弧指向 v (“入度要求”)。换句话说, 对于每条边 $\{u, v\}$, 我们需要决定是将其定向为 (u, v) 还是 (v, u) , 并且我们希望至多有 $p(v)$ 个弧指向 v 。

1. 证明这个问题可以被表述为一个最大流问题。也就是说, 展示如何创建一个最大流问题, 通过其解决方案, 你可以决定图是否可以被定向, 如果可以, 还可以给出定向。
2. 考虑图无法定向且满足入度要求的情况。
从最大流最小割定理证明必然存在一个集合 $S \subseteq V$ 使得 $|E(S)| > \sum_{v \in S} p(v)$, 其中通常 $E(S)$ 表示两个端点都在 S 内的边的集合。

4.3 最大流算法的效率

最大流最小割定理的证明为寻找最大流提供了一个简单的增广路径算法。从任意可行流开始，在剩余图中沿着增广路径推送流，直到不存在这样的增广路径为止。现在我们要解决的主要问题是：在终止之前，这个算法需要多少次迭代。

正如之前提到的，如果容量是无理数，这个算法可能永远不会终止。对于整数容量的情况，如果我们从一个整数流开始，很容易看出我们始终保持一个整数流，并且我们总是推动整数数量的流量。因此，迭代次数受两个流值之间的最大差异的限制，最多为 \sum 这是有限的，但不是多项式时间复杂度（它仅对容量 u 和 l 的对数依赖）。

最短增广路径变体。Edmonds和Karp提出了一种增广路径算法的变体，它保证在多项式次数的迭代中终止，仅取决于 $n = |V|$ 和 $m = |E|$ 。对容量没有任何假设，并且该算法对于无理数容量也是正确且终止的。

Edmonds和Karp的想法是始终在剩余图中找到一条最短的增广路径，即弧的数量最少。给定一个流 x ，考虑剩余图 G_x 。对于任意顶点 v ，让 $d(v)$ 表示从 s 到 v 在 G_x 中的距离（弧的数量）。最短增广路径算法是选择一条路径 $v_0 - v_1 - \dots - v_k$ 在剩余图中，其中 $v_0 = s$ ， $v_k = t$ ，且 $d(v_i) = i$ 。

算法的分析如下。设 P 是从 s 到 t 在 G_x 中的最短增广路径， x' 是在沿着 P 尽可能推送流后得到的结果流。设 d' 是对应于 $G_{x'}$ 的距离标签。观察到只有沿着 P 的反向弧 (i, j) （满足 $d(i) = d(j) + 1$ ）可能会在 $G_{x'}$ 中引入。因此，在增广后，我们有对于每条弧 $(i, j) \in E_{x'}$ ， $d(j) - d(i) \leq 1$ 。将这些不等式沿着从 s 到 $j \in V$ 的任意路径 P' 在 $G_{x'}$ 上求和，我们得到对于任意 $j \in V$ ， $d(j) \leq d'(j)$ 。特别地，我们有 $d(t) \leq d'(t)$ 。由于距离标签永远不会大于 $n-1$ ，我们得到到 t 的距离只能最多增加 $n-1$ 次。但是 $d'(t)$ 也可能等于 $d(t)$ 。在这种情况下， P 的一条饱和弧意味着在 $G_{x'}$ 中具有 $d(j) = d(i) + 1$ 的弧 (i, j) 比 G_x 中少一个。因此，在最多 m 次迭代后， t 的距离标签必须严格增加。总结起来，这意味着增广的次数最多为 $m(n-1)$ 。在剩余图中构建和找到增广路径所需的时间最多为 $O(m)$ 时间。这意味着最短增广路径算法的总运行时间最多为 $O(m^2n)$ 。这可以进一步改进，但这不是本讲义的重点。

4.4 最小割

从现在开始，我们假设只有上限容量 u 而没有下限容量 l ($l(e) = 0$ ，对于所有的 e)。到目前为止，我们解决的最小 $s - t$ 割问题对应于

如下:

$$\min_{S: s \in S, t \notin S} u(\delta^+(S)).$$

如果我们的图 $G = (V, E)$ 是无向的, 并且我们想要找到最小的 $s - t$ 割, 即

$$\min_{S: s \in S, t \notin S} u(\delta(S)),$$

我们可以简单地将每条边 e 替换为两个相反方向但容量相同的弧, 并将问题转化为在有向图中找到最小的 $s - t$ 割. 正如我们刚刚展示的, 这可以通过最大流计算来完成.

现在, 考虑在图中找到全局最小割的问题. 让我们首先考虑有向情况. 找到全局最小割 (或者只是最小割) 意味着找到 S 使得最小化:

$$\min_{S: \emptyset \neq S \neq V} u(\delta^+(S)).$$

这个问题可以通过以下方式简化为 $2(n-1)$ 个最大流计算 (其中 $n = |V|$)。首先, 我们可以任意选择一个顶点 $s \in V$, 并且 s 要么在 S 中, 要么在 $V \setminus S$ 中。因此, 对于任意的 $t \in V \setminus \{s\}$, 我们解决两个最大流问题, 一个给出最小的 $s - t$ 割, 另一个给出最小的 $t - s$ 割。取所有这些割中的最小值, 我们得到有向图中的全局最小割。

在无向图中找到最小割问题时, 我们甚至不需要为每个 $t \in V \setminus \{s\}$ 解决两个最大流问题, 只需要解决其中一个就足够了。因此, 在无向图中的全局最小割问题可以通过计算 $n-1$ 个最大流问题来解决。目前最快的最大流算法所需的时间略多于 $O(mn)$ (例如, Goldberg 和 Tarjan 的算法 [1] 所需的时间为 $O(mn \log(n^2/m))$)。由于我们需要使用它 $n-1$ 次, 我们可以在 $O(mn^2 \log(n^2/m))$ 的时间内找到一个最小割。然而, 这些 $n-1$ 个最大流问题是相关的, Hao 和 Orlin [2] 已经证明可以通过修改 Goldberg 和 Tarjan 的算法在 $O(mn \log(n^2/m))$ 的时间内解决所有这些问题。因此, 最小割问题可以在这个时间限制内解决。

我们现在将推导出一个不基于网络流的最小割问题的算法, 其运行时间略优于 Hao 和 Orlin 的算法。该算法由 Stoer 和 Wagner [6] 提出, 是 Nagamochi 和 Ibaraki [5] 早期结果的简化。我们还应指出, 由于 Karger 和 Stein [4] 提出的随机算法的运行时间为 $O(n^2 \log^3 n)$, 而 Karger [3] 提出的后续算法的运行时间为 $O(m \log^3 n)$ 。

我们首先需要定义。对于任意两个集合 $A, B \subseteq V$ 的顶点,

$$u(A : B) := \sum_{i \in A, j \in B} \text{定义 } u((i, j)).$$

算法的描述如下。简单来说, 该算法从任意顶点开始, 通过始终选择与前面顶点的总成本最大的顶点来构建顶点的排序; 这被称为最大邻接顺序。考虑到这个最大邻接顺序中最后一个顶点所引起的割,

以及通过将最后两个顶点缩小后应用该过程到所得图形所获得的割。（如果存在从一个顶点 v 到这最后两个顶点的边，则用容量等于这两条边容量之和的一条边替换这两条边。）断言是，在考虑的割中，最好的割是整体的最小割。下面给出正式的描述。

最小割(G)

- 令 v_1 为 G 的任意顶点。 $n = |V(G)|$
- $S = \{v_1\}$
- 对于 $i = 2$ 到 n
 - . 令 v_i 为 V 中的顶点 $\setminus S$ 中的顶点
 - .. 使得 $c(S : \{v\})$ 最大化（对于所有 $v \in V \setminus S$ ）
 - . $S := S \cup \{v_i\}$
- . 结束循环
- . if $n = 2$ then 返回割集 $\delta(\{v_n\})$
- . 否则
 - . 令 G' 为通过收缩 v_{n-1} 得到的图 G
 - v_n
 - . 令 C 为 $\text{MINCUT}(G')$ 返回的割集
 - . 在 C 和 $\delta(\{v_n\})$ 中返回较小的割集（按照代价计算）
- . 结束条件

分析基于以下关键性断言。

断言 4.5 $\{v_n\}$ （或 $\{v_1, v_2, \dots, v_{n-1}\}$ ）在 G 中诱导出一个最小 (v_{n-1}, v_n) -割集。（注意我们事先不知道 v_{n-1} 和 v_n 的值。）

从这里，算法的正确性很容易得出。事实上，最小割要么是一个 (v_{n-1}, v_n) -割，要么不是。如果是的话，我们就很好，多亏了上面的断言。如果不是的话，我们可以假设通过对顶点集合的大小进行归纳，它将被调用 $\text{MINCUT}(G')$ 返回。

证明： 让 $v_1, v_2, \dots, v_i, \dots, v_j, \dots, v_{n-1}, v_n$ 是算法选择的顶点序列，我们用 A_i 表示序列 v_1, v_2, \dots, v_{i-1} 。我们对分割感兴趣，分割将 v_{n-1} 和 v_n 分开。让 C 是任意集合，使得 $v_{n-1} \in C$ 且 $v_n \in C$ 。然后我们想要证明由 C 引起的分割满足

$$u(\delta(C)) \geq u(\delta(A_n)).$$

如果 v_i 或 v_{i-1} 属于 C 但不属于 C ，则将顶点 v_i 定义为相对于 C 的关键点。我们声称，如果 v_i 是关键点，则

$$u(A_i : \{v_i\}) \leq u(C_i : A_i \cup \{v_i\} \setminus C_i)$$

其中 $C_i = (A_i \cup \{v_i\}) \cap C$ 。

注意，这意味着 $u(\delta(C)) \geq u(\delta(A_n))$ ，因为 v_n 是关键点。现在让我们通过对关键点序列进行归纳来证明这个命题。

设 v_i 是第一个关键点。那么

$$u(A_i : \{v_i\}) = u(C_i : A_i \cup \{v_i\} \setminus C_i)$$

因此归纳的基础是真实的。

对于归纳步骤，假设对于关键顶点 v_i 的断言是真实的，并且让 v_j 是（在 v_i 之后的）下一个关键顶点。然后

$$\begin{aligned} u(A_j : \{v_j\}) &= u(A_i : \{v_j\}) + u(A_j \setminus A_i : \{v_j\}) \\ &\leq u(A_i : \{v_i\}) + u(A_j \setminus A_i : \{v_j\}) \\ &\leq u(C_i : A_i \cup \{v_i\} \setminus C_i) + u(A_j \setminus A_i : \{v_j\}) \\ &\leq u(C_j : A_j \cup \{v_j\} \setminus C_j), \end{aligned}$$

第一个不等式是根据 v_i 的定义得到的，第二个不等式是根据归纳假设得到的，最后一个不等式是根据 v_j 是下一个关键顶点得到的。证明的结论是观察到 A_n 诱导出割 $\{v_1, v_2, \dots, v_{n-1}\} : \{v_n\}$ 。△

运行时间取决于具体的实现方式。使用斐波那契堆，我们可以在 $O(m + n \log n)$ 的时间内实现每次迭代，从而得到总运行时间为 $O(mn + n^2 \log n)$ 。

练习4-4。假设 G 是一个无向图，其中每个顶点的度至少为 k 。证明存在两个顶点 u 和 v 之间至少有 k 条边不相交的路径。

4.5 最小 T 奇割问题

给定一个图 $G = (V, E)$ ，其中边的容量非负，由 u 给出，并给定一个偶数集合 T of vertices，最小 T 奇割问题是要找到最小化的 S ：

$$\min_{S \subset V : |S \cap T| \text{ 是奇数}} u(\delta(S)).$$

我们称 S 是 T 奇数的，如果 $|S \cap T|$ 是奇数。观察到如果 S 是 T 奇数的，那么 $V \setminus S$ 也是 T 奇数的，反之亦然。

我们为这个问题提供了一个多项式时间算法。我们不会介绍最高效的算法，而是介绍其中最简单的一个。让 $ALG(G, T)$ 表示这个算法。算法 $ALG(G, T)$ 的第一步是找到至少一边有 T 上的一个顶点的最小割：

$$\min_{S \subset V : \emptyset \neq S \cap T = T} u(\delta(S)).$$

这可以通过进行 $|T| - 1$ 最小割计算来完成，通过固定一个顶点 s 在 T 中，然后尝试所有顶点 $t \in T \setminus \{s\}$ ，并返回以此方式获得的最小割 S 。

现在, 有两种情况可能发生。要么 S 是一个 T -奇割, 这种情况下它必须是最小的并且我们完成了, 要么 S 是 T -偶割 (即 $T \cap S$ 的基数是偶数)。如果 S 是 T -偶割, 我们在下面的引理中证明我们可以假设最小的 T -偶割 A 要么是 S 的子集要么是 $V \setminus S$ 的子集。因此, 我们可以通过递归地解决两个较小的最小 T -奇割问题来找到, 一个在通过将 S 缩小为单个顶点而获得的图 $G_1 = G/S$ 中并且让 $T_1 = T \setminus S$, 另一个在通过将 $V \setminus S$ 缩小而获得的图 $G_2 = G/(V \setminus S)$ 中并且让 $T_2 = T \setminus (V \setminus S) = T \cap S$ 。因此, 算法进行两次调用, $ALG(G_1, T_1)$ 和 $ALG(G_2, T_2)$ 并返回返回的最小 (按容量计算) T -奇割。

乍一看, 这个算法是多项式的并不明显, 因为每次调用可能会生成两个递归调用。然而, 假设 $R(k)$ 表示对于 $|T| = k$ (并且 $|V| \leq n$) 的实例, $ALG(G, T)$ 的运行时间的上界, 我们可以看到

1. $R(2) = A$, 其中 A 是计算最小 $s-t$ 割的时间,

2. $R(k) \leq \max_{k_1 \geq 2, k_2 \geq 2, k=k_1+k_2} ((k-1)A + R(k_1) + R(k_2))$.

