第 1 章 概论和基础知识

1.1 引言

运筹优化是 20 世纪兴起发展起来的一门新兴学科，主要是针对现实中的多种应用问题，研究如何实现有效管理和科学决策 [1]-[11]。运筹学的英文为 Operations Research。而中文运筹一词，取自《史记·高祖本纪》“夫运筹帷幄之中，决胜千里外，吾不如子房”，强调运筹优化中的决策需提前规划之意。

运筹优化已经渗透到当今生产生活的方方面面，在水土建筑工程设计、机械结构设计、制造计划管理、资源分配调度、库存管理、物流管理、电子线路设计、集成电路设计、通信网络传输、控制器和自动装置设计、交通运输调度、员工安排、投资管理、公司运营管理、人工智能系统设计、程序设计等难以一一胜数的场景中得到应用。因此运筹学与应用数学、工业工程、计算机科学、人工智能、控制理论与控制工程、电子通讯、经济管理等多种专业密切相关。

研究运筹优化的基础知识包括数学分析、代数、随机过程、离散数学、数据结构和算法等。但实际上，我们在正式学习运筹优化之前，已经了解积累了一些相关知识。中国高考自 1995 年引入数学应用问题，培养中学生的数学应用意思，突出表现在建立最优化模型和掌握相应求解方法。而高等数学教学大纲（电类专业）已经要在了解函数的极值与最大值最小值：理解函数的极值概念，掌握利用导数求函数的极值，会判断函数的增减性、凸性、求曲线的拐点及函数作图（包括求渐近线）的方法，会解决应用题中简单的最大值和最小值问题。我们在本章进一步梳理了相关的预备知识，方便后续教学。

运筹学作为独立学科成熟起来的标志是：

I）用严谨的数学语言重新定义和描述实际问题；

II）抽象出一般化的共性问题，用完整的数学理论分析和证明所考虑问题及算法的性质；

III）提出高效求解算法，进而形成独立理论体系。

因此，我们希望通过系统学习运筹优化知识，完整培养学生的建模，分析和求解三种能力。

在建模方面，我们希望同学们不仅能掌握常用数学规划模型的结构和特点，以便选择合适的模型解决具体问题;还能独立将应用问题用数学规划模型来描述。

如何将现实需求转化为运筹规划问题没有一定之规。美国著名数学教育家 G. Polya 教授在《How to Solve It: A New Aspect of Mathematical Method》一书中提出的如下一般数学问题求解技巧也同样适合运筹规划问题的建模 [12]：

① 求解过程中需要注意的问题：

② 求解什么未知数？已知什么？

③ 条件是什么？条件充不充分？

④ 但凡能画图，一定要画，把条件分解成各个部分，把自己的问题用自己的话重新讲，反复讲。

通常来说，运筹规划建模要点在于：

1. 对现实问题要看透，抓住问题里面的最重要的因素（因问题而异，无一般方法）；

2. 找到最合适的数学语言和它对应（简单而优雅）；

3. 建立模型，要尽量容易解；

4. 尽量套用已有的现成模型，方便使用成熟的求解方法和计算工具；

5. 把求得的结果从数学语言翻译成我们能看懂的语言。

实际运用的时候，需要尽量多了解已有的现成模型，以便套用。

例 1.1. [IMO 1977] 假设一个有限的实数序列中，任意连续七项的和为负，任意连续十一项之和为正。请确定这个序列最多可以含有多少项。

解：如果从运筹优化的角度来看待本题，应该先将约束条件用我们熟悉的形式表达出来。假设这个实数序列表示为$\{a_1, a_2, a_3 ...\}$，如果这个序列有17 项的话，可以得到如下的矩阵

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_7 \\ a_2 & a_3 & \cdots & a_8 \\ \vdots & \vdots & \vdots & \vdots \\ a_{11} & a_{12} & \cdots & a_{17} \end{pmatrix}$$

这个矩阵的所有元素和 S 按行计算的话，有

$$S = \sum_{i=1}^{7} a_i + \sum_{i=2}^{8} a_i + \cdots + \sum_{i=11}^{17} a_i < 0$$

这个矩阵的所有元素和 S 按列计算的话，有

$$S = \sum_{i=1}^{11} a_i + \sum_{i=2}^{12} a_i + \cdots + \sum_{i=7}^{17} a_i > 0$$

显然矛盾。所以这个序列最多只能有16 项。实际上，我们可以找到多个含有16 项的该种序列，譬如5,5,-13,5,5,5,-13,5,5,-13,5,5,5,-13,5,5 。现在的主流优化软件都可以用极短的时间算出一组还有16 项的可行解。

在分析方面，我们希望同学能灵活应用所学知识，准确判断应用问题的分类，正确推导出问题的有解性和解的特性。

在求解方面，我们希望同学们能自行确定最为合理的求解算法。从有效性角度（Effectiveness），判断是否能找到最优解，还是只能找到次优解。从时间空间复杂性（Time/Space Complexity）角度，分析求解算法的收敛时间和停止条件。最后，我们要求将求解方法整理成计算机能够理解的机器语言，以便应用计算机求解实际问题。

数据结构与算法是通向很多世界的大门。运筹学是否一定要学编程是“有争议”的。但实际上如果不会简单的编程，在未来实践中，很有可能遇到模型稍微复杂，无法用现有软件直接实现。如果不想届时束手无策，我们还是希望同学们能掌握足够的编程能力。为此，本书对于一些问题给出了伪代码。

一些数学系的运筹优化教材偏理论分析，有更为严格和完备覆盖各个知识点的数学证明和算法理论。而一些应用类的运筹优化教材注意力放在建模及优化软件的使用上。我们还是希望同学们具备完整“建模、分析和求解”能力。因此本书在突出主要知识点的前提下，给出了尽量完整和自洽的数学证明和算法理论分析，并通过例题引导同学们培养动手能力。同时，我们给出一些重要概念的英文术语，相信对于同学们今后研究会非常有用。

1.2　常见的运筹规划问题分类

一般来说，运筹规划问题包含四个要素：决策变量，目标函数，约束条件和求解方法。我们可以根据运筹规划问题的四个要素进行常见问题的分类。

根据决策变量取值范围，我们可以分类为连续变量规划问题，整数规划问题，混合整数规划问题，图和网络规划问题等。其中连续变量规划问题又包括线性规划问题（Linear Programming Problems）和非线性规划问题（Nonlinear Programming Problems）。非线性规划问题还可以细分为凸优化问题（Convex Programming Problems）和非凸优化问题（Non-Convex Programming Problems），或者可微优化问题和不可微优化问题。而凸优化问题包括二次规划问题（Quadratic Programming Problems），二次约束二次规划问题（Quadratically Constrained Quadratic Programming Problems），二次锥规划问题（Second Order Cone Programming Problems），几何规划问题（Geometric Programming Problems），半正定规划问题（Semi Definite Programming Problems）等。

根据是否有约束条件，我们可以分类为约束规划问题（Constrained Programming Problems）和无约束规划问题（Unconstrained Programming Problems）。根据约束条件是否随机，我们可以分类为：确定性规划问题（Deterministic Programming Problems）和随机规划问题（Stochastic Programming Problems）。

根据目标函数的差异，我们可以分类为单目标规划问题（Single Objective Problems）、多目标规划问题（Multi-Objective Problems）、多层次规划问题（Multi-Level Optimization Problems）等。

因为篇幅有限，且志在介绍大家入门，我们在本书中仅讨论使用最为广泛的线性规划问题、非线性规划（可微优化和凸优化）问题、整数规划问题、动态规划问题、以及图和网络流规划问题。对于网络计划、排队论、存贮论、对策论、决策分析等内容，本书则不涉及。

1.3 数学符号约定

在本书中，仅涉及实数域。其中，矩阵用大写字母 $\mathbf{A},\mathbf{B},\mathbf{C},...$ 表示。矩阵 $\mathbf{A}$ 的转置表示为 $\mathbf{A}^T$。矩阵 $\mathbf{A}$ 的共轭转置表示为 $\mathbf{A}^H$。矩阵 $\mathbf{A}$ 的逆表示为 $\mathbf{A}^{-1}$。矩阵 $\mathbf{A}$ 的行列式表示为 $\det(\mathbf{A})$。矩阵 $\mathbf{A}$ 的迹表示为 $\mathrm{Tr}(\mathbf{A})$。

多维向量（变量，列向量）用粗体小号字母 $\mathbf{a},\mathbf{b},\mathbf{c},...,\mathbf{x},\mathbf{y},\mathbf{z}$ 表示。向量 $\mathbf{a}$ 的转置表示为行向量 $\mathbf{a}^T$。

一维变量用小写字母 $a,b,c,...,x,y,z$ 表示。

集合通常用大写字母表示 $\Gamma,...,\Omega$。$\mathbb{R}^n$ 表示 n 维实数空间（向量空间），$\mathbb{R}^n_+$ 表示所有元素均非负的 n 维实数空间（向量空间），$\mathbb{R}^n_{++}$ 表示所有元素均正的 n 维实数空间（向量空间）。$\mathbb{N}$ 表示自然数空间。

$\mathbf{0}$ 是具有适当维数的所有元素都是 0 的列向量，$\mathbf{1}$ 是具有适当维数的所有元素都是 1 的列向量。$\mathbf{I}$ 通常表示具有适当维数的单位矩阵（对角线元素全为1，其余元素全为0）。

$\|\cdot\|$ 表示多维向量的范数。$|\cdot|$ 表示一维变量的绝对值。

向量 $\mathbf{x}=[x_1,...,x_n]\in\mathbb{R}^n$ 的 0 范数（0-norm，本质上并非严格意义上的范数，因为不满足三角不等式）定义为向量 $\mathbf{x}$ 的所有非零元素的个数。

向量 $\mathbf{x}\in\mathbb{R}^n$ 的 l_1 范数（l_1-norm）定义为

$$\|\mathbf{x}\|_1=\sum_{k=1}^{n}|x_k| \tag{1.1}$$

向量 $\mathbf{x}\in\mathbb{R}^n$ 的 l_2 范数（l_2 -norm）定义为

$$\|\mathbf{x}\|_2=\sqrt{\sum_{k=1}^{n}x_k^2} \tag{1.2}$$

显然，我们有 $\|\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{x}$ 。

向量 $\mathbf{x} \in \mathbb{R}^n$ 的 l_p 范数（l_p -norm，$p \in [1, +\infty)$）定义为

$$\|\mathbf{x}\|_p = \left(\sum_{k=1}^{n} |x_k|^p \right)^{1/p} \tag{1.3}$$

向量 $\mathbf{x} \in \mathbb{R}^n$ 的 l_∞ 范数（l_∞ -norm）定义为

$$\|\mathbf{x}\|_\infty = \max_{k=1,\ldots,n} |x_k| \tag{1.4}$$

1.4　线性代数基础

一个线性方程组（Linear Equation Set）或者线性系统（Linear System）记为

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{cases} \tag{1.5}$$

其中 $x_i, a_{ij}, b_i, c_i \in \mathbb{R}$，$i = 1, \ldots, m$，$j = 1, \ldots, n$。$x_i$ 为待定的决策变量。

我们也可以矩阵向量的形式将该方程组记为

$$\mathbf{Ax} = \mathbf{b} \tag{1.6}$$

其中向量 $\mathbf{x} = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$，$\mathbf{b} = [b_1, \ldots, b_m]^T \in \mathbb{R}^m$，$\mathbf{A} = [a_{ij}]^T \in \mathbb{R}^{m \times n}$。

令矩阵 $\mathbf{A}$ 的各列向量依次表示为 $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$，我们也可以矩阵向量的形式将该方程组记为

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \cdots \\ a_{m1} \end{bmatrix} + \ldots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \cdots \\ a_{mn} \end{bmatrix} = \sum_{j=1}^{p} x_j \boldsymbol{\alpha}_j = \mathbf{b} \tag{1.7}$$

一般来说，这个线性方程组的解有三种可能：1）无解；2）具有唯一解；3）具有无数多个可行解。我们可以通过如下的增广矩阵的秩来判断

$$\mathbf{A_b} = \begin{bmatrix} a_{11} & \ldots & a_{1n} & b_1 \\ a_{21} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots \\ a_{m1} & \ldots & a_{mn} & b_m \end{bmatrix} = [\mathbf{A} \quad \mathbf{b}] \tag{1.8}$$

1）如果 $\mathrm{rank}(\mathbf{A}) < \mathrm{rank}(\mathbf{A_b})$，这个线性方程组无解。

2）如果 $\operatorname{rank}(\mathbf{A})=\operatorname{rank}(\mathbf{A_b})$，这个线性方程组有解。进一步，

2.1）若 $m=n=\operatorname{rank}(\mathbf{A})$，这个线性方程组有唯一解 $\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$。

2.2）若 $m<n$，这个线性方程组有无穷多个解。若 $\mathbf{x}_1\neq\mathbf{x}_2$ 都是该线性方程组的解，对于任意实数 $\lambda\in(0,1)$，$\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$ 亦是该方程组的解。因为 $\mathbf{A}\left[\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right]=\lambda\mathbf{A}\mathbf{x}_1+(1-\lambda)\mathbf{A}\mathbf{x}_2=\lambda\mathbf{b}+(1-\lambda)\mathbf{b}=\mathbf{b}$。

定义 1.1. 考虑 $m<n$ 的线性方程组 $\mathbf{A}\mathbf{x}=\mathbf{b}$。假设 $\mathbf{A_B}$ 是通过从 $\mathbf{A}$ 的 n 列中选择 m 列而形成的方阵。$\mathbf{x}_B\in\mathbb{R}^m$ 是从 $\mathbf{x}$ 中选择的相应变量所组成的列向量，$\mathbf{x}_N\in\mathbb{R}^{n-m}$ 是代表 $\mathbf{x}$ 的其余变量所组成的列向量。如果 $\mathbf{A_B}$ 是非奇异的，任意给定一个 $\mathbf{x}_N$，我们可确定唯一解 $\mathbf{x}=\begin{bmatrix}\mathbf{x}_B\\ \mathbf{x}_N\end{bmatrix}=\begin{bmatrix}\mathbf{A}_B^{-1}\mathbf{b}-\mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N\\ \mathbf{x}_N\end{bmatrix}$。

进一步，我们将 $\mathbf{x}_N=\mathbf{0}$ 时确定的 $\mathbf{x}=\begin{bmatrix}\mathbf{x}_B\\ \mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{A}_B^{-1}\mathbf{b}\\ \mathbf{0}\end{bmatrix}$ 称为 $\mathbf{A}\mathbf{x}=\mathbf{b}$ 的一个基本解（Basic Solution），并将 $\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}$ 对应的这 m 个变量称为一组基变量（Basic Variables），而剩下的 $n-m$ 个变量则称为一组非基变量（Non-Basic Variables）。$\mathbf{x}_B$ 中存在元素值为 0，则称之为一个退化基本解（Degenerate Basic Solution）；否则称之为一个非退化基本解（Non-Degenerate Basic Solution）。

定义 1.2. 我们用集合 $S^{n\times n}$ 表示所有 $n\times n$ 维的对称实矩阵构成集合。

对称实矩阵 $\mathbf{A}\in\mathbb{R}^{n\times n}$ 是半正定矩阵，如果矩阵 $\mathbf{A}$ 的所有特征值都是非负的。我们用集合 $S_+^{n\times n}$ 表示所有 $n\times n$ 维的半正定矩阵构成集合。如果矩阵 $\mathbf{A}\in S_+^{n\times n}$，则对于任意向量 $\mathbf{x}\in\mathbb{R}^n$，满足 $\mathbf{x}^T\mathbf{A}\mathbf{x}\geq 0$。反之，如果对称实矩阵 $\mathbf{A}$ 对于任意向量 $\mathbf{x}\in\mathbb{R}^n$，都满足 $\mathbf{x}^T\mathbf{A}\mathbf{x}\geq 0$，则 $\mathbf{A}\in S_+^{n\times n}$。

对称矩阵 $\mathbf{A}\in\mathbb{R}^{n\times n}$ 是正定矩阵，如果矩阵 $\mathbf{A}$ 的所有特征值都是正的。我们用

集合 $S_{++}^{n\times n}$ 表示所有 $n\times n$ 维的正定矩阵构成集合。如果矩阵 $\mathbf{A}\in S_{++}^{n\times n}$，则对于任意向量 $\mathbf{x}\in\mathbb{R}^n$，满足 $\mathbf{x}^T\mathbf{A}\mathbf{x}>0$。反之，如果对称实矩阵 $\mathbf{A}$ 对于任意向量 $\mathbf{x}\in\mathbb{R}^n$，都满足 $\mathbf{x}^T\mathbf{A}\mathbf{x}>0$，则 $\mathbf{A}\in S_{++}^{n\times n}$。

1.5　凸分析基础

为了更好对于线性规划问题和非线性规划问题进行分析，我们需要掌握一定的凸分析知识。本节中，我们将依次介绍凸集的定义，球、超平面、半平面、多面体、锥等常见的凸集，极点和极线的定义，凸集分隔定理，支撑超平面定理，下有界的凸集支撑超平面必有极点定理，Farkas 引理，凸函数的定义，多元凸函数和一元凸函数之间的关系，光滑凸函数的一阶和二阶条件。

定义 1.3. 凸集（Convex Set）：如果某个集合中任意两点连起来的直线段都属于该集合，则称其为凸集，否则为非凸集。或者等价的，如果对于集合 $\Omega\in\mathbb{R}^n$，任取 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，对于任意实数 $\lambda\in[0,1]$ 存在

$$\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\in\Omega \tag{1.9}$$

则集合 Ω 为凸集。

定理 1.1. 有限个凸集的交集是凸集。

证明：假设我们有 m 个凸集 $\Omega_1,\Omega_2,...,\Omega_n$，$m\in\mathbb{N}$。这 m 个凸集的交集为 $\Omega=\bigcap_{k=1}^{m}\Omega_k$，对于任意两个属于 Ω 的点 $\mathbf{x}_1,\mathbf{x}_2$，根据交集的定义可知满足 $\mathbf{x}_1,\mathbf{x}_2\in\Omega_1,\Omega_2,...,\Omega_m$。由于这些集合都是凸集，我们直接可知，对于任意实数 $\lambda\in[0,1]$ 存在

$$\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\in\Omega_k,\quad k=1,...,m$$

则根据凸集的定义，集合 Ω 显然为凸集。□

定义 1.4. 圆心位于 $\mathbf{x}^*\in\mathbb{R}^n$，半径为 $r\in\mathbb{R}$ 的开球（Open Ball）为

$$B_r(\mathbf{x}^*)=\left\{\mathbf{x}\,|\,\left\|\mathbf{x}-\mathbf{x}^*\right\|_2<r,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.10}$$

圆心位于 $\mathbf{x}^*\in\mathbb{R}^n$，半径为 $r\in\mathbb{R}$ 的闭球（Closed Ball）为

$$\bar{B}_r\left(\mathbf{x}^*\right)=\left\{\mathbf{x}\,|\,\left\|\mathbf{x}-\mathbf{x}^*\right\|_2\le r,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.11}$$

定理 1.2. 任意一个开球和闭球都是凸集。

证明：我们仅证明开球是凸集，闭球证明与之类似。

对于任意两个点 $\mathbf{x}_1,\mathbf{x}_2\in B_r\left(\mathbf{x}^*\right)$ 和任意实数 $\lambda\in[0,1]$，我们可以得到

$$\begin{aligned}&\left\|\left[\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right]-\mathbf{x}^*\right\|_2=\left\|\left[\lambda\left(\mathbf{x}_1-\mathbf{x}^*\right)+(1-\lambda)\left(\mathbf{x}_2-\mathbf{x}^*\right)\right]\right\|_2\\&\le\left\|\lambda\left(\mathbf{x}_1-\mathbf{x}^*\right)\right\|_2+\left\|\left[(1-\lambda)\left(\mathbf{x}_2-\mathbf{x}^*\right)\right]\right\|_2<\lambda r+(1-\lambda)r=r\end{aligned}$$

其中第一个不等号根据范数的三角不等式关系获得，第二个不等号根据开球定义获得。所以，根据凸集的定义，开球 $B_r\left(\mathbf{x}^*\right)$ 是凸集。□

定义 1.5. 超平面（Hyperplane）定义为

$$H=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}=z,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.12}$$

其中 $\mathbf{h}\in\mathbb{R}^n/\{\mathbf{0}\}$ 是 $\mathbb{R}^n$ 中的给定的非零 n 维向量，也被称为超平面 H 的法线方向。$z\in\mathbb{R}$ 为给定实数。

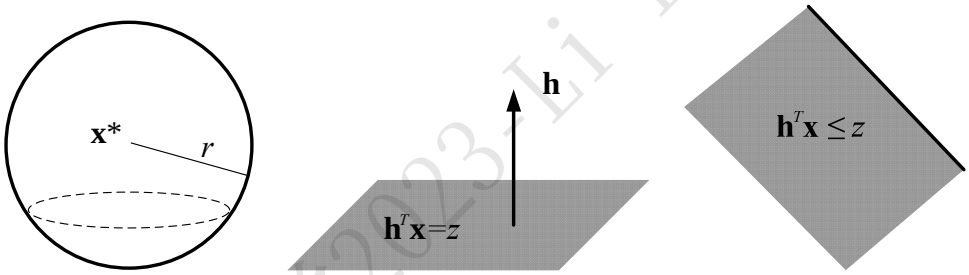


图 1.1 典型凸集示意图：（a）球；（b）超平面；（c）半空间。

定理 1.3. 法线方向 $\mathbf{h}$ 和超平面 H 中所有直线正交。

证明：对于超平面 H 中任意两点 $\mathbf{x}_1,\mathbf{x}_2\in H$，过这两点的直线方向为 $\mathbf{x}_1-\mathbf{x}_2$，必然满足 $\mathbf{h}^T\left(\mathbf{x}_1-\mathbf{x}_2\right)=\mathbf{h}^T\mathbf{x}_1-\mathbf{h}^T\mathbf{x}_2=z-z=0$，也即该直线必然与法线方向 $\mathbf{h}$ 正交。则上述结论显然。□

定义 1.6. 开半空间（Open Half-Space）定义为

$$S=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}<z,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.13}$$

闭半空间（Closed Half-Space）定义为

$$S=\left\{\mathbf{x}\mid\mathbf{h}^T\mathbf{x}\le z,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.14}$$

其中 $\mathbf{h}\in\mathbb{R}^n/\left\{\mathbf{0}\right\}$ 是 $\mathbb{R}^n$ 中的给定的非零 n 维向量，$z\in\mathbb{R}$ 为给定实数。

定理 1.4. 任意一个超平面，半空间都是凸集。

证明：我们仅证明超平面是凸集，半空间证明与之类似。

对于任意两点 $\mathbf{x}_1,\mathbf{x}_2\in H$ 和任意实数 $\lambda\in[0,1]$，我们可以得到

$$\mathbf{h}^T\left[\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right]=\lambda\mathbf{h}^T\mathbf{x}_1+(1-\lambda)\mathbf{h}^T\mathbf{x}_2=\lambda z+(1-\lambda)z=z$$

所以根据凸集的定义，超平面 H 是凸集。□

定义 1.7. 多面体（Polyhedron）定义为同时满足有限个仿射不等式的 n 维向量所构成的集合

$$P=\left\{\mathbf{x}\mid\mathbf{a}_i^T\mathbf{x}\le b_i,i=1,...,m,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.15}$$

其中 $\mathbf{a}_i\in\mathbb{R}^n$ 为给定的 n 维向量，$b_i\in\mathbb{R}$ 为给定实数。很显然，一个多面体可以视为有限个闭半空间的交集。我们也可以将其记为 $P=\left\{\mathbf{x}\mid\mathbf{A}\mathbf{x}\le\mathbf{b},i=1,...,m,\mathbf{x}\in\mathbb{R}^n\right\}$，其中 $\mathbf{A}\in\mathbb{R}^{m\times n}$，$\mathbf{b}\in\mathbb{R}^m$。

有界多面体（Polytope）定义为同时满足有限个仿射不等式的有界 n 维向量所构成的集合

$$P=\left\{\mathbf{x}\mid\mathbf{a}_i^T\mathbf{x}\le b_i,i=1,...,m,l_j\le x_j\le u_j,j=1,...,n,\mathbf{x}\in\mathbb{R}^n\right\} \tag{1.16}$$

其中 $l_i,u_i\in\mathbb{R}$ 为给定实数，表示 x_j 的上下界。

定理 1.5. 任意一个多面体是凸集。

证明：注意到半平面是凸集，结合定理 1 和多面体的定义，结论显然。□

定义 1.8. 假设集合 $\Omega\in\mathbb{R}^n$，如果以点 $\mathbf{x}$ 为中心的任意一个开球都包含集合 Ω 中的一个点和集合 $\mathbb{R}^n-\Omega$ 中的一个点，则点 $\mathbf{x}$ 为集合 Ω 的边界点（Boundary Point）。Ω 的所有边界点构成的集合记为 $\partial\Omega$，称为 Ω 的边界（Boundary）。

如果 $\partial\Omega\subset\Omega$，则 Ω 为闭集（Closed Set）。如果 $\mathbb{R}^n-\Omega$ 为闭集，则 Ω 为开集（Open Set）。

集合 Ω 的闭包（Closure）定义为 $\overline{\Omega}=\Omega\bigcup\partial\Omega$。集合 Ω 的内部（Interior）定义为 $\Omega^o=\Omega-\partial\Omega$。属于集合 Ω 的内部的点称为内点（Interior Point）。Ω 为闭集，当且仅当 $\overline{\Omega}=\Omega$。Ω 为开集，当且仅当 $\Omega=\Omega^o$。

定义 1.9. 点 $\mathbf{x}$ 属于凸集 Ω，如果不存在两个不同的点 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$ 和实数 $\lambda\in[0,1]$，使得点 $\mathbf{x}=\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$，则点 $\mathbf{x}$ 为凸集 Ω 的极点（Extreme Point）。

（a） （b）

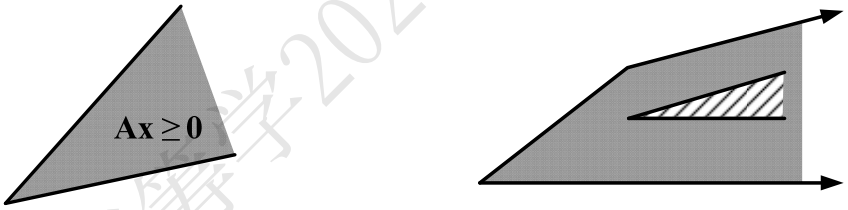
图 1.2 示意图：（a）$\mathbf{x}_1$ 是集合 Ω 的边界点，$\mathbf{x}_2$ 是集合 Ω 的内部点，$\mathbf{x}_3$ 是集合 Ω 之外的点；（b）$\mathbf{x}_1$ 是集合 Ω 的极点，$\mathbf{x}_2$ 和 $\mathbf{x}_3$ 是集合 Ω 的边界点但非极点。

定义 1.10. 如果一个 C 中的任意一个点 $\mathbf{x} \in C$，满足对于任意的 $\lambda > 0$，$\lambda\mathbf{x} \in C$，则该集合 C 称为锥（Cone）。如果集合 C 同时是凸集，则称为凸锥（Convex Cone）。显然，多面体 $P = \{\mathbf{x} \mid \mathbf{Ax} \geq 0, \mathbf{x} \in \mathbb{R}^n\}$ 满足锥的定义，我们称这样的多面体为多面体锥（Polyhedral Cone）。

定义 1.11. 考虑一个非空多面体 $P = \{\mathbf{x} \mid \mathbf{Ax} \geq \mathbf{b}, \mathbf{x} \in \mathbb{R}^n\}$，如果固定某个点 $\mathbf{y} \in P$，我们将点 $\mathbf{y}$ 处的回收锥（Recession Cone）定义为所有方向 $\mathbf{d} \in \mathbb{R}^n$ 的集合，沿着这些方向 $\mathbf{d}$ 我们可以无限远离点 $\mathbf{y}$，而不会离开集合 P。换句话说，回收锥表示为集合

$$R = \{\mathbf{d} \mid \mathbf{A}[\mathbf{y} + \lambda\mathbf{d}] \geq \mathbf{b}, \text{for all } \lambda \geq 0, \mathbf{d} \in \mathbb{R}^n\} \tag{1.17}$$

实际上，很容易证明回收锥的定义等价为 $R = \{\mathbf{d} \mid \mathbf{Ad} \geq \mathbf{0}, \mathbf{d} \in \mathbb{R}^n\}$。因此，回收锥实际上是与 $\mathbf{y}$ 无关的一个多面体锥。



（a） （b）

图 1.3 典型凸集示意图：（a）多面体锥；（b）多面体内的回收锥（斜纹阴影所示）和极线（箭头射线所示）。

定义 1.12. 回收锥 $R = \{\mathbf{d} \mid \mathbf{Ad} \geq \mathbf{0}, \mathbf{d} \in \mathbb{R}^n\}$ 的任意一个非零成员为多面体 $P = \{\mathbf{x} \mid \mathbf{a}_i^T\mathbf{x} \geq b_i, i = 1, ..., m, \mathbf{x} \in \mathbb{R}^n\}$ 的射线（Ray）。而如果该非零成员使得上述不等式组中恰有 $n-1$ 个线性独立的约束条件 $\mathbf{a}_i^T\mathbf{x} \leq b_i$ 起作用（Active），也即 $n-1$

个线性独立的不等式约束条件恰好取等号满足，则该非零成员为多面体 $P=\left\{\mathbf{x}\,|\,\mathbf{a}_i^T\mathbf{x}\geq b_i, i=1,...,m, \mathbf{x}\in\mathbb{R}^n\right\}$ 的极线（Extreme Ray）。

定理 1.6. [凸集分隔定理] 如果某个点 $\mathbf{y}\in\mathbb{R}^n$ $\mathbf{y}\in\mathbb{R}^n$ 位于闭凸集 $\Omega\subset\mathbb{R}^n$ 之外，则必然存在一个超平面 $H=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}=z, \mathbf{x}\in\mathbb{R}^n\right\}$ 通过点 $\mathbf{y}$，且满足 $\Omega\subset\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}>z\right\}$；$\mathbf{h}\in\mathbb{R}^n/\left\{\mathbf{0}\right\}$ 是 $\mathbb{R}^n$ 中的某个非零 n 维向量，$z\in\mathbb{R}$ 为某个实数。

更进一步，同样必然存在另外一个超平面 $H=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}=z, \mathbf{x}\in\mathbb{R}^n\right\}$ 同时满足 $\Omega\subset\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}>z\right\}$ 和 $\mathbf{h}^T\mathbf{y}<z$。我们称之为存在超平面分隔点 $\mathbf{y}$ 和闭凸集 Ω。

证明：我们可以定义点 $\mathbf{y}$ 到集合 Ω 的距离为 $\inf_{\mathbf{x}\in\Omega}\left\|\mathbf{x}-\mathbf{y}\right\|_2>0$。由于 Ω 是闭凸集，因此必然存在一个点 $\mathbf{x}_0$ 使得连续函数 $\left\|\mathbf{x}-\mathbf{y}\right\|_2$ 取得最小值，参见图 1.4。

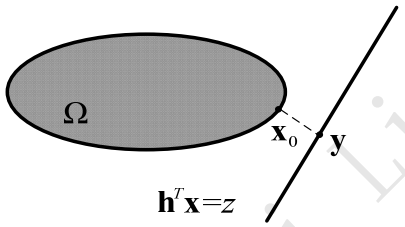


图 1.4　凸集分隔定理证明示意图。

首先，我们使用反证法证明这个点 $\mathbf{x}_0$ 一定是个边界点。如果 $\mathbf{x}_0\in\Omega^o$，则必然存在 $2\varepsilon>0$，使得以 $\mathbf{x}_0$ 为球心，2ε 为半径的开球位于 Ω 内部，也即 $B_{2\varepsilon}\left(\mathbf{x}_0\right)\subset\Omega$。我们可以由此确定一个新点

$$\mathbf{x}_1=\mathbf{x}_0+\varepsilon\frac{\mathbf{y}-\mathbf{x}_0}{\left\|\mathbf{x}_0-\mathbf{y}\right\|_2}\in B_{2\varepsilon}\left(\mathbf{x}_0\right)\subset\Omega$$

不难发现，该点 $\mathbf{x}_1$ 到点 $\mathbf{y}$ 的距离比点 $\mathbf{x}_0$ 到点 $\mathbf{y}$ 的距离更小

$$\left\|\mathbf{x}_1-\mathbf{y}\right\|_2=\left\|\mathbf{x}_0+\varepsilon\frac{\mathbf{y}-\mathbf{x}_0}{\left\|\mathbf{y}-\mathbf{x}_0\right\|_2}-\mathbf{y}\right\|=\left(1-\frac{\varepsilon}{\left\|\mathbf{y}-\mathbf{x}_0\right\|_2}\right)\left\|\mathbf{x}_0-\mathbf{y}\right\|_2<\left\|\mathbf{x}_0-\mathbf{y}\right\|_2$$

这和我们假设的点 $\mathbf{x}_0$ 到点 $\mathbf{y}$ 的距离最小矛盾，因此点 $\mathbf{x}_0$ 一定是个边界点。

接着，我们定义 $\mathbf{h}=\mathbf{x}_0-\mathbf{y}$ 和 $z=\mathbf{h}^T\mathbf{y}$，然后证明 $\Omega\subset\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}>z\right\}$。

对于所有的 $\mathbf{x}\in\Omega$ 和任意的实数 $\lambda\in(0,1)$，由于 Ω 是凸集，我们有

$$\bar{\mathbf{x}}=\mathbf{x}_0+\lambda(\mathbf{x}-\mathbf{x}_0)\in\Omega，且\left\|\bar{\mathbf{x}}-\mathbf{y}\right\|_2\geq\left\|\mathbf{x}_0-\mathbf{y}\right\|_2$$

也即有

$$\left\|\bar{\mathbf{x}}-\mathbf{y}\right\|_2^2\geq\left\|\mathbf{x}_0-\mathbf{y}\right\|_2^2\Rightarrow(\bar{\mathbf{x}}-\mathbf{y})^T(\bar{\mathbf{x}}-\mathbf{y})\geq(\mathbf{x}_0-\mathbf{y})^T(\mathbf{x}_0-\mathbf{y})$$

$$\Rightarrow 2\lambda(\mathbf{x}_0-\mathbf{y})^T(\mathbf{x}-\mathbf{x}_0)+\lambda^2(\mathbf{x}-\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)\geq 0$$

注意到 $\lambda>0$，可以从上式两端同时除以 λ 保持不等式不变号成立，再带入 $\mathbf{h}=\mathbf{x}_0-\mathbf{y}$，整理可得

$$\mathbf{h}^T(\mathbf{x}-\mathbf{x}_0)+\lambda(\mathbf{x}-\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)\geq 0$$

接下来，我们使用反证法证明 $\mathbf{h}^T(\mathbf{x}-\mathbf{x}_0)\geq 0$。假设 $\mathbf{h}^T(\mathbf{x}-\mathbf{x}_0)<0$，我们总能够找到一个足够小的 $\lambda\to 0$ $\lambda\to 0$ 使得 $\lambda(\mathbf{x}-\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)\to 0$，从而使得 $\mathbf{h}^T(\mathbf{x}-\mathbf{x}_0)+\lambda(\mathbf{x}-\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)<0$，但这和我们上面推出的不等式矛盾。

因此，我们可以得到

$$\mathbf{h}^T\mathbf{x}=\mathbf{h}^T(\mathbf{x}-\mathbf{x}_0)+\mathbf{h}^T(\mathbf{x}_0-\mathbf{y})+\mathbf{h}^T\mathbf{y}\geq\mathbf{h}^T\mathbf{h}+z>z$$

因此，$\Omega\subset\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}>z\right\}$。

更进一步，如果我们定义 $\mathbf{h}=\mathbf{x}_0-\mathbf{y}$ 和 $z_1=\mathbf{h}^T\left(\frac{1}{2}\mathbf{y}+\frac{1}{2}\mathbf{x}_0\right)$，则可以发现超平面 $\mathbf{h}^T y<z_1$ 分隔点 $\mathbf{y}$ 和闭凸集 Ω，同时满足 $\Omega\subset\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}>z_1\right\}$ 和 $\mathbf{h}^T\mathbf{x}=z_1$。□

定义 1.13. 考虑闭凸集 $\Omega\subset\mathbb{R}^n$ 的边界点 $\mathbf{x}$，如果存在一个超平面 $H=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}=z,\mathbf{x}\in\mathbb{R}^n\right\}$ 通过点 $\mathbf{x}$，且满足 $\Omega\subseteq\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}\geq z\right\}$；$\mathbf{h}\in\mathbb{R}^n/\{\mathbf{0}\}$ 是 $\mathbb{R}^n$ 中的某个非零 n 维向量，$z\in\mathbb{R}$ 为某个实数。我们称这样的超平面为闭凸集 Ω 的支撑超平面。

定理 1.7. [支撑超平面定理] 对于闭凸集 $\Omega\subset\mathbb{R}^n$ 的任意一个边界点 $\mathbf{y}$，必然存在一个支撑超平面 $H=\left\{\mathbf{x}\,|\,\mathbf{h}^T\mathbf{x}=z,\mathbf{x}\in\mathbb{R}^n\right\}$；$\mathbf{h}\in\mathbb{R}^n/\{\mathbf{0}\}$ 是 $\mathbb{R}^n$ 中的某个非零 n 维向量，$z\in\mathbb{R}$ 为某个实数。

证明：由于 $\mathbf{y}$ 为闭凸集 Ω 的边界点，则对于任意 $n\in\mathbb{N}$，以 $\mathbf{y}$ 为球心，$1/n$ 为半径的球中必然含有一个点 $\mathbf{y}_n$ 不属于集合 Ω 的闭包，也即

$$\mathbf{y}_n\notin\bar{\Omega}，但\lim_{n\to+\infty}\mathbf{y}_n=\mathbf{y}$$

根据定理 1.6，可得到一系列分隔超平面 $\Omega \subseteq H = \left\{ \mathbf{x} \,|\, \mathbf{h}_n^T \mathbf{x} > \mathbf{h}_n^T \mathbf{y}_n, \mathbf{x} \in \mathbb{R}^n \right\}$，$n \in \mathbb{N}$，将闭凸集 Ω 和点 $\mathbf{y}_n$ 分隔开来。

我们可以进一步要求 $\left\| \mathbf{h}_n \right\|_2 = 1$，则 $\left\{ \mathbf{h}_n \right\}$ 是紧空间中的有界点列，因此其中必有一个子列 $\left\{ \mathbf{h}_{n_k}, k \in \mathbb{N} \right\}$ 收敛于 $\mathbf{h} = \lim\limits_{k \to +\infty} \mathbf{h}_{n_k}$。

我们定义 $z = \mathbf{h}^T \mathbf{y}$，则对于任意点 $\mathbf{x} \in \Omega$，我们有

$$\mathbf{h}^T \mathbf{x} = \lim_{n \to +\infty} \mathbf{h}_{n_k}^T \mathbf{x} \geq \lim_{n \to +\infty} \mathbf{h}_{n_k}^T \mathbf{y}_{n_k} = \mathbf{h}^T \mathbf{y} = z$$

因此，超平面 $H = \left\{ \mathbf{x} \,|\, \mathbf{h}^T \mathbf{x} = z, \mathbf{x} \in \mathbb{R}^n \right\}$ 是闭凸集 Ω 的支撑超平面。□

定义 1.14. 一个集合 $\Omega \subset \mathbb{R}^n$ 被称为下有界（Bounded from Below），如果对于任意一个点 $\mathbf{x} = \left[x_1, ..., x_n \right] \in \Omega$，存在

$$\inf \left\{ x_i \right\} > -\infty, \quad i = 1, ..., n \tag{1.18}$$

定理 1.8. 如果闭凸集 $\Omega \subset \mathbb{R}^n$ 下有界，则集合 Ω 的任意一个支撑超平面上至少含有一个集合 Ω 的极点。

证明：假设点 $\mathbf{x}_0$ 是闭凸集 Ω 的一个边界点，而 $H = \left\{ \mathbf{x} \,|\, \mathbf{h}^T \mathbf{x} = z, \mathbf{x} \in \mathbb{R}^n \right\}$ 是通过点 $\mathbf{x}_0$ 的相对于集合 Ω 的一个支撑超平面，满足 $\Omega \subseteq \left\{ \mathbf{x} \,|\, \mathbf{h}^T \mathbf{x} \geq z \right\}$。我们定义该支撑超平面和集合 Ω 的交集为 $\Delta = \Omega \bigcap H$。显然，由于 $\mathbf{x}_0 \in \Delta$，Δ 不是空集。

接下来，我们先证明任意一个集合 Δ 的极点同时也是集合 Ω 的极点。实际上，我们转而证明该命题的逆否命题，也即如果一个点 $\mathbf{t}$ 不是集合 Ω 的极点，则该点 $\mathbf{t}$ 也一定不是集合 Δ 的极点。具体而言，此时有两种情况。

第一种情况，如果 $\mathbf{t} \notin \Omega$，则结论显然。

第二种情况，$\mathbf{t} \in \Omega$。如果 $\mathbf{t}$ 不是集合 Ω 的极点，则存在两个不同的点 $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ 和某个实数 $\lambda \in (0,1)$，满足 $\mathbf{t} = \lambda \mathbf{x}_1 + (1-\lambda) \mathbf{x}_2$。而根据支撑超平面的性质，我们有

$$\mathbf{h}^T \mathbf{t} = \lambda \mathbf{h}^T \mathbf{x}_1 + (1-\lambda) \mathbf{h}^T \mathbf{x}_2 = z, \text{ 同时 } \mathbf{h}^T \mathbf{x}_1 \geq z, \ \mathbf{h}^T \mathbf{x}_2 \geq z$$

上述关系同时成立，则必然有 $\mathbf{h}^T \mathbf{x}_1 = \mathbf{h}^T \mathbf{x}_2 = z$，所以点 $\mathbf{x}_1, \mathbf{x}_2 \in \Delta$，故而 $\mathbf{t}$ 在集合 Δ 内不同两点 $\mathbf{x}_1, \mathbf{x}_2$ 的连线段上，因此 $\mathbf{t}$ 不是集合 Δ 的极点。该命题的逆否命题得证。根据原命题和逆否命题的关系，原命题亦得证。

然后，我们证明集合 Δ 至少存在一个极点。显然，如果集合 Δ 仅含一个元素 $\mathbf{x}_0$，则 $\mathbf{x}_0$ 必然为一个极点，我们已经找到一个满足要求的极点，可以停止。如果集合 Δ 含有多个元素，我们使用如下有限递降法进行证明。

我们构造新的集合 $\Delta^1 = \arg\min \left\{ x_1 \,|\, \mathbf{x} \in \Omega \right\}$。因为 $\Delta \subset \Omega$ 且集合 Ω 下有界，闭集合 Δ^1 实际上定义为集合 Ω 中，所有 x_1 取得最小值的点 $\mathbf{x}$ 所构成的集合。

如果Δ^1只含一个元素，则我们已经找到一个满足要求的极点，可以停止。否则，我们进一步构造新的集合$\Delta^j = \arg\min\left\{x_j \mid \mathbf{x} \in \Delta^{j-1}\right\}$，$j = 2,...,n$。如此进行下去，这一构造过程必然在最多$n$次之后停止；因为$\Delta \subset \mathbb{R}^n$，构造过程执行$\Delta \subset \mathbb{R}^n$ n次之后将得到唯一的一个点。假设我们在k次之后停止，得到单点集合$\Delta^k = \left\{\mathbf{x}^k\right\}$，此时$\Delta^k \subseteq \Delta^{k-1} \subseteq ... \subseteq \Delta^1 \subseteq \Delta$。

最后，我们使用反证法证明点$\mathbf{x}^k = \left[x_1^k,...,x_n^k\right]$必为集合$\Delta$的一个极点。假设存在两个不同的点$\mathbf{x}^a = \left[x_1^a,...,x_n^a\right], \mathbf{x}^b = \left[x_1^b,...,x_n^b\right] \in \Delta$和某个实数$\lambda \in (0,1)$，满足$\mathbf{x}^k = \lambda\mathbf{x}^a + (1-\lambda)\mathbf{x}^b$。则对于任意$i \le k$，存在$x_i^k = \lambda x_i^a + (1-\lambda)x_i^b$。根据集合$\Delta^1$的构造方法，$x_1^k$为$x_1$能取到的最小值，则必然有$x_1^k = x_1^a = x_1^b$，也即$\mathbf{x}^a, \mathbf{x}^b \in \Delta^1$。类似的，我们可以依次得到$x_j^k = x_j^a = x_j^b$，也即$\mathbf{x}^a, \mathbf{x}^b \in \Delta^j$，$j = 2,...,k$，并最终得出$\mathbf{x}^a = \mathbf{x}^b = \mathbf{x}^k \in \Delta^k$，引出矛盾。因此，点$\mathbf{x}^k$必为集合$\Delta$的一个极点。□

定理 1.9. [Farkas 引理] 给定矩阵$\mathbf{A} \in \mathbb{R}^{m\times n}$，向量$\mathbf{b} \in \mathbb{R}^m$，下述两个线性系统有且仅有一个存在可行解 [13]：

线性系统 I：对于$\mathbf{x} \in \mathbb{R}^n$，$\mathbf{Ax} = \mathbf{b}$，$\mathbf{x} \ge \mathbf{0}$ （1.19）

线性系统 II：对于$\mathbf{y} \in \mathbb{R}^m$，$\mathbf{A}^T\mathbf{y} \ge \mathbf{0}$，$\mathbf{b}^T\mathbf{y} < 0$ （1.20）

证明：根据多面体的定义，上述两个线性系统的可行解构成的集合Ω_1, Ω_2必然是凸集。为了应用凸集分隔定理，我们还需要先证明集合Ω_1是闭集。这个证明有两种思路。其一是引入锥包（Conic Hull）的概念，接着证明锥包是闭集，最后证明Ω_1为锥包。其二是将$\mathbf{A} \in \mathbb{R}^{m\times n}$看成是$\mathbb{R}^n$到$\mathbf{A}$的列空间$\mathrm{span}(\mathbf{A})$的一个映射，接着分析这个映射的性质，考察闭集$\mathbb{R}^n$在映射下，原像和像之间的关系证明$\Omega_1$为锥包。因为这些证明较为繁琐且和本书主线相距较远，留给大家作为练习。

接下来，我们应用凸集分隔定理证明 Farkas 引理。

首先我们使用反证法证明上述两个线性系统不可能同时有可行解。如果集合Ω_1, Ω_2同时非空，则存在可行解$\mathbf{y} \in \mathbb{R}^m$满足$\mathbf{b}^T\mathbf{y} < 0$，这表明$\mathbf{b}^T \ne \mathbf{0}$。而$\mathbf{b}^T \ne \mathbf{0}$，则如果有可行解$\mathbf{x} \in \mathbb{R}^n$满足$\mathbf{Ax} = \mathbf{b}$，则可行解$\mathbf{x} \ne \mathbf{0}$。考虑到$\mathbf{x} \ge \mathbf{0}$，$\mathbf{A}^T\mathbf{y} \ge \mathbf{0}$，我们得到

$$\mathbf{b}^T\mathbf{y} = (\mathbf{Ax})^T\mathbf{y} = \mathbf{x}^T(\mathbf{Ay}) \ge 0$$

这与$\mathbf{b}^T\mathbf{y}<0$的假设矛盾，因此集合Ω_1,Ω_2不能同时非空。

最后，我们证明上述两个线性系统必然有一个有解。具体而言，此时有两种情况。

第一种情况，线性系统 I 有解，则结论显然。

第二种情况，线性系统 I 无解，这说明点$\mathbf{b}\notin\Omega=\{\mathbf{Ax},\mathbf{x}\geq\mathbf{0}\}$，而我们前面证明了$\Omega=\{\mathbf{Ax},\mathbf{x}\geq\mathbf{0}\}$是闭凸集，则根据凸集分隔定理，存在一个超平面$H=\{\mathbf{x}\,|\,\mathbf{y}^T\mathbf{s}=z,\mathbf{s}\in\mathbb{R}^m\}$分隔点$\mathbf{b}$和闭凸集$\Omega$。也即$\mathbf{y}^T\mathbf{b}<z$，同时任取一点$\mathbf{s}\in\Omega$，满足$\mathbf{y}^T\mathbf{s}\geq z$。

考虑到$\mathbf{0}\in\Omega$，必然有$z<0$，因此$\mathbf{y}^T\mathbf{b}<z<0$。

另一方面，对于任意一点$\mathbf{x}\geq\mathbf{0}$，满足$\mathbf{y}^T\mathbf{Ax}>0$，则一定有$\mathbf{y}^T\mathbf{A}>\mathbf{0}$。我们使用反证法证明此命题。若向量$\mathbf{y}^T\mathbf{A}$的第$i$个元素是负值，$i=1,...,m$，我们可以选择除了对应的第$i$个元素是正值，其余元素全为 0 的某个特定的$\hat{\mathbf{x}}$使得$\mathbf{y}^T\mathbf{A}\hat{\mathbf{x}}<0$，产生矛盾。

综上所示，Farkas 引理成立。□

实际上，根据公式（1.7），我们可以$\{\mathbf{Ax}\,|\,\mathbf{x}\geq\mathbf{0}\}$视为矩阵$\mathbf{A}$的列向量的所有非负加权组合所构成的点集。不难证明这个集合实际是一个凸锥。如图 1.5 所示，如果向量$\mathbf{b}$落在该凸锥内，线性系统 I 一定有解。反之，如果向量$\mathbf{b}$未落在该凸锥内，线性系统 I 一定无解；并且此时，根据凸集分隔定理，一定能够找到一个过原点的超平面$\mathbf{y}^T\mathbf{x}=0$，使得向量$\mathbf{b}$在超平面$\mathbf{y}^T\mathbf{x}=0$的一边，凸锥$\{\mathbf{Ax}\,|\,\mathbf{x}\geq\mathbf{0}\}$在超平面$\mathbf{y}^T\mathbf{x}=0$的另外一边。因此，线性系统 II 一定有解。

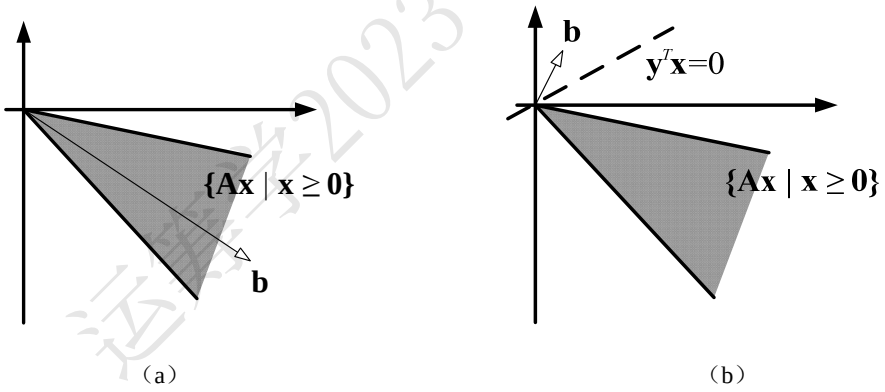


图 1.5　Farkas 引理示意图：（a）向量$\mathbf{b}$落在凸锥$\{\mathbf{Ax},\mathbf{x}\geq\mathbf{0}\}$内，线性系统 I 一定有解；（b）向量$\mathbf{b}$未落在凸锥$\{\mathbf{Ax},\mathbf{x}\geq\mathbf{0}\}$内，线性系统 I 一定无解。

定义 1.15. 假设集合$S\subseteq\mathbb{R}^n$，我们将如下集合称为S的仿射包（Affine Hull）：

$\left\{\mathbf{x}\,|\,\mathbf{x}=\sum_{i=1}^{k}\theta_i\mathbf{x}_i,\sum_{i=1}^{k}\theta_i=1,\mathbf{x}_i\in S,i=1,...,k\right\}$。

定义 1.16. $f:\Omega\to\mathbb{R}$ 是定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上函数，如果对于任意的 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，和任意实数 $\lambda\in[0,1]$ 存在

$$f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)\le\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right) \tag{1.21}$$

则函数 f 为凸集 Ω 上的凸函数（Convex Function）。

如果对于任意的 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，和任意实数 $\lambda\in(0,1)$ 存在

$$f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)<\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right) \tag{1.22}$$

则函数 f 为凸集 Ω 上的严格凸函数（Strictly Convex Function）。

如果对于任意的 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，和任意实数 $\lambda\in[0,1]$ 存在

$$f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)\ge\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right) \tag{1.23}$$

则函数 f 为凸集 Ω 上的凹函数（Concave Function）。

如果对于任意的 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，和任意实数 $\lambda\in(0,1)$ 存在

$$f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)>\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right) \tag{1.24}$$

则函数 f 为凸集 Ω 上的严格凹函数（Strictly Concave Function）。

从几何角度来看，如图 1.6 所示，凸函数曲线上任意两个点连线段上的点位于凸函数曲线对应点的上方。而凹函数正好与之相反。

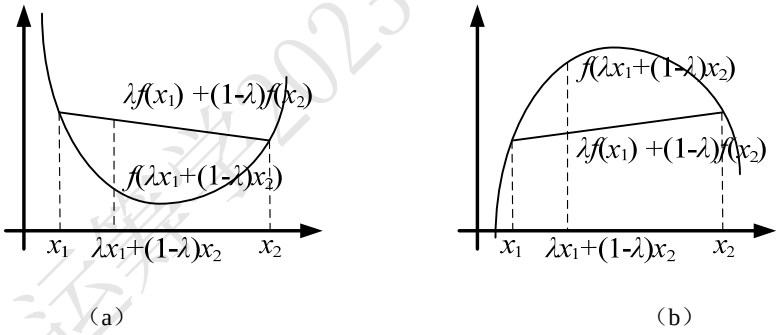


图 1.6 （a）凸函数示意图；（b）凹函数示意图。

定理 1.10. [Jensen 不等式（Jensen's Inequality）] 若 $f:\Omega\to\mathbb{R}$ 是定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上凸函数，对于任意的 $\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_m\in\Omega$，任意实数 $\lambda_i\in[0,1]$ 且满足 $\sum_{i=1}^{m}\lambda_i=1$，$m\in\mathbb{N}$，则成立

$$f\left(\lambda_1\mathbf{x}_1+\lambda_2\mathbf{x}_2+...+\lambda_m\mathbf{x}_m\right)\le\lambda_1 f\left(\mathbf{x}_1\right)+\lambda_2 f\left(\mathbf{x}_2\right)+...+\lambda_m f\left(\mathbf{x}_m\right)$$

（1.25）

反之，如果成立 Jensen 不等式，$f:\Omega\to\mathbb{R}$ 是定义凸集 Ω 上的凸函数。

证明：我们使用数学归纳法来证明正向论断。显然，$m=2$ 时，根据凸函数的定义，Jensen 不等式成立。假设 $m=k$ 时，Jensen 不等式成立，则当 $m=k+1$ 时，我们利用凸函数的性质，可以得到

$$\begin{aligned}
&f\left(\lambda_1\mathbf{x}_1+...+\lambda_k\mathbf{x}_k+\lambda_{k+1}\mathbf{x}_{k+1}\right)\\
&\le\left(1-\lambda_{k+1}\right)f\left(\frac{\lambda_1\mathbf{x}_1+...+\lambda_k\mathbf{x}_k}{1-\lambda_{k+1}}\right)+\lambda_{k+1}f\left(\mathbf{x}_{k+1}\right)\\
&\le\left(1-\lambda_{k+1}\right)\left[\frac{\lambda_1 f\left(\mathbf{x}_1\right)}{1-\lambda_{k+1}}+...+\frac{\lambda_k f\left(\mathbf{x}_k\right)}{1-\lambda_{k+1}}\right]+\lambda_{k+1}f\left(\mathbf{x}_{k+1}\right)\\
&=\lambda_1 f\left(\mathbf{x}_1\right)+...+\lambda_k f\left(\mathbf{x}_k\right)+\lambda_{k+1}f\left(\mathbf{x}_{k+1}\right)
\end{aligned}$$

故当 $m=k+1$ 时，Jensen 不等式也成立。根据数学归纳法，正向命题成立。

注意到 $m=2$ 时，Jensen 不等式即为凸函数定义，因此反向命题成立。□

类似的，我们也可以定义关于凹函数的 Jensen 不等式，此处从略。

Jensen 不等式的一个典型应用是通过构造特定凸函数来证明多种经典不等式。

例 1.2. 试证明下述不等式

[算术平均-几何平均不等式 Arithmetic Mean-Geometric Mean Inequality] 假设 $a_i\in\mathbb{R}_{++}$，$i=1,...,m$，则

$$\left(\prod_{i=1}^{m}a_i\right)^{\frac{1}{m}}\le\frac{1}{m}\sum_{i=1}^{m}a_i$$

[Hölder 不等式 Hölder's Inequality] 假设 $a_i,b_i\in\mathbb{R}_+$，$i=1,...,m$，$p>1$，且 $\frac{1}{p}+\frac{1}{q}=1$，则

$$\sum_{i=1}^{m}a_i b_i\le\left(\sum_{i=1}^{m}|a_i|^p\right)^{\frac{1}{p}}\left(\sum_{i=1}^{m}|b_i|^q\right)^{\frac{1}{q}}$$

[Cauchy-Schwarz 不等式 Cauchy-Schwarz Inequality] 假设 $a_i,b_i\in\mathbb{R}$，$i=1,...,m$，则

$$\left(\sum_{i=1}^{m}a_i^2\right)\left(\sum_{i=1}^{m}b_i^2\right)\ge\left(\sum_{i=1}^{m}a_i b_i\right)^2$$

解：对于算术平均-几何平均不等式，考虑凹函数 $f\left(x\right)=\ln x:\mathbb{R}_{++}\to\mathbb{R}$。

如果我们将 $a_i\in\mathbb{R}_{++}$ 视为该函数定义域内若干点，则 $\lambda_i=\dfrac{1}{m}\in\left[0,1\right]$，则直接应用 Jensen 不等式，有

$$\ln\sqrt[m]{\prod_{i=1}^{m}a_i}=\frac{1}{m}\sum_{i=1}^{m}\ln a_i=\frac{1}{m}\sum_{i=1}^{m}f(a_i)\le f\left(\frac{1}{m}\sum_{k=1}^{m}a_i\right)=\ln\left(\frac{1}{m}\sum_{k=1}^{m}a_i\right)$$

由于该函数为单调增函数，则结论显然。

对于 Hölder 不等式，考虑凸函数 $f\left(x\right)=x^p:\mathbb{R}_{++}\to\mathbb{R}$，$p>1$。如果我们将 $x_i\in\mathbb{R}_{++}$ 视为该函数定义域内若干点，则对于 $\lambda_i\in\left[0,1\right]$，直接应用 Jensen 不等式，有

$$\left(\sum_{i=1}^{m}\lambda_i x_i\right)^p\le\sum_{i=1}^{m}\lambda_i x_i^p$$

我们可以将 $a_i\in\mathbb{R}_{++}$ 表示成 $a_i=\lambda_i^{1/p}x_i$，并定义 $c_i=\lambda_i^{1/p}$，则有

$$\sum_{i=1}^{m}a_ic_i\le\left(\sum_{i=1}^{m}a_i^p\right)^{\frac{1}{p}},\quad\sum_{i=1}^{m}c_i^q=1$$

进一步我们可以将 $b_i\in\mathbb{R}_{++}$ 表示成

$$b_i=\frac{c_i}{\left(\sum_{i=1}^{m}|b_i|^q\right)^{\frac{1}{q}}}$$

则结论显然。

当 $p=q=2$ 时，我们可以直接从 Hölder 不等式得到 Cauchy-Schwarz 不等式。

定理 1.11. [一元凸函数和多元凸函数的关系] 假设 $f:\Omega\to\mathbb{R}$ 是定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的函数；对于任意点 $\mathbf{x}\in\Omega$ 和方向 $\mathbf{d}\in\mathbb{R}^n$，我们定义一族一元函数 $\varphi\left(t\,|\,\mathbf{x},\mathbf{d}\right):\mathbb{R}\to\mathbb{R}$ 为

$$\varphi\left(t\,|\,\mathbf{x},\mathbf{d}\right)=f\left(\mathbf{x}+t\mathbf{d}\right),\quad\forall\mathbf{x}+t\mathbf{d}\in\Omega\tag{1.26}$$

则函数 f 为凸集 Ω 上的凸函数的充分必要条件是，这一族函数 φ 均为凸函数。

证明：我们先证明必要性。对于给定的点 $\mathbf{x}\in\Omega$ 和方向 $\mathbf{d}\in\mathbb{R}^n$，如果实数 t_1,t_2 满足 $\mathbf{x}+t_1\mathbf{d},\mathbf{x}+t_2\mathbf{d}\in\Omega$，则根据凸集的定义，对于任意的实数 $\lambda\in\left[0,1\right]$，满足 $\lambda\left(\mathbf{x}+t_1\mathbf{d}\right)+\left(1-\lambda\right)\left(\mathbf{x}+t_2\mathbf{d}\right)\in\Omega$。进一步根据 f 是凸函数的性质，有

$$\varphi\left(\lambda t_1+\left(1-\lambda\right)t_2\,|\,\mathbf{x},\mathbf{d}\right)=f\left(\lambda\left(\mathbf{x}+t_1\mathbf{d}\right)+\left(1-\lambda\right)\left(\mathbf{x}+t_2\mathbf{d}\right)\right)$$

$$\le\lambda f\left(\mathbf{x}+t_1\mathbf{d}\right)+\left(1-\lambda\right)f\left(\mathbf{x}+t_2\mathbf{d}\right)=\lambda\varphi\left(t_1\,|\,\mathbf{x},\mathbf{d}\right)+\left(1-\lambda\right)\varphi\left(t_2\,|\,\mathbf{x},\mathbf{d}\right)$$

显然根据定义，对于任意点 $\mathbf{x}\in\Omega$ 和方向 $\mathbf{d}\in\mathbb{R}^n$，一族函数 φ 均为凸函数。

我们接着证明充分性。给定 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，实数 $\lambda\in[0,1]$，考虑函数 φ 是凸函数的性质，我们可以得到

$$\begin{aligned}
&f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)=f\left(\mathbf{x}_1+(1-\lambda)(\mathbf{x}_2-\mathbf{x}_1)\right)\\
&=\varphi\left(\lambda\cdot 0+(1-\lambda)\cdot 1\,|\,\mathbf{x}_1,\mathbf{x}_2-\mathbf{x}_1\right)\\
&\le\lambda\varphi\left(0\,|\,\mathbf{x}_1,\mathbf{x}_2-\mathbf{x}_1\right)+(1-\lambda)\varphi\left(1\,|\,\mathbf{x}_1,\mathbf{x}_2-\mathbf{x}_1\right)\\
&=\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right)
\end{aligned}$$

显然根据定义，函数 f 均为凸函数。□

定理 1.12. [可微凸函数的一阶条件（First-Order Condition）] 假设 $f:\Omega\to\mathbb{R}$ 是定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上可微函数；换而言之，函数 f 的梯度（Gradient）∇f 在凸集 Ω 中的每个点都存在。函数 f 为凸集 Ω 上的凸函数的充分必要条件是，对于任意的 $\mathbf{x},\mathbf{y}\in\Omega$，存在

$$f\left(\mathbf{y}\right)\ge f\left(\mathbf{x}\right)+\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{y}-\mathbf{x}\right)\tag{1.27}$$

函数 f 为凸集 Ω 上的严格凸函数的充分必要条件是，对于任意的 $\mathbf{x},\mathbf{y}\in\Omega$，$\mathbf{x}\ne\mathbf{y}$，存在

$$f\left(\mathbf{y}\right)>f\left(\mathbf{x}\right)+\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{y}-\mathbf{x}\right)\tag{1.28}$$

证明：我们仅证明凸函数的结论，严格凸函数结论的证明与之类似。

首先证明必要性。给定 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，对于任意的实数 $\lambda\in(0,1)$，根据凸集的性质，有 $\mathbf{x}=\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\in\Omega$，应用一阶条件，得到

$$f\left(\mathbf{x}_1\right)\ge f\left(\mathbf{x}\right)+\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}\right)=f\left(\mathbf{x}\right)+(1-\lambda)\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)$$

$$f\left(\mathbf{x}_2\right)\ge f\left(\mathbf{x}\right)+\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_2-\mathbf{x}\right)=f\left(\mathbf{x}\right)-\lambda\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)$$

移项整理有

$$f\left(\mathbf{x}_1\right)-f\left(\mathbf{x}\right)\ge(1-\lambda)\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)$$

$$f\left(\mathbf{x}_2\right)-f\left(\mathbf{x}\right)\ge-\lambda\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)$$

加权合并这两个不等式有

$$\begin{aligned}
&\lambda\left[f\left(\mathbf{x}_1\right)-f\left(\mathbf{x}\right)\right]+(1-\lambda)\left[f\left(\mathbf{x}_2\right)-f\left(\mathbf{x}\right)\right]\\
&\ge\lambda(1-\lambda)\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)-\lambda(1-\lambda)\nabla f\left(\mathbf{x}\right)^T\left(\mathbf{x}_1-\mathbf{x}_2\right)=0
\end{aligned}$$

也即

$$\lambda f\left(\mathbf{x}_1\right)+(1-\lambda)f\left(\mathbf{x}_2\right)\ge f\left(\mathbf{x}\right)=f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)$$

则根据定义，函数 f 为凸集 Ω 上的凸函数。

接着证明充分性。给定$\mathbf{x}_1,\mathbf{x}_2\in\Omega$，$t\in(0,1)$，有

$$tf(\mathbf{x}_2)+(1-t)f(\mathbf{x}_1)\geq f(\mathbf{x}_1+t(\mathbf{x}_2-\mathbf{x}_1))$$

将上式移项，两边同时除以t，注意到$t>0$，有

$$f(\mathbf{x}_2)-f(\mathbf{x}_1)\geq\frac{f(\mathbf{x}_1+t(\mathbf{x}_2-\mathbf{x}_1))-f(\mathbf{x}_1)}{t}$$

令$t\to 0$，由极限保号性和方向导数的性质可得

$$f(\mathbf{x}_2)-f(\mathbf{x}_1)\geq\lim_{t\to 0}\frac{f(\mathbf{x}_1+t(\mathbf{x}_2-\mathbf{x}_1))-f(\mathbf{x}_1)}{t}=\nabla f(\mathbf{x}_1)^T(\mathbf{x}_2-\mathbf{x}_1)$$

则函数f满足一阶条件。□

从几何角度来看，如图 1.7 所示，凸函数曲线上任意点位于凸函数曲线切线对应点的上方。而凹函数正好与之相反。

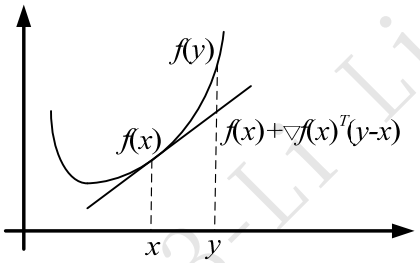


图 1.7　可微凸函数的一阶条件示意图。

定理 1.13. [二次可微凸函数的二阶条件（Second-Order Condition）] 假设$f:\Omega\to\mathbb{R}$是定义在凸集$\Omega\subseteq\mathbb{R}^n$上二次可微函数；换而言之，函数$f$的二阶导（Second-Order Derivative，或 Hessian）$\nabla^2 f$在凸集Ω中的每个点都存在。函数f为凸集Ω上的凸函数的充分必要条件是，对于任意的$\mathbf{x}\in\Omega$，Hessian 矩阵$\nabla^2 f(\mathbf{x})$是半正定矩阵

$$\nabla^2 f(\mathbf{x})\in S_+^{n\times n}\tag{1.29}$$

函数f为凸集Ω上的严格凸函数的充分必要条件是，对于任意的$\mathbf{x}\in\Omega$，Hessian 矩阵$\nabla^2 f(\mathbf{x})$是正定矩阵

$$\nabla^2 f(\mathbf{x})\in S_{++}^{n\times n}\tag{1.30}$$

证明：我们仅证明凸函数的结论，严格凸函数结论的证明与之类似。

我们首先使用反证法证明必要性。假设在点$\mathbf{x}\in\Omega$，存在$\nabla^2 f(\mathbf{x})$的某个特

征值为负。则一定存在非零向量 $\mathbf{d}\in\mathbb{R}^n$ 使得 $\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d}<0$。考察函数 f 在点 $\mathbf{x}$ 附近的带 Peano 余项的二次 Taylor 展开，我们有

$$f(\mathbf{x}+t\mathbf{d})=f(\mathbf{x})+t\nabla f(\mathbf{x})^T\mathbf{d}+\frac{t^2}{2}\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d}+o(t^2)$$

其中 $t>0$。

移项后等式两边同时除以 t^2，有

$$\frac{f(\mathbf{x}+t\mathbf{d})-f(\mathbf{x})-t\nabla f(\mathbf{x})^T\mathbf{d}}{t^2}=\frac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d}+o(1)$$

显然，当 t 充分小时，有

$$\frac{f(\mathbf{x}+t\mathbf{d})-f(\mathbf{x})-t\nabla f(\mathbf{x})^T\mathbf{d}}{t^2}<0$$

也即 $f(\mathbf{x}+t\mathbf{d})<f(\mathbf{x})+t\nabla f(\mathbf{x})^T\mathbf{d}$。这和定理 1.13 的一阶条件矛盾，因此必有 $\nabla^2 f(\mathbf{x})\in S_+^{n\times n}$。

我们接着证明充分性。给定 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，考察函数 f 在点 $\mathbf{x}$ 附近的二次 Taylor 展开，我们有

$$f(\mathbf{x}_2)=f(\mathbf{x}_1)+\nabla f(\mathbf{x}_1)^T(\mathbf{x}_2-\mathbf{x}_1)+\frac{1}{2}(\mathbf{x}_2-\mathbf{x}_1)^T\nabla^2 f\big(\mathbf{x}_1+t(\mathbf{x}_2-\mathbf{x}_1)\big)(\mathbf{x}_2-\mathbf{x}_1)$$

其中 $t\in(0,1)$ 是和 $\mathbf{x}_1,\mathbf{x}_2$ 有关的常数。

如果 $\nabla^2 f(\mathbf{x})\in S_+^{n\times n}$，则

$$f(\mathbf{x}_2)\geq f(\mathbf{x}_1)+\nabla f(\mathbf{x}_1)^T(\mathbf{x}_2-\mathbf{x}_1)$$

由定理 1.13 可知函数 f 为凸集 Ω 上的凸函数。□

定理 1.14. 下述函数是凸函数：

定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的指数函数 $f=e^{\mathbf{x}}$，$\mathbf{x}\in\Omega$。

定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的负对数函数 $f=-\ln(\mathbf{x})$，$\mathbf{x}\in\Omega$。

定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的负对数函数 $f=\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{q}^T\mathbf{x}+r$，$\mathbf{x}\in\Omega$，其中 $\mathbf{P}\in S_+^{n\times n}$，$\mathbf{q}\in\mathbb{R}^n$，$r\in\mathbb{R}$。

下述函数是凸函数：

定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的对数函数 $f=\ln(\mathbf{x})$，$\mathbf{x}\in\Omega$。

下述函数既是凸函数，又是凹函数：

定义在凸集 $\Omega\subseteq\mathbb{R}^n$ 上的仿射函数（Affine Function）$f=\mathbf{a}^T\mathbf{x}+b$，$\mathbf{x}\in\mathbb{R}^n$，其中 $\mathbf{a}\in\mathbb{R}^n$ 为给定的 n 维向量，$b\in\mathbb{R}$ 为给定实数。

证明：均可根据定义证明。本书从略。□

定理 1.15. 有限个凸函数在共同的定义域内逐点取最大值（Pointwise

Maximum）所得到的函数还是凸函数。有限个凹函数在共同的定义域内逐点取最小值（Pointwise Minimum）所得到的函数还是凹函数。

无限个凸函数在共同的定义域内逐点取极大值（Pointwise Supremum）所得到的函数还是凸函数。无限个凹函数在共同的定义域内逐点取最小值（Pointwise Infinimum）所得到的函数还是凹函数。

证明：我们只需证明两个凸函数的情况，考虑两个凸函数 $f_1(\mathbf{x})$ 和 $f_2(\mathbf{x})$，对它们逐点取极大值所得到的函数为 $f(\mathbf{x})=\max\{f_1(\mathbf{x}),f_2(\mathbf{x})\}$，$\mathrm{dom}f=\mathrm{dom}f_1\cap\mathrm{dom}f_2$。对于任意的 $\mathbf{x}_1,\mathbf{x}_2\in\Omega$，和任意实数 $\lambda\in[0,1]$ 存在

$$
\begin{aligned}
&f\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)=\max\left\{f_1\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right),f_2\left(\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right)\right\}\\
&\le\max\left\{\lambda f_1(\mathbf{x}_1)+(1-\lambda)f_1(\mathbf{x}_2),\lambda f_2(\mathbf{x}_1)+(1-\lambda)f_2(\mathbf{x}_2)\right\}\\
&\le\lambda\max\left\{f_1(\mathbf{x}_1),f_2(\mathbf{x}_1)\right\}+(1-\lambda)\max\left\{f_1(\mathbf{x}_2),f_2(\mathbf{x}_2)\right\}\\
&=\lambda f(\mathbf{x}_1)+(1-\lambda)f(\mathbf{x}_2)
\end{aligned}
$$