通过归纳, 我们可以看到 $R(k) \leq k^2 A$, 因为对于 $k=2$, 这是成立的, 并且归纳步骤也满足:

$$\begin{aligned} R(k) &\leq \max_{k_1 \geq 2, k_2 \geq 2, k=k_1+k_2} ((k-1)A + k_1^2 A + k_2^2 A) \\ &\leq (k-1)A + 4A + (k-2)^2 A \\ &= (k^2 - 3k + 7)A \\ &\leq k^2 A, \end{aligned}$$

对于 $k \geq 4$. 因此, 这个算法是多项式的。

我们需要陈述并证明以下引理。

引理 4.6 如果 S 是至少在每一侧都有一个 T 顶点的最小割, 且 $|S \cap T|$ 是偶数, 则存在一个最小 T -奇割 A , 其中 $A \subseteq S$ 或 $A \subseteq V \setminus S$ 。

证明: 设 B 为任意最小 T -奇割。将 T 划分为 T_1, T_2, T_3 和 T_4 如下所示: $T_1 = T \setminus (B \cup S)$, $T_2 = (T \cap S) \setminus B$, $T_3 = T \cap B \cap S$, $T_4 = (T \cap B) \setminus S$ 。根据 B 和 S 的定义, 我们有 $T_1 \cup T_2 = \emptyset$, $T_2 \cup T_3 = \emptyset$, $T_3 \cup T_4 = \emptyset$ 和 $T_4 \cup T_1 = \emptyset$, 因此我们必须要么 T_1 和 T_3 非空, 要么 T_2 和 T_4 非空。可能将 B 替换为 $V \setminus B$, 我们可以假设 T_1 和 T_3 非空。

通过割函数的次模性, 我们知道

$$\sum_{e \in \delta(S)} u(e) + \sum_{e \in \delta(B)} u(e) \geq \sum_{e \in \delta(S \cup B)} u(e) + \sum_{e \in \delta(S \cap B)} u(e). \quad (3)$$

由于 $T_1 = \emptyset$ 和 $T_3 = \emptyset$, $S \cup B$ 和 $S \cap B$ 都将顶点 T 分开。此外, 它们中的一个不必是 T -偶数, 另一个不必是 T -奇数, 因为 $|(S \cap B) \cap T| + |(S \cup B) \cap T| = |T_2| + 2|T_3| + |T_4| = |S \cap T| + |B \cap T|$ 是奇数。因此, $S \cup B$ 和 $S \cap B$ 中的一个的割值不大于 B 的割值, 而另一个的割值不大于 S 的割值。这意味着 $S \cap B$ 或 $S \cup B$ 是最小的 T -奇割。

△

参考文献

- [1] A.V. Goldberg 和 R.E. Tarjan, “最大流问题的新方法”, *ACM期刊*, **35**, 921–940, 1988.
- [2] X. Hao 和 J.B. Orlin, “在图中找到最小割的更快算法”, 第三届 *ACM-SIAM* 离散算法研讨会论文集, 165–174, 1992.
- [3] D. Karger, “近线性时间的最小割”, 第28届 *STOC* 研讨会论文集, 56–63, 1996.
- [4] D. Karger 和 C. Stein, “一个 $\tilde{O}(n^2)$ 的最小割算法”, 第25届 *STOC* 研讨会论文集, 757–765, 1993.
- [5] H. Nagamochi 和 T. Ibaraki, “计算多图和容量图的边连通性”, *SIAM* 离散数学期刊, **5**, 54–66, 1992.
- [6] M. Stoer and F. Wagner, “一个简单的最小割算法”, *ESA 94* 会议论文集, 计算机科学讲义, **855**, 141–147, 1994年。

5. 集合优化

5.1 集合优化的定义

集合优化是一种推广矩阵中线性独立概念的组合优化结构。集合优化有许多等价定义，我们将使用一种侧重于其独立集的定义。集合优化 M 定义在有限的基础集合 E 上（或者 $E(M)$ 如果我们想强调集合优化 M ）， E 的子集的集合被称为独立集。独立集的集合被表示为 \mathcal{I} 或 $\mathcal{I}(M)$ ，我们通常通过列出其基础集合和独立集的集合来引用一个集合优化 M ： $M = (E, \mathcal{I})$ 。对于 M 成为一个集合优化， \mathcal{I} 必须满足两个主要公理：

(I_1) 如果 $X \subseteq Y$ 且 $Y \in \mathcal{I}$ 则 $X \in \mathcal{I}$,

(I_2) 如果 $X \in \mathcal{I}$ 且 $Y \in \mathcal{I}$ 且 $|Y| > |X|$ 则 $\exists e \in Y \setminus X : X \cup \{e\} \in \mathcal{I}$.

换句话说，第二公理表明，如果 X 是独立的，并且存在一个更大的独立集合 Y ，那么可以通过添加 $Y \setminus X$ 的元素来扩展 X 为一个更大的独立集合。公理(I_2) 意味着每个最大（按包含关系）的独立集合都是最大的；换句话说，所有最大独立集合的基数相同。一个最大独立集合被称为该拟阵的一个基。

例子。

- 一个平凡的拟阵 $M = (E, \mathcal{I})$ 的一个例子是一个均匀拟阵，其中

$$\mathcal{I} = \{X \subseteq E : |X| \leq k\},$$

对于给定的 k . 通常表示为 $U_{k,n}$ 其中 $|E| = n$. 基是任何基数为 k 的集合（除非 $k > |E|$ ，在这种情况下，唯一的基是 $|E|$ ）。

自由拟阵是所有集合都是独立的拟阵；它是 $U_{n,n}$ 。

- 另一个是分割拟阵，其中 E 被分割成（不相交的）集合 E_1, E_2, \dots, E_l 和

$$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ 对于所有的 } i = 1, \dots, l\},$$

对于一些给定的参数 k_1, \dots, k_l . 作为练习，让我们检查一下 (I_2) 是否满足。如果 $X, Y \in \mathcal{I}$ 且 $|Y| > |X|$ ，那么必定存在 i 使得 $|Y \cap E_i| > |X \cap E_i|$ ，这意味着将 $E_i \cap (Y \setminus X)$ 中的任意元素 e 添加到 X 将保持独立性。

观察到如果集合 E_i 不是不相交的，则 M 将不是一个拟阵。例如，如果 $E_1 = \{1, 2\}$ 且 $E_2 = \{2, 3\}$ ，其中 $k_1 = 1$ 且 $k_2 = 1$ ，则 $Y = \{1, 3\}$ 和 $X = \{2\}$ 都最多只有一个 E_i 的元素，但无法找到一个 Y 的元素添加到 X 。

- 线性拟阵（或可表示拟阵）是由矩阵 A 定义的，这就是拟阵这个术语的来源。令 E 表示矩阵 A 的列索引集。对于 E 的子集 X ，令 A_X 表示 A 中仅包含那些由 X 索引的列的子矩阵。现在，定义

$$\mathcal{I} = \{X \subseteq E : \text{rank}(A_X) = |X|\},$$

即，如果相应的列线性无关，则集合 X 是独立的。基 B 对应于基数为 $\text{rank}(A)$ 的线性无关列集。

观察到 (I_1) 是显然成立的，因为如果列线性无关，则它们的子集也是线性无关的。 (I_2) 不太显然，但对应于一个基本的线性代数性质。如果 A_X 具有满列秩，它的列张成一个维度为 $|X|$ 的空间，类似地，对于 Y ，如果 $|Y| > |X|$ ，则必定存在一个 A_Y 的列不在 A_X 的列张成的空间中；将这个列添加到 A_X 会增加其秩1。

线性拟阵可以在任何域 \mathbb{F} 上定义（不仅仅是实数）；我们称其为在 \mathbb{F} 上可表示的拟阵。如果域是 \mathbb{F}_2 （2元素域，带有模2的运算），那么拟阵被称为二进制的。如果域是 \mathbb{F}_3 ，那么拟阵被称为三进制的。

例如，与矩阵对应的二进制拟阵

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

对应于 $U_{2,3}$ ，因为取模2时，3列的和为0向量。如果 A 在实数域或 \mathbb{F}_3 上观察，则拟阵是由3个元素构成的自由拟阵。

并非所有的拟阵都是线性的。在那些线性的拟阵中，有些可以在某些域 \mathbb{F} 上表示，但不是所有的。例如，有些二进制拟阵不是三进制的，反之亦然（例如， $U_{2,4}$ 是三进制的但不是二进制的）。可以在任何域上表示的拟阵被称为正则的。可以证明正则拟阵正好是那些可以通过完全单模矩阵在实数上表示的线性拟阵。（由于这种联系，Seymour的一个深刻结果提供了一个多项式时间算法来判断一个矩阵是否是 TU_0 。）

- 这是一个不是拟阵的例子。取一个图 $G = (V, E)$ ，并且令 $\mathcal{I} = \{F \subseteq E : F \text{ 是一个匹配}\}$ 。这不是一个拟阵，因为 (I_2) 不一定满足（ (I_1) 是满足的¹，然而）。例如，考虑一个有4个顶点的图令 $X = \{(2, 3)\}$ 和 $Y = \{(1, 2), (3, 4)\}$ 。既 X 又 Y 都是匹配，但是不能将 Y 的一条边添加到 X 中并仍然得到一个匹配。

¹当只有 (I_1) 满足时， (E, \mathcal{I}) 被称为一个独立系统。

- 然而，与二分图匹配相关的另一个拟阵与图 $G=(V, E)$ 相关，但这次的拟阵的底集对应于 V 。在匹配拟阵中， $\mathcal{I} = \{S \subseteq V : S \text{ 被某个匹配 } M \text{ 覆盖}\}$ 。在这个定义中，匹配不需要精确地覆盖 S ；其他顶点也可以被覆盖。
- 组合优化中一个非常重要的拟阵类是图拟阵（也称为环拟阵）。给定一个图 $G=(V, E)$ ，我们定义独立集为那些不包含任何环的边的子集，即森林。这被称为图拟阵 $M=(E, \mathcal{I})$ ，或 $M(G)$ 。

(我₁) 很明显满足。要检查 (我₂)，首先注意到如果 F 是一棵森林，那么图 (V, F) 的连通分量数目由 $\kappa(V, F) = |V| - |F|$ 给出。因此，如果 X 和 Y 是两个森林且 $|Y| > |X|$ ，则 $\kappa(V, Y) < \kappa(V, X)$ ，因此必定存在一条连接 X 的两个不同连通分量的边；将这条边添加到 X 会得到一个更大的森林。这证明了 (我₂)。

如果图 G 是连通的，则任何基底都对应于图的一棵生成树 T 。
如果原始图是不连通的，则基底对应于在 G 的每个连通分量中取一棵生成树。

图形拟阵是一个线性拟阵。我们首先证明可以选择域 \mathbb{F} 为实数。考虑矩阵 A ，每个顶点 $i \in V$ 对应一行，每条边 $(i, j) \in E$ 对应一列。对于列 (i, j) ，除了 i 或 j （任意一个）处为1，其他位置都为0，另一个位置为 -1 。为了证明原始拟阵 M 和新构造的线性拟阵 M' 的等价性，我们需要证明任何 M 的独立集在 M' 中也是独立的，反之亦然。这留作练习。

事实上，图形拟阵是正则的；可以用任何域 \mathbb{F} 表示。实际上，上述矩阵 A 可以证明是TU的。为了得到域 \mathbb{F} 的表示，只需将上述对 \mathbb{R} 的表示中的所有 -1 替换为1的加法逆元。

5.1.1 电路

在一个拟阵中，一个最小的（按包含关系）相关集被称为一个电路。在一个图拟阵 $M(G)$ 中，一个电路将是图 G 中一个常见的环的概念；在图拟阵中，要成为相关集，需要包含一个环和包含一个环的最小边集就是环本身。在一个分割拟阵中，一个电路将是一个集合 $C \subseteq E_i$ ，其中 i 为某个值，使得 $|C \cap E_i| = k_i + 1$ 。

根据电路 C 的定义，我们知道如果我们移除电路的任何一个元素，我们得到一个独立集。电路的一个关键属性由以下性质给出，

定理 5.1 (唯一电路性质) 设 $M = (E, \mathcal{I})$ 为一个拟阵。设 $S \in \mathcal{I}$ 且存在一个元素 e 使得² $S + e \in \mathcal{I}$ 。那么存在一个唯一的电路 $C \subseteq S + e$ 。

独特性非常重要。事实上，如果我们考虑任意 $f \in C$ 其中 C 是这个唯一的回路，那么我们有 $C + e - f \in \mathcal{I}$ 。事实上，如果 $C + e - f$ 是相关的，那么它将包含一个与 C 不同的回路 C' ，这与 $f \notin C'$ 矛盾。

作为定理的一个特殊情况，考虑一个图形拟阵。如果我们向一个森林添加一条边，并且结果图形有一个循环，那么它有一个唯一的循环。

证明：

假设 $S + e$ 包含多于一个回路，比如 C_1 和 C_2 ，其中 $C_1 \neq C_2$ 。通过 C_1 和 C_2 的最小性，我们知道存在 $f \in C_1 \setminus C_2$ 。由于 $C_1 - f \in \mathcal{I}$ (根据 C_1 的最小性)，我们可以将其扩展为 $S + e$ 的最大独立集 X of。由于 S 也是独立的，我们必须有 $|X| = |S|$ ，并且由于 $e \in C_1 - f$ ，我们必须有 $X = S + e - f \in \mathcal{I}$ 。但这意味着 $C_2 \subseteq S + e - f = X$ ，这与 C_2 是相关的矛盾。

△

练习5-1. 证明任何分割拟阵也是线性拟阵，其中 $\mathbb{F} = \mathbb{R}$ 。(不需要给出一个精确的矩阵 A 来表示它；只需证明其存在。)

练习5-2. 证明匹配拟阵确实是一个拟阵。

练习5-3. 证明 $U_{2,4}$ 在 \mathbb{F}_3 上是可表示的。

练习5-4. 考虑由 3×5 矩阵定义的线性拟阵 (在实数上)：

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 & -1 \\ 1 & 2 & 0 & 1 & -1 \end{pmatrix}.$$

底集 $E = \{1, 2, 3, 4, 5\}$ ，基数为5，对应于矩阵 A 的列，而独立集是线性无关的列集合 (在实数上)。

1. 给出这个拟阵的所有基。
2. 给出这个拟阵的所有电路。
3. 选择一个基 B 和一个不在 B 中的元素 e ，并验证对于 $B + e$ 的唯一电路性质。

练习5-5. 给定一个集合族 A_1, A_2, \dots, A_n (它们不一定是相交的)，一个横切集是一个集合 T ，使得 $T = \{a_1, a_2, \dots, a_n\}$ ，其中 a_i 是不同的，并且 $a_i \in A_i$ 对于所有的 i 。一个部分横切集是 A_i 的一个横切集 $A_{i_1}, A_{i_2}, \dots, A_{i_k}$ 对于一些 A_i 的子族。

证明所有部分横切集构成一个拟阵 (在基础集 $E = \cup A_i$ 上)。

(提示：考虑二分图匹配。)

² 对于一个集合 S 和一个元素 e ，我们通常用 $S + e$ 表示 $S \cup \{e\}$ ，用 $S - e$ 表示 $S \setminus \{e\}$ 。

练习5-6。设 $M=(E, \mathcal{I})$ 是一个拟阵。设 $k \in \mathbb{N}$ 并定义

$$\mathcal{I}_k = \{X \in \mathcal{I} : |X| \leq k\}.$$

证明 $M_k=(E, \mathcal{I}_k)$ 也是一个拟阵。这被称为截断拟阵。

练习5-7。如果对于任意两个集合 $A, B \in \mathcal{F}$, 要么 (i) $A \subseteq B$, 要么 (ii) $B \subseteq A$, 要么 (iii) $A \cap B = \emptyset$, 则集合族 \mathcal{F} 被称为层次的。假设我们有一个层次子集族 \mathcal{F} 和一个整数 k (对于每个集合 $A \in \mathcal{F}$)。证明 (E, \mathcal{I}) 定义了一个拟阵 (一个层次拟阵), 其中:

$$\mathcal{I} = \{X \subseteq E : |X \cap A| \leq k(A) \text{ 对于所有 } A \in \mathcal{F}\}.$$

5.2 集合优化

给定一个集合 $M=(E, \mathcal{I})$ 和一个成本函数 $c: E \rightarrow \mathbb{R}$, 我们对于找到一个最大总成本的独立集合 S 感兴趣, 即 $c(S) = \sum_{e \in S} c(e)$ 。这是一个基础问题。

如果所有 $c(e) \geq 0$, 那么这个问题等价于在集合中找到一个最大成本的基。

如果对于某个元素 e , $c(e) < 0$, 那么根据 (I_1) , e 将不会包含在任何最优解中, 因此我们可以从集合中删除这样的元素。在图论中, 如果集合 $M(G)$ 定义在一个连通图 G 上, 那么这个问题等价于最大生成树问题, 可以通过简单的贪心算法解决。