根据数学归纳法，我们可以得知道对于有限个凸函数，上述结论成立。类似可以证明对于有限个凸函数，上述结论成立。

而对于无限个凸函数或者凹函数的情况，可以引入上境图（Epigragh）概念之后，将上述证明稍作修改得证。本书从略。□

例 1.3. 试证明 $f(\mathbf{x})=\ln\left(\sum_{i=1}^{n}\exp(x_i)\right):\mathbb{R}^n\to\mathbb{R}$ 为凸函数。

解：令 $z_i=\exp(x_i)$，$\mathbf{z}=[z_1,z_2,...,z_n]^n$，则

$$
\nabla^2 f(\mathbf{x})=\frac{1}{\mathbf{1}^T\mathbf{z}}\begin{bmatrix}z_1&0&...&0\\0&z_2&...&...\\...&...&...&0\\0&...&0&z_n\end{bmatrix}-\frac{1}{\left(\mathbf{1}^T\mathbf{z}\right)^2}\mathbf{z}^T\mathbf{z}
$$

对于任意 $\mathbf{y}\in\mathbb{R}^n$，我们可以得到

$$
\mathbf{y}^T\nabla^2 f(\mathbf{x})\mathbf{y}=\frac{\left(\sum_{i=1}^{n}z_iy_i^2\right)\left(\sum_{i=1}^{n}z_i\right)-\left(\sum_{i=1}^{n}z_iy_i\right)^2}{\left(\sum_{i=1}^{n}z_i\right)^2}
$$

根据 Cauchy-Schwarz 不等式，$\left(\sum_{i=1}^{n}z_iy_i^2\right)\left(\sum_{i=1}^{n}z_i\right)\ge\left(\sum_{i=1}^{n}z_iy_i\right)^2$，因此

$\nabla^2 f(\mathbf{x}) \in S_+^{n\times n}$，该函数是凸函数。

1.6　计算复杂性

计算复杂性理论（Computational complexity theory）是理论计算机科学和数学的一个分支，主要研究不同问题的可计算性，以及求解这些问题的算法的复杂性[14]-[15]。

我们常用时间复杂性（即把计算分成一些原子操作，需要通过多少步原子操作才能解决问题）和空间复杂性（需要多少内存才能解决问题）来衡量一个有明确定义的算法（Algorithm）的计算复杂度。以下我们重点讨论时间复杂性，相关结论稍作转换也适用于空间复杂性。

对于运筹优化问题而言，假设我们考虑一个 n 个决策变量的问题。给定算法的最坏情况复杂度（Worst-Case Complexity）是指：在最差的问题条件下，算法花费的求解时间和 n 的关系。给定算法的平均情况复杂度（Average-Case Complexity）是指：在所有可能的问题条件下，算法平均花费的求解时间和 n 的关系。一般来说，如果使用“复杂度”一词且不进行进一步说明，我们指的是考虑最坏情况时间复杂度。

复杂度的评估依赖于计算模型（Model of Computation）的选择。计算模型主要设定哪些操作是可允许的操作，以及哪些操作是在一个时间单位内完成的原子操作。对于运筹优化问题而言，我们主要关注确定性模型和非确定模型。

确定性模型（Deterministic Model）是一种计算机器的后续状态和要执行的操作完全由前面的状态决定的计算模型。历史上提出的 lambda 演算模型（lambda Calculus）、图灵机（Turing Machines）、随机存取机器（Random Access Machines），随机存取储存程序机器（Random Access Stored Program Machines）都属于确定性模型。可以证明这些确定性计算模型求解问题的计算能力是等价的，但在计算速度上有不同。目前，冯·诺依曼架构真实计算机的计算模型基本等效为随机存取储存程序机器。确定性模型规定了在任何给定情况下，最多只能执行一个动作，因此计算能力受限。

非确定性模型（Non-Deterministic Model）允许在某些给定情况下同时执行了多个可能的动作，并衡量执行所有可能的选择后做出最佳选择时所需的时间，以所有动作中花费的最少时间做出最佳选择的动作来计算执行时间。换句话说，我们可以认为计算是多个（相同的）处理器上同时进行的，而不确定性模型花费的计算时间是第一个完成计算的处理器所花费的时间。

如果一个问题可以在确定性模型上以多项式时间（Polynomial Time）求解，

则该问题属于 P 类问题。例如，我们优化如下问题：$\min_{\mathbf{y}}\left\{\mathbf{d}^T\mathbf{y} \mid \mathbf{y}\in\Gamma\right\}$，集合 Γ 只含有可数的 n 个元素。则我们枚举一遍 $\mathbf{y}$ 所有可能的取值，就能找到该问题的最优解。此时我们花费的时间和空间资源均为 n 若干倍即可，我们用 $O(n)$ 来衡量。

如果我们求解一个问题花费的时间和空间资源可以用 $O(n^k)$ 来衡量，$k\in\mathbb{Z}$ 为某常数，则称之为以多项式时间求解。

如果一个问题可以在非确定性机器上以多项式时间（Polynomial Time）求解，则该问题属于 NP 类问题。可以证明，所有 P 类问题都属于 NP 类问题。但我们尚不知道是否所有 NP 类问题也属于 P 类问题。亦即，我们不知道是否 P 类问题和 NP 类问题是等价的。这就是所谓的“P=NP?”问题

如果一个问题属于 NP 问题，且不比其它任何 NP 问题更简单，则称该问题为 NP 完全问题（Non-Deterministic Polynomial-Complete Problems，NP-Complete）。例如背包问题（Knapsack Problem）、旅行推销员问题（Travelling Salesman Problem）都是 NP 完全问题 [14]。如果这些问题中的任何一个可以在确定性模型上在多项式时间内求解，那就意味着所有 NP 问题都可以在多项式时间内求解，我们立即可以得出结论：P = NP。但目前所有这些问题的算法在确定性模型上都具有指数复杂度，我们尚不知道是否 P = NP。

1.7 数据结构

数据结构是指存在一种或多种特定相互关系的数据元素的集合。所谓“结构”通常指数据元素之间存在的逻辑结构和存储结构。在理论和现实计算机中，我们往往采用特定数据结构来关联和存储数据。针对特定的问题选择合适的数据结构将带来更高的运行或者存储效率。数据结构往往同各种算法联系在一起。算法的设计取决于数据的逻辑结构，而算法的实现依赖于给定的存储结构 [16]。

我们一般按照数据的逻辑结构将常见的数据结构分为线性结构和非线性结构两类。如果我们将每个数据的储存单元看作一个节点，则线性结构中各个结点具有线性关系。线性结构有且仅有一个开始结点和一个终端结点。线性结构所有结点都最多只有一个直接前驱结点和一个直接后继结点。非线性结构中各个结点之间具有多个对应关系。换而言之，非线性结构的一个结点可能有多个直接前驱结点和多个直接后继结点。

下面我们介绍几个常用的数据结构。

数组（Array）是将具有相同类型的若干变量有序组织在一起的集合，可以有一维、二维以及多维等表现形式。数组中元素可以顺序依次访问，也可以按数组

下标直接访问。

栈（Stack）将具有相同类型的若干变量按照先进后出（或着等价的后进先出）原则来存储的集合。我们只能在一个栈的一个固定端进行数据的插入和删除操作。先插入的数据将被压入栈底，最后插入的数据在栈顶，读出数据时，只能从栈顶开始逐个读出和删除操作，直到栈变空。

队列（亦称队，Queue）将具有相同类型的若干变量按照先进先出（或着等价的后进后出）原则来存储的集合。和栈不同的是，我们只能在一个队列的一端进行插入操作；而在另一端进行读出和删除操作，直到队列变空。进行插入操作的一端称为队尾，进行删除操作的一端称为队头。

树（Tree）中有且仅有一个根结点，该结点没有前驱结点。树中的其它结点都有且仅有一个前驱结点，但可能有多个后继结点。

图（Graph）中的数据结点一般称为顶点，而边是顶点的有序偶对。如果两个顶点之间存在一条边，那么就表示这两个顶点具有相邻关系。

显然，栈、队列等数据结构属于线性结构。数组、树和图等数据结构属于非线性结构。

1.8　作业题

1.1　假设矩阵 $\mathbf{A}\in\mathbb{R}^{n\times n}$ 是满秩的，$\mathrm{rank}(\mathbf{A})=n$。请证明或证否：可以通过改变 $\mathbf{A}$ 的至多 m^2 个元素的值，$m\le n$，使得新得到的矩阵 $\bar{\mathbf{A}}$ 满足 $\mathrm{rank}(\bar{\mathbf{A}})=n-m$。

1.2　如果一个集合 $S\in\mathbb{R}^n$，且 S 和 $\mathbb{R}^n$ 空间中任意一条直线的交集为凸集，试证明集合 S 为凸集。

1.3　试证明一个多面体是有界多面体，当且仅当该多面体不包含任何射线。

1.4　假设以下集合均为非空集合，请判断哪些集合一定有极点，并给出理由：

a）$\Omega=\{\mathbf{x}\mid\mathbf{Ax}=\mathbf{b},\mathbf{x}\ge\mathbf{0},\mathbf{x}\in\mathbb{R}^n,\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$。

b）$\Omega=\{\mathbf{x}\mid\mathbf{Ax}\ge\mathbf{b},\mathbf{x}\in\mathbb{R}^n,\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$，其中 $\mathbf{A}$ 为行满秩矩阵。

c）$\Omega=\{\mathbf{x}\mid\mathbf{Ax}\ge\mathbf{b},\mathbf{x}\in\mathbb{R}^n,\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$，其中 $\mathbf{A}$ 为列满秩矩阵。

d）$\Omega=\{\mathbf{x}\mid\mathbf{Ax}\ge\mathbf{0},\mathbf{c}^T\mathbf{x}=-1,\mathbf{x}\in\mathbb{R}^n,\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m,\mathbf{c}\in\mathbb{R}^n\}$，其中 $\mathbf{A}$ 为列满秩矩阵。

1.5　试证明多面体可行域 $\{\mathbf{x}\in\mathbb{R}^n\mid\mathbf{Ax}\ge\mathbf{b}\}$ 存在至少一个极点，必有 $\mathrm{rank}(\mathbf{A})=n$。

1.6　试证明 $\Omega=\{\mathbf{x}\in\mathbb{R}^n\mid\mathbf{Ax}=\mathbf{b},\mathbf{x}\ge\mathbf{0},\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$ 为闭集。

1.7　[Gordan's Theorem] 试证明给定矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$，下述两个线性系统有且仅有一个存在可行解 [17]：

线性系统 I：对于 $\mathbf{x}\in\mathbb{R}^n$， $\mathbf{Ax}<\mathbf{0}$

线性系统 II：对于 $\mathbf{y}\in\mathbb{R}^m$， $\mathbf{A}^T\mathbf{y}=\mathbf{0}$， $\mathbf{y}\geq 0$ 但 $\mathbf{y}\neq 0$

1.7 [Stiemke's Theorem] 试证明给定矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$，下述两个线性系统有且仅有一个存在可行解 [18]：

线性系统 I：对于 $\mathbf{x}\in\mathbb{R}^n$， $\mathbf{Ax}\geq\mathbf{0}$ 但 $\mathbf{Ax}\neq\mathbf{0}$

线性系统 II：对于 $\mathbf{y}\in\mathbb{R}^m$， $\mathbf{A}^T\mathbf{y}\geq\mathbf{0}$， $\mathbf{y}>0$

1.8 [Motzkin's Transportation Theorem] 试证明给定矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$， $\mathbf{B}\in\mathbb{R}^{k\times n}$，$\mathbf{b}\in\mathbb{R}^m$， $\mathbf{d}\in\mathbb{R}^k$，如果线性系统 I 无解，则线性系统 II 或者线性系统 III 中至少有一个存在可行解 [19]：

线性系统 I：对于 $\mathbf{x}\in\mathbb{R}^n$， $\mathbf{Ax}\geq\mathbf{b}$， $\mathbf{Bx}>\mathbf{d}$

线性系统 II：对于 $\mathbf{y}\in\mathbb{R}^m$， $\mathbf{v}\in\mathbb{R}^k$， $\begin{cases}\mathbf{y}^T\mathbf{A}+\mathbf{v}^T\mathbf{B}=\mathbf{0},\mathbf{y}^T\mathbf{b}+\mathbf{v}^T\mathbf{d}>0\\ \mathbf{y}\geq\mathbf{0},\mathbf{v}\geq\mathbf{0}\end{cases}$

线性系统 III：对于 $\mathbf{y}\in\mathbb{R}^m$， $\begin{cases}\mathbf{y}^T\mathbf{A}+\mathbf{v}^T\mathbf{B}=\mathbf{0},\mathbf{y}^T\mathbf{b}+\mathbf{v}^T\mathbf{d}\geq 0\\ \mathbf{y}\geq\mathbf{0},\mathbf{v}\geq\mathbf{0},\mathbf{v}\neq\mathbf{0}\end{cases}$

1.9 试证明定理 1.14。

1.10 [Carlson Inequality] 试证明 $m\times n$ 的非负实数矩阵 $\mathbf{X}\in\mathbb{R}_{++}^{m\times n}$ 中，n 列每列元素之和的几何平均值不小于矩阵中 m 行每行元素的几何平均值之和

$$\left[\prod_{i=1}^{m}\left(\sum_{j=1}^{n}x_{ij}\right)\right]^{\frac{1}{m}}\geq\sum_{j=1}^{n}\left(\prod_{i=1}^{m}x_{ij}\right)^{\frac{1}{m}}$$

1.9 参考文献

[1] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.

[2] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1996.

[3] D. Bertsimas, J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific Press, 1997.

[4] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, 2003.

[5] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[6] J. Nocedal, S. J. Wright, *Numerical Optimization*, 2nd edition, Springer, 2006.

[7] D. P. Bertsekas, *Convex Optimization Algorithms*, Athena Scientific, 2015.

[8] D. G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, 4th edition, Springer, 2016.

[9] 《运筹学》教材编写组，*运筹学*，第 4 版本科版，清华大学出版社, 2013.

[10] 刁在筠，刘桂真，戎晓霞，王光辉，*运筹学*，第 4 版，高等教育出版社, 2016.

[11] 胡运权，郭耀煌，*运筹学教程*，第 5 版，清华大学出版社, 2018.

[12] G. Polya, *How to Solve It: A New Aspect of Mathematical Method*, Princeton University Press, 1945.

[13] G. Farkas, "A Fourier-féle mechanikai elv alkalmazásának algebrai alapjam," *Mathematikai Természettudományi Értesitö*, vol. 16, pp. 361-364, 1898.

[14] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computation* , R. E. Miller, J. W. Thacher, eds., Plenum Press, pp. 85 103, 1972.

[15] O. Goldreich, P*, NP, and NP-Completeness: The Basics of Computational Complexity*, Cambridge University Press, 2010.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 4th edition, The MIT Press, 2022.

[17] P. Gordan, "Ueber die Auflösung linearer Gleichungen mit reellen Coefficienten," *Mathematische Annalen*, vol. 6, pp. 23-28, 1873.

[18] E. Stiemke, "Über positive Lösungen homogener linearer Gleichungen" *Mathematische Annalen*, vol. 76, pp. 340-342, 1915.

[19] T. S. Motzkin, *Beiträge zur Theorie der linearen Ungleichungen*, Inaugural Dissertation, Basel, Jerusalem, 1936.

第2章 线性规划

2.1 引言

线性规划（Linear Programming，LP）是运筹学中的一个重要分支，研究线性约束条件下线性目标函数的实数极值规划问题。线性规划有着研究较早、应用广泛、方法较成熟的特点，常被作为运筹学的基础。随着计算机的逐渐普及，我们已经可以轻松求解数十万变量乃至更大的线性规划问题。目前，线性规划在工程技术、商业活动、军事行动和工农业生产中发挥着越来越广泛的作用。

本章先介绍线性规划的基本概念；接着通过几何和代数的观点分析线性规划问题；然后介绍线性规划问题中广泛使用的对偶理论和单纯型法；最后讨论一些相关问题。

2.2 线性规划基础

2.2.1 线性规划的历史

1939 年，苏联学者 Kantorovich 关注到具有线性约束的极值优化问题，并提出了解乘数法，对资源最优分配理论做出了贡献。该方法是线性规划的雏形。1975 年，Kantorovich 获得诺贝尔经济学奖。

与此同时，美国研究者们也开始关注线性规划问题。1941 年，Hitchcock 提出运输问题 [4]；1945 年，Koopmans 同样对运输问题进行了研究 [5]。1947 年，Dantzig 担任美国空军数学顾问，为找到解决问题的机制化工具，完整提出了在一组线性方程或不等式约束下，求某一线性形式极小值问题的数学模型，也即线性规划问题；并提出了单纯型法（Simplex Method）。1948 年，Koopmans 和 Dantzig 在海滩散步时共同提出使用“线性规划”作为该类问题的名称 [6]。

1971 年，Klee 和 Minty 构造出一个例子，证明在最坏情况下，单纯形法是一个指数时间算法 [7]。

1979 年，Khachiyan 提出了第一个解决线性规划问题的多项式时间算法——椭球法（Ellipsoid Method）[8]。这个算法证明了线性规划问题存在多项式时间算法，但椭球法实际计算效率不高。

1984 年，Karmarkar 发明了内点法（Interior Point Method），这是线性规划第一个实际可用的多项式时间算法 [9]。

2004 年，Spielman 和 Teng 提出算法复杂性平滑分析法（Smoothed Analysis of Algorithms），证明单纯型法在很多线性规划问题的计算中只需多项式时间即可收敛到最优解 [10]。

现在有多种成熟的软件包来计算求解大规模线性规划问题。这些软件包大多同时集成实现了单纯型法和内点法。

2.2.2 典型问题和图解法

在各类经济活动中，经常遇到这样的问题：在特定的生产条件下，如何通过统筹安排，合理安排人力、物力资源，组织生产过程，使总的经济效益最好。这种问题常常可以化成或近似地化成线性规划问题。本节举例介绍线性规划的典型问题，然后给出解决线性规划最基本的图解法。

例 2.1. 某公司可以生产 A、B 两种产品，已知每生产 1 吨这两种产品的价格，必须的原料 1、2、3 的多少及原料每日供及量的上限如下表所示，试问该公司应如何制定生产计划，使得在满足要求的情况下总的产品价格最多。

	每吨产品 A	每吨产品 B	每日该原料最高需求量
原料 1	0 吨	5 吨	15 吨
原料 2	6 吨	2 吨	24 吨
原料 3	1 吨	1 吨	5 吨
产品价格	2 万元	1 万元	

解：首先，建立数学规划模型：

① 确定决策变量：

设 x_1 为计划生产产品 A 的量， x_2 为计划生产产品 B 的量

② 确定目标函数：

目标函数为： $\max\limits_{x_1,x_2} 2x_1 + x_2$

③ 确定约束条件：

$$\begin{cases} 5x_2 \le 15 \\ 6x_1 + 2x_2 \le 24 \\ x_1 + x_2 \le 5 \\ x_1 \ge 0, x_2 \ge 0 \end{cases}$$

④ 建立线性规划模型：

$$\begin{aligned} \max_{x_1,x_2} \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & 5x_2 \le 15 \\ & 6x_1 + 2x_2 \le 24 \\ & x_1 + x_2 \le 5 \\ & x_1 \ge 0, x_2 \ge 0 \end{aligned}$$

其次，利用图解法求得具体数值解：

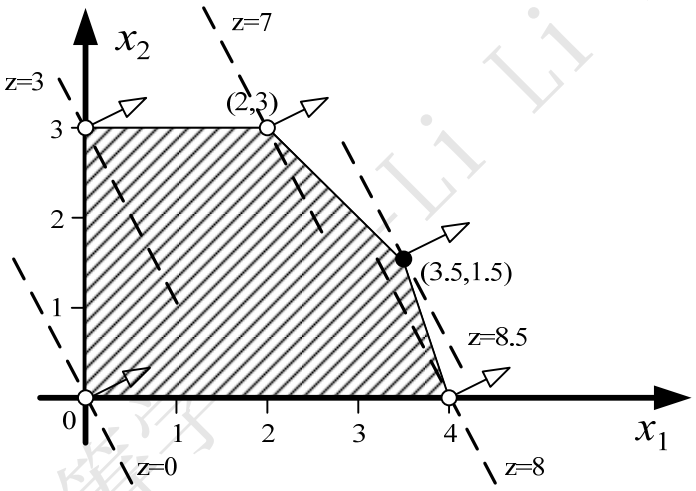


图 2.1　生产问题的图解法示例

对于少于 4 个变量的线性规划模型可以建立图解表示。如图 2.1 所示，生产问题所有可行解围成的区域为斜纹阴影区域，也即点$(0,0)$，$(0,3)$，$(2,3)$，$(3.5,1.5)$，$(4,0)$围成的五边形，这些点同时是该五边形的 5 个极点。虚线是目标函数的若干条等值线，空心箭头是目标函数的梯度方向。点$(0,0)$，$(0,3)$，$(2,3)$，$(3.5,1.5)$，$(4,0)$处的目标函数分别为0，3，7，8.5，8。不难观察得知，实心点$(3.5,1.5)$是该问题的最优解。

例 2.2. 农民特德有 500 英亩的土地种植小麦、玉米或甜菜。他需要 200 吨小麦和 240 吨玉米来养牛。这些农作物可以在他的土地上种植，也可以从批发商那里购买。任何超过这些数量的产品都可以出售：小麦 170 美元/吨,玉米 150 美元/吨。任何短缺必须以小麦 238 美元/吨，玉米 210 美元/吨的价格从批发商处购买。特德也能种甜菜。少于 6000 吨甜菜可以卖 36 美元/吨。但由于甜菜生产的经济配额，超过 6000 吨的甜菜只能以每吨 10 美元的价格出售。亩产量：小麦 2.5 吨/英亩,玉米 3 吨/英亩,甜菜 20 吨/英亩。种植花费：小麦 150 美元/英亩,玉米 230 美元/英亩,甜菜 260 美元/英亩。请问特德如何种地才能获得最大收益？

解：① 确定决策变量：

小麦、玉米和甜菜种植的亩数分别为：x_W， x_C， x_B

小麦、玉米和甜菜（高价）卖出的吨数：w_W， w_C， w_B

甜菜（低价）卖出的吨数：e_B

从批发商处购买的小麦和玉米吨数：y_W， y_C

② 确定目标函数：

$$\max -150x_W - 230x_C - 260x_B - 238y_W + 170w_W - 210y_C + 150w_C + 36w_B + 10e_B$$

③ 约束条件：

$$x_W + x_C + x_B \le 500$$

$$2.5x_W + y_W - w_W = 200$$

$$3x_C + y_C - w_C = 240$$

$$20x_B - w_B - e_B = 0$$

$$w_B \le 6000$$

$$x_W, x_C, x_B, y_W, y_C, e_B, w_W, w_C, w_B \ge 0$$

由于该问题的特殊性，少于 6000 吨时甜菜卖价要小于多于 6000 吨时甜菜卖价，这使得我们可以简单设计如上的决策变量和线性约束关系。请读者思考，如果少于 6000 吨时甜菜卖价要大于多于 6000 吨时甜菜卖价，还是否能设计如此简单的决策变量和线性约束关系。

④ 数值求解：

可利用单纯型法求出最优解为：

	小麦	玉米	甜菜
种植亩数/亩	120	80	300
产量/吨	300	240	6000
销售量/吨	100	0	6000
购买量/吨	0	0	0

最终的利润为：118600 美元。

2.2.3　线性规划问题的定义和表达形式转换

研究者从实际问题中抽象得到了线性规划问题的一般定义，进而给出了不同线性规划问题表达形式的转换方法，以便针对标准形式的线性规划问题进行分析和求解。

定义 2.1. 线性规划问题的标准形式（Standard Form）为如下形式

$$\begin{aligned} \min_{x_1,x_2,\ldots,x_n} \quad & z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ & \ldots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \\ & x_1 \geq 0, \ldots, x_n \geq 0 \end{aligned} \tag{2.1}$$

其中 $x_i, a_{ij}, b_i, c_i \in \mathbb{R}$，$i = 1, \ldots, m$，$j = 1, \ldots, n$。$x_i$ 为待定决策变量。一般的，我们有 $m < n$，也即约束条件个数小于决策变量的个数。

我们也可以矩阵向量的形式将标准形式记为

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.2}$$

其中向量 $\mathbf{x} \in \mathbb{R}^n$ 也被称为决策变量（Decision Variables），$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{A} \in \mathbb{R}^{m \times n}$，$\mathbf{c} \in \mathbb{R}^n$。这里，向量 $\mathbf{c} = \begin{bmatrix} c_1 & \ldots & c_n \end{bmatrix}^T$ 称为价值向量，$\mathbf{c}^T\mathbf{x}$ 称为目标函数。向量 $\mathbf{b} = \begin{bmatrix} b_1 & \ldots & b_m \end{bmatrix}^T$ 称为右侧向量。

需要说明的是，有些教材将线性规划问题的标准形式设为最大化目标函数 $z = \mathbf{c}^T\mathbf{x}$。这在分析和求解时没有太大影响，仅需要在后面谈到的单纯型算法条件数检验的时候考虑相反的符号设置即可。我们这里设置线性规划问题的标准形式设为最小化目标函数 $z = \mathbf{c}^T\mathbf{x}$，是为了和后续非线性规划问题的分析保持一致。这也是国际教材中常见的定义方式。

定义 2.2. 线性规划问题的规范形式（Canonical Form）为如下形式

$$\begin{aligned}\min_{x_1,x_2,...,x_n} \quad & z = c_1x_1 + c_2x_2 + ... + c_nx_n \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \geq b_2 \\ & ... \\ & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \geq b_m \\ & x_1 \geq 0, ..., x_n \geq 0\end{aligned} \tag{2.3}$$

其中 $x_i, a_{ij}, b_i, c_i \in \mathbb{R}$，$i = 1, ..., m$，$j = 1, ..., n$。$x_i$ 为待定决策变量。

我们也可以矩阵向量的形式将规范形式记为

$$\begin{aligned}\min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}\end{aligned} \tag{2.4}$$

其中向量 $\mathbf{x} \in \mathbb{R}^n$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{A} \in \mathbb{R}^{m \times n}$，$\mathbf{c} \in \mathbb{R}^n$。

实际上，线性规划问题的标准形式和规范形式是等价的。我们也可以将任意的线性规划问题通过变换归为标准形式的线性规划问题。变换规则如下：

1）如果一个线性规划问题的目标函数是求价值函数 $\mathbf{c}^T\mathbf{x}$ 的最大值，则我们可以等价的求新的价值函数 $-\mathbf{c}^T\mathbf{x}$ 的最小值；反之亦然。

2）对于一个线性规划问题的等式约束条件

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \tag{2.5}$$

我们可以用如下两个不等式约束条件等价替换

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i, \quad -\sum_{j=1}^{n} a_{ij}x_j \geq -b_i \tag{2.6}$$

3）对于一个线性规划问题的不等式约束条件

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \tag{2.7}$$

我们可以引入一个新的决策变量 x_{n+1}，将上述不等式约束条件等价写作

$$\sum_{j=1}^{n} a_{ij}x_j - x_{n+1} = b_i, \quad x_{n+1} \geq 0 \tag{2.8}$$

4）对于一个无非负约束的变量 x_j，我们可以引入两个新的决策变量 $x_j^+ \geq 0$ 和 $x_j^- \geq 0$，将变量 x_j 替换为 $x_j = x_j^+ - x_j^-$。

2.3　线性规划问题的有解性和几何解释

定义 2.3. 满足线性规划问题所有约束条件的解被称为可行解（Feasible Solution）。线性规划问题所有可行解所组成的集合称为可行区域（Feasible Region，或者 Feasible Set）。其中最小化目标函数的可行解被称为最优解。

定理 2.1. 标准形式线性规划问题（2.3）或者规范形式线性规划问题（2.4）的可行区域构成一个下有界的闭凸集。

证明：注意到标准形式线性规划问题（2.3）的约束条件分为两个部分，一部分是有限个超平面的交集，另一部分是闭凸集 $\mathbb{R}^n_+$。这两个部分的交集构成最终的可行区域。因为闭凸集 $\mathbb{R}^n_+$ 下有界。而且闭集合的交集保持为闭集。结论显然。

类似的，注意到标准形式线性规划问题（2.4）的约束条件分为两个部分，一部分是有限个半平面的交集，另一部分是闭凸集 $\mathbb{R}^n_+$。这两个部分的交集构成最终的可行区域。同样，结论显然。□

而根据等价转换方式（2.6），标准线性规划问题也可以等价转换为规范形式线性规划问题，其对应的可行区域同样可以视为一个下有界的多面体。

一般而言，求解线性规划问题有三种可能的情况。

第一种情况，线性规划问题无解（Infeasible）。例如如下问题

$$\begin{aligned} \min_{x_1,x_2} \quad & z = x_1 + x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 2 \\ & -x_1 - x_2 \geq -1 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

第二种情况，线性规划问题存在无界解（Unbounded），也即随着决策变量趋向于无穷，目标函数值也会趋向于无穷。例如如下问题

$$\begin{aligned} \min_{x_1,x_2} \quad & z = x_1 - x_2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

第三种情况，线性规划问题存在有界的最优解。从 2.2.2 节介绍的图解法可以观察到，例 2.1 中的线性规划问题最优解在对应的可行区域多面体的一个极点上取得。下面，我们将这一观察结果推广到一般线性规划问题。

定理 2.2. 如果标准形式线性规划问题（2.3）或者规范形式线性规划问题（2.4）存在有界的最优解，则一定在其可行区域多面体的一个极点上取得最优解。

证明：我们首先使用反证法证明，该最优解必然在可行域的一个边界点取得。假设最优解在可行域内部的一个点$\mathbf{x}_0$处取得，则存在$2\varepsilon > 0$，使得以$\mathbf{x}_0$为球心，2ε为半径的开球位于Ω内部，也即$B_{2\varepsilon}(\mathbf{x}_0) \subset \Omega$。我们可以由此确定一个新点

$$\mathbf{x}_1 = \mathbf{x}_0 - \varepsilon \frac{\mathbf{c}}{\|\mathbf{c}\|_2} \in B_{2\varepsilon}(\mathbf{x}_0) \subset \Omega$$

不难发现，该点对应的目标函数值更小

$$\mathbf{c}^T\mathbf{x}_1 = \mathbf{c}^T\mathbf{x}_0 - \varepsilon\mathbf{c}^T \frac{\mathbf{c}}{\|\mathbf{c}\|_2^2} = \mathbf{c}^T\mathbf{x}_0 - \varepsilon < \mathbf{c}^T\mathbf{x}_0$$

这和我们假设的点$\mathbf{x}_0$处取得最优解的假设矛盾，因此最优解不可能在可行域的一个内部点取得。

假设该问题有界的最优解值为z。对于任意可行的$\mathbf{x}$，$\mathbf{c}^T\mathbf{x} \le z$，则根据定理1.7，$H = \{\mathbf{x} \mid \mathbf{c}^T\mathbf{x} = z, \mathbf{x} \in \mathbb{R}^n\}$为可行域下有界多面体的一个支撑超平面。而根据定理1.8，该支撑超平面上一定存在一个极点，该极点同时也是可行域多面体的一个极点；在这个极点上，线性规划问题取得最优解。□

进一步细分的话，存在有界最优解的线性规划问题还可以分为有唯一最优解和有无穷多个最优解的情况。例如下述问题就存在无穷多个最优解

$$\begin{aligned} \min_{x_1,x_2} \quad & z = x_1 + x_2 \\ \text{s.t.} \quad & x_1 + x_2 \ge 1 \\ & x_1 \ge 0, x_2 \ge 0 \end{aligned}$$

例 2.3. 如果某个线性规划问题在两个不同的点取得最优解，则该问题存在无穷多个最优解。

解：假设某个线性规划问题的可行域为Ω，在集合Ω内两个不同的点$\mathbf{x}_1, \mathbf{x}_2 \in \Omega$取得最优解$\mathbf{c}^T\mathbf{x}_1 = \mathbf{c}^T\mathbf{x}_2 = z$。则任意取实数$\lambda \in (0,1)$，可以得到

$$\mathbf{c}^T\left[\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2\right] = \lambda\mathbf{c}^T\mathbf{x}_1 + (1-\lambda)\mathbf{c}^T\mathbf{x}_2 = \lambda z + (1-\lambda)z = z$$

显然，该两点连线段上的任意一点同样取得最优解；故这样点有无穷个。□

从几何角度来看，我们可以搜索线性规划问题的所有极点，其中对应目标函数值最低的极点可能就是该线性规划问题的最优解（相关讨论作为作业题2.1-2.2留给大家练习）。不过在具体计算求解线性规划问题时，我们还需要利用代数方法来刻画极点搜索的过程。

2.4　线性规划的代数解释

定义 2.4. 如果标准形式线性规划问题（2.3）的一个可行解 $\mathbf{x}$，根据定义 1.1 同时是方程组 $\mathbf{Ax}=\mathbf{b}$ 的基本解，则 $\mathbf{x}$ 是标准形式线性规划问题（2.3）的一个基本可行解（Basic Feasible Solution）。如果一个基本可行解是非退化的，则称之为一个非退化基本可行解（Non-Degenerate Basic Feasible Solution），否则称之为一个退化基本可行解（Degenerate Basic Feasible Solution）

不失一般性，我们可以通过从 $\mathbf{A}$ 的 n 列中选择线性无关的 m 列而形成的可逆方阵 $\mathbf{A_B}\in\mathbb{R}^{m\times m}$，称之为基矩阵。约束条件写为 $\mathbf{b}=\mathbf{Ax}=\begin{bmatrix}\mathbf{A}_B & \mathbf{A}_N\end{bmatrix}\begin{bmatrix}\mathbf{x}_B\\ \mathbf{x}_N\end{bmatrix}$。$\mathbf{x}_B\in\mathbb{R}^m$ 是从 $\mathbf{x}$ 中选择的相应变量所组成的列向量。如果 $\mathbf{x}_B\geq\mathbf{0}$，则 $\mathbf{A}_B$ 被称为可行基矩阵，$\mathbf{x}_B$ 被称为可行基或基变量。$\mathbf{x}_N\in\mathbb{R}^{n-m}$ 是代表 $\mathbf{x}$ 的其余变量所组成的列向量，被称为非可行基或非基变量。对应的，我们可以把目标函数也记为 $z=\mathbf{c}_B^T\mathbf{x}_B+\mathbf{c}_N^T\mathbf{x}_N$，$\mathbf{c}_B\in\mathbb{R}^m$ 是可行基对应的价值向量，$\mathbf{c}_N\in\mathbb{R}^{n-m}$ 是非可行基对应的价值向量。

特殊的，一个基本可行解 $\mathbf{x}$ 可写为 $\mathbf{x}=\begin{bmatrix}\mathbf{x}_B\\ 0\end{bmatrix}$，此时 $\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}$，目标函数值为 $\mathbf{z}=\mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{b}$。

我们可以进一步建立线性规划问题的基本可行解和对应可行域的极点之间的关系，得到如下几个重要的定理，将线性规划的几何解释和代数解释联系起来，为后续的单纯型法分析提供理论基础。

定理 2.3. 标准形式线性规划问题（2.3）的一个基本可行解必然对应可行域的一个极点。

证明：下面使用反证法来证明。

假设 $\mathbf{x}$ 是一个基本可行解。不失一般性，我们假设 $\mathbf{x}$ 可以写为 $\mathbf{x}=\begin{bmatrix}\mathbf{x}_B\\ 0\end{bmatrix}$，$\mathbf{b}=\mathbf{Ax}=\begin{bmatrix}\mathbf{A}_B & \mathbf{A}_N\end{bmatrix}\begin{bmatrix}\mathbf{x}_B\\ 0\end{bmatrix}$，$\mathbf{A}_B$ 为对应的可逆基矩阵。

若 $\mathbf{x}$ 不是一个极点，则存在两个不同的可行解 $\mathbf{x}_1=\begin{bmatrix}\mathbf{u}_1\\ \mathbf{v}_1\end{bmatrix},\mathbf{x}_2=\begin{bmatrix}\mathbf{u}_2\\ \mathbf{v}_2\end{bmatrix}$ 以及实数 $\lambda\in(0,1)$ 使得 $\mathbf{x}=\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$，其中 $\mathbf{u}_1,\mathbf{u}_2\in\mathbb{R}^m$ 和 $\mathbf{v}_1,\mathbf{v}_2\in\mathbb{R}^{n-m}$ 分别是 m 维和 $n-m$ 维列向量。

显然，此时有$\mathbf{0}=\lambda\mathbf{v}_1+(1-\lambda)\mathbf{v}_2$，由于$\mathbf{x}_1,\mathbf{x}_2$是可行解，$\mathbf{v}_1,\mathbf{v}_2\geq\mathbf{0}$，所以只能是$\mathbf{v}_1=\mathbf{v}_2=\mathbf{0}$。故

$$\mathbf{b}=\mathbf{A}\mathbf{x}=\begin{bmatrix}\mathbf{A}_B & \mathbf{A}_N\end{bmatrix}\begin{bmatrix}\mathbf{x}_B\\0\end{bmatrix}=\mathbf{A}_B x_B=\begin{bmatrix}\mathbf{A}_B & \mathbf{A}_N\end{bmatrix}\begin{bmatrix}\mathbf{u}_1\\0\end{bmatrix}=\begin{bmatrix}\mathbf{A}_B & \mathbf{A}_N\end{bmatrix}\begin{bmatrix}\mathbf{u}_2\\0\end{bmatrix}$$

由于$\mathbf{A}_B$可逆，有$\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}=\mathbf{u}_1=\mathbf{u}_2$，则$\mathbf{x}_1=\mathbf{x}_2=\mathbf{x}$，推出矛盾。故原命题成立。□

定理 2.4. 标准形式线性规划问题（2.3）可行域的一个极点必然对应一个基本可行解。

证明：假设$\mathbf{x}$是一个极点，有p个正元素。为了方便起见，假设$\mathbf{x}$的前p个元素是正的，令矩阵$\mathbf{A}$的各列向量依次表示为$\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_n$，则有

$$\mathbf{x}=\left[x_1,...,x_p,0,...,0\right]^T,\ \sum_{j=1}^{p}x_j\boldsymbol{\alpha}_j=\mathbf{b}$$

我们使用反证法证明$\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_p$是线性独立的。假设这些列向量线性相关，则存在一组不全为0的实数$\{\lambda_j,j=1,...,p\}$使得$\sum_{j=1}^{p}\lambda_j\boldsymbol{\alpha}_j=\mathbf{0}$。进一步取$\varepsilon\in\left(0,\min_{\lambda_j\neq0}\frac{x_j}{|\lambda_j|}\right)$，我们有$x_j\pm\varepsilon\lambda_j>0$，$j=1,...,p$。

记$\mathbf{y}=\left[\lambda_1,...,\lambda_p,0,...,0\right]^T$，我们可以得到两个新的可行解$\mathbf{x}_1=\mathbf{x}+\varepsilon\mathbf{y}\geq\mathbf{0}$，$\mathbf{x}_2=\mathbf{x}-\varepsilon\mathbf{y}\geq\mathbf{0}$，$\mathbf{A}\mathbf{x}_1=\mathbf{A}\mathbf{x}+\varepsilon\mathbf{A}\mathbf{y}=\mathbf{b}+\mathbf{0}=\mathbf{b}$，$\mathbf{A}\mathbf{x}_2=\mathbf{A}\mathbf{x}-\varepsilon\mathbf{A}\mathbf{y}=\mathbf{b}-\mathbf{0}=\mathbf{b}$。注意到$\mathbf{x}=\frac{1}{2}\mathbf{x}_1+\frac{1}{2}\mathbf{x}_2$。这与我们假设的$\mathbf{x}$是一个极点矛盾。

因此，$\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_p$是线性独立的，$\mathbf{x}$是一个基本可行解。□

定理 2.5. [线性规划基本定理] 如果标准形式线性规划问题（2.3）存在可行解，则必然存在基本可行解。

如果标准形式线性规划问题（2.3）存在有界最优解，则必然存在一个基本可行解处取得最优值。

证明：不妨设矩阵$\mathbf{A}$是行满秩的，不然的话，我们可以消除多余的约束再进一步讨论。同时，我们设增广矩阵$\mathbf{A}_\mathbf{b}=\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}$和矩阵A的秩相当，否则线性规划问题没有可行解。若$m=n=\mathrm{rank}(\mathbf{A})$，则该问题有唯一解，且这个解必然是基本可行解。我们下面几种讨论$m<n$的情况。

如果$\mathbf{b}=\mathbf{0}$，则$\mathbf{x}=\mathbf{0}$显然是一个没有正元素的基本可行解。以下讨论$\mathbf{b}\neq\mathbf{0}$的情况。假设$\mathbf{x}$是一个可行解，有p个正元素。为了方便起见，假设$\mathbf{x}$的前p个

元素是正的，令矩阵 $\mathbf{A}$ 的各列向量依次表示为 $\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_n$，则有

$$\mathbf{x}=\left[x_1,...,x_p,0,...,0\right]^T,\ \sum_{j=1}^{p}x_j\boldsymbol{\alpha}_j=\mathbf{b}$$

接下来我们分两种情况讨论。

第一种情况，如果 $\left\{\boldsymbol{\alpha}_j,j=1,...,p\right\}$ 是线性独立的一组向量，则我们有 $p\le\operatorname{rank}(\mathbf{A})=m$。如果进一步 $p=m$，则根据定义，$\mathbf{x}$ 是一个非退化的基本可行解。如果 $p<m$，那么一定还存在 $m-p$ 个向量 $\left\{\boldsymbol{\alpha}_j,j=p+1,...,m\right\}$ 使得 $\left\{\boldsymbol{\alpha}_j,j=1,...,m\right\}$ 构成线性独立的一组向量。由于 $\mathbf{x}$ 的第 $p+1$ 个到第 m 个元素都是 0，因此有

$$\sum_{j=1}^{m}x_j\boldsymbol{\alpha}_j=\sum_{j=1}^{p}x_j\boldsymbol{\alpha}_j=\mathbf{b}$$

那么根据定义，$\mathbf{x}$ 是一个退化的基本可行解。

第二种情况，如果 $\left\{\boldsymbol{\alpha}_j,j=1,...,p\right\}$ 是线性相关的一组向量，则存在一组不全为 0 的实数 $\left\{\lambda_j,j=1,...,p\right\}$ 使得 $\sum_{j=1}^{p}\lambda_j\boldsymbol{\alpha}_j=\mathbf{0}$。不妨设这组实数 $\left\{\lambda_j,j=1,...,p\right\}$ 中一定有正数。因为如果 λ_j 全部非正，则一组全部非负的实数 $\left\{-\lambda_j,j=1,...,p\right\}$ 同样可以满足 $\sum_{j=1}^{p}\left(-\lambda_j\right)\boldsymbol{\alpha}_j=\mathbf{0}$，那么我们考虑 $\left\{-\lambda_j,j=1,...,p\right\}$ 也是一样的。

对于任意一个 $\lambda_r>0$，则可以有 $\lambda_r=-\sum_{j=1,j\ne r}^{p}\frac{\lambda_j}{\lambda_r}\boldsymbol{\alpha}_j$。将此式代入约束条件，可得 $\sum_{j=1,j\ne r}^{p}\left(\mathbf{x}_j-\mathbf{x}_r\frac{\lambda_j}{\lambda_r}\right)\boldsymbol{\alpha}_j=\mathbf{b}$。由此，我们可以得到一个新的向量

$$\hat{\mathbf{x}}=\left[x_1-x_r\frac{\lambda_1}{\lambda_r},...,x_{r-1}-x_r\frac{\lambda_{r-1}}{\lambda_r},0,x_{r+1}-x_r\frac{\lambda_{r+1}}{\lambda_r},...,x_p-x_p\frac{\lambda_p}{\lambda_r},0,...,0\right]^T$$

显然，这个 $\hat{\mathbf{x}}$ 中非零元素的个数不超过 $p-1$。但任意选择 $\lambda_r>0$ 进行上述操作不能保证 $\hat{\mathbf{x}}$ 中的每个元素都非负。

因此，我们可以选择 $r=\arg\min_j\left\{\frac{x_j}{\lambda_j}:\lambda_j>0,\ j=1,...,m\right\}$，保证 $\hat{\mathbf{x}}$ 中的每个元素都非负。因为，如此选择的话，若 $\lambda_j>0$，上述取最小的操作保证 $\frac{x_j}{\lambda_j}-\frac{x_r}{\lambda_r}\geq 0$，也即 $x_j-x_r\frac{\lambda_j}{\lambda_r}\geq 0$，$j=1,...,p$ 成立；若 $\lambda_j<0$，则 $\frac{x_j}{\lambda_j}-\frac{x_r}{\lambda_r}\leq 0$ 自然成立，同样有 $x_j-x_r\frac{\lambda_j}{\lambda_r}\geq 0$，$j=1,...,p$。

如果 $\hat{\mathbf{x}}$ 的正元素对应的矩阵 $\mathbf{A}$ 的列向量是线性独立的，那么如前所述，这个 $\hat{\mathbf{x}}$ 是一个基本可行解。不然的话，我们可以重复上述过程，得到一个另一个可行解，其正元素的个数不超过 $p-2$。由于 p 是有限的，上述过程最多可能执行 $p-1$ 次，最终我们得到一个仅有一个非零元素的可行解，其对应的 A 的列自然是线性独立的。因此，执行上述过程一定可以找到一个基本可行解。

进一步，根据定理 2.2，标准形式线性规划问题（2.3）存在有界最优解，则一定在对应的可行区域多面体的一个极点上取得最优解。根据定理 2.4，这个极点一定对应一个基本可行解。□

定理 2.6. 多面体 $P=\left\{\mathbf{x}\,|\,\mathbf{Ax}\geq\mathbf{b},\mathbf{x}\in\mathbb{R}_+^n,\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\right\}$ 仅有有限个极点和/或极线。

证明：根据定义 1.9，每一条极线由 $n-1$ 个线性独立的约束条件确定，则我们最多有 $C_m^{n-1}=\frac{m!}{(n-1)!(\mathrm{m}-n+1)!}$ 条极线。

而根据转换公式（2.8），我们可以通过增加 m 个新的决策变量，将上述多面体的 m 个线性不等式约束条件等价转换为含有 $n+m$ 个变量的 m 个线性等式约束条件。根据定理 2.4，每个多面体的极点对应这 m 个线性等式的至少一个基本可行解。因此，多面体的极点最多有 $C_{n+m}^n=\frac{(n+m)!}{n!m!}$ 个。□

标准形式线性规划问题（2.3）的一个基本可行解，一定对应对应的可行区域多面体的一个极点。但是可行区域多面体的一个极点可能对应多个基本可行解；这多个基本可行解一定对应退化基本可行解。

2.5　线性规划的对偶理论

每一个线性规划问题（Primal Problem）都有一个伴生的线性规划问题，我们称之为对偶问题（Dual Problem）。这两个线性规划问题存在密切的联系，对于分析和求解线性规划问题有着重要的意义。

定义 2.5. 标准形式线性规划问题（2.3）的对偶问题写作

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{b}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u} \le \mathbf{c} \end{aligned} \tag{2.9}$$

其中向量 $\mathbf{u}\in\mathbb{R}^m$，$\mathbf{b}\in\mathbb{R}^m$，$\mathbf{A}\in\mathbb{R}^{m\times n}$，$\mathbf{c}\in\mathbb{R}^n$。

规范形式线性规划问题（2.4）的对偶问题写作

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{b}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u} \le \mathbf{c} \\ & \mathbf{u} \ge \mathbf{0} \end{aligned} \tag{2.10}$$

其中向量 $\mathbf{u}\in\mathbb{R}^m$ 也被成为对偶变量（Dual Variables），$\mathbf{b}\in\mathbb{R}^m$，$\mathbf{A}\in\mathbb{R}^{m\times n}$，$\mathbf{c}\in\mathbb{R}^n$。

其它形式的线性规划问题可以先化为标准形式线性或者规范形式规划问题，根据上述定义推出对偶问题，再进行化简。或者根据如下的对应关系进行推导

1） 每个对偶变量对应原问题的一个约束
2） 原问题是等式约束则对偶变量是自由变量（无非负约束）
3） 原问题是不等式约束则对偶变量有非负约束
4） 原问题变量和对偶问题的约束同样符合上述规律

上述关系也可以总结为对应表。

表 2.1. 线性规划原问题和对偶问题的对应关系

原问题		对偶问题	
$\min_{\mathbf{x}}\ \mathbf{c}^T\mathbf{x}$		$\max_{\mathbf{u}}\ \mathbf{b}^T\mathbf{u}$	
s.t. $a_{i1}x_1+\ldots+a_{in}x_n=b_i$	$i=1,\ldots,p$	s.t. u_i是自由变量	
$a_{i1}x_1+\ldots+a_{in}x_n\ge b_i$	$i=p+1,\ldots,m$	$u_i\ge 0$	
$x_j\ge 0$	$j=1,\ldots,q$	$\boldsymbol{\alpha}_j^T\mathbf{u}\le c_i$	
x_j是自由变量	$j=q+1,\ldots,n$	$\boldsymbol{\alpha}_j^T\mathbf{u}=c_i$	

其中矩阵 $\mathbf{A}$ 的各列向量依次表示为 $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n$。

例 2.4. 根据规范形式线性规划问题（2.3）的对偶问题定义推导标准形式线性规划问题（2.4）的对偶问题。

解：先将标准形式线性规划问题（2.4）改写为规范形式线性规划问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & -\mathbf{A}\mathbf{x} \geq -\mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

则根据定义 2.5，其对应的对偶问题为

$$\begin{aligned} \max_{\mathbf{u}_1,\mathbf{u}_2} \quad & \mathbf{b}^T\mathbf{u}_1 - \mathbf{b}^T\mathbf{u}_2 \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u}_1 - \mathbf{A}^T\mathbf{u}_2 \leq \mathbf{c} \\ & \mathbf{u}_1, \mathbf{u}_2 \geq \mathbf{0} \end{aligned}$$

令 $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2$，化简可得线性规划问题（2.9）。

定理 2.7. 一个线性规划问题的对偶问题的对偶问题是原问题。

证明：我们仅对规范形式线性规划问题（2.4）及其对偶问题（2.10），其它形式的结论可转化为规范形式之后类似得到。

先将对偶问题（2.10）写成规范形式

$$\begin{aligned} \min_{\mathbf{u}} \quad & -\mathbf{b}^T\mathbf{u} \\ \text{s.t.} \quad & -\mathbf{A}^T\mathbf{u} \geq -\mathbf{c} \\ & \mathbf{u} \geq \mathbf{0} \end{aligned}$$

根据定义 2.5，其对应的对偶问题为

$$\begin{aligned} \max_{\mathbf{x}} \quad & -\mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & -\mathbf{A}\mathbf{x} \leq -\mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

而这显然就是规范形式线性规划问题（2.4）。□

线性规划原问题和对偶问题之间的联系由下面三个重要的定理给出。虽然我们下面以规范形式线性规划问题进行讨论，但对于标准形式线性规划问题这些定理的结论不变。

定理 2.8. [弱对偶性定理（Weak Duality Theorem）] 如果 $\mathbf{x} \in \mathbb{R}^n$ 是规范形式线性规划问题（2.4）的一个可行解，$\mathbf{u} \in \mathbb{R}^m$ 是其对偶问题（2.10）的一个可行解，

则

$$\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{u} \tag{2.11}$$

证明：$\mathbf{x} \in \mathbb{R}^n$ 是规范形式线性规划问题（2.4）的一个可行解，则因为其满足约束条件，有 $\mathbf{Ax} \geq \mathbf{b}$。故对于任何 $\mathbf{u} \geq \mathbf{0}$，我们有

$$\mathbf{u}^T\mathbf{Ax} = \mathbf{u}^T\left(\mathbf{Ax}\right) \geq \mathbf{u}^T\mathbf{b} = \mathbf{b}^T\mathbf{u}$$

$\mathbf{u} \in \mathbb{R}^m$ 是其对偶问题（2.10）的一个可行解，则因为其满足约束条件，有 $\mathbf{A}^T\mathbf{u} \leq \mathbf{c}$。故对于任何 $\mathbf{x} \geq \mathbf{0}$，我们有

$$\mathbf{x}^T\mathbf{Au} = \mathbf{x}^T\left(\mathbf{Au}\right) \leq \mathbf{x}^T\mathbf{c} = \mathbf{c}^T\mathbf{x}$$

结合上述两个不等式有 $\mathbf{c}^T\mathbf{x} \geq \mathbf{x}^T\mathbf{Au} = \mathbf{u}^T\mathbf{Ax} \geq \mathbf{b}^T\mathbf{u}$。□

推论 2.1. 如果规范形式线性规划问题（2.4）的一个可行解 $\mathbf{x} \in \mathbb{R}^n$ 和对偶问题（2.10）的一个可行解 $\mathbf{u} \in \mathbb{R}^m$，满足 $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{u}$，则 $\mathbf{x}$ 是原问题的最优解，$\mathbf{u}$ 是对偶问题的最优解。

定理 2.9. [强对偶性定理（Strong Duality Theorem）] 如果规范形式线性规划问题（2.4）存在有界的最优解 $\mathbf{x}^* \in \mathbb{R}^n$，当且仅当对偶问题（2.10）也存在有界的最优解 $\mathbf{u}^* \in \mathbb{R}^m$，且满足 $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{u}^*$。反之亦然。

证明：充分条件也就是推论 2.1 很容易证明。根据定理 2.8，对于规范形式线性规划问题（2.4）的任意一个可行解 $\mathbf{x} \in \mathbb{R}^n$，总有 $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{u}^* = \mathbf{c}^T\mathbf{x}^*$，因此 $\mathbf{x}^*$ 必然是原问题的最优解。类似的，对于对偶问题的任意一个可行解 $\mathbf{u} \in \mathbb{R}^m$，总有 $\mathbf{b}^T\mathbf{u}^* = \mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{u}$，因此 $\mathbf{u}^*$ 必然是对偶问题的最优解。

接下来用 Farkas 引理和构造法证明必要性。假设原问题存在有界的最优解 $\mathbf{x}^* \in \mathbb{R}^n$，我们需要找到对应的 $\mathbf{u}^* \in \mathbb{R}^m$，满足对偶问题的最优解。假设 $\mathbf{x}^*$ 使得约束条件 $\mathbf{Ax} \geq \mathbf{b}$ 中取等号那些约束的下标们构成集合 I，则我们有

$$\mathbf{a}_i^T\mathbf{x}^* = b_i, i \in I，\text{以及}\ \mathbf{a}_i^T\mathbf{x}^* > b_i, i \notin I$$

$\mathbf{x}^*$ 是最优解，则集合 $\left\{\mathbf{d} \mid \mathbf{a}_i^T\mathbf{d} \geq 0, \mathbf{c}^T\mathbf{d} < 0, i \in I\right\}$ 必为空集。否则，总可以找到一个足够小的 $\varepsilon > 0$ 使得 $\mathbf{a}_i^T\left(\mathbf{x}^* + \varepsilon\mathbf{d}\right) \geq b_i$，$\mathbf{c}^T\left(\mathbf{x}^* + \varepsilon\mathbf{d}\right) < \mathbf{c}^T\mathbf{x}^*$，$i = 1, \ldots, m$。

根据定理 1.9 所述的 Farkas 引理，我们知道集合 $\left\{\mathbf{d} \mid \mathbf{a}_i^T\mathbf{d} \geq 0, \mathbf{c}^T\mathbf{d} < 0, i \in I\right\}$ 为空集，则必然存在一组实数 λ_i，$i \in I$ 满足

$$\lambda_i \geq 0，\ \sum_{i \in I} \lambda_i \mathbf{a}_i^T = \mathbf{c}，$$

则我们可以根据

$$\mathbf{u}_i^* = \lambda_i, i \in I，\text{以及}\ \mathbf{u}_i^* = 0, i \notin I$$

来构造对偶问题的可行解 $\mathbf{u}^* \in \mathbb{R}^m$。此时成立

$$\mathbf{b}^T\mathbf{u}_i^* = \sum_{i\in I} b_i\mathbf{u}_i = \sum_{i\in I}\left(\mathbf{a}_i^T\mathbf{x}_i^*\right)\mathbf{u}_i = \mathbf{u}^*\mathbf{A}\mathbf{x}^* = \mathbf{c}^T\mathbf{x}^*$$

则根据弱对偶性定理 2.8，$\mathbf{u}^*$ 是对偶问题的最优解，所以对于原问题，强对偶性得证。类似的可以证明对于对偶问题亦然。□

定理 2.10. [互补松弛性定理（Complementary Slackness）] 如果规范形式线性规划问题（2.4）的一个可行解 $\mathbf{x}\in\mathbb{R}^n$ 和对偶问题（2.10）的一个可行解 $\mathbf{u}\in\mathbb{R}^m$，则 $\mathbf{x}$ 是原问题的最优解且 $\mathbf{u}$ 是对偶问题的最优解的充分必要条件是同时满足

$$u_i\left(\mathbf{a}_i^T\mathbf{x}-b_i\right)=0 \text{ 和 } \left(c_j-\mathbf{u}^T\alpha_j\right)x_j=0 \qquad (2.12)$$

其中 $\left\{\alpha_j\in\mathbb{R}^m, j=1,...,n\right\}$ 是矩阵 $\mathbf{A}$ 的各列向量。

证明：我们首先证明充分性。如果（2.12）成立，则有 $\left(\mathbf{u}^T\mathbf{A}-\mathbf{c}^T\right)\mathbf{x}=0$，也即 $\mathbf{u}^T\mathbf{A}\mathbf{x}=\mathbf{u}^T\mathbf{b}=\mathbf{c}^T\mathbf{x}$。则根据强对偶性定理，$\mathbf{x}$ 是原问题的最优解且 $\mathbf{u}$ 是对偶问题的最优解。

我们接下来证明必要性。如果 $\mathbf{x}$ 是原问题的最优解且 $\mathbf{u}$ 是对偶问题的最优解，则根据强对偶性定理，我们得到 $\left(\mathbf{u}^T\mathbf{A}-\mathbf{c}^T\right)\mathbf{x}=0$ 且 $\mathbf{u}^T\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)=0$。考虑到 $\mathbf{x}\geq\mathbf{0}$ 且 $\mathbf{u}^T\mathbf{A}-\mathbf{c}^T\geq\mathbf{0}$，$\mathbf{u}\geq\mathbf{0}$ 且 $\mathbf{A}\mathbf{x}-\mathbf{b}\geq\mathbf{0}$，则（2.12）必然成立。□

根据上述上个定理，我们可以进一步分析线性规划原问题和对偶问题有解性之间的联系。这些联系将在我们的后续分析证明中反复使用。

定理 2.11. 原问题存在有界的最优解，则对偶问题存在有界的最优解。原问题存在无界解，则对偶问题无解。原问题无解，则对偶问题无解或者存在无界解。

对偶问题存在有界的最优解，则原问题存在有界的最优解。对偶问题存在无界解，则原问题无解。对偶问题无解，则原问题无解或者存在无界的解。

证明：如果原始问题是无界的，这意味着 $\mathbf{c}^T\mathbf{x}$ 可以趋向 $-\infty$，根据弱对偶性定理 2.8，我们有 $\mathbf{b}^T\mathbf{u}\leq\mathbf{c}^T\mathbf{x}$，这是不可能的，这意味着对偶问题一定无解。

如果原始问题存在有界的最优解，根据强对偶定理 2.9，则对偶问题存在有界的最优解。

如果原问题无解，根据弱对偶定理 2.8，对偶问题可能是无界的。我们提出下面的例子来证明对偶问题也可能是无解。考虑原问题

$$\min_{x_1,x_2} \quad \begin{bmatrix}-2 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$
$$\text{s.t.} \quad \begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \geq \begin{bmatrix}1\\-0.5\end{bmatrix}$$
$$x_1, x_2 \geq 0$$

其对偶问题为

$$\max_{u_1,u_2} \quad \begin{bmatrix}1 & -0.5\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix}$$
$$\text{s.t.} \quad \begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix} \leq \begin{bmatrix}-2\\1\end{bmatrix}$$
$$u_1, u_2 \geq 0$$

很显然，这两个线性规划都无解。

对于对偶问题，结论类似可以证明。□

总之，原问题和对偶问题的解的对应关系如下表。

表 2.2. 线性规划原问题和对偶问题的解的对应关系

原始 \ 对偶	存在有界最优解	存在无界解	无解
存在有界最优解	√	×	×
存在无界解	×	×	√
无解	×	√	√

接下来，我们顺势给出线性规划多面体可行域的极点—极线表示法，为第 4 章的算法分析提供理论基础。

定理 2.12. $\mathbf{0}$ 是多面体锥 $\left\{\mathbf{x}\in\mathbb{R}^n \mid \mathbf{A}\mathbf{x}\geq\mathbf{0}\right\}$ 的极点，当且仅当 $\operatorname{rank}(\mathbf{A}) = n$。

证明：首先，我们使用反证法证明必要性。

若 $\mathbf{0}$ 是该多面体锥的极点。$\operatorname{rank}(\mathbf{A}) < n$，必然存在一个点 $\mathbf{y}\neq\mathbf{0}$ 满足 $\mathbf{A}\mathbf{y}=\mathbf{0}$。进一步，点 $-\mathbf{y}\neq\mathbf{0}$ 也满足 $\mathbf{A}(-\mathbf{y})=\mathbf{0}$。所以点 $\mathbf{y}\neq\mathbf{0}$ 和 $-\mathbf{y}\neq\mathbf{0}$ 都在这个多面体锥中，且 $\mathbf{y}+(-\mathbf{y})=\mathbf{0}$，这和 $\mathbf{0}$ 是该多面体锥的极点矛盾。因此，若 $\mathbf{0}$ 是该多面体锥的极点，则 $\operatorname{rank}(\mathbf{A}) = n$。

接着，我们使用反证法证明充分性。

若 $\operatorname{rank}(\mathbf{A}) = n$，$\mathbf{y}=\mathbf{0}$ 是 $\mathbf{A}\mathbf{y}=\mathbf{0}$ 的唯一解。假设存在 $\mathbf{x}_1,\mathbf{x}_2\neq\mathbf{y}$ 以及实数 $\lambda\in(0,1)$ 使得 $\mathbf{0}=\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2$。则

$$\mathbf{A}\left[\lambda\mathbf{x}_1+(1-\lambda)\mathbf{x}_2\right]=\lambda\mathbf{A}\mathbf{x}_1+(1-\lambda)\mathbf{A}\mathbf{x}_2=\mathbf{0}$$

但 $\mathbf{A}\mathbf{x}_1\geq\mathbf{0}$，$\mathbf{A}\mathbf{x}_2\geq\mathbf{0}$，$\lambda\geq 0$，$1-\lambda\geq 0$。必有 $\mathbf{A}\mathbf{x}_1=\mathbf{A}\mathbf{x}_2=\mathbf{0}$，这和 $\mathbf{y}=\mathbf{0}$ 是 $\mathbf{A}\mathbf{y}=\mathbf{0}$ 的唯一解矛盾。因此，若 $\mathrm{rank}(\mathbf{A})=n$，$\mathbf{0}$ 是该多面体锥的极点。□

定理 2.13. 如果线性规划问题 $\left\{\min_{\mathbf{x}}\mathbf{c}^T\mathbf{x}\mid\mathbf{A}\mathbf{x}\geq\mathbf{0}\right\}$ 存在无界解，则锥可行域 $\{\mathbf{x}\mid\mathbf{A}\mathbf{x}\geq\mathbf{0}\}$ 有且必有一条射线 $\mathbf{r}$ 满足 $\mathbf{c}^T\mathbf{r}\leq\mathbf{0}$。

证明：根据定义，上述充分性命题显而易见。我们着重分析必要性。

如果原始问题是无界的，则根据强对偶性定理，它的对偶问题一定是无解的。换句话说，对偶问题的可行域集合 $\left\{\mathbf{u}\in\mathbb{R}^m\mid\mathbf{A}^T\mathbf{u}=\mathbf{c},\mathbf{u}\geq\mathbf{0}\right\}$ 一定为空集。根据定理 1.9 所述的 Farkas 引理，存在向量 $\mathbf{r}$ 使得 $\mathbf{A}\mathbf{r}\geq\mathbf{0}$ 和 $\mathbf{c}^T\mathbf{r}\leq\mathbf{0}$。显然，$\mathbf{r}$ 是满足要求的锥多面体 $\Omega=\left\{\mathbf{x}\in\mathbb{R}^n\mid\mathbf{A}\mathbf{x}\geq\mathbf{0}\right\}$ 的一条射线。□

定理 2.14. 如果线性规划问题 $\left\{\min_{\mathbf{x}}\mathbf{c}^T\mathbf{x}\mid\mathbf{A}\mathbf{x}\geq\mathbf{b}\right\}$ 存在无界解，且其定义域中存在至少一个极点，则多面体可行域 $\left\{\mathbf{x}\in\mathbb{R}^n\mid\mathbf{A}\mathbf{x}\geq\mathbf{b}\right\}$ 有且必有一条极线 $\mathbf{d}$ 满足 $\mathbf{c}^T\mathbf{d}\leq\mathbf{0}$。

证明：根据定义，上述充分性命题显而易见。我们着重分析必要性。

首先，考虑该线性规划问题 $\left\{\min_{\mathbf{x}}\mathbf{c}^T\mathbf{x}\mid\mathbf{A}\mathbf{x}\geq\mathbf{b}\right\}$ 的对偶问题

$$\begin{aligned}\max_{\mathbf{u}}\quad & z=\mathbf{b}^T\mathbf{u}\\ \text{s.t.}\quad & \mathbf{A}^T\mathbf{u}=\mathbf{c},\mathbf{u}\geq\mathbf{0}\end{aligned}$$

如果原问题存在无界解，则对偶问题无解。因此可行域 $\left\{\mathbf{u}\in\mathbb{R}^m\mid\mathbf{A}^T\mathbf{u}=\mathbf{c},\mathbf{u}\geq\mathbf{0}\right\}$ 是空集，因此下述线性规划问题也无解

$$\begin{aligned}\max_{\mathbf{u}}\quad & \mathbf{0}^T\mathbf{u}=0\\ \text{s.t.}\quad & \mathbf{A}^T\mathbf{u}=\mathbf{c},\mathbf{u}\geq\mathbf{0}\end{aligned}$$

而这个新的线性规划问题的对偶问题

$$\begin{aligned}\min_{\mathbf{x}}\quad & z=\mathbf{c}^T\mathbf{x}\\ \text{s.t.}\quad & \mathbf{A}\mathbf{x}\geq\mathbf{0}\end{aligned}$$

要么存在无界解，要么无解。但 $\mathbf{x}=\mathbf{0}$ 显然是该问题的一个平凡解，则线性规划

问题 $\left\{\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\right\}$ 显然存在无界解，根据定理 2.13，锥多面体 $\Omega = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\right\}$ 存在一条射线 $\mathbf{r}$ 满足 $\mathbf{A}\mathbf{r} \geq \mathbf{0}$ 和 $\mathbf{c}^T\mathbf{r} \leq \mathbf{0}$。

接着，考虑如下的线性规划问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1 \end{aligned}$$

显然，约束 $\mathbf{c}^T\mathbf{x} = -1$ 确保该线性规划问题存在有界的最优解。而且，这个最优解显然可以取到。例如，上面证明存在一个向量 $\mathbf{r}$ 满足使得 $\mathbf{A}\mathbf{r} \geq \mathbf{0}$ 和 $\mathbf{c}^T\mathbf{r} \leq \mathbf{0}$。令向量 $\mathbf{x}_1 = -\dfrac{\mathbf{r}}{\mathbf{c}^T\mathbf{r}}$，我们可以看到 $\mathbf{A}\mathbf{x}_1 = -\dfrac{\mathbf{A}\mathbf{r}}{\mathbf{c}^T\mathbf{r}} \geq 0$，$\mathbf{c}^T\mathbf{x}_1 = -\dfrac{\mathbf{c}^T\mathbf{r}}{\mathbf{c}^T\mathbf{r}} = -1$，则线性规划问题 $\left\{\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1\right\}$ 在点 $\mathbf{x}_1$ 处取得最优解 $z = \mathbf{c}^T\mathbf{x}_1 = -1$。

然后，由于多面体可行域 $\left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\right\}$ 存在至少一个极点，根据作业题 1.5 的结论，必有 $\text{rank}(\mathbf{A}) = n$。根据作业题 1.4 的结论，若 $\text{rank}(\mathbf{A}) = n$，多面体 $\Gamma = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1\right\}$ 必然存在极点。根据定理 2.3，线性规划问题 $\left\{\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1\right\}$ 存在有界的最优解，且定义域存在极点，则最优解必然在某一个极点处取得。不妨设在多面体可行域 $\Gamma = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1\right\}$ 的极点 $\mathbf{d}$ 处取得最优解。

如果在点 $\mathbf{d}$ 处存在 n 个线性独立的约束条件取等号 $\mathbf{a}_i^T\mathbf{d} = 0$，$i = 1, \ldots, n$，则点 $\mathbf{d}$ 也会是锥多面体 $\Omega = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\right\}$ 的极点。$\text{rank}(\mathbf{A}) = n$。并且 $\mathbf{d} \neq \mathbf{0}$，因为 $\mathbf{c}^T\mathbf{d} = -1$。而这会与定理 2.12 矛盾。

如果在点 $\mathbf{d}$ 处存在 $n-1$ 个线性独立的约束条件取等号 $\mathbf{a}_i^T\mathbf{d} = 0$，$i = 1, \ldots, n-1$，则根据定义 1.12，点 $\mathbf{d}$ 就是我们寻找的满足原命题要求的极线。

如果在点 $\mathbf{d}$ 处存在少于 $n-1$ 个线性独立的约束条件取等号，则我们可以简单的使用反证法证明点 $\mathbf{d}$ 不可能是多面体 $\Gamma = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}, \mathbf{c}^T\mathbf{x} = -1\right\}$ 的极点。□

定理 2.15. [弱化的 Minkowski 表示定理（Almost Minkowski's Resolution Theorem）] 假设$\Omega=\left\{\mathbf{x}\in\mathbb{R}^n \mid \mathbf{Ax}\geq\mathbf{b}\right\}$表示的多面体非空，且含有至少一个极点。如果多面体$\Omega$的所有极点构成集合$\left\{\mathbf{x}_1,...,\mathbf{x}_k\in\mathbb{R}^n\right\}$，$\Omega$的所有极线构成集合$\left\{\mathbf{d}_1,...,\mathbf{d}_r\in\mathbb{R}^n\right\}$。定义集合

$$\Gamma=\left\{\sum_{i=1}^{k}\lambda_i\mathbf{x}_i+\sum_{j=1}^{r}\theta_j\mathbf{d}_j \mid \lambda_i\geq 0,\theta_j\geq 0,\sum_{i=1}^{k}\lambda_i=1\right\} \tag{2.12}$$

则$\Gamma=\Omega$。

证明：我们首先证明$\Gamma\subseteq\Omega$。对于任意一个$\mathbf{x}=\sum_{i=1}^{k}\lambda_i\mathbf{x}_i+\sum_{j=1}^{r}\theta_j\mathbf{d}_j\in\Gamma$，有

$$\mathbf{A}\sum_{i=1}^{k}\lambda_i\mathbf{x}_i=\sum_{i=1}^{k}\lambda_i\mathbf{A}\mathbf{x}_i\geq\sum_{i=1}^{k}\lambda_i\mathbf{b}=\mathbf{b},\quad \mathbf{A}\sum_{j=1}^{r}\theta_j\mathbf{d}_j=\sum_{j=1}^{r}\theta_j\mathbf{A}\mathbf{d}_j\geq\sum_{j=1}^{r}\theta_j\mathbf{0}=\mathbf{0}$$

因此$\mathbf{Ax}\geq\mathbf{b}$，也即有$\Gamma\subseteq\Omega$。

接着，我们使用反证法证明$\Omega\subseteq\Gamma$。假设存在一个点$\mathbf{y}$使得$\mathbf{y}\in\Omega$同时$\mathbf{y}\notin\Gamma$。注意到多面体是闭集，则根据定理 1.6 所述的凸集分隔定理，必有一个超平面分隔点$\mathbf{y}$和集合Γ。

考虑如下的线性规划问题

$$\begin{aligned}\max_{\boldsymbol{\lambda},\boldsymbol{\theta}}\quad & \mathbf{0}^T\boldsymbol{\lambda}+\mathbf{0}^T\boldsymbol{\theta}\\ \text{s.t.}\quad & \mathbf{y}=\sum_{i=1}^{k}\lambda_i\mathbf{x}_i+\sum_{j=1}^{r}\theta_j\mathbf{d}_j\\ & \boldsymbol{\lambda}^T\mathbf{1}=1\\ & \boldsymbol{\lambda}\geq\mathbf{0},\boldsymbol{\theta}\geq\mathbf{0}\end{aligned} \tag{2.13}$$

其中$\boldsymbol{\lambda}=[\lambda_1,...,\lambda_k]\in\mathbb{R}^k$，$\boldsymbol{\theta}=[\theta_1,...,\theta_k]\in\mathbb{R}^r$。

$\mathbf{y}\notin\Gamma$，则上述线性规划问题无解，则下述对偶问题无解或者存在无解的解。

$$\begin{aligned}\min_{\mathbf{p},q}\quad & \mathbf{p}^T\mathbf{y}+q\\ \text{s.t.}\quad & \mathbf{p}^T\mathbf{x}_i+q\geq 0,i=1,...,k\\ & \mathbf{p}^T\mathbf{d}_j\geq 0,\ j=1,...,\mathrm{r}\end{aligned} \tag{2.14}$$

但显然$\mathbf{p}=\mathbf{0}$，$q=0$是上述对偶问题的平凡解，则该对偶问题存在无解的解。则意味着存在一个解$\mathbf{p}_1$，q_1使得$\mathbf{p}_1^T\mathbf{y}+q_1<0$。同时观察到$\mathbf{p}_1^T\mathbf{x}_i+q_1\geq 0,i=1,...,k$。则有$\mathbf{p}_1^T\mathbf{y}\leq\mathbf{p}_1^T\mathbf{x}_i,i=1,...,k$。

考虑如下一个新的线性规划问题

$$\begin{aligned}\min_{\mathbf{x}}\quad & z=\mathbf{p}_1^T\mathbf{x}\\ \text{s.t.}\quad & \mathbf{Ax}\geq\mathbf{b}\end{aligned}$$

由于我们假设多面体Ω非空，且含有至少一个极点；则该线性规划问题是有解的。

如果这个线性规划问题存在有界的最优解，则根据定理 2.5 表述的线性规划基本定理，该问题一定在一个极点处取得最优解，但$\mathbf{y}\in\Omega$，$\mathbf{y}$显然不是极点，则应该存在$\mathbf{p}_1^T\mathbf{y}>\mathbf{p}_1^T\mathbf{x}_i, i=1,...,k$，这导致矛盾。

如果这个线性规划问题存在无界的解；则根据定理 2.14，存在一条极线$\mathbf{d}_j$使得$\mathbf{p}_1^T\mathbf{d}_j<0$，这和$\mathbf{p}_1$，$q_1$是上述对偶问题（2.14）的解，满足$\mathbf{p}^T\mathbf{d}_j\geq 0, j=1,...,\mathrm{r}$相矛盾。因此，只有可能$\Omega\subseteq\Gamma$。

因为$\Gamma\subseteq\Omega$和$\Omega\subseteq\Gamma$同时成立，则$\Gamma=\Omega$。□

定理 2.16. 如果集合Γ定义为（2.12）形式，则集合Γ为多面体。

证明：如果存在点$\mathbf{y}$满足$\mathbf{y}\in\Gamma$，则必然线性规划问题（2.13）有解。而线性问题（2.13）有解，则最优解的值必为0。根据强对偶性定理，对偶问题（2.14）也有存在有界的最优解，值为0。

我们引入$\mathbf{p}^+,\mathbf{p}^-\geq\mathbf{0}$和$q^+,q^-\geq 0$，将变量$\mathbf{p}$和$q$替换为$\mathbf{p}=\mathbf{p}^+-\mathbf{p}^-$和$q=q^+-q^-$，将对偶问题（2.14）改写成标准形式的线性规划问题，则任意可行解点$\mathbf{y}$对应的目标函数值要大于或者等于最优解的值0，也即

$$\mathbf{p}^T\mathbf{y}+q=\left(\mathbf{p}^+-\mathbf{p}^-\right)^T\mathbf{y}+q^+-q^-\geq 0 \qquad (2.15)$$

根据定理 2.14，对于改写成标准形式的对偶问题（2.14），其可行域的所有极线都必须满足不等式（2.15），而根据定理 2.6，极线的个数是有限的，因此任意可行解$\mathbf{y}$只需要满足有限个形如（2.15）的不等式约束条件；根据多面体的定义，集合Γ为多面体。□

2.6　线性规划问题的单纯型法

2.6.1　单纯型法

秉持着“如果线性规划问题的最优解存在，则一定可以在其可行区域的定点中找到”的基本理念，George Dantzig 于 1947 年提出了一个寻找最优解的思路：先找出可行域的一个极点，据一定规则判断其是否最优；若否，则转换到与之相邻的另一极点，并使目标函数值更优；如此往复，直到找到某最优解为止。本节将会讨论这种“在极点中搜索最优解的算法”——单纯型法。

我们首先给出标准形式线性规划问题（2.3）的单纯型法的一般步骤，然后逐一解释为什么这些步骤不可或缺，并证明单纯型法的收敛性和复杂度。

定义 2.6. 对于标准形式线性规划问题（2.3），给定的可行基矩阵 $\mathbf{A}_B$，非可行基矩阵 $\mathbf{A}_N$，我们定义检验数（Condition）为向量 $\boldsymbol{\sigma}_N = \mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N \in \mathbb{R}^{n-m}$。

值得注意的是，有部分国内教材的检验数定义与国际大多数教材恰好反号，则对应的判断是否达到的准备也恰好反号。本书还是按国际主流方式来定义检验数。

算法 2.1. 单纯型法

Step 1. [确定初始可行解] 如果观察原问题即能找到明显的初始可行基矩阵 $\mathbf{A}_B$ 和对应的可行基 $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b} \geq \mathbf{0}$，则直接进入 Step 2。否则使用大 M 法或者两阶段法找到初始可行基矩阵 $\mathbf{A}_B$ 和对应的可行基 $\mathbf{x}_B = \mathbf{A}_B^{-1}\mathbf{b} \geq \mathbf{0}$，再进入 Step 2。

Step 2. [判断最优性] 注意到标准形式线性规划问题（2.3）的约束条件和目标函数可以写成

$$\begin{cases} \mathbf{A}_B\mathbf{x}_B + \mathbf{A}_N\mathbf{x}_N = \mathbf{b} \\ \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N = z \end{cases} \tag{2.16}$$

或等价的

$$\begin{cases} \mathbf{x}_B + \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N = \mathbf{A}_B^{-1}\mathbf{b} \\ \mathbf{c}_B^T\left(\mathbf{A}_B^{-1}\mathbf{b} - \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N\right) + \mathbf{c}_N^T\mathbf{x}_N = z \end{cases}$$

移项化简为

$$\begin{cases} \mathbf{x}_B + \mathbf{A}_B^{-1}\mathbf{A}_N\mathbf{x}_N = \mathbf{A}_B^{-1}\mathbf{b} \\ \left(\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N\right)\mathbf{x}_N = z - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{b} \end{cases} \tag{2.17}$$

我们根据当前的可行基矩阵 $\mathbf{A}_B$，构建如下的单纯型表（Tableau）

	$\mathbf{x}_B$	$\mathbf{x}_N$	RHS
$\mathbf{x}_B$	$\mathbf{I}$	$\mathbf{A}_B^{-1}\mathbf{A}_N$	$\mathbf{A}_B^{-1}\mathbf{b}$
	$\mathbf{0}$	$\boldsymbol{\sigma}^T$	$z - \mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{b}$

令 $\bar{\mathbf{A}} = \mathbf{A}_B^{-1}\mathbf{A}_N = \left[\bar{\boldsymbol{\alpha}}_{m+1}, ..., \bar{\boldsymbol{\alpha}}_n\right] = \left[\bar{a}_{ij}\right] \in R^{m\times(n-m)}$，$\bar{\mathbf{b}} = \mathbf{A}_B^{-1}\mathbf{b} = \left[\bar{b}_i\right] \in R_+^m$ 来简化表述上表中的对应矩阵分块，其中 $\bar{\boldsymbol{\alpha}}_j$，$j = 1, ..., n-m$，表示矩阵 $\bar{\mathbf{A}}$ 的各个列

向量。$\bar{a}_{ij}$，$i=1,...,m$，$j=m+1,...,n$ 表示矩阵 $\bar{\mathbf{A}}$ 的各个元素。$\bar{b}_i$ 表示向量 $\bar{\mathbf{b}}$ 的各个元素。$\boldsymbol{\sigma}_N=\left[\sigma_{m+1},...,\sigma_n\right]^T=\left[c_{m+1}-\mathbf{c}_B^T\mathbf{A}_B^{-1}\bar{\boldsymbol{\alpha}}_{m+1},...,c_n-\mathbf{c}_B^T\mathbf{A}_B^{-1}\bar{\boldsymbol{\alpha}}_n\right]^T$。我们可以将上面的单纯型表的各列详细表述出来

$\mathbf{x}_B$	x_1	x_2	...	x_{m+1}	...	x_{m+k}	...	RHS
x_1	1	0	...	$\bar{a}_{1,m+1}$	...	$\bar{a}_{1,m+k}$	...	$\bar{b}_1$
x_2	0	1	...	$\bar{a}_{2,m+1}$	...	$\bar{a}_{2,m+k}$	...	$\bar{b}_2$
...	...	...	...	...	...	...	...	...
	0	0	0	σ_{m+1}	...	σ_{m+k}	...	$z-\mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{b}$

如果 $\boldsymbol{\sigma}\geq\mathbf{0}$，则当前解为最优解，停止。

如果检验数向量 $\boldsymbol{\sigma}$ 的第 k 个元素 $\sigma_k<0$，而对应的列向量 $\bar{\boldsymbol{\alpha}}_k\leq\mathbf{0}$，则原问题无界，停止。

如果检验数向量 $\boldsymbol{\sigma}$ 的第 k 个元素 $\boldsymbol{\sigma}_k<0$，而对应的列向量 $\bar{\boldsymbol{\alpha}}_k$ 中至少有一个元素 $\bar{a}_{jk}$ 为正，注意到我们可以将对应的非基变量 x_{m+k} 从 0 变为正来减少目标函数值，则进入 Step 3，通过翻转找到一个新的可行基 $\hat{\mathbf{x}}_B$，使得 $\mathbf{c}_B^T\hat{\mathbf{x}}_B\leq\mathbf{c}_B^T\mathbf{x}_B$。

Step 3. [翻转（Pivots）] 首先确定进基变量，按照 Bland 法则（Bland's Rule）[13]，则选取检验数向量 $\boldsymbol{\sigma}$ 的所有非正元素中下标最小的元素对应的决策变量作为进基变量，由非基变量变为基变量。

$$k=\min\left\{j\,|\,\sigma_j<0,j=m+1,...,n\right\}\tag{2.18}$$

此时 $\mathbf{x}_N$ 中的第 k 个元素 $\mathbf{x}_{m+k}$ 即为选定的进基变量。

接着计算求解下述问题，确定可能的出基变量的下标所组成的集合

$$L=\left\{\underset{i}{\operatorname{arcmin}}\left\{\frac{\bar{b}_i}{\bar{a}_{ik}}\middle|\,\bar{a}_{ik}>0,i=1,...,m\right\}\right\}\tag{2.19}$$

如果有多个 $\bar{b}_i/\bar{a}_{ik}$ 同时取得最小值，则同样按照 Bland 法则，选取这其中下标最小的 i 作为出基变量下标 l

$$l=\min\left\{i\,|\,i\in L,i=1,...,m\right\}\tag{2.20}$$

则出基变量为此时可行基 $\mathbf{x}_B$ 中的第 l 个元素 x_l。

根据原有可行基 $\mathbf{x}_B$，进基变量 $\mathbf{x}_{m+k}$ 和出基变量 $\mathbf{x}_l$ 构造新的可行基矩阵 $\hat{\mathbf{A}}_B$，进而求解除新的可行基本解 $\hat{\mathbf{x}}_B=\hat{\mathbf{A}}_B^{-1}\mathbf{b}\geq\mathbf{0}$。这一步可以直接在表中，由进基变量所在列和出基变量所在行决定的约束矩阵系数通过初等变换得到，参见例 2.5 的求解过程。最后回到 Step 2 进行下一轮迭代。

如果我们使用单纯型法求解线性规划问题$\left\{\max_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \mid \mathbf{Ax}=\mathbf{b}, \mathbf{x}\geq \mathbf{0}\right\}$，则有两种求解选择。第一种选择是转化为标准形式线性规划（2.3），也即求解$\left\{\min_{\mathbf{x}} -\mathbf{c}^T\mathbf{x} \mid \mathbf{Ax}=\mathbf{b}, \mathbf{x}\geq \mathbf{0}\right\}$。第二种选择是在 Step 2 根据如下规则来直接确定问题的翻转方向：

如果$\boldsymbol{\sigma}\leq \mathbf{0}$，则当前解为最优解，停止。

如果检验数向量$\boldsymbol{\sigma}$的第k个元素$\sigma_k>0$，而对应的列向量$\bar{\boldsymbol{\alpha}}_k\leq \mathbf{0}$，则原问题无界，停止。

如果检验数向量$\boldsymbol{\sigma}$的第k个元素$\boldsymbol{\sigma}_k>0$，而对应的列向量$\bar{\boldsymbol{\alpha}}_k$中至少有一个元素$\bar{a}_{jk}$为正，注意到我们可以将对应的非基变量$x_{m+k}$从 0 变为正来增加目标函数值，则进入 Step 3，通过翻转找到一个新的可行基$\hat{\mathbf{x}}_{\boldsymbol{B}}$，使得$\mathbf{c}_B^T\hat{\mathbf{x}}_B\geq \mathbf{c}_B^T\mathbf{x}_B$。

接着，我们依次证明几个定理，解释上述单纯型算法几个关键点的必要性和可行性。

定理 2.17. 如果单纯型法迭代到某一步，满足$\boldsymbol{\sigma}\geq \mathbf{0}$，则当前解为最优解。

证明：若满足$\boldsymbol{\sigma}\geq \mathbf{0}$的解是$\mathbf{x}$，对任意可行解$\mathbf{y}$有$\mathbf{Ax}-\mathbf{Ay}=\mathbf{b}-\mathbf{b}=\mathbf{0}$。

我们定义$\mathbf{d}=\mathbf{y}-\mathbf{x}=\begin{bmatrix}\mathbf{d}_B\\ \mathbf{d}_N\end{bmatrix}$，则上式改写为$\mathbf{Ad}=\mathbf{0}$。或者进一步考虑对应的可行基矩阵$\mathbf{A}_B$，写为$\mathbf{A}_B\mathbf{d}_B+\sum_{i\in N}\boldsymbol{\alpha}_i d_i=\mathbf{0}$，其中$N$表示所有非基变量的下标构成的集合，$\{\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_n\}$表示矩阵$\mathbf{A}$的各个列向量。

因矩阵$\mathbf{A}_B$可逆，有$\mathbf{d}_B=-\sum_{i\in N}\mathbf{A}_B\boldsymbol{\alpha}_i d_i$和$\mathbf{c}^T\mathbf{d}=\mathbf{c}_B^T\mathbf{d}_B+\mathbf{c}_N^T\mathbf{d}_N=\sum_{i\in N}\sigma_i d_i$。因为对于非基变量$x_i=0$，对于可行解$\mathbf{y}$有$y_i\geq 0$。因此$d_i\geq 0$。同时根据假设，$\boldsymbol{\sigma}\geq \mathbf{0}$。所以$\mathbf{c}^T\mathbf{d}\geq 0$。故而对于任意可行解$\mathbf{y}$，$\mathbf{c}^T\mathbf{y}\geq \mathbf{c}^T\mathbf{x}$。□

与之相关，我们还可以得到其它一些有用的推论，这里作为作业题 2.3-2.4 留给大家练习。

定理 2.18. 如果单纯型法迭代到某一步，满足检验数向量 $\boldsymbol{\sigma}$ 的第 k 个元素 $\boldsymbol{\sigma}_k < 0$，且对应的列向量 $\bar{\boldsymbol{\alpha}}_k \le \mathbf{0}$，则原问题无界，且单纯型法停止时，我们可以直接构造出标准形式线性规划问题（2.3）可行域的一条极线。

证明：停止时可行基矩阵为 $\mathbf{A}_B$，矩阵 $\bar{\mathbf{A}} = \mathbf{A}_B^{-1}\mathbf{A}_N = [\bar{\boldsymbol{\alpha}}_1, ..., \bar{\boldsymbol{\alpha}}_{n-m}] \in R^{m\times(n-m)}$ 的第 k 列 $\bar{\boldsymbol{\alpha}}_k \le \mathbf{0}$，则我们可以构造如下的方向 $\mathbf{d} = [\mathbf{d}_B, \mathbf{d}_N] \in R^n$，其中 $\mathbf{d}_B = -\bar{\boldsymbol{\alpha}}_k \in R^m$，而除了 $d_{m+k} = 1$，$\mathbf{d}_N \in R^{n-m}$ 的其它各元素都等于 0 。

我们把标准形式线性规划问题（2.3）的可行域 $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \ge \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$ 等价变换为 $\Omega = \{\mathbf{x} \in \mathbb{R}^n \mid [\mathbf{A} \quad -\mathbf{A} \quad \mathbf{I}]\mathbf{x} \ge [\mathbf{b} \quad -\mathbf{b} \quad \mathbf{0}]^{\mathrm{T}}\}$。则该可行域对应的回归锥为 $\Gamma = \{\mathbf{x} \in \mathbb{R}^n \mid [\mathbf{A} \quad -\mathbf{A} \quad \mathbf{I}]\mathbf{x} \ge \mathbf{0}\}$。

不难验证 $\mathbf{d} \ge \mathbf{0}$，$\mathbf{Ad} = \mathbf{0}$，因此 $\mathbf{d}$ 属于标准形式线性规划问题（2.3）可行域的回归锥 Γ。同时，根据 $\boldsymbol{\sigma}_k < 0$，不难验证 $\mathbf{c}^T\mathbf{d} \le \mathbf{0}$。

最后，因为 $\mathbf{A}_B$ 可逆，我们一定可以挑出 $\mathbf{Ad} = \mathbf{0}$ 中对应可行基矩阵 $\mathbf{A}_B$ 的 m 个线性独立的等式约束。再加上 $d_{m+j} = 0$，$j = 1, ..., n-m$，$j \ne k$，这 $n - m + 1$ 个线性独立的等式约束。我们确认 $\mathbf{d}$ 恰好满足 $n - 1$ 个线性独立的等式约束。□

定理 2.19. 算法 2.1 因为采用 Bland 法则，不会在退化的基本可行解处产生迭代循环（Cycle），一定会在有限步内收敛。如果放弃 Bland 法则，则可能出现循环 [11]-[14]。

证明：显然，下标越大的决策变量元素越难被选中进基。对于决策变量 $\mathbf{x} = [x_1, ..., x_n]^T \in \mathbb{R}^n$，$x_n$ 最多可能被选中 1 次进基。我们容易递归证明，x_{n-i} 只可能被选中 2^i 次进基，$i = 1, ..., n$。因此，采用 Bland 法则保证会在有限步内收敛。而翻转次数最多的线性规划问题例子可以参见定理 2.21。

如果放弃 Bland 法则，Hoffman，Beale 等研究者构造了一些有趣的例子来证实会发生循环。以下的例子来自[12]，考虑如下的线性规划问题

$$\begin{aligned}
\max_{x_1,x_2,\ldots,x_7} \quad & \frac{3}{4}x_4 - 20x_5 + \frac{1}{2}x_6 - 6x_7 \\
\text{s.t.} \quad & x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0 \\
& x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0 \\
& x_3 + x_6 = 1 \\
& x_i \geq 0, i = 1,2,\cdots,7
\end{aligned}$$

假设我们选择的初始基变量是$\{x_1, x_2, x_3\}$，则得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_1	1	0	0	1/4	–8	–1	9	0
x_2	0	1	0	1/2	–12	–1/2	3	0
x_3	0	0	1	0	0	1	0	1
	0	0	0	3/4	–20	1/2	–6	$z-0$

第一次选择x_4作为进基变量，x_1作为出基变量，进行翻转，基变量变为$\{x_4, x_2, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_4	4	0	0	1	–32	–4	36	0
x_2	–2	1	0	0	4	3/2	–15	0
x_3	0	0	1	0	0	1	0	1
	–3	0	0	0	4	7/2	–33	$z-0$

第二次选择x_5作为进基变量，x_2作为出基变量，进行翻转，基变量变为$\{x_4, x_5, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_4	–12	8	0	1	0	8	–84	0
x_5	–1/2	1/4	0	0	1	3/8	–15/4	0
x_3	0	0	1	0	0	1	0	1
	–1	–1	0	0	0	2	–18	$z-0$

第三次选择x_6作为进基变量，x_4作为出基变量，进行翻转，基变量变为$\{x_6, x_5, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_6	$-2/3$	1	0	$1/8$	0	1	$-21/2$	0
x_5	$1/16$	$-1/8$	0	$-3/64$	1	0	$3/16$	0
x_3	$3/2$	-1	1	$-1/8$	0	0	$21/2$	1
	2	-3	0	$-1/4$	0	0	3	$z-0$

第四次选择 x_7 作为进基变量，x_5 作为出基变量，进行翻转，基变量变为 $\{x_6, x_7, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_6	2	-6	0	$-5/2$	56	1	0	0
x_7	$1/3$	$-2/3$	0	$-1/4$	$16/3$	0	1	0
x_3	-2	6	1	$5/2$	-56	0	0	1
	1	-1	0	$1/2$	-16	0	0	$z-0$

第五次选择 x_1 作为进基变量，x_6 作为出基变量，进行翻转，基变量变为 $\{x_1, x_7, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_1	1	-3	0	$-5/4$	28	$1/2$	0	0
x_7	0	$1/3$	0	$1/6$	-4	$-1/6$	1	0
x_3	0	0	1	0	0	1	0	1
	0	2	0	$7/4$	-44	$-1/2$	0	$z-0$

第六次选择 x_2 作为进基变量，x_7 作为出基变量，进行翻转，基变量变为 $\{x_1, x_2, x_3\}$，得到如下的单纯型表

$\mathbf{x}_B$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
x_1	1	0	0	$1/4$	-8	-1	9	0
x_2	0	1	0	$1/2$	-12	$-1/2$	3	0
x_3	0	0	1	0	0	1	0	1
	0	0	0	$3/4$	-20	$1/2$	-6	$z-0$

则此时循环到了初始状态。实际上在整个迭代过程中，虽然可行基矩阵不断改变，但对应的基本可行解始终是 $\mathbf{x}=[0,0,1,0,0,0,0]^T$，没有变化。

而如果采用 Bland 法则，则在第四次选择 x_1 作为进基变量，x_5 作为出基变量，进行翻转，基变量变为 $\{x_6, x_1, x_3\}$，得到如下的单纯型表

这样就跳出了循环。

□

本书仅介绍了使用 Bland 法则避免循环，还有其它的避免循环的方法，参见文献 [6]。

定理 2.20. 如果不按照（2.19）定义的方式选择出基变量，不能保证在迭代过程中 $\mathbf{x}_B \geq \mathbf{0}$。

该定理很容易构造数值反例证明，留做作业题 2.6，给大家作为练习。

定理 2.21. 在最差情况下，线性规划问题的单纯型法非多项式时间复杂度。

证明：Klee 和 Minty 构造了如下的特殊线性规划问题来证明这一点 [7]

$$\begin{aligned} \min \quad & -x_n \\ \text{s.t.} \quad & x_1 \geq 0 \\ & x_1 \leq 1 \\ & x_i \geq \varepsilon x_{i-1}, i = 2,...,n \\ & x_i \leq 1 - \varepsilon x_{i-1}, i = 2,...,n \end{aligned}$$

其中 $\varepsilon \in (0, 0.5)$。

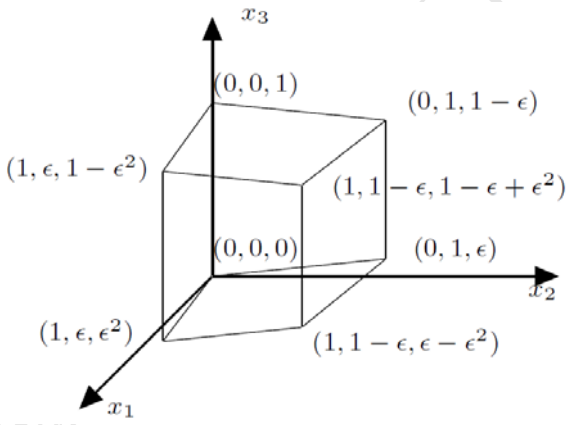


图 2.6 $n = 3$ 时，发生变形的 3 维立方体各极点坐标示意图。

显然该问题的有界多面体可行域为一个发生变形的 n 维立方体，参见图 2.6。不难得知该可行域有 2^n 个极点。我们需要证明，最差的情况是初始解选择原点，需要经过 $2^n - 1$ 次翻转，遍历所有的极点，才能找到最优解 $\mathbf{x} = (0,...,0,1)$。

首先，我们对于 n 维立方体的极点进行 $\{0,1\}^n$ 上的二进制编码。例如，图 2.6 所示的立方体各极点编码为

极点	编码
$(0,0,0)$	$(0,0,0)$
$(0,\varepsilon,\varepsilon^2)$	$(1,0,0)$
$(0,1-\varepsilon,\varepsilon-\varepsilon^2)$	$(1,1,0)$
$(0,1,\varepsilon)$	$(0,1,0)$
$(0,1,1-\varepsilon)$	$(0,1,1)$
$(0,1-\varepsilon,1-\varepsilon+\varepsilon^2)$	$(1,1,1)$
$(0,\varepsilon,1-\varepsilon^2)$	$(1,0,1)$
$(0,0,1)$	$(0,0,1)$

再者，我们定义$\{0,1\}^n$上的一条特殊 Hamiltonian 路径，并证明沿着该路径，目标函数不断下降。实际上这条特殊 Hamiltonian 路径可能看作一类特殊的递归定义的 Gray 码。Gray 码是一种要求相邻的两个数字仅在一个数位上相差 1 的数字编码。

$$\Theta_n=\begin{matrix}0...00\\ 1...00\\ \Theta_{n-1}\\ ...\\ 1...10\\ 0...11\\ ...\\ 1...01\\ 反向\Theta_{n-1}\\ 0...01\end{matrix}$$

接下来，我们使用数学归纳法证明对于n维立方体，我们沿着类似的 Hamiltonian 路径遍历所有极点时，目标函数一直是下降的。

当$n=2$时，考虑对应的线性规划问题

$$\begin{aligned}\min\quad & -x_2\\ \text{s.t.}\quad & x_1\geq 0, x_1\leq 1\\ & x_2\geq \varepsilon x_1, x_2\leq 1-\varepsilon x_1\end{aligned}$$

显然，若初始解选择原点$\mathbf{x}=(0,0)$，需要经过 3 次翻转$\Theta_2=00\to 10\to 11\to 01$，遍历所有的极点，才能找到最优解$\mathbf{x}=(0,0,1)$。

当$n=3$时，对于图 2.6 中的 3 维立方体，若初始解选择原点$\mathbf{x}=(0,0)$，易证单纯型法经过 7 次翻转，才能找到最优解$\mathbf{x}=(0,0,1)$。走过的 Hamiltonian

路径为$\Theta_3 = 000 \to 100 \to 110 \to 010 \to 011 \to 111 \to 101 \to 001$。而且不难发现，这条 3 维 Hamiltonian 路径Θ_3可以分解为两段 2 维 Hamiltonian 路径Θ_2和反向Θ_2，在第一段 2 维 Hamiltonian 路径Θ_2，$x_3 = \varepsilon x_2$。在第二段 2 维 Hamiltonian 路径和反向Θ_2，$x_3 = 1 - \varepsilon x_2$。每次翻转都保持目标函数下降。

类似的，若$n = k$时，上述结果成立。则$n = k+1$时，我们可以得到一条$k+1$维 Hamiltonian 路径Θ_{k+1}，这条$k+1$维 Hamiltonian 路径Θ_3可以分解为两段k维 Hamiltonian 路径Θ_k和反向Θ_k。在第一段k维 Hamiltonian 路径Θ_k，$x_{k+1} = \varepsilon x_k$。在第二段$k$维 Hamiltonian 路径反向$\Theta_k$，$x_{k+1} = 1 - \varepsilon x_k$。对于每段分别应用$n = k$时结论，可以证明每次翻转都保持目标函数下降。易证单纯型法经过$2^{k+1} - 1$次翻转，才能找到最优解$\mathbf{x} = (0,...,1)$。

因此，根据数学归纳法，定理得证。□

例 2.5. 我们使用单纯型法求解例 2.1 中的线性规划问题。

解：

如果我们在一开始不能找到明显的初始可行基，可以使用大 M 法或者两阶段法找到初始可行基。

其中大 M 法在标准形式线性规划问题（2.3）中添加人工变量$\mathbf{u} \in \mathbb{R}^m$，得到如下的线性规划问题

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & z = \mathbf{c}^T\mathbf{x} + \mathbf{m}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{u} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{u} \geq \mathbf{0} \end{aligned} \tag{2.21}$$

其中向量$\mathbf{x} \in \mathbb{R}^n$，$\mathbf{u} \in \mathbb{R}^m$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{A} \in \mathbb{R}^{m \times n}$，$\mathbf{c} \in \mathbb{R}^n$。向量$\mathbf{m} = \begin{bmatrix} M & ... & M \end{bmatrix}^T \in \mathbb{R}^m$的所有元素均为足够大的正数$M$。

因为$\mathbf{u} \geq \mathbf{0}$，$M$又很大，则在线性规划问题（2.21）的求解过程中，我们自然而然的会倾向于令$\mathbf{u} \to \mathbf{0}$，使得目标函数$z = \mathbf{c}^T\mathbf{x} + \mathbf{m}^T\mathbf{u} \to \mathbf{c}^T\mathbf{x}$。如果该问题存在有界的最优解，当应用单纯型法到停止时，会发现$\mathbf{u} = \mathbf{0}$，即人工变量最终能全部出基，最后得到的结果稍作整理就可知原问题的最优解。

但是由于 M 很大，在应用单纯型法时会造成较大的计算舍入误差，因此现在该方法很少使用。为了克服这一困难，两阶段法在第一阶段先考虑问题

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & z = \mathbf{1}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}+\mathbf{u}=\mathbf{b} \\ & \mathbf{x}\geq\mathbf{0},\mathbf{u}\geq\mathbf{0} \end{aligned} \tag{2.22}$$

显然，线性规划问题（2.22）如果存在有界的最优解 $\mathbf{x}_0,\mathbf{u}_0$，必然有 $\mathbf{u}_0=\mathbf{0}$，即人工变量最终能全部出基。而此时得到的 $\mathbf{x}_0$ 的解就标准形式线性规划问题（2.3）的初始可行解。此时，再开始第二阶段，正式求解原问题。

2.6.2　对偶单纯型法

考虑标准形式线性规划问题（2.3）及其对偶问题（2.9）。单纯型法的本质是针对原问题，在保证 $\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}\geq\mathbf{0}$ 的前提下，迭代寻找新的基矩阵直到满足 $\boldsymbol{\sigma}_N=\mathbf{c}_N^T-\mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N\geq\mathbf{0}$。根据强对偶性，此时对偶问题同时达到最优解。

但从对偶问题的角度来看，我们也可以设计如下的对偶单纯型法，在保证 $\boldsymbol{\sigma}_N=\mathbf{c}_N^T-\mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N\geq\mathbf{0}$ 的前提下，迭代寻找新的基矩阵直到满足 $\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}\geq\mathbf{0}$。这样在迭代的过程中，我们保证每次迭代找到的都是对偶问题的可行解，直到找到对偶问题的最优解。根据强对偶性，此时原问题同时达到最优解。

算法 2.2. 对偶单纯型法

Step 1. [确定初始对偶可行解] 如果观察原问题即能找到明显的初始基矩阵 $\mathbf{A}_B$，满足 $\boldsymbol{\sigma}_N=\mathbf{c}_N^T-\mathbf{c}_B^T\mathbf{A}_B^{-1}\mathbf{A}_N\geq\mathbf{0}$，但对应的 $\mathbf{x}_B=\mathbf{A}_B^{-1}\mathbf{b}$ 中存在小于0的元素，则进入 Step 2。

Step 2. [判断最优性] 求 $r=\arg\min_i\{\bar{b}_i\mid i=1,\ldots,m\}$。

如果 $\bar{b}_r\geq 0$，且 $\boldsymbol{\sigma}\geq\mathbf{0}$，则已找到原始问题最优解，停止。

如果 $\bar{b}_r<0$，但矩阵 $\mathbf{A}$ 的第 r 行所有元素非负，$\mathbf{x}\geq\mathbf{0}$，则下式不可能成立

$$x_r+\bar{a}_{r,m+1}x_{m+1}+\ldots+\bar{a}_{r,n}x_n=\bar{b}_r$$

此时可以得出该线性规划问题无解，停止。

如果 $\bar{b}_r<0$，但矩阵 $\mathbf{A}$ 的第 r 行并非所有元素非负。注意到我们可以将对应的基变量 x_r 从负值变为 0 来增加解的可行性并同时减少目标函数值，则进入 Step 3，通过翻转找到一个新的基矩阵 $\mathbf{A}_B$。

Step 3. [翻转（Pivots）] 出基变量选为 x_r，计算求解下述问题，确定可能的进基变量的下标所组成的集合

$$L=\left\{\underset{i}{\arg\min}\left\{-\frac{\sigma_i}{\bar{a}_{ir}}\middle|\sigma_i<0,i=m+1,...,n\right\}\right\} \tag{2.23}$$

如果有多个$-\sigma_i/\bar{a}_{ik}$同时取得最小值，则同样按照 Bland 法则，选取这其中下标最小的i作为进基变量下标l

$$l=\min\{i\mid i\in L,i=m+1,...,n\} \tag{2.24}$$

则进基变量为此时非基变量x_l。

根据原有基本解$\mathbf{x}_B$，出基变量$\mathbf{x}_r$和进基变量$\mathbf{x}_l$构造新的可行基矩阵$\hat{\mathbf{A}}_B$，进而求解除新的基本解$\hat{\mathbf{x}}_B$。这一步可以直接在表中，由进基变量所在列和出基变量所在行决定的约束矩阵系数通过初等变换得到。最后回到 Step 2 进行下一轮迭代。

定理 2.22. 对偶单纯型法一定能在有限时间内停止。

证明：对偶单纯型法每次迭代都是从一个基本解前进到另外一个基本解。根据定理 2.6 和定理 2.4，线性规划问题的基本解个数是有限的。

如果遇到非退化基本解。考虑到$\sigma_i<0$，$\bar{b}_r<0$，$\bar{a}_{ik}>0$，不难发现迭代前后，目标函数值单调下降

$$z'=z-\frac{\sigma_i\bar{b}_r}{\bar{a}_{ik}}<z$$

如果遇到退化基本解，则采用 Bland 方法保证不发生循环。则对偶单纯型法一定能在有限时间内停止。□

对偶单纯型法特别适合如下类型的线性规划问题

$$\begin{aligned}\min_{\mathbf{x}}\quad & z=\mathbf{c}^T\mathbf{x}\\ \text{s.t.}\quad & \mathbf{A}\mathbf{x}\geq\mathbf{b}\\ & \mathbf{x}\geq\mathbf{0}\end{aligned} \tag{2.25}$$

通过添加人工变量$\mathbf{u}\in\mathbb{R}^m$，我们可以得到如下的线性规划问题

$$\begin{aligned}\min_{\mathbf{x},\mathbf{u}}\quad & z=\mathbf{c}^T\mathbf{x}\\ \text{s.t.}\quad & \mathbf{A}\mathbf{x}+\mathbf{u}=\mathbf{b}\\ & \mathbf{x}\geq\mathbf{0},\mathbf{u}\geq\mathbf{0}\end{aligned} \tag{2.26}$$

该问题直接可以观测到初始的对偶可行解$\mathbf{x}=\mathbf{0}$，$\mathbf{u}=\mathbf{b}$；然后开始应用对偶单纯型法求解。我们在本书 3.3 节可以看到该方法的一个典型应用。

例 2.6. 我们使用对偶单纯型法求解例 2.1 中的线性规划问题。

解：

2.7　其他

当我们考虑生产建设的时候，常可以建模得到如下的线性规划问题及其对偶问题

$$\begin{array}{lll} \max\limits_{\mathbf{x}} & z=\mathbf{c}^T\mathbf{x} & \min\limits_{\mathbf{u}} \quad \mathbf{b}^T\mathbf{u} \\ \text{s.t.} & \mathbf{A}\mathbf{x}\le\mathbf{b}, & \text{s.t.} \quad \mathbf{A}^T\mathbf{u}\ge\mathbf{c} \\ & \mathbf{x}\ge\mathbf{0} & \mathbf{u}\ge\mathbf{0} \end{array} \tag{2.27}$$

其中 b_i 表示投入的第 i 种生产资源的总量，a_{ij} 表示投入的单位数量的第 i 种生产资源产出第 j 种产品的转化率，c_j 表示单位数量的第 j 种产品的价值，x_j 表示计划产出第 j 种产品的数量；$i=1,...,m$，$j=1,...,n$。

假设对偶问题最优解为 $\hat{\mathbf{u}}$，由强对偶性知，原问题的目标函数最优值为 $\mathbf{b}^T\hat{\mathbf{u}}=\sum_{i=1}^{m}b_i\hat{u}_i$。所以，原问题的目标函数在最优解处关于 b_i 的偏导数分别是 $\hat{u}_i$。这说明增加一个单位的第 i 种生产资源，最优目标值可望增加 $\hat{u}_i$，故称 $\hat{u}_i$ 为 b_i 的影子价格。

进一步考虑检验数 $\sigma_j=c_j-\mathbf{c}_B^T\mathbf{A}_B^{-1}\bar{\boldsymbol{\alpha}}_j=c_j-\hat{\mathbf{u}}^T\bar{\boldsymbol{\alpha}}_j=c_j-\sum_{i=1}^{m}\hat{u}_i\bar{\alpha}_{ij}$，则 σ_j 等于生产一个单位第 j 种产品的利润减去按影子价格计算的资源的总成本，如果差值 σ_j 大于零，应继续生产，所以最优解必须满足所有检验数非正。

实际上，我们可以进一步分析目标函数带有参数的线性规划问题。例如考虑目标函数中含参数 λ 的线性规划问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \left(\mathbf{c} + \lambda \bar{\mathbf{c}}\right)^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.28}$$

我们可以首先令 $\lambda = 0$ 解无参数的问题，然后确定保持当前最优基矩阵不变的 λ 各个取值区间，最后在 λ 各个取值区间里，求解带参数的问题最优解。

对于约束条件右边常数含参数的线性规划问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} + \lambda \bar{\mathbf{b}} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.29}$$

我们可以转而考虑其对偶问题

$$\begin{aligned} \max_{\mathbf{u}} \quad & \left(\mathbf{b} + \lambda \bar{\mathbf{b}}\right)^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T \mathbf{u} \leq \mathbf{c} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{2.30}$$

转换为上述目标函数带有参数的线性规划问题进行求解。显然，这一变换的想法就是保持可行域不含参数，方便我们讨论分析。

例 2.7. 我们求解如下带参数的线性规划问题。

解：

2.8 作业题

2.1 如果某个标准形式线性规划问题的一个极点比其周围的所有极点目标函数值都小，试证明或证否该极点处取得最优解。

2.2 如果某个标准形式线性规划问题的一个极点比其它所有极点目标函数值都小，试证明或证否该极点处取得最优解。

2.3 如果一个可行解是标准形式线性规划问题的最优解，则其对应的检验数满足 $\boldsymbol{\sigma} \geq \mathbf{0}$。

2.4 如果从标准形式线性规划问题的约束条件矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 的 n 列中挑出 m 列构成的矩阵 $\mathbf{B}$ 可逆，剩下的 $n-m$ 列构成的矩阵 $\mathbf{N}$。满足 $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$，

$\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N} \geq \mathbf{0}$，则其对应的解是该线性规划问题的最优解。

2.5　试证明或证否：如果线性规划问题$\left\{\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\right\}$存在无界解，则锥可行域$\left\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{0}\right\}$有且必有一条极线$\mathbf{d}$满足$\mathbf{c}^T\mathbf{d} \leq \mathbf{0}$。

2.6　试构造反例证明，如果不按照（2.19）定义的方式选择出基变量，不能保证在迭代过程中$\mathbf{x}_B \geq \mathbf{0}$。

2.7　试证明或证否：如果一个线性规划问题的可行域是有界的多面体，则其对偶问题的可行域一定也构成有界的多面体。

2.9　参考文献

[1] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.

[2] D. Bertsimas, J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific Press, 1997.

[3] D. G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, 4th edition, Springer, 2016.

[4] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *Journal of Mathematics and Physics*, vol. 20, no. 1-4, pp. 224-230 1941.

[5] T. C. Koopmans, "Optimum utilization of the transportation system," *Proceedings of the International Statistical Conference*, Washington, DC, 1947.

[6] G. B. Dantzig, "Linear Programming," *Operations Research*, vol. 50, no. 1, pp. 42-47, 2002.

[7] V. Klee, G. J. Minty, "How good is the simplex algorithm?" in *Inequalities, III*, O. Shisha, ed., pp. 159-175, Academic Press, 1972.

[8] L. G. Khachiyan, "A polynomial algorithm for linear programming," *Doklady Akademii Nauk SSSR*, vol. 244, pp. 1093-1096, 1979.

[9] N. K. Karmarkar, "A new polynomial-time algorithmfor linear programming," *Combinatorica*, vol. 4, pp. 373-395, 1984.

[10] D. A. Spielman, S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, no. 3, pp. 385-463, 2004.

[11] A. J. Hoffman, "Cycling in the simplex algorithm," Technical report 2974, National Bureau of Standards, Gaithersburg, USA, December 16th, 1953.

[12] E. M. L. Beale, "Cycling in the dual simplex algorithm," *Naval Research Logistics Quarterly*, vol. 2, no. 4, pp. 269-275, 1955.

[13] R. G. Bland, "New finite pivoting rules for the simplex method," *Mathematics of Operations Research*, vol. 2, no. 2, pp. 103-107, 1977.

[14] T. Terlaky, "Lexicographic pivoting rules," in *Encyclopedia of Optimization*, C A. Floudas, P. M. Pardalos, eds., Springer, pp. 1870-1873, 2009.

第3章 非线性规划

3.1 引言

非线性规划（Non-linear Programming），特别是凸优化（Convex Optimization），是运筹学中的一个重要分支，研究具有非线性约束条件或者非线性线性目标函数的实数极值规划问题。由于现实中，更多的应用问题属于非线性优化，因此在越来越多领域发挥着广泛的作用。本章先介绍非线性规划（特别是凸优化问题和可微优化问题）的基本概念和分类，接着介绍非线性规划问题的有解性和解的最优性条件，然后介绍非线性规划的对偶理论，最后介绍非线性规划问题的搜索求解的一般思路，典型算法及其复杂度。

3.2 非线性规划基础

3.2.1 非线性规划问题的定义和典型凸优化问题

定义 3.1. 非线性规划问题常写为如下形式

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \le 0 \\ & h_j(\mathbf{x}) = 0 \end{aligned} \tag{3.1}$$

其中 $\mathbf{x} \in \mathbb{R}^n$ 为待定决策变量，$f_0(\cdot): \mathbb{R}^n \to \mathbb{R}$ 为目标函数，函数 $f_i(\cdot): \mathbb{R}^n \to \mathbb{R}$ 刻画不等式约束条件，$i \in I = \{1,...,m\}$，函数 $h_i(\cdot): \mathbb{R}^n \to \mathbb{R}$ 刻画等式约束条件，$j \in J = \{1,...,l\}$。

如果问题不显式包含约束条件，则为无约束优化问题（Unconstrained Optimization Problem）。

如果 $f_i(\cdot)$ 均为凸函数，同时 $h_j(\cdot)$ 均为仿射函数，则该问题称为凸优化问题。根据定义不难证明，凸优化问题的可行域为凸集。

如果 $f_i(\cdot)$ 均为可微函数，同时 $h_j(\cdot)$ 均为仿射函数，则该问题称为可微优化问题。

很多问题同时是凸优化问题和可微优化问题。我们称之为可微的凸优化问题。

本书讲解的非线性规划问题基本都是凸优化问题或者可微优化问题，这是因为凸优化问题和可微优化问题有着良好的理论分析特性，高效的实际可计算性和强大的应用建模能力。很大一部分凸优化问题是确定性机器上多项式时间可解的；我们已经建立了十分有效的求解算法，可以快速求得全局最优解。而且现实生活中，很多问题都属于可微的凸优化问题。例如，线性规划就是凸优化问题的特例。因此，凸优化问题的分析和求解具有重要的研究意义和实用价值。

此外，凸优化是研究连续变量优化的起点和基础。目前很多非凸优化问题中非凸性的刻画都脱胎于凸优化，相关问题的求解也和凸优化联系在一起，常常有赖于找到这些非凸优化问题中“凸”的结构。很多非凸优化或 NP-Hard 的问题可以近似转化为多项式时间可解的凸优化问题，并由此给出原问题的界。

认识典型的凸优化问题，可以帮助我们迅速确定一个应用问题是否可以用凸优化求解算法求解。常见的凸优化问题包括线性规划及以下这些。

定义 3.2. 凸二次规划问题（Quadratic Programming Problems）

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} + r \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{G}\mathbf{x} = \mathbf{d} \end{aligned} \tag{3.2}$$

其中 $\mathbf{P} \in S_+^{n\times n}$，$\mathbf{q} \in \mathbb{R}^n$，$r \in \mathbb{R}$，$\mathbf{A} \in \mathbb{R}^{m\times n}$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{G} \in \mathbb{R}^{l\times n}$，$\mathbf{d} \in \mathbb{R}^l$。

定义 3.3. 凸二次约束二次规划问题（Quadratically Constrained Quadratic Programming Problems）

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{x}^T\mathbf{P}_0\mathbf{x} + \mathbf{q}_0^T\mathbf{x} + r_0 \\ \text{s.t.} \quad & \mathbf{x}^T\mathbf{P}_i\mathbf{x} + \mathbf{q}_i^T\mathbf{x} + r_i \le 0 \\ & \mathbf{G}\mathbf{x} = \mathbf{d} \end{aligned} \tag{3.3}$$

其中 $\mathbf{P}_i \in S_+^{n\times n}$，$\mathbf{q}_i \in \mathbb{R}^n$，$r_i \in \mathbb{R}$，$i = 0,...,m$，$\mathbf{G} \in \mathbb{R}^{l\times n}$，$\mathbf{d} \in \mathbb{R}^l$。

定义 3.4. 二次锥规划问题（Second-Order Cone Programming Problems）

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{q}_0^T\mathbf{x} \\ \text{s.t.} \quad & \left\|\mathbf{A}_i\mathbf{x} + \mathbf{b}_i\right\|_2 \le \mathbf{c}_i^T\mathbf{x} + r_i \\ & \mathbf{G}\mathbf{x} = \mathbf{d} \end{aligned} \tag{3.4}$$

其中 $\mathbf{q}_0 \in \mathbb{R}^n$，$\mathbf{A}_i \in \mathbb{R}^{k_i \times n}$，$\mathbf{b}_i \in \mathbb{R}^{k_i}$，$\mathbf{c}_i \in \mathbb{R}^n$，$r_i \in \mathbb{R}$，$k_i \in \mathbb{N}$，$i=1,...,m$，$\mathbf{G} \in \mathbb{R}^{l \times n}$，$\mathbf{d} \in \mathbb{R}^l$。当 $r_i = 0$ 时，退化为二次约束二次规划问题。当 $k_i = 0$ 时，退化为线性规划问题。

定义 3.5. 半正定规划问题（Semi-Definite Programming Problems）

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{q}^T \mathbf{x} \\ \text{s.t.} \quad & x_1 \mathbf{F}_1 + x_2 \mathbf{F}_2 + ... + x_n \mathbf{F}_n + \mathbf{G} \in S_+^{n \times n} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{3.5}$$

其中 $\mathbf{G} \in S^{n \times n}$，$\mathbf{F}_i \in S^{n \times n}$，$i=1,...,m$，$\mathbf{A} \in \mathbb{R}^{m \times n}$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{q} \in \mathbb{R}^n$。

很多时候，我们需要将问题进行转化，以期化为凸优化问题。

例 3.1. 请将几何规划（Geometric Programming Problems）中单项式函数（Monomial Function）等式约束

$$c x_1^{a_1} x_2^{a_2} ... x_n^{a_n} = 1$$

和正项式函数（Posynomial Function）不等式条件

$$\sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} ... x_n^{a_{nk}} \le 1$$

转化为凸函数约束。其中 $c > 0$，$c_k > 0$，$k=1,...,K$。

解：我们可以引入变换 $y_i = \log x_i$，将上述单项式函数等式约束变为仿射等式约束

$$\mathbf{a}^T \mathbf{y} + \log c = 0$$

将正项式函数不等式条件变为凸函数不等式约束

$$\log\left(\sum_{k=1}^{K} \exp\left(\mathbf{a}_{ik}^T \mathbf{y} + \log c_k \right) \right) \le 0$$

这样即可得到凸优化问题。

3.2.2　非线性规划问题解的特性

定义 3.6. 非线性规划问题（3.1）的目标函数和所有约束函数定义域的交集称为域（Domain）

$$D = \operatorname{dom} f_0 \cap \left(\bigcap_{i=1}^{m} \operatorname{dom} f_i \right) \cap \left(\bigcap_{i=1}^{m} \operatorname{dom} h_i \right) \tag{3.6}$$

域中满足所有约束条件的解 $\mathbf{x}$ 被称为可行解（Feasible Solution）。所有可行

解所组成的集合 X 称为可行区域（Feasible Region，或者 Feasible Set）

$$X=\left\{\mathbf{x}\,|\,\mathbf{x}\in D, g_i\left(\mathbf{x}\right)\le 0, h_i\left(\mathbf{x}\right)=0\right\} \tag{3.7}$$

而非线性规划问题（3.1）的最优值（Optimal Value）定义为

$$p^*=\inf\left\{f_0\left(\mathbf{x}\right)\,|\,\mathbf{x}\in X\right\} \tag{3.8}$$

如果同时存在 $\mathbf{x}^*\in X$ 使得 $f_0\left(\mathbf{x}^*\right)=p^*$，则 $\mathbf{x}^*\in X$ 称为非线性规划问题（3.1）的最优解（Optimal Solution，或者 Optimal Point）。

我们如此定义最优值，是因为非线性规划问题和线性规划问题不同，最优值可能无法在可行域内取到。一般而言，求解非线性规划问题有四种可能的情况。其中前三中情况与线性规划问题的解类似。

第一种情况，非线性规划问题无解（Infeasible）。

第二种情况，非线性规划问题存在无界解（Unbounded），也即随着决策变量趋向于无穷，目标函数值也会趋向于无穷。例如如下问题

$$\min_{x>0}-\log x$$

第三种情况，非线性规划问题存在有界的最优解，且可以取到最优解。例如如下问题

$$\min_{x>0} x\log x$$

最优解在 $x=1/e$，最优值为 $-1/e$。

第四种情况，非线性规划问题存在有界的最优解，但不能取到最优解。例如如下问题

$$\min_{x>0}\frac{1}{x}$$

最优值为 0，但无法取到。

不仅如此，我们还关心一个非线性规划问题是否存在局部最优解。

定义 3.7. 如果一个点 $\mathbf{x}$ 满足非线性规划问题（3.1）的约束条件，且存在一个以 $\mathbf{x}$ 为球心，半径为 r 的闭球 $B_r\left(\mathbf{x}\right)$ 使得 $B_r\left(\mathbf{x}\right)$ 内满足非线性规划问题（3.1）的约束条件的任意一点 $\mathbf{y}$ 都成立都满足 $f_0\left(\mathbf{x}\right)\le f_0\left(\mathbf{y}\right)$，那么 $\mathbf{x}$ 可以被称为非线性规划问题（3.1）的局部最优点。同时，点 $\mathbf{x}$ 应当如下优化问题的解

$$\begin{aligned}\min_{\mathbf{y}} \quad & z = f_0(\mathbf{y}) \\ \text{s.t.} \quad & f_i(\mathbf{y}) \le 0 \\ & h_j(\mathbf{y}) = 0 \\ & \|\mathbf{y}-\mathbf{x}\|_2 \le r\end{aligned} \tag{3.9}$$

例如对于问题 $\min_{x>0} x^3 - 3x$，$x = 1$ 是一个局部最优解。

定理 3.1. 凸优化问题（3.1）的局部最优解也是全局最优解。

证明：我们使用反证法证明。假设 $\mathbf{x}$ 是凸优化问题（3.1）的局部最优解，$\mathbf{y}$ 是全局最优解，且 $f_0(\mathbf{y}) \le f_0(\mathbf{x})$。

$\mathbf{x}$ 是局部最优解，则存在以 $\mathbf{x}$ 为球心，半径为 r 的闭球 $B_r(\mathbf{x})$ 使得 $B_r(\mathbf{x})$ 内满足非线性规划问题（3.1）的约束条件的任意一点 $\mathbf{z}$ 都成立都满足 $f_0(\mathbf{x}) \le f_0(\mathbf{z})$。

考虑 $\mathbf{x}$ 和 $\mathbf{y}$ 相连线段上的特定点 $\mathbf{z}' = \dfrac{r}{2\|\mathbf{y}-\mathbf{x}\|_2}\mathbf{y} + \left(1 - \dfrac{r}{2\|\mathbf{y}-\mathbf{x}\|_2}\right)\mathbf{x}$，因为 $f_0(\mathbf{y}) \le f_0(\mathbf{x})$，因此 $\|\mathbf{y}-\mathbf{x}\|_2 > r$，故而 $0 < \dfrac{r}{2\|\mathbf{y}-\mathbf{x}\|_2} < \dfrac{1}{2}$，因此 $\|\mathbf{z}-\mathbf{x}\|_2 \le r$。

且根据 $f_0(\cdot)$ 是凸函数的性质有

$$f_0(\mathbf{z}') \le \frac{r}{2\|\mathbf{y}-\mathbf{x}\|_2} f_0(\mathbf{x}) + \left(1 - \frac{r}{2\|\mathbf{y}-\mathbf{x}\|_2}\right) f_0(\mathbf{y}) \le f_0(\mathbf{x})$$

这与我们假设的 $\mathbf{x}$ 是局部最优解矛盾。□

3.2.3　无约束可微优化问题的最优解特性

接下来，我们讨论无约束可微优化问题的最优解应该满足哪些条件。给定一个点，如果从定义出发判断该点是否是目标函数的一个局部最优解，我们需要对其邻域内的所有点进行检验，这通常很难办或者不可行。因此，我们通常使用所谓最优性条件来快速判断一个点是否为无约束可微优化问题的局部最优解，这样的最优性条件主要包含一阶最优性条件和二阶最优性条件。

定义 3.8. [可微函数的下降方向（Descent Direction）] 对于可微函数 $f(\cdot): \mathbb{R}^n \to \mathbb{R}$ 和点 $\mathbf{x} \in \mathbb{R}^n$，如果存在向量 $\mathbf{d} \in \mathbb{R}^n$ 满足

$$\nabla f\left(\mathbf{x}\right)^T \mathbf{d} < 0 \tag{3.10}$$

则我们称 $\mathbf{d}$ 为 $f\left(\cdot\right)$ 在点 $\mathbf{x}$ 处的一个下降方向。

定理 3.2. [无约束可微优化问题一阶必要条件] 对于可微函数 $f\left(\cdot\right):\mathbb{R}^n \to \mathbb{R}$，如果点 $\mathbf{x}^* \in \mathbb{R}^n$ 是无约束可微优化问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的一个局部最优解，则 $\nabla f\left(\mathbf{x}^*\right) = 0$。

证明：任取方向 $\mathbf{d} \in \mathbb{R}^n$，考察函数 f 在点 $\mathbf{x}^*$ 附近的带 Peano 余项的一次 Taylor 展开，我们有

$$f\left(\mathbf{x}^* + t\mathbf{d}\right) = f\left(\mathbf{x}^*\right) + t\nabla f\left(\mathbf{x}^*\right)^T \mathbf{d} + o\left(t\right)$$

其中 $t > 0$。

移项后等式两边同时除以 t^2，有

$$\frac{f\left(\mathbf{x}^* + t\mathbf{d}\right) - f\left(\mathbf{x}^*\right)}{t} = \nabla f\left(\mathbf{x}^*\right)^T \mathbf{d} + o\left(1\right)$$

因为点 $\mathbf{x}^*$ 是局部最优解，所以

$$\lim_{t\to 0^+} \frac{f\left(\mathbf{x}^* + t\mathbf{d}\right) - f\left(\mathbf{x}^*\right)}{t} = \nabla f\left(\mathbf{x}^*\right)^T \mathbf{d} \geq 0$$

$$\lim_{t\to 0^-} \frac{f\left(\mathbf{x}^* + t\mathbf{d}\right) - f\left(\mathbf{x}^*\right)}{t} = \nabla f\left(\mathbf{x}^*\right)^T \mathbf{d} \leq 0$$

因此，对于任意方向 $\mathbf{d}$ 均有 $\nabla f\left(\mathbf{x}^*\right)^T \mathbf{d} = 0$。考虑到 $\mathbf{d}$ 的任意性，必有 $\nabla f\left(\mathbf{x}^*\right) = 0$。□

上述条件只是必要条件。例如对于问题 $f\left(\mathbf{x}\right) = x^3$，$x = 0$ 满足 $\nabla f\left(\mathbf{x}\right) = 0$，但不是一个局部最优解。因此，还需其它条件进一步判断是否一个点是最优解。

对于二次可微函数 $f\left(\cdot\right):\mathbb{R}^n \to \mathbb{R}$，我们可以借助当前点处的二阶泰勒展开来逼近该函数在该点附近的取值情况，从而来判断最优性。

定理 3.3. [无约束可微优化问题二阶必要条件] 对于二次可微函数 $f\left(\cdot\right):\mathbb{R}^n \to \mathbb{R}$，如果点 $\mathbf{x}^* \in \mathbb{R}^n$ 是无约束可微优化问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的一个局部最优解，则 $\nabla f\left(\mathbf{x}^*\right) = 0$，$\nabla^2 f\left(\mathbf{x}^*\right) \in S_+^{n\times n}$。

[无约束可微优化问题二阶充分条件] 对于二次可微函数 $f\left(\cdot\right):\mathbb{R}^n \to \mathbb{R}$，如果点 $\mathbf{x}^* \in \mathbb{R}^n$ 满足 $\nabla f\left(\mathbf{x}^*\right) = 0$，$\nabla^2 f\left(\mathbf{x}^*\right) \in S_{++}^{n\times n}$，则 $\mathbf{x}^*$ 是无约束可微优化问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的一个局部最优解。

证明：我们首先利用反证法证明二阶必要条件。假设局部最优解点 $\mathbf{x}^*$ 处，函

数 f 存在 $\nabla f\left(\mathbf{x}^*\right)=0$，但 $\nabla^2 f\left(\mathbf{x}^*\right)\notin S_+^{n\times n}$。也即 $\nabla^2 f\left(\mathbf{x}^*\right)$ 存在负的特征值 $\lambda^-<0$。不妨取向量 $\mathbf{d}\in\mathbb{R}^n$ 为特征值 λ^- 对应的特征向量。进一步考察函数 f 在点 $\mathbf{x}^*$ 附近沿着该特征向量方向 $\mathbf{d}$ 的带 Peano 余项的二次 Taylor 展开，我们有

$$f\left(\mathbf{x}^*+\mathbf{d}\right)=f\left(\mathbf{x}^*\right)+\nabla f\left(\mathbf{x}^*\right)^T\mathbf{d}+\frac{1}{2}\mathbf{d}^T\nabla^2 f\left(\mathbf{x}^*\right)\mathbf{d}+o\left(\left\|d\right\|_2^2\right)$$

$$=f\left(\mathbf{x}^*\right)+\frac{1}{2}\mathbf{d}^T\nabla^2 f\left(\mathbf{x}^*\right)\mathbf{d}+o\left(\left\|d\right\|_2^2\right)$$

对于上式稍微变换，可以得到

$$\frac{f\left(\mathbf{x}^*+\mathbf{d}\right)-f\left(\mathbf{x}^*\right)}{\left\|d\right\|_2^2}=\frac{1}{2}\lambda^-+o(1)$$

当 $\left\|d\right\|_2^2$ 充分小时，$f\left(\mathbf{x}^*+\mathbf{d}\right)-f\left(\mathbf{x}^*\right)<0$，这和点 $\mathbf{x}^*$ 是局部最优解矛盾。因此二阶必要条件成立。

证明二阶必要条件与之类似。因为 $\nabla^2 f\left(\mathbf{x}^*\right)\in S_{++}^{n\times n}$，则 $\nabla^2 f\left(\mathbf{x}^*\right)$ 的最小特征值 $\lambda_{\min}>0$，则对于任意向量 $\mathbf{d}\in\mathbb{R}^n$，存在

$$\mathbf{d}^T\nabla^2 f\left(\mathbf{x}^*\right)\mathbf{d}\geq\lambda_{\min}\left\|d\right\|_2^2>0$$

考察函数 f 在点 $\mathbf{x}^*$ 附近沿着任意方向 $\mathbf{d}$ 的带 Peano 余项的二次 Taylor 展开，我们有

$$\frac{f\left(\mathbf{x}^*+\mathbf{d}\right)-f\left(\mathbf{x}^*\right)}{\left\|d\right\|_2^2}\geq\frac{1}{2}\lambda_{\min}+o(1)$$

当 $\left\|d\right\|_2^2$ 充分小时，$f\left(\mathbf{x}^*+\mathbf{d}\right)-f\left(\mathbf{x}^*\right)>0$，因此二阶必要条件成立。□

定义 3.9. 我们把满足 $\nabla f\left(\mathbf{x}\right)=0$ 的点 $\mathbf{x}$ 称为函数 f 的稳定点，或者驻点。如果点 $\overline{\mathbf{x}}\in\mathbb{R}^n$ 满足一阶最优性条件 $\nabla f\left(\overline{\mathbf{x}}\right)=0$，且该点处的 Hessian 矩阵 $\nabla^2 f\left(\overline{\mathbf{x}}\right)$ 既有正特征值又有负特征值，我们称稳定点 $\overline{\mathbf{x}}$ 为一个鞍点（Saddle Point）。

例 3.2. 请求下述无约束严格凸二次规划问题的最优解

$$\min_{\mathbf{x}} f_0\left(\mathbf{x}\right)=\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{q}^T\mathbf{x}+r \tag{3.2}$$

其中 $\mathbf{P}\in S_+^{n\times n}$，$\mathbf{q}\in\mathbb{R}^n$，$r\in\mathbb{R}$。

解：因为 $\nabla^2 f_0\left(\mathbf{x}\right)=\mathbf{P}\in S_+^{n\times n}$，因此，满足 $\nabla f_0\left(\mathbf{x}^*\right)=\mathbf{P}\mathbf{x}^*+\mathbf{q}=0$ 的解 $\mathbf{x}^*$ 为该问题的最优解。不难求得 $\mathbf{x}^*=-\mathbf{P}^{-1}\mathbf{q}$。

3.2.4 有约束可微优化问题的最优解特性

本节讨论更为复杂的有约束可微优化问题的最优解应该满足哪些条件。

定义 3.10. [闭的非空集合的可行方向（Feasible Direction）] 对于闭的非空集合$\Omega \subseteq \mathbb{R}^n$和点$\mathbf{x} \in \Omega$，如果存在向量$\mathbf{d} \in \mathbb{R}^n$和$\bar{\alpha} > 0$满足

$$\alpha \in [0, \bar{\alpha}], \quad \mathbf{x} + \alpha \mathbf{d} \in \Omega \tag{3.11}$$

则我们称$\mathbf{d}$为集合Ω在点$\mathbf{x}$处的一个可行方向。而所有点$\mathbf{x}$处的可行方向构成的集合用$F_\Omega(\mathbf{x})$表示。$F_\Omega(\mathbf{x})$显然是锥，但不一定是闭锥，也不一定是凸锥。

定理 3.4. 如果可行解$\mathbf{x}^*$是可微凸优化规划问题（3.1）的一个局部极小点，且该问题的可行域为闭的非空集合$\Omega \subseteq \mathbb{R}^n$，则

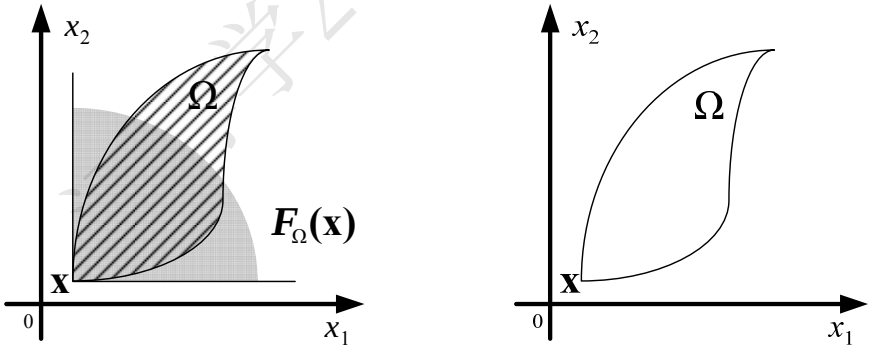
$$\forall \mathbf{d} \in F_\Omega(\mathbf{x}^*), \quad \nabla f_0(\mathbf{x}^*)^T \mathbf{d} \geq 0 \tag{3.12}$$

或者等价的

$$F_\Omega(\mathbf{x}^*) \cap \left\{ \mathbf{d} \mid \nabla f_0(\mathbf{x}^*)^T \mathbf{d} < 0, \mathbf{d} \in \mathbb{R}^n \right\} = \phi \tag{3.13}$$

证明：类似定理 3.5 的证明，可用反证法直接得出。本书从略。□

该定理对于可微凸优化问题比较容易应用，但很多可微非线性规划问题的可行域并非凸集，导致可行方向锥$F_\Omega(\mathbf{x})$可能仅含零向量，难以进一步分析；参见图 3.1。因此，我们还需要引入更加松弛的约束条件。



（a） （b）

图 3.1 示意图：（a）斜纹阴影区域所示的不等式约束构成闭非空集合$\Omega \subset \mathbb{R}^2$，$F_\Omega(\mathbf{x})$

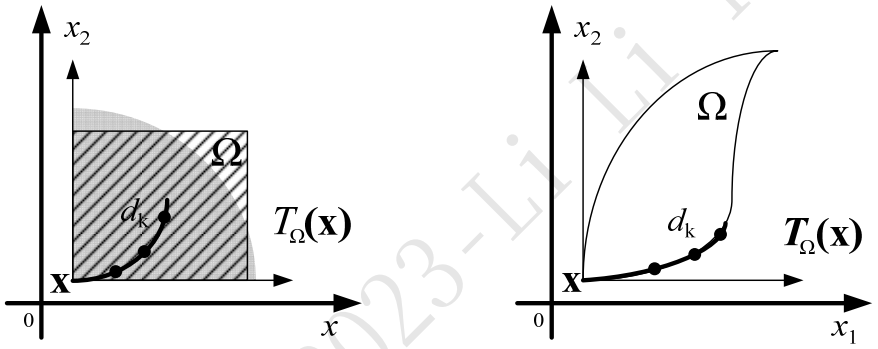
构成一个锥；（b）曲线所示的不等式约束构成闭非空集合$\Omega \subset \mathbb{R}^2$，$F_\Omega(\mathbf{x})$仅含零向量。

定义 3.11. 给定闭的非空集合$\Omega \subseteq \mathbb{R}^n$及其内一点$\mathbf{x}$，若存在可行序列$\{\mathbf{z}_k\}_{k=1}^{\infty} \subset \Omega$逼近点$\mathbf{x}$，即$\lim\limits_{k\to+\infty} \mathbf{z}_k = \mathbf{x}$；及正的标量序列$\{t_k\}_{k=1}^{\infty}$，$t_k \to 0$，满足

$$\lim_{k\to+\infty} \frac{\mathbf{z}_k - \mathbf{x}}{t_k} = \mathbf{d}$$

则称向量$\mathbf{d} \in \mathbb{R}^n$为集合$\Omega$在点$\mathbf{x}$的一个切向量（Tangent Vector）。而所有点$\mathbf{x}$处的切向量构成的集合称为切锥（Tangent Cone），我们用$T_\Omega(\mathbf{x})$表示。

如图 3.2（a）所示，如果集合Ω是凸集时，$F_\Omega(\mathbf{x}) \subseteq T_\Omega(\mathbf{x})$，该证明留给大家自行完成。而如下图 3.2（b）所示，集合Ω非凸集时，$T_\Omega(\mathbf{x})$是闭的非凸集合，$F_\Omega(\mathbf{x})$仅含零向量。



（a）　　（b）

图 3.2　示意图：（a）斜纹阴影区域所示的不等式约束构成闭非空集合$\Omega \subset \mathbb{R}^2$，$T_\Omega(\mathbf{x})$构成一个锥；（b）曲线所示的不等式约束构成闭非空集合$\Omega \subset \mathbb{R}^2$，$T_\Omega(\mathbf{x})$为在点$\mathbf{x}$处与$\Omega$相切的两条射线。

定理 3.5. 如果可行解$\mathbf{x}^*$是可微非线性规划问题（3.1）的一个局部极小点，且该问题的可行域为闭的非空集合$\Omega \subseteq \mathbb{R}^n$，则

$$\forall \mathbf{d} \in T_\Omega(\mathbf{x}^*),\ \nabla f_0(\mathbf{x}^*)^T \mathbf{d} \geq 0 \tag{3.14}$$

或者等价的

$$T_\Omega(\mathbf{x}^*) \cap \{\mathbf{d} \mid \nabla f_0(\mathbf{x}^*)^T \mathbf{d} < 0\} = \phi \tag{3.15}$$

证明：还是用反证法，假设可行解 $\mathbf{x}^*$ 处有 $T_\Omega\left(\mathbf{x}^*\right)\bigcap\left\{\mathbf{d}\,|\,\nabla f_0\left(\mathbf{x}^*\right)^T\mathbf{d}<0\right\}\neq\phi$，存在 $\bar{\mathbf{d}}\in T_\Omega\left(\mathbf{x}^*\right)\bigcap\left\{\mathbf{d}\,|\,\nabla f_0\left(\mathbf{x}^*\right)^T\mathbf{d}<0\right\}$。根据切向量的定义，存在正的标量序列 $\left\{t_k\right\}_{k=1}^{\infty}$，使得 $\mathbf{x}^*+t_k\bar{\mathbf{d}}_k\in\Omega$，其中 $t_k\to 0$ 且 $\mathbf{d}_k\to\mathbf{d}$。由于 $\nabla f_0\left(\mathbf{x}^*\right)^T\mathbf{d}<0$，对于充分大的 k，我们有

$$\begin{aligned}
&f_0\left(\mathbf{x}^*+t_k\bar{\mathbf{d}}_k\right)=f_0\left(\mathbf{x}^*\right)+t_k\nabla f_0\left(\mathbf{x}^*\right)^T\bar{\mathbf{d}}_k+o\left(t_k\right)\\
&=f_0\left(\mathbf{x}^*\right)+t_k\nabla f_0\left(\mathbf{x}^*\right)^T\mathbf{d}+t_k\nabla f_0\left(\mathbf{x}^*\right)^T\left(\bar{\mathbf{d}}_k-\mathbf{d}\right)+o\left(t_k\right)\\
&=f_0\left(\mathbf{x}^*\right)+t_k\nabla f_0\left(\mathbf{x}^*\right)^T\mathbf{d}+o\left(t_k\right)\\
&<f_0\left(\mathbf{x}^*\right)
\end{aligned}$$

这与 $\mathbf{x}^*$ 是局部极小点矛盾。□

实际上，我们还可以放松条件为：$f_i(\cdot)$ 和 $g_i(\cdot)$ 只需在 $\mathbf{x}^*$ 处为可微函数，而不需要要求他们处处可微。

但是切锥是根据可行域的几何性质来定义的。类似于线性规划问题从几何性质到代数计算的演进，研究者又提出了线性化可行方向锥这一代数方法来计算可行方向，以便更容易判断最优性条件。

定义 3.12. 可微非线性规划问题（3.1）的闭的非空可行域 $\Omega\subseteq\mathbb{R}^n$ 内一点 $\mathbf{x}\in\Omega$ 处的积极集（Active Set）$A\left(\mathbf{x}\right)$ 为两部分下标的并集，一部分是等式约束对应的下标，另外一部分是不等式约束中等号成立的约束对应的下标

$$A\left(\mathbf{x}\right)=J\bigcup\left\{i\in I:\,f_i\left(\mathbf{x}\right)=0\right\}\tag{3.16}$$

而点 $\mathbf{x}$ 处的线性化可行方向锥（the Cone of Linearized Feasible Directions）定义为

$$L_\Omega\left(\mathbf{x}\right)=\left\{\mathbf{d}\left|\begin{array}{l}\mathbf{d}^T\nabla h_j\left(\mathbf{x}\right)=0,\forall j\in J\\ \mathbf{d}^T\nabla f_i\left(\mathbf{x}\right)\le 0,\forall i\in A\left(\mathbf{x}\right)\bigcap J\end{array}\right.\right\}\in\mathbb{R}^n\tag{3.17}$$

定理 3.6. 对于可微非线性规划问题（3.1）的闭的非空可行域 $\Omega\subseteq\mathbb{R}^n$ 内一点 $\mathbf{x}\in\Omega$，$T_\Omega\left(\mathbf{x}\right)\subseteq L_\Omega\left(\mathbf{x}\right)$。

证明：假设存在可行序列 $\left\{\mathbf{z}_k\right\}_{k=1}^{\infty}\subset\Omega$ 逼近点 $\mathbf{x}$，即 $\lim\limits_{k\to+\infty}\mathbf{z}_k=\mathbf{x}$；及正的标