实际上, 这对于任何拟阵都是成立的, 这也是本节的主题。

我们描述的贪婪算法实际上对于每个 k , 返回一个最大化 $c(S)$ 的集合 S_k , 其中 S_k 是所有大小为 k 的独立集合中的最大集合。因此, 整体最优解可以通过输出其中的最佳解来简单地获得。贪婪算法如下所示: 对元素进行排序 (并重新编号),

使得 $c(e_1) \geq c(e_2) \geq \dots \geq c(e_{|E|})$. $S_0 = \emptyset$, $k=0$. 对于 j 从 1 到 $|E|$

```

. if  $S_k + e_j \in \mathcal{I}$  then
.    $k \leftarrow k + 1$ 
.    $S_k \leftarrow S_{k-1} + e_j$ 
.    $s_k \leftarrow e_j$ 
. 输出  $S_1, S_2, \dots, S_k$ 
```

定理5.2 对于任意的拟阵 $M=(E, \mathcal{I})$, 上述贪心算法可以找到每个 k 大小的最大代价独立集 S_k 中的一个。

证明: 假设不成立。设 $S_k = \{s_1, s_2, \dots, s_k\}$, 其中 $c(s_1) \geq c(s_2) \geq \dots \geq c(s_k)$, 且假设 T_k 有更大的代价 ($c(T_k) > c(S_k)$), 其中 $T_k = \{t_1, t_2, \dots, t_k\}$, 且 $c(t_1) \geq c(t_2) \geq \dots \geq c(t_k)$ 。设 p 是第一个满足 $c(t_p) > c(s_p)$ 的索引。由于 $|A| > |B|$, 存在 $t_i \notin B$ such that $B + t_i \in \mathcal{I}$ 。由于

$c(t_i) \geq c(t_p) > c(s_p)$ ，当考虑时应选择 t_i 。更加精确和详细地说，当考虑 t_i 时，贪心算法会检查是否可以将 t_i 添加到当前集合 S 中。但是由于 $S \subseteq B$ ，将 t_i 添加到 S 中应该得到一个独立集合（根据 (I_1) ），因为将其添加到 B 中会得到一个独立集合。这产生了矛盾并完成了证明。

△观察

到，只要 $c(s_k) \geq 0$ ，我们有 $c(S_k) \geq c(S_{k-1})$ 。因此，要找到所有独立集合中的最大成本集合，我们只需替换循环。对于 $j=1$ 到 $|E|$ 通过对于 $j=1$ 到 q

其中 q 是这样的 $c(e_q) \geq 0 > c(e_{q+1})$ ，并输出最后的 S_k 。

对于最大代价生成树问题，贪心算法简化为克鲁斯卡尔算法，该算法按照非递增的代价考虑边，并在之前选择的边中添加一条边，如果它不形成环。

可以证明贪心算法实际上刻画了拟阵。如果 M 是一个独立系统，即满足 (I_1) ，那么 M 是一个拟阵，当且仅当贪心算法对于每个 k 和每个代价函数找到一个最大代价为 k 的集合。

练习 5-8。我们有 N 个任务，每个任务需要一个单位的处理时间。所有任务在时间 0 可用，任务 j 的利润为 c_j ，截止时间为 d_j 。只有当任务 j 在时间 d_j 之前完成时，才能获得任务 j 的利润。问题是找到一个任务的排序，使得总利润最大化。首先，证明如果一组任务可以按时完成，那么如果按照它们的截止时间排序，它们仍然可以按时完成。现在，令 $E(M) = \{1, 2, \dots, n\}$ ，令 $\mathcal{I}(M) = \{J \subseteq E(M) : J \text{ 可以按时完成的}\}$ 。证明 M 是一个拟阵，并描述如何找到任务的最优排序。

5.3 集合系统的秩函数

类似于矩阵的秩的概念，可以为任何集合系统定义一个秩函数。集合系统 M 的秩函数，可以用 $r(\cdot)$ 或 $r_M(\cdot)$ 表示，定义如下：

$$r_M : 2^E \rightarrow \mathbb{N} : r_M(X) = \max\{|Y| : Y \subseteq X, Y \in \mathcal{I}\}.$$

以下是一些具体的秩函数：

- 对于线性集合系统， X 的秩恰好是与 X 中的列对应的矩阵 A 在 X 在线性代数意义下的秩。
- 对于分割集合系统 $M = (E, \mathcal{I})$ ，其中

$$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ for } i = 1, \dots, l\}$$

(将 E_i 的形成一个分割 E) 其秩函数为:

$$r(X) = \sum_{i=1}^l \min(|E_i \cap X|, k_i)。$$

•对于定义在图 $G=(V, E)$ 上的图形拟阵 $M(G)$, 秩函数等于:

$$r_{M(G)}(F) = n - \kappa(V, F),$$

其中 $n = |V|$, $\kappa(V, F)$ 表示具有边 F 的图的连通分量 (包括孤立顶点) 的数量。

任何拟阵 $M=(E, \mathcal{I})$ 的秩函数具有以下性质:

(R_1) $0 \leq r(X) \leq |X|$, 并且对于所有 $X \subseteq E$, 其值为整数

(R_2) $X \subseteq Y \Rightarrow r(X) \leq r(Y)$

(R_3) $r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)。$

最后一个属性被称为子模性, 是组合优化中的一个关键概念。

很明显, 根据定义, 任何秩函数都满足 (R_1) 和 (R_2)。证明秩函数满足子模性需要一个证明。

引理 5.3 任何拟阵的秩函数都是子模的。

证明: 考虑任意两个集合 $X, Y \subseteq E$ 。设 J 是 $X \cap Y$ 的极大独立子集; 因此, $|J| = r(X \cap Y)$ 。根据 (I_2), J 可以扩展为 X 的极大 (因此是最大) 独立子集, 称为 J_X 。我们有 $J \subseteq J_X \subseteq X$ 且 $|J_X| = r(X)$ 。此外, 根据 J 在 $X \cap Y$ 中的极大性, 我们知道 $J_X \setminus Y = J_X \setminus J$ 。

(1)

现在将 J_X 扩展为 $X \cup Y$ 的最大独立集 J_{XY} 。因此, $|J_{XY}| = r(X \cup Y)$ 。

为了能够证明

$$r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$$

或者等价地

$$|J_X| + r(Y) \geq |J| + |J_{XY}|,$$

我们需要证明 $r(Y) \geq |J| + |J_{XY}| - |J_X|$ 。观察到 $J_{XY} \cap Y$ 是独立的 (根据 (I_1)) , 且是 Y 的子集, 因此 $r(Y) \geq |J_{XY} \cap Y|$ 。现在观察到

$$J_{XY} \cap Y = J_{XY} \setminus (J_X \setminus Y) = J_{XY} \setminus (J_X \setminus J),$$

第一个等式是因为 J_X 是 X 的最大独立子集, 第二个等式是根据 (1) 得到的。因此,

$$r(Y) \geq |J_{XY} \cap Y| = |J_{XY} \setminus (J_X \setminus J)| = |J_{XY}| - |J_X| + |J|,$$

证明了引理。

△

5.3.1 跨度

下面的定义也受到线性代数背景的启发。

定义5.1 给定一个拟阵 $M=(E, \mathcal{I})$ 和给定 $S \subseteq E$ ，令

$$\text{span}(S) = \{e \in E : r(S \cup \{e\}) = r(S)\}.$$

集合的跨度也称为闭包。观察到 $S \subseteq \text{span}(S)$ 。我们声称 $r(S) = r(\text{span}(S))$ ；换句话说，如果将一个元素添加到 S 中不增加秩，那么添加多个这样的元素也不会增加秩。确实，取 S 的一个最大独立子集 J 。如果 $r(\text{span}(S)) > |J|$ ，则存在 $e \in \text{span}(S) \setminus J$ 使得 $J + e \in \mathcal{I}$ 。因此， $r(S + e) \geq r(J + e) = |J| + 1 > |J| = r(S)$ ，与 $e \in \text{span}(S)$ 的事实相矛盾。

定义5.2 如果集合 S 满足 $S = \text{span}(S)$ ，则称其为闭集。

练习题5-9。给定一个秩函数为 r 的拟阵 M 和一个整数 $k \in \mathbb{N}$ ，截断拟阵 M_k 的秩函数是什么（参见练习题5-6的定义）。

练习题5-10。什么是分层拟阵的秩函数，参见练习题5-7？

5.4 拟阵多面体

令

$$X = \{\chi(S) \in \{0, 1\}^{|E|} : S \in \mathcal{I}\}$$

记 $M=(E, \mathcal{I})$ 的所有独立集的关联（或特征）向量为 X ，并且定义拟阵多面体为 $\text{conv}(X)$ 。在本节中，我们通过线性不等式给出了 $\text{conv}(X)$ 的完整特征描述。此外，我们还展示了用于证明多面体完整描述的不同技术，这些技术在多面体章节中提出。

定理5.4 假设

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \leq r(S) & \forall S \subseteq E \\ x_e \geq 0 & \forall e \in E \end{array}\}$$

其中 $x(S) := \sum_{e \in S} x_e$ 。那么 $\text{conv}(X) = P$ 。

很明显 $\text{conv}(X) \subseteq P$ since $X \subseteq P$ 。更难的部分是要证明 $P \subseteq \text{conv}(X)$ 。在接下来的三个小节中，我们提供基于三种不同技术的三个不同证明来证明完整的多面体描述。

5.4.1 算法证明

在这里，我们提供基于贪婪算法的算法证明。从 $\text{conv}(X) \subseteq P$ ，我们知道

$$\max\{c^T x : x \in X\} = \max\{c^T x : x \in \text{conv}(X)\} \leq \max\{c^T x : \begin{array}{ll} x(S) \leq r(S) & S \subseteq E \\ x_e \geq 0 & e \in E \end{array}\}.$$

使用LP对偶性，我们得到最后一个表达式等于：

$$\min\left\{\sum_S r(S)y_S : \begin{array}{ll} \sum_{S:e \in S} y_S \geq c(e) & \forall e \in E \\ y_S \geq 0 & S \subseteq E \end{array}\right\}.$$

我们现在的目标是，对于任意的成本函数 c ，得到一个独立集 S 和一个对偶可行解 y such that $\sum_S r(S)y_S = c(S)$ (证明了 $\text{conv}(X) = P$). 考虑任意的代价函数 c . 我们知道最大

代价独立集可以通过贪心算法获得. 更准确地说，当我们只考虑那些元素直到 e_q ，其中 $c(e_q) \geq 0 \geq c(e_{q+1})$ 时，贪心算法返回的最后一个集合 S_k . 我们现在需要展示一个与 S_k 相同值的对偶解. 对偶中有指数多的变量，但这不是问题. 实际上，我们将大部分变量设置为 0.

对于任意的索引 $j \leq k$ ，我们有 $S_j = \{s_1, s_2, \dots, s_j\}$ ，我们定义 U_j 为我们排序中的所有元素，直到并排除 s_{j+1} ，即 $U_j = \{e_1, e_2, \dots, e_l\}$ 其中 $e_{l+1} = s_{j+1}$. 换句话说， U_j 是在 s_{j+1} 之前的排序中的所有元素. U_j 的一个重要性质是

$$r(U_j) = r(S_j) = j.$$

事实上，由于独立性， $r(S_j) = |S_j| = j$ ，根据 (R_2) ， $r(U_j) \geq r(S_j)$ 。如果 $r(U_j) > r(S_j)$ ，那么存在一个元素 $e \in U_j \setminus S_j$ ，使得 $S_j \cup \{e\} \in \mathcal{I}$ 。但是贪心算法会选择该元素（根据 (I_1) ），这与 $e \in U_j \setminus S_j$ 的事实相矛盾。

按照以下方式设置 y_S 的非零元素。对于 $j = 1, \dots, k$ ，令

$$y_{U_j} = c(s_j) - c(s_{j+1}),$$

在这里我们理解 $c(s_{k+1}) = 0$. 根据 $c(\cdot)$ 的排序，我们有 $y_S \geq 0$ 对于所有 S . 此外，对于任意的 $e \in E$ ，我们有

$$\sum_{S:e \in S} y_S = \sum_{j=t}^k y_{U_j} = c(s_t) \geq c(e),$$

其中 t 是最小的索引使得 $e \in U_t$ (意味着 e 不在 S_t 之前). 这表明 y 是对偶问题的可行解。此外，它的对偶值为：

$$\sum_S r(S)y_S = \sum_{j=1}^k r(U_j)y_{U_j} = \sum_{j=1}^k j(c(s_j) - c(s_{j+1})) = \sum_{j=1}^k (j - (j-1))c(s_j) = \sum_{j=1}^k c(s_j) = c(S_k).$$

这表明对偶解与贪婪算法输出的独立集具有相同的值，对于所有成本函数都成立。这完成了算法证明。

5.4.2 顶点证明

在这里，我们将重点放在任何顶点 x 上

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \leq r(S) & \forall S \subseteq E \\ x_e \geq 0 & \forall e \in E \end{array}\}$$

并且证明 x 是一个整数向量。由于 $x(\{e\}) \leq r(\{e\}) \leq 1$ ，我们得到 $x \in \{0, 1\}^{|E|}$ 并且因此它是一个独立集的关联向量。

对于任何 $x \in P$ ，考虑 tight 集 S ，即满足 $x(S) = r(S)$ 的集合。下一个引理表明这些 tight 集在取交集或并集时是封闭的。这个引理真的很核心，并且是由次模性得出的。

引理 5.5 设 $x \in P$. 设

$$\mathcal{F} = \{S \subseteq E : x(S) = r(S)\}.$$

那么

$$\text{若 } S \in \mathcal{F}, T \in \mathcal{F}, \text{ 则 } S \cap T \in \mathcal{F}, S \cup T \in \mathcal{F}.$$

观察到即使 S 和 T 是不相交的，引理仍然适用。在这种情况下，它表明 $\emptyset \in \mathcal{F}$

(这总是成立的，因为 $x(\emptyset) = 0 = r(\emptyset)$) 且 $S \cup T \in \mathcal{F}$ 。

证明：事实上， $S, T \in \mathcal{F}$ 意味着：

$$r(S) + r(T) = x(S) + x(T). \quad (2)$$

由于 $x(S) = \sum_{e \in S} x_e$ ，我们有

$$x(S) + x(T) = x(S \cap T) + x(S \cup T), \quad (3)$$

即函数 $x(\cdot)$ 是模块化的 (x 和 $-x$ 都是次模的)。由于 $x \in P$ ，我们知道

$x(S \cap T) \leq r(S \cap T)$ (即使 $S \cap T = \emptyset$ 也成立)，同样地 $x(S \cup T) \leq r(S \cup T)$ ；这意味着

$$x(S \cap T) + x(S \cup T) \leq r(S \cap T) + r(S \cup T). \quad (4)$$

根据次模性，我们有

$$r(S \cap T) + r(S \cup T) \leq r(S) + r(T). \quad (5)$$

结合(2)–(5)，我们得到

$$r(S) + r(T) = x(S) + x(T) = x(S \cap T) + x(S \cup T) \leq r(S \cap T) + r(S \cup T) \leq r(S) + r(T),$$

因此我们在整个过程中都有等式成立。这意味着 $x(S \cap T) = r(S \cap T)$ 和 $x(S \cup T) = r(S \cup T)$ ，即 $S \cap T$ 和 $S \cup T$ 在 \mathcal{F} 中。 \triangle

为了证明任何顶点或极点 P 是整数的，我们首先对 P 的任何面进行了特征化。链 \mathcal{C} 是一组集合，对于所有的 $S, T \in \mathcal{C}$ ，我们有 $S \subseteq T$ 或者 $T \subseteq S$ (或者两者都成立，如果 $S = T$)。

定理5.6 考虑 P 的任何面 F 。那么存在一个链 \mathcal{C} 和一个子集 $J \subseteq E$ ，使得：

$$F = \{x \in \mathbb{R}^{|E|} : \begin{aligned} x(S) &\leq r(S) & \forall S \subseteq E \\ x(C) &= r(C) & \forall C \in \mathcal{C} \\ x_e &\geq 0 & \forall e \in E \setminus J \\ x_e &= 0 & \forall e \in J. \end{aligned}\}$$

证明：根据多面体笔记的定理3.5，我们知道任何面都可以通过将 P 的一些不等式设置为等式来进行特征化。特别地， F 可以表示为

$$F = \{x \in \mathbb{R}^{|E|} : \begin{aligned} x(S) &\leq r(S) & \forall S \subseteq E \\ x(C) &= r(C) & \forall C \in \mathcal{F} \\ x_e &\geq 0 & \forall e \in E \setminus J \\ x_e &= 0 & \forall e \in J. \end{aligned}\}$$

为了证明这个定理，我们需要证明方程组：

$$x(C) = r(C) \quad \forall C \in \mathcal{F}$$

可以被一个等价的（子）系统所替代，其中 \mathcal{F} 被一个链 \mathcal{C} 所替代。为了等价，我们需要

$$\text{span}(\mathcal{F}) = \text{span}(\mathcal{C}), \text{ 其中 } \text{span}(\mathcal{L}) \text{ 表示}$$

$$\text{span}(\mathcal{L}) := \text{span}\{\chi(C) : C \in \mathcal{L}\}.$$

设 \mathcal{C} 是 \mathcal{F} 的一个最大子链，即 $\mathcal{C} \subseteq \mathcal{F}$ ， \mathcal{C} 是一个链，对于所有 $S \in \mathcal{F} \setminus \mathcal{C}$ ，存在 $C \in \mathcal{C}$ 使得 $S \subseteq C$ 且 $C \subseteq S$ 。我们声称 $\text{span}(\mathcal{C}) = \text{span}(\mathcal{F})$ 。

假设不成立，即 $H = \text{span}(\mathcal{F})$ 其中 $H := \text{span}(\mathcal{C})$ 。这意味着存在一个 $S \in \mathcal{F} \setminus \mathcal{C}$ 使得 $\chi(S) \notin H$ 但是 S 不能被添加到 \mathcal{C} 而不破坏链式结构。换句话说，对于任何这样的 S ，'链式违规'的集合

$$V(S) := \{C \in \mathcal{C} : C \subseteq S \text{ 且 } S \subseteq C\}$$

是非空的。在所有这样的集合 S 中，选择一个使得 $|V(S)|$ 尽可能小

（ $|V(S)|$ 不能为0，因为我们假设对于所有可能的 S ， $V(S) \neq \emptyset$ ）。现在固定一个集合 $C \in V(S)$ 。根据引理5.5，我们知道 $C \cap S \in \mathcal{F}$ 且 $C \cup S \in \mathcal{F}$ 。观察到 $\chi(C)$, $\chi(S)$, $\chi(C \cup S)$, $\chi(C \cap S)$ 之间存在线性依赖关系：

$$\chi(C) + \chi(S) = \chi(C \cup S) + \chi(C \cap S).$$

这意味着，由于 $\chi(C) \in H$ 和 $\chi(S) \notin H$ ，我们必须要么 $\chi(C \cup S) \notin H$ ，要么 $\chi(C \cap S) \notin H$ （否则 $\chi(S)$ 将在 H 中）。假设 $\chi(B) \notin H$ ，其中 B 可以是 $C \cup S$ 或 $C \cap S$ 。这是一个矛盾，因为 $|V(B)| < |V(S)|$ ，与我们选择的 S 相矛盾。事实上，我们可以看到 $V(B) \subset V(S)$ 且 $C \in V(S) \setminus V(B)$ 。 \triangle

作为一个推论，我们还可以得到一个类似的性质，对于一个极点，从定理3.6开始。

推论5.7 设 x 是 P 的任意一个极点。那么存在一个链 C 和一个子集 $J \subseteq E$ ，使得 x 是以下方程的唯一解：

$$\begin{aligned} x(C) &= r(C) & \forall C \in \mathcal{C} \\ x_e &= 0 & \forall e \in J. \end{aligned}$$

从这个推论中，每个极点的整数性质很容易得出。实际上，如果推论中给出的链包含 $C_1 \subset C_2 \subset \dots \subset C_p$ ，则系统简化为

$$\begin{aligned} x(C_i \setminus C_{i-1}) &= r(C_i) - r(C_{i-1}) & i = 1, \dots, p \\ x_e &= 0 & \forall e \in J, \end{aligned}$$

为了有唯一解，我们最好有 $|C_i \setminus C_{i-1} \setminus J| \leq 1$ ，对于所有的 i ，并且结果 x_e 的值是整数。由于 $0 \leq x_e \leq r(\{e\}) \leq 1$ ，我们有 x 是一个 $0-1$ 向量，因此 $x = \chi(S)$ 对于某个集合 S 。由于 $|S| \leq r(S) \leq |S|$ ，我们有 $|S| = r(S)$ ，因此 $S \in \mathcal{I}$ 并且因此 x 是一个独立集的关联向量。这完成了证明。

5.4.3 面证明

我们对定理 5.4 的面进行最后的证明。

首先，我们需要证明我们缺少任何等式。让我们专注于每个单例集合都是独立的（有趣的）情况： $\{e\} \in \mathcal{I}$ 对于每个 $e \in E$ 。在这种情况下，由于我们可以展示 X 中 $|E|+1$ 个仿射独立的点：零向量和所有单位向量 $\chi(\{e\})$ 对于每个 $e \in E$ ，所以 X 的维数为 $(\text{conv}(X)) = |E|$ 。因此，我们不需要任何等式。如果我们不假设每个单例集合都是独立的，请参见练习 5-11。

现在考虑 X 的任何一个面 F 。这个面是由一个有效的不等式 $\alpha^T x \leq \beta$ 引起的，其中 $\beta = \max \left\{ \sum_{e \in I} \alpha_e : I \in \mathcal{I} \right\}$ 。让

$$\mathcal{O} = \{I \in \mathcal{I} : \sum_{e \in I} \alpha_e = \beta\},$$

即 \mathcal{O} 是所有独立集合的集合，其关联向量属于该面。我们将证明在我们对 P 的描述中存在一个不等式，该不等式在所有集合 $I \in \mathcal{O}$ 的关联向量上都满足等式。

我们考虑两种情况。如果存在 $e \in E$ 使得 $\alpha_e < 0$ ，则 $I \in \mathcal{O}$ 意味着 $e \notin I$ ，这意味着我们的面 F 包含在由 $x_e \geq 0$ （这是我们对 P 的描述中的面）引起的面中。

对于另一种情况，我们假设对于所有 $e \in E$ ，我们有 $\alpha_e \geq 0$ 。我们可以进一步假设 $\alpha_{\max} := \max_{e \in E} \alpha_e > 0$ ，否则 F 是平凡的。现在，定义 S 为

$$S = \{e \in E : \alpha_e = \alpha_{\max}\}.$$

命题 5.8 对于任意 $I \in \mathcal{O}$ ，我们有 $|I \cap S| = r(S)$ 。

这意味着面 F 包含在由不等式 $x(S) \leq r(S)$ 引起的面中，因此我们在描述 P 时有一个引起每个凸包面的不等式。因此，我们对 $\text{conv}(X)$ 有一个完整的描述。

为了证明这个命题，假设 $|I \cap S| < r(S)$ 。因此 $I \cap S$ 可以扩展为一个独立集 $X \in \mathcal{I}$ ，其中 $X \subseteq S$ 且 $|X| > |I \cap S|$ 。令 $e \in X \setminus (I \cap S)$ ；观察到 $e \in S$ 是根据我们选择的 X 。由于 $\alpha e > 0$ ，我们有 $I + e \in \mathcal{I}$ ，因此存在一个回路 $C \subseteq I + e$ 。根据唯一回路性质（见定理 5.1），对于任意 $f \in C$ ，我们有 $I + e - f \in \mathcal{I}$ 。但是 $C \setminus S = \emptyset$ ，因为 $(I \cap S) + e \in \mathcal{I}$ ，因此我们可以选择 $f \in C \setminus S$ 。对于 $I + e - f$ 的代价满足：

$$c(I + e - f) = c(I) + c(e) - c(f) > c(I),$$

与 \mathcal{O} 的定义相矛盾。

5.5 面？

现在我们已经用线性不等式描述了拟阵多面体，人们可能会想知道这些（指数级多的）不等式中哪些定义了 $\text{conv}(X)$ 的面。

为了简单起见，让我们假设对于所有的 $e \in E$ （ e 属于某个独立集），都有 $r(\{e\}) = 1$ 。那么，每个非负约束都定义了 $P = \text{conv}(X)$ 的一个面。事实上，除了 $\chi(\{e\})$ 之外的零向量和所有单位向量都构成了 $|E|$ 个仿射独立的点，满足 $x_e = 0$ 。这意味着对应的面的维度至少为 $|E| - 1$ ，而由于 P 本身的维度为 $|E|$ ，所以该面是一个面。

我们现在考虑约束条件 $x(S) \leq r(S)$ ，其中 $S \subseteq E$ 。如果 S 不是闭合的（见定义 5.2），那么 $x(S) \leq r(S)$ 明确地不定义 $P = \text{conv}(X)$ 的一个面，因为它被约束条件 $x(\text{span}(S)) \leq r(S)$ 和 $x_e \geq 0$ 对于 $e \in \text{span}(S) \setminus S$ 所隐含。

另一种情况是，如果 S 可以表示为 $U = \emptyset$ 和 $S \setminus U = \emptyset$ 的不相交并集，并且 $r(U) + r(S \setminus U) = r(S)$ ，那么 $x(S) \leq r(S)$ 不明确地定义一个面。在这种情况下， S 的不等式由 U 和 $S \setminus U$ 的不等式隐含。

定义 5.3 如果没有 U 满足 $\emptyset \neq U \subset S$ 且 $r(S) = r(U) + r(S \setminus U)$ ，则 S 被称为不可分的。

根据我们刚才的讨论，对于 $x(S) \leq r(S)$ 来定义 $P = \text{conv}(X)$ 的一个面的必要条件是 S 是闭合且不可分的。虽然证明被省略了，但这也可以被证明是充分的。

举个例子，考虑一个分割拟阵与 $M = (E, \mathcal{I})$ 其中

$$\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i \text{ 对于所有的 } i = 1, \dots, l\},$$

对于不相交的 E_i 。假设对于所有的 i ， $k_i \geq 1$ 。这个拟阵的秩函数是：

$$r(S) = \sum_{i=1}^l \min(k_i, |S \cap E_i|).$$

对于一个集合 S 来说, 如果存在(i) $i \in \{1, \dots, l\}$ 使得 $S \subseteq E_i$, 并且(ii) $|S \cap E_i|$ 要么 ≤ 1 , 要么 $> k_i$ 对于每个 i 来说。此外, 对于 $S \subseteq E_i$ 来说, 如果 $|S \cap E_i| \geq k_i$, 则 $S \cap E_i = E_i$ 。因此, 描述一个分割鞍点多面体所需的唯一集合是(i) 对于 i 满足 $|E_i| > k_i$ 的集合 $S = E_i$, 以及(ii) 对于每个 $e \in E$ 的单元集合 $\{e\}$ 。因此, 分割鞍点多面体由以下给出:

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{l} x(E_i) \leq k_i \quad i \in \{1, \dots, l\} : |E_i| > k_i \\ 0 \leq x_e \leq 1 \quad e \in E \end{array}\}.$$

举个例子, 将 M 取为图形拟阵 $M(G)$ 。对于一组边 $F \subseteq E$ 要使其不可分割, 我们需要子图 (V, F) 要么是单条边, 要么只有一个非平凡的 (即具有多于1个顶点的) 2-连通分量³ (这是一个最大的顶点集, 对于其中的任意两个顶点, 存在两条 (内部) 顶点不相交的路径连接它们)。实际上, 如果我们将 F 划分为边集 F_1, \dots, F_c (对于那些不属于任何2-连通分量的边, 单独成组), 我们有 $c(F) = \sum_{i=1}^c c(F_i)$, 因此 c 必须为1, 才能使 F 不可分割。给定一组边 F , 其对于图形拟阵的生成 (span) 包括所有两个端点都在 F 的任何连通分量内的边; 这些边的添加不会增加最大森林的大小。因此, 对于 F 要不可分割且闭合, 要么它是单条边, 要么存在一个顶点集 $S \subseteq V$ 使得 $F = E(S)$ ($E(S)$ 表示所有两个端点都在 S 内的边) 且 $(S, E(S))$ 是2-连通的。因此 (图 $G = (V, E)$ 中所有森林的凸包的最小描述) 森林多面体由以下给出:

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{l} x(E(S)) \leq |S| - 1 \quad S \subseteq V, E(S) \text{ 2-连通或} \quad |S| = 2 \\ 0 \leq x_e \leq 1 \quad e \in E \end{array}\}.$$

(通常情况下, $x(E(S))$ 表示 $\sum_{e \in E(S)} x_e$ 。) 我们也可以包括所有的顶点集 S ; 当然, 这也会给出生成树多面体, 尽管不是最小描述。注意, 即使我们只包括定义面的不等式, 这个多面体描述仍然有非常多的不等式。

从这个描述中, 我们还可以很容易地推导出图的生成树多面体, 即图中所有生成树的关联向量的凸包。事实上, 这是通过将 $S = V$ ($x(E) \leq |V| - 1$) 的不等式替换为一个不等式得到的森林多面体的一个面:

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{l} x(E) = |V| - 1 \\ x(E(S)) \leq |S| - 1 \quad S \subset V, E(S) \text{ 2-连通或} \quad |S| = 2 \\ 0 \leq x_e \leq 1 \quad e \in E \end{array}\}.$$

例如, 如果图由3个只有一个公共顶点 (v) 的顶点不相交的循环组成, 则每个循环都形成一个2-连通分量。一般来说, 2-连通分量的集合上有一棵森林结构。

练习5-11. 设 $M = (E, \mathcal{I})$ 是一个拟阵, 令 $S = \{e \in E : \{e\} \in \mathcal{I}\}$. 证明 $\dim(\text{conv}(X)) = |S|$ (其中 X 是独立集的关联向量集合), 并证明 P 的描述具有所需数量的线性无关等式。

练习5-12. 设 $M = (E, \mathcal{I})$ 是一个拟阵, P 是相应的拟阵多面体, 即独立集的特征向量的凸包. 证明两个独立集 I_1 和 I_2 在 P 上相邻当且仅当 (i) $I_1 \subseteq I_2$ 且 $|I_1| + 1 = |I_2|$, 或者 (ii) $I_2 \subseteq I_1$ 且 $|I_2| + 1 = |I_1|$, 或者 (iii) $|I_1 \setminus I_2| = |I_2 \setminus I_1| = 1$ 且 $I_1 \cup I_2 \notin \mathcal{I}$.