量序列$\{t_k\}_{k=1}^{\infty}$，$t_k \to 0$，满足$\lim\limits_{k\to+\infty}\dfrac{\mathbf{z}_k-\mathbf{x}}{t_k}=\mathbf{d}$。因此，有

$$\mathbf{z}_k=\mathbf{x}+t_k\mathbf{d}+o(t_k)$$

对于任意$i\in J$，考虑在点$\mathbf{x}$附近的带 Peano 余项的一次 Taylor 展开，我们有

$$0=\frac{c_i(\mathbf{z}_k)}{t_k}=\frac{c_i(\mathbf{x})+t_k\nabla c_i(\mathbf{x})^T\mathbf{d}+o(t_k)}{t_k}=\nabla c_i(\mathbf{x})^T\mathbf{d}+\frac{o(t_k)}{t_k}$$

当$k\to+\infty$，$\nabla c_i(\mathbf{x})^T\mathbf{d}=0$。其中$c(\mathbf{x})=\begin{bmatrix}f_i(\mathbf{x})\\h_i(\mathbf{x})\end{bmatrix}_{i\in I\cup J}\in\mathbb{R}^{m+l}$为代表所有不等式约束和等式约束构成的列向量值函数，其的第i个分量为$c_i(\mathbf{x})$。

类似的，我们可以证明：对于任意$i\in I$，当$k\to+\infty$，$\nabla c_i(\mathbf{x})^T\mathbf{d}\le 0$。

因为点$\mathbf{x}$的任意性，$T_\Omega(\mathbf{x})\subseteq L_\Omega(\mathbf{x})$。□

线性化可行方向锥容易计算和使用，但线性化可行方向锥$L_\Omega(\mathbf{x})$一般比切锥$T_\Omega(\mathbf{x})$要大。因此研究者进一步引入约束品性（Constraint Qualification）这个概念。当某种约束品性成立时，可以有$L_\Omega(\mathbf{x})=T_\Omega(\mathbf{x})$，进而更方便地研究问题的最优性条件 [1]-[3]。

定理 3.7. [线性约束品性（Affine Constraints Qualification，ACQ，或 Linear Constraints Qualification，LCQ）] 如果非线性规划问题（3.1）的所有的约束函数$f_i(\cdot)$，$i\in I=\{1,...,m\}$，$h_j(\cdot)$，$j\in J=\{1,...,l\}$，均为仿射函数，则称线性约束品性成立。

证明：假设可微非线性规划问题（3.1）的不等式约束条件写作$h_i(\mathbf{x})=\mathbf{a}_i^T\mathbf{x}-b_i\le 0$，等式约束条件写作$h_j(\mathbf{x})=\mathbf{g}_j^T\mathbf{x}-d_j=0$。

假设$\mathbf{x}$是可微非线性规划问题（3.1）的闭的非空可行域$\Omega\subseteq\mathbb{R}^n$内一点，且$L_\Omega(\mathbf{x})$非空，则存在$\mathbf{d}\in\mathbb{R}^n$，满足

$$\left\{\mathbf{d}\left|\begin{array}{l}\mathbf{d}^T\nabla h_j(\mathbf{x})=\mathbf{d}^T\mathbf{g}_j=0,\forall j\in J\\ \mathbf{d}^T\nabla f_i(\mathbf{x})=\mathbf{d}^T\mathbf{a}_i\le 0,\forall i\in A(\mathbf{x})\bigcap J\end{array}\right.\right\}\neq\phi$$

则构造$\mathbf{z}_k=\mathbf{x}+t_k\mathbf{d}$，序列$\{t_k\}_{k=1}^{\infty}$为正的标量序列，$t_k\to 0$。显然有

$$t_k\mathbf{d}^T\mathbf{g}_j=0,\forall j\in J,\ t_k\mathbf{d}^T\mathbf{a}_i\le 0,\forall i\in A(\mathbf{x})\bigcap J$$

则可行序列$\{\mathbf{z}_k\}_{k=1}^{\infty}\subset\Omega$，且逼近点$\mathbf{x}$，$\lim\limits_{k\to+\infty}\mathbf{z}_k=\mathbf{x}$。所以此时$L_\Omega(\mathbf{x})\subseteq T_\Omega(\mathbf{x})$。

若点 $\mathbf{x}$ 处，$L_\Omega(\mathbf{x})$ 为空集，显然亦成立 $L_\Omega(\mathbf{x}) \subseteq T_\Omega(\mathbf{x})$。

根据定理 3.6，又有 $T_\Omega(\mathbf{x}) \subseteq L_\Omega(\mathbf{x})$，因此，线性约束品性成立时两个集合相同，$L_\Omega(\mathbf{x}) = T_\Omega(\mathbf{x})$。□

定理 3.8. [线性无关约束品性（Linear Independence Constraints Qualification，LICQ）] 如果非线性规划问题（3.1）的可行点 $\mathbf{x}$ 处的积极集 $A(\mathbf{x})$ 对应的约束函数的梯度是线性无关的，则称线性无关约束品性成立。

证明：我们仅考虑积极集 $A(\mathbf{x}) = I \cup J$ 的情况，其它情况类似证明。此时，定义矩阵

$$\mathbf{A}(\mathbf{x}) = \left[\nabla f_i(\mathbf{x}), \nabla h_i(\mathbf{x})\right]_{i\in I\cup J}^T \in \mathbb{R}^{(m+l)\times n}$$

若约束函数的梯度是线性无关的，则 $\operatorname{rank}\left(\mathbf{A}(\mathbf{x})\right) = m+l$。

令矩阵 $\mathbf{Z}(\mathbf{x}) \in \mathbb{R}^{n\times(n-m-l)}$ 为矩阵 $\mathbf{A}(\mathbf{x})$ 的零空间的基矩阵，则 $\mathbf{Z}(\mathbf{x})$ 满足 $\operatorname{rank}\left(\mathbf{Z}(\mathbf{x})\right) = n-m-l$，$\mathbf{A}(\mathbf{x})\mathbf{Z}(\mathbf{x}) = \mathbf{0}$。

令 $\mathbf{d} \in L_\Omega(\mathbf{x})$ 为任意线性化可行方向，定义映射 $R: \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$

$$R(\mathbf{z}, t) = \begin{bmatrix} c(\mathbf{x}) - t\mathbf{A}(\mathbf{x})\mathbf{d} \\ \mathbf{Z}(\mathbf{x})^T(\mathbf{z} - \mathbf{x} - t\mathbf{d}) \end{bmatrix}$$

其中 $c(\mathbf{x}) = \begin{bmatrix} f_i(\mathbf{x}) \\ h_i(\mathbf{x}) \end{bmatrix}_{i\in I\cup J} \in \mathbb{R}^{m+l}$ 为代表所有不等式约束和等式约束构成的列向量值函数，其的第 i 个分量为 $c_i(\mathbf{x})$。

接下来，我们将证明给定任意一列满足 $t_k \to 0$ 的正标量 $\{t_k\}_{k=1}^\infty$，可以根据映射 R 确定对应的一列 $\{\mathbf{z}_k\}_{k=1}^\infty$ 使得 $\mathbf{z}_k \to \mathbf{x}$，且保持 $\mathbf{z}_k$ 为可行解。

考虑映射 R 的零点，也即满足 $R(\mathbf{z}, t) = \mathbf{0}$ 的点 $(\mathbf{z}, t)$。在点 $(\mathbf{x}, 0)$ 处，我们有 $R(\mathbf{x}, 0) = \mathbf{0}$，且 $\dfrac{\partial R(\mathbf{x}, t)}{\partial \mathbf{z}} = \begin{bmatrix} \mathbf{A}(\mathbf{x}) \\ \mathbf{Z}(\mathbf{x})^T \end{bmatrix}$。根据 $\mathbf{Z}(\mathbf{x})$ 的定义方式，雅可比矩阵 $\dfrac{\partial R(\mathbf{x}, t)}{\partial \mathbf{z}}$ 是非奇异的。因此，由隐函数定理，对任意充分小的 t_k，都存在唯一的 $\mathbf{z}_k$，使得 $R(\mathbf{z}_k, t_k) = \mathbf{0}$，$i \in I \bigcup J$。根据线性化可行方向的定义，我们有

$$c_i(\mathbf{z}_k) = t_k c_i(\mathbf{x})^T \mathbf{d} \le 0, \quad \forall i \in I \cap A(\mathbf{x})$$

$$c_i(\mathbf{z}_k) = t_k c_i(\mathbf{x})^T \mathbf{d} = 0, \quad \forall i \in J$$

则 $\mathbf{z}_k$ 为可行点。

进一步，考虑在点$\mathbf{z}_k$附近的带 Peano 余项的一次 Taylor 展开，我们有

$$R(\mathbf{z}_k,t_k)=\begin{bmatrix} c(\mathbf{z}_k)-t_k\mathbf{A}(\mathbf{x})\mathbf{d} \\ \mathbf{Z}(\mathbf{x})^T(\mathbf{z}_k-\mathbf{x}-t_k\mathbf{d}) \end{bmatrix}$$
$$=\begin{bmatrix} \mathbf{A}(\mathbf{x})(\mathbf{z}_k-\mathbf{x})+o(\|\mathbf{z}_k-\mathbf{x}\|)-t_k\mathbf{A}(\mathbf{x})\mathbf{d} \\ \mathbf{Z}(\mathbf{x})^T(\mathbf{z}_k-\mathbf{x}-t_k\mathbf{d}) \end{bmatrix}$$
$$=\begin{bmatrix} \mathbf{A}(\mathbf{x}) \\ \mathbf{Z}(\mathbf{x})^T \end{bmatrix}(\mathbf{z}_k-\mathbf{x}-t_k\mathbf{d})+o(\|\mathbf{z}_k-\mathbf{x}\|)$$

因为雅可比矩阵$\dfrac{\partial R(\mathbf{x},t)}{\partial \mathbf{z}}$是非奇异的，我们进一步得到

$$\frac{\mathbf{z}_k-\mathbf{x}}{t_k}=\mathbf{d}+\begin{bmatrix} \mathbf{A}(\mathbf{x}) \\ \mathbf{Z}(\mathbf{x})^T \end{bmatrix}^{-1} o\left(\frac{\|\mathbf{z}_k-\mathbf{x}\|}{t_k}\right)$$

则有$\lim\limits_{k\to+\infty}\dfrac{\mathbf{z}_k-\mathbf{x}}{t_k}=\mathbf{d}$。因为$\mathbf{d}\in L_\Omega(\mathbf{x})$为任意线性化可行方向，所以$L_\Omega(\mathbf{x})\subseteq T_\Omega(\mathbf{x})$。根据定理 3.6，又有$T_\Omega(\mathbf{x})\subseteq L_\Omega(\mathbf{x})$。因此，线性无关约束品性成立时两个集合相同，$L_\Omega(\mathbf{x})=T_\Omega(\mathbf{x})$。□

有了约束品性，我们可以较容易判断在可微非线性规划问题（3.1）的局部最优解$\mathbf{x}^*$处是否有$F_\Omega(\mathbf{x}^*)=T_\Omega(\mathbf{x}^*)$，如果线性化可行方向锥和切锥重合，则集合

$$\left\{\mathbf{d}\left|\begin{array}{l}\mathbf{d}^T\nabla f_0(\mathbf{x})<0\\ \mathbf{d}^T\nabla h_j(\mathbf{x})=0,\forall j\in J\\ \mathbf{d}^T\nabla f_i(\mathbf{x})\le 0,\forall i\in A(\mathbf{x})\cap I\end{array}\right.\right\} \tag{3.18}$$

应该为空集。我们可以进一步将这个判定式子进行变形得到如下的 Karush-Kuhn-Tucker（KKT）条件。

定理 3.9. [KKT 条件] 在可微非线性规划问题（3.1）的局部最优解$\mathbf{x}^*$处有$L_\Omega(\mathbf{x}^*)=T_\Omega(\mathbf{x}^*)$，则存在$\boldsymbol{\lambda}\in\mathbb{R}_+^m$，$\boldsymbol{\mu}\in\mathbb{R}^l$，使得如下 KKT 条件（3.19）-（3.22）成立：

稳定性条件（Stationarity）：

$$\nabla f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i\nabla f_i(\mathbf{x})+\sum_{j=1}^{l}\mu_j\nabla h_j(\mathbf{x})=\mathbf{0} \tag{3.19}$$

原问题可行性条件（Primary Feasibility）：

$$f_i(\mathbf{x}) \le 0,\ i \in I = \{1,...,m\},\ h_j(\mathbf{x}) = 0,\ j \in J = \{1,...,l\} \quad (3.20)$$

对偶问题可行性条件（Dual Feasibility）：

$$\lambda_i \ge 0,\ i \in I = \{1,...,m\} \quad (3.21)$$

互补松弛条件（Complementary Slackness）：

$$\lambda_i f_i(\mathbf{x}) = 0,\ i \in I = \{1,...,m\} \quad (3.22)$$

我们将满足（3.19）-（3.22）条件的点$\mathbf{x}^*$称为 KKT 点或者 KKT 解。

证明：因为$L_\Omega(\mathbf{x}^*) = T_\Omega(\mathbf{x}^*)$，根据定理 3.5，（3.18）式对应的集合为空集。因此，根据 Farkas 引理，存在$\boldsymbol{\lambda} \in \mathbb{R}_+^m$，$\boldsymbol{\mu} \in \mathbb{R}^l$使得（3.19）式成立。令$\lambda_i = 0$，$i \in I \setminus A(\mathbf{x}^*)$，结合$\mathbf{x}^*$的可行性，KKT 条件（3.19）-（3.22）成立。□

值得注意的是，只有在切锥与线性化可行方向锥相同时，在局部最优点$\mathbf{x}^*$处才需要满足 KKT 条件；换而言之，只有在切锥与线性化可行方向锥相同时，KKT 条件才是局部最优解的必要条件。

反过来，即使在在切锥与线性化可行方向锥相同时，KKT 条件也不是局部最优解的充分条件。换而言之，满足 KKT 点不一定是局部最优点。

例 3.3. 请问是否可以使用 KKT 条件找到规划问题$\min\limits_{x^2 \le 0} x$的最优解？

解：该问题的最优解显然为 $x = 0$，约束起作用。该点 $x = 0$ 处$\nabla f_1(x)\big|_{x=0} = 0$，只有一个全零向量，线性相关，不满足 LICQ 条件。进一步检测，对于任意 $\lambda \ge 0$，该点 $x = 0$ 处梯度$\nabla f_0(x)\big|_{x=0} + \lambda \nabla f_1(x)\big|_{x=0} = 1$无法满足 KKT 条件，也即最优解不是 KKT 解。

例 3.4. 请问是否可以使用 KKT 条件找到规划问题$\min\limits_{x^2 \le 0} x^2$的最优解？

解：该问题的最优解显然为 $x = 0$，约束起作用。该点 $x = 0$ 处$\nabla f_1(x)\big|_{x=0} = 0$，只有一个全零向量，线性相关，不满足 LICQ 条件。进一步检测，对于任意 $\lambda \ge 0$，该点 $x = 0$ 处梯度$\nabla f_0(x)\big|_{x=0} + \lambda \nabla f_1(x)\big|_{x=0} = 0$满足 KKT 条件，因此，虽然不满足 LICQ 条件，但最优解是 KKT 解。

对于类似例 3.3 的情况，其实我们可以稍微修正一下，得到如下覆盖面更广的条件。

定理 3.10. [Fritz John 条件] 在可微非线性规划问题（3.1）的局部最优解 $\mathbf{x}^*$ 处有 $L_\Omega\left(\mathbf{x}^*\right)=T_\Omega\left(\mathbf{x}^*\right)$，则存在 $\boldsymbol{\lambda}\in\mathbb{R}_+^m$，$\boldsymbol{\mu}\in\mathbb{R}^l$，使得如下 Fritz John 条件成立：

稳定性条件：

$$\lambda_0\nabla f_0(X)+\sum_{i=1}^{m}\lambda_i\nabla f_i(X)+\sum_{j=1}^{l}\mu_j\nabla h_j(X)=\mathbf{0}$$

原问题可行性条件：

$$f_i(\mathbf{x})\le 0,\ i\in I=\{1,...,m\},\ h_j(\mathbf{x})=0,\ j\in J=\{1,...,l\}$$

对偶问题可行性条件：

$$\lambda_i\ge 0,\ i=\{0,1,...,m\}$$

互补松弛条件：

$$\lambda_i f_i(X)=0,\ i\in I=\{1,...,m\}$$

我们将满足上述条件的点 $\mathbf{x}^*$ 称为 Fritz John 点或者 Fritz John 解。

证明：可从定理 3.9 证明稍加修改得到。本书从略。□

显然，Fritz John 解 $x=0$ 是例 3.3 问题的最优解。

使用 KKT 条件可以比较容易的验证任意的一个解是不是可能为最优解，帮助我们把可行域里边很多不是最优的解排除掉，方便我们进一步寻找真正的最优解。当不容易验证约束品性是否成立的时候，我们也可以先求出所有 KKT 解，然后从中比较，找出最优解。但需要小心，有时局部最优点不一定满足 KKT 条件。

例 3.5. 请问是否可以使用 KKT 条件找到规划问题 $\min\limits_{(x+y-2)^2\le 0} -xy$ 的最优解？

解：该题中满足 KKT 条件的解不存在，但该题存在最优解 $x=y=1$。

实际上，KKT 条件里的 $\boldsymbol{\lambda}\in\mathbb{R}_+^m$，$\boldsymbol{\mu}\in\mathbb{R}^l$ 为非线性规划问题（3.1）的 Lagrange 乘子，参见 3.3 节的深入分析。

3.3　非线性规划的对偶理论

定义 3.13. 对于非线性规划问题（3.1），我们可以给其中每一个约束指定一个 Lagrange 乘子（Lagrange Multipliers），以乘子为加权系数将约束增加到目标函数中，得到 Lagrange 函数（Lagrange Function）

$$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})=f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i f_i(\mathbf{x})+\sum_{i=1}^{l}\mu_i h_i(\mathbf{x}) \tag{3.23}$$

其中$\boldsymbol{\lambda}\in\mathbb{R}^m$，$\boldsymbol{\lambda}\geq\mathbf{0}$，$\boldsymbol{\mu}\in\mathbb{R}^l$。这些 Lagrange 乘子可以被视为违反不同约束所带来负面影响的权重。

对拉格朗日函数$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})$中的$\mathbf{x}$在域$D$内逐点取下确界，我们可以定义 Lagrange 对偶函数（Lagrange Dual Function）

$$g(\boldsymbol{\lambda},\boldsymbol{\mu})=\inf_{\mathbf{x}\in D}L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) \tag{3.24}$$

显然 Lagrange 对偶函数$g(\cdot,\cdot):\mathbb{R}_+^m\times\mathbb{R}^l\to[-\infty,+\infty)$是逐点定义的一族关于$(\boldsymbol{\lambda},\boldsymbol{\mu})$的仿射函数的下确界，根据定理 1.15，无论原始问题是否为凸优化问题，其必为凹函数。

定义 3.14. 为了求解从 Lagrange 对偶函数获得的最优下界，我们定义如下的 Lagrange 对偶问题定义为

$$\begin{aligned}\max_{\boldsymbol{\lambda},\boldsymbol{\mu}}\quad & z=g(\boldsymbol{\lambda},\boldsymbol{\mu})\\ \text{s.t.}\quad & \boldsymbol{\lambda}\geq\mathbf{0}\end{aligned} \tag{3.25}$$

此时，我们称 Lagrange 乘子$(\boldsymbol{\lambda},\boldsymbol{\mu})$为对偶变量（Dual Variable）。由于其目标函数的凹性和约束集合的凸性，Lagrange 对偶问题是一个凸优化问题。当$g(\boldsymbol{\lambda},\boldsymbol{\mu})=-\infty$时，对偶函数提供的关于原问题最优解的下界没有实际意义。只有当$g(\boldsymbol{\lambda},\boldsymbol{\mu})>-\infty$时，对偶函数生成的关于原始问题最优解的下界才是非平凡的。因此我们规定拉格朗日对偶函数的定义域为

$$\mathrm{dom}g=\left\{(\boldsymbol{\lambda},\boldsymbol{\mu})\,|\,\boldsymbol{\lambda}\geq 0,g(\boldsymbol{\lambda},\boldsymbol{\mu})>-\infty\right\} \tag{3.26}$$

当$(\boldsymbol{\lambda},\boldsymbol{\mu})\in\mathrm{dom}g$时，称其为对偶可行解。假设$(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)$是使得对偶问题取得最优值的解，称其为对偶最优解或者最优 Lagrange 乘子。

例 3.6. 请用 Lagrange 对偶问题的形式给出标准形式线性规划问题（2.2）的对偶问题。

解：对于标准形式线性规划问题（2.2），我们可以构造 Lagrange 函数为

$$\begin{aligned}L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})&=\mathbf{c}^T\mathbf{x}-\boldsymbol{\lambda}^T\mathbf{x}+\boldsymbol{\mu}^T(\mathbf{A}\mathbf{x}-\mathbf{b})\\&=-\mathbf{b}^T\boldsymbol{\mu}+(\mathbf{A}^T\boldsymbol{\mu}-\boldsymbol{\lambda}+\mathbf{c})^T\mathbf{x}\end{aligned}$$

其中$\boldsymbol{\lambda}\geq\mathbf{0}$。

而 Lagrange 对偶函数为

$$g(\boldsymbol{\lambda},\boldsymbol{\mu})=\inf_{\mathbf{x}\in\mathbb{R}^n}L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})=\begin{cases}-\mathbf{b}^T\boldsymbol{\mu},\text{if }\mathbf{A}^T\boldsymbol{\mu}-\boldsymbol{\lambda}+\mathbf{c}=\mathbf{0}\\-\infty,\text{otherwise}\end{cases}$$

仅考虑其中有意义的下界，则有对偶问题

$$\max_{\mathbf{u}} \quad \mathbf{b}^T\mathbf{u}$$
$$\text{s.t.} \quad \mathbf{A}^T\mathbf{u}+\boldsymbol{\lambda}=\mathbf{c},\boldsymbol{\lambda}\geq\mathbf{0}$$

约去$\boldsymbol{\lambda}$，我们就不需要根据第 2 章的定义，而是直接从 Lagrange 对偶理论推导出了标准形式线性规划问题（2.2）的对偶问题（2.9）。

$$\max_{\mathbf{u}} \quad \mathbf{b}^T\mathbf{u}$$
$$\text{s.t.} \quad \mathbf{A}^T\mathbf{u}\leq\mathbf{c}$$

定理 3.11. [弱对偶性定理] 记原问题（Primary Problem）的最优值为 p^*

$$p^*=\inf\left\{f_0(\mathbf{x})\middle|\text{s.t. } f_i(\mathbf{x})\leq 0,1\leq i\leq m,h_i(\mathbf{x})=0,1\leq i\leq p\right\} \tag{3.27}$$

对偶问题的最优值为d^*

$$d^*=\sup_{\boldsymbol{\lambda}\geq\mathbf{0}} g\left(\boldsymbol{\lambda},\boldsymbol{\mu}\right) \tag{3.28}$$

我们称 p^*-d^* 为对偶间隙，存在 $p^*-d^*\geq 0$。如果对偶间隙为 0 ，称强对偶性成立。

证明：实际上，假设$\left(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}\right)$是任意的原对偶可行对（指 $\mathbf{x}$ 是原问题可行解，$\left(\boldsymbol{\lambda},\boldsymbol{\mu}\right)$是对偶问题可行解），则总是成立

$$g\left(\boldsymbol{\lambda},\boldsymbol{\mu}\right)\leq d^*\leq p^*\leq f_0\left(\mathbf{x}\right) \tag{3.29}$$

因为

$$f_0\left(\mathbf{x}\right)\geq L\left(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}\right)\geq\inf_{\mathbf{x}\in D}L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu})=g\left(\boldsymbol{\lambda},\boldsymbol{\mu}\right)$$

而因为 p^* 是原问题的最优值，则 $p^*\geq d^*$。□

基于弱对偶性定理，我们常考虑原问题的对偶问题，试图首选寻求原问题最优解的一个下界，来帮助我们进一步分析求解。如果有强对偶性成立，则可以完全转而求解对偶问题。对于凸优化问题，我们常根据 Slater 约束品性（Slater Condition）来判断一个具体问题是否满足强对偶性。

定义 3.15. [相对内点] 给定集合 $D\subseteq\mathbb{R}^n$，根据定义 1.15，记其仿射包为 $\text{affine}D$。则集合 D 的相对内点集定义为

$$\text{relint}D=\left\{\mathbf{x}\in\mathbf{D}\,|\,\exists r>0,B_r\left(\mathbf{x}\right)\bigcap\text{affine}D\subseteq D\right\} \tag{3.30}$$

相对内点是内点的推广。根据定义 1.8，若 $\mathbf{x}$ 是集合 D 的内点，则存在一个以 $\mathbf{x}$ 为球心，合适半径的 n 维球含于集合 D。若 D 的“维数”较低，则 D 不可能有内点。例如超平面不可能有内点。但可以引入上述定义，使得 D 可能有相对内

点。

定义 3.16. [Slater 约束品性] 若下述凸优化问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \le 0 \\ & \mathbf{Ax} = \mathbf{b} \end{aligned} \tag{3.31}$$

存在 $\mathbf{x} \in \text{relint}D$ 满足 $f_i(\mathbf{x}) < 0$，$i = 1,...,m$，$\mathbf{Ax} = \mathbf{b}$，$\mathbf{A} \in \mathbb{R}^{l\times n}$，$\mathbf{b} \in \mathbb{R}^l$，则称对此问题 Slater 约束品性满足。

Slater 约束品性实际上是要求域 D 的相对内点中存在使得不等式约束严格成立的点。当一些不等式约束是仿射函数时，Slater 约束品性可以适当放宽，因为对线性不等式约束无需要求其存在严格可行点。

定理 3.12. 如果凸优化问题(3.31)满足 Slater 约束品性，则强对偶原理成立。

证明：注意到若 $p^* = -\infty$，由弱对偶性知结论成立，以下假定 $p^* > -\infty$。

第一步，我们先证明最简单的情况，然后逐渐扩展到更加一般的情况。假设域 D 内部非空，$\mathbf{A}$ 行满秩（否则可以去掉多余的线性等式约束）以及原始问题最优函数值 p^* 有限。

我们定义两个集合

$$\Omega_1 = \bigcup_{x\in D} \left\{ (\mathbf{u},\mathbf{v},t) \in R^{m\times p\times 1} \,\middle|\, f_i(\mathbf{x}) \le u_i, i = 1,\cdots,m, \mathbf{Ax} - \mathbf{b} = \mathbf{v}, f_0(\mathbf{x}) \le t \right\}$$

$$\Omega_2 = \left\{ (0,0,s) \in R^{m\times p\times 1} \,\middle|\, s < p^* \right\}$$

容易验证这两个集合都是凸集。接下来，我们用反证法证明 $\Omega_1 \bigcap \Omega_2$ 为空集。

如果我们假设 $(\mathbf{u},\mathbf{v},t) \in \Omega_1 \bigcap \Omega_2$，则 $(\mathbf{u},\mathbf{v},t) \in \Omega_2$ 可以推出 $\mathbf{u} = \mathbf{0}$，$\mathbf{v} = \mathbf{0}$ $t < p^*$。同时 $(\mathbf{u},\mathbf{v},t) \in \Omega_1$，则存在 $\mathbf{x}$ 使得 $f_i(\mathbf{x}) \le 0$，$\mathbf{Ax} - \mathbf{b} = \mathbf{0}$，$f_0(\mathbf{x}) \le t < p^*$。这和 p^* 是原问题可行目标的下确界相矛盾。

根据凸集分隔定理，我们知道存在一个超平面的法线方向 $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \upsilon) \ne 0$，和截距参数 α 使得

$$\tilde{\boldsymbol{\lambda}}^T \mathbf{u} + \tilde{\boldsymbol{\mu}}^T \mathbf{v} + \upsilon t \ge \alpha \quad \forall (\mathbf{u},\mathbf{v},t) \in \Omega_1 \tag{3.32}$$

$$\tilde{\boldsymbol{\lambda}}^T \mathbf{u} + \tilde{\boldsymbol{\mu}}^T \mathbf{v} + \upsilon t < \alpha \quad \forall (\mathbf{u},\mathbf{v},t) \in \Omega_2 \tag{3.33}$$

从（3.32）可以得知 $\tilde{\boldsymbol{\lambda}} \ge 0$，$\upsilon \ge 0$。否则可以取 u_i 和 t 为任意大的正实数以及 $\mathbf{v} = \mathbf{0}$，则 $\tilde{\boldsymbol{\lambda}}^T \mathbf{u} + \tilde{\boldsymbol{\mu}}^T \mathbf{v} + \upsilon t = \tilde{\boldsymbol{\lambda}}^T \mathbf{u} + \upsilon t$ 在 Ω_1 集合中可以趋向 $-\infty$，这和其有下界 α 矛盾。同时，（3.33）表明 $\upsilon t < \alpha \quad \forall t < p^*$，所以 $\upsilon p^* \le \alpha$。

因此，任取域 D 的内点 $\forall \mathbf{x} \in D$，对于 $(f_1(\mathbf{x}),\cdots, f_m(\mathbf{x}), \mathbf{Ax} - \mathbf{b}, f_0(\mathbf{x})) \in \Omega_1$，我们可以得到

$$\sum_{1\le i\le m} \tilde{\lambda}_i f_i(\mathbf{x}) + \tilde{\boldsymbol{\mu}}^T (\mathbf{Ax} - \mathbf{b}) + \upsilon f_0(\mathbf{x}) \ge \alpha \ge \upsilon p^*, \forall \mathbf{x} \in D \tag{3.34}$$

然后，我们用反证法证明$\upsilon \neq 0$。如果$\upsilon = 0$，代入（3.34）两端，我们可以得到

$$\sum_{1\le i\le m} \tilde{\lambda}_i f_i(\mathbf{x}) + \tilde{\boldsymbol{\mu}}^T \left(\mathbf{A}\mathbf{x} - \mathbf{b}\right) \ge 0,\ \forall \mathbf{x} \in D \tag{3.35}$$

对于满足 Slater 的点$\hat{\mathbf{x}} \in D,\ \mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$，我们可以从（3.35）得出$\sum_{1\le i\le m} \tilde{\lambda}_i f_i(\hat{\mathbf{x}}) \ge 0$。而此时$\tilde{\lambda}_i \ge 0,\ f_i(\hat{\mathbf{x}}) < 0, 1 \le i \le m$，因此$\tilde{\boldsymbol{\lambda}} = \mathbf{0}$。

而前述$\left(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \upsilon\right) \neq 0$，且$\tilde{\boldsymbol{\lambda}} = \mathbf{0}$，$\upsilon = 0$，则必有$\tilde{\boldsymbol{\mu}} \neq \mathbf{0}$。代入（3.35），对于$\forall \mathbf{x} \in D$，我们有$\tilde{\boldsymbol{\mu}}^T \left(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\right) \ge 0$。但同时对于满足 Slater 的点$\hat{\mathbf{x}} \in D,\ \mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$，则$\tilde{\boldsymbol{\mu}}^T \left(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\right) = 0$。而由于$\hat{\mathbf{x}} \in \operatorname{int} D, \mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$，必然有$\hat{\mathbf{x}} \in D$满足$\tilde{\boldsymbol{\mu}}^T \left(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\right) < 0$，除非$\tilde{\boldsymbol{\mu}}^T \mathbf{A} = 0$。而我们假设的$\mathbf{A}$行向量线性无关，也就是$\tilde{\boldsymbol{\mu}} = \mathbf{0}$。这和$\left(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \upsilon\right) = 0$矛盾。

因为$\upsilon \ge 0$，且$\upsilon \neq 0$，则一定有$\upsilon > 0$。我们可以从（3.34）式得到

$$\sum_{1\le i\le m} \frac{\tilde{\lambda}_i}{\upsilon} f_i(\mathbf{x}) + \frac{\tilde{\mu}^T}{\upsilon}\left(\mathbf{A}\mathbf{x} - \mathbf{b}\right) + f_0(\mathbf{x}) \ge p^*,\quad \forall \mathbf{x} \in D$$

进一步令$\lambda = \dfrac{\tilde{\lambda}}{\upsilon}$，$\mu = \dfrac{\overline{\mu}}{\upsilon}$，我们可以得到

$$d^* \ge g\left(\boldsymbol{\lambda}, \boldsymbol{\mu}\right) = \inf_{x\in D} L\left(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) \ge p^*$$

结合弱对偶性可知强对偶性成立。

第二步，我们扩展证明：对于域D的相对内点$\hat{\mathbf{x}} \in \operatorname{relint} D$，结论仍然成立。

$\hat{\mathbf{x}}$是域D的相对内点意味着域D的仿射包$\operatorname{affine} D$不等于全空间，因此，存在$\mathbf{Q} \in R^{n\times q}$（对应线性空间的基矩阵）满足

$$\operatorname{affine} D = \left\{\mathbf{x} \middle| \mathbf{x} = \hat{\mathbf{x}} + \mathbf{Q}\mathbf{y}, \mathbf{y} \in \mathbb{R}^q\right\}$$

令$\overline{f}_i(\mathbf{y}) = f_i\left(\hat{\mathbf{x}} + Q\mathbf{y}\right)$，$\overline{D} = \left\{y \in R^q \middle| \hat{x} + Qy \in D\right\}$，则$\hat{\mathbf{y}} = \mathbf{0} \in \mathbb{R}^q$是$\overline{D}$的内点，满足$\overline{f}_i(\hat{\mathbf{y}}) < 0$，$i = 1,...,m$，并且，$p^*$是以下问题的最优值

$$\min_{\mathbf{y}\in\overline{D}} \left\{\overline{f}_0(\mathbf{y}) \middle| \text{s.t. } \overline{f}_i(\mathbf{y}) \le 0, i = 1,...,m, \overline{\mathbf{A}}\mathbf{y} = 0\right\}$$

其中$\overline{\mathbf{A}} = \mathbf{A}\mathbf{Q}$。本质可以转为对行满秩的$\overline{\mathbf{A}}$来证明，过程类比第一步证明。

用$\overline{d}$表示上述问题的对偶问题的最大值，根据前面证明的结论，可知$\overline{d} = p^*$。

令 $\overline{g}(\boldsymbol{\lambda},\boldsymbol{\mu})=\inf_{\mathbf{y}\in D}\overline{f}_0(\mathbf{y})+\sum_{1\le i\le m}\lambda_i\overline{f}_i(\mathbf{y})+\boldsymbol{\mu}^T\overline{\mathbf{A}}\mathbf{y}$，则有

$$\begin{aligned}\overline{g}(\boldsymbol{\lambda},\boldsymbol{\mu})&=\inf_{\hat{x}+Qy\in D}f_0(\hat{\mathbf{x}}+\mathbf{Q}\mathbf{y})+\sum_{1\le i\le m}\lambda_i f_i(\hat{\mathbf{x}}+\mathbf{Q}\mathbf{y})+\boldsymbol{\mu}^T\left(\mathbf{A}\left(\hat{\mathbf{x}}+\mathbf{Q}\mathbf{y}\right)-\mathbf{b}\right)\\&=\inf_{x\in D}f_0(\mathbf{x})+\sum_{1\le i\le m}\lambda_i f_i(\mathbf{x})+\boldsymbol{\mu}^T\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)\\&=g(\boldsymbol{\lambda},\boldsymbol{\mu})\end{aligned}$$

因此

$$d^*=\sup_{\boldsymbol{\lambda}\ge\mathbf{0}}g(\boldsymbol{\lambda},\boldsymbol{\mu})=\sup_{\boldsymbol{\lambda}\ge\mathbf{0}}\overline{g}(\boldsymbol{\lambda},\boldsymbol{\mu})=\overline{d}=p^*$$

结论成立。

第三步，我们扩展证明：只需要对非线性不等式成为严格不等式，即线性不等式可以是等式，结论仍然成立。

假设用 $f_i(\mathbf{x})=\overline{\mathbf{a}}_i^T\mathbf{x}-\overline{b}_i, i=1,\ldots,\overline{m}\le m$ 表示在 $\hat{\mathbf{x}}$ 处起作用，即 $\overline{\mathbf{a}}_i^T\hat{\mathbf{x}}-\overline{b}_i=0$ 的线性不等式约束。考虑等式和不等式方程组

$$\overline{\mathbf{a}}_i^T\mathbf{d}<0, i=1,\ldots,\overline{m}, \mathbf{A}\mathbf{d}=0 \qquad (3.36)$$

如果方程组（3.36）存在解 $\mathbf{d}$，则我们取充分小的 $t>0$，令 $\hat{\mathbf{x}}'=\hat{\mathbf{x}}+t\mathbf{d}$，容易验证，$\hat{\mathbf{x}}'$ 是满足 Slater 约束品性的可行解，结论成立。

如果方程组（3.36）无解，我们定义两个集合

$$C=\left\{(\overline{y},\hat{y})\middle|\overline{y}_i=\overline{\mathbf{a}}_i^T\mathbf{d}, i=1,\ldots,\overline{m}, \hat{y}=\mathbf{A}\mathbf{d}, \mathbf{d}\in R^n\right\}$$

$$D=\left\{(\overline{z},\hat{z})\middle|\overline{z}_i<0, i=1,\ldots,\overline{m}, \hat{z}=0\right\}$$

容易验证这两个集合是凸集。方程组（3.36）无解代表这两个凸集无交点。根据凸集分隔定理，存在不全为零的 $(\boldsymbol{\lambda},\boldsymbol{\mu})$ 满足

$$\left(\sum_{i=1}^{\overline{m}}\lambda_i\overline{\mathbf{a}}_i^T+\boldsymbol{\mu}^T\mathbf{A}\right)\mathbf{d}\le\sum_{i=1}^{\overline{m}}\lambda_i\overline{z}_i, \forall\mathbf{d}\in R^n, \overline{z}_i<0$$

由上式可推出 $\forall i$，$\lambda_i\le 0$，$\sum_{i=1}^{\overline{m}}\lambda_i\overline{\mathbf{a}}_i^T+\boldsymbol{\mu}^T\mathbf{A}=0$。由于可假设 $\mathbf{A}$ 行满秩，必有某些 $\lambda_i\ne 0$，用 I_+ 表示这些 λ_i 的下标所构成的集合，则有 $\forall i$，$\lambda_i\le 0$，$\sum_{i\in I_+}\lambda_i\overline{\mathbf{a}}_i^T+\boldsymbol{\mu}^T\mathbf{A}=\mathbf{0}$。

任取不等于 $\hat{\mathbf{x}}$ 的可行解 $\mathbf{x}$，利用上式（分别乘 $\hat{\mathbf{x}}$ 和 $\mathbf{x}$ 再相减）又可得到

$$\forall i,\ \lambda_i\le 0,\ \sum_{i\in I_+}\lambda_i\left(\overline{\mathbf{a}}_i^T\mathbf{x}-\overline{b}_i\right)=0$$

由此可知，对任意可行解 $\mathbf{x}$ 均成立：$\forall i\in I_+$，$\overline{\mathbf{a}}_i^T\mathbf{x}=\overline{b}_i$。于是，可以把这些不等式约束视为等式约束。

上述过程表明，我们总可做到，或者获得一个满足 Slater 约束品性的可行解，或者减少不满足 Slater 约束品性的不等式的数目，如此继续，有限递降，最终一定可以满足 Slater 约束品性，完成证明。

第四步，我们扩展证明：矩阵 $\mathbf{A}$ 行满秩的假设可以去掉，结论仍然成立。

实际上，只需要证明如果不是行满秩（存在冗余等式），那么可以通过一个线性变换，使得约束行满秩，且经过变换之后 Lagrange 函数的最优值不变，即强对偶性不变。

假设 $\mathbf{A}$ 的行向量线性相关。由于 $\mathbf{Ax}=\mathbf{b}$ 有解 ，可以将其分为两组 $\mathbf{A}_1\mathbf{x}=\mathbf{b}_1$ 和 $\mathbf{A}_2\mathbf{x}=\mathbf{b}_2$，满足以下条件：1） $\mathbf{A}_1$ 行向量线性无关；2） $\mathbf{A}_2$ 的每行向量都可以表示成 $\mathbf{A}_1$ 的行向量的线性组合，即有矩阵 $\mathbf{P}$ 使得 $\mathbf{A}_2=\mathbf{PA}_1$，$\mathbf{b}_2=\mathbf{Pb}_1$。此时，原问题等价于满足前面假设的问题

$$\begin{aligned}\min_{\mathbf{x}} \quad & z=f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x})\le 0 \\ & \mathbf{A}_1\mathbf{x}=\mathbf{b}_1\end{aligned} \tag{3.37}$$

优化问题（3.37）的最小值显然等于原问题最小值 p^*，下面说明其对偶问题的最大值，记为 $\bar{d}$，也等于 d^*，从而完成定理的证明。令

$$g(\boldsymbol{\lambda},\bar{\boldsymbol{\mu}})=\inf_{\mathbf{x}\in D} f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i f_i(\mathbf{x})+\boldsymbol{\mu}^T(\mathbf{A}_1\mathbf{x}-\mathbf{b}_1)$$

则有

$$\begin{aligned}g(\boldsymbol{\lambda},\boldsymbol{\mu}) &= \inf_{\mathbf{x}\in D} f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i f_i(\mathbf{x})+\boldsymbol{\mu}_1^T(\mathbf{A}_1\mathbf{x}-\mathbf{b}_1)+\boldsymbol{\mu}_2^T(\mathbf{A}_2\mathbf{x}-\mathbf{b}_2) \\ &= \inf_{\mathbf{x}\in D} f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i f_i(\mathbf{x})+(\boldsymbol{\mu}_1^T+\boldsymbol{\mu}_2^T\mathbf{P})(\mathbf{A}_1\mathbf{x}-\mathbf{b}_1) \\ &= g(\boldsymbol{\lambda},\boldsymbol{\mu}_1^T+\boldsymbol{\mu}_2^T\mathbf{P})\end{aligned}$$

于是

$$d^*=\sup_{\boldsymbol{\lambda}\ge\mathbf{0}} g(\boldsymbol{\lambda},\boldsymbol{\mu})=\sup_{\boldsymbol{\lambda}\ge\mathbf{0}} g(\boldsymbol{\lambda},\boldsymbol{\mu}_1^T+\boldsymbol{\mu}_2^T\mathbf{P})=\sup_{\boldsymbol{\lambda}\ge\mathbf{0}} g(\boldsymbol{\lambda},\bar{\boldsymbol{\mu}})=\bar{d}$$

证明完成。□

需要注意的是，Slater 约束品性是凸优化问题强对偶原理成立的充分条件，而非必要条件。以下兹举数例讨论其它可能出现的情况。

例 3.7. 请证明下述非凸问题满足强对偶性。

$$\min_{x,y\in\mathbb{R}}\left\{x^2-y^2 \,\middle|\, \text{s.t. } x^2+y^2\le 1\right\}$$

解：不难看出原问题非凸优化问题，其最优值为 $p^*=-1$，在 $x=0$，$y=1$

以及 $x=0$， $y=-1$ 两个点取得。

而 Lagrange 函数为

$$L(x,y,\lambda)=x^2-y^2+\lambda\left(x^2+y^2-1\right)$$

其中 $\lambda\geq 0$。

Lagrange 对偶函数为

$$\begin{aligned}g(\lambda)&=\inf_{x,y}L(x,y,\lambda)=\inf_{x,y}\left[(1+\lambda)x^2+(\lambda-1)y^2-\lambda\right]\\&=\inf_{x=0,y}\left[(\lambda-1)y^2-\lambda\right]\\&=\begin{cases}-1 & \text{if } \lambda-1\leq 0,\text{ then } y \text{ should be } \pm 1\\ -\lambda<-1 & \text{if } \lambda-1>0,\text{ then } y \text{ should be } 0\\ -\infty & \text{otherwise}\end{cases}\end{aligned}$$

显然，对偶问题为

$$\min_{\lambda\in\mathbb{R}}\left\{-1\middle|\ \text{s.t. } 0\leq\lambda\leq 1\right\}$$

对偶问题的最优值为 $d^*=-1$。强对偶性成立。

例 3.8. 请证明下述二次约束二次规划问题满足强对偶性，但不一定满足 Slate 约束品性。

$$\min_{x\in\mathbb{R}}\left\{x^2\middle|\ \text{s.t. } x^2\leq 0\right\}$$

解：不难看出原问题是凸优化问题，其最优值为 $p^*=0$，在 $x=0$ 点取得。但显然不存在其他可行点 $\hat{x}$ 满足 $\hat{x}^2<0$。因此不满足 Slate 约束品性。

Lagrange 对偶函数为

$$L(x,\lambda)=x^2+\lambda x^2=(1+\lambda)x^2$$

其中 $\lambda\geq 0$。

Lagrange 对偶函数为

$$g(\lambda)=\inf_x L(x,\lambda)=\inf_x\left[(1+\lambda)x^2\right]=0$$

显然，对偶问题为

$$\min_{\lambda\in\mathbb{R}}\left\{0\middle|\ \text{s.t. } \lambda\geq 0\right\}$$

因此，对偶问题最优值为 $d^*=0$。强对偶性成立。

定理 3.13. 如果可微的凸优化问题（3.31）满足 Slater 约束品性，则 KKT 条件为全局最优解的充要条件。此时，KKT 条件里的 $\boldsymbol{\lambda}\in\mathbb{R}_+^m$， $\boldsymbol{\mu}\in\mathbb{R}^l$ 即为对应的 Lagrange 乘子，

证明：我们首先证明必要性。由于满足 Slater 约束品性，则强对偶性成立，

因此存在解 $\mathbf{x}^* \in \mathbb{R}^n$ 和对应的 $\boldsymbol{\lambda}^* \in \mathbb{R}_+^m$，$\boldsymbol{\mu}^* \in \mathbb{R}^l$ 满足

$$f_0(\mathbf{x}^*) = g\left(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) = \min_{\mathbf{x} \in D} f_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i f_i(\mathbf{x}) + \boldsymbol{\mu}^T \left(\mathbf{A}\mathbf{x} - \mathbf{b}\right)$$

$$\leq \min_{\mathbf{x} \in D} f_0(\mathbf{x}) = f_0(\mathbf{x}^*)$$

上式成立必然有原问题可行性条件和对偶问题可行性条件成立，互补松弛条件成立。因为 $\mathbf{x}^*$ 是 $\min_{\mathbf{x} \in D} L\left(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right)$ 的最优解，则根据定理 3.2，有稳定性条件成立。因此，KKT 条件为全局最优解的必要条件。

接下来，我们证明充分性。如果存在解 $\mathbf{x}^* \in \mathbb{R}^n$ 和对应的 $\boldsymbol{\lambda}^* \in \mathbb{R}_+^m$，$\boldsymbol{\mu}^* \in \mathbb{R}^l$ 满足 KKT 条件（3.19）-（3.22），则

$$\begin{aligned} g\left(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*\right) &= \min_{\mathbf{x} \in D} L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \\ &= f_0(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i^* f_i(\mathbf{x}^*) + \boldsymbol{\mu}^{*T} \left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) \\ &= f_0(\mathbf{x}^*) \end{aligned}$$

其中倒数第二个等式应用了稳定性条件，最后一个等式应用了互补松弛条件和对偶问题可行性条件。

这意味着对偶间隙为 0，解 $\mathbf{x}^*$ 和对应的 $\boldsymbol{\lambda}^* \in \mathbb{R}_+^m$，$\boldsymbol{\mu}^* \in \mathbb{R}^l$ 分别为原问题的对偶问题的最优解。因此，KKT 条件为全局最优解的充分条件。□

实际上，不可微的凸优化问题如果满足 Slater 约束品性，则考虑次梯度（Subgradient）的扩展 KKT 条件同样为全局最优解的充要条件。但相关证明要麻烦很多，本书不涉及。

为了确定非凸二次可微优化问题满足 KKT 条件的点是否是最优解，我们还有如下的定理。

定义 3.17. [临界锥] 对于可微非线性规划问题（3.1），如果存在解 $\mathbf{x}^*$ 和对应的 $\boldsymbol{\lambda}^* \in \mathbb{R}_+^m$，$\boldsymbol{\mu}^* \in \mathbb{R}^l$ 满足 KKT 条件（3.19）-（3.22），则定义临界锥为

$$C_\Omega\left(\mathbf{x}^*, \boldsymbol{\lambda}^*\right) = \left\{\mathbf{d} \in L_\Omega\left(\mathbf{x}^*\right) \mid \mathbf{d}^T \nabla f_i\left(\mathbf{x}^*\right) = 0, \forall i \in A(\mathbf{x}) \cap I,\ \text{且} \boldsymbol{\lambda}^* > 0\right\}$$

其中 $L_\Omega\left(\mathbf{x}^*\right)$ 为点 $\mathbf{x}^*$ 处的线性化可行方向锥。

定理 3.14. [可微约束优化问题的二阶必要条件] 在二次可微非线性规划问题（3.1）的局部最优解 $\mathbf{x}^*$ 处有 $L_\Omega\left(\mathbf{x}^*\right) = T_\Omega\left(\mathbf{x}^*\right)$，且存在 $\boldsymbol{\lambda} \in \mathbb{R}_+^m$，$\boldsymbol{\mu} \in \mathbb{R}^l$，使得 KKT 条件（3.19）-（3.22）成立，则对于任意 $\mathbf{d} \in C_\Omega\left(\mathbf{x}^*, \boldsymbol{\lambda}^*\right)$，有

$$\mathbf{d}^T\nabla_{\mathbf{xx}}L\left(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*\right)\mathbf{d}\geq 0$$

[可微约束优化问题的二阶充分条件] 在二次可微非线性规划问题（3.1）的可行解$\mathbf{x}^*$处存在$\boldsymbol{\lambda}^*\in\mathbb{R}_+^m$，$\boldsymbol{\mu}^*\in\mathbb{R}^l$，使得 KKT 条件（3.19）-（3.22）成立，若对于任意$\mathbf{d}\in C_\Omega\left(\mathbf{x}^*,\boldsymbol{\lambda}^*\right)$，$\mathbf{d}\neq\mathbf{0}$，有

$$\mathbf{d}^T\nabla_{\mathbf{xx}}L\left(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*\right)\mathbf{d}>0$$

则$\mathbf{x}^*$是二次可微非线性规划问题（3.1）的一个严格局部极小解。

证明：可以类似定理 3.3 的证明得证。本书从略。□

3.4 非线性规划的通用算法

求解非线性规划问题通常有两种思路。

第一种是逐点搜索算法，在每一步迭代过程中依次确定前进方向和前进步长，保证迭代过程中目标函数单调下降，保持在问题的可行域内进行，直到收敛。

算法 3.1. 通用逐点搜索算法

Step 1. 确定最初的探索点$\mathbf{x}^{(1)}\in\Omega\subseteq\mathbb{R}^n$，$k:=1$，进入 Step 2。

Step 2. 确定前进方向$\mathbf{d}^{(k)}$。确定前进步长$t^{(k)}$。前进至新的探索点$\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\in\Omega$。进入 Step 3。

Step 3. 如果$\left\|\mathbf{x}^{(k+1)}-\mathbf{x}^{(k)}\right\|\leq\varepsilon$，$\varepsilon>0$为预先设定的阈值，$\|\cdot\|$表示某种合适的范数。则已经收敛找到最优解，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

有的研究者在 Step 2 先判断是否收敛，如果还未收敛，再在 Step 3 决定如何前进。这样也是可以的，和上述算法几无差别。

第二种是区域压缩算法，在每一步迭代过程中缩小待考察域，保证迭代过程中解所在域单调减小，保持最优解在待考察域进行，直到收敛。

算法 3.2. 通用区域压缩算法

Step 1. 确定最初的待考察域$\Omega^{(1)}\subseteq\mathbb{R}^n$，$k:=1$，进入 Step 2。

Step 2. 确定新的待考察域$\Omega^{(k+1)}\subseteq\Omega^{(k)}$。进入 Step 3。

Step 3. 如果$\mathrm{size}\left(\Omega^{(k+1)}\right)\leq\varepsilon$，$\varepsilon>0$为预先设定的阈值，则已经收敛找到最优解，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

根据用于迭代的问题信息，我们可以将相关搜索算法大致分为三种：

第一种，零阶算法（Zero-Order Algorithms）。当目标函数的梯度和 Hessian 阵信息难以获得时，我们采用零阶算法求解。该类算法特别适合于没有给出显式函数形式，或者函数不可微且属于低维优化问题。代表性的零阶算法包括 DIRECT 算法 []。

第二种，一阶算法（First-Order Algorithms）利用目标函数的梯度信息来寻找前进方向；特别适用于大规模数据优化，机器学习、统计预测等问题。代表性的算法包括梯度下降法，最速下降法。

第三种，二阶算法（Second-Order Algorithms）利用目标函数的梯度和 Hessian 阵信息来寻找前进方向；特别适用于具有高精度需求的非高维优化问题。代表性的算法包括牛顿法，内点法，序列二次规划法（Sequential Quadratic Programming，SQP）。

实际上，我们还可以在搜索过程的不同阶段，灵活采用不同的算法，实现混合搜索算法。例如对于非凸的低维优化问题，首先利用 DIRECT 算法确定一些含有最优解的更小的可行区域，然后利用序列二次规划法在这些区域中迅速找到全局最优解。

为了衡量算法收敛的快慢，我们引入了如下的概念。

定义 3.18. 逐点搜索算法中 $\left\{\mathbf{x}^{(k)}\right\}$ 收敛到 $\mathbf{x}^*$，当且仅当

$$\lim_{k\to+\infty}\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\|=0 \tag{3.38}$$

其中 $\|\cdot\|$ 表示某种合适的范数。

如果存在正数 $\gamma>0$ 满足

$$\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\|\le\frac{O(1)}{k^{\gamma}}\left\|\mathbf{x}^{(0)}-\mathbf{x}^*\right\| \tag{3.39}$$

则称 $\left\{\mathbf{x}^{(k)}\right\}$ 以 γ 次幂算术收敛于 $\mathbf{x}^*$（converges arithmetically with power γ）。

如果存在正数 $\gamma>0$ 满足

$$\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\|\le\gamma^k\left\|\mathbf{x}^{(0)}-\mathbf{x}^*\right\| \text{ 或者 } \left\|\mathbf{x}^{(k+1)}-\mathbf{x}^*\right\|\le\gamma\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\| \tag{3.40}$$

则称 $\left\{\mathbf{x}^{(k)}\right\}$ 以速率 γ 线性收敛（或几何收敛）于 $\mathbf{x}^*$（linearly or geometrically with rate γ）。

如果存在正数 $\gamma\in[0,1)$，在 $\gamma\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\|<1$ 之后，满足

$$\left\|\mathbf{x}^{(k+1)}-\mathbf{x}^*\right\|\le\gamma\left\|\mathbf{x}^{(k)}-\mathbf{x}^*\right\|^2 \tag{3.41}$$

则称 $\left\{\mathbf{x}^{(k)}\right\}$ 以二次收敛于 $\mathbf{x}^*$（converges quadratically）。

类似的，我们可以用待考察域 $\Omega^{(k)}$ 的压缩比率来定义区域压缩算法的快慢，

本书不再赘述。

3.5 无约束优化问题的搜索算法

3.5.1 无约束优化问题逐点搜索算法的步长选择

当我们可以应用目标函数的梯度信息时，多采用逐点搜索算法进行求解。本节我们先讨论如何在确定前进方向后选择步长，下一节再讨论如何选择前进方向。

具体而言，确定步长有两类算法：步长精确搜索算法（Exact Search）和步长非精确搜索算法（Inexact Search）。其中，步长精确搜索算法力图在每一次迭代中，求解出让目标函数值一次性下降最多的步长。也即每一次迭代时求解一维优化问题

$$\min_{t>0} f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right) \tag{3.42}$$

为此，衍生出了 0.618 法，Fibonaci 法，折半搜索法，Newton 法等。除了 Newton 法之外，这些方法都采用区间压缩算法求解优化问题（3.42），常写成如下形式：

算法 3.3. 通用步长区域压缩算法

Step 1. 确定最初的待考察一维区间$\left[a^{(0)},b^{(0)}\right]$，保证优化问题（3.42）的最优解属于该区间，$k:=0$，进入 Step 2。

Step 2. 确定新的待考察一维区间$\left[a^{(k+1)},b^{(k+1)}\right]\subset\left[a^{(k)},b^{(k)}\right]$，保证优化问题（3.42）的最优解属于该区间。进入 Step 3。

Step 3. 如果$\left|b^{(k+1)}-a^{(k+1)}\right|\le\varepsilon$，$\varepsilon>0$ 为预先设定的阈值，则已经收敛找到最优步长，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

定义 3.19. 对于函数$f(t):\mathbb{R}\to\mathbb{R}$，如果在一维区间$[a,b]$存在$c\in(a,b)$，满足$f(t)$在区间$[a,c]$上单调下降，在区间$[c,b]$上单调上升，则称函数$f(t)$为一维区间$[a,b]$上的单谷函数，而区间$[a,b]$为函数$f(t)$的单谷区间。

对于函数$f(t):\mathbb{R}\to\mathbb{R}$，如果在一维区间$[a,b]$存在$c\in(a,b)$，满足$f(t)$在区间$[a,c]$上单调上升，在区间$[c,b]$上单调下降，则称函数$f(t)$为一维区间$[a,b]$上的单峰函数，而区间$[a,b]$为函数$f(t)$的单峰区间。

单谷函数和单峰函数统称为单模态函数（Unimodal Function），这类函数在给定区间上只包含一个最小值或最大值。

步长精确搜索算法一般假定已经确定了函数 $g(t)=f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right)$ 合适的单谷区间。

0.618 法（Golden Section Search Method）的基本想法是尽量每多计算出一个点的目标函数值，就能将待考察的一维区间长度压缩一个固定的比值 γ 。如图 3.3 所示，在最初的单谷区间 $\left[a^{(0)},b^{(0)}\right]$ 上，选取两点 $t_1^{(1)}$ 、$t_2^{(1)}$，如果 $g\left(t_1^{(1)}\right)<g\left(t_2^{(1)}\right)$，区间将被压缩为 $\left[a^{(0)},t_2^{(1)}\right]$；反之，区间被压缩为 $\left[t_1^{(1)},b^{(0)}\right]$。于是，$t_1^{(1)}$ 、$t_2^{(1)}$ 应该满足：

$$\frac{t_2^{(1)}-a^{(0)}}{b^{(0)}-a^{(0)}}=\frac{b^{(0)}-t_1^{(1)}}{b^{(0)}-a^{(0)}}=\gamma$$

假设新的区间选为 $\left[a^{(0)},t_2^{(1)}\right]$，我们再选取一个点 $t_1^{(2)}$，和上一次迭代的点 $t_1^{(1)}$ 处的目标函数值进行比较，如果 $g\left(t_2^{(1)}\right)<g\left(t_1^{(1)}\right)$，区间将被压缩为 $\left[a^{(0)},t_1^{(1)}\right]$；反之，区间被压缩为 $\left[t_1^{(2)},t_2^{(1)}\right]$。于是，$t_1^{(1)}$ 、$t_2^{(1)}$ 应该满足：

$$\frac{t_1^{(1)}-a^{(0)}}{t_2^{(1)}-a^{(0)}}=\frac{t_2^{(1)}-t_1^{(2)}}{t_2^{(1)}-a^{(0)}}=\gamma$$

不难推出，γ 应该满足

$$\gamma^2+\gamma-1=0$$

舍去负值解，我们得到 $\gamma=\dfrac{\sqrt{5}-1}{2}\approx 0.618$。所以，在 0.618 法的每一次迭代中，区间长度都被压缩为原来的 0.618 倍。为了表述更为简单清晰，我们可以采用每轮迭代计算两个点的方式（重复计算了一个点），将 0.618 法表述如下。

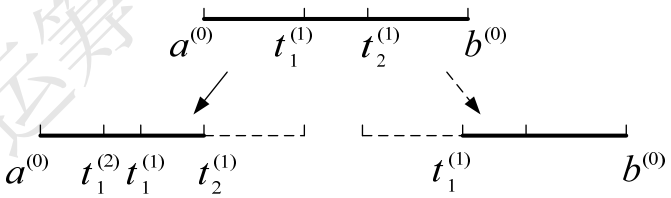


图 3.3　0.618 法示意图。

算法 3.4. 0.618 法

Step 1. 确定最初的待考察一维区间$\left[a^{(0)},b^{(0)}\right]$，确定误差阈值$\delta$及满足$0.618^{n-1}\left(b-a\right)\le\delta$的迭代次数$n$，$k:=1$，进入 Step 2。

Step 2. 选择区间$\left[a^{(k-1)},b^{(k-1)}\right]$内两个点

$$t_1^{(k)}=a^{(k-1)}+0.618\left(b^{(k-1)}-a^{(k-1)}\right),\quad t_2^{(k)}=b^{(k-1)}-0.618\left(b^{(k-1)}-a^{(k-1)}\right) \tag{3.43}$$

对比两点的目标函数值，如果$g\left(t_1^{(k)}\right)<g\left(t_2^{(k)}\right)$，函数$g(t)$的最小值一定在区间$\left[a^{(k-1)},t_2^{(k)}\right]$中，则令$a^{(k)}=a^{(k-1)}$，$b^{(k)}=t_2^{(k)}$；反之，函数$g(t)$的最小值一定在区间$\left[t_1^{(k)},b^{(k-1)}\right]$中，则令$a^{(k)}=t_1^{(k)}$，$b^{(k)}=b^{(k-1)}$；确定新的待考察一维区间$\left[a^{(k)},b^{(k)}\right]$。进入 Step 3。

Step 3. 如果$k=n$，则已经收敛到最优步长$t=0.5\left(a^{(n)}+b^{(n)}\right)$，停止。否则，$k:=k+1$，回到 Step 2，继续迭代。

Fibonaci 法（Fibonaci Search Method）的基本想法是分析计算过n个点目标函数值之后，能够将其压缩为一个单位长度的最大初始区间长度。如果我们用F_n表示该最大初始区间长度，则通过计算该区间中n个点目标函数值，可以把该初始区间压缩为单位长度区间。只要初始区间长度大于F_n，就无法凭借计算该区间中的n个点目标函数值做到这一点。

由于不计算函数值和只计算一点的函数值都不能压缩区间，所以$F_0=F_1=1$。

当$n=2$时，由于两个分点可以任意接近中点，所以能将两个单位的区间压缩为一个单位的区间，即$F_2=2$。可见，前三个数的关系是$F_2=F_1+F_0$。

如图 3.4 所示，在单谷区间$\left[a^{(k-1)},b^{(k-1)}\right]$上，选取两点$t_1^{(k)}$、$t_2^{(k)}$，如果$g\left(t_1^{(k)}\right)<g\left(t_2^{(k)}\right)$，区间将被压缩为$\left[a^{(k-1)},t_2^{(k)}\right]$；反之，区间被压缩为$\left[t_1^{(k)},b^{(k-1)}\right]$。于是，$t_1^{(1)}$、$t_2^{(1)}$应该满足：

$$\frac{t_2^{(k)}-a^{(k-1)}}{b^{(k-1)}-a^{(k-1)}}=\frac{b^{(k-1)}-t_1^{(k)}}{b^{(k-1)}-a^{(k-1)}}=\frac{F_{n-1}}{F_n}$$

类似于 0.618 法，实际上$t_1^{(k)}$、$t_2^{(k)}$中的一个点就是上一轮迭代中未被选为区间边界点的那个计算点，因此，一般情况下，F_n、F_{n-1}、F_{n-2}必须满足以下关系：

$$F_n = F_{n-1} + F_{n-2}$$

在 Fibonaci 数列中，有 $F_n = F_{n-1} + F_{n-2}$，可变形为 $\frac{F_n}{F_{n-1}} = 1 + \frac{F_{n-2}}{F_{n-1}}$。令 $n \to \infty$，定义 $F_* = \lim_{n\to\infty} \frac{F_{n-1}}{F_n}$，可得 $\frac{1}{F_*} = 1 + F_*$。舍去负值解，可得 $F_* = \frac{\sqrt{5}-1}{2} \approx 0.618$。所以，0.618 法实际上就是用 Fibonaci 法中分数数列的极限代替每个分数值所得到的方法。

在搜索过程中，0.618 法每一步迭代使区间缩小 0.618 倍，而 Fibonaci 法第 k 步迭代使区间缩小 $\left(F_{n-k}/F_{n-k+1} \to 0.618\right)$ 倍。对于足够复杂的问题，0.618 法和 Fibonaci 法的迭代次数相差无几。但是，Fibonaci 法每一步迭代的计算量要比 0.618 法更大，所以，0.618 法的运算速度应该会更快。

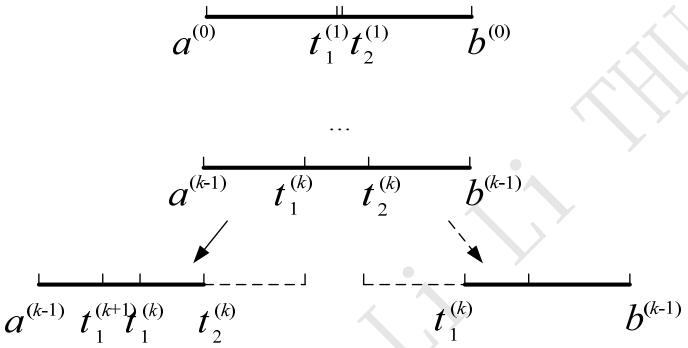


图 3.4　Fibonaci 法示意图。

算法 3.5. Fibonaci 法

Step 1. 确定最初的待考察一维区间 $\left[a^{(0)}, b^{(0)}\right]$，确定误差阈值 δ 及满足 $F_n^{-1}(b-a) \le \delta$ 的迭代次数 n，$k := 1$，进入 Step 2。

Step 2. 选择区间 $\left[a^{(k-1)}, b^{(k-1)}\right]$ 内两个点

$$\begin{aligned} t_1^{(k)} &= a^{(k-1)} + \left(F_{n-k}/F_{n-k+1}\right)\left(b^{(k-1)} - a^{(k-1)}\right), \\ t_2^{(k)} &= b^{(k-1)} - \left(F_{n-k}/F_{n-k+1}\right)\left(b^{(k-1)} - a^{(k-1)}\right), \end{aligned} \tag{3.44}$$

对比两点的目标函数值，如果$g\left(t_1^{(k)}\right)<g\left(t_2^{(k)}\right)$，函数$g(t)$的最小值一定在区间$\left[a^{(k-1)},t_2^{(k)}\right]$中，则令$a^{(k)}=a^{(k-1)}$，$b^{(k)}=t_2^{(k)}$；反之，函数$g(t)$的最小值一定在区间$\left[t_1^{(k)},b^{(k-1)}\right]$中，则令$a^{(k)}=t_1^{(k)}$，$b^{(k)}=b^{(k-1)}$；确定新的待考察一维区间$\left[a^{(k)},b^{(k)}\right]$。进入 Step 3。

Step 3. 如果$k=n$，则已经收敛到最优步长$t=0.5\left(a^{(n)}+b^{(n)}\right)$，停止。
否则，$k:=k+1$，回到 Step 2，继续迭代。

很显然，0.618 法和 Fibonaci 法不需要假设函数$g(t)$光滑，没有使用函数$g(t)$的导数信息，因此属于零阶算法。

与这两者不同，折半搜索法（Bisection Search Method）假设函数$g(t)$光滑，并使用函数$g(t)$的导数信息加速搜索。显然，函数$g(t)$的最优解t^*就是函数$g(t)$的导数函数$g'(t)$满足$g'(t)=0$的解。注意到这点，折半搜索法法表述如下。

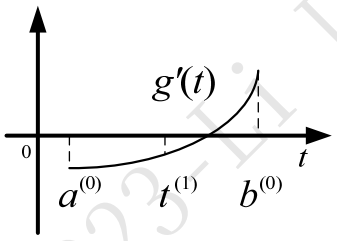


图 3.5　折半搜索法示意图。

算法 3.6. 折半搜索法

Step 1. 确定最初的待考察一维区间$\left[a^{(0)},b^{(0)}\right]$，确定误差阈值$\delta$及满足$0.5^{n-1}(b-a)\le\delta$的迭代次数$n$，$k:=1$，进入 Step 2。

Step 2. 选择区间$\left[a^{(k-1)},b^{(k-1)}\right]$的中点

$$t^{(k)}=a^{(k-1)}+0.5\left(b^{(k-1)}-a^{(k-1)}\right) \tag{3.45}$$

如果此点的目标函数导数值 $g'\left(t^{(k)}\right)<0$，函数 $g(t)$ 的最小值一定在区间 $\left[t^{(k-1)},b^{(k-1)}\right]$ 中，则令 $a^{(k)}=t^{(k)}$，$b^{(k)}=b^{(k-1)}$；反之，函数 $g(t)$ 的最小值一定在区间 $\left[a^{(k-1)},t^{(k)}\right]$ 中，则令 $a^{(k)}=a^{(k-1)}$，$b^{(k)}=t^{(k)}$；确定新的待考察一维区间 $\left[a^{(k)},b^{(k)}\right]$。进入 Step 3。

Step 3. 如果 $k=n$，则已经收敛到最优步长 $t=0.5\left(a^{(n)}+b^{(n)}\right)$，停止。否则，$k:=k+1$，回到 Step 2，继续迭代。

折半搜索法假设函数 $g(t)$ 光滑，并使用函数 $g(t)$ 的导数信息加速搜索，因此属于一阶算法。折半搜索法的区间压缩比等于 0.5，比 0.618 法和 Fibonaci 法区间压缩率更高，但实际效果还要取决于导数的计算量。

Newton 法同样考虑如何快速找到 $g'(t)=0$ 的解，其基本想法是：假定函数 $g(t)$ 二次可微，在给定的初始点 $t^{(0)}$ 用切线 $g'(t)=g'\left(t^{(0)}\right)+g''\left(t^{(0)}\right)\left(t-t^{(0)}\right)$ 近似 $g'(t)$，并求出此切线的零点作为 $t^{(1)}$。再在 $t^{(1)}$ 重复该过程。如图 3.6 所示，反复迭代，直到已经满足误差阈值要求或者 $\varphi'\left(t^{(k)}\right)=0$，终止迭代过程。Newton 法具体表述如下。

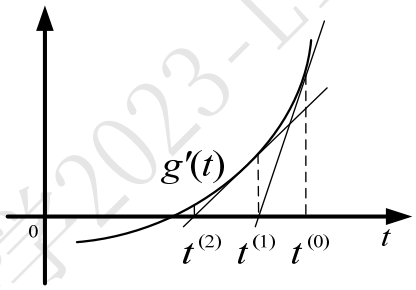


图 3.6　Newton 法示意图。

算法 3.7. Newton 法

Step 1. 确定最初的待考察一维区间 $\left[a^{(0)},b^{(0)}\right]$，确定误差阈值 δ，$k:=1$，选择区间中点 $t^{(0)}=0.5\left(a^{(0)}+b^{(0)}\right)$，进入 Step 2。

Step 2. 计算

$$t^{(k+1)}=t^{(k)}-\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)} \tag{3.46}$$

进入 Step 3。

Step 3. 如果点$t^{(k+1)}$的目标函数导数值$\left|g'\left(t^{(k+1)}\right)\right|\le\delta$，则已经收敛到最优步长，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

Newton 法并使用函数$g(t)$的二阶导数信息加速搜索，因此属于二阶算法。和上述算法不同，Newton 法属于逐点搜索算法而不是区间压缩算法。对于函数$g(t)$为凸函数，Newton 法一定收敛，且收敛速度比折半搜索收敛速度更快。限于篇幅，本书没有深入讨论$g''\left(t^{(k)}\right)=0$的特殊情况。

定理 3.15. 对于可微凸规划问题$\min\limits_t g(t)$，函数$g(\cdot):\mathbb{R}\to\mathbb{R}$二次可微且严格凸，算法 3.7 一定收敛。

证明：考虑函数$g(\cdot)$在点$t^{(k)}$的带 Peano 余项的二次 Taylor 展开，我们有

$$\begin{aligned}g\left(t^{(k+1)}\right)&=g\left(t^{(k)}-\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right)=g\left(t^{(k)}\right)+g'\left(t^{(k)}\right)\left(-\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right)\\&+\frac{1}{2}\left(-\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right)g''\left(t^{(k)}\right)\left(-\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right)+o\left(\left|\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right|^2\right)\\&=g\left(t^{(k)}\right)-\frac{1}{2}\frac{\left(g'\left(t^{(k)}\right)\right)^2}{g''\left(t^{(k)}\right)}+o\left(\left|\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right|^2\right)\end{aligned}$$