6. 拟阵交集讲义

关于拟阵的一个好处是，简单的贪心算法可以优化其独立集或基数。与此同时，这也显示了使用拟阵的局限性：对于许多组合优化问题，贪心算法无法提供最优解。然而，在本章中我们将展示，一旦考虑两个拟阵的独立集的交集，拟阵的表达能力就会大大增强。

考虑两个拟阵 $M_1 = (E, \mathcal{I}_1)$ 和 $M_2 = (E, \mathcal{I}_2)$ 在相同的底集 E 上，以及考虑两个拟阵的公共独立集族 $\mathcal{I}_1 \cap \mathcal{I}_2$ 。这通常被称为两个拟阵的交集。

在本章中，我们将给出一些拟阵交集的例子，并展示找到两个拟阵的最大公共独立集可以高效地完成，并提供最大值的最小-最大关系。我们还考虑加权设置（推广了分配问题），尽管我们不会给出一般情况下的算法（尽管存在一种），我们只限制在特殊情况下，即树形问题。我们将通过表征拟阵交集多面体并给出其最小-最大关系（一个 $\text{NP} \cap \text{co-NP}$ 的特征化）来暗示一般情况下的算法。最后，我们还讨论了拟阵的并集；这是一种从其他拟阵构建拟阵的强大方法，在其中拟阵交集起着核心作用。（术语“拟阵并集”是误导性的，因为它与我们定义的拟阵交集不符合...它不对应于 $\mathcal{I}_1 \cup \mathcal{I}_2$ 。）

6.1 例子

6.1.1 二分图匹配

在一个二分图 $G = (V, E)$ 中，具有二分割 (A, B) 的匹配不构成一个拟阵的独立集。然而，它们可以被看作是两个拟阵的公共独立集；这是拟阵交集的典型例子。

设 M_A 是一个由集合 E 的划分构成的拟阵，其中 $E = \bigcup \{\delta(v) : v \in A\}$ ， $\delta(v)$ 表示与 v 相邻的边。注意这是一个划分，因为所有的边都恰好有一个端点在 A 中。我们还定义对于每个 $v \in A$ ， $kv = 1$ 。因此， M_A 的独立集族由以下集合组成

$$\mathcal{I}_A = \{F : |F \cap \delta(v)| \leq 1, \text{ 对于所有的 } v \in A\}.$$

换句话说，如果一组边对于 M_A 是独立的，那么它最多只有一条边与 A 的每个顶点关联（而与 B 的每个顶点关联的边可以有任意数量）。同样地，我们可以定义 $M_B = (E, \mathcal{I}_B)$ 为

$$\mathcal{I}_B = \{F : |F \cap \delta(v)| \leq 1, \text{ 对于所有 } v \in B\}.$$

现在观察到任何 $F \in \mathcal{I}_A \cap \mathcal{I}_B$ 都对应于 G 中的一个匹配，反之亦然。而 \mathcal{I}_A 和 \mathcal{I}_B 的最大公共独立集对应于 G 中的最大匹配。

6.1.2 有向树

给定一个有向图 $D = (V, A)$ 和一个特殊的根节点 $r \in V$ ，一个 r -有向树（或者只是有向树）是一个从根节点 r 指向其他节点的生成树（当视为无向图时）。因此，在一个 r -有向树中，每个顶点都可以从根节点 r 到达。由于 r -有向树没有指向根节点的弧，我们假设 D 没有这样的弧。

r -树可以被看作是同时在两个拟阵中独立的集合。让 G 表示忽略弧的方向而得到的 D 的无向对应物。

注意，如果在 D 中同时存在弧 $a_1 = (u, v)$ 和 $a_2 = (v, u)$ ，那么在 G 中也会有两条无向边，标记为 a_1 和 a_2 ，连接 u 和 v 。定义 $M_1 = (A, \mathcal{I}_1) = M(G)$ 为与 G 对应的图拟阵， $M_2 = (A, \mathcal{I}_2)$ 为分割拟阵，其中独立集是每个顶点 $v \neq r$ 最多有一条入弧的集合。换句话说，我们定义

$$\mathcal{I}_2 = \{F : |F \cap \delta^-(v)| \leq 1, \text{ 对于所有 } v \in V \setminus \{r\}\}.$$

其中 $\delta^-(v)$ 表示入度为 v 的弧集合 $\{(u, v) \in A\}$ 。因此，任何一个 r -树都是独立于两个子群 M_1 和 M_2 的。反之，任何一个在 M_1 和 M_2 中都是独立的集合 T ，且基数为 $|V| - 1$ （因此在两个子群中都是基）都是一个 r -树。

实际上，这样一个 T 在 G 中作为生成树， T 和任何顶点 v 之间都有唯一的路径；这条路径必须从根节点 r 指向其他节点，否则要么存在一条入弧指向 t ，要么存在两条入弧指向同一个顶点。

在最小成本树问题中，我们还给出了一个成本函数 $c: A \rightarrow \mathbb{R}$ ，我们希望找到最小成本的 r -树。这是最小生成树问题的有向对应，但是贪心算法不能解决这个问题。

6.1.3 定向

给定一个无向图 $G = (V, E)$ ，我们考虑将所有边定向为有向弧；也就是说，每条（无向）边¹ $\{u, v\}$ 要么被替换为从 u 到 v 的弧² (u, v) ，要么被替换为从 v 到 u 的弧 (v, u) 。我们的目标是，给定 $k: V \rightarrow \mathbb{N}$ ，判断是否存在一种定向，使得对于每个顶点 $v \in V$ ，顶点 v 的入度（进入 v 的弧的数量）不超过 $k(v)$ 。显然，这并不总是可能的，这个问题可以使用拟阵交集（或网络流）来解决。

为了通过拟阵交集来解决这个问题，考虑有向图 $D = (V, A)$ ，其中每条边 $e = \{u, v\}$ of E 被替换为两条弧 (u, v) 和 (v, u) 。使用

¹通常，我们用 (u, v) 来表示一个（无向）边。然而，在本节中，我们使用符号

$\{u, v\}$ 而不是 (u, v) 来强调边是无向的。

²在有向图的情况下，我们使用弧，而在无向图的情况下使用边。

将弧集合 A 作为基准集合，我们定义两个分割拟阵， M_1 和 M_2 。在 M_1 中独立的意思是，对于每个 $(u, v) \in E$ ，最多只能选择 $\{(u, v), (v, u)\}$ 中的一个，即 $\mathcal{I}_1 = \{$

$F \subseteq A : |F \cap \{(u, v), (v, u)\}| \leq 1, \text{ 对于所有 } (u, v) \in E\}$ 。在 M_2 中独立的意思是，对于每个 v ，最多可以选择 $k(v)$ 个弧在 $\delta^-(v)$ 中，即 $\mathcal{I}_2 = \{F \subseteq A$
 $: |F \cap \delta^-(v)| \leq k(v), \text{ 对于所有 } v \in V\}$ 。

观察到这确实定义了一个分割拟阵，因为对于所有的 v ， $\delta^-(v)$ 都是分割 A 的子集。

因此，如果存在一个具有所需入度限制的公共独立集合 M_1 和 M_2 ，使其基数恰好为 $|E|$ （在这种情况下，我们选择 (u, v) 或 (v, u) 但不选择两者都有），则存在一个满足要求的有向图。

6.1.4 色彩丰富的生成树

假设我们有一个无向图 $G = (V, E)$ ，每条边都有一种颜色。这可以通过将 E 表示为 $E_1 \cup \dots \cup E_k$ 的分割来表示，其中每个 E_i 表示一组具有相同颜色 i 的边。通过拟阵交集可以判断该图是否存在一棵生成树，其中所有边的颜色都不同的问题。这样的生成树被称为色彩丰富的。

多彩的生成树是图形拟阵 $M_1 = M(G)$ 的基，也是分割拟阵 $M_2 = (E, \mathcal{I}_2)$ 中的独立集，其中 $\mathcal{I}_2 = \{F : |F \cap E_i| \leq 1, \text{ 对于所有的 } i\}$ 定义。

6.1.5 两个森林的并集

在第6.5节中，我们展示了如何通过拟阵交集来判断一个图 G 是否有两个边不相交的生成树。

6.2 最大公共独立集

通常，一个问题是找到最大基数的公共独立集，另一个问题是证明它确实是最优的。这可以通过最小-最大关系来完成。

给定两个拟阵 $M_1 = (E, \mathcal{I}_1)$ 和 $M_2 = (E, \mathcal{I}_2)$ ，分别具有秩函数 r_1 和 r_2 ，考虑任意集合 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ 和任意集合 $U \subseteq E$ 。观察到

$$|S| = |S \cap U| + |S \cap (E \setminus U)| \leq r_1(U) + r_2(E \setminus U),$$

因为 $S \cap U$ 和 $S \cap (E \setminus U)$ 在 M_1 和 M_2 中都是独立的（根据性质 (I_1) ）；特别地（这似乎更弱一些）， $S \cap U$ 对于 M_1 是独立的，而 $S \cap (E \setminus U)$ 对于 M_2 是独立的。现在，我们可以对 S 取最大值，对 U 取最小值，并得出：

$$\max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S| \leq \min_{U \subseteq E} [r_1(U) + r_2(E \setminus U)].$$

有点令人惊讶的是，我们将证明我们总是有相等性：

定理6.1 (Matroid Intersection) 对于任意两个秩函数分别为 r_1 和 r_2 的拟阵 $M_1 = (E, \mathcal{I}_1)$ 和 $M_2 = (E, \mathcal{I}_2)$ ，我们有：

$$\max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S| = \min_{U \subseteq E} [r_1(U) + r_2(E \setminus U)]. \quad (1)$$

在描述证明这个定理的交叉集算法之前，我们先考虑一些特殊情况下的最小-最大结果。首先，观察到我们总是可以将注意力限制在对于交叉集 M_1 是闭合的集合 U 上。实际上，如果不是这种情况，我们可以用 $V = \text{span}_{M_1}(U)$ 替换 U ，这样我们就有 $r_1(V) = r_1(U)$ 且 $r_2(E \setminus V) \leq r_2(E \setminus U)$ 。这表明总是存在一个对于 M_1 是闭合的集合 U 达到最小值。类似地，我们可以假设 $E \setminus U$ 对于 M_2 是闭合的（但不能同时做出这两个假设）。

当将交叉集定理特化为本章前面讨论的图方向问题时，我们可以得出以下结论。

定理 6.2 $G = (V, E)$ 存在一种方向使得每个顶点 v 的入度至多为 $k(v)$ 对于每个 $v \in V$ 当且仅当对于所有的 $P \subseteq V$ 我们有³：

$$|E(P)| \leq \sum_{v \in P} k(v).$$

同样地，对于有色生成树，我们得到：

定理 6.3 给定一个图 $G = (V, E)$ 其中边 E_i 被着色为 i 对于 $i = 1, \dots, k$ ，如果删除任意 c 个颜色的边（对于任意 $c \in \mathbb{N}$ ）产生至多 $c + 1$ 个连通分量，则存在一棵有色生成树。

我们通过展示一个算法来证明定理 6.1，该算法可以找到两个拟阵的最大基数独立集合，并找到一个相应的集合 U 使得在 (1) 中我们有相等。对于算法，我们将从 $S = \emptyset$ 开始，在每一步中要么增加 S 要么产生一个给出相等的 U 。我们的算法将在一个称为交换图的结构上大量依赖。我们首先只关注一个拟阵。

定义6.1 给定一个拟阵 $M = (E, \mathcal{I})$ 和一个独立集 $S \in \mathcal{I}$ ，交换图 $\mathcal{G}_M(S)$ （或者简称 $\mathcal{G}(S)$ ）是具有二分划分 S 和 $E \setminus S$ 的二分图，如果 $S - y + x \in \mathcal{I}$ ，则 S 中的 y 和 $E \setminus S$ 中的 x 之间有一条边。

引理6.4 设 S 和 T 是 M 中的两个独立集，且 $|S| = |T|$ 。则在 $\mathcal{G}_M(S)$ 中存在一个完美匹配，连接 $S \setminus T$ 和 $T \setminus S$ 。

证明略。引理6.4的逆命题不成立。接下来我们证明一个命题，它是上述引理的部分逆命题。

³ $E(P)$ 表示两个端点都在 P 中的边的集合。

命题6.5 设 $S \in \mathcal{I}$, 其交换图 $\mathcal{G}_M(S)$ 。设 T 是一个满足 $|T| = |S|$ 且 $\mathcal{G}_M(S)$ 有唯一完美匹配的集合 $S \setminus T$ 和 $T \setminus S$ 。那么 $T \in \mathcal{I}$ 。

证明: 设 N 是唯一的匹配。将 N 中的边从 $T \setminus S = \{x_1, \dots, x_t\}$ 指向 $S \setminus T = \{y_1, \dots, y_t\}$, 其余的边从 $S \setminus T$ 指向 $T \setminus S$ 。如果我们收缩 N 的边, 观察到得到的有向图没有有向环, 否则, 我们在收缩之前可以找到一个交替环, 这将违反匹配的唯一性。因此, 可以对 $\mathcal{G}_M(S)$ 的顶点进行编号 (通过拓扑排序), 使得 (i) 匹配的端点按顺序编号, (ii) 所有边都从编号较小的顶点指向编号较大的顶点。因此, 对 $S \setminus T$ 和 $T \setminus S$ 进行编号, 使得 $N = \{(y_1, x_1), (y_2, x_2), \dots, (y_t, x_t)\}$, 且对于 $i < j$, (y_i, x_j) 不是一条边。

现在假设为了推导出矛盾, $T \notin \mathcal{I}$ 。那么 T 有一个回路 C 。取最小的 i 使得 $x_i \in C$ (因为 C 中至少存在一个元素在 $T \setminus S$ 中, 由于 $C \subseteq T$ 且 S 是独立的)。根据构造, 对于所有的 $x \in C - x_i$, (y_i, x) 不是一条边。这意味着对于所有的 $x \in C - x_i$, $x \in \text{span}(S - y_i)$ 。因此 $C - x_i \subseteq \text{span}(S - y_i)$, 所以 $\text{span}(C - x_i) \subseteq \text{span}(\text{span}(S - y_i)) = \text{span}(S - y_i)$ 。 C 是一个循环, 所以 $x_i \in \text{span}(C - x_i)$, 因此 $x_i \in \text{span}(S - y_i)$ 。这是一个矛盾, 因为根据假设, $(y_i, x_i) \in \mathcal{G}_M(S)$ 。

因此 T 必须在 \mathcal{I} 中, 这证明了命题。 \triangle

我们现在准备描述证明最小最大公式的算法。首先, 我们定义了一种新类型的交换图, 用于处理两个拟阵的情况。

定义6.2 对于 $S \in \mathcal{I}_1 \cap \mathcal{I}_{2, M_2}$, 交换图具有分割 S 和 $E \setminus S$ 的有向二分图, 如果 $S - y + x \in \mathcal{I}_1$, 则 (y, x) 是一条弧, 如果 $S - y + x \in \mathcal{I}_2$, 则 (x, y) 是一条弧。

还定义 $X_1 := \{x \in S \mid S + x \in \mathcal{I}_1\}$, 源集合, 和 $X_2 := \{x \in S \mid S + x \in \mathcal{I}_2\}$, 汇集集合。然后算法是找到一条路径 (我们称之为增广路径), 从 X_1 到 X_2 , 该路径不包含任何捷径 (从路径上的一个较早的顶点指向路径上的一个非相邻后继顶点的弧)。例如, 可以通过选择从 X_1 到 X_2 的最短路径来获得这个路径。然后用路径上的顶点集合 P 替换 S , 即将 S 替换为 $S \triangle P$ 。作为特殊情况, 如果 $X_1 \cap X_2 = \emptyset$, 则得到的路径由一个单一顶点组成, 我们可以将该元素直接添加到 S 中。如果没有这样的路径, 则设置 $U := \{z \in S \mid z \text{ 可以到达 } X_2 \text{ 中的某个顶点在 } \mathcal{D}_M\}$ 。或者, 我们可以将 $E \setminus U$ 定义为可以从 X_1 中的顶点到达的顶点的集合; 这可能会得到一个不同的集合。

为了证明这个算法的正确性, 我们需要展示

1. 当我们停止时, 集合 S 和 U 确实满足最小最大公式 (1) 中的等式。
2. 在算法的每个阶段, $S \triangle P \in \mathcal{I}_1 \cap \mathcal{I}_2$ 。

证明1: 首先注意到 $X_2 \subseteq U$ 并且 $X_1 \cap U = \emptyset$ (否则我们可以继续运行算法来增加 S 的大小)。我们声称 $r_1(U) = |S \cap U|$ 并且

综上所述, 这将意味着 $|S| = r_1(U) + r_2(E \setminus U)$, 这正是我们需要的。

首先假设 $|S \cap U| \neq r_1(U)$. 由于 $S \cap U \subseteq U$ 且 $S \cap U$ 是独立的, 这将意味着 $|S \cap U| < r_1(U)$. 那么必然存在某个 $x \in U \setminus S$ 使得 $(S \cap U) + x \in \mathcal{I}_1$. 由于 $S \in \mathcal{I}_1$, 我们可以重复将 S 的元素添加到 $(S \cap U) + x$ 中, 从而得到一个形如 $S + x - y$ 的集合, 其中 $y \in S \setminus U$ 且 $S + x - y \in \mathcal{I}_1$. 但是, 这样一来, (y, x) 就成为 \mathcal{D}_M 中的一条弧。这是一个矛盾, 因此我们必须有 $|S \cap U| = r_1(U)$.