函数$g(\cdot)$是严格凸函数，在有限一维区间，存在$\varepsilon>0$，满足$\varepsilon\ge g''\left(t^{(k)}\right)>0$。因此，对于还未收敛时$\left|g'\left(t^{(k)}\right)\right|>\delta$，有

$$g\left(t^{(k+1)}\right)=g\left(t^{(k)}\right)-\frac{1}{2}\frac{\left(g'\left(t^{(k)}\right)\right)^2}{g''\left(t^{(k)}\right)}+o\left(\left|\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right|^2\right)$$

$$<g\left(t^{(k)}\right)-\frac{1}{2}\frac{\delta^2}{\varepsilon}+o\left(\left|\frac{g'\left(t^{(k)}\right)}{g''\left(t^{(k)}\right)}\right|^2\right)$$

$$<g\left(t^{(k)}\right)-c_0$$

其中 $c_0>0$ 为某固定常数。

考虑到有限一维区间中，函数 $g(\cdot)$ 必有下界。因此，算法 3.7 的迭代过程单调有界必然收敛。□

在很多问题中，我们无法确定合适的初始搜索区间。例如对于含有几百亿个参数的深度神经网络，我们很难确定何处为单谷区间。而且某些问题中，步长精确搜索算法可能花费过多的时间来求得最优步长。因此，我们多使用步长非精确搜索算法，在每一次迭代中，选定前进方向之后，找出让目标函数值下降的一个可行步长即进入下一次迭代。但若选取不合适的步长将会导致整个逐点搜索算法无法收敛到局部最优解。因此，步长非精确线搜索算法的步长选择还是要满足一定的要求，这些要求被称为线搜索准则（Line Search Conditions）。

定义 3.20. [Armijo 准则 Armijo（Sufficient Decrease）Condition] 设 $\mathbf{d}^{(k)}$ 是点 $\mathbf{x}^{(k)}$ 处的前进方向，若

$$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\le f\left(\mathbf{x}^{(k)}\right)+c_1t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)} \tag{3.47}$$

则称步长 $t^{(k)}$ 满足 Armijo 准则，其中 $c_1\in(0,1)$ 为预设常数。在实际应用中，参数 c_1 通常选为一个很小的正数，例如 $c_1=10^{-3}$。

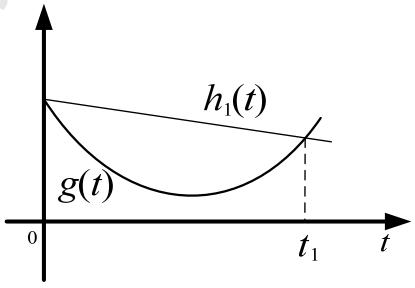


图 3.7　Armijo 准则示意图。

如图3.7所示，Armijo准则的几何意义是$h_1(t)=f\left(\mathbf{x}^{(k)}\right)+c_1t\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$为过$\left(0,f\left(\mathbf{x}^{(k)}\right)\right)$点的直线。注意到可行方向要求$\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}<0$，则该直线的斜率是负的。定义该直线和函数$g(t)=f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right)$在$t=t_1$处，产生$t=0$之后的第一个交点。则从点$\mathbf{x}^{(k)}$出发，沿着$\mathbf{d}^{(k)}$方向，我们应该选择步长在区间$[0,t_1]$之内。考虑到$t$在区间$(0,t_1)$之内，函数$g(t)=f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right)$位于直线$h_1(t)$的下方。因此，步长$t^{(k)}$在区间$(0,t_1)$之内保证前进到的新点$\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$时，目标函数值满足$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)<f\left(\mathbf{x}^{(k)}\right)$。

但是仅仅使用 Armijo 准则并不能保证迭代收敛。例如，步长$t^{(k)}=0$显然满足 Armijo 准则，但这样的步长是没有意义。我们多半将 Armijo 准则加上其它准则共同使用。

定义 3.21. [Armijo-Goldstein 准则，Goldstein 准则，Goldstein Condition] 设$\mathbf{d}^{(k)}$是点$\mathbf{x}^{(k)}$处的前进方向，若

$$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\le f\left(\mathbf{x}^{(k)}\right)+c_2t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$$

$$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\ge f\left(\mathbf{x}^{(k)}\right)+(1-c_2)t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\qquad(3.48)$$

则称步长$t^{(k)}$满足 Goldstein 准则，其中$c_2\in(0,0.5)$为预设常数。

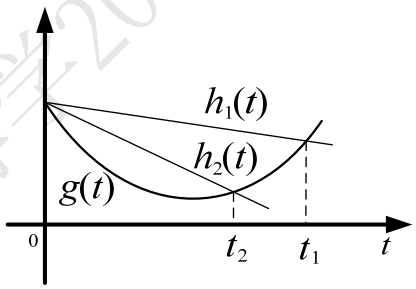


图 3.8　Goldstein 准则示意图。

Goldstein 准则考虑过$\left(0,f\left(\mathbf{x}^{(k)}\right)\right)$点的$h_1(t)=f\left(\mathbf{x}^{(k)}\right)+c_2t\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$

和 $h_1(t)=f\left(\mathbf{x}^{(k)}\right)+(1-c_2)t\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$ 这两条直线，定义他们在 $t=t_1$ 处和 $t=t_2$ 处，各自和函数 $g(t)=f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right)$ 产生 $t=0$ 之后的第一个交点。则从点 $\mathbf{x}^{(k)}$ 出发，沿着 $\mathbf{d}^{(k)}$ 方向，我们应该选择步长在区间 $[t_2,t_1]$ 之内。这样既保证目标函数值满足 $f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)<f\left(\mathbf{x}^{(k)}\right)$，又避免出现过小的步长。但最优解经常不在 Goldstein 准则所定义的区间 $[t_2,t_1]$ 中。

定义 3.22. [Armijo-Wolfe 准则，弱 Wolfe 准则，Strong Wolfe Condition] 设 $\mathbf{d}^{(k)}$ 是点 $\mathbf{x}^{(k)}$ 处的前进方向，若

$$\begin{aligned}&f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\le f\left(\mathbf{x}^{(k)}\right)+c_3t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\\&\nabla f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)^T\mathbf{d}^{(k)}\ge c_4\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\end{aligned}\tag{3.49}$$

则称步长 $t^{(k)}$ 满足 Weak Wolfe 准则，其中 $c_3,c_4\in(0,1)$ 为预设常数，且 $c_3<c_4$。在实际应用中，通常取 $c_4=0.9$。

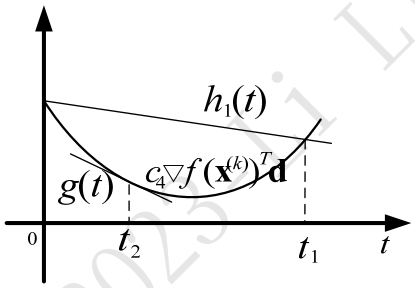


图 3.9　Wolfe 准则示意图。

（3.49）中第一个不等式就是 Armijo 准则，而第二个不等式也被称为曲率准则（Curvature Condition）。注意到函数 $g(t)$ 在 $t=t^{(k)}$ 处导数为 $g'(t)=\nabla f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)^T\mathbf{d}^{(k)}$。如图 3.9 所示，曲率准则实际要求被选择的步长 $t^{(k)}$ 满足 $g(t)$ 在 $t=t^{(k)}$ 处的切线斜率不能小于 $g'(0)$ 的 c_4 倍。假设 $t=0$ 之后的第一个满足曲率准则的步长为 $t=t_2$。进一步注意到，函数 $g(t)$ 的极小值点 $t=t^*$ 处，应该有 $g'(t^*)=\nabla f\left(\mathbf{x}^{(k)}+t^*\mathbf{d}^{(k)}\right)^T\mathbf{d}^{(k)}=0$。则 t^* 一定满足曲率准则。

进一步，我们可以较容易选择较小的c_3可使得$t<t_1$的步长都满足 Armijo 准则，且$t^*<t_1$。此时，t^*在所有满足 Wolfe 准则的点构成的区间$\left[t_2,t_1\right]$中。因此，选择 Wolfe 准则在绝大多数情况下可以得到包含步长搜索精确搜索算法最优解的步长区间。

我们还可以进一步改写曲率准则以便将区间$\left[t_2,t_1\right]$压缩得更小，并保持$t^*\in\left[t_2,t_1\right]$。

定义 3.23. [Armijo-Wolfe 准则，强 Wolfe 准则，Strong Wolfe Condition] 设$\mathbf{d}^{(k)}$是点$\mathbf{x}^{(k)}$处的前进方向，若

$$\begin{aligned} f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)&\le f\left(\mathbf{x}^{(k)}\right)+c_3t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\\ \left|\nabla f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)^T\mathbf{d}^{(k)}\right|&\le c_4\left|\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\right| \end{aligned} \tag{3.50}$$

则称步长$t^{(k)}$满足 Strong Wolfe 准则，其中$c_3,c_4\in(0,1)$为预设常数，且$c_3<c_4$。在实际应用中，通常取$c_4=0.9$。

基于上述准则，我们稍加修改可以构建多种步长非精确线搜索算法。

算法 3.8. Goldstein 搜索法 Goldstein Line Search

Step 1. 选定$\alpha>1$（一般取$\alpha=2$），$0<c_1<0.5<c_2<1$。选择$0=a^{(0)}<t^{(0)}<b^{(0)}=+\infty$，$k:=0$。进入 Step 2。

Step 2. 如果$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\le f\left(\mathbf{x}^{(k)}\right)+c_1t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$，进入 Step 3。

否则，令$a^{(k+1)}=a^{(k)}$，$b^{(k+1)}=t^{(k)}$；进入 Step 4。

Step 3. 如果$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\ge f\left(\mathbf{x}^{(k)}\right)+c_2t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$，已经找到合适的步长$t^{(k)}$，停止迭代。

否则，令$a^{(k+1)}=t^{(k)}$，$b^{(k+1)}=b^{(k)}$；进入 Step 4。

Step 4. 令

$$t^{(k+1)}=\begin{cases}\alpha t^{(k)} & \text{if } b^{(k+1)}=+\infty\\ 0.5\left(a^{(k+1)}+b^{(k+1)}\right) & \text{otherwise}\end{cases}$$

$k := k+1$，回到 Step 2，继续迭代。

算法 3.9. Weak Wolfe 搜索法 Weak Wolfe Line Search

Step 1. 选定 $\alpha>1$（一般取 $\alpha=2$），$0<c_3<0.5<c_4<1$。选择 $0=a^{(0)}<t^{(0)}<b^{(0)}=+\infty$，$t^{(0)}=1$，$k:=0$。进入 Step 2。

Step 2. 如果 $f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)>f\left(\mathbf{x}^{(k)}\right)+c_3t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$，令 $a^{(k+1)}=a^{(k)}$，$b^{(k+1)}=t^{(k)}$，进入 Step 4。

否则，进入 Step 3。

Step 3. 如果 $\nabla f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)^T\mathbf{d}^{(k)}<c_4\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$，令 $a^{(k+1)}=t^{(k)}$，$b^{(k+1)}=b^{(k)}$，进入 Step 4。

否则，已经找到合适步长步长 $t^{(k)}$，停止迭代。

Step 4. 令

$$t^{(k+1)}=\begin{cases}\alpha t^{(k)} & \text{if } b^{(k+1)}=+\infty\\ 0.5\left(a^{(k+1)}+b^{(k+1)}\right) & \text{otherwise}\end{cases}$$

$k := k+1$，回到 Step 2，继续迭代。

由于寻找一个满足 Armijo 准则的步长是比较容易的，特别对于凸优化问题，我们经常使用如下的回溯搜索法。

算法 3.10. 回溯搜索法 Backtracking Line Search

Step 1. 选定 $\alpha,\beta\in(0,1)$。选择 $t^{(0)}<+\infty$，$k:=0$，进入 Step 2。

Step 2. 如果 $f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)<f\left(\mathbf{x}^{(k)}\right)+\alpha t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$，则找到合适步长，停止。

否则，进入 Step 2。

Step 3. 令 $t^{(k+1)}=\beta t^{(k)}$，$k:=k+1$，回到 Step 2，继续迭代。

该算法的本质就是 Armijo 准则。其被称为回溯搜索法是因为我们从大到小来搜索可行的步长，可以确保找到的可行步长尽可能的大。注意到 $\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}<0$，我们对于足够小的 $t^{(k)}$，总能满足 Armijo 准则

$$f\left(\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}\right)\approx f\left(\mathbf{x}^{(k)}\right)+t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^{T}\mathbf{d}^{(k)}$$
$$\leq f\left(\mathbf{x}^{(k)}\right)+\alpha t^{(k)}\nabla f\left(\mathbf{x}^{(k)}\right)^{T}\mathbf{d}^{(k)}$$

因此，回溯搜索法必然在有限步内停止。在实际应用中我们通常也会设置一个最小可行步长，防止步长过小。

例 3.9. 请用不同步长搜索方法确定。

最后，我们探讨如何同时合适选择前进方向和步长，保证算法收敛。我们先针对弱 Wolfe 准则，证明如下 Zoutendijk 条件。实际上采用 Goldstein 准则也可推出类似的条件。

定理 3.16. [Zoutendijk 条件，Zoutendijk Condition] 对于可微非线性规划问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$，函数 $f\left(\cdot\right):\mathbb{R}^{n}\to\mathbb{R}$ 下有界、连续可微且梯度 Lipschitz 连续（Lipschitz Continuity）

$$\left\|\nabla f\left(\mathbf{x}\right)-\nabla f\left(\mathbf{y}\right)\right\|\leq L\left\|\mathbf{x}-\mathbf{y}\right\|,\ \ \forall\mathbf{x},\mathbf{y}\in\mathbb{R}^{n}$$

采用通用逐点搜索算法 3.1，如果迭代过程中步长选择满足弱 Wolfe 准则（3.49），则满足 Zoutendijk 条件

$$\sum_{j=0}^{+\infty}\cos^{2}\theta^{(j)}\left\|\nabla f\left(\mathbf{x}^{(j)}\right)\right\|_{2}^{2}<+\infty \tag{3.51}$$

其中 $\cos\theta^{(k)}$ 为搜索点 $\mathbf{x}^{(k)}$ 处，负梯度方向 $-\nabla f\left(\mathbf{x}^{(k)}\right)$ 和前进方向 $\mathbf{d}^{(k)}$ 夹角的余弦值

$$\cos\theta^{(k)}=\frac{-\nabla f\left(\mathbf{x}^{(k)}\right)^{T}\mathbf{d}^{(k)}}{\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_{2}\left\|\mathbf{d}^{(k)}\right\|_{2}} \tag{3.52}$$

证明：根据弱 Wolfe 准则（3.49），我们有

$$\left[\nabla f\left(\mathbf{x}^{(k+1)}\right)-\nabla f\left(\mathbf{x}^{(k)}\right)\right]^T\mathbf{d}^{(k)}\geq\left(c_4-1\right)\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}$$

根据 Cauchy-Schwarz 不等式和梯度 Lipschitz 连续性质，我们有

$$\begin{aligned}&\left[\nabla f\left(\mathbf{x}^{(k+1)}\right)-\nabla f\left(\mathbf{x}^{(k)}\right)\right]^T\mathbf{d}^{(k)}\\&\leq\left\|\nabla f\left(\mathbf{x}^{(k+1)}\right)-\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_2\left\|\mathbf{d}^{(k)}\right\|_2\leq t^{(k)}L\left\|\mathbf{d}^{(k)}\right\|_2^2\end{aligned}$$

综合上述两个不等式，有

$$t^{(k)}\geq\frac{c_4-1}{L}\frac{\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}}{\left\|\mathbf{d}^{(k)}\right\|_2^2}$$

同时由于是可行方向，$\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\mathbf{d}^{(k)}<0$，我们可以从 Armijo 准则得到

$$f\left(\mathbf{x}^{(k+1)}\right)\leq f\left(\mathbf{x}^{(k)}\right)+c_3\frac{c_4-1}{L}\frac{\left(\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\right)^2}{\left\|\mathbf{d}^{(k)}\right\|_2^2}$$

进一步考虑（3.52）给出的$\cos\theta^{(k)}$的定义，我们可以得到

$$f\left(\mathbf{x}^{(k+1)}\right)\leq f\left(\mathbf{x}^{(k)}\right)+c_3\frac{c_4-1}{L}\cos^2\theta^{(k)}\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_2^2$$

对上述系列不等式的前 k 项求和，我们得到

$$f\left(\mathbf{x}^{(k+1)}\right)\leq f\left(\mathbf{x}^{(0)}\right)+c_3\frac{c_4-1}{L}\sum_{j=0}^{k}\cos^2\theta^{(j)}\left\|\nabla f\left(\mathbf{x}^{(j)}\right)\right\|_2^2$$

因为我们假设函数 $f(\cdot)$ 下有界，且由 $0<c_3<c_4<1$ 可知 $0<c_3<c_4<1$，$k\to+\infty$ 时，不等式（3.51）必然成立。□

定理 3.17. [线搜索算法的收敛性] 采用通用逐点搜索算法 3.1 时，如果按（3.52）定义搜索点 $\mathbf{x}^{(k)}$ 处，负梯度方向 $-\nabla f\left(\mathbf{x}^{(k)}\right)$ 和前进方向 $\mathbf{d}^{(k)}$ 夹角为 $\theta^{(k)}$，并假设对任意第 k 次迭代，存在常数 $\gamma>0$，满足

$$\theta^{(k)}<\frac{\pi}{2}-\gamma\tag{3.53}$$

则在满足 Zoutendijk 条件成立的条件下，存在

$$\lim_{k\to+\infty}\nabla f\left(\mathbf{x}^{(k)}\right)=\mathbf{0}$$

证明：我们用反证法证明。假设结论不成立，即存在下标子列 $\{k_l\}$ 和正常数 $\delta>0$，使得

$$\left\|\nabla f\left(\mathbf{x}^{(k_l)}\right)\right\|_2\geq\delta,\ l=1,...$$

根据 $\theta^{(k)}$ 的定义和假设，有

$$\cos\theta^{(k)} > \sin\gamma > 0\ ,\ \ \forall k$$

那么考虑下标子列$\{k_l\}$对应的求和式，我们可以得到

$$\sum_{j=0}^{+\infty}\cos^2\theta^{(j)}\left\|\nabla f\left(\mathbf{x}^{(j)}\right)\right\|_2^2 \geq \sum_{l=0}^{+\infty}\cos^2\theta^{(k_l)}\left\|\nabla f\left(\mathbf{x}^{(k_l)}\right)\right\|_2^2$$

$$\geq \sum_{l=0}^{+\infty}\left(\sin^2\gamma\right)\delta^2 \to +\infty$$

这与定理 3.16 矛盾。因此，必然有结论成立。□

从几何意义上来看，定理 3.17 要求是搜索过程中，每一步的前进方向$\mathbf{d}^{(k)}$和负梯度方向$-\nabla f\left(\mathbf{x}^{(k)}\right)$不能趋于正交即可。这一要求显然是合理的。因为当前进方向$\mathbf{d}^{(k)}$和和负梯度方向$-\nabla f\left(\mathbf{x}^{(k)}\right)$正交时，根据在当前搜索点出的一阶泰勒展开，无论步长如何选，目标函数值几乎保持不变，无法有效的前进到新的搜索点使得目标函数实质性下降。

3.5.2 无约束优化问题的一阶和二阶逐点搜索算法

本节我们讨论如何选择前进方向，并和上节的结论一起完成逐点搜索算法。

梯度下降（Gradient Descent）算法是最简单也最常用的无约束可微优化问题的一阶逐点搜索算法，其基本想法是基于定义 3.8 选择当前点处的函数负梯度方向作为可行方向，选择合适的步长进行逐点迭代搜索，并根据定理 3.2 确定停止条件。该算法常写为如下形式

算法 3.11. 梯度下降算法

Step 1. 确定最初的探索点$\mathbf{x}^{(1)}\in\mathbb{R}^n$，$k:=1$，进入 Step 2。

Step 2. 确定前进方向$\mathbf{d}^{(k)}=-\nabla f\left(\mathbf{x}^{(k)}\right)$。使用精确搜索法或者非精确搜索法确定合适前进步长$t^{(k)}$。前进至新的探索点$\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$。进入 Step 3。

Step 3. 如果$\left\|\nabla f\left(\mathbf{x}^{(k+1)}\right)\right\|\leq\varepsilon$，$\varepsilon>0$为预先设定的阈值，$\|\cdot\|$表示某种合适的范数（一般选$l_2$范数）。则已经收敛找到最优解，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

定理 3.18. 对于可微非线性规划问题$\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$，函数$f\left(\cdot\right):\mathbb{R}^n\to\mathbb{R}$，如果该问题有有限的最优值且可以取到，梯度下降算法 3.11 可以在有限时间内收敛。

证明：根据定义 3.8，当$\nabla f\left(\mathbf{x}^{(k)}\right)\neq\mathbf{0}$时，选择当前点处的函数负梯度方向作

为前进方向，显然存在常数 $\gamma = \frac{\pi}{4}$，满足（3.53）。根据定理 3.17，算法 3.11 可以在有限时间内收敛。□

定理 3.19 对于可微非线性规划问题 $\min_{\mathbf{x}} f(\mathbf{x})$，函数 $f(\cdot):\mathbb{R}^n \to \mathbb{R}$，采用梯度下降算法 3.11，并以步长精确搜索算法确定每次迭代时的步长，则算法相邻两次迭代的搜索方向正交。

证明：根据算法 3.11，$f\left(\mathbf{x}^{(k+1)}\right)$ 是在点 $\mathbf{x}^{(k)}$ 处，沿方向 $\mathbf{d}^{(k)}$ 进行步长精确搜索法得到的解 $t^{(k)}$，因此满足

$$t^{(k)} = \arg\min_{t>0} f\left(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}\right)$$

由于函数 $f(\cdot)$ 可微，我们 $t^{(k)}$ 满足

$$\frac{f\left(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}\right)}{\partial t} = 0$$

则

$$\frac{f\left(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}\right)}{\partial \mathbf{x}^T}\frac{d\left(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}\right)}{dt} = \nabla f\left(\mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}\right)^T \mathbf{d}^{(k)} = 0$$

也即

$$\nabla f\left(\mathbf{x}^{(k+1)}\right)^T \mathbf{d}^{(k)} = \mathbf{d}^{(k+1)T}\mathbf{d}^{(k)} = 0$$

因此，算法相邻两次迭代的搜索方向正交。□

定理 3.18 说明梯度下降算法 3.3 能收敛，但定理 3.19 说明该算法的效率可能不高。为此，研究者提出了多种方式来修正前进方向，加速收敛。

最速下降算法（Steepest Descent Method）的基本想法是用当前搜索点的目标函数的一阶 Taylor 展开来近似目标函数

$$f\left(\mathbf{x}^{(k)} + \mathbf{y}\right) \approx f\left(\mathbf{x}^{(k)}\right) + \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{y}$$

其中 $\mathbf{y} \subseteq \mathbb{R}^n$。

然后选择合适范数，使得在该范数定义的单位步长下，近似目标函数的值下降最大的方向作为前件方向。

算法 3.12. 最速下降算法

Step 1. 确定最初的探索点 $\mathbf{x}^{(1)} \in \mathbb{R}^n$，$k := 1$，进入 Step 2。

Step 2. 确定负梯度方向 $-\nabla f\left(\mathbf{x}^{(k)}\right)$。选定合适的范数，求解下述优化问题

$$\begin{aligned}\min_{\mathbf{d}\in\mathbb{R}^n}\quad & \nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}\\ \text{s.t.}\quad & \|\mathbf{d}\|\le 1\end{aligned}\tag{3.54}$$

以最优解 $\mathbf{d}^{(k)}\in\mathbb{R}^n$ 作为前进方向（也有研究者将约束条件写作 $\|\mathbf{d}\|=1$，不难证明两种形式等价）。

Step 3. 使用精确搜索法或者非精确搜索法确定合适前进步长 $t^{(k)}$。前进至新的探索点 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$。进入 Step 3。

Step 4. 如果 $\left\|\nabla f\left(\mathbf{x}^{(k+1)}\right)\right\|\le\varepsilon$，$\varepsilon>0$ 为预先设定的阈值，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。则已经收敛找到最优解，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

我们可以证明最速下降算法 3.12 每次前进方向都是可行方向。

定理 3.20. 优化问题（3.54）的最优解 $\mathbf{d}^{(k)}$ 是可微非线性规划问题 $\min_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的下降方向。

证明：如果在点 $\mathbf{x}^{(k)}$ 处尚未达到局部最优解，则 $\nabla f\left(\mathbf{x}^{(k+1)}\right)\neq\mathbf{0}$，则必然存在 $\tilde{\mathbf{d}}\in\mathbb{R}^n$，满足 $\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\tilde{\mathbf{d}}\neq 0$。否则，由于 $\tilde{\mathbf{d}}$ 的任意性，必然有 $\nabla f\left(\mathbf{x}^{(k+1)}\right)=\mathbf{0}$，产生矛盾。

注意到根据范数的对称性，$\hat{\mathbf{d}}=\tilde{\mathbf{d}}/\|\tilde{\mathbf{d}}\|$ 和 $-\hat{\mathbf{d}}=-\tilde{\mathbf{d}}/\|\tilde{\mathbf{d}}\|$ 都是优化问题（3.54）的可行解，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。$\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\tilde{\mathbf{d}}\neq 0$，则要么 $\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\hat{\mathbf{d}}<0$，要么 $\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\left(-\hat{\mathbf{d}}\right)<0$。因此，优化问题（3.54）的最优解最优解 $\mathbf{d}^{(k)}$ 必然满足

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\le\min\left\{\nabla f\left(\mathbf{x}^{(k)}\right)^T\hat{\mathbf{d}},\nabla f\left(\mathbf{x}^{(k)}\right)^T\left(-\hat{\mathbf{d}}\right)\right\}<0$$

因此，$\mathbf{d}^{(k)}$ 是可微非线性规划问题 $\min_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的下降方向。□

类似于定理 3.16，我们可以进一步证明最速下降算法 3.9 可以在有限时间内收敛。最后，由于定理 3.2 保证，停止条件只在最优解附近成立。限于篇幅，本书不再给出证明。

定理 3.21. 当我们选择 l_1 范数定义优化问题（3.54）时，求出前进方向 $\mathbf{d}^{(k)}$ 为

$$\mathbf{d}_i^{(k)}=\begin{cases}\operatorname{sgn}\left(-\dfrac{\partial f(\mathbf{x}^{(k)})}{\partial x_i}\right) & \text{if } \left|\dfrac{\partial f(\mathbf{x}^{(k)})}{\partial x_i}\right|=\left\|\nabla f(\mathbf{x}^{(k)})\right\|_\infty \\ 0 & \text{if } \left|\dfrac{\partial f(\mathbf{x}^{(k)})}{\partial x_i}\right|\neq\left\|\nabla f(\mathbf{x}^{(k)})\right\|_\infty\end{cases} \tag{3.55}$$

如果有多个 x_i 满足 $\left|\dfrac{\partial f(\mathbf{x}^{(k)})}{\partial x_i}\right|=\left\|\nabla f(\mathbf{x}^{(k)})\right\|_\infty$，则任意保留其中一项 $\mathbf{d}_i^{(k)}=\operatorname{sgn}\left(-\dfrac{\partial f(\mathbf{x}^{(k)})}{\partial x_i}\right)$，其余的置为 $\mathbf{d}_i^{(k)}=0$。前进方向 $\mathbf{d}^{(k)}$ 满足 $\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}=-\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_\infty$。

当我们选择 l_∞ 范数定义优化问题（3.54）时，求出前进方向为

$$\mathbf{d}_i^{(k)}=\operatorname{sgn}\left(-\frac{\partial f\left(\mathbf{x}^{(k)}\right)}{\partial x_i}\right),\quad \forall i \tag{3.56}$$

满足 $\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}=-\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_1$。

当我们选择 l_2 范数定义优化问题（3.54）时，求出的前进方向为 $\mathbf{d}^{(k)}=-\nabla f\left(\mathbf{x}^{(k)}\right)$。

当我们选择 l_p 范数定义优化问题（3.54）时，求出前进方向 $\mathbf{d}^{(k)}$ 为

$$\mathbf{d}_i^{(k)}=\operatorname{sgn}\left(-\frac{\partial f\left(\mathbf{x}^{(k)}\right)}{\partial x_i}\right)\left|\frac{\partial f\left(\mathbf{x}^{(k)}\right)}{\partial x_i}\right|^{q-1}\left(\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_q\right)^{-\frac{q}{p}} \tag{3.57}$$

其中 $q=\dfrac{p}{p-1}$，满足 $\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}=-\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_q$。

当我们选择定义成如下形式的 $\mathbf{P}$ 范数，矩阵 $\mathbf{P}\in S_{++}^{n\times n}$ 为特定的正定矩阵

$$\|\mathbf{d}\|_{\mathbf{P}}=\left(\mathbf{d}^T\mathbf{P}\mathbf{d}\right)^{\frac{1}{2}} \tag{3.58}$$

定义优化问题（3.54）时，求出的前进方向为

$$\mathbf{d}^{(k)}=-\mathbf{P}^{-1}\nabla f\left(\mathbf{x}^{(k)}\right) \tag{3.59}$$

证明：当我们选择 l_1 范数和 l_∞ 范数定义优化问题（3.54）时，上述结论显然。

当我们选择 l_2 范数定义优化问题（3.54）时，根据 Cauchy-Schwarz 不等式有

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}\geq-\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_2\left\|\mathbf{d}^{(k)}\right\|_2=-\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_2$$

上述不等式等号成立时，必然有

$$\mathbf{d}^{(k)}=\frac{-\nabla f\left(\mathbf{x}^{(k)}\right)}{\left\|\nabla f\left(\mathbf{x}^{(k)}\right)\right\|_2}$$

这样我们就直接得到了此时问题（3.54）的解析解，而该解等价于 $\mathbf{d}^{(k)}=-\nabla f\left(\mathbf{x}^{(k)}\right)$。

当我们选择 l_p 范数定义优化问题（3.54）时，可以根据 Hölder 不等式，类似得出结论。

当我们选择 $\mathbf{P}$ 范数定义优化问题（3.54）时，可以通过换元 $\hat{\mathbf{d}}=\mathbf{P}^{\frac{1}{2}}\mathbf{d}$，得到采用 l_2 范数定义优化问题

$$\begin{aligned}\min_{\hat{\mathbf{d}}\in\mathbb{R}^n}\quad & \nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{P}^{\frac{1}{2}}\hat{\mathbf{d}}\\ \text{s.t.}\quad & \left\|\hat{\mathbf{d}}\right\|_2\le 1\end{aligned}$$

根据上面关于采用 l_2 范数定义优化问题的讨论，该换元后问题最优解应该为 $\hat{\mathbf{d}}=-\mathbf{P}^{\frac{1}{2}}\nabla f\left(\mathbf{x}^{(k)}\right)$，原问题最优解为 $\mathbf{d}^{(k)}=-\mathbf{P}^{-\frac{1}{2}}\mathbf{P}^{-\frac{1}{2}}\nabla f\left(\mathbf{x}^{(k)}\right)=-\mathbf{P}^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)$。□

一般情况下，在最优解附近时，算法相邻两次迭代的搜索方向正交会使得搜索效率变低。为了进一步加速收敛，研究者提出了共轭梯度算法（Conjugate Gradient）。我们首先针对严格凸二次规划问题引入共轭梯度算法，然后将其扩展为针对一般可微优化问题的形式。

定义 3.24. 对于实对称矩阵 $\mathbf{A}\in\mathbb{R}^{n\times n}$，如果非零向量 $\mathbf{p},\mathbf{q}\in\mathbb{R}^n$ 满足

$$\mathbf{p}^T\mathbf{A}\mathbf{q}=0 \tag{3.60}$$

则称向量 $\mathbf{p},\mathbf{q}$ 对于矩阵 $\mathbf{A}$ 相互共轭。如果存在一组非零向量 $\mathbf{q}_i\in\mathbb{R}^n$，$i=1,...,n$，满足 $\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j=0$，$i\neq j$，$i,j=1,...,n$。则称 $\{\mathbf{p}_i\}$，$i=1,...,n$，为矩阵 $\mathbf{A}$ 的一组共轭基底。

很显然，矩阵 $\mathbf{A}$ 为单位阵时，共轭关系就是正交关系。因此，我们将共轭概念视为正交概念的一种扩展。此外，共轭关系还有如下的性质。

定理 3.22. 对于正定矩阵 $\mathbf{A}\in S_{++}^{n\times n}$，若 $\{\mathbf{p}_i\}$，$i=1,...,n$，为矩阵 $\mathbf{A}$ 的一组共轭基底，则 $\{\mathbf{p}_i\}$ 为一组线性无关的列向量。

证明：我们用反证法证明。假设这些列向量线性相关，则存在一组不全为 0

的实数$\left\{\lambda_j, j=1,...,n\right\}$使得$\sum_{j=1}^{n}\lambda_j\mathbf{p}_j=\mathbf{0}$。

则任意取$\mathbf{p}_i$，我们有$\mathbf{p}_i^T\mathbf{A}\left(\sum_{j=1}^{n}\lambda_j\mathbf{p}_j\right)=\sum_{j=1}^{n}\lambda_j\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j=0$。根据共轭的定义，我们有$\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j=0$，$i\neq j$。则此时，$\sum_{j=1}^{n}\lambda_j\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j=\lambda_i\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i=0$。注意到矩阵$\mathbf{A}$为正定，$\mathbf{p}_i\neq\mathbf{0}$，$\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i>0$，因此必然有$\lambda_i=0$。

考虑到我们是任意取$\mathbf{q}_i$，显然可推知$\lambda_i=0$，$i=1,...,n$。这与我们假设的一组不全为 0 的实数矛盾。因此结论成立。□

根据定理 3.22，正定矩阵$\mathbf{A}\in S_{++}^{n\times n}$最多只有$n$个共轭基底，因为整个$\mathbb{R}^n$空间最多只有$n$个线性无关的列向量基地。

应用定理 3.2，我们可以将求解线性方程组$\mathbf{Ax}=\mathbf{b}$的问题，等价转化为严格凸二次规划问题

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})=\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}-\mathbf{b}^T\mathbf{x} \tag{3.61}$$

$\mathbf{A}\in S_{++}^{n\times n}$，$\mathbf{b}\in\mathbb{R}^n$。考虑到上述性质，该问题的最优解$\mathbf{x}^*$应该可以表示为矩阵$\mathbf{A}$的一组共轭基底$\left\{\mathbf{p}_i\right\}$的线性加权和

$$\mathbf{x}^*=\alpha_1\mathbf{p}_1+...+\alpha_n\mathbf{p}_n$$

任意取$\mathbf{p}_i$，我们有

$$\mathbf{p}_i^T\mathbf{b}=\mathbf{p}_i^T\mathbf{A}\mathbf{x}^*=\mathbf{p}_i^T\mathbf{A}\left(\sum_{j=1}^{n}\alpha_j\mathbf{p}_j\right)=\sum_{j=1}^{n}\alpha_j\mathbf{p}_i^T\mathbf{A}\mathbf{p}_j=\alpha_i\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i$$

则

$$\alpha_i=\frac{\mathbf{p}_i^T\mathbf{b}}{\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i}$$

进而有

$$\mathbf{x}^*=\alpha_1\mathbf{p}_1+...+\alpha_n\mathbf{p}_n=\sum_{i=1}^{n}\frac{\mathbf{p}_i^T\mathbf{b}\mathbf{p}_i}{\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i}$$

定理 3.23. 考虑严格凸二次规划问题（3.61），$\left\{\mathbf{p}_i\right\}$为矩阵$\mathbf{A}$的一组共轭基底，从任意可行点$\mathbf{x}^{(1)}$出发，根据公式

$$\alpha^{(k)}=-\frac{\left(\mathbf{A}\mathbf{x}^{(k)}-\mathbf{b}\right)^T\mathbf{p}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k},\quad \mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{p}_k$$

能在最多 n 次迭代后找到最优解。

证明：假设该问题的最优解 $\mathbf{x}^*$ 和点 $\mathbf{x}^{(1)}$ 的差值表述为共轭基底 $\{\mathbf{p}_i\}$ 的线性加权和如下

$$\mathbf{x}^* - \mathbf{x}^{(1)} = \alpha_1\mathbf{p}_1 + \ldots + \alpha_n\mathbf{p}_n$$

则任意取 $\mathbf{p}_i$，$i=1,\ldots,n$，我们有

$$\mathbf{p}_i^T\mathbf{A}\left(\mathbf{x}^* - \mathbf{x}^{(1)}\right) = \mathbf{p}_i^T\mathbf{A}\left(\alpha_1\mathbf{p}_1 + \ldots + \alpha_n\mathbf{p}_n\right) = \alpha_i\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i$$

于是

$$\alpha_i = \frac{\mathbf{p}_i^T\mathbf{A}\left(\mathbf{x}^* - \mathbf{x}^{(1)}\right)}{\mathbf{p}_i^T\mathbf{A}\mathbf{p}_i}$$

另一方面

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(1)} = \alpha_1\mathbf{p}_1 + \ldots + \alpha_k\mathbf{p}_k$$

注意到 $\{\mathbf{p}_i\}$ 为矩阵 $\mathbf{A}$ 的一组共轭基底，则有

$$\mathbf{p}_{k+1}^T\mathbf{A}\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(1)}\right) = \mathbf{p}_{k+1}^T\mathbf{A}\left(\alpha_1\mathbf{p}_1 + \ldots + \alpha_k\mathbf{p}_k\right) = 0$$

代入上式有

$$\alpha_{k+1} = \frac{\mathbf{p}_k^T\mathbf{A}\left(\mathbf{x}^* - \mathbf{x}^{(1)}\right)}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k} = \frac{\mathbf{p}_k^T\mathbf{A}\left(\mathbf{x}^* - \mathbf{x}^{(1)}\right) - \mathbf{p}_k^T\mathbf{A}\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(1)}\right)}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}$$

$$= \frac{\mathbf{p}_k^T\mathbf{A}\left(\mathbf{x}^* - \mathbf{x}^{(k+1)}\right)}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k} = \frac{\mathbf{p}_k^T\mathbf{A}\mathbf{x}^* - \mathbf{p}_k^T\mathbf{A}\mathbf{x}^{(k+1)}}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k} = \frac{\mathbf{p}_k^T\mathbf{b} - \mathbf{p}_k^T\mathbf{A}\mathbf{x}^{(k+1)}}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}$$

$$= \frac{\left(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}\right)^T\mathbf{p}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k} = \alpha^{(k+1)}$$

两者显然是一样的，结论得证。□

考虑到直接用 Gram-Schmidt 正交化方法来寻找一组共轭基底 $\{\mathbf{p}_i\}$ 比较麻烦，研究者提出来依次迭代求取共轭基底 $\{\mathbf{p}_i\}$ 的算法。同时，如果最优解 $\mathbf{x}^*$ 仅由部分共轭基底 $\{\mathbf{p}_i\}$ 的线性加权和表示，则我们可能在迭代过程中提前结束，无需算出所有基底。相关共轭梯度算法如下。

算法 3.13. 针对严格凸二次规划问题的共轭梯度算法

Step 1. 考虑严格凸二次规划问题（3.61），从任意可行点 $\mathbf{x}^{(1)}$ 出发，如果 $\left\|\mathbf{A}\mathbf{x}^{(1)}-\mathbf{b}\right\|\le\varepsilon$，$\varepsilon>0$ 为预先设定的阈值，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。则已经收敛找到最优解，停止。

否则，$\mathbf{d}^{(1)}=-\nabla f\left(\mathbf{x}^{(1)}\right)=\mathbf{b}-\mathbf{A}\mathbf{x}^{(1)}$，$k:=1$，进入 Step 2。

Step 2. 前进至新的探索点

$$\alpha^{(k)}=-\frac{\left(\mathbf{A}\mathbf{x}^{(k)}-\mathbf{b}\right)^T\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)}},\quad \mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{d}^{(k)} \tag{3.62}$$

确定新的前进方向

$$\beta^{(k)}=\frac{\left(\mathbf{A}\mathbf{x}^{(k+1)}-\mathbf{b}\right)^T\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)}},\quad \mathbf{d}^{(k+1)}=\mathbf{b}-\mathbf{A}\mathbf{x}^{(k+1)}+\beta^{(k)}\mathbf{d}^{(k)} \tag{3.63}$$

进入 Step 3。

Step 3. 如果 $\left\|\mathbf{A}\mathbf{x}^{(k+1)}-\mathbf{b}\right\|\le\varepsilon$，$\varepsilon>0$ 为预先设定的阈值，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。则已经收敛找到最优解，停止。

否则，$k:=k+1$，回到 Step 2，继续迭代。

定理 3.24. 算法 3.13 本质是构建了一组共轭基底，最多在 n 次迭代后找到最优解。

证明：利用数学归纳法，我们很容易证明 $\left\{\mathbf{A}\mathbf{x}^{(i)}-\mathbf{b}\right\}$ 和 $\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,n$，各自张成了整个 $\mathbb{R}^n$ 空间的一组线性无关的基底。

接下里，我们利用数学归纳法证明 $\left(\mathbf{A}\mathbf{x}^{(k+1)}-\mathbf{b}\right)^T\mathbf{d}^{(i)}=0$，$i=1,...,k$，$k\le n-1$。容易验证 $k=1$ 时，该论断成立。假设对于 $k=\hat{k}$ 时，该论断都成立。

根据（3.62）中 $\mathbf{x}^{(\hat{k}+1)}=\mathbf{x}^{(\hat{k})}+\alpha^{(\hat{k})}\mathbf{d}^{(\hat{k})}$，我们有

$$\begin{aligned}\left(\mathbf{A}\mathbf{x}^{(\hat{k}+1)}-\mathbf{b}\right)^T\mathbf{d}^{(i)}&=\left[\mathbf{A}\left(\mathbf{x}^{(\hat{k})}+\alpha^{(\hat{k})}\mathbf{d}^{(\hat{k})}\right)-\mathbf{b}\right]^T\mathbf{d}^{(i)}\\&=\left(\mathbf{A}\mathbf{x}^{(\hat{k})}-\mathbf{b}\right)^T\mathbf{d}^{(i)}+\alpha^{(\hat{k})}\mathbf{d}^{(\hat{k})T}\mathbf{A}\mathbf{d}^{(i)}\end{aligned}$$

当 $i=1,...,k-1$ 时，根据数学归纳法假设，上式第一项等于 0，根据共轭性质，$\mathbf{d}^{(\hat{k})T}\mathbf{A}\mathbf{d}^{(i)}=0$，上式第二项等于 0。

当 $i=k$ 时，根据 $\alpha^{(k)}$ 的定义，有 $\left(\mathbf{A}\mathbf{x}^{(k+1)}-\mathbf{b}\right)^T\mathbf{d}^{(k)}=0$。

因此，对于 $k=\hat{k}+1$ 时，该论断也成立。故而，根据数学归纳法，

$\left(\mathbf{A}\mathbf{x}^{(k+1)}-\mathbf{b}\right)$ 不在 $\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,k$，构成的线性子空间中。

最后，我们用数学归纳法证明 $\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,n$，构成一组关于矩阵 $\mathbf{A}$ 的共轭基底。易知 $i=1$ 时，该论断成立。假设已经证明 $\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,k$，构成一组关于矩阵 $\mathbf{A}$ 的共轭列向量。

根据公式（3.63），我们有

$$\mathbf{d}^{(k+1)T}\mathbf{A}\mathbf{d}^{(i)}=\left(\mathbf{b}-\mathbf{A}\mathbf{x}^{(k+1)}\right)^T\mathbf{A}\mathbf{d}^{(i)}+\beta^{(k)T}\mathbf{d}^{(k)}\mathbf{A}\mathbf{d}^{(i)},\quad i=1,...,k$$

当 $i<k$ 时，根据归纳法假设，$\mathbf{d}^{(k)}$ 和 $\mathbf{d}^{(i)}$ 关于矩阵 $\mathbf{A}$ 共轭，则 $\mathbf{d}^{(k)}\mathbf{A}\mathbf{d}^{(i)}=0$。同时，$\mathbf{A}\mathbf{d}^{(i)}$ 在 $\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,k$，构成的线性子空间中。根据我们上面证明的结论，有 $\left(\mathbf{b}-\mathbf{A}\mathbf{x}^{(k)}\right)^T\mathbf{A}\mathbf{d}^{(i)}=0$，$i=1,...,k-1$。因此，$i<k$ 时，$\mathbf{d}^{(k+1)T}\mathbf{A}\mathbf{d}^{(i)}=0$。

当 $i=k$ 时，根据 $\beta^{(k)}$ 的定义，$\mathbf{d}^{(k+1)T}\mathbf{A}\mathbf{d}^{(i)}=0$。

故而

$$\mathbf{d}^{(k+1)T}\mathbf{A}\mathbf{d}^{(i)}=0,\quad i=1,...,k$$

根据数学归纳法，$\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,k+1$，也构成一组关于矩阵 $\mathbf{A}$ 的共轭列向量。进一步可知，$\left\{\mathbf{d}^{(i)}\right\}$，$i=1,...,n$，构成一组关于矩阵 $\mathbf{A}$ 的共轭基底。且 $\alpha^{(k)}$ 的定义也满足定理 3.23 的形式，因此根据定理 3.23，可知结论成立。□

我们还可以进一步证明上述迭代过程中 $\alpha^{(k)}$ 就是每次迭代时，选定前进方向之后，用精确步长搜索算法解出的步长 $\alpha^{(k)}=\arg\min_{\alpha>0} f\left(\mathbf{x}^{(k)}+\alpha\mathbf{d}^{(k)}\right)$。类似于定理 3.19 的证明，我们可以得到

$$\frac{f\left(\mathbf{x}^{(k)}+t\mathbf{d}^{(k)}\right)}{\partial t}=\nabla f\left(\mathbf{x}^{(k+1)}\right)^T\mathbf{d}^{(k)}=0=\left(\mathbf{b}-\mathbf{A}\mathbf{x}^{(k+1)}\right)^T\mathbf{d}^{(k)}$$

结合 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{d}^{(k)}$，即有

$$\left[\mathbf{b}-\mathbf{A}\left(\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{d}^{(k)}\right)\right]^T\mathbf{d}^{(k)}=0$$

稍加整理有

$$\alpha^{(k)}=-\frac{\left(\mathbf{A}\mathbf{x}^{(k)}-\mathbf{b}\right)^T\mathbf{d}^{(k)}}{\mathbf{d}^{(k)T}\mathbf{A}\mathbf{d}^{(k)}}$$

正是（3.62）中给出的 $\alpha^{(k)}$ 计算公式。

实际上，我们可以将针对严格凸二次规划问题的共轭梯度算法推导到针对可微优化问题 $\min_{\mathbf{x}} f\left(\mathbf{x}\right)$ 的共轭梯度算法。

算法 3.14. 针对可微优化问题的共轭梯度算法

Step 1. 从任意可行点 $\mathbf{x}^{(1)}$ 出发，如果 $\left\|\nabla f\left(\mathbf{x}^{(1)}\right)\right\| \le \varepsilon$，$\varepsilon > 0$ 为预先设定的阈值，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。则已经收敛找到最优解，停止。

否则，$\mathbf{d}^{(1)} = -\nabla f\left(\mathbf{x}^{(1)}\right)$，$k := 1$，进入 Step 2。

Step 2. 使用精确步长搜索算法确定步长 $t^{(k)} = \arg\min_{t>0} f\left(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}\right)$，前进到新的探索点 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$。进入 Step 3。

Step 3. 如果 $\left\|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}\right\| \le \varepsilon$，$\varepsilon > 0$ 为预先设定的阈值，$\|\cdot\|$ 表示某种合适的范数（一般选 l_2 范数）。则已经收敛找到最优解，停止。

否则，任选下述一种形式确定新的方向修正系数：

Fletcher-Reeves 形式

$$\alpha^{(k+1)} = \frac{\nabla f\left(\mathbf{x}^{(k+1)}\right)^T \nabla f\left(\mathbf{x}^{(k+1)}\right)}{\nabla f\left(\mathbf{x}^{(k)}\right)^T \nabla f\left(\mathbf{x}^{(k)}\right)} \tag{3.64}$$

Polak-Ribière-Polyak 形式

$$\alpha^{(k+1)} = \frac{\nabla f\left(\mathbf{x}^{(k+1)}\right)^T \left[\nabla f\left(\mathbf{x}^{(k+1)}\right) - \nabla f\left(\mathbf{x}^{(k)}\right)\right]}{\nabla f\left(\mathbf{x}^{(k)}\right)^T \nabla f\left(\mathbf{x}^{(k)}\right)} \tag{3.65}$$

Beale-Sorenson/Hestenes-Stiefel 形式

$$\alpha^{(k+1)} = \frac{\nabla f\left(\mathbf{x}^{(k+1)}\right)^T \left[\nabla f\left(\mathbf{x}^{(k+1)}\right) - \nabla f\left(\mathbf{x}^{(k)}\right)\right]}{\left[\nabla f\left(\mathbf{x}^{(k+1)}\right) - \nabla f\left(\mathbf{x}^{(k)}\right)\right]^T \mathbf{d}^{(k)}} \tag{3.66}$$

Dai-Yuan 形式

$$\alpha^{(k+1)} = \frac{\nabla f\left(\mathbf{x}^{(k+1)}\right)^T \nabla f\left(\mathbf{x}^{(k+1)}\right)}{\left[\nabla f\left(\mathbf{x}^{(k+1)}\right) - \nabla f\left(\mathbf{x}^{(k)}\right)\right]^T \mathbf{d}^{(k)}} \tag{3.67}$$

令 $\mathbf{d}^{(k+1)} = -\nabla f\left(\mathbf{x}^{(k+1)}\right) + \alpha^{(k+1)}\mathbf{d}^{(k)}$，$k := k+1$，回到 Step 2，继续迭代。

定理 3.25. 算法 3.13 和算法 3.14 中每次迭代时都选择了下降方向。

证明：对于算法 3.13 而言，在第 k 次迭代时

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k)} = \left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)^T \left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} + \beta^{(k)}\mathbf{d}^{(k-1)}\right)$$
$$= -\nabla f\left(\mathbf{x}^{(k)}\right)^T \nabla f\left(\mathbf{x}^{(k)}\right) + \beta^{(k)} \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k-1)}$$

由于采用精确搜索算法，类似于定理 3.19 的证明，我们可以得到

$$\frac{f\left(\mathbf{x}^{(k-1)} + t\mathbf{d}^{(k-1)}\right)}{\partial t} = \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k-1)} = 0$$

因此，没有达到收敛前，有

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k)} = \left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)^T \left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} + \beta^{(k)}\mathbf{d}^{(k-1)}\right)$$
$$= -\nabla f\left(\mathbf{x}^{(k)}\right)^T \nabla f\left(\mathbf{x}^{(k)}\right) < -\delta^2$$

类似的，对于算法 3.14 而言，在第 k 次迭代时

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k)} = -\nabla f\left(\mathbf{x}^{(k)}\right)^T \nabla f\left(\mathbf{x}^{(k)}\right) + \alpha^{(k)} \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k-1)} < -\delta^2$$

因此，两个算法的 $\mathbf{d}^{(k)}$，$k = 1,...,n$，均是下降方向。□

最速下降算法 3.12 和共轭梯度算法 3.13 及算法 3.14 只考虑目标函数在搜索点 $\mathbf{x}^{(k)}$ 处的一阶梯度信息，如果进一步考虑目标函数在搜索点 $\mathbf{x}^{(k)}$ 处的 Hessian 阵信息，则可以采用 Newton 算法（Newton's Method）。该算法可以视为将针对一维函数的算法 3.7 扩展到高维的形式。

定义 3.25. [Newton 方向] 对于可微非线性规划问题 $\min_{\mathbf{x}} f\left(\mathbf{x}\right)$，函数 $f\left(\cdot\right): \mathbb{R}^n \to \mathbb{R}$ 二次可微且严格凸，则称如下 $\mathbf{d}^{(k)}$ 为点 $\mathbf{x}^{(k)}$ 处的 Newton 方向

$$\mathbf{d}^{(k)} = -\left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1} \nabla f\left(\mathbf{x}^{(k)}\right) \tag{3.68}$$

我们可以从三种角度来引出 Newton 方向。

第一种角度，假设我们采用 $\mathbf{P}$ 范数定义的最速下降方向，只不过我们每次令 $\mathbf{P} = \nabla^2 f\left(\mathbf{x}^{(k)}\right)$，直接得到解析解（3.68）。

第二种角度，假设我们采用目标函数在点 $\mathbf{x}^{(k)}$ 处的二阶 Talyor 展开作为近似目标函数，令步长 $t^{(k)} = 1$，则有

$$f\left(\mathbf{x}^{(k)} + \mathbf{d}^{(k)}\right) \approx f\left(\mathbf{x}^{(k)}\right) + \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k)} + \frac{1}{2}\mathbf{d}^{(k)T} \nabla^2 f\left(\mathbf{x}^{(k)}\right) \mathbf{d}^{(k)} \tag{3.69}$$

求解近似目标函数的极小值

$$\min_{\mathbf{d} \in \mathbb{R}^n} f\left(\mathbf{x}^{(k)}\right) + \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \nabla^2 f\left(\mathbf{x}^{(k)}\right) \mathbf{d}$$

直接得到解析解（3.68）。

第三种角度，假设我们采用目标函数的梯度 $\nabla f\left(\mathbf{x}^{(k)}\right)$ 在点 $\mathbf{x}^{(k)}$ 处的一阶 Talyor 展开作为梯度 $\nabla f\left(\mathbf{x}^{(k)}\right)$ 的近似

$$\nabla f\left(\mathbf{x}^{(k)}+\mathbf{d}^{(k)}\right)\approx\nabla f\left(\mathbf{x}^{(k)}\right)+\nabla^2 f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)} \tag{3.70}$$

目标函数的极小值处应该有 $\nabla f\left(\mathbf{x}^{(k)}\right)=\mathbf{0}$，则直接得到解析解（3.68）。

定理 3.26. 对于可微非线性规划问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$，函数 $f\left(\cdot\right):\mathbb{R}^n\to\mathbb{R}$ 二次可微且严格凸，则点 $\mathbf{x}^{(k)}$ 处的 Newton 方向（3.72）为下降方向。

证明：注意到

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T\mathbf{d}^{(k)}=-\nabla f\left(\mathbf{x}^{(k)}\right)^T\left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)<0$$

结论成立。□

为了衡量采用 Newton 方向之后，每一步

定义 3.26. [Newton 减量（Newton Decrement）] 对于可微非线性规划问题 $\min\limits_{\mathbf{x}} f\left(\mathbf{x}\right)$，函数 $f\left(\cdot\right):\mathbb{R}^n\to\mathbb{R}$ 二次可微且严格凸，若在点 $\mathbf{x}^{(k)}$ 处采用 Newton 方向（3.72），且令步长 $t^{(k)}=1$，带入目标函数在点 $\mathbf{x}^{(k)}$ 处的二阶 Talyor 展开作为近似（3.73），则称如下 $\lambda^{(k)}$ 为 Newton 减量

$$f\left(\mathbf{x}^{(k)}+\mathbf{d}^{(k)}\right)-f\left(\mathbf{x}^{(k)}\right)\approx\left(\nabla f\left(\mathbf{x}^{(k)}\right)^T\left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)\right)^{\frac{1}{2}}=\lambda^{(k)} \tag{3.71}$$

可以用来衡量前进之后，目标函数值的变化。

算法 3.15. 无约束优化问题的 Newton 算法

Step 1. 确定最初的探索点 $\mathbf{x}^{(1)}\in\mathbb{R}^n$，$k:=1$，进入 Step 2。

Step 2. 计算 $\lambda^{(k)}=\left(\nabla f\left(\mathbf{x}^{(k)}\right)^T\left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)\right)^{\frac{1}{2}}$。如果

$\left(\lambda^{(k)}\right)^2\le 2\varepsilon$，$\varepsilon>0$ 为预先设定的阈值。则已经收敛找到最优解，停止。

否则，进入 Step 3。

Step 3. 确定 Newton 方向 $\mathbf{d}^{(k)}=-\left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1}\nabla f\left(\mathbf{x}^{(k)}\right)$。

如果是纯 Newton 算法（Pure Newton's method），选择前进步长 $t^{(k)}=1$。

如果是阻尼 Newton 算法（Damped Newton's method），使用回溯搜索法确定合适前进步长 $t^{(k)}$。

前进至新的探索点 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$。$k:=k+1$，回到 Step 2，继续迭代。

定理 3.27. 对于严格凸二次规划问题 $\min\limits_{\mathbf{x}\in\mathbb{R}^n} f\left(\mathbf{x}\right)=\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{q}^T\mathbf{x}+r$，$\mathbf{P}\in S_{++}^{n\times n}$，$\mathbf{q}\in\mathbb{R}^n$，$r\in\mathbb{R}$，从任意可行点 $\mathbf{x}^{(0)}$ 出发，采用纯 Newton 算法，则可以迭代一次即找到最优解。

证明：假设问题最优解在 $\mathbf{x}^*$ 处取得，根据定理 3.2，此时有 $\nabla f\left(\mathbf{x}^*\right)=\mathbf{0}$，也即 $\mathbf{P}\mathbf{x}^*+\mathbf{q}=\mathbf{0}$。则 $\mathbf{x}^*=-\mathbf{P}^{-1}\mathbf{q}$。而对于任意可行解 $\mathbf{x}$，有 $\nabla^2 f\left(\mathbf{x}\right)=\mathbf{P}$。

注意到 $\nabla f\left(\mathbf{x}^{(0)}\right)=\mathbf{P}\mathbf{x}^{(0)}+\mathbf{q}$。我们得到 $\mathbf{x}^{(0)}$ 处的 Newton 方向满足

$$-\left(\nabla^2 f\left(\mathbf{x}^{(0)}\right)\right)^{-1}\nabla f\left(\mathbf{x}^{(0)}\right)=-\mathbf{P}^{-1}\left(\mathbf{P}\mathbf{x}^{(0)}+\mathbf{q}\right)=-\mathbf{x}^{(0)}+\mathbf{x}^*$$

因此，Newton 方向直接从点 $\mathbf{x}^{(0)}$ 指向最优解 $\mathbf{x}^*$。选择前进步长 $t^{(k)}=1$。直接可以找到最优解。□

采用 Newton 方向有可能快速的的收敛的局部最优解，但每次迭代都要计算新的 Hessian 阵信息，可能会花费更多的时间。

例 3.10. 请用不同搜索算法确定。

3.6 约束优化问题的搜索算法

3.6.1　等式约束优化问题的搜索算法

本书主要讨论如下的仿射等式约束凸优化问题

$$\begin{aligned}\min_{\mathbf{x}} \quad & z=f_0(\mathbf{x})\\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}=\mathbf{b}\end{aligned} \tag{3.72}$$

其中 $\mathbf{A}\in\mathbb{R}^{l\times n}$，$\operatorname{rank}(\mathbf{A})=l<n$，$\mathbf{b}\in\mathbb{R}^{l}$。目标函数 $f_0(\cdot):\mathbb{R}^n\to\mathbb{R}$ 为具有连续二阶导数的凸函数。且该问题最优值 $p^*=f(\mathbf{x}^*)=\min\{f(\mathbf{x})\mid \text{s.t. } \mathbf{A}\mathbf{x}=\mathbf{b}\}$ 为有限值。

具体求解时，我们一般有四种选择。

第一种选择，利用换元法。令 $\mathbf{x}=\mathbf{F}\mathbf{y}+\hat{\mathbf{x}}$，$\mathbf{y}\in\mathbb{R}^{n-l}$。将优化问题（3.72）等价转化为无约束优化问题

$$\min_{\mathbf{y}} z=f_0(\mathbf{F}\mathbf{y}+\hat{\mathbf{x}}) \tag{3.73}$$

其中矩阵 $\mathbf{F}\in\mathbb{R}^{n\times(n-l)}$ 为矩阵 $\mathbf{A}$ 的零空间 $\{\mathbf{x}\mid \mathbf{A}\mathbf{x}=\mathbf{0}\}$ 的基矩阵，满足 $\mathbf{A}\mathbf{F}=\mathbf{0}$。也即有 $\operatorname{rank}(\mathbf{F})=n-l$，$\{\mathbf{x}\mid \mathbf{A}\mathbf{x}=\mathbf{0}\}=\{\mathbf{F}\mathbf{y}\mid \mathbf{y}\in\mathbb{R}^{n-l}\}$。$\hat{\mathbf{x}}\in\mathbb{R}^n$ 满足 $\mathbf{A}\hat{\mathbf{x}}=\mathbf{b}$，则 $\{\mathbf{x}\mid \mathbf{A}\mathbf{x}=\mathbf{b}\}=\{\mathbf{F}\mathbf{y}+\hat{\mathbf{x}}\mid \mathbf{y}\in\mathbb{R}^{n-l}\}$。

第二种选择，利用 KKT 条件。优化问题（3.72）只有仿射约束条件，Slater 条件成立，根据 KKT 条件，存在 $\mathbf{x}^*\in\mathbb{R}^n$ 和 $\mathbf{u}^*\in\mathbb{R}^m$ 满足

$$\nabla f_0(\mathbf{x}^*)+\mathbf{A}^T\mathbf{u}^*=0,\quad \mathbf{A}\mathbf{x}^*=\mathbf{b} \tag{3.74}$$

对于某些比较简单的问题，我们可以直接根据（3.74）求解。

例 3.11. 请求解等式约束的凸二次规划问题

$$\begin{aligned}\min_{\mathbf{x}} \quad & z=\frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{q}^T\mathbf{x}+r\\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}=\mathbf{b}\end{aligned} \tag{3.75}$$

其中 $\mathbf{P}\in S_+^{n\times n}$，$\mathbf{q}\in\mathbb{R}^n$，$r\in\mathbb{R}$，$\mathbf{A}\in\mathbb{R}^{l\times n}$，$\operatorname{rank}(\mathbf{A})=l<n$，$\mathbf{b}\in\mathbb{R}^l$。

解：根据 KKT 条件，我们可以得到

$$\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}^*\\ \mathbf{u}^*\end{bmatrix}=\begin{bmatrix}-\mathbf{q}\\ \mathbf{b}\end{bmatrix} \tag{3.76}$$

则该问题的解分为三种情况。

第一种情况，如果 $\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}$ 非奇异。我们可进一步证明，这等价于 $\begin{bmatrix}\mathbf{P}\\ \mathbf{A}\end{bmatrix}$ 满

秩；或者等价的 $\mathbf{P}+\mathbf{A}^T\mathbf{A}\in S_{++}^{n\times n}$。此时，该问题有唯一解 $\begin{bmatrix}\mathbf{x}^*\\ \mathbf{u}^*\end{bmatrix}=\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}^{-1}\begin{bmatrix}-\mathbf{q}\\ \mathbf{b}\end{bmatrix}$。

第二种情况，如果 $\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}$ 奇异，但方程组（3.76）有解，则该问题存在无穷多个最优解。

第三种情况，如果方程组（3.76）无解，则该问题存在无界的解。根据 Farkas 引理，此时存在 $\mathbf{w}\in\mathbb{R}^n$，$\mathbf{v}\in\mathbb{R}^l$ 满足

$$\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{w}\\ \mathbf{v}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix},\quad -\mathbf{q}^T\mathbf{w}+\mathbf{b}^T\mathbf{v}>0$$

假设存在 $\hat{\mathbf{x}}\in\mathbb{R}^n$ 满足 $\mathbf{A}\hat{\mathbf{x}}=\mathbf{b}$，则对于任意 $t>0$，$\mathbf{x}=\hat{\mathbf{x}}+t\mathbf{w}$ 均满足 $\mathbf{A}\hat{\mathbf{x}}=\mathbf{b}$，而

$$\begin{aligned}&\mathbf{x}^T\mathbf{P}\mathbf{x}+\mathbf{q}^T\mathbf{x}+r\\ &=\hat{\mathbf{x}}^T\mathbf{P}\hat{\mathbf{x}}+\mathbf{q}^T\hat{\mathbf{x}}+r+t\left(\mathbf{w}^T\mathbf{P}\hat{\mathbf{x}}+\mathbf{q}^T\mathbf{w}\right)+\frac{1}{2}t^2\mathbf{w}^T\mathbf{P}\mathbf{w}\\ &=\hat{\mathbf{x}}^T\mathbf{P}\hat{\mathbf{x}}+\mathbf{q}^T\hat{\mathbf{x}}+r+t\left(-\hat{\mathbf{x}}^T\mathbf{P}\mathbf{v}+\mathbf{q}^T\mathbf{w}\right)-\frac{1}{2}t^2\mathbf{v}^T\mathbf{A}\mathbf{w}\\ &=\hat{\mathbf{x}}^T\mathbf{P}\hat{\mathbf{x}}+\mathbf{q}^T\hat{\mathbf{x}}+r+t\left(-\mathbf{b}^T\mathbf{v}+\mathbf{q}^T\mathbf{w}\right)\end{aligned}$$

显然，当 $t\to+\infty$ 时，目标函数值趋于 $-\infty$。

此外还可以证明，如果 $\mathbf{P}\in S_{++}^{n\times n}$，$\begin{bmatrix}\mathbf{P} & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}$ 必然非奇异，该问题有唯一解。

第三种选择，在已知 $\mathbf{A}\mathbf{x}=\mathbf{b}$ 的一个可行解 $\mathbf{x}^{(1)}$ 的情况下，我们可以不断进行形如 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$ 的迭代搜索来求解优化问题（3.72）。我们依旧采用（3.69）所示的原问题目标函数在点 $\mathbf{x}^{(k)}$ 处的二阶 Talyor 展开作为近似目标函数，令步长 $t^{(k)}=1$，同时要求保证迭代到的新解 $\mathbf{x}^{(k+1)}$ 仍为满足 $\mathbf{A}\mathbf{x}=\mathbf{b}$ 的可行解。则搜索方向 $\mathbf{d}^{(k)}$ 应该满足

$$\mathbf{d}^{(k)}=\underset{\mathbf{A}\mathbf{y}=\mathbf{0}}{\arg\min}\nabla f\left(\mathbf{x}^{(k)}\right)^T\left(\mathbf{y}-\mathbf{x}^{(k)}\right)+\frac{1}{2}\left(\mathbf{y}-\mathbf{x}^{(k)}\right)^T\nabla^2 f\left(\mathbf{x}^{(k)}\right)\left(\mathbf{y}-\mathbf{x}^{(k)}\right)\tag{3.77}$$

此时得到形如例 3.11 所示的等式约束的严格凸二次规划问题，则 $\mathbf{d}^{(k)}$ 和某个对偶向量 $\mathbf{v}^{(k)}\in\mathbb{R}^m$ 应该一起满足

$$\begin{bmatrix} \nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix} = \begin{bmatrix} -\nabla f\left(\mathbf{x}^{(k)}\right) \\ \mathbf{0} \end{bmatrix} \tag{3.78}$$

解这个方程组即可得到仿射等式约束问题的可行初始点 Newton 方向（Newton Step at Feasible Points）。

算法 3.16. 仿射等式优化问题的可行初始点 Newton 算法

Step 1. 确定最初的探索点 $\mathbf{x}^{(1)} \in \mathbb{R}^n$，满足 $\mathbf{A}\mathbf{x}^{(1)} = \mathbf{b}$。令 $k := 1$，进入 Step 2。

Step 2. 计算 $\lambda^{(k)} = \left(\nabla f\left(\mathbf{x}^{(k)}\right)^T \left(\nabla^2 f\left(\mathbf{x}^{(k)}\right)\right)^{-1} \nabla f\left(\mathbf{x}^{(k)}\right)\right)^{\frac{1}{2}}$。如果 $\left(\lambda^{(k)}\right)^2 \leq 2\varepsilon$，$\varepsilon > 0$ 为预先设定的阈值。则已经收敛找到最优解，停止。

否则，进入 Step 3。

Step 3. 确定 Newton 方向 $\begin{bmatrix} \nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix} = \begin{bmatrix} -\nabla f\left(\mathbf{x}^{(k)}\right) \\ \mathbf{0} \end{bmatrix}$。

使用回溯搜索法确定合适前进步长 $t^{(k)}$。前进至新的探索点 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$。$k := k+1$，回到 Step 2，继续迭代。

定理 3.28. 算法 3.16 中每次迭代时都选择了可行下降方向。

证明：如果是严格凸二次规划问题，则对于第 k 次迭代，$\nabla^2 f\left(\mathbf{x}^{(k)}\right) \in S_{++}^{n\times n}$，不难进一步证明 $\begin{bmatrix} \nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \in S_{++}^{n\times n}$。则如果迭代没有收敛时，$\nabla f\left(\mathbf{x}^{(k)}\right) \neq \mathbf{0}$，我们有

$$\nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d}^{(k)} = -\nabla f\left(\mathbf{x}^{(k)}\right) \begin{bmatrix} \nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}^{-1} \nabla f\left(\mathbf{x}^{(k)}\right) < 0$$

因此，$\mathbf{d}^{(k)}$ 为下降方向。

同时，根据（3.78），我们有 $\mathbf{A}\mathbf{d}^{(k)} = \mathbf{0}$，所以

$$\mathbf{A}\mathbf{x}^{(k+1)} = \mathbf{A}\left(\mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}\right) = \mathbf{A}\mathbf{x}^{(k)} + t^{(k)}\mathbf{A}\mathbf{d}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$$

因此，$\mathbf{d}^{(k)}$ 为可行方向。综上所述，结论成立。□

第四种选择，在已知 $\mathbf{Ax}=\mathbf{b}$ 的一个不可行解 $\mathbf{x}^{(1)}$ 的情况下，我们可以不断进行形如 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}^{(k)}$ 的迭代搜索来求解优化问题（3.72）。我们依旧采用（3.69）所示的原问题目标函数的梯度在点 $\mathbf{x}^{(k)}$ 处的一阶 Talyor 展开作为近似目标函数，令步长 $t^{(k)}=1$，同时要求保证迭代到的新解 $\mathbf{x}^{(k+1)}$ 尽可能满足 $\mathbf{Ax}=\mathbf{b}$。则根据 KKT 条件，搜索方向 $\mathbf{d}^{(k)}$ 应该和某个对偶向量 $\mathbf{v}^{(k)}\in\mathbb{R}^m$ 应该一起满足

$$\mathbf{A}\left(\mathbf{x}^{(k)}+\mathbf{d}^{(k)}\right)=\mathbf{b},\quad \nabla f\left(\mathbf{x}^{(k)}\right)+\nabla^2 f\left(\mathbf{x}^{(k)}\right)\mathbf{d}^{(k)}+\mathbf{A}^T\mathbf{v}^{(k)}=\mathbf{0}$$

也即

$$\begin{bmatrix}\nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{d}^{(k)}\\ \mathbf{v}^{(k)}\end{bmatrix}=\begin{bmatrix}-\nabla f\left(\mathbf{x}^{(k)}\right)\\ \mathbf{Ax}^{(k)}-\mathbf{b}\end{bmatrix}\tag{3.79}$$

解这个方程组即可得到仿射等式约束问题的不可行初始点 Newton 方向（Newton Step at Infeasible Points）。如果迭代到某一步满足 $\mathbf{Ax}^{(k)}=\mathbf{b}$，则方程组（3.79）退化为方程组（3.78），之后迭代的解都能保持可行。

我们也可以从原对偶 Newton 法（Primal-Dual Newton Method）的角度来推导。我们定义原对偶残差

$$r\left(\mathbf{y}\right)=r\left(\mathbf{x},\mathbf{v}\right)=\begin{pmatrix}\nabla f\left(\mathbf{x}\right)+\mathbf{A}^T\mathbf{v}\right)\\ \mathbf{Ax}-\mathbf{b}\end{pmatrix}\triangleq\begin{pmatrix}r_{\text{dual}}\left(\mathbf{y}\right)\\ r_{\text{pri}}\left(\mathbf{y}\right)\end{pmatrix}\tag{3.80}$$

其中 $r_{\text{dual}}\left(\mathbf{y}\right)$ 称为对偶问题残差（Dual Residual），$r_{\text{pri}}\left(\mathbf{y}\right)$ 称为原问题残差（Primal Residual），则在点 $\mathbf{y}\in\mathbb{R}^{n+l}$ 处对于 $r\left(\mathbf{y}\right)$ 进行一阶 Talyor 展开作为近似目标函数，有

$$r\left(\mathbf{y}+\mathbf{d}_{\mathbf{y}}\right)\approx r\left(\mathbf{y}\right)+\text{Dr}\left(\mathbf{y}\right)\mathbf{d}_{\mathbf{y}}\tag{3.81}$$

则使得 $r\left(\mathbf{y}+\mathbf{d}_{\mathbf{y}}\right)\approx\mathbf{0}$ 的 $\mathbf{d}_{\mathbf{y}}$ 可以用 Newton 法求出为

$$\mathbf{d}_{\mathbf{y}}=-\text{Dr}\left(\mathbf{y}\right)^{-1}r\left(\mathbf{y}\right)=-\begin{pmatrix}\nabla^2 f\left(\mathbf{x}\right) & \mathbf{A}^T\\ \mathbf{A} & 0\end{pmatrix}^{-1}\begin{pmatrix}\nabla f\left(\mathbf{x}\right)+\mathbf{A}^T\mathbf{v}\\ \mathbf{Ax}-\mathbf{b}\end{pmatrix}\triangleq\begin{pmatrix}\mathbf{d}_{\mathbf{x}}\\ \mathbf{d}_{\mathbf{v}}\end{pmatrix}$$

其中 $\text{Dr}\left(\mathbf{y}\right)\in\mathbb{R}^{(n+l)\times(n+l)}$ 是 $r\left(\mathbf{y}\right)$ 在点 $\mathbf{y}\in\mathbb{R}^{n+l}$ 处的梯度。