现在假设 $|S \cap (E \setminus U)| \neq r_2(E \setminus U)$. 那么和之前一样, 我们必须有 $|S \cap (E \setminus U)| < r_2(E \setminus U)$. 因此存在一个 $x \in (E \setminus U) \setminus S$ 使得 $(S \cap (E \setminus U)) + x \in \mathcal{I}_2$. 所以, 根据之前的逻辑, 我们可以找到一个 $y \in S \setminus (E \setminus U)$ 使得 $S - y + x \in \mathcal{I}_2$. 但是 $S \setminus (E \setminus U) = S \cap U$, 所以我们有一个 $y \in S \cap U$ 使得 $S - y + x \in \mathcal{I}_2$. 但是 (x, y) 是 \mathcal{D}_M 中的一条弧。这是一个矛盾, 所以我们有 $|S \cap (E \setminus U)| = r_2(E \setminus U)$. Δ

证明2: 回想一下, 我们需要证明当 P 是一条从 X_1 到 X_2 的路径且没有捷径时, $S \Delta P \in \mathcal{I}_1 \cap \mathcal{I}_2$. 我们首先证明 $S \Delta P \in \mathcal{I}_1$. 我们首先定义一个新的拟阵 M'_1 从 M_1 中 $M'_1 := (E \cup \{t\}, \{J \mid J \setminus \{t\} \in \mathcal{I}_1\})$. 换句话说, 我们只是在拟阵中添加一个与其他元素都独立的新元素 $\{t\}$. 然后我们知道 $S \cup \{t\}$ 在 M'_1 和 M'_2 中都是独立的 (我们类似地定义 M'_2 与 M'_1 相同)。另一方面, 如果我们将 \mathcal{D}_M

$(S \cup \{t\})$ 作为 $\mathcal{D}_{M'_1}$ 的子图) 中存在一个完美匹配 $(S \cup \{t\})$ 在 $(S \cap P) \cup \{t\}$ 和 $P \setminus S$ 之间存在 (由 P 中 $(S \cup \{t\})$, 以及 $\{t\}$ 和 P 中的第一个顶点之间的弧。此外, 这个匹配是唯一的, 因为 P 没有捷径, 所以根据命题, 我们知道 $(S \cup \{t\}) \Delta P$ 在 M'_1 中是独立的, 因此 $S \Delta P$ 在 M_1 中是独立的。

证明 $S \Delta P \in \mathcal{I}_2$ 的方法是相同的, 只是这次匹配由 P 中也是 $\mathcal{D}_{M'_2}$ 中的弧组成

$(S \cup \{t\})$, 连同 $\{t\}$ 和 P 中的最后一个顶点之间的弧。 Δ

因此, 我们已经证明了我们的算法是正确的, 并且作为结果建立了极小极大公式。

练习6-1. 从两个独立集的最大公共子集的极小极大关系推导出König定理, 关于二分图中匹配的最大大小。

6.3 集合交的多面体

在本节中, 我们用线性不等式来描述集合交的多面体, 即由两个独立集的特征向量的凸包组成。设 $M_1 = (E, \mathcal{I}_1)$ 和 $M_2 = (E, \mathcal{I}_2)$ 是两个独立集, 且

$$X = \{\chi(S) \in \{0, 1\}^{|E|} : S \in \mathcal{I}_1 \cap \mathcal{I}_2\}.$$

主要结果是 $\text{conv}(X)$ 精确地由 matroid polytopes 的交集给出
对于 M_1 和 M_2 .

定理 6.6 让

$$P = \{x \in \mathbb{R}^{|E|} : \begin{array}{ll} x(S) \leq r_1(S) & \forall S \subseteq E \\ x(S) \leq r_2(S) & \forall S \subseteq E \\ x_e \geq 0 & \forall e \in E \end{array} \}.$$

然后 $\text{conv}(X) = P$.

我们的证明将基于顶点。我们将证明 P 的任何极点都是整数的，并且很容易看出它对应于一个公共独立集。证明将以一种微妙的方式依赖于完全单调性。尽管定义 P 的整体矩阵不是完全单调的，但我们将证明，对于每个极点 x^* ， x^* 可以被看作是一个基于完全单调矩阵的方程组的解。这是一种可以应用于许多情况的强大方法。

证明： 假设 x^* 是 P 的一个极点。我们知道一旦我们知道了在 P 的描述中紧束缚的不等式， x^* 就可以唯一地确定。让

$$\mathcal{F}_i = \{S \subseteq E : x^*(S) = r_i(S)\},$$

对于 $i=1,2$ 。我们知道 x^* 是以下方程的唯一解

$$\begin{array}{ll} x(S) = r_1(S) & S \in \mathcal{F}_1 \\ x(S) = r_2(S) & S \in \mathcal{F}_2 \\ x_e = 0 & e \in E_0. \end{array}$$

考虑对于 $i=1,2$ 的 M_i 的拟阵多面体 P_i ，并定义面 F_i of P_i
(for $i=1,2$) 为

$$F_i = \{x \in P_i : \begin{array}{ll} x(S) = r_1(S) & \forall S \in \mathcal{F}_i \\ x_e = 0 & \forall e \in E_0 \end{array} \}.$$

观察到 $F_1 \cap F_2 = \{x^*\}$. 此外，根据 matroid optimization 章节的定理 4.6，我们可以通过链 \mathcal{C}_i 来替代定义 F_i . 因此， x^* 是唯一的解

$$\begin{array}{ll} x(S) = r_1(S) & S \in \mathcal{C}_1 \\ x(S) = r_2(S) & S \in \mathcal{C}_2 \\ x_e = 0 & e \in E_0. \end{array}$$

在消除 E_0 中的所有变量后，该系统可以写成 $Ax = b$ ，其中 A 的行是 $\mathcal{C}_1 \cup \mathcal{C}_2$ 的特征向量。

这样的矩阵 A 是完全单调的，可以通过使用定理 3.14 来证明。

考虑任意行的子集；这对应于将我们的注意力限制在链 \mathcal{C}'_1 和 \mathcal{C}'_2 上。

首先考虑 \mathcal{C}'_1 。如果我们将最大的集合分配给 R_1 ，然后在考虑越来越小的集合时，交替分配给 R_2 和 R_1 ，我们得到

$$\sum_{i \in \mathcal{C}'_1 \cap R_1} a_{ij} - \sum_{i \in \mathcal{C}'_1 \cap R_2} a_{ij} \in \{0, 1\},$$

对于所有的 j 。如果对于 C'_2 ，我们从最大的集合开始放在 R_2 中，我们得到

$$\sum_{i \in C'_2 \cap R_1} a_{ij} - \sum_{i \in C'_2 \cap R_2} a_{ij} \in \{0, -1\},$$

对于所有 j 。结合两者，我们可以得出对于每个 j ，我们得到一个值在 $\{0, 1, -1\}$ 中，表明矩阵是完全单调的。因此， x^* 是整数，因此对应于一个公共独立集的特征向量。

△

6.4 有向树问题

最小成本 r -有向树是指在给定的有向图 $D = (V, A)$ ，一个根节点 $r \in V$ 和每条弧 $a \in A$ 的成本 c_a ，找到 D 中的一个 r -有向树，使其总成本最小。因此，这可以看作是一个加权拟阵交集问题，我们可以使用拟阵交集算法和结果的全部机制。然而，在这里，我们将使用类似于分配问题的匈牙利方法的概念来开发一个更简单的算法。我们将假设成本是非负的。

作为一个整数规划问题，可以如下表述。令 x_a 为 1，表示一棵 r -树的弧，我们有以下表述：

$$\text{最小化目标函数 } OPT \quad \sum_{a \in A} c_a x_a$$

满足条件：

$$\begin{aligned} \sum_{a \in \delta^-(S)} x_a &\geq 1 && \text{对于所有的 } S \subseteq V \setminus \{r\} \\ \sum_{a \in \delta^-(v)} x_a &= 1 && \text{对于所有的 } v \in V \setminus \{r\} \\ x_a &\in \{0, 1\} && a \in A. \end{aligned}$$

在这个表述中， $\delta^-(S)$ 表示弧的集合 $\{(u, v) \in A : u \notin S, v \in S\}$ 。可以验证，上述问题的任何可行解都对应于一棵 r -树的关联向量。注意，这个优化问题有指数数量的约束条件。我们将展示可以放宽整数限制条件为 $x_a \geq 0$ ，并且移除等式约束条件 \sum

$\sum_{a \in \delta^-(v)} x_a = 1$ ，仍然存在一个最优的 r -树，该树对于这个放松后的（现在是线性的）问题是最优的。放松后的线性问题（仍然有指数数量的约束）是：

$$LP = \min \sum_{a \in A} c_a x_a$$

满足以下条件：

$$(P) \quad \begin{aligned} \sum_{a \in \delta^-(S)} x_a &\geq 1 && \text{对于所有的 } S \subseteq V \setminus \{r\} \\ x_a &\geq 0 && a \in A. \end{aligned}$$

这个线性问题的对偶问题是：

$$LP = \max \sum_{S \subseteq V \setminus \{r\}} y_S$$

受限于：

$$(D) \quad \begin{aligned} \sum_{S: a \in \delta^-(S)} y_S &\leq c_a \\ y_S &\geq 0 && S \subseteq V \setminus \{r\}. \end{aligned}$$

算法将构建一棵树 T （以及相应的关联向量 x ，其中 $x_a = 1$ ，当 $a \in T$ 时，否则为0），以及一个可行的对偶解 y ，满足互补松弛条件，这将表明 T 对应于 (P) 的最优解，因此是最优的树。互补松弛条件是：

$$1. y_S > 0 \implies |T \cap \delta^-(S)| = 1, \text{ 并且}$$

$$2. a \in T \implies \sum_{S: a \in \delta^-(S)} y_S = c_a.$$

该算法将分为两个阶段。在第一阶段，它将构建一个对偶可行解 y 和一个从根节点到每个顶点的有向路径的集合 F 。这可能不是一个 r -树，因为可能有太多的弧。在 F 中的弧将满足上述条件2（但不满足条件1）。在第二阶段，算法将删除不必要的弧，并得到满足条件1的 r -树。

第一阶段初始化时 $F = \emptyset$ 和 $y_S = 0$ ，对于所有的 S 。当 F 不包含从根节点到每个顶点的有向路径时，算法选择一个集合 S ，使得 (i) 在 S 内， F 是强连通的（即每个顶点都可以到达每个顶点），并且 (ii) $F \cap \delta^-(S) = \emptyset$ 。这个集合 S 存在，因为我们可以收缩所有强连通分量，并在结果的无环有向图中找到一个顶点（可能来自强连通分量的收缩），它没有入弧（否则从该顶点回溯，我们要么到达根节点，要么发现一个新的有向环（我们可以收缩））。现在我们尽可能地增加 y_S ，直到出现一个新的不等式，比如对于弧 a_k ， $\sum_{S: a_k \in \delta^-(S)} y_S \leq c_{a_k}$ 变成一个等式。通过这样做，解决方案仍然满足对偶可行性并且仍然满足条件2。

我们现在可以将 a_k 添加到 F 中而不违反互补松弛条件2，然后

我们递增 k (在开始时我们初始化为 $k=1$)。然后我们继续选择另一个集合 S , 依此类推, 直到每个顶点都可以从 F 中的 r 到达。现在我们有这样一个集合 $F = \{a_1, a_2, \dots, a_k\}$ 和一个满足条件2的对偶可行解 y_0 。

在第2步中, 我们消除尽可能多的弧, 但是我们按照它们添加到 F 中的相反顺序考虑。因此, 我们让 i 从 k 到 1, 如果 $F \setminus \{a_i\}$ 仍然包含从 r 到每个顶点的有向路径, 我们将 a_i 从 F 中移除, 并继续。然后我们输出结果集 T 的弧。

第一个断言是 T 是一棵有根树。事实上, 我们断言它恰好有 $|V| - 1$ 条弧, 每个顶点 $v \in V \setminus \{r\}$ 都有一条入弧。事实上, 如果不是这样, 就会有两条弧 a_i 和 a_j 入射到某个顶点 v ; 假设 $i < j$ 。在反向删除步骤中, 我们应该删除 a_j ; 事实上, 从 r 通过 a_j 可达的任何顶点都可以通过 a_i 到达 (除非 a_i 是不必要的, 在这种情况下我们可以稍后摆脱 a_i)。

第二个 (也是最后一个) 断言是互补松弛条件1也得到满足。事实上, 假设不是这样, 并假设我们有一个集合 S , 其中 $y_S > 0$, 且 $|T \cap \delta^-(S)| > 1$ 。 S 在算法的某个时刻被选择, 并且在那个时候我们将 $a_k \in \delta^-(S)$ 添加到 F 中。由于在将 a_k 添加到 F 之前, $\delta^-(S)$ 中没有其他弧, 这意味着 $T \cap \delta^-(S)$ 中的所有其他弧必须具有形式 a_j , 其中 $j > k$ 。此外, 当选择 S 时, F 在 S 内部已经是强连通的; 这意味着从 S 内部的任何顶点, 可以使用 a_i (其中 $i < k$) 到达 S 内的任何其他顶点。我们断言, 在考虑删除 a_j 时, 它应该被删除。事实上, 假设需要 a_j 去到顶点 v , 并且沿着路径 P 到 v 时, S 中的最后一个顶点是 $w \in S$ 。然后, 我们可以使用导致 S 中某个地方的 a_k 到达 v , 然后沿着路径 P 继续使用 a_i (其中 $i < k$, 这些弧都还没有被删除) 到达 $w \in S$ 。因此, a_j 实际上是不必要的, 应该被删除。这表明互补松弛条件1也得到满足, 因此构建的有根树是最优的。

6.5 集合系统的并

从任意集合系统 $M = (E, \mathcal{I})$ 可以构造出对偶集合系统 $M^* = (E, \mathcal{I}^*)$ 。

定理 6.7 令 $\mathcal{I}^* = \{X \subseteq E : E \setminus X \text{ 包含 } M \text{ 的一个基}\}$ 。那么 $M^* = (E, \mathcal{I}^*)$ 是一个集合系统, 其秩函数为

$$r_{M^*}(X) = |X| + r_M(E \setminus X) - r_M(E)。$$

有几种方法可以证明这一点。其中一种方法是首先证明 X 中最大子集的大小确实是 $|X| + r_M(E \setminus X) - r_M(E)$, 然后证明 M^* 的秩函数满足集合系统秩函数需要满足的三个条件 (第三个条件, 次模性, 可以从 M 的秩函数的次模性推导出来)。可以使用定理 6.7 和集合系统交的方法来得到图 $G = (V, E)$ 有两个边不相交的生成树的一个很好的刻画。

事实上, 令 M 为图 G 的图集合系统, 当且仅当 G 有两个边不相交的生成树时, 我们得到

$$\max_{S \in \mathcal{I} \cap \mathcal{I}^*} |S| = |V| - 1。$$

对于图形拟阵, 我们知道 $r_M(F) = n - \kappa(F)$ 其中 $n = |V|$ 且 $\kappa(F)$ 表示 (V, F) 的连通分量数。但是根据拟阵交集定理, 我们可以写成:

$$\begin{aligned} \max_{S \in \mathcal{I} \cap \mathcal{I}^*} |S| &= \min_{E_1 \subseteq E} [r_M(E_1) + r_{M^*}(E \setminus E_1)] \\ &= \min_{E_1 \subseteq E} [(n - \kappa(E_1)) + (|E \setminus E_1| + \kappa(E) - \kappa(E_1))] \\ &= \min_{E_1 \subseteq E} [n + 1 + |E \setminus E_1| - 2\kappa(E_1)], \end{aligned}$$

我们将 $\kappa(E)$ 替换为1, 因为否则 G 将会有有一个生成树。重新排列项, 我们得到 G 有两个边不相交的生成树, 当且仅当对于所有的 $E_1 \subseteq E$, 我们有 $|E \setminus E_1| \geq 2(\kappa(E_1) - 1)$ 。如果对于某个 E_1 , 这个不等式被违反了, 我们可以添加任何不减少 $\kappa(E_1)$ 的边到 E_1 。换句话说, 如果 E_1 的连通分量是 V_1, V_2, \dots, V_p , 那么我们可以假设 $E_1 = E \setminus \delta(V_1, V_2, \dots, V_p)$ 其中 $\delta(V_1, \dots, V_p) = \{(u, v) \in E : u \in V_i, v \in V_j \text{ 且 } i = j\}$ 。因此我们已经证明了:

定理6.8 G 如果且仅如果对于所有的划分 V_1, V_2, \dots, V_p of V , G 有两个边不相交的生成树

$$|\delta(V_1, \dots, V_p)| \geq 2(p-1).$$

定理6.8可以推广到任意数量的边不相交的生成树。
这个结果在这里没有被证明。

定理6.9 G 如果且仅如果对于所有的划分 V_1, V_2, \dots, V_p of V , G 有 k 个边不相交的生成树

$$|\delta(V_1, \dots, V_p)| \geq k(p-1).$$

从两个拟阵 $M_1 = (E, \mathcal{I}_1)$ 和 $M_2 = (E, \mathcal{I}_2)$, 我们也可以定义它们的并集为 $M_1 \cup M_2 = (E, \mathcal{I})$ 其中 $\mathcal{I} = \{S_1 \cup S_2 : S_1 \in \mathcal{I}_1, S_2 \in \mathcal{I}_2\}$. 请注意, 我们不要求两个拟阵相同, 就像我们刚刚对于边不相交的生成树所做的那样。

我们可以证明:

定理6.10 (拟阵的并) $M_1 \cup M_2$ 是一个拟阵。此外, 它的秩函数定义如下:

$$r_{M_1 \cup M_2}(S) = \min_{F \subseteq S} \{|S \setminus F| + r_{M_1}(F) + r_{M_2}(F)\}.$$

证明: 为了证明它是一个拟阵, 假设 $X, Y \in \mathcal{I}$ 且 $|X| < |Y|$ 。我们还可以进一步假设 X_i 的不相交, Y_i 的也不相交。最后, 我们假设在所有的选择中, X_1, X_2, Y_1 和 Y_2 的选择使得 $|X_1 \cap Y_1| + |X_2 \cap Y_2|$ 最大。由于 $|Y| > |X|$, 我们可以假设 $|Y_1| > |X_1|$ 。因此, 存在一个元素 $e \in (Y_1 \setminus X_1)$, 使得 $X_1 \cup \{e\}$ 对于 M_1 是独立的。最大性意味着 $e \notin X_2$ (否则考虑

$X_1 \cup \{e\}$ 和 $X_2 \setminus \{e\}$). 但这意味着 $X \cup \{e\} \in \mathcal{I}$, 正如我们所希望的。

我们现在展示秩函数的表达式。它是 \leq 是显而易见的，因为一个独立集合 $S \in \mathcal{I}$ 的大小 $|S \setminus F| + |S \cap F| \leq |S \setminus F| + r_{M_1}(F) + r_{M_2}(F)$ 对于任何 F 都成立。

对于逆命题，让我们证明对于整个基准集 $S = E$ 成立。一旦我们证明了

$$r_{M_1 \cup M_2}(E) = \min_{F \subseteq S} \{|E \setminus F| + r_{M_1}(F) + r_{M_2}(F)\},$$

对于任何集合 S 的相应命题将通过将我们的拟阵限制在 S 上来得到。让 X 是 $M_1 \cup M_2$ 的一个基。事实上 $X \in \mathcal{I}$ 意味着 $X = X_1 \cup X_2$ ，其中 $X_1 \in \mathcal{I}_1$ 且 $X_2 \in \mathcal{I}_2$ 。我们还可以假设 X_1 和 X_2 是不相交的，并且 $r_{M_2}(X_2) = r_{M_2}(E)$ （否则将元素添加到 X_2 中，并可能从 X_1 中删除它们）。

因此我们可以假设 $|X| = |X_1| + r_{M_2}(E)$ 。我们有 $X_1 \in \mathcal{I}_1$ ，并且 X_1 对于 M_2 的对偶是独立的（因为 X_1 的补集包含 M_2 的一个基）。换句话说， $X_1 \in \mathcal{I}_1 \cap \mathcal{I}_2^*$ 。通过使用拟阵交集定理和定理6.7，证明完成：

$$\begin{aligned} r_{M_1 \cup M_2}(E) = |X| &= \max_{X_1 \in \mathcal{I}_1 \cap \mathcal{I}_2^*} (|X_1| + r_{M_2}(E)) \\ &= \min_{E_1 \subseteq E} (r_{M_1}(E_1) + \text{半径}_{M_2^*}(E \setminus E_1) + \text{半径}_M(E)) \\ &= \min_{E_1 \subseteq E} (\text{半径}_{M_1}(E_1) + |E \setminus E_1| + \text{半径}_M(E_1) - \text{半径}_{M_2}(E) + \text{半径}_{M_2}(E)) \\ &= \min_{E_1 \subseteq E} (|E \setminus E_1| + r_{M_1}(E_1) + \text{半径}_{M_2}(E_1)), \end{aligned}$$

按要求。 △

由定理6.10可知 $M_1 \cup M_2$ 是一个拟阵，我们知道它的秩函数是次模的。然而，从定理中给出的公式中并不明显。

6.5.1 生成树游戏

生成树游戏是一个2人游戏。每个玩家轮流选择一条边。玩家1首先删除一条边，然后玩家2固定一条边（尚未被删除的边）；固定的边后来不能被另一位玩家删除。如果玩家2成功构建了图的生成树，则玩家2获胜；否则，玩家1获胜。问题是哪些图对于玩家1来说有获胜策略（无论对手做什么），哪些图对于玩家2来说有获胜策略。

定理6.11对于图 $G = (V, E)$ 上的生成树游戏，如果玩家1有一个获胜策略，那么 G 不具有两个边不相交的生成树。否则，玩家2有一个获胜策略。

如果 G 没有两个边不相交的生成树，那么根据定理6.8，我们知道存在一个划分 V_1, \dots, V_p 使得 $|\delta(V_1, \dots, V_p)| \leq 2(p-1) - 1$ 。玩家1的获胜策略是始终从 $\delta(V_1, \dots, V_p)$ 中删除一条边。由于玩家1先行动，所以在玩家2能够修复 $p-1$ 条边之前， $\delta(V_1, \dots, V_p)$ 中的边将被用尽，因此玩家2失败。反之是练习6-4的主题。

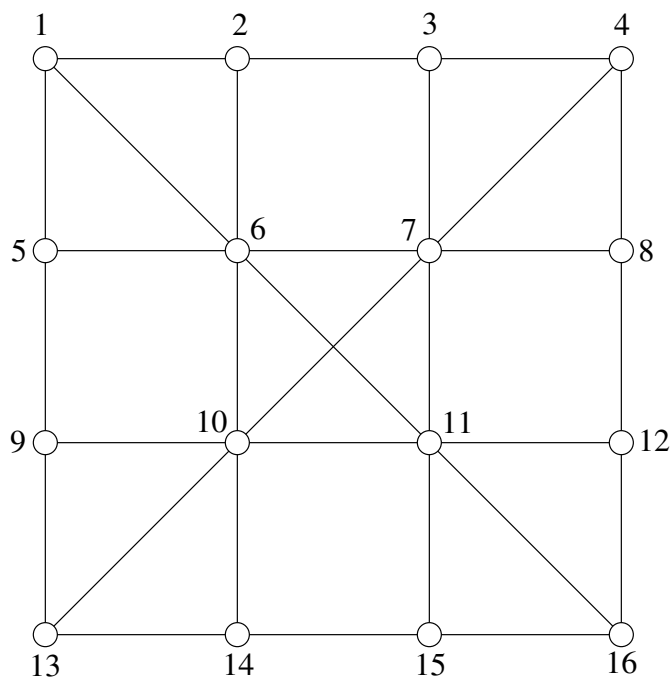
练习6-2. 从定理6.10推导出 k 个 M_1, M_2, \dots, M_k 的并集是一个拟阵, 其秩函数为

$$r_{M_1 \cup M_2 \cup \dots \cup M_k}(S) = \min_{F \subseteq S} \{|S \setminus F| + r_{M_1}(F) + r_{M_2}(F) + \dots + r_{M_k}(F)\}.$$

练习6-3. 从练习6-2推导出定理6.9.

练习6-4. 假设 G 有2个边不相交的生成树. 给出第2个玩家在生成树游戏中的获胜策略。

练习6-5. 在具有16个顶点和30条边的图中找到两个边不相交的生成树, 或证明不存在这样的树。



7. 椭球算法讲义

单纯形算法是最早提出的线性规划算法，虽然该算法在实践中非常快速，但目前尚无已知的多项式时间变体。椭球算法是最早发现的线性规划多项式时间算法。椭球算法由俄罗斯数学家Shor于1977年提出，用于解决一般的凸优化问题，并由Khachyan于1979年应用于线性规划。与单纯形算法相反，椭球算法在实践中并不是非常快速；然而，它的理论多项式性对组合优化问题有重要的影响，我们将会看到。另一个多项式时间算法，或者更准确地说是一类算法，用于线性规划的是内点算法，从1984年的Karmarkar算法开始；内点算法也非常实用，但对组合优化问题的影响不如椭球算法重要。

椭球算法考虑的问题是：

给定一个有界凸集 $P \in \mathbb{R}^n$ ，找到一个点 $x \in P$ 。

我们将看到，我们可以将线性规划简化为寻找一个点 x 在 P 中的问题，其中 $P = \{x \in \mathbb{R}^n : Cx \leq d\}$ 。

椭球算法的工作原理如下。我们从一个包含 P 的大椭球 E 开始。然后我们检查椭球的中心是否在 P 中。如果是的话，我们就完成了，找到了 P 中的一个点。否则，我们找到一个不满足我们中心点的不等式 $c^T x \leq d_i$ ，该不等式对 P 中的所有点都成立（例如，在 P 的描述中明确给出），但对我们的中心点不成立。椭球算法的一个迭代过程如图7.1所示。椭球算法如下所示。

- 让 E_0 是包含 P 的椭球

- 当椭球 E 的中心 a_k 不在 P 中时，执行以下操作：

- 让 $c^T x \leq c^T a_k$ ，使得 $\{x : c^T x \leq c^T a_k\} \supseteq P$
- 让 E_{k+1} 是包含 $E_k \cap \{x : c^T x \leq c^T a_k\}$ 的最小体积椭球
- 令 $k \leftarrow k + 1$

椭球算法具有重要的性质，即随着算法的进行，构造的椭球体积逐渐缩小；这在下一个引理中被准确地说明。这意味着如果集合 P 的体积为正，我们最终将找到一个点在 P 中。我们需要处理的情况是当集合 P 没有体积（即 P 只有一个点）时，以及讨论何时可以停止并确保我们要么有一个点在 P 中，要么知道 P 为空。

引理 7.1 $\frac{\text{Vol}(E_{k+1})}{\text{Vol}(E_k)} < e^{-\frac{1}{2(n+1)}}.$

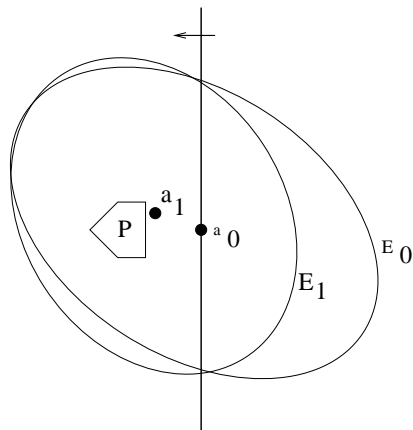


图 7.1：椭圆体算法的一次迭代。

在我们能够更准确地陈述算法之前，我们需要定义椭圆体。

定义 7.1 给定一个中心 a 和一个正定矩阵 A ，椭圆体 $E(a, A)$ 被定义为 $\{x \in \mathbb{R}^n: (x - a)^T A^{-1}(x - a) \leq 1\}$ 。

关于正定矩阵 A 的一个重要事实是存在 B 使得 $A = B^T B$ ，因此 $A^{-1} = B^{-1}(B^{-1})^T$ 。事实上，椭圆体实际上只是单位球的仿射变换。为了看到这一点，考虑到 (双射) 仿射变换 $T: x \rightarrow y = (B^{-1})^T(x - a)$ 。它将 $E(a, A)$ 映射到 $\{y: y^T y \leq 1\} = E(0, I)$ 。