不难看出，这一结果和方程组（3.79）求出的 Newton 方向是一致的。

接下来，我们讨论选定 Newton 方向作为前进方向之后，前进步长 t 如何选择。注意到

$$\begin{aligned}\frac{d}{dt}f\left(\mathbf{x}+t\mathbf{d}_{\mathbf{x}}\right)\Big|_{t=0}&=-\nabla f\left(\mathbf{x}\right)\mathbf{d}_{\mathbf{x}}\\ &=-\mathbf{d}_{\mathbf{x}}^T\left(\nabla^2 f\left(\mathbf{x}\right)\mathbf{d}_{\mathbf{x}}+\mathbf{A}^T\mathbf{v}\right)\\ &=-\mathbf{d}_{\mathbf{x}}^T\nabla^2 f\left(\mathbf{x}\right)\mathbf{d}_{\mathbf{x}}+\left(\mathbf{Ax}-\mathbf{b}\right)^T\mathbf{v}\end{aligned}$$

从不可行初始点出发的 Newton 方向，不一定是下降方向。当满足 $\mathbf{Ax}=\mathbf{b}$ 之

后，才能保证是下降方向。

进一步研究原对偶残差关于前进步长 t 的关系，有

$$\frac{d}{dt}\left\|r\left(\mathbf{y}+t\mathbf{d}_{\mathrm{y}}\right)\right\|_2^2\Bigg|_{t=0}=2r\left(\mathbf{y}\right)^T\mathrm{Dr}\left(\mathbf{y}\right)\mathbf{d}_{\mathrm{x}}=-r\left(\mathbf{y}\right)^T r\left(\mathbf{y}\right)$$

则

$$\frac{d}{dt}\left\|r\left(\mathbf{y}+t\mathbf{d}_{\mathrm{y}}\right)\right\|_2\Bigg|_{t=0}=-\left\|r\left(\mathbf{y}\right)\right\|_2$$

因此，我们可以通过检查 $\left\|r\left(\mathbf{y}\right)\right\|_2$ 来判断步长搜索是否已经收敛。

算法 3.17. 仿射等式优化问题的不可行初始点 Newton 算法

Step 1. 确定最初的探索点 $\mathbf{x}^{(1)}\in\mathbb{R}^n$，$\mathbf{v}^{(1)}\in\mathbb{R}^l$。选定 $\alpha\in\left(0,1/2\right)$，$\beta\in\left(0,1\right)$。令 $k:=1$，进入 Step 2。

Step 2. 根据 $\mathbf{x}^{(k)}$，$\mathbf{v}^{(k)}$ 计算出 $r\left(\mathbf{y}^{(k)}\right)=r\left(\mathbf{x}^{(k)},\mathbf{v}^{(k)}\right)$，检查是否满足 $\left\|\mathbf{A}\mathbf{x}^{(1)}-\mathbf{b}\right\|_2\le\varepsilon_1$，$\left\|r\left(\mathbf{y}^{(k)}\right)\right\|_2\le\varepsilon_2$，$\varepsilon_1,\varepsilon_2>0$ 为预先设定的阈值。如果满足这两个条件，则已经收敛找到可行的最优解，停止。

否则，进入 Step 3。

Step 3. 求解方程组 $\begin{bmatrix}\nabla^2 f\left(\mathbf{x}^{(k)}\right) & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{d}^{(k)}\\ \mathbf{v}^{(k)}\end{bmatrix}=\begin{bmatrix}-\nabla f\left(\mathbf{x}^{(k)}\right)\\ \mathbf{A}\mathbf{x}^{(k)}-\mathbf{b}\end{bmatrix}$ 确定 Newton 方向。令 $t_*=1$，进入 Step 4。

Step 4. 如果 $\left\|r\left(\mathbf{x}^{(k)}+t_*\mathbf{d}_{\mathrm{x}}^{(k)},\mathbf{v}^{(k)}+t_*\mathbf{d}_{\mathrm{v}}^{(k)}\right)\right\|_2<\left(1-\alpha t_*\right)\left\|r\left(\mathbf{x}^{(k)},\mathbf{v}^{(k)}\right)\right\|_2$，则已经用回溯搜索法确定合适前进步长，令 $t^{(k)}=t_*$，进入 Step 5。

否则，令 $t_*=\beta t_*$，回到 Step 4，继续迭代。

Step 5. 前进至新的探索点 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}_{\mathrm{x}}^{(k)}$，$\mathbf{v}^{(k+1)}=\mathbf{v}^{(k)}+t^{(k)}\mathbf{d}_{\mathrm{v}}^{(k)}$。$k:=k+1$，回到 Step 2，继续迭代。

具体计算时，α 一般取 0.01 到 0.1 之间，β 一般取 0.3 到 0.8 之间。

实践测试可知，仿射等式优化问题的不可行初始点 Newton 算法不一定保证

能收敛到可行解，可能迭代多次后依然无法进入可行域。但从（3.80）和（3.81）可以推出

$$r_{\text{pri}}\left(\mathbf{y}+t\mathbf{d}_{\text{y}}\right)=\mathbf{A}\left(\mathbf{x}+t\mathbf{d}_{\text{x}}\right)-\mathbf{b}=\left(1-t\right)\left(\mathbf{Ax}-\mathbf{b}\right)=\left(1-t\right)r_{\text{pri}}\left(\mathbf{y}\right)$$

因此，迭代多次后有

$$r_{\text{pri}}\left(\mathbf{y}+t\mathbf{d}_{\text{y}}\right)=r_{\text{pri}}\left(\mathbf{y}^{(k)}\right)=\left(\prod_{i=1}^{k-1}\left(1-t^{(i)}\right)\right)r_{\text{pri}}\left(\mathbf{y}^{(1)}\right)$$

不难看出，一旦有某一步 $t^{(i)}=1$，则可以得到 $\mathbf{Ax}^{(i+1)}=\mathbf{b}$。这说明，如果在某一步进入可行域之后，则可以一直保持在可行域内直到收敛。

例 3.11. 请用不同算法求解。

3.6.2 等式及不等式约束优化问题的搜索算法

对于同时含有非线性等式及不等式约束的优化问题（3.1），我们可以引入二次罚函数（Penalty Function），将其近似转化为无约束优化问题

$$\min_{\mathbf{x}} f_{QP}\left(\mathbf{x},\sigma\right)=f_0\left(\mathbf{x}\right)+\sum_{i=1}^{m}\sigma\left(\max\left\{f_i\left(\mathbf{x}\right),0\right\}\right)^2+\sum_{j=1}^{l}\sigma\left(h_j\left(\mathbf{x}\right)\right)^2 \tag{3.82}$$

其中 $\sigma>0$ 为罚因子。

很显然，如果 $\mathbf{x}$ 位于可行域外，惩罚项（Penalty Term）为正，使得目标函数值变大；如果 $\mathbf{x}$ 位于可行域内，惩罚项为 0，此时优化的就是原问题的目标函数。当 σ 较小时，求解无约束优化问题（3.82）得到的局部最优解一般位于可行域外。因此，我们可以逐步增大 σ，迫使无约束优化问题（3.82）得到的局部最优解向可行域靠近。这样我们得到一系列无约束优化问题（3.82）的局部最优解点列。因为，该点列位于可行域外，因此被称为外点罚函数（Exterior Penalty Function）。当 σ 趋于无穷时，惩罚项的引入会促使无约束优化问题（3.82）的最优解落在可行域内。

算法 3.17. 二次罚函数算法

Step 1. 确定最初的探索点 $\mathbf{x}^{(1)}\in\mathbb{R}^n$。设置初始罚因子 $\sigma^{(1)}$，罚因子增长系数 $\rho>0$。令 $k:=1$，进入 Step 2。

Step 2. 验证 $\mathbf{x}^{(k)}$ 是原问题（3.1）的可行解。则已经收敛找到最优解，停止。否则，进入 Step 3。

Step 3. 以 $\mathbf{x}^{(k)}$ 为初始解，$\sigma^{(k)}$ 为罚因子，求解优化问题（3.82）。令 $\sigma^{(k+1)}=\rho\sigma^{(k)}$。$k:=k+1$，回到 Step 2，继续迭代。

定理 3.29. 如果 $\mathbf{x}^{(k)}$ 是无约束优化问题 $\min f_{EP}\left(\mathbf{x},\sigma^{(k)}\right)$ 的全局最优解，假设我们令 $\sigma^{(k)}$ 单调上升趋于无穷，则 $\left\{\mathbf{x}^{(k)}\right\}$ 的极限点 $\bar{\mathbf{x}}$ 是原问题（3.1）的全局最优解。

证明：假设 $\mathbf{x}^*$ 是原问题（3.1）的全局最优解，满足所有约束条件，因此，$f_{QP}\left(\mathbf{x}^*,\sigma^{(k)}\right)=f_0\left(\mathbf{x}^*\right)$。而对于所有的可行解 $\mathbf{x}\in X$，还满足

$$f_0\left(\mathbf{x}^*\right)\le f_0\left(\mathbf{x}\right)$$

而 $\mathbf{x}^{(k)}$ 是无约束优化问题 $\min f_{EP}\left(\mathbf{x},\sigma^{(k)}\right)$ 的全局最优解，则

$$f_{QP}\left(\mathbf{x}^{(k)},\sigma^{(k)}\right)\le f_{QP}\left(\mathbf{x}^*,\sigma^{(k)}\right)=f_0\left(\mathbf{x}^*\right)$$

也即

$$f_0\left(\mathbf{x}^{(k)}\right)+\sum_{i=1}^{m}\sigma^{(k)}\left(\max\left\{f_i\left(\mathbf{x}^{(k)}\right),0\right\}\right)^2+\sum_{j=1}^{l}\sigma^{(k)}\left(h_j\left(\mathbf{x}^{(k)}\right)\right)^2\le f_0\left(\mathbf{x}^*\right)$$

变形可得

$$\sum_{i=1}^{m}\left(\max\left\{f_i\left(\mathbf{x}^{(k)}\right),0\right\}\right)^2+\sum_{j=1}^{l}\left(h_j\left(\mathbf{x}^{(k)}\right)\right)^2\le\frac{1}{\sigma^{(k)}}\left[f_0\left(\mathbf{x}^*\right)-f_0\left(\mathbf{x}^{(k)}\right)\right]$$

当 $\mathbf{x}^{(k)}\to\bar{\mathbf{x}}$，$\sigma^{(k)}\to+\infty$，我们有

$$\sum_{i=1}^{m}\left(\max\left\{f_i\left(\mathbf{x}^{(k)}\right),0\right\}\right)^2+\sum_{j=1}^{l}\left(h_j\left(\mathbf{x}^{(k)}\right)\right)^2\to 0$$

因此，$\bar{\mathbf{x}}$ 也是可行解，且

$$f_0\left(\bar{\mathbf{x}}\right)\le f_0\left(\mathbf{x}^*\right)$$

显然，必然有 $f_0\left(\bar{\mathbf{x}}\right)=f_0\left(\mathbf{x}^*\right)$，结论成立。□

实际计算时，由于罚因子 σ 太大后会使得我们对于无约束优化问题（3.82）变得日益困难。因此，研究者针对凸优化问题（3.31）又提出了内点罚函数（Interior Penalty Function）方法，也称对数障碍函数方法（Barrier Function Method），将其近似转化为等式约束凸优化问题

$$\begin{aligned}\min_{\mathbf{x}} \quad & f_{LP}(\mathbf{x},\sigma)=f_0(\mathbf{x})-\sum_{i=1}^{m}\frac{1}{\sigma}\ln\left(-f_i(\mathbf{x})\right)\\ \text{s.t.} \quad & \mathbf{A}\mathbf{x}=\mathbf{b}\end{aligned} \tag{3.83}$$

函数 $f_i(\cdot):\mathbb{R}^n\to\mathbb{R}$ 刻画不等式约束条件，$i\in I=\{1,...,m\}$，$\mathbf{A}\in\mathbb{R}^{l\times n}$，$\mathbf{b}\in\mathbb{R}^l$。

显然，等式约束凸优化问题（3.83）可以使用前述等式约束优化问题的求解方法来求解。这样设计罚函数，可以保证求解出的解 $\mathbf{x}^{(k)}$ 一直在无约束优化问题（3.83）的可行域内。如果原问题的最优解位于可行域边界上，则我们需要调整罚因子 σ 使其趋于无穷时，减弱对数罚函数在边界附近的惩罚效果。但类似于外点罚函数方法，罚因子 σ 太大后会使得我们对于等式约束凸优化问题（3.83）变得日益困难。

二次罚函数不能保证外点罚函数是二阶可导的，因此不能直接利用二阶算法（如牛顿法）求解子问题。而对数函数保证内点罚函数二阶可导，因此可以使用利用二阶算法求解子问题。

因此，我们进一步考虑等式束凸优化问题（3.83）的 KKT 条件。显然，此时 Slater 条件成立，对于给定的 $\sigma^{(k)}$，等式约束凸优化问题（3.83）的最优解为 $\overline{\mathbf{x}}^{(k)}$，必然存在对偶问题最优解 $\overline{\mathbf{w}}^{(k)}\in\mathbb{R}^l$ 满足

$$\nabla f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\frac{-1}{\sigma^{(k)}f_i\left(\overline{\mathbf{x}}^{(k)}\right)}\nabla f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\mathbf{A}^T\overline{\mathbf{w}}^{(k)}=\mathbf{0},\quad \mathbf{A}\overline{\mathbf{x}}^{(k)}=\mathbf{b} \tag{3.84}$$

我们可以直接令

$$\lambda_i^{(k)}=\frac{-1}{\sigma^{(k)}f_i\left(\overline{\mathbf{x}}^{(k)}\right)},\quad i\in I=\{1,...,m\},\quad \boldsymbol{\mu}^{(k)}=\overline{\mathbf{w}}^{(k)} \tag{3.85}$$

注意到，如果 $f_i\left(\mathbf{x}^{(k)}\right)<0$，则 $\lambda_i^{(k)}>0$。且

$$\nabla f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\mathbf{A}^T\boldsymbol{\mu}^{(k)}=\mathbf{0}$$

所以，$\overline{\mathbf{x}}^{(k)}$ 最小化了，选定 $(\boldsymbol{\lambda},\boldsymbol{\mu})=\left(\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)$ 时的原优化问题（3.31）的 Lagrangian 函数

$$L\left(\mathbf{x},\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)=f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i^{(k)}f_i(\mathbf{x})+\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)^T\boldsymbol{\mu}^{(k)}$$

因此，点 $\left(\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)$ 位于原优化问题（3.31）的 Lagrangian 对偶函数 $g(\boldsymbol{\lambda},\boldsymbol{\mu})$ 的定义域内，$g\left(\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)>-\infty$。因此，$\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}$ 应该是原优化问题（3.31）的对偶可行解。

整理之后，我们不难发现，$\overline{\mathbf{x}}^{(k)}$，$\boldsymbol{\lambda}^{(k)}$ 和 $\boldsymbol{\mu}^{(k)}$ 应该一起满足如下的扰动 KKT 条件（Perturbed KKT Conditions）

$$\nabla f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\mathbf{A}^T\boldsymbol{\mu}^{(k)}=\mathbf{0}$$

$$f_i\left(\overline{\mathbf{x}}^{(k)}\right)\le 0,\ \ i\in I=\{1,...,m\},\ \ \mathbf{A}\overline{\mathbf{x}}^{(k)}=\mathbf{b}$$

$$\lambda_i^{(k)}\ge 0,\ \ i\in I=\{1,...,m\}$$

$$\lambda_i^{(k)}f_i\left(\overline{\mathbf{x}}^{(k)}\right)=-\frac{1}{\sigma^{(k)}},\ \ i\in I=\{1,...,m\} \tag{3.86}$$

和定理 3.9 相比，唯一的差别在于互补松弛条件被替换成了（3.86）。

不仅如此，我们还可以进一步推出

$$g\left(\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)=\min_{x\in D}f_0(\mathbf{x})+\sum_{i=1}^{m}\lambda_i^{(k)}f_i(\mathbf{x})+\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)^T\boldsymbol{\mu}^{(k)}$$

$$=f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\lambda_i^{(k)}f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\left(\mathbf{A}\overline{\mathbf{x}}^{(k)}-\mathbf{b}\right)^T\boldsymbol{\mu}^{(k)}$$

$$=f_0\left(\overline{\mathbf{x}}^{(k)}\right)-\frac{m}{\sigma^{(k)}}$$

根据强对偶性，原优化问题（3.31）的最优解 $\mathbf{x}^*$ 应该满足

$$g\left(\boldsymbol{\lambda}^{(k)},\boldsymbol{\mu}^{(k)}\right)\le f_0\left(\mathbf{x}^*\right)\le f_0\left(\overline{\mathbf{x}}^{(k)}\right)$$

当 $\sigma^{(k)}\to+\infty$ 时，则有

$$\lim_{k\to\infty}f_0\left(\overline{\mathbf{x}}^{(k)}\right)=p^*,\ \ \lim_{k\to\infty}\overline{\mathbf{x}}^{(k)}=\mathbf{x}^*$$

定义 3.27. [中心路径（Central Path）] 对于等式约束凸优化问题（3.83），给定任意一个罚因子 $\sigma>0$，我们可以确定一个最优解 $\mathbf{x}_\sigma$，所有这样的 $\mathbf{x}_\sigma$ 构成一个在 $\mathbb{R}^n$ 空间的连续曲线集合，我们称为中心路径。

显然，满足如下的扰动 KKT 条件（Perturbed KKT Conditions）的 $\left\{\overline{\mathbf{x}}^{(k)}\right\}$ 位于收敛到最优解的中心路径（Central Path）上。

假设优化问题（3.31）的非线性目标函数和不等式约束函数均二次可导，由此得到如下算法。

算法 3.18. 中心路径障碍函数法

Step 1. 确定最初可行解 $\mathbf{x}^{(1)}\in\mathbb{R}^n$，满足 $f_i\left(\mathbf{x}^{(1)}\right)<0$，$i\in I=\{1,...,m\}$，$\mathbf{A}\mathbf{x}^{(1)}=\mathbf{b}$。设置初始罚因子 $\sigma^{(1)}$，罚因子增长系数 $\rho>0$，停止条件阈值 $\varepsilon>0$。令 $k:=1$，进入 Step 2。

Step 2. [Centering Step] 以 $\mathbf{x}^{(k)}$ 为初始点用等式约束 Newton 法解以下问题

$$\min\left\{\sigma^{(k)} f_0(\mathbf{x}) - \sum_{i=1}^{m} \ln\left(-f_i(\mathbf{x})\right) \middle| \text{s.t. } \mathbf{Ax} = \mathbf{b}\right\} \tag{3.87}$$

得到该问题的最优解 $\mathbf{x}^{(k+1)}$。这里为了避免除很大的 $\sigma^{(k)}$ 造成舍入误差，而改成了等价的乘 $\sigma^{(k)}$ 的目标函数形式。因为这一步可以被视为将 $\mathbf{x}^{(k+1)}$ 放到中心路径上，所以又被称为 Centering Step。进入 Step 3。

Step 3. 如果 $\frac{m}{\sigma^{(k)}} \le \varepsilon$，已经收敛，停止。

否则，令 $\sigma^{(k+1)} = \rho\sigma^{(k)}$。$k := k+1$，回到 Step 2，继续迭代。

类似于线性规划的两阶段方法，我们可以求解如下两个问题中的任意一个来找到满足 $f_i\left(\mathbf{x}^{(1)}\right) < 0$，$i \in I = \{1,...,m\}$，$\mathbf{Ax}^{(1)} = \mathbf{b}$ 的最初可行解 $\mathbf{x}^{(1)}$。

极小化最大不可行值问题：

$$\min\left\{s \middle| \text{ s.t. } f_i(\mathbf{x}) \le s, i = 1,\cdots,m, \mathbf{Ax} = \mathbf{b}\right\} \tag{3.88}$$

极小化不可行值之和问题：

$$\min\left\{\sum_{i=1}^{m} s_i \middle| \text{ s.t. } f_i(\mathbf{x}) \le s_i, i = 1,\cdots,m, \mathbf{Ax} = \mathbf{b}, s_i \ge 0\right\} \tag{3.89}$$

我们依然可以用中心路径障碍函数法来求解优化问题（3.88）或者（3.89）。我们直接根据这些问题最优目标值是否小于 0，来判断是否可行并停止，得到最初可行解 $\mathbf{x}^{(1)}$。求解极小化不可行值之和问题的一个优点是：当原始系统不可行时，最终得到解中的非零项将告诉我们哪些约束是无法满足的。

我们知道在 $\mathbf{x}^{(k)}$ 处的目标函数值与最优解的差肯定小于 $\frac{m}{\sigma^{(k)}}$，但我们不是直接给一个很大的 $\sigma^{(1)}$ 值通过较少的几次迭代就能算出最优解。这是因为中心路径障碍函数法的计算量是由内外两层迭代组成的，外层迭代增大 $\sigma^{(k)}$，内层用等式约束 Newton 法迭代求解优化问题（3.87）。如果 ρ 太小，那么可能需要许多外部迭代；如果 ρ 太大，那么用等式约束 Newton 方法求解 Step 2 可能需要多次迭代。如果 $\sigma^{(1)}$ 太小，那么可能需要许多外部迭代；如果 $\sigma^{(1)}$ 太大，那么第一次用等式约束 Newton 方法求解 Step 2 可能需要多次迭代。因此，需要对 ρ 与 $\sigma^{(1)}$ 进行权衡，在内层迭代次数和外层迭代次数之间折衷。好在实践测试发现，中心路径障碍函数法的性能通常对 ρ 和 $\sigma^{(1)}$ 的选择相当稳健。

一般来说，ρ 的取值在 10 到 20 之间比较合适。而注意到停止条件要求

$\nabla f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\mathbf{A}^T\boldsymbol{\mu}^{(k)}\to\mathbf{0}$，找到最初可行解 $\mathbf{x}^{(1)}$ 之后，我们以用最小二乘法求解如下问题来得到合适的 $\sigma^{(1)}$。

$$\min_{\sigma^{(1)},\boldsymbol{\mu}^{(1)}}\left\|\sigma^{(1)}\nabla f_0\left(\overline{\mathbf{x}}^{(1)}\right)-\sum_{i=1}^{m}\frac{\nabla f_i\left(\overline{\mathbf{x}}^{(1)}\right)}{f_i\left(\overline{\mathbf{x}}^{(1)}\right)}+\mathbf{A}^T\boldsymbol{\mu}^{(1)}\right\|_2^2$$

例 3.12. 请用中心路径障碍函数法求解如下线性规划问题

$$\begin{aligned}\min_{\mathbf{x}}\quad & z=\mathbf{c}^T\mathbf{x}\\ \text{s.t.}\quad & \mathbf{D}\mathbf{x}\le\mathbf{e}\end{aligned}\tag{3.90}$$

其中 $\mathbf{c}\in\mathbb{R}^n$，$\mathbf{D}\in\mathbb{R}^{l\times n}$，$\operatorname{rank}(\mathbf{D})=l<n$，$\mathbf{e}\in\mathbb{R}^l$。

解：显然，此时需要用等式约束 Newton 法求解的问题是

$$\min\left\{\sigma^{(k)}\mathbf{c}^T\mathbf{x}-\sum_{i=1}^{m}\ln\left(e_i-\mathbf{d}_i^T\mathbf{x}\right)\right\}$$

直接令该目标函数的梯度为 $\mathbf{0}$，则最优解 $\mathbf{x}^*$ 应该满足

$$\sigma^{(k)}\mathbf{c}-\sum_{i=1}^{m}\frac{\mathbf{d}_i}{e_i-\mathbf{d}_i^T\mathbf{x}^*}=\mathbf{0}\tag{3.91}$$

如果我们定义 $\phi(\mathbf{x})=-\sum_{i=1}^{m}\ln\left(e_i-\mathbf{d}_i^T\mathbf{x}\right)$，则 $\nabla\phi\left(\mathbf{x}^*\right)=-\sum_{i=1}^{m}\frac{\mathbf{d}_i}{e_i-\mathbf{d}_i^T\mathbf{x}^*}$，则上式（3.91）说明 $\nabla\phi\left(\mathbf{x}^*\right)$ 和向量 $\mathbf{c}$ 是平行的。图 3.10 中的多面体显示了线性规划问题（3.90）的可行域，其中每一条虚线就对应了给定不同 $\sigma^{(k)}$ 时，对数障碍函数的等值线，$\sigma^{(k)}$ 越大等值线越靠近边界。从内部出发到多边形顶点的那一条实曲线就是中心路径。可以看出，如果在中心路径与等值线的交点处作等值线的法线，就是 $\phi\left(\mathbf{x}^*\right)$ 的梯度方向 $\nabla\phi\left(\mathbf{x}^*\right)$，那么这一条法线一定是与向量 $\mathbf{c}$ 平行。随着 $\sigma^{(k)}$ 增大，可以看到 $\mathbf{x}^{(k)}$ 的确是从内部出发逐步走到边界上的最优解。

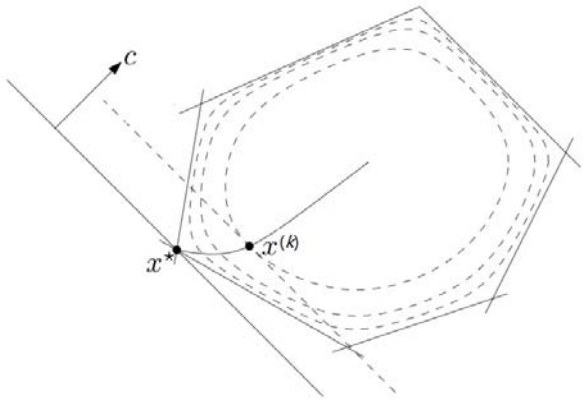


图 3.10　线性规划问题（3.87）的中心路径示意图。

例 3.13. 请用对数障碍函数法求解如下线性规划问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{1}^T\mathbf{x} \le 1 \\ & \mathbf{x} \ge \mathbf{0} \end{aligned}$$

其中 $\mathbf{c} \in \mathbb{R}^n$。

解：使用对视障碍函数法，给定 $\sigma^{(k)}$，我们可以得到如下的无约束优化问题

$$\min\left\{ f_{IP}(\mathbf{x},\sigma) = \mathbf{c}^T\mathbf{x} - \frac{1}{\sigma}\ln\left(1-\mathbf{1}^T\mathbf{x}\right) - \frac{1}{\sigma}\sum_{i=1}^{m}\ln\left(-x_i\right) \right\}$$

显然有

$$\nabla f_{IP}(\mathbf{x},\sigma) = \frac{1}{\sigma}\left[\sigma\mathbf{c}^T - \frac{\mathbf{1}}{1-\mathbf{1}^T\mathbf{x}} - \left[\frac{1}{x_1},\ldots,\frac{1}{x_n} \right]^T \right]$$

$$\nabla^2 f_{IP}(\mathbf{x},\sigma) = \frac{1}{\sigma}\left[\frac{\mathbf{1}^T\mathbf{1}}{\sigma\left(1-\mathbf{1}^T\mathbf{x}\right)^2} - \begin{bmatrix} \frac{1}{x_1^2} & 0 & \ldots & 0 \\ 0 & \frac{1}{x_2^2} & \ldots & \ldots \\ \ldots & \ldots & \ldots & 0 \\ 0 & \ldots & 0 & \frac{1}{x_n^2} \end{bmatrix} \right]$$

在我们使用等式约束 Newton 方法求解 Step 2 时，必须要关心 $\nabla^2 f_{LP}(\mathbf{x},\sigma)$ 是否病态（Ill-Conditioned）。

第一种情况，$\mathbf{x}$ 接近可行集的边界，σ 是适度的。在这种情况下，向量 $1-\mathbf{1}^T\mathbf{x}$ 或向量 $\mathbf{x}$ 中的一些元素值接近于零，而其它元素值则不接近于零。因此，矩阵 $\nabla^2 f_{LP}(\mathbf{x},\sigma)$ 变得非常病态，同时 $\nabla f_{LP}(\mathbf{x},\sigma)$ 的一些元素值可能非常小，而其它元素值则非常大。这都使得精确求解 Newton 方程 $\nabla^2 f_{LP}(\mathbf{x},\sigma)\mathbf{d}=-\nabla f_{LP}(\mathbf{x},\sigma)$ 希望渺茫。

第二种情况，$\mathbf{x}$ 不接近可行集的边界，σ 很大。同样会造成 Newton 方程难以求解。

第三种情况，$\mathbf{x}$ 不接近可行集的边界，σ 是适度的。在这种情况下，矩阵 $\nabla^2 f_{LP}(\mathbf{x},\sigma)$ 不会病态，同时，$\nabla f_{LP}(\mathbf{x},\sigma)$ 中各元素值大小接近。因此，精确求解 Newton 方程 $\nabla^2 f_{LP}(\mathbf{x},\sigma)\mathbf{d}=-\nabla f_{LP}(\mathbf{x},\sigma)$。

第四种情况，$\mathbf{x}$ 接近可行集的边界，σ 很大。虽然此时向量 $1-\mathbf{1}^T\mathbf{x}$ 或向量 $\mathbf{x}$ 中的一些元素值接近于零，而其它元素值则不接近于零。但 $\mathbf{x}$ 接近可行集的边界是，根据例 3.12 的结论，如果我们希望保持在中心路径上，则迭代的前进方向应该保持和向量 $\mathbf{c}$ 平行，因此，我们可以大致忽略 $\nabla f_{LP}(\mathbf{x},\sigma)$ 中关于向量 $1-\mathbf{1}^T\mathbf{x}$ 的后面那项。同时，虽然 $\nabla^2 f_{LP}(\mathbf{x},\sigma)$ 可能是病态的，难以准确计算。然而，仔细观察不难发现，$\nabla^2 f_{LP}(\mathbf{x},\sigma)$ 是 $\mathbf{c}^T$ 和 $\nabla f_{LP}(\mathbf{x},\sigma)$ 的主特征向量的加权和。因此，Newton 方程可以被相当精确的求解。

保持沿着中心路径搜素可以确保我们总是落在第三种情况或者第四种情况中。由此可以看出，我们在迭代过程中保持在中心路径上的必要性。

类似于前述分析，我们也可以从原对偶 Newton 法（Primal-Dual Newton Method）的角度来重新审视上述算法。可以将上述扰动 KKT 条件中的核心部分视为一个非线性方程组

$$\begin{cases}\nabla f_0\left(\overline{\mathbf{x}}^{(k)}\right)+\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i\left(\overline{\mathbf{x}}^{(k)}\right)+\mathbf{A}^T\boldsymbol{\mu}^{(k)}=\mathbf{0}\\ \lambda_i^{(k)}f_i\left(\overline{\mathbf{x}}^{(k)}\right)=-\dfrac{1}{\sigma^{(k)}},i=1,\ldots,m\\ \mathbf{A}\overline{\mathbf{x}}^{(k)}=\mathbf{b}\end{cases}$$

我们将其简记为

$$r(\mathbf{y})=r(\mathbf{x},\boldsymbol{\lambda},\mathbf{v})=\begin{pmatrix}\nabla f_0(\mathbf{x})+\mathrm{D}q(\mathbf{x})^T\boldsymbol{\lambda}+\mathbf{A}^T\mathbf{v}\\ -diag(\boldsymbol{\lambda})q(\mathbf{x})-\dfrac{1}{\sigma}\mathbf{1}\\ \mathbf{A}\mathbf{x}-\mathbf{b}\end{pmatrix}\triangleq\begin{pmatrix}r_{dual}(\mathbf{y})\\ r_{cent}(\mathbf{y})\\ r_{pri}(\mathbf{y})\end{pmatrix}\quad(3.92)$$

并定义对偶问题残差 $r_{\text{dual}}(\mathbf{y})$，原问题残差 $r_{\text{pri}}(\mathbf{y})$ 和中心路径残差 $r_{cent}(\mathbf{y})$。其中

$$q(\mathbf{x})=\begin{pmatrix} f_1(\mathbf{x}) \\ \cdots \\ f_m(\mathbf{x}) \end{pmatrix},\ \mathrm{D}q(\mathbf{x})=\begin{bmatrix} \nabla f_1(\mathbf{x})^T \\ \cdots \\ \nabla f_m(\mathbf{x})^T \end{bmatrix},\ diag(\boldsymbol{\lambda})=\begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & \ldots \\ \ldots & \ldots & \ldots & 0 \\ 0 & \ldots & 0 & \lambda_m \end{bmatrix}。$$

类似于前述（3.81）式的方式，我们可以在点 $\mathbf{y}\in\mathbb{R}^{n+m+l}$ 处对于 $r(\mathbf{y})$ 进行一阶 Talyor 展开作为近似目标函数 $r(\mathbf{y}+\mathbf{d}_{\mathrm{y}})\approx r(\mathbf{y})+\mathrm{Dr}(\mathbf{y})\mathbf{d}_{\mathrm{y}}$，然后用 Newton 法求出使得 $r(\mathbf{y}+\mathbf{d}_{\mathrm{y}})\approx\mathbf{0}$ 的 $\mathbf{d}_{\mathrm{y}}$ 作为前进方向。其中 $\mathrm{Dr}(\mathbf{y})\in\mathbb{R}^{(n+m+l)\times(n+m+l)}$ 是 $r(\mathbf{y})$ 在点 $\mathbf{y}\in\mathbb{R}^{n+m+l}$ 处的梯度。

如果我们进一步采用（3.85）选择 $\lambda_i^{(k)}=\dfrac{-1}{\sigma^{(k)}f_i(\overline{\mathbf{x}}^{(k)})}$，$i\in I=\{1,...,m\}$。则可以进一步简化得到

$$r(\mathbf{x},\mathbf{v})=\begin{pmatrix} \nabla f_0(\mathbf{x})+\sum_{i=1}^{m}\left(-\dfrac{1}{\sigma f_i(\mathbf{x})}\right)\nabla f_i(\mathbf{x})+\mathbf{A}^T\mathbf{v} \\ \mathbf{A}\mathbf{x}-\mathbf{b} \end{pmatrix}$$

则此时的 Newton 方向满足

$$\mathbf{d}_{\mathrm{y}}=-\mathrm{Dr}(\mathbf{y})^{-1}r(\mathbf{y})=-\begin{pmatrix} H_{LQ}(\mathbf{x}) & \mathbf{A}^T \\ \mathbf{A} & 0 \end{pmatrix}^{-1}r(\mathbf{y})\triangleq\begin{pmatrix} \mathbf{d}_{\mathbf{x}} \\ \mathbf{d}_{\mathbf{v}} \end{pmatrix} \qquad (3.93)$$

其中 $H_{LQ}(\mathbf{x})=\nabla^2 f_0(x)+\sum_{i=1}^{m}\dfrac{1}{\sigma f_i(x)^2}\nabla f_i(x)\nabla f_i(x)^T+\sum_{i=1}^{m}\dfrac{1}{\sigma f_i(x)^2}\nabla^2 f_i(x)$。

（3.93）就是用 Newton 法求解扰动 KKT 条件进行一次迭代之后前进方向。这一做法和算法 3.18 中心路径障碍函数法的本质相同。

但我们也可以不采用（3.85）消去 $\boldsymbol{\lambda}$，而是直接从（3.92）得出如下的 Newton 前进方向

$$\begin{aligned}\mathbf{d}_{\mathrm{y}}&=-\mathrm{Dr}(\mathbf{y})^{-1}r(\mathbf{y})\\&=-\begin{bmatrix} H_{pd}(\mathbf{x}) & \mathrm{D}q(\mathbf{x})^T & \mathbf{A}^T \\ -diag(\boldsymbol{\lambda})\mathrm{D}q(\mathbf{x}) & -diag(q(\mathbf{x})) & 0 \\ \mathbf{A} & 0 & 0 \end{bmatrix}^{-1}\begin{pmatrix} r_{dual}(\mathbf{y}) \\ r_{cent}(\mathbf{y}) \\ r_{pri}(\mathbf{y}) \end{pmatrix}\triangleq\begin{pmatrix} \mathbf{d}_{\mathbf{x}} \\ \mathbf{d}_{\boldsymbol{\lambda}} \\ \mathbf{d}_{\mathbf{v}} \end{pmatrix}\end{aligned} \qquad (3.94)$$

其中 $diag\left(q\left(\mathbf{x}\right)\right)=\begin{bmatrix} f_1\left(\mathbf{x}\right) & ... & 0 \\ ... & ... & ... \\ 0 & ... & f_m\left(\mathbf{x}\right) \end{bmatrix}$，$H_{pd}\left(\mathbf{x}\right)=\nabla^2 f_0\left(\mathbf{x}\right)+\sum_{i=1}^{m}\lambda_i\nabla^2 f_i\left(\mathbf{x}\right)$。

类似于不可行初始点 Newton 算法，（3.94）所定义的 Newton 前进方向需要同时关注原对偶变量及中心路径残差的更新，因此被称为原对偶内点法（Primal-Dual Interior-Point Method）。

注意到，迭代时 $\mathbf{v}$ 不一定是对偶问题的解。因此对于原对偶内点法，我们需要构建一个代理对偶间隙来考察问题是否收敛：

$$\eta\left(\mathbf{x},\boldsymbol{\lambda}\right)=-q\left(\mathbf{x}\right)^T\boldsymbol{\lambda}=-\sum_{i=1}^{m}\lambda_i f_i\left(\mathbf{x}\right) \tag{3.95}$$

如果我们能找到可行的点即 $r_{pri}\left(\mathbf{y}\right)=\mathbf{0}$ 以及 $r_{pri}\left(\mathbf{y}\right)=\mathbf{0}$，这个代理对偶间隙应当是一个真实的对偶间隙。

由扰动 KKT 条件中（3.86）式可以进一步推出

$$\eta\left(\mathbf{x},\boldsymbol{\lambda}\right)=-\sum_{i=1}^{m}\lambda_i f_i\left(\mathbf{x}\right)=\frac{m}{\sigma}$$

因此，我们可以直接计算 $\eta\left(\mathbf{x},\boldsymbol{\lambda}\right)\le\varepsilon$ 来判断算法是否收敛。

算法 3.19. 原对偶内点法

Step 1. 确定最初探索点 $\mathbf{x}^{(1)}\in\mathbb{R}^n$，满足 $f_i\left(\mathbf{x}^{(1)}\right)<0$，$i\in I=\left\{1,...,m\right\}$，$\boldsymbol{\lambda}^{(1)}\in\mathbb{R}_+^l$，$\mathbf{v}^{(1)}\in\mathbb{R}^l$。$\eta^{(1)}=-q\left(\mathbf{x}^{(1)}\right)^T\boldsymbol{\lambda}^{(1)}$。选定 $\alpha\in\left(0,1/2\right)$，$\beta\in\left(0,1\right)$。设定罚因子增长系数 $\rho>0$，停止条件阈值 $\varepsilon_1,\varepsilon_2>0$。令 $k:=1$，进入 Step 2。

Step 2. 令 $\sigma^{(k)}=\dfrac{\rho m}{\eta^{(k)}}$。求解下述方程组计算 Newton 前进方向

$$\begin{bmatrix} H_{pd}\left(\mathbf{x}^{(k)}\right) & \mathrm{D}q\left(\mathbf{x}^{(k)}\right)^T & \mathbf{A}^T \\ -diag\left(\boldsymbol{\lambda}\right)Dq\left(\mathbf{x}^{(k)}\right) & -diag\left(q\left(\mathbf{x}^{(k)}\right)\right) & 0 \\ \mathbf{A} & 0 & 0 \end{bmatrix}\begin{pmatrix} \mathbf{d}_{\mathbf{x}}^{(k)} \\ \mathbf{d}_{\boldsymbol{\lambda}}^{(k)} \\ \mathbf{d}_{\mathbf{v}}^{(k)} \end{pmatrix}=-\begin{pmatrix} r_{dual}\left(\mathbf{y}^{(k)}\right) \\ r_{cent}\left(\mathbf{y}^{(k)}\right) \\ r_{pri}\left(\mathbf{y}^{(k)}\right) \end{pmatrix} \tag{3.96}$$

计算能保持对偶可行性的最大步长

$$t_{\max}=\min\left\{1,\min\left\{-\frac{\lambda_i}{d_{\boldsymbol{\lambda},i}^{(k)}}:d_{\boldsymbol{\lambda},i}^{(k)}<0\right\}\right\} \tag{3.97}$$

令 $t_*=t_{\max}$，进入 Step 3。

Step 3. 如果当前步长能保持原问题可行性

$$f_i\left(\mathbf{x}^{(k)}+t_*\mathbf{d}_{\mathbf{x}}^{(k)}\right)<0,\ \ i\in I=\{1,...,m\} \tag{3.98}$$

则进入 Step 4。

否则，令 $t_*=\beta t_*$，回到 Step 3，继续迭代。

Step 4. 如果当前步长能使下一个探索点的目标函数值更小

$$\left\|r\left(\mathbf{x}^{(k)}+t_*\mathbf{d}_{\mathbf{x}}^{(k)},\boldsymbol{\lambda}^{(k)}+t_*\mathbf{d}_{\boldsymbol{\lambda}}^{(k)},\mathbf{v}^{(k)}+t_*\mathbf{d}_{\mathbf{v}}^{(k)}\right)\right\|_2<(1-\alpha t_*)\left\|r\left(\mathbf{x}^{(k)},\mathbf{v}^{(k)}\right)\right\|_2 \tag{3.99}$$

则已经用回溯搜索法确定合适前进步长，令 $t^{(k)}=t_*$，进入 Step 5。

否则，令 $t_*=\beta t_*$，回到 Step 4，继续迭代。

Step 5. 前进至新的探索点 $\mathbf{x}^{(k+1)}=\mathbf{x}^{(k)}+t^{(k)}\mathbf{d}_{\mathbf{x}}^{(k)}$，$\boldsymbol{\lambda}^{(k+1)}=\boldsymbol{\lambda}^{(k)}+t^{(k)}\mathbf{d}_{\boldsymbol{\lambda}}^{(k)}$，$\mathbf{v}^{(k+1)}=\mathbf{v}^{(k)}+t^{(k)}\mathbf{d}_{\mathbf{v}}^{(k)}$。计算 $\eta^{(k+1)}=-q\left(\mathbf{x}^{(k+1)}\right)^T\boldsymbol{\lambda}^{(k+1)}$，如果 $\eta^{(k+1)}\le\varepsilon_1$ 且 $\left(\left\|r_{pri}\left(\mathbf{y}^{(k+1)}\right)\right\|_2^2+\left\|r_{dual}\left(\mathbf{y}^{(k+1)}\right)\right\|_2^2\right)^{\frac{1}{2}}\le\varepsilon_2$，则算法已经收敛，停止。

否则，令 $k:=k+1$，回到 Step 2，继续迭代。

例 3.12. 请用不同算法求解。

3.7　作业题

3.1　试证明：给定闭的非空凸集合 $\Omega \subseteq \mathbb{R}^n$ 及其内一点 $\mathbf{x}$，$F_\Omega(\mathbf{x}) \subseteq T_\Omega(\mathbf{x})$。

3.2　试证明：给定闭的非空凸集合 $\Omega \subseteq \mathbb{R}^n$ 及其内一点 $\mathbf{x}$，线性化可行方向锥 $L_\Omega(\mathbf{x})$ 一般比切锥 $T_\Omega(\mathbf{x})$ 要大。

3.3　试证明：[Mangasarian-Fromovitz 约束品性] 如果非线性规划问题（3.1）的可行点 $\mathbf{x}$ 处的积极集 $A(\mathbf{x})$，如果存在向量 $\mathbf{d} \in \mathbb{R}^n$ 使得

$$\mathbf{d}^T \nabla f_i(\mathbf{x}) < 0, \forall i \in A(\mathbf{x}) \bigcap I$$

$$\mathbf{d}^T \nabla h_j(\mathbf{x}) = 0, \forall j \in J$$

并且等式约束对应的梯度集 $\{\nabla h_j(\mathbf{x}), \forall j \in J\}$ 是线性无关的，则称 Mangasarian-Fromovitz 约束品性在点 $\mathbf{x}$ 处成立。

3.4　试证明：定理 3.10 所描述的 Fritz John 条件。

3.5　已知 $a,b>0$，$a+b=2$，试求 $\frac{1}{a}+\frac{4}{b}$ 的最大值和最小值。

3.6　给定探索点 $\mathbf{x}^{(k)}$ 处的负梯度方向 $-\nabla f\left(\mathbf{x}^{(k)}\right)$，试证明下面两个问题的最优解一样。

优化问题 I：

$$\begin{aligned} \min_{\mathbf{d}\in\mathbb{R}^n} \quad & \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d} \\ \text{s.t.} \quad & \|\mathbf{d}\| \le 1 \end{aligned}$$

优化问题 II：

$$\begin{aligned} \min_{\mathbf{d}\in\mathbb{R}^n} \quad & \nabla f\left(\mathbf{x}^{(k)}\right)^T \mathbf{d} \\ \text{s.t.} \quad & \|\mathbf{d}\| = 1 \end{aligned}$$

3.6　试证明最速下降算法 3.9 满足定理 3.17。

3.7　试证明共轭梯度算法 3.13 和算法 3.14 满足定理 3.17。

3.8　试证明 Newton 算法 3.15 和算法 3.16 满足定理 3.17。

3.9　试证明：如果 $\mathbf{P} \in S_{++}^{n\times n}$，$\mathbf{A} \in \mathbb{R}^{m\times n}$，$\text{rank}(\mathbf{A}) = m < n$，$\begin{bmatrix} \mathbf{P} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$ 必然非奇异。

3.8 参考文献

[1] F. J. Gould, J. W. Tolle, "A necessary and sufficient qualification for constrained optimization," *SIAM Journal on Applied Mathematics*, vol. 20, no. 2, pp. 164-172, 1971.

[2] D. W. Peterson, "A review of constraint qualifications in finite-dimensional spaces," *SIAM Review*, vol. 15, no. 3, pp. 639-654, 1973.

[3] W. Sun, Y. X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Science & Business Media, 2006.

[4] Y. Nesterov, A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, 1994.

[5] C. Roos, T. Terlaky, J.-P. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, Wiley, 1997.

[6] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, 2001.

[7] T. Terlaky, "Twenty-Five years of interior point methods," *INFORMS Tutorials in Operations Research*, 2009.

[8] J. Gondzio, "Interior point methods 25 years later," *European Journal of Operational Research*, vol. 218, no. 3, pp. 587-601, 2012.

第4章 整数规划

4.1 引言

整数规划（Integer Programming，简称 IP）也是运筹学中的一个重要分支，研究含有整数决策变量的极值规划问题 [1]-[2]。整数规划和线性规划研究密切联系在一起，常见的求解方法是在线性规划解法的基础上建立起来的。目前，整数规划，特别是混合整数线性规划（Mixed-Integer Linear Programming，简称 MILP）在工程技术、商业活动、军事行动和工农业生产中发挥着越来越广泛的作用。

本章先介绍整数规划的基本概念及整数规划问题的建模技巧；之后介绍整数规划问题中广泛使用的切平面法和分支定界法；最后介绍 Benders 分解法求解混合整数线性规划问题。切平面法和分支定界法本质都是用线性约束去等价替代整数约束进行松弛求解，区别在于加什么样的线性约束才能等价替代整数约束，从而求出最优解。

4.2 整数规划基础

4.2.1 整数规划问题的定义

定义 4.1. 混合整数线性规划问题的标准形式（standard form）为如下形式

$$
\begin{aligned}
\min_{x_1,\ldots,x_n,y_1,\ldots,y_l} \quad & z = c_1x_1 + \ldots + c_nx_n + d_1y_1 + \ldots + d_ly_l \\
\text{s.t.} \quad & a_{11}x_1 + \ldots + a_{1n}x_n + f_{11}y_1 \ldots + f_{1l}y_l = b_1 \\
& a_{21}x_1 + \ldots + a_{2n}x_n + f_{21}y_1 \ldots + f_{2l}y_l = b_2 \\
& \ldots \\
& a_{m1}x_1 + \ldots + a_{mn}x_n + f_{m1}y_1 \ldots + f_{ml}y_l = b_m \\
& x_1 \geq 0, \ldots, x_n \geq 0, y_1, \ldots, y_l \in \mathbb{Z}
\end{aligned}
\tag{4.1}
$$

其中 $x_i, a_{ij}, f_{ik}, b_i, c_i, d_k \in \mathbb{R}$，$y_1, ..., y_l \in \mathbb{Z}$，$i = 1, ..., m$，$j = 1, ..., n$，$k = 1, ..., l$。$x_i$，$y_k$ 为待定的决策变量。一般的，我们有 $m < n + l$，也即约束条件个数小于决策变量的个数。

我们也可以矩阵向量的形式将此规范形式记为

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z = \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{F}\mathbf{y} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \in \mathbb{Z} \end{aligned} \tag{4.2}$$

其中向量 $\mathbf{x} \in \mathbb{R}^n$，$\mathbf{y} \in \mathbb{Z}^k$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{A} \in \mathbb{R}^{m \times n}$，$\mathbf{F} \in \mathbb{R}^{m \times k}$，$\mathbf{c} \in \mathbb{R}^n$，$\mathbf{d} \in \mathbb{R}^k$。

如果仅含整数的整数线性规划问题（Integer Linear Programming），则写为

$$\begin{aligned} \min_{y_1,...,y_l} \quad & z = d_1 y_1 + ... + d_l y_l \\ \text{s.t.} \quad & f_{11} y_1 ... + f_{1l} y_l = b_1 \\ & f_{21} y_1 ... + f_{2l} y_l = b_2 \\ & ... \\ & f_{m1} y_1 ... + f_{ml} y_l = b_m \\ & y_1, ..., y_l \in \mathbb{Z} \end{aligned} \tag{4.3}$$

或

$$\begin{aligned} \min_{\mathbf{y}} \quad & z = \mathbf{d}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{F}\mathbf{y} = \mathbf{b} \\ & \mathbf{y} \in \mathbb{Z} \end{aligned} \tag{4.4}$$

在很多应用中，我们还会遇到 0-1 型整数线性规划（Binary Linear Programming）问题，其中所有变量只能等于 0 或 1，称之为 0-1 型变量。0-1 型整数线性规划问题属于最为简单的一类整数规划问题；但研究者早就证明了，即使 0-1 型整数线性规划问题在最差情况下也是多项式复杂程度的非确定性问题 [3]。

此外，我们还可能遇到非线性整数规划（Nonlinear Integer Programming）问题。也即目标函数和约束中至少有一个非线性，决策变量只含有整数变量的整数规划问题。非线性整数规划的难度往往要比线性整数规划问题更难求解 [4]-[7]。

一般而言，我们建模的整数规划问题要么无解，要么存在有界的最优解。本书以下讨论不涉及存在无界的解的整数规划问题。

4.2.2 整数规划问题的建模

应用整数规划建模实际应用问题的时候，有些技巧需要注意。兹举数例如下。

例 4.1.［背包问题（Knapsack Problems）[8]］一个背包可以容纳的重量上限为W。目前有n件物品可以放入背包，第i件物品的重量为w_i，价值为v_i，$i=1,...,n$。试问如何选择物品装入背包，使总价值最大。

解：我们设 0-1 型变量$x_i \in \{0,1\}$，$i=1,...,n$，表示是否选择第i件物品。$x_i=1$表示选择第i件物品；$x_i=0$表示不选择第i件物品。由此建立数学规划模型：

$$\max_{x_i} \quad z=\sum_{i=1}^{n} v_i x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \le W, x_i \in \{0,1\}$$

例 4.2. 现有资金B，可以投资 7 个项目，第i件项目的投资额和收益分别为a_i和c_i，$i=1,...,7$。但要考虑三个条件：1）若选择项目 1 就必须选择项目 2；2）项目 3 和项目 4 至少选一个；3）项目 5、6、7 中选两个。试问如何投资使总效益最大？

解：我们设 0-1 型变量$x_i \in \{0,1\}$，$i=1,...,7$，表示是否投资第i个项目。$x_i=1$表示投资第i个项目；$x_i=0$表示不投资第i个项目。由此建立规划模型：

$$\max_{x_i} \quad z=\sum_{i=1}^{7} c_i x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{7} a_i x_i \le B, x_i \in \{0,1\}$$

$$x_2 \ge x_1, x_3 + x_4 \ge 1, x_5 + x_6 + x_7 = 2$$

例 4.3. 某航空公司计划在全国选择若干个机场组建基地。设在机场i组建基地所需费用为c_i，$i=1,...,n$。若该公司在机场i和机场j的基地组建完成，则可开通往返两地的航班并获得票款收益r_{ij}，$1 \le i < j \le n$。该航空公司基地组建费用上限为B。试问应选择在哪些机场组建基地才能使获得的收益最大？

解：我们设 0-1 型变量$x_i \in \{0,1\}$，$i=1,...,n$，表示是否在机场i设立基地。$x_i=1$表示设立基地；$x_i=0$表示不设立基地。由此建立数学规划模型：

$$\max_{x_i} \quad z=\sum_{1\le i<j\le n} r_{ij}x_ix_j-\sum_{i=1}^{n}c_ix_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n}c_ix_i\le B, x_i\in\{0,1\}$$

但这个数学规划问题的目标函数不是关于变量 x_i 的线性函数。实际上，我们还可以引入变量 0-1 型变量 $y_{ij}\in\{0,1\}$，$1\le i<j\le n$，表示是否开通机场 i 和机场 j 之间的航班。$y_{ij}=1$ 表示开通航班；$y_{ij}=0$ 表示不开通航班。由此，我们可以将目标函数设为 x_i 和 y_{ij} 的函数，建立整数线性规划模型为：

$$\max_{x_i,y_{ij}} \quad z=\sum_{1\le i<j\le n} r_{ij}y_{ij}-\sum_{i=1}^{n}c_ix_i$$
$$\text{s.t.} \quad \sum_{i=1}^{n}c_ix_i\le B, x_i\in\{0,1\}$$
$$y_{ij}\le x_i, y_{ij}\le x_j, y_{ij}\ge x_i+x_j-1, y_{ij}\in\{0,1\}$$

例 4.4. [旅行商问题（Traveling Salesman Problems，TSP）[9]] 一个推销员从城市 v_0 出发，要遍访城市 $\{v_1,...,v_n\}$ 有且仅有一次，最后返回城市 v_0。从城市 v_i 到城市 v_j 的旅费为 c_{ij}，$i,j=1,...,n$。试问如何安排城市的访问顺序使得总旅费最低？

解：我们先看一种不正确的做法，设 0-1 型变量 $x_{ij}\in\{0,1\}$，$i=1,...,n$，表示是否选择从城市 v_i 到城市 v_j 的路径或者从城市 v_j 到城市的路径 v_i。$x_{ij}=1$ 表示选择该种路径；$x_{ij}=0$ 表示不选择该种路径。假设 $c_{ij}=c_{ji}$，我们可以建立数学规划模型：

$$\min_{x_{ij}} \quad z=\sum_{i=0}^{n}\sum_{j=0}^{n}c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n}x_{ij}=2, i=0,...,n$$

$$x_{ij}=x_{ji}, x_{ij}\in\{0,1\}$$

但该模型不能避免出现多于一个的互不联通的旅行路线回路；参见图 4.1（a）。

（a）　　（b）

图 4.1　旅行商问题求解示意图：（a）一个含有两个互不联通旅行路线回路的错误解；（a）一个满足要求的可行解。

为此，研究者提出了多种解决方案，例如下述的 Miller-Tucker-Zemlin 模型。设 0-1 型变量 $x_{ij}\in\{0,1\}$，$i=1,...,n$，表示是否选择从城市 v_i 到城市 v_j 的路径。$x_{ij}=1$ 表示选择该路径；$x_{ij}=0$ 表示不选择该路径。由此建立数学规划模型：

$$\min_{x_{ij},u_i} \quad z=\sum_{i=0}^{n}\sum_{j=0}^{n}c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{n}x_{ij}=1, i=0,...,n$$

$$\sum_{j=1}^{n}x_{ij}=1, i=0,...,n$$

$$u_i-u_j+nx_{ij}\le n-1, i,j=1,...,n$$

$$x_{ij}\in\{0,1\}, i,j=0,...,n, \quad u_i\in R, i=1,...,n$$

上面的约束条件中，第一条保证每个城市都被离开一次，第二条保证每个城市都被进入一次，第三条仅仅使得原问题增加了 n 个连续变量和 n^2 个的不等式约束，即可保证不会出现多于一个的互不联通的旅行路线回路。为了理解这点，不妨考虑一个包含城市 v_3 到城市 v_4 的路径，则有 $u_3-u_4+nx_{34}\le n-1$，注意到

$x_{34}=1$，实际上就是$u_3+1\le u_4$。不难看出，有第三条约束之后，u_i的取值必须沿着回路增加，每通过一个城市至少增加1。如果存在一个不包含城市v_0的长度为m的回路，则无论u_i如何取值，必然最终出现形如$u_m+m\le u_m$这样的矛盾。

例 4.5. 如果一个规划问题的可行域如下图 4.2 所示，目标函数为$\max\limits_{x_1,x_2} x_2$，请引入整数变量将起建模为整数规划问题。

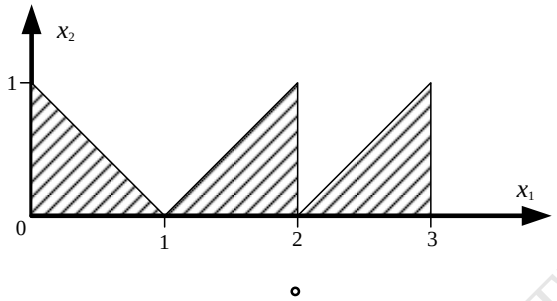


图 4.2　一个非凸的可行域示意图。

解：我们不难看出该问题的整个可行域可以视为三个凸集的并集。

第一个凸集可以表述为：$-x_1\le 0,-x_2\le 0,x_1+x_2\le 1$；

第二个凸集可以表述为：$x_1\le 2,-x_2\le 0,-x_1+x_2\le -1$；

第三个凸集可以表述为：$x_1\le 3,-x_2\le 0,-x_1+x_2\le -2$。

我们可以引入如下的 0-1 整数变量$y_1,y_2,y_3\in\{0,1\}$，其中$y_i=0$表示我们选择第i个凸集作为实际可行域，$y_i=1$表示我们未选择第i个凸集作为实际可行域；$i=1,2,3$。则我们可以将整个可行域表述为

$$\begin{cases}-x_1\le 0+y_1M_1,-x_2\le 0+y_1M_1,x_1+x_2\le 1+y_1M_1\\ x_1\le 2+y_2M_2,-x_2\le 0+y_2M_2,-x_1+x_2\le -1+y_2M_2\\ x_1\le 3+y_3M_3,-x_2\le 0+y_3M_3,-x_1+x_2\le -2+y_3M_3\\ y_1+y_2+y_3=2\end{cases}$$

其中M_1,M_2,M_3为足够大的正实数。

不难看出，当我们选择第一个凸集作为实际可行域时，$y_1=0$而$y_2=y_3=1$，则只有第一个凸集的约束条件起作用，而第二个凸集和第三个凸集的约束条件自然满足。当我们选择第二个凸集作为实际可行域时，$y_2=0$而$y_1=y_3=1$，则只有第二个凸集的约束条件起作用，而第一个凸集和第三个凸集的约束条件自然满足。当我们选择第三个凸集作为实际可行域时，$y_3=0$而$y_1=y_2=1$，则只有第三个凸集的约束条件起作用，而第一个凸集和第二个凸集的约束条件自然满足。

实际上，若要从 p 个约束条件中选用 q 个约束条件同时满足的时候，我们以可以类似的引入 0-1 整数变量 $y_1,...,y_p \in \{0,1\}$ 和足够大的正实数 $M_1,...,M_p$，类似上面表述整个可行域的，并且令 $\sum_{i=1}^{p} y_i = p - q$。

例 4.6. [选址覆盖问题（Facility Covering Problem）] 假设已知所有 m 个需求点的位置和 n 个备选的设施位置，以及每个设施的服务范围。请建模求解在满足覆盖所有需求点的前提下，选取若干个点建造设施，以使得建设总成本最小。

解：设 0-1 型变量 $x_j \in \{0,1\}$，$j = 1,...,n$，表示是否选择在第 j 个备选设施位置设立服务设施，$x_j = 1$ 表示选择修建；$x_j = 0$ 表示不选择。以 a_{ij}，$i = 1,...,m$，$j = 1,...,n$，表示覆盖范围，$a_{ij} = 1$ 表示在第 j 个备选设施位置设立服务设施能够服务第 i 个需求点；$a_{ij} = 0$ 表示在第 j 个备选设施位置设立服务设施不能服务第 i 个需求点。则相应的数学规划模型为

$$\begin{aligned} \min_{x_j} \quad & z = \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1,\ i = 1,...,m \\ & x_j \in \{0,1\},\ j = 1,...,n \end{aligned}$$

例 4.7. [选址匹配问题（Facility Match Problem）] 假设已知所有 m 个需求点的位置和 n 个备选的设施位置，以及每对需求点和备选设置之间的空间距离。请建模求解在满足覆盖所有需求点的前提下，选取修建最多 p 个设施，并指定需求点和设置的配对方式，以使得所有需求点到达其所属设施的距离的最大值最小。如果进一步已知每个需求点需要服务的人数，目标函数改为使得所有需求点的客户到达其所属设施的距离总和最小，重新规划。

解：设 0-1 型变量 $x_j \in \{0,1\}$，$j = 1,...,n$，表示是否选择在第 j 个备选设施位置设立服务设施，$x_j = 1$ 表示选择修建；$x_{ij} = 0$ 表示不选择。0-1 型变量

$y_{ij} \in \{0,1\}$，$i=1,...,m$，$j=1,...,n$，表示是否将第i个需求点分配给第j个服务设施，$y_{ij}=1$表示选择分配；$y_{ij}=0$表示不选择。d_{ij}表示第i个需求点到第j个备选设施位置的距离。则所有需求点到达其所属设施的距离的最大值最小相应的数学规划模型为

$$\begin{aligned}
\min_{x_j, y_{ij}, w} \quad & w \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_j = p \\
& \sum_{j=1}^{n} y_{ij} = 1,\ i=1,...,m \\
& y_{ij} \le x_j,\ i=1,...,m,\ j=1,...,n \\
& \sum_{j=1}^{n} d_{ij} y_{ij} \le w,\ i=1,...,m,\ j=1,...,n \\
& x_j \in \{0,1\}, y_{ij} \in \{0,1\},\ j=1,...,n,\ i=1,...,m
\end{aligned}$$

如果第i个需求点有h_i个人需要服务，则所有需求点的客户到达其所属设施的距离总和最小相应的数学规划模型为

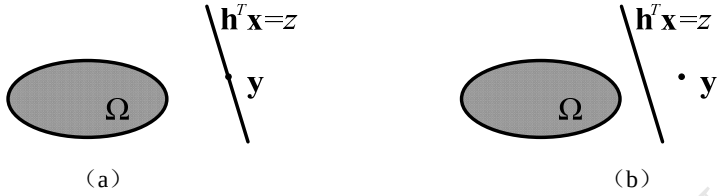
$$\begin{aligned}
\min_{x_j, y_{ij}} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} h_i d_{ij} y_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_j = p \\
& \sum_{j=1}^{n} y_{ij} = 1,\ i=1,...,m \\
& y_{ij} \le x_j,\ i=1,...,m,\ j=1,...,n \\
& x_j \in \{0,1\}, y_{ij} \in \{0,1\},\ j=1,...,n,\ i=1,...,m
\end{aligned}$$

4.3 切平面法

在介绍整数规划算法之前，我们先介绍一下一般化的切平面法的概念，因为在很多问题都可以应用切平面的思想进行求解 [10]-[15]。

定义 4.2. 给定集合$\Omega \in \mathbb{R}^n$和点$\mathbf{y} \in \mathbb{R}^n$，一个切平面预言（Cutting Plane Oracle）要么确定$\mathbf{y} \in \Omega$，要么确定一个超平面$H = \left\{\mathbf{x} \mid \mathbf{h}^T\mathbf{x} = z, \mathbf{x} \in \mathbb{R}^n\right\}$分隔集合$\Omega$和点$\mathbf{y}$，也即满足$\Omega \subset \left\{\mathbf{x} \mid \mathbf{h}^T\mathbf{x} > z\right\}$且$\mathbf{h}^T\mathbf{y} \le z$。如果$\Omega$是凸集，根据凸集分隔定理，我们知道这个预言必然是成立的。

如果$\mathbf{y}$在该超平面上，也即$\mathbf{h}^T\mathbf{y} = z$，我们称之为中性切（Neutral Cut）；如果$\mathbf{y}$不在该超平面上，也即$\mathbf{h}^T\mathbf{y} < z$，我们称之为深度切（Deep Cut）。参见图 4.3。



（a）　　（b）

图 4.3　切平面示意图：（a）中性切；（a）深度切。

考虑如下优化问题

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \le 0, i = 1, ..., m \end{aligned} \tag{4.5}$$

其中$\mathbf{x} \in \mathbb{R}^n$，函数$f_i : \mathbb{R}^n \to \mathbb{R}$，$i = 0, ..., m$。

我们可以引入变量$t \in \mathbb{R}$，将上述问题等价变形为

$$\begin{aligned} \min_{\mathbf{x},t} \quad & t \\ \text{s.t.} \quad & f_0(\mathbf{x}) \le t \\ & f_i(\mathbf{x}) \le 0, i = 1, ..., m \end{aligned} \tag{4.6}$$

这样可以统一的考虑目标函数的优化和可行区域的界定。

假设约束条件构成的非空可行域为$\Omega \in \mathbb{R}^n$。对于优化问题（4.6），一般化的切平面法的想法是通过产生一系列的探索点，在每一个探索点处进行一次切平面预言，利用一系列新产生的切平面来逐渐缩小可行域，直到找到最优解。

特别的，如果优化问题（4.6）是凸优化问题，且函数$f_i : \mathbb{R}^n \to \mathbb{R}$，$i = 0, ..., m$是光滑的。任意给一个点$\mathbf{z} \in \mathbb{R}^n$，如果$\mathbf{z} \in \Omega$，则我们可以增加如下的最优性切平面（Optimality Cut）来改善我们对于最优目标函数值的下界估计

$$f_0(\mathbf{z}) + \nabla f_0(\mathbf{z})^T (\mathbf{x} - \mathbf{z}) \le t \tag{4.7}$$

以及对于最优目标函数值的上界估计

$$t \le f_0(\mathbf{z}) \tag{4.8}$$

如果 $\mathbf{z}\notin\Omega$，不妨假设其违反了如下约束条件 $f_j(\mathbf{z})>0$， $j\in\{1,...,m\}$，则我们可以增加如下的可行性切平面（Feasibility Cut）来缩小可行域，改善我们对于最优解所在位置的估计

$$f_j(\mathbf{z})+\nabla f_j(\mathbf{z})^T(\mathbf{x}-\mathbf{z})\le 0,\quad j\in\{1,...,m\} \tag{4.9}$$

我们把相应得到线性规划问题记为

$$\begin{aligned}\min_{\mathbf{x},t}\quad & t\\ \text{s.t.}\quad & f_0(\mathbf{z}_1^{(k)})+\nabla f_0(\mathbf{z}_1^{(k)})^T(\mathbf{x}-\mathbf{z}_1^{(k)})\le t\\ & t\le f_0(\mathbf{z}_1^{(k)})\\ & f_j(\mathbf{z}_2^{(k)})+\nabla f_j(\mathbf{z}_2^{(k)})^T(\mathbf{x}-\mathbf{z}_2^{(k)})\le 0,\ j=\{1,...,m\}\end{aligned} \tag{4.10}$$

其中 $\mathbf{z}_1^{(k)}\in\Omega$， $\mathbf{z}_2^{(k)}\notin\Omega$ 表示产生的一系列切平面探索点，$k\in\mathbb{N}$。

定理 4.1. 如果 $\mathbf{x}^*$ 是凸优化问题（4.6）的最优解， $\mathbf{x}_*,t_*$ 是对应切平面法产生的线性规划问题（4.10）的最优解，则 $t_*\le f_0(\mathbf{x}^*)=t^*$。

证明：显然 $\mathbf{x}_*,t_*$ 是对应切平面法产生的线性规划问题（4.10）的最优解，则其必然位于搜索过程中产生的某个最优性切平面之上，不妨设其位于如下的超平面

$$t_*=f_0(\mathbf{z}_1^{(k)})+\nabla f_0(\mathbf{z}_1^{(k)})^T(\mathbf{x}_*-\mathbf{z}_1^{(k)})$$

考虑到规划问题（4.6）是凸优化问题，根据定理 1.13 所述凸函数的一阶条件，位于这个超平面上的点应该满足

$$f_0(\mathbf{z}_1^{(k)})+\nabla f_0(\mathbf{z}_1^{(k)})^T(\mathbf{x}^*-\mathbf{z}_1^{(k)})\le t^*$$

由于 $\mathbf{x}_*,y_*$ 是对应切平面法产生的线性规划问题（4.10）的最优解，则

$$t_*=f_0(\mathbf{z}_1^{(k)})+\nabla f_0(\mathbf{z}_1^{(k)})^T(\mathbf{x}_*-\mathbf{z}_1^{(k)})\le f_0(\mathbf{z}_1^{(k)})+\nabla f_0(\mathbf{z}_1^{(k)})^T(\mathbf{x}^*-\mathbf{z}_1^{(k)})\le t^*$$

因此，结论成立。□

定理 4.2. 如果我们把迭代到第 k 次的线性规划问题（4.10）的最优目标函数值记为 $t^{(k)}$，如果我们把迭代到第 $k+1$ 次的线性规划问题（4.10）的最优目标函数值记为 $t^{(k+1)}$，则 $t^{(k)}\le t^{(k+1)}$。

证明：如果迭代到第 $k+1$ 次的探索点不可行，则迭代到第 $k+1$ 次的线性规划问题将增加一个可行性切面，这意味着新的线性规划问题多了一个约束条件。换句话说，新的线性规划问题的可行区域变小了。由于这一系列求最小值的线性

规划问题的目标函数相同，较小的可行区域不可能产生较大的目标值。此时有 $t^{(k)} \le t^{(k+1)}$。

如果迭代到第 $k+1$ 次的探索点可行，则迭代到第 $k+1$ 次的线性规划问题将增加一个最优性切面。这直接为新的线性规划问题的最优解设定了一个上限。此时同样有 $t^{(k)} \le t^{(k+1)}$。

综上所述，我们有 $t^{(k)} \le t^{(k+1)}$。□

定理 4.1 和定理 4.2 告诉我们，切平面法通过建立松弛话的线性规划问题，并不断加入新的线性约束条件，得到凸优化问题（4.6）最优解的一系列下界估计，如果原凸优化问题存在有界最优解，则切平面方面会最终收敛到最优解。对于如何产生探索点，在每次迭代时尽可能多的切去不含最优解的部分可行域有不少研究成果，参见[14]-[15]。

推广到一般情况，假定针对最小化问题，我们得到切平面算法流程如下：

算法 4.1. 通用切平面算法

Step 1. 确定最初的多面体可行域 $\Omega^{(1)} \in \mathbb{R}^n$，保证最优解在 $\Omega^{(1)}$ 中。令 $\text{LowerBound} = -\infty$， $\text{UpperBound} = +\infty$。取最初的探索点 $\mathbf{z}^{(1)} \in \Omega^{(1)}$，$k := 1$，进入 Step 2。

Step 2. 求得当前最优化问题的最优解 $t^{(k)} \in \mathbb{R}$。

如果 $\mathbf{z}^{(k)} \notin \Omega^{(k)}$，令 $\text{LowerBound} = t^{(k)}$，并加入可行性切平面约束条件 $\mathbf{h}_k^T \mathbf{x} \le \mathbf{h}_k^T \mathbf{z}^{(1)}$， $\mathbf{h}_k \in \mathbb{R}^n$ 为恰当的法向量，得到新的多面体可行域 $\Omega_{k+1} \in \mathbb{R}^n$，进入 Step 3。

否则， $\mathbf{z}^{(k)} \in \Omega^{(k)}$，令 $\text{UpperBound} = t^{(k)}$，并加入最优性切平面约束条件 $t^{(k)} + \mathbf{h}_k^T \left(\mathbf{x} - \mathbf{z}^{(k)} \right) \le t$， $t \le t^{(k)}$， $\mathbf{h}_k \in \mathbb{R}^n$ 为恰当的法向量，得到新的多面体可行域 $\Omega_{k+1} \in \mathbb{R}^n$，进入 Step 3。

Step 3. 如果 $\mathbf{z}^{(k)} \in \Omega^{(k)}$，且 $\text{UpperBound-LowerBound} \le \varepsilon$， $\varepsilon > 0$ 为预先设定的阈值，则找到最优解，停止。

如果 $\mathbf{z}^{(k)} \notin \Omega^{(k)}$，且 $\Omega_{k+1} = \phi$，则问题无解，停止。

否则，产生新的探索点 $\mathbf{z}^{(k+1)} \in \Omega^{(k+1)}$， $k := k+1$，回到 Step 2，继续迭代。

例 4.8. 用切平面法优化如下非线性问题

$$\begin{aligned} \min_{x} \quad & \left(0.5x - 2\right)^2 + 3 \\ \text{s.t.} \quad & 2 \le x \le 7 \end{aligned}$$

解：假设初始探索点 $z_1 = 2$，我们可以得到

$$\begin{aligned}\min_{x,y}\quad & y\\ \text{s.t.}\quad & -(x-1)+4\le y\\ & 2\le x\le 7\end{aligned}$$

该线性规划问题最优解在 $x=7$ 处取得，令新探索点 $z_2=7$，我们可以得到

$$\begin{aligned}\min_{x,y}\quad & y\\ \text{s.t.}\quad & -(x-1)+4\le y, 1.5(x-7)+\frac{21}{4}\le y\\ & 2\le x\le 7\end{aligned}$$

该线性规划问题最优解在 $x=9/2$ 处取得，令新探索点 $z_2=9/2$，我们可以得到

$$\begin{aligned}\min_{x,y}\quad & y\\ \text{s.t.}\quad & -(x-1)+4\le y, 1.5(x-7)+\frac{21}{4}\le y, \frac{1}{4}\left(x-\frac{9}{2}\right)+\frac{49}{16}\le y\\ & 2\le x\le 7\end{aligned}$$

如图 4.4 所示，我们可以通过增加切平面逐渐逼近原非线性问题的最优解。

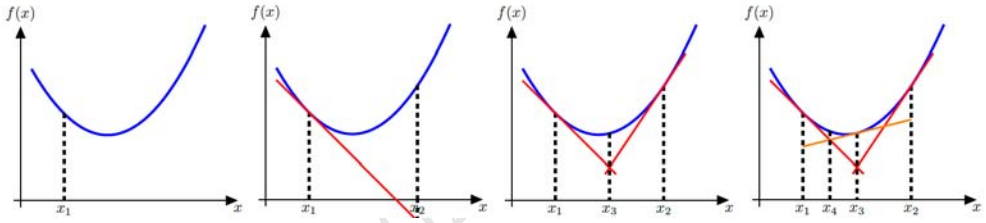


图 4.4　例 4.8 的每一步求解过程中加入的超平面示意图。

4.4　纯整数规划的 Gomory 切平面法

很多整数规划问题的可行域包含有限多个整数可行点（可行解）。一种最简单直观的想法就是枚举所有的整数可行点，找到目标函数值最低的整数可行点。但这样算法往往效率很低。例如，如果一个问题含有 n 个 0-1 型整数变量，则其可行域最多可以包含 2^n 个整数可行点。当 $n\ge 50$，我们用当前的计算机可能需要花费数月以上的时间枚举所有的整数可行点，这显然无法实用。

第二种方法是考虑去掉整数约束，转为线性规划问题求解之后再对求得的解就近取整。但这种简单的先松弛求解再就近取整策略无法得到最优解。

例 4.9. 考虑如下的整数线性规划问题

$$\max_{y_1,y_2} \quad y_1+5y_2$$

$$\text{s.t.} \quad -\frac{5}{8}y_1+y_2\le 1, y_1+\frac{8}{5}y_2\le 8$$

$$y_1,y_2\ge 0, y_1,y_2\in\mathbb{Z}$$

如图 4.1 所示，上述整数规划问题的可行域Ω是多面体P内部和边界上的所有的整数点。如果我们去掉整数约束的话该问题就变为所谓的松弛线性规划问题，这样的松弛线性规划问题的可行域就是多面体P。很显然，松弛线性规划问题的可行域P包含了整数规划问题的可行域Ω，也即$\Omega\subset P$。

利用图解法，我们可以得到这里松弛线性规划问题的最优解是$(16/5,3)$。通过就近取整，我们可以得到离这个解最近的两个整数可行点为$(3,2)$。但这个点不是该问题的最优解。从图上不难得知，原问题真正的最优解为整数可行点$(4,2)$。这说明，简单取整策略往往不能得到原问题的最优解。

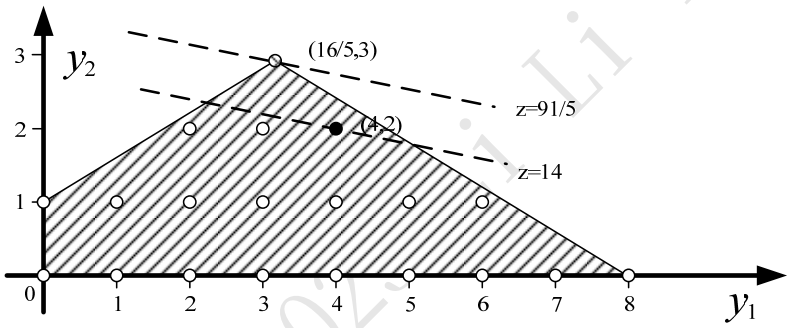


图 4.5　例 4.7 的可行域示意图。

为此，研究者对于纯整数线性规划问题提出了第三种方法，Gomory 切平面法（Gomory's Fractional Cutting Plane Algorithm）[16]。首先利用松弛线性规划问题找到一个初始解。然后判断当前解是否是整数解。如果不是，则利用切平面法，在当前解的基础上往松弛线性规划问题引进更多的线性约束条件，减小当前松弛线性规划问题的可行域。反复迭代这一过程，直到找到整数解，则该解必为原问题的最优解。

考虑如下的纯整数线性规划问题

$$\begin{aligned} \min_{\mathbf{y}} \quad & z=\mathbf{d}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{F}\mathbf{y}=\mathbf{b} \\ & \mathbf{y}\geq\mathbf{0},\mathbf{y}\in\mathbb{Z} \end{aligned} \tag{4.11}$$

Gomory 切平面法的基本流程如下：

算法 4.2. Gomory 切平面法

Step 1. 首先将规划问题（4.10）松弛为

$$\begin{aligned} \min_{\mathbf{y}} \quad & z=\mathbf{d}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{F}\mathbf{y}=\mathbf{b} \\ & \mathbf{y}\geq\mathbf{0} \end{aligned} \tag{4.12}$$

并求解。如果线性规划问题（4.12）无解，则原问题（4.11）也无解，停止；否则线性规划问题(4.12)存在最优解 $\mathbf{y}^{(0)}$，如果 $\mathbf{y}^{(0)}$ 是整数，则已经找到原问题(4.11)最优解，停止；否则进入 Step 2。

Step 2. 对于当前规划问题最优解 $\mathbf{y}^{(k)}$，我们可以将对应的基本解和非基本解记为 $\mathbf{y}_B$ 和 $\mathbf{y}_N$，可行基矩阵和非可行基矩阵记为 $\mathbf{F}_B$ 和 $\mathbf{F}_N$，则最优解满足

$$\begin{cases} \mathbf{y}_B+\mathbf{F}_B^{-1}\mathbf{F}_N\mathbf{y}_N=\mathbf{F}_B^{-1}\mathbf{b} \\ \left(\mathbf{d}_N^T-\mathbf{d}_B^T\mathbf{F}_B^{-1}\mathbf{F}_N\right)\mathbf{y}_N=z-\mathbf{d}_B^T\mathbf{F}_B^{-1}\mathbf{b} \end{cases}$$

令 $\overline{\mathbf{F}}=\mathbf{F}_B^{-1}\mathbf{F}_N=\left[\overline{\boldsymbol{\alpha}}_{m+1},...,\overline{\boldsymbol{\alpha}}_n\right]=\left[\overline{a}_{ij}\right]\in R^{m\times(n-m)}$，$\overline{\mathbf{b}}=\mathbf{F}_B^{-1}\mathbf{b}=\left[\overline{b}_i\right]\in R_+^m$ 来简化表述上表中的对应矩阵分块，其中 $\overline{\boldsymbol{\alpha}}_j$，$j=1,...,n-m$，表示矩阵 $\overline{\mathbf{F}}$ 的各个列向量。$\overline{a}_{ij}$，$i=1,...,m$，$j=m+1,...,n$ 表示矩阵 $\overline{\mathbf{F}}$ 的各个元素。$\overline{b}_i$ 表示向量 $\overline{\mathbf{b}}$ 的各个元素。

任意取 $\mathbf{y}^{(k)}$ 的一个非整数基变量，不妨设为第 i 个元素 $\mathbf{y}_{B_i}$，可以得到对应的约束条件

$$\mathbf{y}_{B_i}+\sum_{j=m+1}^{n}\overline{a}_{ij}y_j=\overline{b}_i \tag{4.13}$$

我们用 $\lfloor a\rfloor$ 表示不超过实数 a 的最大整数，例如 $\lfloor 3.5\rfloor=3$，$\lfloor -1.6\rfloor=-2$，$\lfloor 0\rfloor=0$。令

$$\overline{a}_{ij}=\lfloor\overline{a}_{ij}\rfloor+f_{ij},\quad j=m+1,...,n,\quad \overline{b}_i=\lfloor\overline{b}_i\rfloor+h_i \tag{4.14}$$

注意到 $\mathbf{y}\geq\mathbf{0}$，有

$$\sum_{j=m+1}^{n} \lfloor \bar{a}_{ij} \rfloor y_j \le \sum_{j=m+1}^{n} \bar{a}_{ij} y_j$$

代入等式（4.13）中有

$$\mathbf{y}_{B_i} + \sum_{j=m+1}^{n} \lfloor \bar{a}_{ij} \rfloor y_j \le \bar{b}_i \tag{4.15}$$

由于 $\mathbf{y} \in \mathbb{Z}$，因此不等式（4.15）左侧为整数。因此，将不等式（4.15）右侧用 $\lfloor \bar{b}_i \rfloor$ 代替后，不等式依然成立

$$\mathbf{y}_{B_i} + \sum_{j=m+1}^{n} \lfloor \bar{a}_{ij} \rfloor y_j \le \lfloor \bar{b}_i \rfloor \tag{4.16}$$

将（4.13）带入（4.16），可得到一个可行性切平面（Gomory 切平面）

$$\sum_{j=m+1}^{n} f_{ij} y_j \ge h_i \tag{4.17}$$

并进入 Step 3。

Step 3. 我们将 Step 2 中新得到的切平面不等式（4.17）两边同时乘以 -1，然后引入一个松弛变量 s，等到一个等式约束条件加入当前规划问题

$$-\sum_{j=m+1}^{n} f_{ij} y_j + s = -h_i \tag{4.18}$$

并用对偶单纯型法求解这个新的松弛规划问题，得到新的最优解 $\mathbf{y}^{(k+1)}$，如果 $\mathbf{y}^{(k+1)}$ 是整数，则已经找到原问题（4.11）最优解，停止；否则返回 Step 2。

定理 4.3. 如果原问题存在最优整数解，则上述 Gomory 切平面法可以找到。

证明：由于等式约束条件（4.18）是由原纯整数规划问题的整数约束推出来的，所以其并未割掉任何原问题的整数可行点。同时根据（4.13）和（4.17），容易验证上一轮求解出的规划问题最优解 $\mathbf{y}^{(k)}$ 不满足新增加的等式约束条件（4.17），因此该切平面加入后割掉了上一轮的最优解 $\mathbf{y}^{(k)}$，也即每次迭代都确实缩小了松弛问题的可行域，直到找到最优解停止。

此外，每轮迭代时新引入的松弛变量 s 和原来的基变量一起构成了新松弛规划问题的新基本解。当 $\bar{b}_i$ 不是整数时，$h_i > 0$，因此新松弛规划问题的基本解中有 $s = -h_i < 0$。不仅如此，由于增加新约束条件（4.18）后，单纯型表中的检验数未改变，因此新基本解满足对偶可行性。因此，可以使用对偶单纯型法求解新松弛规划问题。□

注意，上述 Gomory 切平面法不能用于混合整数规划问题，因为对于混合整数规划问题，我们不能保证从（4.15）推出（4.16）。

例 4.10. 用 Gomory 切平面法求解例题 4.7。

解：

4.5 分支定界法

分支定界法（Branch and Bound，也被称为分枝界限法）属于分而治之算法（Divide and Conquer），对于整数规划、组合优化等问题都是一种有效的求解方法 [17]-[20]。其基本思想是将问题的可行域逐渐分割为越来越小的若干子区域（称为分支），并在各自子区域内部求解计算出一个最小化目标函数的下界（称为定界），在每次分支后，凡是可能的最小化目标函数值会大于已知的可行解对应最小目标函数的那些子区域不再进一步分支（称为剪枝），比较得到不同子区域内获得局部最优解，最终得到全局最优解。

分支定界法是目前求解整数规划问题的最常用算法。这种方法不但可以求解纯整数规划，还可以求解混合整数规划问题。例如对于整数线性规划问题（4.2），我们常考虑如下分支方式，将问题的可行域分隔成两个部分：

$$\begin{array}{lll} \min\limits_{\mathbf{x},\mathbf{y}} & z=\mathbf{c}^T\mathbf{x}+\mathbf{d}^T\mathbf{y} & \min\limits_{\mathbf{x},\mathbf{y}} \quad z=\mathbf{c}^T\mathbf{x}+\mathbf{d}^T\mathbf{y} \\ \text{s.t.} & \mathbf{Ax}+\mathbf{Fy}=\mathbf{b} & \text{s.t.} \quad \mathbf{Ax}+\mathbf{Fy}=\mathbf{b} \\ & \mathbf{x}\geq\mathbf{0},\mathbf{y}\in\mathbb{Z} & \mathbf{x}\geq\mathbf{0},\mathbf{y}\in\mathbb{Z} \\ & y_i\leq z & y_i\geq z+1 \end{array} \qquad (4.19)$$

其中 z 为选定的可行的一个整数，y_i 是决策变量 $\mathbf{y}$ 的其中第 i 个元素。

在使用分支定界法求解整数规划问题时，会自然而然的构建出一个求解树。这颗树的根节点代表问题的整个可行域；这颗树的每一次分支代表将上一层节点的可行域进行一次不重不漏的分割；这颗树的叶节点代表不需要被进一步分支的可行域。在分支定界法求解过程中，我们不再进一步分支的当前节点被称为死点，还需要被进一步分支的当前节点被称为活点。

算法 4.3. 针对整数规划的分支定界法

Step 1. 令活点集合 Ω 代表最初的可行域。令 $\text{UpperBound}=+\infty$。令当前最优整数解为空集 $\mathbf{y}^*:=\phi$，进入 Step 2。

Step 2. 如果当前的活点集合为空集，则转向 Step 4。否则，从当前的活点集合中选择一个代表可行域 Ω_k 的分支节点 k，并从当前的活点集合中删掉该分支节点 k，进入 Step 3。

Step 3. 解所选分支节点 k 所对应松弛线性规划问题。

若此松弛问题无解，则进入 Step 2。

若此松弛问题有解 $\mathbf{y}_k$，但其对应的目标函数值 z_k 大于已知的最优整数解对应的目标函数值 UpperBound，则不再进一步分支该分支节点，进入 Step 2；否则，进一步考虑找到的松弛问题最优解 $\mathbf{y}_k$ 是否是整数解：如果是整数解，则更新当前已知的最优整数解为 $\mathbf{y}^* := \mathbf{y}_k$，$\text{UpperBound} = z_k$，进入 Step 2；如果不是整数解，则将 $\mathbf{y}_k$ 某个非整数元素生成分支节点 k 的两个后代节点，令这两个后代节点为活点并加入活点集合，进入 Step 2。

Step 4. 若当前最优整数解为空集，$\text{UpperBound} = +\infty$，则原整数规划问题无解。否则找到最优整数解 $\mathbf{y}^*$，最优目标函数值为 UpperBound，停止。

显然，上述 Step 2 中选择活点集合中的哪一个待分支节点对整个分支定界法的计算效率有着重要影响。一般来说，我们有两种选择策略。

第一种选择策略是选择当前目标函数值最小的待分支节点（对应优先队列式搜索，参见第 6 章的讨论）：我们在 Step 3 产生后代点的同时，让每个后代点记住此时前代点松弛问题目标函数值 z_k。在 Step 2，我们比较活点集合中所有的待分支节点所记住的前代点松弛问题目标函数值 z_k，找出其中最小的 z_k 所对应的待分支节点，做进一步分支。该方法的优点是分支数经常较少，能较快地求得最佳解；缺点是要存储很多待分支节点的前代点松弛问题目标函数值，花费更多内存空间。

第二种选择策略是选择尽可能靠近根节点的待分支节点（对应宽度优先式搜索，参见第 6 章的讨论）。该方法的优点是节省了空间；缺点是往往需要较多的分枝运算，耗费的时间较多。

此外，上述 Step 3 中选择 $\mathbf{y}_k$ 的哪一个非整数元素产生后代节点同样对整个分支定界法的计算效率有着重要影响。我们一般会定义一个打分函数来评价选择分支每个非整数元素的倾向性，然后选择打分最高的分支节点做进一步分支 [20]。常见的打分函数定义为：

$$\text{score}\left(\Delta_j^-, \Delta_j^+\right) = \lambda \min\left(\Delta_j^-, \Delta_j^+\right) + (1-\lambda)\max\left(\Delta_j^-, \Delta_j^+\right) \qquad (4.20)$$

其中 $\Delta_j^- = \max\left\{c_{Q_j^-} - c_Q, \varepsilon\right\}$，$\Delta_j^+ = \max\left\{c_{Q_j^+} - c_Q, \varepsilon\right\}$，$c_Q$ 表示当前松弛问题的最优解，$c_{Q_j^-}$ 和 $c_{Q_j^+}$ 表示如果用 $\mathbf{y}_k$ 的第 j 个元素做分支形成的两个子问题的目标函数。Δ_j^- 和 Δ_j^+ 表示采用如果用决策变量 $\mathbf{y}$ 的其中第 j 个元素做分支形成的两个子问题相对于当前松弛问题的函数值增加程度。引入小的正数 ε 是防止分支后的松弛问题无解的情况。加权因子 $\lambda \in [0,1]$ 由经验决定，一般选 $\lambda = 0.6$。

因为子问题是在更小的可行域上求最小值，所以子问题的目标函数值只会比

当前问题的目标函数更大。所以，显然，$\Delta_j^-, \Delta_j^+ \geq 0$。我们在分支定界的搜索过程中总是希望得到的松弛问题的目标函数值越大越好，这对于加速我们整个分支定界算法显然有着很大的帮助。由此，Δ_j^-, Δ_j^+ 越大，打分函数值越高，分支的效果越好。

进一步应用打分函数进行分支的时候，我们有如下四种方式：

第一种方式是强分支（Strong Branching）：对于 $\mathbf{y}_k$ 的所有可分支非整数元素进行二分支，求解每一个分支的 2 个线性松弛子问题，获得所有可分支元素的打分函数值，选择打分函数值最高的进行分支。数值实验表明强分支可以有效加速算法收敛，但是强分支每做一次分支需要求解多个线性规划问题，计算量太大。

第二种方式是伪打分分支（Pseudocost Branching）：在整个分支定界过程中来连续跟踪决策变量 $\mathbf{y}$ 的所有可分支元素在多次分支过程中对于目标函数的累计改进，进而估计选择某一个可分支元素对于目标函数下降的提升程度。

令 y_j 表示决策变量 $\mathbf{y}$ 的第 j 个非整数元素，我们记录每次对于 y_j 分支时，所选定的 $\hat{y}_j$ 向下取整和向上取整之后，两个子问题的目标函数针对取整残差的变化幅度

$$\mu_j^- = \frac{\Delta_j^-}{\hat{y}_j - \lfloor \hat{y}_j \rfloor}, \quad \mu_j^+ = \frac{\Delta_j^+}{\lceil \hat{y}_j \rceil - \hat{y}_j} \tag{4.21}$$

然后，利用历史信息的均值去估计针对 y_j 分支对于目标函数的影响

$$\eta_j^- = \frac{\sum \mu_j^-}{M}, \quad \eta_j^+ = \frac{\sum \mu_j^+}{M} \tag{4.22}$$

其中 M 表示截至当前分支定界过程中，我们对变量 y_j 曾经进行过 M 次分支。

最后，针对当前分支定界局面下 $\mathbf{y}_k$ 的所有可分支非整数元素 y_j，当前需要用如下的打分函数进行评估

$$\begin{aligned} &\text{score}\left(\eta_j^-\left[y_j - \lfloor y_j \rfloor\right], \eta_j^+\lceil y_j \rceil - y_j\right) \\ &= \lambda \min\left(\eta_j^-\left[y_j - \lfloor y_j \rfloor\right], \eta_j^+\lceil y_j \rceil - y_j\right) \\ &+ (1-\lambda)\max\left(\eta_j^-\left[y_j - \lfloor y_j \rfloor\right], \eta_j^+\lceil y_j \rceil - y_j\right) \end{aligned} \tag{4.23}$$

第三种方式是混合分支（Hybrid Branching）：将上述两种分支方法结合，达到却长补短的效果。例如热启动混合分支的做法是在分支定界初期做强分支，然后切换到伪打分分支。因为在搜索初期没有多少历史信息来做估计，会导致伪打分分支效果变差。我们就分支定界初期采用强分支收集信息，当分支定界树的深度达到一定深度之后，切换到伪打分分支。这样就可以保证保证伪打分分支的准确性，同时又可以有效地降低计算量。

例 4.11. 用分支定界法求解例题 4.7。

解：

4.6　Benders 分解法

Benders 分解法（Benders' Decomposition Algorithm）是一种行生成算法（Row Generation Algorithm）[21]-[23]。在使用 Benders 分解法求解复杂问题的过程中，我们先解简化后的问题，然后基于当前得到的解迭代添加线性约束并再次求解，也即逐渐增加约束条件矩阵的行数，直到求解原问题的解。

我们先考虑如下的线性规划问题

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & z = \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{F}\mathbf{y} \geq \mathbf{b} \\ & \mathbf{x}, \mathbf{y} \geq \mathbf{0} \end{aligned} \tag{4.24}$$

其中向量 $\mathbf{x} \in \mathbb{R}^n$，$\mathbf{y} \in \mathbb{Z}^k$，$\mathbf{b} \in \mathbb{R}^m$，$\mathbf{A} \in \mathbb{R}^{m\times n}$，$\mathbf{F} \in \mathbb{R}^{m\times k}$，$\mathbf{c} \in \mathbb{R}^n$，$\mathbf{d} \in \mathbb{R}^k$。

显然，我们可以将上述线性规划问题（4.24）等价改写为

$$\min_{\mathbf{y}\geq\mathbf{0}} \left[\mathbf{d}^T\mathbf{y} + \min_{\mathbf{x}\geq\mathbf{0}} \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{F}\mathbf{y} \right\} \right] \tag{4.25}$$

可以看到优化问题（4.25）由内层优化问题 $\min_{\mathbf{x}\geq\mathbf{0}} \left\{ \mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{F}\mathbf{y} \right\}$ 和外层优化问题共同组成。如果我们固定选择 $\hat{\mathbf{y}} \geq \mathbf{0}$，则上述问题（4.25）退化为

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\hat{\mathbf{y}} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{F}\hat{\mathbf{y}} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{4.26}$$

显然线性规划问题（4.26）的解取决于 $\hat{\mathbf{y}}$ 的选择，此时可能遇到三种情况：

第一种情况，线性规划问题（4.26）无解，则下述线性规划问题同样无解

$$\begin{aligned} \min_{\mathbf{x}} \quad & z = \mathbf{0}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{F}\hat{\mathbf{y}} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{4.27}$$

而根据弱对偶性定理，该问题的对偶问题或者无解或者存在无界的解

$$\begin{aligned} \max_{\mathbf{u}} \quad & \left[\mathbf{b}-\mathbf{F}\hat{\mathbf{y}}\right]^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u} \le \mathbf{0} \\ & \mathbf{u} \ge \mathbf{0} \end{aligned} \tag{4.28}$$

其中对偶变量 $\mathbf{u} \in \mathbb{R}^m$。

注意到 $\mathbf{u}=\mathbf{0}$ 显然是线性规划问题（4.28）的一个平凡解，则线性规划问题（4.28）存在无界的解。

第二种情况，线性规划问题（4.27）存在无界的解，根据定理 2.14，线性规划问题（4.27）的可行域中存在一条极线 $\mathbf{x_d}$ 满足 $\mathbf{c}^T\mathbf{x_d} < 0$。因为 $\begin{bmatrix}\mathbf{x_d} & \hat{\mathbf{y}}\end{bmatrix}$ 是线性规划问题（4.24）的可行解，则线性规划问题（4.24）存在无界的解。

第二种情况，线性规划问题（4.27）存在有界的最优解，根据强对偶性定理，我们可以将优化问题（4.25）等价改写为

$$\min_{\mathbf{y}\ge\mathbf{0}} \left[\mathbf{d}^T\mathbf{y} + \max_{\mathbf{u}\ge\mathbf{0}} \left\{ \left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T \mathbf{u} \mid \mathbf{A}^T\mathbf{u} \le \mathbf{c} \right\} \right] \tag{4.29}$$

我们更偏爱求解优化问题（4.29），因为其内层优化问题的约束条件 $\mathbf{A}^T\mathbf{u} \le \mathbf{c}$ 跟 $\mathbf{y}$ 无关。考虑到这点，Benders 分解法针对线性规划问题（4.24）设计了如下的双层优化算法：

算法 4.4. Benders 分解法

Step 1. 设置初始可行解 $\mathbf{y}^{(1)}$，初始的上界 $\text{UpperBound}^{(0)} = +\infty$，初始的下界 $\text{LowerBound}^{(0)} = -\infty$。令 $k := 1$，进入 Step 2。

Step 2. 求解如下的子问题（Sub Problem）

$$\begin{aligned} \max_{\mathbf{u}} \quad & z_{\text{sub}}^{(k)} = \mathbf{d}^T\mathbf{y}^{(k)} + \left[\mathbf{b}-\mathbf{F}\mathbf{y}^{(k)}\right]^T \mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u} \le \mathbf{c} \\ & \mathbf{u} \ge \mathbf{0} \end{aligned} \tag{4.30}$$

如果子问题（4.30）存在有界的最优解，则最优解必然在多面体可行域 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \le \mathbf{c}, \mathbf{u} \ge \mathbf{0}\right\}$ 的一个极点 $\mathbf{u}_i^e$ 处取得，我们将 $\mathbf{d}^T\mathbf{y} + \left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T \mathbf{u}_i^e \le z$ 作为最优性切平面条件加入 Step 3 中描述的受限主问题。根据最优目标函数值 $z_{\text{sub}}^{(k)}$，更新原问题最优解的上界估计 $\text{UpperBound}^{(k)} = \min\left\{z_{\text{sub}}^{(k)}, \text{UpperBound}^{(k-1)}\right\}$。进入 Step 3。

如果子问题（4.30）存在无界的解，找出没有用过的一条使得子问题（4.30）无界的多面体可行域 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \le \mathbf{c}, \mathbf{u} \ge \mathbf{0}\right\}$ 的极线 $\mathbf{u}_i^r$，将 $\left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T \mathbf{u}_j^r \le 0$ 作为可行性切平面条件加入 Step 3 中描述的受限主问题（因为使得子问题（4.30）无界的极线可能不止一条）。进入 Step 3。

如果子问题（4.30）无解，则原问题要么无解，要么存在无界的解，停止。

Step 3. 求解如下的受限主问题（Restricted Master Problem）

$$\begin{aligned}\min_{\mathbf{y},z}\quad & z\\ \text{s.t.}\quad & \mathbf{d}^T\mathbf{y}+\left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T\mathbf{u}_i^e\le z, i=1,...,m\\ & \left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T\mathbf{u}_j^r\le 0, j=1,...,k\\ & \mathbf{y}\ge\mathbf{0}\end{aligned}\tag{4.31}$$

其中 $\mathbf{u}_i^e, i=1,...,m$ 是求解子问题时，加入的多面体可行域 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\le\mathbf{c},\mathbf{u}\ge\mathbf{0}\right\}$ 的不同极点，加入后使得受限主问题尽量存在有界的解。$\mathbf{u}_j^r, j=1,...,k$ 是求解子问题时，加入的多面体可行域 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\le\mathbf{c},\mathbf{u}\ge\mathbf{0}\right\}$ 的不同极线，加入后使得受限主问题尽量存在可行解。

如果受限主问题（4.31）存在有界的最优解 $\left[\mathbf{y}_{\text{master}}\quad \mathrm{z}_{\text{master}}^{(k)}\right]$，则更新原问题最优解的上界估计 $\text{LowerBound}^{(k)}=\max\left\{\mathrm{z}_{\text{master}}^{(k)},\text{LowerBound}^{(k-1)}\right\}$。进入 Step 4。

如果受限主问题（4.31）无解，则原问题亦无解，停止。

如果受限主问题（4.31）在第 k 轮迭代的 Step 2 增加了新的切平面条件，但存在无界的解，则跳过，直接进入 Step 2。

如果受限主问题（4.31）在第 k 轮迭代的 Step 2 没有增加新的切平面条件，依然存在无界的解，则原问题存在无界的解，停止。

Step 4. 如果 $\text{LowerBound}^{(k)}=\text{UpperBound}^{(k)}$，已经找到原问题最优解，停止。

否则，令 $\mathbf{y}^{(k+1)}=\mathbf{y}_{\text{master}}$，$k:=k+1$，进入 Step 2。

定理 4.3. Benders 分解法可以正确求解线性规划问题（4.24）。

证明。显然我们需要分三种情况讨论。

第一种情况，线性规划问题（4.24）存在有界的最优解。则子问题（4.30）不可能存在无界的解。因为对于任意可行 $\mathbf{y}^{(k)}$，及多面体可行域 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\le\mathbf{c},\mathbf{u}\ge\mathbf{0}\right\}$ 的极线 $\mathbf{u}_j^r, j=1,...,k$，都应该满足受限主问题的约束条件 $\left[\mathbf{b}-\mathbf{F}\mathbf{y}^{(k)}\right]^T\mathbf{u}_j^r\le 0$，则沿着多面体可行域 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\le\mathbf{c},\mathbf{u}\ge\mathbf{0}\right\}$ 的任意一条极线 $\mathbf{u}_j^r$，都会有

$$\mathbf{d}^T\mathbf{y}^{(k)}+\left[\mathbf{b}-\mathbf{F}\mathbf{y}^{(k)}\right]^T\mathbf{u}_j^r\le\mathbf{d}^T\mathbf{y}^{(k)}$$

而多面体可行域 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\le\mathbf{c},\mathbf{u}\ge\mathbf{0}\right\}$ 的所有极点也都是有界的，则根据定理 2.15 所述弱化的 Minkowski 表示定理，$\mathbf{d}^T\mathbf{y}^{(k)}+\left[\mathbf{b}-\mathbf{F}\mathbf{y}^{(k)}\right]^T\mathbf{u}$ 必然有界。

而且，线性规划问题（4.24）存在有界的最优解，则子问题（4.30）必然有解。否则根据弱对偶性，线性规划问题（4.26）会存在无界的解，导致矛盾。

进一步，根据强对偶性定理，我们可以得知原问题的最优解可以写为

$$z^* = \min_{\mathbf{y}\geq\mathbf{0}}\left[\mathbf{d}^T\mathbf{y} + \min_{\mathbf{x}\geq\mathbf{0}}\left\{\mathbf{c}^T\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{F}\mathbf{y}\right\}\right]$$

$$= \min_{\mathbf{y}\geq\mathbf{0}}\left[\mathbf{d}^T\mathbf{y} + \max_{\mathbf{u}\geq\mathbf{0}}\left\{\left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}\right\}\right]$$

而且，根据定理 2.5，子问题（4.29）的最优解必然在某个极点处取得，因此

$$\max_{\mathbf{u}\geq\mathbf{0}}\left\{\left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}\right\} = \max\left\{\left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u}_i^e, i = 1,...,m\right\}$$

进一步，受限主问题可以改写为

$$z^* = \min_{\mathbf{y}\geq\mathbf{0}, \left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T\mathbf{u}_j^r \leq 0,\, j=1,...,k}\left[\mathbf{d}^T\mathbf{y} + \max\left\{\left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u}_i^e, i = 1,...,m\right\}\right]$$

$$= \min_{\mathbf{y}\geq\mathbf{0}, \left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T\mathbf{u}_j^r \leq 0,\, j=1,...,k}\left[z \mid \mathbf{d}^T\mathbf{y} + \left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u}_i^e \leq z, i = 1,...,m\right]$$

此时可以看出，如果线性规划问题（4.24）存在有界的最优解。最差情况下，在我们将多面体可行域 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 的所有极点和极线都加入受限主问题之后，一定会有 $\text{LowerBound}^{(k)} = \text{UpperBound}^{(k)}$。一般而言，无需加入所有极点和极线，就会有 $\text{LowerBound}^{(k)} = \text{UpperBound}^{(k)}$，此时 Benders 分解法满足 Step 4 的停止条件，找到解就是该最优解。

第二种情况，线性规划问题（4.24）无解。则对于任意可行 $\hat{\mathbf{y}} \geq \mathbf{0}$，线性规划问题（4.26）都必须无解；而对偶问题（4.28）要么无解，要么存在无界的解。

如果对偶问题（4.28）存在无界的解，则根据定理 2.14，存在多面体可行域 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 的一条极线 $\mathbf{u}_j^r$ 使得 $\left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T\mathbf{u}_j^r > 0$，但这与受限主问题的约束条件矛盾，因此，对于任意可行 $\hat{\mathbf{y}} \geq \mathbf{0}$，子问题也都无解。因此，满足 Benders 分解法 Step 2 的停止条件，确认原问题无解。

第三种情况，线性规划问题（4.24）存在无界的解。则如下线性规划问题必然无解

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{b}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u} \leq \mathbf{c} \\ & \mathbf{F}^T\mathbf{u} \leq \mathbf{d} \\ & \mathbf{u} \geq \mathbf{0} \end{aligned} \tag{4.32}$$

则多面体 $\left\{\mathbf{u} \mid \mathbf{F}^T\mathbf{u} \leq \mathbf{d}\right\}$ 和多面体 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 的交集为空集，此时又有两种细分情况。

细分情况（I），多面体 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 为空集，则受限主问题始终不能添加任何切平面约束条件，因此最终只会获得无界的解。因此，原问题要么无解，要么存在无界的解，满足 Benders 分解法 Step 2 的停止条件。

细分情况（II），如果多面体 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 至少存在一个极点，则根据定理 2.15 所述弱化的 Minkowski 表示定理，多面体 $\left\{\mathbf{u} \mid \mathbf{A}^T\mathbf{u} \leq \mathbf{c}, \mathbf{u} \geq \mathbf{0}\right\}$ 里的任意一点 $\mathbf{u}$ 可以表示为

$$\mathbf{u}=\sum_{i=1}^{k}\lambda_i\mathbf{u}_i^e+\sum_{j=1}^{r}\theta_j\mathbf{u}_j^r$$

其中 $\lambda_i\geq 0$，$i=1,...,m$，$\theta_j\geq 0$，$j=1,...,k$，$\sum_{i=1}^{m}\lambda_i=1$。

因此，根据线性规划问题（4.32）无解，下面的线性规划问题亦无解

$$\begin{aligned}\max_{\boldsymbol{\lambda},\boldsymbol{\theta}}\quad & \mathbf{b}^T\left[\sum_{i=1}^{k}\lambda_i\mathbf{u}_i^e+\sum_{j=1}^{r}\theta_j\mathbf{u}_j^r\right]\\ \text{s.t.}\quad & \mathbf{F}^T\left[\sum_{i=1}^{k}\lambda_i\mathbf{u}_i^e+\sum_{j=1}^{r}\theta_j\mathbf{u}_j^r\right]\leq\mathbf{d}\\ & \sum_{i=1}^{m}\lambda_i=1\\ & \lambda_i\geq 0,i=1,...,m,\theta_j\geq 0,\ j=1,...,k\end{aligned}\tag{4.33}$$

容易验证，线性规划问题（4.33）的对偶问题就是受限主问题（4.31）。因此，此时受限主问题（4.31）要么无解，要么存在无界的解。

最后，我们用反证法证明，如果线性规划问题（4.24）存在无界的解，则当所有切平面都加入后，受限主问题（4.31）必然存在无界的解。

因为我们适当选择 z，总可以使得 $\mathbf{d}^T\mathbf{y}^{(k)}+\left[\mathbf{b}-\mathbf{F}\mathbf{y}^{(k)}\right]^T\mathbf{u}_j^r\leq z$ 满足。所以受限主问题（4.31）无解，必须对于任意可行 $\mathbf{y}^{(k)}$，多面体 $\left\{\mathbf{u}\mid\mathbf{A}^T\mathbf{u}\leq\mathbf{c},\mathbf{u}\geq\mathbf{0}\right\}$ 的极线 $\mathbf{u}_j^r,j=1,...,k$，都满足 $\left[\mathbf{b}-\mathbf{F}\mathbf{y}\right]^T\mathbf{u}_j^r>0$。则线性规划问题（4.28）一定存在无界的界。根据弱对偶性原理，线性规划问题（4.27）一定无解；进而对于任意可行 $\hat{\mathbf{y}}$，线性规划问题（4.26）一定无解。这和线性规划问题（4.24）存在无界的解矛盾。

因此，如果线性规划问题（4.24）存在无界的解，则当所有切平面都加入后，受限主问题（4.31）必然存在无界的解，满足 Benders 分解法 Step 2 的停止条件。□

比较受限主问题（4.31）和原始问题（4.24），我们可以发现 Benders 分解法确实应用了切割平面法。每次求解受限主问题时，我们都会更新下界估计；每次求解子问题时，我们都会更新上界估计。由于极点和极线的数量都是有限的，所以我们总是会在添加完所有约束条件后最终停止。

注意到多面体可行域的极点和极线的数量是指数级的。Bender 分解法采取依次添加约束的方法进行求解。我们经常可以在不将所有这些约束添加到受限主问题中的情况下获得最终最优解。

Benders 分解法的收敛速度可能较慢。但是，Benders 分解法同时提供了问题的上下界估计，这可能会在很多应用中对我们有所帮助。

接下来我们考虑混合整数线性规划问题（4.2）。显然，Benders 分解法只需要将受限主问题（4.31）改写为

$$\begin{aligned} \min_{\mathbf{y},z} \quad & z \\ \text{s.t.} \quad & \mathbf{d}^T\mathbf{y} + \left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T \mathbf{u}_i^e \le z, i = 1,...,m \\ & \left[\mathbf{b} - \mathbf{F}\mathbf{y}\right]^T \mathbf{u}_j^r \le 0, j = 1,...,k \\ & \mathbf{y} \in \mathbb{Z} \end{aligned} \tag{4.31}$$

即可继续使用。此时，子问题为纯粹的线性规划问题，容易求解。而受限主问题为纯整数规划问题，可以使用 Gomory 切平面法或分支定界法求解。因此，Benders 分解法将显著简化问题的难度。

例 4.12. 用 Benders 分解法求解如下的混合整数线性规划问题

$$\begin{aligned} \min_{x,y} \quad & z_0 = 2x + 3y \\ \text{s.t.} \quad & x + 3y \ge 6 \\ & 4x - y \ge 0 \\ & x - 2y \ge -4 \\ & x \ge 0 \\ & y \in \{0,...,6\} \end{aligned}$$

解：首先选定初始化基本可行解 $\hat{y}^{(0)} = 0$，初始的上界 $\text{UpperBound}^{(0)} = +\infty$，初始的下界 $\text{LowerBound}^{(0)} = -\infty$。

第一轮迭代：子问题为

$$\begin{aligned} \max_{\mathbf{u}} \quad & z_1 = (6 - 3y)u_1 + yu_2 + (-4 + 2y)u_3 + 3y \\ \text{s.t.} \quad & u_1 + 4u_2 + u_3 \le 2 \\ & u_1, u_2, u_3 \ge 0 \end{aligned}$$

带入 $\hat{y}^{(0)} = 0$ 整理可得

$$\begin{aligned} \max_{\mathbf{u}} \quad & z_1 = 6u_1 - 4u_3 \\ \text{s.t.} \quad & u_1 + 4u_2 + u_3 \le 2 \\ & u_1, u_2, u_3 \ge 0 \end{aligned}$$

解得最优解为 $\hat{\mathbf{u}}^{(1)}=[2,0,0]^T$，是子问题的一个极点。对应的最优目标函数值为 $z_{\text{sub}}^{(1)}=12$，因此 $\text{UpperBound}^{(1)}=\min\{12,+\infty\}=12$。我们增加一个最优性切平面约束条件，将受限原问题写为

$$\begin{aligned}\min_{y}\quad & z_1\\ \text{s.t.}\quad & z_1\geq -3y+12\\ & y\in\{0,...,6\}\end{aligned}$$

这个受限原问题的最优解显然为 $y^{(1)}=6$，对应的最优目标函数值为 $z_{\text{master}}^{(1)}=-6$，因此 $\text{LowerBound}^{(1)}=\max\{-6,-\infty\}=-6$。

因为 $\text{LowerBound}^{(1)}\neq\text{UpperBound}^{(1)}$，还需要继续迭代。

第二轮迭代：子问题为

$$\begin{aligned}\max_{\mathbf{u}}\quad & z_2=-12u_1+6u_2+8u_3+18\\ \text{s.t.}\quad & u_1+4u_2+u_3\leq 2\\ & u_1,u_2,u_3\geq 0\end{aligned}$$

解得最优解为 $\hat{\mathbf{u}}^{(2)}=[0,0,2]^T$，也是子问题的一个极点。对应的最优目标函数值为 $z_{\text{sub}}^{(2)}=34$，因此 $\text{UpperBound}^{(2)}=\min\{34,12\}=12$。我们增加一个最优性切平面约束条件，将受限原问题写为

$$\begin{aligned}\min_{y}\quad & z_2\\ \text{s.t.}\quad & z_2\geq -3y+12\\ & z_2\geq 7y-8\\ & y\in\{0,...,6\}\end{aligned}$$

这个受限原问题的最优解显然为 $y^{(2)}=2$，对应的最优目标函数值为 $z_{\text{master}}^{(2)}=6$，因此 $\text{LowerBound}^{(2)}=\max\{2,-6\}=2$。

因为 $\text{LowerBound}^{(2)}\neq\text{UpperBound}^{(2)}$，还需要继续迭代。

第三轮迭代：子问题为

$$\begin{aligned}\max_{\mathbf{u}}\quad & z_3=2u_2+6\\ \text{s.t.}\quad & u_1+4u_2+u_3\leq 2\\ & u_1,u_2,u_3\geq 0\end{aligned}$$

解得最优解为 $\hat{\mathbf{u}}^{(3)}=[0,0.5,0]^T$，也是子问题的一个极点。对应的最优目标函数值为 $z_{\text{sub}}^{(3)}=7$，因此 $\text{UpperBound}^{(3)}=\min\{7,12\}=7$。我们增加一个最优性切平面约束条件，将受限原问题写为

$$\begin{aligned}\min_{y} \quad & z_3 \\ \text{s.t.} \quad & z_3 \geq -3y+12 \\ & z_3 \geq 7y-8 \\ & z_3 \geq 3.5y \\ & y \in \{0,...,6\}\end{aligned}$$

这个受限原问题的最优解显然为 $y^{(3)}=2$，对应的最优目标函数值为 $z_{\text{master}}^{(3)}=7$，因此 $\text{LowerBound}^{(3)}=\max\{7,2\}=7$。

因为 $\text{LowerBound}^{(3)}=\text{UpperBound}^{(3)}$，停止。原问题的最优解为 $x=0.5$，$y=2$，最优目标函数值为 $z=7$。

4.7 作业题

4.1 切平面法有时会花费非常多的步骤去求解一些不那么复杂的问题。请用切平面法求解如下的整数线性规划问题

$$\begin{aligned}\max_{y_1,y_2} \quad & y_1+5y_2 \\ \text{s.t.} \quad & -\frac{5}{8}y_1+y_2 \leq 1, y_1+\frac{8}{5}y_2 \leq 8 \\ & y_1,y_2 \geq 0, y_1,y_2 \in \mathbb{Z}\end{aligned}$$

4.2 请用分支定界法求解作业题 4.1 中的整数线性规划问题。

4.8 参考文献

[1] M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, L. A. Wolsey, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, Springer, 2010.

[2] L. A. Wolsey, *Integer Programming*, 2nd edition, John Wiley & Sons, 2021.

[3] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computation* , R. E. Miller, J. W. Thacher, eds., Plenum Press, pp. 85 103, 1972.

[4] S. Burer, A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97-106, 2012.

[5] F. Trespalacios, I. E. Grossmann, "Review of mixed‐integer nonlinear and generalized disjunctive programming methods," *Chemie Ingenieur Technik*, vol. 86, no. 7, pp. 991-1012, 2014.

[6] F. Boukouvala, R. Misener, C. A. Floudas, "Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO," *European Journal of Operational Research*, vol. 252, no. 3, 1, pp. 701-727, 2016.

[7] M. Locatelli, F. Schoen, "(Global) Optimization: Historical notes and recent developments," *EURO Journal on Computational Optimization*, vol. 9, 100012., 2021.

[8] S. Martello, P. Toth, *Knapsack Problems Algorithms and Computer Implementations*, John Wiley & Sons, 1990.

[9] E. Miller, W. Tucker, A. Zemlin, "Interger programming formulation of traveling salesman problems," *Journal of the ACM*, vol. 7, no. 4, pp. 326-329, 1960.

[10] Jr. J. E. Kelley, "The cutting-plane method for solving convex programs," *Journal of Society of Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703-712, 1960.

[11] A. F. Veinott Jr., "The supporting hyperplane method for unimodal programming," *Operations Research*, vol. 15, no. 1, pp. 147-152, 1967.

[12] B. C. Eaves, W. I. Zangwill, "Generalized cutting plane algorithms," *SIAM Journal on Control*, vol. 9, no. 4, pp. 529-542, 1971.

[13] J. Gondzio, O. du Merle, R. Sarkissian, J.-R. Vial, "ACCPM A library for convex optimization based on an analytic center cutting plane method," *European Journal of Operational Research*, vol. 94, no. 1, pp. 206-211, 1996.

[14] Y. Ye, "Complexity analysis of the analytic center cutting plane method that uses multiple cuts," *Mathematical Programming*, vol. 78, no. 1, pp. 85-104, 1997.

[15] J. E. Mitchell, "Cutting plane methods and subgradient methods," *INFORMS Tutorials in Operations Research*, pp. 34-61, 2017.

[16] R. E. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, pp. 5, pp. 275-278, 1958.

[17] A. H. Land, A. G. Doig, "An automatic method for solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497-520, 1960.

[18] J. D. C. Little, K. G. Murty, D. W. Sweeney, C. Karel, "An algorithm for the traveling salesman problem," *Operations Research*, vol. 11, no. 6, pp. 972-989, 1963.

[19] R. J. Dakin, "A tree search algorithm for mixed integer programming problems," *Computer Journal*, vol. 8, pp. 250-255, 1965.

[20] T. Achterberg, T. Koch, A. Martin, "Branching rules revisited," *Operations Research Letters*, vol. 33, no. 1, pp. 42-54, 2005.

[21] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 3, pp. 238-252, 1962.

[22] A. M. Geoffrion, "Generalized Benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237-260, 1972.

[23] R. K. Martin, *Large Scale Linear and Integer Optimisation: A Unified Approach*, Springer, 1998.

第 5 章 图和网络规划

5.1 引言

很多优化问题与图（Graph）和网络（Network）联系在一起。20 世纪 50 年代以来，Bellman 等研究者提出了最短路算法、最大网络流算法等经典算法，在经济、社会、工业生产、军事等重要领域取得了卓有成效的应用 [1]-[3]。

本章先介绍图和网络的基本概念。接着依次介绍解决最小支撑树问题、最短路问题、最大流问题、最大流和最小费用流问题、运输问题、指派问题中的典型算法。最后介绍了幺模矩阵的概念，并进一步分析了一些网络问题的求解方式。

5.2 图和网络基础

5.2.1 图和网络的基本定义

定义 5.1. 图（Graph）由有穷非空的顶点（Vertex）集合和顶点之间边（Edge）的集合构成，表示为 $G=(V,E)$，$V=\{v_1,...,v_n\}$ 是顶点的集合，$E=\{e_1,...,e_m\}$ 是边的集合。也有的研究者用点（Node）来指代顶点。

定义 5.2. 若顶点 v_i 到顶点 v_j 之间的边没有方向，则这条边为无向边；反之，则为有向边。有的研究者用 (v_i,v_j) 或 (v_j,v_i) 表示无向边，用 $\langle v_i,v_j\rangle$ 表示有向边。若图中所有边都为无向边，则该图为无向图。反之，若都为有向边，则为有向图（Directed Graph，或简写为 Digraph）。有的研究者特别用边（Edge）来指代无向边；用弧（Arc）来指代有向边。其中 v_i 称为弧尾，v_j 称为弧头。本书后续还是采用顶点和边这一描述方式。统一以 (v_i,v_j) 表示边，对于有向图称 v_i 为起点，v_j 为终点。

定义 5.3. 如果顶点 $v_1,v_2\in V$，边 $(v_1,v_2)\in E$，则称 v_1,v_2 相邻（Adjacent），

v_1, v_2 称为 (v_i, v_j) 的端点。如果边 $e_1, e_2 \in E$，并且有公共端点 $v \in V$，则称 e_1, e_2 相邻，称 e_1, e_2 为 v 的关联边。

定义 5.4. 对于图 $G = (V, E)$，我们分别用 $|E|$ 和 $|V|$ 来表示边数和顶点数。

定义 5.5. 对于图 $G = (V, E)$，如果边 e_1 连接的两个端点相同，则称 e_1 为自回路（Self-Loop，或 Cycle）。

定义 5.5. 如果图的两个顶点之间有多于一条的不同的边，则称这些边为多重边（Parallel Edges）。

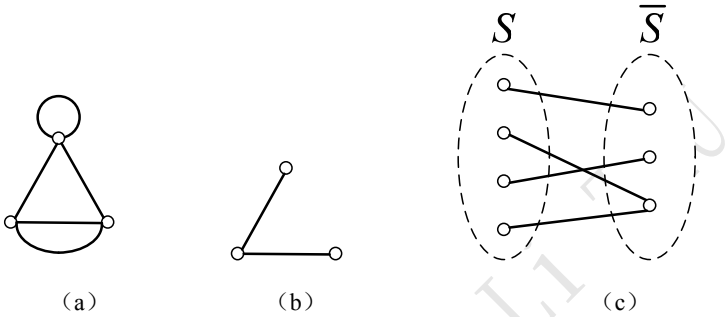


图 5.1 （a）具有环和多重边的无向图示意图；（b）图（a）的一个支撑子图；（c）二分图示意图。

定义 5.7. 不含有自回路和多重边的图称为简单图（Simple Graph）；含有多重边的图称为多重图（Multigraph）。

定义 5.8. 任意两个顶点间都有边相连的无向简单图称为完全图（Complete Graph）；任意两个顶点之间都有且仅有一条边相连的有向简单图称为有向完全图（Complete Digraph）。

例 5.1. 假设一个无向完全图有 n 个顶点，请问该图有多少条边？假设一个有向完全图有 n 个顶点，请问该图有多少条边？

解：显然，一个有着 n 个顶点的无向完全图有 $\frac{n(n-1)}{2}$ 条边。一个有着 n 个顶点的有向完全图有 $n(n-1)$ 条边。

定义 5.9. 对于图 $G = (V, E)$ 如果 V 可以划分为两个不相交的子集 $S, \overline{S}$，使

得 E 中每条边的两个端点必有一个属于 S，一个属于 $\overline{S}$，称 $G=(V,E)$ 为二分图（Bipartite Graph），记为 $G(L,R,E)$。

定义 5.10. 以顶点 v 为端点的边数称为 v 的度（Degree）或者次，记为 $\deg(v)$，或简记为 $d(v)$。度为 0 的顶点称为孤立点，度为 1 的顶点称为悬挂点，连接悬挂点的边成为悬挂边，度为奇数的顶点称为奇点，度为偶数的顶点称为偶点。

定义 5.11. 在有向图中，以顶点 v 为起点的边数称为顶点 v 的出度或出次，用 $d^+(v)$ 表示，以顶点 v 为终点的边数称为顶点 v 的入度或入次，用 $d^-(v)$ 表示。

定理 5.1. 任何图中，顶点度数总和等于边数的 2 倍。

证明：因为每条边会在顶点的出度和入度各计算一遍，因此结论成立。□

定理 5.2. 任何图中，奇点的个数为偶数个。

证明：根据定理 1，我们可知：奇点度数综合 + 偶点度数总和 = 偶数。其中，偶点度数总和为偶数，若有奇数个奇点，其度数总和为奇数，则上式不成立，故原假设不成立，奇点的个数为偶数个。□

例 5.2. Erdos 和 Renyi 提出了随机图模型（Erdos-Renyi Random Graph）[3]，设定对于图中每对顶点，以相等的概率 p 在它们之间连一条边。假设该图有 n 个顶点，请问该图有 m 条边的概率是多少？请进一步算出该图中每个顶点度的分布。

解：考虑例 5.1 的结果，该随机图有 m 条边的概率是 $C_{\frac{n(n-1)}{2}}^{m}p^m(1-p)^{\frac{n(n-1)}{2}-m}$。

进一步，我们可以推出第 i 个顶点的度等于 k 的概率为

$$\Pr\left(\deg\left(v_i\right)=k\right)=C_{n-1}^{k}p^k(1-p)^{n-1-k}$$

对于很大 n，这一概率近似满足 Poisson 分布

$$\Pr\left(\deg\left(v_i\right)=k\right)\approx e^{-pn}\frac{(pn)^k}{k!}$$

例 5.3. Barabási 和 Albert 提出了无标度网络（Scal-Free Network）[4]，要求按如下方式建立网络：

初始过程：初始网络是空集；

增长过程：每个时刻加入一个新顶点，它有 m 条边与现有顶点连接，允许自环；

优先连接过程：新顶点与顶点 i 相连的概率取决于顶点 i 的度

$$\Pr\left(\text{connect } v_i\right)=\frac{\deg\left(v_i\right)}{\sum_i \deg\left(v_i\right)}$$

重复上述增长和有限连接过程，可以发现虽然大多数顶点只和少数几个其它顶点连接，但有一些顶点逐渐变成网络的中心点（Hub）。出现富者更富（Rich-Gets-Richer）现象，即新顶点偏向连接度数高的顶点，而度数高的顶点会有更多新顶点连接。请计算足够长的时间之后，该图中每个顶点的度的分布。

解：我们用连续实变量来近似$\deg\left(v_i\right)$增长过程中的期望值。显然顶点i获取新顶点连接的速率为

$$\frac{d\deg\left(v_i\right)}{dt}=m\frac{\deg\left(v_i\right)}{\sum_i \deg\left(v_i\right)}$$

在t个时刻之后，网络中一共有t个顶点，mt条边。而所有顶点的总度值等于边的数目的两倍，所以$\sum_i \deg\left(v_i\right)=2mt$。所以，上式可以变为

$$\frac{d\deg\left(v_i\right)}{\deg\left(v_i\right)}=\frac{1}{2}\frac{dt}{t}$$

由于每个顶点是逐一加入到网络中的，在顶点i加入的t_i时刻，有$\deg\left(v_i\right)=m$。带入上述常微分方程，我们可以推出

$$\deg\left(v_i\right)=m\left(\frac{t}{t_i}\right)^{\frac{1}{2}}$$

进一步将所有顶点的度值组成一个度序列（Degree Sequence），并统计不同顶点度值出现的频率就可以得到度分布（Degree Distribution），Barabási 和 Albert 发现不同网络的度分布在 log-log 坐标系下均呈现近似线性的排布，即 Power-Law 现象。

现实生活中有很多网络大致符合这一度分布规律。例如，社交网络中的少量大 V 用户拥有极其多的粉丝关注，而大多数用户则只有少数人关注。

定义 5.12. 对于图$G=\left(V,E\right)$，如果E'是E的子集，V'是V的子集，并且E'中每条边的端点都属于V'，则称$G'=\left(V',E'\right)$是G的子图（Subgraph），如果$V'=V$，则称G'是G的支撑子图（Supporting Subgraph）。

定义 5.13. 每边赋有权（实数或实数向量）的图称为网络（Network），记为$G=\left(V,E,W\right)$，其中$W=\{w(e),e\in E\}$是权的集合，无向图赋权构成无向网络，有向图赋权构成有向网络。

图是高度抽象后的用邻接矩阵所包含的结构信息，而网络（Network）是建立在图的拓扑结构之上，通过在边和顶点上增加新的属性来包含更多的信息。

5.2.2　图和网络的矩阵描述和连通性

定义 5.14. 一个图中可重复的顶点、边交替的序列称为路或链（Walk）。无重复边的路称为简单路，无重复顶点的路称为初级路（Path）。至少包含一条边，且起点和终点为同一顶点的路称为回路（Closed Walk，或 Circuit)）。如果一条回路中没有重复的边，则成为简单回路。

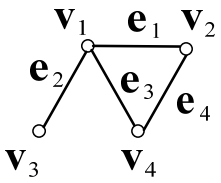


图 5.2 $v_1 \to e_1 \to v_2 \to e_4 \to v_4 \to e_3 \to v_1 \to e_2 \to v_3$ 为一条简单路。$v_1 \to e_1 \to v_2 \to e_4 \to v_4$ 为一条初级路。$v_1 \to e_1 \to v_2 \to e_4 \to v_4 \to e_3 \to v_1$ 为一条简单回路路。e_2 为该图的一条割边。$\{e_2, e_3, e_4\}$ 为该图的一个边割。$\{e_1, e_4\}$ 为该图的一个割集。

定义 5.15. 一个图中两顶点间若存在连接两点的一条路，称这两顶点连通（Connected）。任意两顶点都连通的图称为连通图（Connected Graph）。一个不连通的图可以含有多个连通子图，称为连通分支。

定义 5.15. 如果删去一个图中某条边，会使得图的连通分支数严格增加，则称该边为图的割边（Cut Edge）。

定义 5.17. 对于图 $G=(V,E)$，任取 $S \subset V$，其补集为 $\overline{S}=V-S$，若 S 和 $\overline{S}$ 都不是空集，称两个端点分属 S 和 $\overline{S}$ 的边的集合为 G 的一个边割（Edge Cut，或者称为 S 的 Coboundary），记为 $\{S,\overline{S}\}$。如果对于一个连通图，除去某个边割后，连通图变为不连通，而除去这个边割的真子集后，连通图仍然连通，则称这个边割为边割集或割集（Cuts）。

定义 5.18. 表示各个顶点和每条边之间的关系的矩阵称为有向图的关联矩阵（Incidence Matrix），描述了有向简单图的拓扑性质。对于图 $G=(V,E)$，用 g_{ij}

表示在关联矩阵中顶点 i 和边 j 之间的关系。若 $g_{ij}=1$，表示边 j 离开顶点 i；若 $g_{ij}=-1$，表示边 j 进入顶点 i；若 $g_{ij}=0$，表示边 j 和顶点 i 不相关联。显然，关联矩阵每列必然有一个元素为 1 和另一个元素为 -1，其余元素为 0。

定义 5.19. 表示顶点之间相邻关系的矩阵称为邻接矩阵（Adjacency Matrix），描述了简单图的拓扑性质。对于图 $G=(V,E)$，用 g_{ij} 表示在关联矩阵中顶点 i 和顶点 j 之间的关系。若 $g_{ij}=k$，表示有 k 条边从顶点 i 到顶点 j；否则 $g_{ij}=0$，即没有边从顶点 i 与顶点 j。显然，对于无向图而言，邻接矩阵为对称矩阵。

例 5.4. 请给出图 5.3（a）中有向图的关联矩阵，图 5.3（b）中无向图的邻接矩阵。

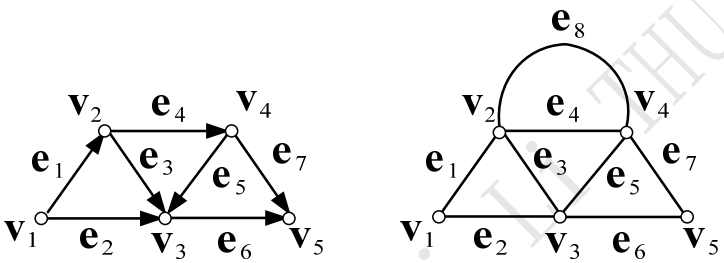


图 5.3 （a）某有向图示意图；（b）某无向图示意图。

解：图 5.3（a）的关联矩阵为

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}\begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{pmatrix} 1 & 1 & & & & & \\ -1 & & 1 & 1 & & & \\ & -1 & -1 & & -1 & 1 & \\ & & & -1 & 1 & & 1 \\ & & & & & -1 & -1 \end{pmatrix} \end{array}$$

图 5.3（b）的邻接矩阵为

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 2 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

基于上述描述，我们可以进一步检验某个给定图中各顶点的连通性。无向图连通性的经典算法包括：深度优先搜索、广度优先搜索等。

算法 5.1. 无向图连通性的深度优先搜索算法

```
输入: 存放图的一维顶点数组 vertice[]，二维邻接矩阵数组 adjacency[][]
输出: 图是否连通的 Bool 值
中间变量: 存放图的顶点是否访问过信息的一维数组 visitedvertice []

function connectivity_DFS(v)
{   visitedvertice [v] = True;   // 标记已经访问过的顶点
    for u = 1 : sizeof(vertice)
    {
        if adjacency[v][u] >= 1 && visitedvertice [u] == False
          {connectivity_DFS(u);   // 递归访问尚未展开的顶点
          }
    }
}

for u = 1 : sizeof(vertice)   // 先将所有顶点设为没有访问过
{   visitedvertice [u] = False;
}

connectivity_DFS(1);     // 从第一个顶点开始深度优先搜索

connectedflag = True;
for u = 1 : sizeof(vertice)   // 深度优先搜索之后，遍历查询访问记录
{
    if visitedvertice [u] == False
      {connectedflag = False;
```

```
        }
}

output connectedflag;   // 输出结果
```

算法 5.2. 无向图连通性的广度优先搜索算法

```
输入: 存放图的一维顶点数组 vertice[]，二维邻接矩阵数组 adjacency[][]
输出: 图是否连通的 Bool 值
中间变量: 存放待访问顶点的队列 tobecheckedqueue

tobecheckedqueue = new queue;   // 建立用于广度优先搜索的队列

for u = 1 : sizeof(vertice)   // 先将所有顶点设为没有访问过
{   visitedvertice [u] = False;
}

function connectivity_BFS(v)
{   tobecheckedqueue.append(v);
    visitedvertice [v] = True;
    while tobecheckedqueue is not empty
    {
       u = tobecheckedqueue.pop();   //
       for u = 1 : sizeof(vertice)
       {
           if adjacency[v][u] >= 1 && visitedvertice [u] == False
           {
               tobecheckedqueue.append (u);   // 将待考察顶点加入 queue
               visitedvertice [u] = True;
           }
     }
   }
}

connectivity_BFS(1);   // 从第一个顶点开始广度优先搜索

connectedflag = True;
for u = 1 : sizeof(vertice)   // 广度优先搜索之后，遍历查询访问记录
```

```
{
    if visitedvertice [u] == False
      {connectedflag = False;
      }
}

output connectedflag;   // 输出结果
```

对于有向图而言，我们可以从图中每一个顶点出发，用上述算法检测是否可以从该顶点连通到图中其它所有顶点。如果从每一个顶点出发都能连通到图中其它所有顶点，则该有向图是完全连通的。

1731 年，Euler 在俄国彼得堡科学院担任物理教授，他接收到一封看似随意的询问信，里面包含着一个在当时广为人知但却令许多人困惑的问题：在普鲁士的哥尼斯堡有一个被称为奈发夫的岛屿；普雷格尔河的两条支流从岛的两旁流过，且有七座桥横跨这两条支流。一个人如何才能一次走遍七座桥，且每座桥只走过一次，最后回到出发点？1735 年 8 月 26 日，Euler 向当时俄国的圣彼得堡科学院递交了一篇名为《有关位置几何的一个问题的解》的论文，否定哥尼斯堡七桥问题能一次走完，并提出了一笔画定理。

定义 5.20. 能起点出发，经过各点一笔画出所有边并回到起点的无向图称为 Euler 图。对应的路称为 Euler 路（Eulerian path）

定理 5.3. [一笔画定理] 凡是由偶点组成的连通图，一定是 Euler 图。画时可以把任一偶点为起点，最后一定能以这个点为终点画完此图。凡是只有两个奇点的连通图（其余都为偶点），一定是 Euler 图。画时必须把一个奇点为起点，另一个奇点终点。其它情况的图都不是 Euler 图。

证明：我们首先用数学归纳法证明第一种情况。假设所考虑的图 G 是偶点组成的连通图。

当边数为时 1，G 必定为单顶点的环，显然此时 G 为 Euler 图。

假设已经证明边数少于 m 的连通图，在所有顶点的度均为偶数时必为 Euler 图。进一步考虑有 m 条边的图 G。设想从 G 的任一顶点出发，沿着边构画，使笔不离开图且不在构画过的边上重新构画。由于每个顶点都是偶数度，笔在进入一个顶点后总能离开那个顶点，除非笔回到了起点。在笔回到起点时，它构画出一条环路，记为 H。

如果环路 H 覆盖了所有 G 的边，则环路 H 就是我们需要寻找的 Euler 路。否则，我们可以从图 G 中删去环路 H 的所有边，所得图记为 G'。G' 未必连通，但其各顶点的度数仍均为偶数。考虑 G' 的各连通分支，由于它们都连通，且其中顶

点的度数均为偶数，而边数均小于m。根据归纳假设，它们都是 Euler 图。此外，由于 G 连通，它们都与H共有一个或若干个公共顶点。因此，它们与 H 一起，可以构成一个 Euler 路。

根据数学归纳法，我们就证明了 G 是一个 Euler 图。

对于第二种情况。假设所考虑的图G是只有两个奇点的连通图。我们可以增加一条边将这两个奇点连起来，则划归成第一种情况，可以找到一条 Euler 路。从这条 Euler 路中删去我们增加的边，则依然有一条从其中一个奇点为起点，另一个奇点为终点的 Euler 路。因此，满足第二种情况的图是 Euler 图。

对于第三种情况，假设存在 Euler 路H，由于H每经过一个非起点或终点的顶点一次，便给这一顶点增加度2，因而各顶点的度均为该路经历此顶点的次数的两倍。这导致与假设矛盾。因此，第三种情况的图都不是 Euler 图。□

5.3 树

5.3.1 树和支撑树

定义 5.21. 设无向图$T=(V,E)$，如果下述任何一条满足，则T是一个树。

（1）T连通且无回路。

（2）T任意两顶点都有唯一的路连接。

（3）T无回路，但任意加上一个新的边就构成唯一回路。

（4）T连通且每条边都是割边。

（5）T连通且有$|V|-1$条边。

（6）T无回路且有$|V|-1$条边。

上述树定义可以扩充到有向图的树，但本书仅讨论无向图的树的情况。

定理 5.4. 定义 5.18 中的各条陈述等价。

证明：我们按照（1）推出（2），（2）推出（3），（3）推出（4），（4）推出（5），（5）推出（6），（6）推出（1）的方式来证明该定理。

第一步，如果结论（1）成立，任取两顶点$u_1,u_2\in V$，因为T连通，所以u_1,u_2之间有路连接。我们使用反证法证明，如果u_1,u_2之间存在两条以上的路连接，不妨设其中两条路为p_1和p_2，且$p_1\neq p_2$，则p_1和p_2构成回路，这与我们的假设相矛盾。因此结论（2）成立。

第二步，如果结论（2）成立，类似于上一步，我们可以用反证法证明如果T任意两顶点都有唯一的路连接，则T无回路。进一步，如果任取两顶点$u_1,u_2\in V$，存在唯一一条路为p_1，则增加一条边e_1直接连接u_1,u_2，则得到唯一的回路

p_1+e_1。因此结论（3）成立。

第三步，如果结论（3）成立，则T无回路，因此，T中的每一个边都是割边。接着，我们用反证法证明T必然是连通的。假设T不连通，其中至少有两个不连通的子图T_1和T_2，显然，在T_1中找一顶点u_1，在T_2中找一顶点u_2，增加一条边e_1直接连接u_1,u_2，则图中依然无法形成环，这与我们的假设相矛盾。因此结论（4）成立。

第四步，如果结论（4）成立，我们使用数学归纳法进行推导。显然$|V|=1$和$|V|=2$时，必然成立$|E|=|V|-1$。假设当$|V|\le k$时，都有$|E|=|V|-1$。则当$|V|=k+1$时，因为T中的每一个边都是割边，我们可以任意取其中一个边割去，得到两个不连通的子图$T_1=(V_1,E_1)$和$T_2=(V_2,E_2)$，且这两个子图中的顶点数目都小于k，因此满足$|E_1|=|V_1|-1$，$|E_2|=|V_2|-1$。所以，T的总边数应该满足$|E|=|E_1|+|E_2|+1=|V_1|-1+|V_2|-1+1=|V|-1$。因此，$|V|=k+1$时，结论也成立。根据数学归纳法，结论（5）成立。

第五步，如果结论（5）成立，我们使用数学归纳法进行推导。当$|V|=2$时，显然$|E|=1$，连通无回路。假设当$|V|\le k$时，都满足连通无回路。则则当$|V|=k+1$时，因为T是连通的，所以T中任意一顶点的度大于1，而且T中至少存在一顶点u_1的度等于1。否则，如果T中任意一顶点的度大于等于2，则T的总度数应该为$2|V|\le 2|E|=2|V|-2$，这会导致矛盾。根据归纳假设，去掉该顶点u_1得到的子图$T_1=(V_1,E_1)$的边数为$|E_1|=|E|-1$，顶点数为$|V_1|=|V|-1$，且无回路。则T显然也无回路。根据数学归纳法，结论（6）成立。

第六步，如果结论（6）成立，此时T有$|V|-1$条边，且无回路，所以每条边都是割边。我们可以用反证法证明T连通。假设T不连通，有$m>1$个不联通的子图。则我们可以通过在这些子图间添加$m-1$条边得到一个全部顶点连通的图$T'=(V',E')$。显然，T'中的边数为$|E|-1+m-1$。但另一方面，结论（6）指出应该T'应该含有$|E|-1$条边。两者矛盾。因此结论（1）成立。□

定理 5.5. 每个树至少有两个度为1的顶点。若$T=(V,E)$恰好有两个度为1的顶点，则其它顶点的度必为2，因此$T=(V,E)$是一条路。

证明：我们用反证法证明。假设只有 0 个或者 1 个度为 1 的顶点（叶子），那么剩下的顶点的度都大于或等于 2，所以图的总度数大于或等于$2(n-1)+1$。但树只有$n-1$条边，这就产生了矛盾。

类似的，根据树只有$n-1$条边的的性质，可以推出，如果恰好有两个度为 1 的顶点，则其它顶点的度必为 2，又根据连通性，得出是一条路。□

定义 5.22. 如果$G=(V,E)$的支撑子图$T=(V,E')$是树，则称其为G的支撑树（Spanning Tree）。G中属于T的边成为树枝，不属于T的边称为弦。

定理 5.5. 无向图$G=(V,E)$有支撑树的充要条件是G是连通图。

证明：上述结论的必要性显然。充分性的证明等价于对任意的连通图给出确定支撑树的算法。参见下述算法。□

算法 5.3. 无向图的深度优先支撑树算法

```
输入: 存放图的一维顶点数组 vertice[]，二维邻接矩阵数组 adjacency[][]
输出: 存放支撑树的二维邻接矩阵数组 spanningtree[][] 或者 不连通标志
中间变量: 存放图的顶点是否访问过信息的一维数组 visitedvertice []

function RecurssiveDFS(v)
{   visitedvertice [v] = True;   // 标记已经访问过的顶点
    for u = 1 : sizeof(vertice)
    {
        if adjacency[v][u] >= 1 && visitedvertice [u] == False
          { spanningtree [v][u] = 1;   // 将这条边加入支撑树
            RecurssiveDFS(u);   // 递归访问尚未展开的顶点
          }
    }
}

function spanningtree_DFS(vertice[], adjacency[][])
{
 for u = 1 : sizeof(vertice)   // 先将所有顶点设为没有访问过
    { visitedvertice [u] = False;
    }

 RecurssiveDFS(1);    // 从第一个顶点开始深度优先搜索

 connectedflag = True;
 for u = 1 : sizeof(vertice)   // 深度优先搜索之后，遍历查询访问记录
 {
    if visitedvertice [u] == False
       {connectedflag = False;
       }
 }

 if connectedflag == False
```

```
    {return connectedflag;   // 输出图不连通的结果
    }
 else
    {return spanningtree;   // 输出找到的 spanning tree
    }
}
```

算法 5.4. 无向图的广度优先支撑树算法

```
输入：存放图的一维顶点数组 vertice[]，二维邻接矩阵数组 adjacency[][]
输出：存放支撑树的二维邻接矩阵数组  spanningtree[][]
中间变量：存放待访问顶点的队列  tobecheckedqueue

function spanningtree_BFS()
{
 tobecheckedqueue = new queue;   // 建立用于广度优先搜索的队列

 for u = 1 : sizeof(vertice)   // 先将所有顶点设为没有访问过
    { visitedvertice [u] = False;
    }

 tobecheckedqueue.append(1);   // 从第一个顶点开始广度优先搜索
 visitedvertice [1] = True;

 while tobecheckedqueue is not empty
 {
     v = tobecheckedqueue.pop();
     for u = 1 : sizeof(vertice)
     {
         if adjacency[v][u] >= 1 && visitedvertice [u] == False
             { spanningtree [v][u] = 1;   // 将这条边加入支撑树
               tobecheckedqueue.append (u);
               visitedvertice [u] = True;
             }
      }
    }

 connectedflag = True;
```

```
for u = 1 : sizeof(vertice)   // 广度优先搜索之后，遍历查询访问记录
{if visitedvertice [u] == False
    {connectedflag = False;
    }
}

if connectedflag == False
  {return connectedflag;   // 输出图不连通的结果
  }
else
  {return spanningtree;   // 输出找到的 spanning tree
  }
}
```