我们首先考虑简单情况，即椭球 E_k 为单位球体，我们生成的不等式为 $x_1 \geq 0$ 。我们声称包含 $E_k \cap \{x: x_1 \geq 0\}$ 的椭球是

$$E_{k+1} = \left\{ x : \left(\frac{n+1}{n} \right)^2 \left(x - \frac{1}{n+1} \right)^2 + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \leq 1 \right\}.$$

事实上, 如果我们考虑一个属于 $E_k \cap \{x : x_1 \geq 0\}$ 的 x , 我们可以看到

$$\begin{aligned}
 & \left(\frac{n+1}{n}\right)^2 \left(x - \frac{1}{n+1}\right)^2 + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \\
 &= \frac{n^2+2n+1}{n^2} x_1^2 - \left(\frac{n+1}{n}\right)^2 \frac{2x_1}{n+1} + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \\
 &= \frac{2n+2}{n^2} x_1^2 - \frac{2n+2}{n^2} x_1 + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=1}^n x_i^2 \\
 &= \frac{2n+2}{n^2} x_1(x_1-1) + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=2}^n x_i^2 \\
 &\leq \frac{1}{n^2} + \frac{n^2-1}{n^2} \leq 1.
 \end{aligned}$$

在这个特殊情况下, 我们可以轻松证明引理7.1。

证明: 椭球的体积与其边长的乘积成比例。

因此, 单位椭球 E_k 和 E_{k+1} 之间的比率是

$$\begin{aligned}
 \frac{Vol(E_{k+1})}{Vol E_k} &= \frac{\left(\frac{n}{n+1}\right) \left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}}}{1} \\
 &= \left(\frac{n}{n+1}\right) \left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}} \\
 &< e^{-\frac{1}{n+1}} e^{\frac{n-1}{(n^2-1)2}} = e^{-\frac{1}{n+1}} e^{\frac{1}{2(n+1)}} = e^{-\frac{1}{2(n+1)}},
 \end{aligned}$$

在这里我们使用了对于所有的 x , 有 $1+x \leq e^x$ 的事实, 如果 $x=0$, 则严格不等式成立。 \triangle

现在考虑稍微一般一点的情况, 椭球也是单位球但是我们有一个任意的约束 $d^T x \leq 0$ 。我们想要找到一个包含 $E(0, I) \cap \{x : d^T x \leq 0\}$ 的椭球 (我们令 $\|d\|=1$; 这可以通过缩放两边来实现), 很容易验证我们可以取 $E_{k+1} = E(-$

$\frac{1}{n+1}d, F)$, 其中 $F = \frac{n^2}{n^2-1}(I - \frac{2}{n+1}dd^T)$, 且体积比率为 $\leq \exp\left(-\frac{1}{2(n+1)}\right)$ 。

现在我们处理 E_k 不是单位球的情况。我们利用线性变换保持体积比率的事实。

$$\begin{array}{ccc}
 E_k & \xrightarrow{T} & E(0, 1) \\
 & & \downarrow \\
 E_{k+1} & \xleftarrow{T^{-1}} & E'
 \end{array} \tag{1}$$

设 a_k 为 E_k 的中心, $c^T x \leq c^T a_k$ 为通过 a_k 的半空间, 包含 F 的半椭球为 $E(a_k, A) \cap \{x : c^T x \leq c^T a_k\}$ 。

让我们看看在变换 T 定义为 $y =$

$T(x) = (B^{-1})^T(x - a)$ 后, 这个半椭球会发生什么。这个变换将 $E_k = E(a_k, A)$ 转换为 $E(0, I)$ 。此外,

$$\{x : c^T x \leq c^T a_k\} \xrightarrow{T} \{y : c^T(a_k + B^T y) \leq c^T a_k\} = \{y : c^T B^T y \leq 0\} = \{y : d^T x \leq 0\}, \quad (2)$$

其中 d 由以下方程给出:

$$d = \frac{Bc}{\sqrt{c^T B^T B c}} = \frac{Bc}{\sqrt{c^T A c}}. \quad (3)$$

令 $b = B^T d = \frac{Ac}{\sqrt{c^T A c}}$ 。这意味着:

$$E_{k+1} = E\left(a_k - \frac{1}{n+1}b, \frac{n^2}{n^2-1}B^T\left(I - \frac{2}{n+1}dd^T\right)B\right) \quad (4)$$

$$= E\left(a_k - \frac{1}{n+1}b, \frac{n^2}{n^2-1}\left(A - \frac{2}{n+1}bb^T\right)\right). \quad (5)$$

总结一下, 这是椭圆体算法:

1. 从 $k=0$, $E_0 = E(a_0, A_0) \supseteq P$, $P = \{x : Cx \leq d\}$ 开始。

2. 当 $a_k \in P$ 时执行以下操作 :

- 令 $c^T x \leq d$ 为一个对于所有 $x \in P$ 但 $c^T a_k > d$ 成立的不等式。
- 令 $b = \frac{A_k c}{\sqrt{c^T A_k c}}$ 。
- 令 $a_{k+1} = a_k - \frac{1}{n+1}b$ 。
- 令 $A_{k+1} = \frac{n^2}{n^2-1}(A_k - \frac{2}{n+1}bb^T)$ 。

命题 7.2 $\frac{Vol(E_{k+1})}{Vol(E_k)} < \exp\left(-\frac{1}{2(n+1)}\right)$ 。

经过 k 次迭代, $Vol(E_k) \leq Vol(E_0) \exp\left(-\frac{k}{2(n+1)}\right)$ 。如果 P 非空, 则椭球算法最多在 $2(n+1) \ln \frac{Vol(E_0)}{Vol(P)}$ 步内找到一个 $x \in P$ 。

一般来说, 对于描述为 $P = \{x : Cx \leq d\}$ 的满维多面体, 可以证明 $\frac{Vol(E_0)}{Vol(P)}$ 在编码长度上是多项式的 C 和 d 。我们不会一般性地展示这一点, 但我们专注于组合优化中最重要的情况, 即给定一个集合 $S \subseteq \{0, 1\}^n$ (不是明确地给出, 但例如作为图中所有匹配的关联向量), 我们希望在 $P = \text{conv}(S)$ 上进行优化。我们将假设 P 是满维的; 否则, 可以消除一个或多个变量并得到一个较小的满维问题。

从可行性到优化首先，让我们展示如何将这样一个优化问题归约为在多面体中找到一个可行点的问题。让 $c^T x$ 与 $c \in \mathbb{R}^n$ 是我们希望在 P 上最小化的目标函数。不失一般性地假设 $c \in \mathbb{Z}^n$. 我们可以检查非空性而不是进行优化

$$P' = P \cap \{x : c^T x \leq d + \frac{1}{2}\}$$

对于 $d \in \mathbb{Z}$, 我们的最优值对应于最小的 d 。由于 $S \subseteq \{0, 1\}^n$, d 必须在 $[-nc_{max}, nc_{max}]$ 范围内取值, 其中 $c_{max} = \max_i c_i$ 。要找到 d , 我们可以使用二分搜索 (并使用椭球算法检查 P' 的非空性)。这将需要 $O(\log(nc_{max})) = O(\log n + \log c_{max})$ 步, 这是多项式时间。

开始椭球体。现在, 我们需要考虑使用椭球体来找到一个可行点在 P' 中或确定 P' 为空。作为起始椭球体, 我们可以使用以向量 $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$ 为中心的球体。

$(E_0) = \{x : \sum_{i=1}^n (x_i - \frac{1}{2})^2 \leq \frac{1}{2}\}$ 和半径 $\frac{1}{\sqrt{2}}\sqrt{n}$ (通过所有 $\{0, 1\}^n$ 向量的) 这个球的体积 $Vol(E_0) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$ 体积 (B_n) , 其中 B_n 是单位球。我们有 $Vol(B_n) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$, 在这里我们甚至可以使用 (非常弱的) 上界 $\pi^{n/2}$ (甚至是 2^n)。这表明 $\log(Vol(E_0)) = O(n \log n)$ 。

终止准则。我们现在将证明如果 P' 非空, 则其体积不会太小。假设 P' 非空, 即 $v_0 \in P' \cap \{0, 1\}^n$ 。我们假设 P 是满维的, 意味着存在 $v_1, v_2, \dots, v_n \in P \cap \{0, 1\}^n$ such that the “simplex” v_0, v_1, \dots, v_n is full-dimensional. The v_i 's may not be in P' 。相反, 对于 $i = 1, \dots, n$ 定义 w_i 为:

$$w_i = \begin{cases} v_i & \text{如果 } c^T v_i \leq d + \frac{1}{2} \\ v_0 + \alpha(v_i - v_0) & \text{否则} \end{cases}$$

其中 $\alpha = \frac{1}{2nc_{max}}$. 这意味着 $w_i \in P'$ 作为

$$c^T w_i = c^T v_0 + \alpha c^T (v_i - v_0) \leq d + \frac{1}{2nc_{max}} nc_{max} = d + \frac{1}{2}.$$

我们有 P' 包含 $C = \text{conv}(\{v_0, w_1, w_2, \dots, w_n\})$ 并且 $Vol(C)$ 是 $\frac{1}{n!}$ 倍的体积由 $w_i - v_0 = \beta_i(v_i - v_0)$ 张成的平行六面体 (其中 $\beta_i \in \{\alpha, 1\}$) 的体积为 $i = 1, \dots, n$. 这个平行六面体的体积等于 β_i (至少为 α^n) 乘以一个具有整数顶点的平行六面体的体积, 至少为 1。因此,

$$Vol(P') \geq Vol(C) = \frac{1}{n!} \left(\frac{1}{2nc_{max}} \right)^n.$$

取对数后, 我们可以看到在发现 P' 为空或者存在可行点之前, 椭球算法的迭代次数最多为 $\log(Vol(E_0)) - \log(Vol(P')) = O(n \log n + n \log c_{max})$ 。这是多项式的。

分离预言机。为了运行椭球算法，我们需要能够判断给定的 $x \in \mathbb{R}^n$ ，是属于 P 还是找到一个违反的不等式。美妙之处在于我们不需要对 P 进行完整和明确的线性不等式描述。我们将看到一些例子，其中我们甚至可以将其应用于指数大小的描述。我们需要的是 P 的分离预言机：给定 $x^* \in \mathbb{R}^n$ ，要么判断 $x^* \in P$ ，要么找到一个对 P 有效的不等式 $a^T x \leq b$ ，使得 $a^T x^* > b$ 。如果这个分离预言机在多项式时间内运行，我们就成功地找到了在优化 $c^T x$ 在 P （或 S ）上的最优值 d 。

寻找最优解。还有一个问题。该算法给出了一个点 x^* 在 P 中的值最多为 $d + \frac{1}{2}$ 其中 d 是最优值。然而，我们对于找到一个点 $x \in P \cap \{0, 1\}^n = S$ 的值恰好为 d 感兴趣。这可以通过从 x^* 开始，并找到 P 中的任意极端点 x 使得 $c^T x \leq c^T x^*$ 来实现。细节被省略。

总结一下，我们得到了由 Grötschel, Lovász 和 Schrijver 在 1979 年证明的以下重要定理。

定理 7.3 令 $S = \{0, 1\}^n$ 且 $P = \text{conv}(S)$ 。假设 P 是满维度的，并且我们给出了一个对于 P 的分离预言机。然后，给定 $c \in \mathbb{Z}^n$ ，可以通过使用椭球算法和多项式次数的操作和调用分离预言机来找到 $\min\{c^T x : x \in S\}$ 。

更准确地说，上述应用中椭球算法的迭代次数为 $O(n^2 \log^2 n + n^2 \log^2 c_{\max})$ ，每次迭代需要调用分离 oracle 并进行多项式次数的操作（矩阵 A 的秩-1 更新等）。下面是两个可以用这种方法解决的组合优化问题的简要描述。

最小成本树问题。 我们已经看到了一种组合算法来解决最小成本树问题，它依赖于通过原始-对偶方法解决线性规划：

$$LP = \min \sum_{a \in A} c_a x_a$$

满足以下条件：

$$(P) \quad \begin{aligned} \sum_{a \in \delta^-(S)} x_a &\geq 1 && \text{对于所有的 } S \subseteq V \setminus \{r\} \\ x_a &\geq 0 && a \in A. \end{aligned}$$

相反，我们可以使用椭球算法直接在多项式时间内解决这个线性规划。事实上，尽管这个线性规划有指数数量的约束（与图的大小相关），但它的分离 oracle 可以很容易地定义。事实上，考虑 x^* 。如果对于某个 $a \in A$ ，有 $x_a^* < 0$ ，只需返回不等式 $x_a \geq 0$ 。否则，对于每个 $t \in V \setminus \{r\}$ ，

考虑在弧 a 上给定容量为 x_a^* 的最小割问题。正如我们所见，这可以通过最大流计算来解决。如果对于某个 $t \in V \setminus \{r\}$ ，最小割的值小于 1，则我们找到了一个被 x^* 违反的不等式。

否则，我们有 $x^* \in P$ 。因此，我们的分离预言机可以通过进行 $|V| - 1$ 次最大流计算来实现，因此是多项式时间的。我们完成了。

最大权匹配问题。对于一般图 $G = (V, E)$ ，Edmonds 已经证明了所有匹配的凸包是由以下给出的：

$$\begin{aligned} \sum_{e \in E(S)} x_e &\leq \frac{|S| - 1}{2} & S \subset V, |S| \text{ 奇数} \\ \sum_{e \in \delta(v)} x_e &\leq 1 & v \in V \\ x_e &\geq 0 & e \in E. \end{aligned} \quad (6)$$

给定 x^* ，我们可以轻松检查它们是否满足不等式。如果违反，我们将展示如何通过一系列最小割计算来检查它们。

假设 $x \geq 0$ 满足 $\sum_{e \in \delta(v)} x_e \leq 1$ 对于每个 $v \in V$ ，我们想要决定是否

$$\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2}$$

对于所有的 $S \subset V$ ， $|S|$ 是奇数，如果不满足，则产生一个违反的集合 S 。我们可以假设 $|V|$ 是偶数（否则，只需添加一个顶点）。让

$$s_v = 1 - \sum_{e \in \delta(v)} x_e,$$

对于所有 $v \in V$ 。观察到 $\sum_{e \in E(S)} x_e \leq \frac{|S| - 1}{2}$ 等价于

$$\sum_{v \in S} s_v + \sum_{e \in \delta(S)} x_e \geq 1.$$

定义一个新图 H ，其顶点集为 $V \cup \{u\}$ （其中 u 是一个新顶点），边集为 $E \cup \{(u, v) : v \in V\}$ 。在这个新图中，如果 $e \in E$ 或者 $e = (u, v)$ ，则边的容量 u_e 为 x_e 。观察到在这个图中，

$$\sum_{v \in S} s_v + \sum_{e \in \delta(S)} x_e \geq 1$$

当且仅当

$$\sum_{e \in \delta_H(S)} u_e \geq 1,$$

其中 $\delta_H(S)$ 是 H 中以 S 为一个端点的边。因此，要解决我们的分离问题，只需要能够在 H 中找到所有由集合 S 定义的割中的最小割，其中 $S \subseteq V$ ， $|S|$ 为奇数。这是之前所见的最小 T -奇割问题的特殊情况。如果最小 T -奇割问题返回一个值大于等于1的最小割，我们就知道所有不等式都满足；否则，最小 T -奇割提供了一个违反的不等式。