有的研究者也将上述算法称为生成树算法，强调树是逐步构建出来的。

例 5.5. 请分别用深度优先和广度优先生成图 5.3（b）中无向图的支撑树。

解：图 5.4（a）显示了用深度优先支撑树算法求出的支撑树。图 5.4（b）显示了用深度优先支撑树算法求出的支撑树。不难发现，一个图的支撑树可能不止一棵。

图 5.4　支撑树算法示意图：（a）用深度优先支撑树算法求出的支撑树；（b）用深度优先支撑树算法求出的支撑树。

5.3.2　最小支撑树

定义 5.23. 网络 $G=(V,E,W)$ 的任意一个支撑树 T 的所有树枝上的权的总和，记为 $L(T)$，称为这个支撑树的权，具有最小权的支撑树称为最小支撑树（Minimum Spanning Tree），也被称为最小树或者最小生成树。

最小支撑树在交通网、电力网设计等实际问题中广泛应用。假设我们需要把多个地点用交通网或者电力网络连接起来，往往希望总建设费用最低。此时即可应用最小生成树算法求解最优建设方案。

对于给定图求最小支撑树的典型算法包括 Kruskal 算法和 Prim 算法。

Kruskal 算法（避圈法）：将所有边按权值从小到大排序，从权值最小的边开始选树枝加入树中，如果可能形成回路（圈）则跳过，直至已经选够顶点数减 1 的树枝。

算法 5.5. 最小支撑树的 Kruskal 算法

输入：存放图的一维顶点数组 vertice[] ,一维边权元胞数组 edgeweights[] (每个元胞存有边权、入点、出点三种信息)

输出：存放支撑树的二维邻接矩阵数组 spanningtree[][]

中间变量：存放排序后的一维边权元胞数组 sourtededgeweights[] (每个元胞存有边权、入点、出点三种信息) ,查询那些顶点被访问过的哈希表 visitedvertices

```
function minimumspanningtree_Kruskal (vertice, edgeweights)
{
    if sizeof(edgeweights[]) < 1   // 只有零条边直接退出
    {
        return NoNeedtoDoThis;
    }

    sourtededgeweights = sourted(edgeweights);   // 根据边权递增排序
    numberofbranches = 1;   // 记录已经找到几根树枝

    while numberofbranches < sizeof(vertice)   // 已经找到足够树枝则停止
    {
        for u = 1 : sizeof(sourtededgeweights[])   // 遍历所有树枝
        {if sourtededgeweights[u].InVertex is NOT in visitedvertice &&
            sourtededgeweights[u].OutVertex is NOT in visitedvertice
            {spanningtree[sourtededgeweights[u].InVertex]
[sourtededgeweights[u].OutVertex] = 1;   // 加入该树枝到最小支撑树
                visitedvertice.Add(sourtededgeweights[u].InVertex);
                visitedvertice.Add(sourtededgeweights[u].OutVertex);   // 记住相关顶点
                numberofbranches = numberofbranches + 1;
```

```
                }
            }
        }

        return spanningtree;
    }
```

定理 5.7. T 是无向图 $G=(V,E)$ 的最小支撑树的充要条件是：加入任何弦形成的回路中，弦的权值最大。

证明：我们首先证明必要性。如果加入某个弦形成的回路中有比该弦的权值更大的树枝，则用该弦代替最大树枝形成的支撑树的总权值会变小，和最小支撑树定义矛盾。因此，必然性得证。

接着我们使用反证法证明充分性。设 T_1 是满足条件的支撑树，T_2 是所有最小支撑树中和 T_1 不同的树枝数最少的树。这样的树一定存在，假设只有 m 个树枝不一样，不妨将这两棵树分别记为

$$T_1=\left\{e_1,e_2,\cdots,e_m,\hat{T}\right\},\quad T_2=\left\{\overline{e}_1,\overline{e}_2,\cdots,\overline{e}_m,\hat{T}\right\}$$

其中我们根据对应的权重新排列了不同树枝，有 $w(\overline{e}_1)\le w(\overline{e}_i)$，$\forall 2\le i\le m$。

将 $\overline{e}_1$ 加入 T_1 会形成回路，根据算法假设，一定有 $e_k\in T_1\setminus\hat{T}\Rightarrow w(e_k)\le w(\overline{e}_1)$。

将这样找到的 e_k 加入 T_2 同样会形成回路。根据 T_2 是最小支撑树假设，此时一定有 $\overline{e}_j\in T_2\setminus\hat{T}$；$w(e_k)\le w(\overline{e}_1)\le w(\overline{e}_j)$。则我们可以在 T_2 的基础上构建一个新的树 $T_2(e_k\setminus\overline{e}_j)$，也就是将 T_2 中的 $\overline{e}_j$ 边删掉，增加 e_k 边。

如果 $w(e_k)\le w(\overline{e}_j)$ 中的不等式严格成立，也即 $w(e_k)<w(\overline{e}_j)$，则与 T_2 是最小支撑树的假设矛盾。

否则 $w(e_k)\le w(\overline{e}_j)$ 中的等式成立，也即 $w(e_k)=w(\overline{e}_j)$，则 $T_2(e_k\setminus\overline{e}_j)$ 仍是最小支撑树。但此时，$T_2(e_k\setminus\overline{e}_j)$ 和 T_1 的不同树枝数为 $m-1$，与我们假设的 T_2 是所有最小支撑树中和 T_1 不同的树枝数最少的树相矛盾。

因此，假设不成立。充分性得证。□

定理 5.7 直接证明了 Kruskal 算法产生的是最小支撑树。

Prim 算法：从图中任意顶点开始，逐渐往某个顶点集中加入图中尚未加入的顶点。如果将该顶点集记为 S，每次从不在 S 的顶点集里选择距 S 一步距离最小的顶点加入 S，将相应边取为树枝，直至 S 包含所有的顶点。

具体执行时，可以按如下方式简化所需比较的次数。对图 $G=(V,E)$ 设置集合 S，存放已访问的顶点，然后每次从集合 $V-S$ 中选择与集合 S 的最短距离最

小的一个顶点（记为u），将其加入集合S。之后，令顶点u为中介点，优化所有从u能到达的顶点v与集合S之间的最短距离。迭代直到集合S已包含所有顶点。

算法 5.5. 最小支撑树的 Prim 算法

输入：一维数组 vertice[]存放图的所有顶点，一维数组 edgeweights[]存放边权、入点、出点信息

输出：二维邻接矩阵 spanningtree[][] ,表示最小支撑树。如 spanningtree[u][v]表示顶点 u 和顶点 v 之间存在一条边

中间变量：优先队列 Q 用于存放还未加入到最小支撑树中的顶点，按照 key 值属性排序。u.key 表示从顶点 u 到集合 S 任一顶点的最小边权重。u.pi 表示集合 S 中连接到 u 的顶点。Extract-Min(Q)函数从优先队列中提取具有最小 key 值的顶点。Adjacent(u)函数获取顶点 u 的邻接顶点列表

```
function minimumspanningtree_Prim(vertice, edgeweights)
{
    for each vertex u in vertice
    {   u.key = Infinity //  初始 key 值无穷大，表示没有任何边连接到集合 S
        u.pi = NIL //  初始 pi 值为 NIL，表示没有父节点
        vertice[0].key = 0 //  将起始顶点的 key 值设为 0
    }

    Q.buildfrom(vertice); //  建立优先队列 queue，包含所有顶点

    spanningtree = [] //  存放生成树的数组，初始化为空

    while Q is not empty
    {
        u = Extract-Min(Q) //  提取并移除 Q 中 key 值最小的顶点 u

        if u.pi is not NIL
        {   add edge (u, u.pi) to spanningtree //  将顶点 u 加入生成树

            for each v in Adjacent(u) //  对于顶点 u 的每一个邻接顶点 v
            {
                if v is in Q and edgeweights[u][v] < v.key //  如果 v 在 Q 中，并且边(u,v)的权重小于 v 的 key 值
```

```
                {    v.pi = u //  更新 v 的 pi 值为 u
                     v.key = edgeweights[u][v] //更新 v 的 key 值为边(u,v)的权
                }
            }
        }
    }

    return spanningtree
}
```

定理 5.8. T 是无向图 $G=(V,E)$ 的最小支撑树的充要条件是：任何树枝都是所在的唯一割集中权值最小的边。

证明：我们首先证明必要性。如果树枝不是所在的唯一割集中权值最小的边，用权值最小的边代替相应树枝可得总权值更小的支撑树。因此，必然性得证。

接着我们证明充分性。加入任何弦形成的回路中，新加入的弦和回路上任何树枝都在某个唯一的割集上。所以，新加入的弦弦的权值最大。因此，充分性得证。□

定理 5.8 直接证明了 Prim 算法产生的是最小支撑树。

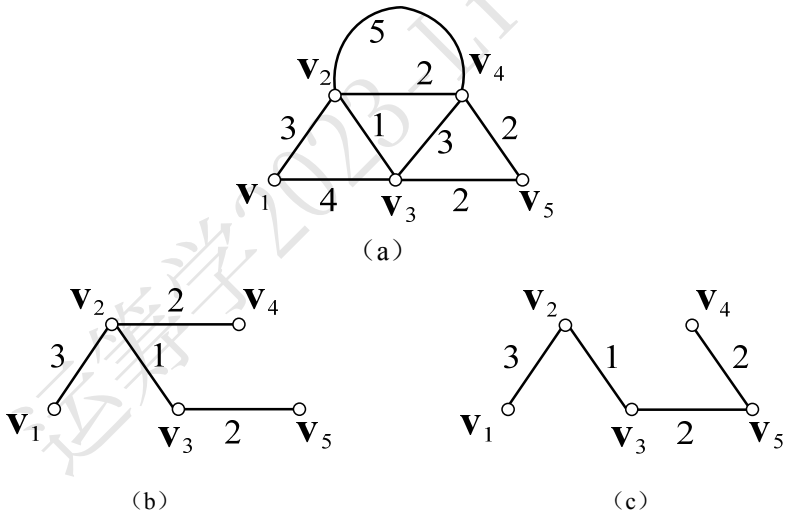


图 5.5　最小支撑树算法示意图：（a）初始网络；（b）用 Kruskal 算法求出的最小支撑树；（c）用 Prim 算法求出的最小支撑树。

例 5.6. 请分别用 Kruskal 算法和 Prim 算法生成图 5.5（a）中网络的最小支撑树。

解：图 5.5（b）显示了用 Kruskal 算法求出的最小支撑树。图 5.5（c）显示了从 v_5 点开始，用 Prim 算法求出的最小支撑树。不难发现，一个图的最小支撑树可能不止一棵。

5.4　最短路问题

定义 5.24. 给定连通网络 $G=(V,E)$ 及各边 $(v_i,v_j)\in E$ 的权 l_{ij}，其中 $v_i,v_j\in V$；如果 $(v_i,v_j)\notin E$，规定 $l_{ij}=+\infty$（在计算机求解时可以设为一个足够大的值）。指定起点 $v_s\in V$ 和终点 $v_t\in V$，设 Ω_{st} 表示从 v_s 到 v_t 的所有路的集合。最短路问题是寻找一条从起点 v_s 到终点 v_t 的路 $\mu\in\Omega_{st}$，使它是从 v_s 到 v_t 的所有道路中总权最小的路（Shortest-Path Problem）

$$L(\mu^*)=\min_{\mu\in\Omega_{st}}L(\mu)=\sum_{(v_i,v_j)\in\mu}l_{ij} \tag{5.1}$$

最短路在交通网、电力网设计等实际问题中广泛应用。假设网络中的每条边都有一个权（长度、成本、时间等），我们需要找出给定起点和终点之间总权最小的路。此时即可应用最短路算法求解最优建设方案。

Floyd-Warshall 算法又称为插点法，是一种利用动态规划的思想寻找给定的无向网络中多起点最短路问题（Multiple-Source Shortest-Path Problem）的算法。其基本想法是进行 n 轮迭代来维护并动态更新一个存有所有顶点到所有顶点的最短距离的的二维数组 dist[][]。在第 k 轮迭代中，我们把第 k 个点加入图中，来试探是否 dist[][]中是否有路长度可以被更改。我们都对任意顶点对 i, j 尝试如下的松弛操作：d[i][j] = min(d[i][j], d[i][k] + d[k][j])。显然，这种松弛操作就相当于考察能否经由 k 点使顶点 i 到顶点 j 的距离变短。如果一个含有 n 个顶点的图中不存在总权为负值的环，则任意两点之间的最短路最多包含 $n-1$ 边，所以我们最多只需要 n 轮迭代即可收敛。如果再多迭代第 $n+1$ 轮，尝试重新加入第 1 个点，再试探是否 dist[][]中是否有路长度可以被更改，如果这轮迭代还有顶点对的最短距离被更新，就说明存在总权为负值的环。

显然，如果没有总权为负值的环，则 Floyd-Warshall 算法的时间复杂度为 $O\left(|V|^3\right)$，空间复杂度 $O\left(|V|^2\right)$。

算法 5.7. 无向网络多源最短路的 Floyd-Warshall 算法

输入：顶点个数 V，图的二维邻接矩阵数组 dist[][]，d[u][v]表示 u->v 的权值，

没有直接相连的两点默认为一个足够大的值 ,d[u][u]则默认为 0; 记录最短路上前驱顶点的二维数组 p[][]，p[i][j]初始默认值为 j

输出: 二维邻接矩阵数组 dist[][]，其中 d[u][v]表示 u->v 的最短路的总权值，二维路矩阵数组 p[][]，其中 p[i][j]表示从 i 到 j 的最短路上的第一个中间顶点

```
function Floyd()
{
    for k = 0: V-1 //试探加入图中的第 k 个点
      {
          for i = 0: V-1 //通过 i、j 双重循环，遍历图中的每一个点
            {
              for j = 0: V-1
                { //判断点对(i,j)间的最小距离是否因为加入的点 k 而发生改变
                  if d[i][j] > d[i][k] + d[k][j]
                    {
                      d[i][j] = d[i][k] + d[k][j];//若发生改变，则修改最短路
                      p[i][j] = p[i][k]; //  更新 i 到 j 的中间顶点
                    }
                }
            }
       }
}

function print_path(s, t) //  输入起点 s，终点 t，打印 s->t 的最短路
{
      temp = p[s][t];
      print(s); //打印起点
      while temp != w //  打印中间点
      {
          print(temp);
          temp = p[temp][t];
      }
      print(t); //打印终点
}
```

Bellman-Ford-Moore 算法是一种利用动态规划的思想寻找给定的有向/无向网络中单起点最短路问题（Single-Source Shortest-Path Problem）的算法。该方法

可以容许网络中权为负值，甚至网络中存在总权为负值的环。其基本想法是进行 $n-1$ 轮迭代来维护并动态更新一个存有起点到所有顶点的最短距离的的一维数组 dist[]。每轮迭代中，我们都对任意顶点对 x, y 尝试如下的松弛操作：d[y] = min(d[y], d[x] + e[x][y])；这里的 e[x][y]表示顶点对 x、y 之间直连边的长度。显然，这种松弛操作就相当于考察能否经由 x 点使起点到 y 点的距离变短。第 1 轮迭代之后，我们得到的是起点最多经过一条边到达其他顶点的最短距离；第 2 轮迭代之后，我们得到的是源点最多经过两条边到达其他顶点的最短距离；第 3 轮迭代之后，我们得到的是源点最多经过三条边到达其他顶点的最短距离。如果一个含有 n 个顶点的图中不存在总权为负值的环，则任意两点之间的最短路最多包含 $n-1$ 边，所以我们最多只需要 $n-1$ 轮迭代即可收敛。如果再多迭代第 $n+1$ 轮，对每条边再尝试松弛一遍，如果这轮迭代还有顶点对的最短距离被更新，就说明存在总权为负值的环。

显然，如果没有总权为负值的环，则 Bellman-Ford-Moore 算法的时间复杂度为 $O(|V||E|)$，空间复杂度 $O(|V|^2)$。

算法 5.8. 有向/无向网络单源最短路的 Bellman-Ford-Moore 算法

输入: 顶点个数 V，边数 E，起点 s，记录起点 s 到每个顶点的最短距离的一维数组 d[],记录最短路上每个顶点的前驱顶点 pre[],存储图中所有边的信息的元胞数组 edges[]，其中 edges[i].from 为边 i 的始点，edges[i].to 为边 i 的终点，edges[i].cost 为边 i 的权值

输出: 最短距离存储在全局变量 d[]中，最短路上各点的前驱节点存储在全局变量 pre[]中

```
function Bellman_Ford_Moore(s)
{
    for i = 0 : V-1   // 将 d[]初始值设为 INF，一个足够大的数
      { d[i] = INF; }

    d[s] = 0;   // 将起点到自身的距离记为 0

    for j = 0 : V-1   // 迭代进行 V-1 轮松弛操作，在没有负权圈的情况下必能获得结果
      {
        for i = 0: E-1 //对所有的边都进行一次松弛操作
          {
            Edge temp = edges[i];
            //若边 i 可使起点 s 到边 i 终点的距离变短，则更新最短距离
```

```
                if (d[temp.from] != INF) &&
                   (d[temp.to] > d[temp.from] + temp.cost)
                    {
                      d[temp.to] = d[temp.from] + temp.cost;   // 更新最短距离
                      pre[temp.to] = temp.from; //更新前驱顶点

                      if j == V-1 // 判断迭代次数是否超过了 V-1 次
                        {
                          print("存在负圈\n");
                          return;
                        }
                    }
                }
        }
}

function print_path(s, t)   // 打印从起点 s 到终点 t 的最短路
{
        if s == t
        {
                print(s);   // 打印起点也即终点
                return;
        }
        print_path(s, pre[t]);   // 递归打印前驱顶点
        print(t); //打印终点
}

Bellman_Ford_Moore(1);   // 寻找顶点 1 到其他顶点的最短路
print(1, V-1);   // 打印顶点 1 到顶点 V-1 的最短路
```

Dijkstra 算法是一种利用动态规划的思想寻找给定的有向/无向网络中单起点最短路问题（Single-Source Shortest-Path Problem）的算法。该方法不容许网络中权为负值。其本质想法是：在 Bellman-Ford-Moore 算法定义的每轮迭代松弛操作中，必然会有一个顶点到起点的最短距离被确定，并在后面的迭代中不会被改变。因此，我们只需要在每轮找到最短路刚刚被确定的顶点，更新从其出发的相邻顶点到起点的最短距离，而不需要再去更新前面迭代中已经确定的那些顶点的相邻顶点。所以没有权为负值的情况下，Dijkstra 算法就比 Bellman-Ford-Moore 算法

少一重遍历，速度更快。

如果我们用数组这一数据结构来记录找到的距离最小的顶点和标记顶点是否使用（已找到最短路），则 Dijkstra 算法的时间复杂度为$O\left(|V|^2\right)$，空间复杂度$O\left(|V|^2\right)$。

算法 5.9. 基于数组实现的有向/无向网络单源最短路的 Dijkstra 算法

如果我们用堆这一数据结构来记录找到的距离最小的顶点和标记顶点是否使用（已找到最短路），则 Dijkstra 算法的时间复杂度为$O\left(|E|\log_2\left(|V|\right)\right)$，空间复杂度$O\left(|V|\right)$。

算法 5.10. 基于堆实现的有向/无向网络单源最短路的 Dijkstra 算法

输入：顶点个数 V，边数 E，起点 s，记录起点 s 到每个顶点的最短距离的一维数组 d[]，记录最短路上每个顶点的前驱顶点 pre[]，存储图的邻接表 G[]，记录各点使用情况的一维数组 used[]

输出：最短距离存储在全局变量 d[]中，最短路上各点的前驱节点存储在全局变量 pre[]中

```
int V;   // 顶点个数
int E;   // 边数
struct edge{int from, to,, cost};   // 边：{始点，终点，权值}
vector<edge> G[];   // 使用邻接表存储图
int d[];   // 从起点到其他点的最短距离
bool used[];   // 已经确定了最短距离的点
struct s_node{int id, dis};   // {顶点编号，该顶点到起点的最短距离}
int pre[];   // 记录前驱顶点

function Dijkstra(s)
{
    for i = 0: N-1 // 初始化
      {
          d[i] = INF; // 最短距离为 INF
          used[i] = False; // 所有点都还没被使用过
```

```
    }
  d[s] = 0; //起点到自身的距离为 0
  priority_queue<s_node> Q;   // 优先队列，存储节点信息
  Q.push(s_node(s, d[s]));   // 起点入队
  while (!Q.empty())   // 队列为空则说明所有顶点都已经被使用过了
  {
   s_node u = Q.top();   // 距离起点 s 最近的顶点 u
   Q.pop();
   if used[u.id]   // u 已经被使用过
     continue;
   else
     used[u.id] = true;

   for i = 0: G[u.id].size - 1
   {
      edge v = G[u.id][i];   // u.id 的第 i 个邻居是 v.to
      if used[v.to]   // 此邻居已经找到最短路
        continue;
      else
        { if d[v.to] > u.dis + v.cost   // 经过 u 可以使 v 的最短距离变小
            {
              d[v.to] = u.dis + v.cost;   // 更新最短距离
              Q.push(s_node(v.to, d[v.to]));   // 新邻居入队
              pre[v.to] = u.id;   // 记录最短路上的前驱顶点
            }
        }
   }
}

function print_path(s, t)   // 打印从起点 s 到终点 t 的最短路
{
   if s == t
   {
        print(s);   // 打印起点
        return;
   }
   print_path(s, pre[t]);   // 递归打印前驱顶点
```

```
        print(t);   // 打印终点
    }

    Dijkstra(1);   // 寻找顶点 1 到其他顶点的最短路
    print(1, V-1);   // 打印顶点 1 到顶点 V-1 的最短路
```

当网络稀疏时，$|V|^2 \gg |E|$，基于堆实现的 Dijkstra 算法更优；当网络稠密时，$|V|^2 \sim |E|$，基于数组实现的 Dijkstra 算法更优。

Floyd-Warshall 算法、Bellman-Ford-Moore 算法和 Dijkstra 算法的证明比较简单，留给大家自行练习。

例 5.7. 对于图 5.6 所示网络，请用 Floyd-Warshall 算法求解其中各点的最短路，请用 Bellman-Ford-Moore 算法和 Dijkstra 算法求解其中 v_1 点到 v_4 点的最短路。

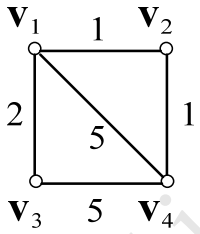


图 5.6　某无向图示意图。

解：用 Floyd 算法求解时，初始距离矩阵和路径矩阵为

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 5 \\ 1 & 0 & \infty & 1 \\ 2 & \infty & 0 & 5 \\ 5 & 1 & 5 & 0 \end{pmatrix} \end{array}, \quad \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} v_1 & v_2 & v_3 & v_4 \\ v_1 & v_2 & \phi & v_4 \\ v_1 & \phi & v_3 & v_4 \\ v_1 & v_2 & v_3 & v_4 \end{pmatrix} \end{array}$$

经过第一轮迭代后距离矩阵和路径矩阵为

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 3 & 1 \\ 2 & 3 & 0 & 5 \\ 2 & 1 & 5 & 0 \end{pmatrix} \end{array}, \quad \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} v_1 & v_2 & v_3 & v_2 \\ v_1 & v_2 & v_1 & v_4 \\ v_1 & v_1 & v_3 & v_4 \\ v_2 & v_2 & v_3 & v_4 \end{pmatrix} \end{array}$$

经过第二轮迭代后距离矩阵和路径矩阵为

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 3 & 1 \\ 2 & 3 & 0 & 4 \\ 2 & 1 & 4 & 0 \end{pmatrix} \end{array}, \quad \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} v_1 & v_2 & v_3 & v_2 \\ v_1 & v_2 & v_1 & v_4 \\ v_1 & v_1 & v_3 & v_1 \\ v_2 & v_2 & v_2 & v_4 \end{pmatrix} \end{array}$$

验证可知此时已经收敛到最优解。

用 Bellman-Ford-Moore 算法求解时，初始和第一轮迭代后的距离矩阵分别为

$$\begin{array}{c} \\ v_1 \end{array}\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & \infty & \infty & \infty \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \\ v_1 \end{array}\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 5 \end{pmatrix} \end{array}$$

第二轮迭代后的距离矩阵为

$$\begin{array}{c} \\ v_1 \end{array}\begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 2 \end{pmatrix} \end{array}$$

验证可知此时已经收敛到最优解。

用 Bellman-Ford-Moore 算法求解时，第一轮迭代可以确定 v_1 到 v_2 的最短路长度为 2，今后不再改变。第二轮迭代可以确定 v_1 到 v_3 的最短路长度为 2，今后不再改变。第二轮迭代可以确定 v_1 到 v_4 的最短路长度为 2，今后不再改变。验证可知此时已经收敛到最优解。

5.5 最大流和最小费用流问题

5.5.1 最大流

定义 5.25. 给定连通有向网络 $G=(V,E)$ 及各边 $(v_i,v_j)\in E$ 的权 c_{ij}，对于任意 $v_i,v_j\in V$，规定 $c_{ij}\geq 0$ 代表该边的容量。仅有一个入次为 0 的顶点 $v_s\in V$，称为源点；仅有一个出次为 0 的顶点 $v_t\in V$，称为汇点。

假设针对每条边赋予一个流量值 x_{ij}，则称 $F=\{x_{ij}\}$ 为网络 G 的一个流。并且称满足下列容量约束条件和流量平衡约束条件的 $F=\{x_{ij}\}$ 为可行流：

$$0\leq x_{ij}\leq c_{ij} \tag{5.2}$$

$$\sum_{(v_i,v_j)\in E} x_{ij}=\sum_{(v_k,v_i)\in E} x_{ki},\quad \forall v_i\in V, i\neq s,t \tag{5.3}$$

显然，任一可行流的网络总流量为

$$W=\sum_{(v_s,v_j)\in E}x_{sj}=\sum_{(v_k,v_t)\in E}x_{kt} \tag{5.4}$$

进一步，我们可以定义最大流问题（Maximum-Flow Problem）

$$\max_{W,x_{ij}\in\mathbb{R}} \quad W$$

$$\text{s.t.} \quad \sum_{(v_i,v_j)\in E}x_{ij}-\sum_{(v_j,v_i)\in E}x_{ji}=\begin{cases}W & \text{if } i=s\\ 0 & \text{if } i\notin\{s,t\}\\ -W & \text{if } i=t\end{cases} \tag{5.5}$$

$$0\le x_{ij}\le c_{ij},\ \forall(v_i,v_j)\in E$$

注意到所有边上流量为 0 的流 $F=\{x_{ij}=0\}$ 是一个可行流，因此最大流问题一定有解。同时因为 $W\le\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\le\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}$，因此最大流问题有界。

最大流问题（5.5）是线性规划问题，所以可以用单纯型法或者原对偶内点法直接求解。但其流量平衡约束条件具有的特殊形式，其中约束矩阵的每一列只有一个元素为 1 和另一个元素为 −1，其余元素全为 0；而对应的右侧向量则除了第一个元素是 W，第一个元素是 $-W$，其余元素都为 0。因此，我们也可以结合网络算法来进行求解。

定义 5.25. 对于最大流问题的连通有向网络图 $G=(V,E,C)$，任取 $S\subset V$，其补集为 $\overline{S}=V-S$，若 S 和 $\overline{S}$ 都不是空集，且 $v_s\in S$，$v_t\in\overline{S}$。我们称 $\{S,\overline{S}\}$ 为该连通网络的割集。而所有属于 $\{S,\overline{S}\}$ 的边的容量总和称为割集容量，并记为 $C(S,\overline{S})$，这些割集中具有最小割集容量的割集称为最小割（Minimum Cuts）。

定理 5.9. 设 $F=\{x_{ij}\}$ 为连通有向网络 $G=(V,E,C)$ 的任一可行流，流量为 W，$\{S,\overline{S}\}$ 是分离 v_s 和 v_t 的任一割集，则有 $W\le C(S,\overline{S})$。

证明：注意到任取一个分离 v_s 和 v_t 的割集 $\{S,\overline{S}\}$，和任意一个可行流 $\{x_{ij}\}$，我们有

$$W=\sum_{v_i\in S}\left(\sum_{(v_i,v_j)\in E}x_{ij}-\sum_{(v_j,v_i)\in E}x_{ji}\right)=\sum_{v_i\in S}\left(\sum_{\substack{v_j\in S\\(v_i,v_j)\in E}}x_{ij}-\sum_{\substack{v_j\in S\\(v_j,v_i)\in E}}x_{ji}\right)$$

$$+\sum_{v_i\in S}\left(\sum_{\substack{v_j\in\bar{S}\\(v_i,v_j)\in E}}x_{ij}-\sum_{\substack{v_j\in\bar{S}\\(v_j,v_i)\in E}}x_{ji}\right)=\sum_{v_i\in S}\left(\sum_{\substack{v_j\in\bar{S}\\(v_i,v_j)\in E}}x_{ij}-\sum_{\substack{v_j\in\bar{S}\\(v_j,v_i)\in E}}x_{ji}\right)$$

$$\le\sum_{v_i\in S}\sum_{\substack{v_j\in\bar{S}\\(v_i,v_j)\in E}}x_{ij}\le C\left(S,\bar{S}\right)$$

因此，结论成立。□

定义 5.27. 假设 $F=\{x_{ij}\}$ 是连通网络 $G=(V,E,C)$ 的一个可行流。我们称 $N=(V,E,C-F)$ 为网络 G 相对于流 F 的残差网络（Residual Network）。假设网络 G 中存在从顶点 v_i 指向顶点 v_j 的边，如果 $f_{ij}<c_{ij}$，则在网络 N 中存在从顶点 v_i 指向顶点 v_j 的边，其容量为 $\hat{c}_{ij}=c_{ij}-f_{ij}$；如果 $f_{ij}>0$，则在网络 N 中存在从顶点 v_j 指向顶点 v_i 的边，其容量为 $\hat{c}_{ij}=f_{ij}$。如果网络 N 中的一条从 v_s 到 v_t 的初级路 μ 中的每个边都具有正容量，则称其为可增广路（Augmenting Path），有的文献也直接称之为增广路。

定义 5.28. 设 μ 是将连通有向网络 $G=(V,E,C)$ 中从起点 v_s 到终点 v_t 的一条初级路。我们定义 μ 的方向为从 v_s 到 v_t 的方向，对于 μ 上的任意有向边，如果其方向和 μ 相同则称其为前向边，否则为后向边，用 μ^+ 和 μ^- 分别表示前向边和后向边的集合，如果网络 G 中的 $F=\{x_{ij}\}$ 是一个可行流，且满足前向边流量可增，后向边流量可减

$$x_{ij}<c_{ij},\quad \forall(v_i,v_j)\in\mu^+ \tag{5.6}$$

$$0<x_{ij},\quad \forall(v_i,v_j)\in\mu^- \tag{5.7}$$

则称 μ 是网络 G 中从起点 v_s 到终点 v_t 的一条可增广路。不难看出定义 5.28 种的可增广路和定义 5.27 中的可增广路实际上是经过同样顶点和有向边序列的同一条路，只不过定义 5.28 的方式不需要引入残差网络。

定理 5.10. 一个可行流 $F=\{x_{ij}\}$ 是连通网络 $G=(V,E,C)$ 的最大流的充要条件是：网络 G 相对于流 F 的残差网络 $N=(V,E,C-F)$ 中不存在可增广路。

证明：我们用反证法证明必要性。假设 $F=\{x_{ij}\}$ 已经是最大流，但存在可增

广路 μ，则我们至少可以沿着其中一条可增广路可以多运送如下数量的货物

$$\delta=\min\left\{\left\{c_{ij}-x_{ij},\forall\left(v_i,v_j\right)\in\mu^{+}\right\},\left\{x_{ij},\forall\left(v_i,v_j\right)\in\mu^{-}\right\}\right\}>0 \quad (5.8)$$

而不违反容量约束，这与假设 $F=\left\{x_{ij}\right\}$ 是最大流相矛盾。

接着我们证明充分性。假设我们用 S 表示的残差网络 $N=\left(V,E,C-F\right)$ 中，从起点 v_s 有可增广路达到的点集，即，对任意的 $v_k\in S$，存在从 v_s 到 v_k 的路，满足以下条件：

如果 $\left(v_i,v_j\right)$ 与路的方向相同，则 $0\le x_{ij}<c_{ij}$

如果 $\left(v_i,v_j\right)$ 与路的方向相反，则 $0<x_{ij}\le c_{ij}$

因此，残差网络 N 不存在可增广路，等价于 $v_t\in\overline{S}$。

由 S 的定义可知，对于任意的 $v_i\in S,v_j\in\overline{S}$，我们有

如果 $(v_i,v_j)\in E$，则一定有 $x_{ij}=c_{ij}$，否则 $v_j\in S$

如果 $(v_j,v_i)\in E$，则一定有 $x_{ji}=0$，否则 $v_j\in S$

故而

$$W=\sum_{v_i\in S}\left(\sum_{\substack{v_j\in\overline{S}\\(v_i,v_j)\in E}}x_{ij}-\sum_{\substack{v_j\in\overline{S}\\(v_j,v_i)\in E}}x_{ji}\right)=\sum_{v_i\in S}\sum_{\substack{v_j\in\overline{S}\\(v_i,v_j)\in E}}c_{ij}=C\left(S,\overline{S}\right)$$

由于定理 5.9，任何可行流的流量 $\hat{W}$ 都满足 $\hat{W}\le C\left(S,\overline{S}\right)$，所以可行流 $F=\left\{x_{ij}\right\}$ 是最大流。□

定理 5.11. [最大流最小割定理] 设 $\left\{x_{ij}\right\}$ 为网络连通网络图 $G=\left(V,E,C\right)$ 的最大可行流，流量为 W，$\left\{S,\overline{S}\right\}$ 是分离 v_s 和 v_t 的最小割集，则有 $W=C\left(S,\overline{S}\right)$。且一个最大可行流至少对应一个最小割。

证明：根据定理 5.9 和 5.10，结论显然成立，而且定理 5.10 充分性的证明实际给出了根据最大流构造出最小割的方式。□

定义 5.29. 设 $\left\{S,\overline{S}\right\}$ 是连通有向网络 $G=\left(V,E,C\right)$ 中分离 v_s 和 v_t 的一个割集，则我们称 $\left\{S,\overline{S}\right\}$ 中的任意一个满足前向边流量可增，后向边流量可减条件的边是可增广边。

根据定理 5.9 到 5.11，我们可以采取“产生可行流，寻找可增广路，改进可行流”这一循环迭代的方式求解最大流问题。每轮迭代时，我们可以首先令 $S=\left\{v_s\right\}$，用 $\overline{S}$ 表示其补集，检查割集 $\left(S,\overline{S}\right)$，如果其中有可增广边，将相应边属于 $\overline{S}$ 的点移入 S，然后对新的割集继续找可增广边，直到 $v_t\in S$，我们就建立

把找到的边连起来构成一条可增广路，并据此改进可行流。如果割集$\left(S,\overline{S}\right)$不含可增广边，则当前可行流已是最大流。从（5.8）式可以看出各边容量皆为整数的整数容量网络如果从整数可行流开始操作，最终得到的是最大流一定是整数流。

基于该想法，我们可以设计如下的 Ford-Fulkerson 算法。其基本想法是：选定初始可行流，给v_t设为已标记待检查的顶点。然后循环选择已标记但未检查的顶点，对其邻点设为已标记待检查的顶点。如果已标记顶点没有未标记邻点，则标记该点为已检查点。如果能标记到v_t，则构造出一条可增广路，进而改进可行流。如果所有点都已检查，停止，构造最小割。否则，返回继续循环。

Ford-Fulkerson 算法没有限定网络存在多条可增广路时应该选择哪一条来改进可行流，因此有可能导致算法不收敛。为此，研究者提出采取广度优先搜索来查找未标记领点，此种算法被称为 Edmonds-Karp 算法。因为广度优先搜索总是选择一条边数最少的路。

算法 5.11. 最大流问题的 Edmonds-Karp 算法

定理 5.12. Edmonds-Karp 算法可以在有限时间内收敛到最优解。

证明：我们首先证明，对于所有的顶点$v_k \in S-\{v_s,v_t\}$，其在残差网络中的最短路距离$\delta_F\left(v_s,v_k\right)$随着每次流量的递增而单调递增。因为，通过 Edmonds-Karp 算法的执行找到一条最短的可增广路后，设置相应流并叠加到原网络中，则新的残差网络中此增广路必定不存在了，在新的残差网络中再次寻找的最短路一定更长。

我们假设 Edmonds-Karp 算法运行在网络$G=\left(V,E,C\right)$上。在残差网络$N=\left(V,E,C-F\right)$中，若增广路μ的残差容量是该条路上边$\left(v_i,v_j\right)$的残差容量，即$c_F\left(\mu\right)=c_{ij}$，则称边$\left(v_i,v_j\right)$在增广路$\mu$上的关键边（Critical Edge）。在沿一条增广路增加流后，处于该条路上的所有关键边都将从残差网络中消失。而且，任何一条增广路上都至少存在一条关键边。

接着，我们证明，对于所有$\left|E\right|$条边中的每一条边，成为关键边的次数最多为$\left|V\right|/2$次。

事实上，当边$\left(v_i,v_j\right)$第一次成为关键边时，有$\delta_{F'}\left(v_s,v_i\right)=\delta_F\left(v_s,v_i\right)+1$。一旦对流进行增加后，边$\left(v_i,v_j\right)$就从残差网络中消失。以后也不能重新出现在另一条增广路上，除非v_i到v_j的网络流减小到使得边$\left(v_i,v_j\right)$再次出现在增广路上。若当这一事件发生时，必然有当时的$\delta_{F''}\left(v_s,v_i\right)=\delta_{F''}\left(v_s,v_i\right)+1$。注意到

Edmonds-Karp 算法运行过程中，我们找到的流是不断增大的。不难推出

$$\delta_F(v_s,v_i)+2\le\delta_{F'''}(v_s,v_i)$$

因此，边(v_i,v_j)从成为关键边到下一次再成为关键边，从源点v_s到顶点v_i的距离至少增加 2，而从源点v_s到顶点v_i的初始距离至少为0。因为边(v_i,v_j)处于一条增广路上且该增广路在v_t结束，所以从源点v_s到顶点v_i的最短路上的中间顶点中不可能有顶点v_i、v_s或v_i。因此，从源点v_s到顶点v_i的最短路最多有$|V|-2$条边。所以，边(v_i,v_j)成为关键边的次数最多为$|V|/2$次。

最后，由于一共有$|E|$对顶点可以在一个残差网络中有一条边直接相连，因此在 Edmonds-Karp 算法执行的全部过程中，关键边的总数为$O(|V||E|)$。又因为，每一条增广路至少有一条关键边。而每次用广度优先搜索找到一条增广路的复杂度为$O(|E|)$。故而 Edmonds-Karp 所执行的流量递增操作的总次数为$O(|V||E|^2)$，结论得证。□

Edmonds-Karp 算法每次用广度优先搜索找到一条增广路的复杂度为$O(|E|)$。其实我们还可以利用顶点距离这样度量来进一步优化这一结果。

定义 5.30. 原网络$G=(V,E,C)$中的每条边在残差网络$N=(V,E,C-F)$中都化为一条或者两条边。在残差网络$N=(V,E,C-F)$中，把从源点到顶点v_k的含有边长最少的最短路所含边数，称为顶点v_k的层次（Level）。源点v_s的层次为0。我们从源点v_s出发，依次将其它顶点加入形成层次图（Level Graph）。

不同于 Ford-Fulkerson 算法，最短可增广路算法的基本思想是：每次在层次图中找一条含弧数最少的可增广路进行增广。其基本想法是：首先初始化容量网络和一个可行流。然后构建残差网络和层次图，若汇点不在层次图中，则算法结束。随后在层次图中不断用广度优先搜索寻找可增广路，直到层次图中没有可增广路为止；每次增广完毕，在层次图中要去掉因改进流量而导致饱和的边。然后返回构建新的残差网络和层次图，循环迭代。

在算法实现的时候，我们并不需要真正“构造”层次图，而只需要对每个顶点标记对应的层次，在增广的时候判断边是否满足这一约束条件即可。

算法 5.12. 最大流问题的最短可增广路算法

定理 5.13. 最短可增广路算法可以在有限时间内收敛到最优解。

证明：最短可增广路算法包括建立层次图和寻找增广路两部分。

根据定理 5.12 的证明，最短可增广路算法每次重新建立层次图，汇点所在的层次是严格递增的。而对于原网络 $G=(V,E,C)$，层次图最多有 $|V|$ 层，所以最多重新建立层次图 $|V|$ 次。而在建立每个层次图时，我们用广度优先搜索一次遍历即可得到。一次广度优先搜索的复杂度为 $O(|E|)$，所以建立层次图的总复杂度为 $O(|V||E|)$。

根据定理 5.12 的证明，在同一个层次图中，因为每一条增广路至少有一条关键边，所以增广路最多 $|E|$ 条。在层次图中使用广度优先搜索在层次图找到每一条增广路时，前进和回溯都最多 $|V|$ 次，合起来的复杂度是 $O(|V||E|)$。因此，对于最多 $|V|$ 个层次图，寻找增广路的总时间复杂度是 $O(|V|^2|E|)$。

综和建立层次图和寻找增广路两部分，最短可增广路算法的复杂度是 $O(|V|^2|E|)$，结论得证。□

此外，常见求解最大流问题的网络算法的还有 Dinic 算法等，本书从略。

例 5.8. 对于图 5.7 所示网络，请用 Edmonds-Karp 算法求解其的最大流。

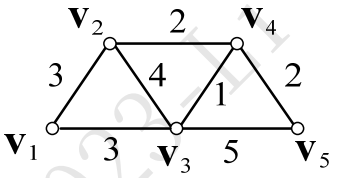


图 5.7　某最大流问题示意图。

解：用 Edmonds-Karp 算法第一轮迭代求出的可增广路是 $v_1 \to v_3 \to v_5$，沿该路能运送的最大货物量是 3，因此得到图 5.8（a）所示的可行流（下述图中，每条边的左边数字为该边容量，右边数字为该边当前流量）。

用 Edmonds-Karp 算法第二轮迭代求出的可增广路是 $v_1 \to v_2 \to v_3 \to v_5$，沿该路能运送的最大货物量是 2，因此得到图 5.8（b）所示的可行流。

用 Edmonds-Karp 算法第三轮迭代求出的可增广路是 $v_1 \to v_2 \to v_3 \to v_5$，沿该路能运送的最大货物量是 2，因此得到图 5.8（b）所示的可行流。

验证可知此时已经收敛到最优解。对应的最小割集 $\{S,\overline{S}\}$ 为 $S=\{v_1\}$，$\overline{S}=V-S$。

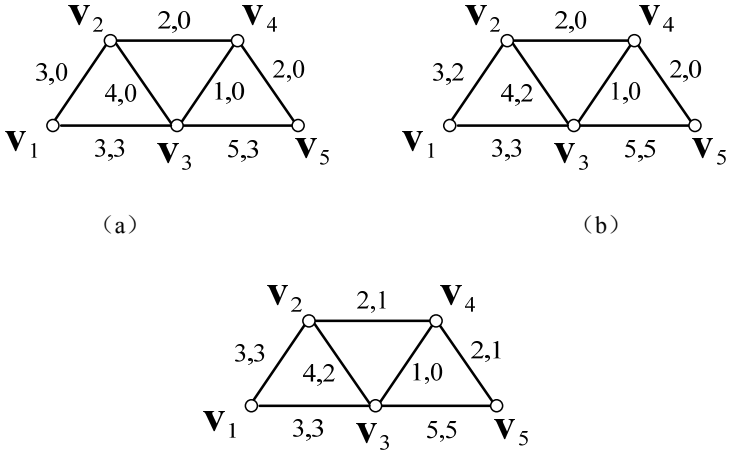


图 5.8 （a）Edmonds-Karp 算法第一轮迭代求出的结果；（b）Edmonds-Karp 算法第二轮迭代求出的结果；（c）Edmonds-Karp 算法第三轮迭代求出的结果。

5.5.2 最小费用流

定义 5.31. 给定连通有向网络 $G=(V,E)$ 及各边 $(v_i,v_j)\in E$ 的权 c_{ij} 和 d_{ij}，对于任意 $v_i,v_j\in V$，规定 $c_{ij}\geq 0$ 代表该边的容量，$d_{ij}\geq 0$ 代表单位货物通过该边的费用。仅有一个入次为 0 的顶点 $v_s\in V$，称为源点；仅有一个出次为 0 的顶点 $v_t\in V$，称为汇点。

若 W 是该网络允许的最大流总流量，假设针对每条边赋予一个流量值 x_{ij}，如果 $F=\{x_{ij}\}$ 为网络 G 的一个可行流，我们可以定义满足总流量为 $w\leq W$ 的最小费用流问题（Minimum-Cost Flow Problem）

$$\min_{x_{ij}\in\mathbb{R}} \sum_{(v_i,v_j)\in E} d_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{(v_i,v_j)\in E} x_{ij}-\sum_{(v_j,v_i)\in E} x_{ji}=\begin{cases} w & \text{if } i=s \\ 0 & \text{if } i\notin\{s,t\} \\ -w & \text{if } i=t \end{cases} \tag{5.9}$$

$$0\leq x_{ij}\leq c_{ij},\forall(v_i,v_j)\in E$$

如果我们在求解之初不清楚该网络的最大流流量值，可以先用上述最大流算法计算并检验需求的总流量 w 是否满足 $w \le W$。我们也可以按照如下的启发式算法来处理：沿用前述最大流问题 Ford-Fulkerson 算法的想法，我们可以每次都选择当前总费用最小的可增广路来尽可能多的运输货物，直到用完该可增广路的容量限制或者是达到总流量 w 需求，因此第 k 次沿可增广路 $\mu^{(k)}$ 运输的货物为

$$\delta^{(k)} = \min\left\{w - w^{(k)}, \left\{c_{ij} - x_{ij}, \forall\left(v_i, v_j\right) \in \mu^{(k)+}\right\}, \left\{x_{ij}, \forall\left(v_i, v_j\right) \in \mu^{(k)-}\right\}\right\} > 0 \tag{5.10}$$

接下来，我们还需要解决两件事情。第一，设计算法方便的找出当前总费用最小的可增广路；第二证明这样的启发式算法在停止时找到的就是最小费用流。

我们先将最小费用流问题（5.9）束简记为

$$\begin{aligned} \min_{x_{ij} \in \mathbb{R}} \quad & \sum_{(v_i, v_j) \in E} d_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{(v_i, v_j) \in E} P_{ij} x_{ij} = \mathbf{w}, -x_{ij} \le 0, x_{ij} - c_{ij} \le 0, \forall\left(v_i, v_j\right) \in E \end{aligned} \tag{5.11}$$

其中 $\mathbf{w} = \left[w, 0, ..., 0, -w\right]^T \in \mathbb{R}^{|V|}$。

进一步写出对应的 Lagrange 函数

$$L\left(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) = \sum_{(v_i, v_j) \in E} \left(d_{ij} + \mathbf{z}^T P_{ij} - \lambda_{ij} + \mu_{ij}\right) x_{ij} - \mathbf{z}^T \mathbf{w} - \sum_{(v_i, v_j) \in E} \theta_{ij} c_{ij}$$

其中 $\mathbf{z} \in \mathbb{R}^{|V|}$，$\boldsymbol{\lambda}, \boldsymbol{\theta} \in \mathbb{R}_+^{|V|}$ 为对偶变量。

因为最小费用流问题（5.9）是线性规划问题，满足强对偶性，对应 KKT 条件中的稳定性条件和互补松弛条件分别为

$$\begin{cases} d_{ij} + \mathbf{z}^T P_{ij} - \lambda_{ij} + \theta_{ij} = 0 \\ \lambda_{ij} x_{ij} = 0, \theta_{ij}\left(c_{ij} - x_{ij}\right) = 0 \end{cases}, \forall\left(v_i, v_j\right) \in E \tag{5.12}$$

我们可以进一步根据约束矩阵的特性，消去 λ_{ij}，将（5.12）改写为

$$\begin{cases} d_{ij} + \mathbf{z}^T P_{ij} + \theta_{ij} \ge 0 \\ \left(d_{ij} + z_i - z_j + \theta_{ij}\right) x_{ij} = 0, \theta_{ij}\left(c_{ij} - x_{ij}\right) = 0 \end{cases}, \forall\left(v_i, v_j\right) \in E \tag{5.13}$$

因此，最小费用流问题（5.9）的对偶问题为

$$\begin{aligned} \max_{z_i \in \mathbb{R}, \mu_{ij} \in \mathbb{R}} \quad & -\mathbf{z}^T \mathbf{w} - \sum_{(v_i, v_j) \in E} \theta_{ij} c_{ij} \\ \text{s.t.} \quad & d_{ij} + \mathbf{z}^T P_{ij} + \theta_{ij} \ge 0, \theta_{ij} \ge 0, \forall\left(v_i, v_j\right) \in E \end{aligned} \tag{5.14}$$

定理 5.14. 可行流 $F = \left\{x_{ij}\right\} = \mathbf{x} \in \mathbb{R}^{|E|}$ 是问题（5.9）最优解的充要条件是：存在满足如下等式约束的对偶变量 $\mathbf{z} \in \mathbb{R}^{|V|}$

$$\begin{cases} x_{ij}=0 & \forall\, d_{ij}+z_i-z_j>0 \\ x_{ij}=c_{ij} & \forall\, d_{ij}+z_i-z_j<0 \end{cases} \tag{5.15}$$

证明：我们首先证明充分性。如果对偶变量$\boldsymbol{\mu}\in\mathbb{R}_+^{|V|}$满足（5.15），我们可以构造如下的对偶变量$\boldsymbol{\lambda},\boldsymbol{\theta}\in\mathbb{R}_+^{|V|}$

$$\theta_{ij}=\max\left\{0,-\left(d_{ij}+z_i-z_j\right)\right\},\quad \lambda_{ij}=d_{ij}+\mathbf{z}^T P_{ij}+\theta_{ij}$$

使得（5.12）成立，显然，变量$\mathbf{x}\in\mathbb{R}^{|E|}$，和对偶变量$\mathbf{z}\in\mathbb{R}^{|V|}$，$\boldsymbol{\lambda},\boldsymbol{\theta}\in\mathbb{R}_+^{|V|}$共同满足 KKT 条件，因此，$\mathbf{x}\in\mathbb{R}^{|E|}$是最小费用流问题（5.9）最优解。

接着，我们证明必要性。如果$d_{ij}+z_i-z_j>0$，考虑到$\boldsymbol{\theta}\in\mathbb{R}_+^{|V|}$，则有$d_{ij}+z_i-z_j+\theta_{ij}>0$，根据互补松弛性条件，必然有$x_{ij}=0$。

如果$d_{ij}+z_i-z_j<0$，根据稳定性条件，$d_{ij}+\mathbf{z}^T P_{ij}+\theta_{ij}\geq 0$，则有$\theta_{ij}>0$，根据互补松弛性条件，必然有$x_{ij}=c_{ij}$。

因此，必要性成立。□

定义 5.32. 对于可行流$\mathbf{x}\in\mathbb{R}^{|E|}$，和变量$\mathbf{z}\in\mathbb{R}^{|V|}$，$\boldsymbol{\lambda},\boldsymbol{\mu}\in\mathbb{R}_+^{|V|}$，我们称$\sigma_{ij}(\mathbf{z})=d_{ij}+z_i-z_j$为边$(v_i,v_j)\in E$的简化成本，并把简化成本为零的可增广边称为可用边。

算法 5.13. 基于原问题对偶问题交替更新的最小费用流算法

Step 1. 令$\left\{x_{ij}^{(k)}=0\right\}$为初始可行流，$\left\{z_i^{(k)}=0\right\}$为初始对偶变量，$w^{(k)}=0$为初始总流量。$k:=1$，进入 Step 2。

Step 2. [寻找可增广路] 令$S=\{v_s\}$，用$\overline{S}$表示其补集。进入 Step 3。

Step 3. 如果割集$\{S,\overline{S}\}$中没有可增广边，原问题没有可行解，停止。

否则用$\hat{E}$表示其所有可增广边的集合，由下式决定η

$$\eta=\left|\sigma_{\hat{i}\hat{j}}(\mathbf{z})\right|=\min\left\{\left|\sigma_{ij}(\mathbf{z})\right|\ \text{s.t.}\ (v_i,v_j)\in\hat{E}\right\} \tag{5.16}$$

对所有$v_i\in\overline{S}$，更新其对应的对偶变量

$$z_i^{(k)}=z_i^{(k)}+\eta$$

如果此时存在多条可增广边，我们类似 Edmonds-Karp 算法进行广度优先搜索。假设我们最终确定针对割集$\{S,\overline{S}\}$的一条可用边$(v_{\hat{i}},v_{\hat{j}})$。令$v=\overline{S}\bigcap\{v_{\hat{i}},v_{\hat{j}}\}$，更新$S$和$\overline{S}$

$$S=S\bigcup\{v\},\quad \overline{S}=\overline{S}-\{v\}$$

如果$\overline{S}$是空集，则已产生一条由可用边组成的可增广链，进入 Step 4。

否则，返回 Step 3 继续迭代。

Step 4. 根据（5.10）式求出沿可增广路 $\mu^{(k)}$ 运输的货物 $\delta^{(k)}$，更新可行流

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + \delta^{(k)}, \forall \left(v_i, v_j\right) \in \mu^{(k)+}$$

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} - \delta^{(k)}, \forall \left(v_i, v_j\right) \in \mu^{(k)-}$$

$$x_{ij}^{(k+1)} = x_{ij}^{(k)}, \forall \left(v_i, v_j\right) \in E - \mu^{(k)+} - \mu^{(k)-}$$

更新总流量

$$w^{(k+1)} = w^{(k)} + \delta^{(k)}$$

如果 $w^{(k+1)} = w$，已经达到需求流量，停止。

否则，$k := k+1$，返回 Step 2。

定理 5.15. 如果最小费用流问题（5.9）存在有限的最优解，则算法 5.13 可以在有限时间内收敛到最优解。

证明：算法 5.13 的本质是原问题的可行解 $\left\{x_{ij}^{(k)}\right\}$ 和对偶问题的可行解 $\left\{z_i^{(k)}\right\}$ 交替迭代更新的过程。在迭代过程中，$x_{ij}^{(k)}$ 从 $x_{ij}^{(k)} = 0$ 开始，由（5.10）式保证了 $x_{ij}^{(k)} \le c_{ij}$ 和流量平衡条件，因此保证了 $\left\{x_{ij}^{(k)}\right\}$ 一直是原问题的可行解。而 $z_i^{(k)}$ 同样从 $z_{ij}^{(k)} = 0$ 开始，保持单调增长。而且根据定理 5.14 的证明，我们总可以构造合适的 $\boldsymbol{\lambda}, \boldsymbol{\theta} \in \mathbb{R}_+^{|V|}$ 使得 $\left(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\theta}\right)$ 一起构成对偶问题的可行解。

同时，算法 5.13 的 Step 2-Step 3 内层循环每次结束，选定一条可增广路更新流量 $\left\{x_{ij}^{(k)}\right\}$ 和对应的对偶变量 $\left\{z_i^{(k)}\right\}$ 之后，总能保证 KKT 条件（5.13）中的互补松弛条件成立。在达到需求流量停止时，一定满足 KKT 条件（5.13）中的稳定性条件。因此，算法 5.13 可以收敛到最优解。

同时，根据定理 5.12 的证明，整个网络的可增广路数目一定是有限的。因此，算法 5.13 每轮循环选择一条可增广路，且不会再重复选择该路，则一定会在有限时间内收敛。□

例 5.9. 请用算法 5.13 求解如下线性规划问题。

$$\min 2x_1+4x_2+x_3+7x_4+x_5+4x_6+5x_7+2x_8$$

$$\text{s.t.}\quad \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \\ -4 \end{pmatrix}$$

$$0 \le x_1 \le 4, 0 \le x_2 \le 1, 0 \le x_3 \le 2, 0 \le x_4 \le 1$$

$$0 \le x_5 \le 1, 0 \le x_6 \le 2, 0 \le x_7 \le 2, 0 \le x_8 \le 3$$

解：该问题等价于求解图 5.9（a）中所示网络中从 v_1 到节点 v_5，需求总流量为 4 的最小费用流问题（其中小括号中数字表示该边对应的价格和容量）。

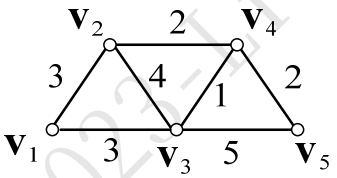


图 5.9 （a）对应的最小费用流问题示意图；（b）。

5.6　运输问题

定义 5.33. [Transportation Problem] 假设某种物品有 m 个产地，记为 $A_1, A_2, ..., A_m$，各产地产量分别是 $a_1, a_2, ..., a_m$；有 n 个销售地 $B_1, B_2, ..., B_n$，各销

售地销量分别是$b_1,b_2,\cdots,b_n$。假定从产地A_i向销地B_j运输单位物品的费用是$c_{ij}\geq 0$；怎样调运这些物品能使总费用最小。

如果是产销平衡，则

$$\sum_{i=1}^{m}a_i=\sum_{j=1}^{n}b_j=Q \tag{5.17}$$

如果总产量大于总销量，$\sum_{i=1}^{m}a_i>\sum_{j=1}^{n}b_j$。如果总产量小于总销量，$\sum_{i=1}^{m}a_i<\sum_{j=1}^{n}b_j$。

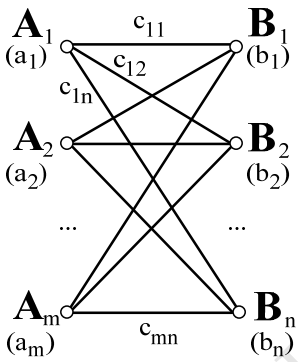


图 5.10　运输问题示意图。

我们可以通过增加新的顶点总产地A_0和新的顶点总产地B_0，并假定A_0点的产量为Q，$A_1,A_2,...,A_m$各点产量分别为0，B_0点的销售量为Q，$B_1,B_2,...,B_n$各点销售量分别为0。A_0点到$A_1,A_2,...,A_m$点均建立边，每条边的容量为$a_1,a_2,...,a_m$，运输单位物品费用为1。B_0点到$B_1,B_2,...,B_n$点均建立边，每条边的容量为$b_1,b_2,\cdots,b_n$，运输单位货物费用为1。而A_i点到B_j点均建立边，每条边的容量为$\min\{a_i,b_j\}$，运输单位货物费用为c_{ij}。这样，我们可以得到一个等价的最小费用流问题。

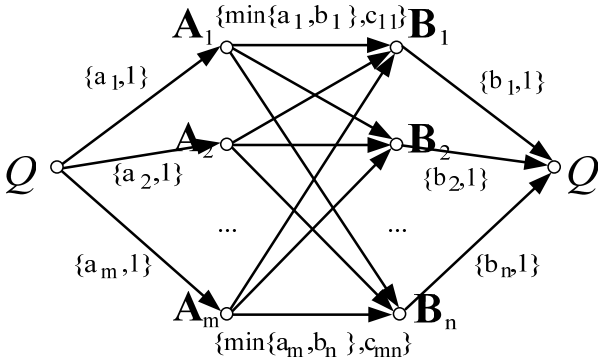


图 5.11　运输问题转化为最大流问题示意图，每条边旁大括号内显示该边的容量限制和单位货物的运输费用。

显然，我们也建立如下线性规划模型来直接求解该问题

$$\begin{aligned} \min_{x_{ij}\in\mathbb{R}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = a_i, \forall 1 \le i \le m \\ & -\sum_{i=1}^{m} x_{ij} = -b_j, \forall 1 \le j \le n \\ & x_{ij} \ge 0, \forall 1 \le i \le m, 1 \le j \le n \end{aligned} \tag{5.18}$$

也可以将上述约束简记为

$$\sum_{i=1}^{m}\sum_{j=1}^{n} P_{ij}x_{ij} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ -b_1 \\ \vdots \\ -b_n \end{pmatrix} \tag{5.19}$$

不难推出，该问题的对偶问题为

$$\begin{aligned} \max_{u_i,v_j\in\mathbb{R}} \quad & \sum_{j=1}^{n} b_j v_j - \sum_{i=1}^{m} a_i u_i \\ \text{s.t.} \quad & v_j - u_i \le c_{ij}, \forall 1 \le i \le m, \forall 1 \le j \le n \\ & u_i, v_j \ge 0, \forall 1 \le i \le m, \forall 1 \le j \le n \end{aligned} \tag{5.20}$$

其中对偶变量 u_i 的经济意义为在 A_i 产地单位货物的买入价格，对偶变量 v_j 的经

济意义为在 B_j 销售地的单位货物的卖出价格。显然对偶问题（5.20）的经济意义为：在价格差 $v_j - u_i$ 不超过沿对应边的实际运输成本 c_{ij} 的前提下，希望运输公司的利润最大化。

尽管可以用求解最小费用流问题的算法求解运输问题，但实践起来手续繁琐。也可以运用单纯型法求解线性规划问题（5.18），但由于存在大量的退化解，经常出现换基不能改进目标函数的情况，这些做法效率不高。为此，研究者将运输问题视为二分网络上的流量规划，提出了运输问题的支撑树算法。

定理 5.15. 对于产销平衡的运输问题，一定存在最优基本可行解。

证明：显然，由于 $x_{ij} \geq 0$，$c_{ij} \geq 0$，目标函数值必然非负。因此该线性规划问题有界。同时，令 $x_{ij} = \dfrac{a_i b_j}{Q}$，代入约束可以满足。根据定理 2.5 可知，产销平衡的运输问题一定存在最优基本可行解。□

定理 5.17. 对于产销平衡的运输问题，任一基本可行解对应运输网络中的一颗支撑树。

证明：注意到

$$\sum_{i=1}^{m} x_{in} = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n-1} x_{ij}\right) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n-1} x_{ij}$$

$$= \sum_{i=1}^{m} a_i - \sum_{j=1}^{n-1}\sum_{i=1}^{m} x_{ij} = Q - \sum_{j=1}^{n-1} b_j = b_n$$

则（5.18）中有一个多余的约束，因此，基变量维度应该是 $m+n-1$。

不妨计最后一个约束多余，我们将约束条件改写为

$$\sum_{j=1}^{n} x_{ij} = a_i,\ \forall 1 \leq i \leq m,\ \sum_{i=1}^{m} x_{ij} = b_j,\ \forall 1 \leq j \leq n-1$$

进一步将上述约束简记为

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \overline{P}_{ij} x_{ij} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ -b_1 \\ \vdots \\ -b_{n-1} \end{pmatrix} \tag{5.21}$$

其中，对于 $j < n$，$\overline{P}_{ij} = [0,\cdots,0,1,0,\cdots,0,-1,0,\cdots,0]^T \in R^{m+n-1}$ 中，只有第 i 个

元素为1，第$m+j$个元素为-1，其余元素为0。而$\overline{P}_{in}=[0,\cdots,1,\cdots,0]^T$，只有第$i$个元素为1，其余元素为0。

如果我们从mn个x_{ij}中选出$m+n-1$个x_{ij}构成基变量，则根据线性规划基变量的定义，这些x_{ij}对应的列向量$\overline{P}_{ij}$应该是线性无关的。注意到$\overline{P}_{ij}$的特殊形式，不难发现：如果我们将选定x_{ij}对应边映射到网络上，则这些边不能构成回路。注意到网络中有$m+n$个节点，$m+n-1$条不构成回路的边，根据定理 5.4，则必然在网络中形成一颗支撑树。□

需要指出的是，产销平衡的运输问题所对应网络上的支撑树个数为C_{mn}^{m+n-1}，对应的基本解个数也是C_{mn}^{m+n-1}。但这些基本解不一定都是基本可行解。因此，任意给定支撑树不一定能产生基本可行解。为了产生初始基本可行解，我们还需要考虑流量平衡方程。

算法 5.14. 运输问题的支撑树算法

Step 1. 假设我们将所有边按运输单位货物费用c_{ij}进行排序，进入 Step 2。

Step 2. [最小元素法产生一棵支撑树作为初始基本可行解] 如果已经找到合适的运输计划满足要求，在对应的网络上给出对应的树，则进入 Step 3。

否则，选取没有用完的边中运输单位货物费用最小的边，不妨记为$x_{i_0j_0}$对应的边。我们沿着该边运输尽可能多的货物，显然此时有$x_{i_0j_0}=\min\{a_{i_0},b_{j_0}\}$。

如果$x_{i_0j_0}=a_{i_0}$，则顶点A_{i_0}的货物已经运完，标记所有与A_{i_0}相连的边为已经用完，同时改写顶点B_{j_0}的货物需求量为$b_{j_0}-a_{i_0}$。

如果$x_{i_0j_0}=b_{j_0}$，则顶点B_{j_0}的需求已经满足，标记所有与B_{j_0}相连的边为已经用完，同时改写顶点A_{i_0}的货物存量为$b_{j_0}-a_{i_0}$。

返回 Step 2。

Step 3. [运用位势法计算所有非基变量的检验数] 从对偶问题角度来看，如果我们确定了一组基本可行解，则对于属于基矩阵的列向量$\overline{P}_{ij}$，应该满足

$$c_{ij}=[u_1\cdots u_m\ v_1\cdots v_{n-1}]\overline{P}_{ij}$$

如果我们假定$v_n=0$，则根据（5.19）和（5.21）的关系可以进一步推出

$$c_{ij}=[u_1\cdots u_m\ v_1\cdots v_{n-1}v_n]P_{ij} \tag{5.22}$$

根据（5.22），我们可以从v_n出发，在运输计划对应的树上，依次算出所有的对偶变量u_i和v_j的值。之后，我们再计算非基变量的检验数

$$\sigma_{ij}=c_{ij}-(u_i-v_j) \tag{5.23}$$

Step 4. 如果所有非基变量的检验数非负，已得最优解，停止。

否则，找出最小下标对应的非基变量（这里我们依然按 Bland 法则选择进出基以保证收敛性，所有可以进基的非基变量 x_{ij} 按下标排序，先按 i 的值从小到大排序，i 的值相同时再按 j 的值从小到大排序）。

这个非基变量对应的边一定和运输计划对应的树上某些边（对应当前的一些基变量）共同组成回路。不妨设这个非基变量对应的边为 (A_i, B_j)，对应的回路为 μ。我们定义回路 μ 中与边 (A_i, B_j) 相邻的边与边 (A_i, B_j) 的距离为 1，与边 (A_i, B_j) 相隔一条边的边距离为 2，并以此类推。

我们希望将边 (A_i, B_j) 上流量从 0 开始，尽可能的增大。为了满足顶点的供需约束条件，μ 中与边 (A_i, B_j) 距离为奇数的边的流量必须跟着等额减小，而 μ 中与边 (A_i, B_j) 距离为偶数的边的流量必须跟着等额增大。随着边 (A_i, B_j) 上流量的增大，最终必然会有回路中另外一条边的流量减为 0。这意味着那条边对应的基变量出基，而边 (A_i, B_j) 对应的非基变量进基。

假设我们不会遇到回路中有多条边的流量被减为 0 的情况，这意味着线性规划问题不会有退化的情况。上述操作就保证我们完成一次线性规划的进基出基操作。回到 Step 3。

定理 5.18. 运输问题的支撑树算法（5.14）可以在有限时间内收敛到最优解。

证明：因为该算法本质是线性规划的单纯型解法，因此必然在有限时间内收敛到最优解。□

例 5.10. 请用支撑树算法求解如下的线性规划问题

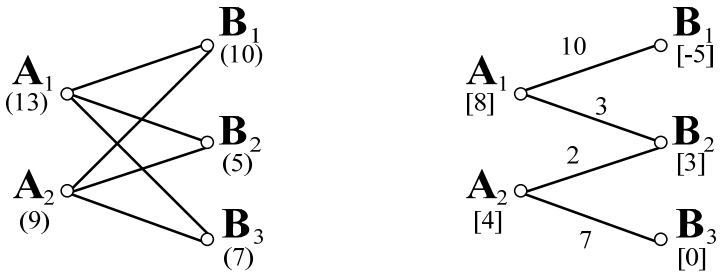
$$
\begin{aligned}
\min\ & 3x_1 + 11x_2 + 10x_3 + 8x_4 + 7x_5 + 4x_6 \\
\text{s.t.}\ & x_1 + x_2 + x_3 = 13 \\
& x_4 + x_5 + x_6 = 9 \\
& x_1 + x_4 = 10 \\
& x_2 + x_5 = 5 \\
& x_3 + x_6 = 7 \\
& x_i \geq 0, i = 1,2,3,4,5,6
\end{aligned}
$$

解：该问题等价于求解图 5.12（a）中所示网络的运输问题（各顶点旁圆括号内的数字表示该顶点的产量或需求量）。

用最小元素法产生初始可行解支撑树如图 5.12（b）所示，（各顶点旁房括号内的数字数字为相应对偶变量值，各边旁的数字为该边流量，其中我们令 $v_3 = 0$。

进一步计算非基变量的检验数可知 $\sigma_3 = 2$，$\sigma_4 = 9$，均大于零，满足最优性

条件。所以此运输问题初始可行解即最优解 $x_1=10$， $x_2=3$， $x_3=0$， $x_4=0$，$x_5=2$， $x_6=7$，最优值为 105。



（a）　　（b）

图 5.12 （a）对应的运输问题示意图；（b）用最小元素法产生初始可行解支撑树。

对于总产量大于总销量的运输问题，可以引入假想销售地 B_{n+1}，令点 B_{n+1} 的销售量为 $\sum_{i=1}^{m}a_i-\sum_{j=1}^{n}b_j$，所有产地 $A_1,A_2,...,A_m$ 到点 B_{n+1} 建立边，从产地 A_i 向销售地 B_{n+1} 运输单位物品的费用是 $c_{i(n+1)}=0$；$i=1,...,m$。即可得到一个等价的产销平衡的运输问题，可用上述算法等价求解。

对于总产量小于总销量的运输问题，可以引入假想产地 A_{n+1}，令点 A_{n+1} 的产量为 $\sum_{j=1}^{n}b_j-\sum_{i=1}^{m}a_i$，所有产地 $B_1,B_2,...,B_n$ 到点 A_{n+1} 建立边，从产地 A_{n+1} 到点 B_j 运输单位物品的费用是 $c_{(m+1)j}=0$；$j=1,...,n$。即可得到一个等价的产销平衡的运输问题，可用上述算法等价求解。

5.7　指派问题

定义 5.34. [Assignment Problem] 假设有 n 件事要 n 个人完成，每人做一件事，已知第 i 个人做第 j 件事的成本是 $c_{ij}\geq 0$，要确定人和事之间一对一的指派方案，使完成这 n 件事的总费用最小。其中 $\mathbf{C}=\left[c_{ij}\right]\in\mathbb{R}^{n\times n}$ 称为指派问题的系数矩阵。

显然，我们也建立如下线性规划模型来直接求解该问题

$$\begin{aligned}\min_{x_{ij}\in\mathbb{N}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij}=1,\ \ i=1,2,\cdots,n \\ & \sum_{i=1}^{n} x_{ij}=1,\ \ j=1,2,\cdots,n \\ & x_{ij}\in\{0,1\},\ \ \forall i,j\end{aligned} \tag{5.24}$$

其中 $x_{ij}=1$ 表示指派第 i 个人做第 j 件事，$x_{ij}=0$ 表示不指派第 i 个人做第 j 件事。

尽管可以用求解运输问题的算法或者单纯型法解线性规划问题求解指派问题，但由于存在大量的退化解，经常出现换基不能改进目标函数的情况，这些做法效率不高。为此，研究者提出了运输问题的匈牙利算法（Hungarian Algorithm）。

定理 5.19. 任取 $1\le k\le n$ 和任意实数 a，用 $\mathbf{C}_1$ 和 $\mathbf{C}_2$ 分别将 $\mathbf{C}$ 的第 k 行或第 k 列减去 a 以后得到的系数矩阵，则以 $\mathbf{C}$，$\mathbf{C}_1$ 或 $\mathbf{C}_2$ 为系数矩阵的指派问题的最优方案相同。

证明：注意到

$$\sum_{\substack{i=1\\ i\neq k}}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}+\sum_{j=1}^{n}(c_{kj}-a)x_{kj}=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}-a$$

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq k}}^{n} c_{ij}x_{ij}+\sum_{i=1}^{n}(c_{ik}-a)x_{ik}=\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}-a$$

变形后的优化问题目标函数差一个常数，约束相同，最优解显然和原问题的最优解相同。□

定理 5.20. 如果 $\mathbf{C}=\left[c_{ij}\right]\in\mathbb{R}^{n\times n}$ 的所有元素中没有负数，且存在 n 个行列号都互不相同的零元素（简称为独立零元素），则最优方案可以由独立零元素的位置确定。

证明：我们可以根据这组 n 个独立零元素位置 (i,j) 的集合 Θ 确定一个指派方案 $\left\{x_{ij}=1,(i,j)\in\Theta;x_{ij}=0,(i,j)\notin\Theta\right\}$。因为独立零元素的行列号都互不相同，所以 $\left\{x_{ij}=1,(i,j)\in\Theta;x_{ij}=0,(i,j)\notin\Theta\right\}$ 满足所有约束条件，而且使得对应的标准指派问题的最优目标值等于 0。注意到 $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}\ge 0$，则这 n 个独立零元素的位置所代表的指派方案 $\left\{x_{ij}=1,(i,j)\in\Theta;x_{ij}=0,(i,j)\notin\Theta\right\}$ 是最优解。□

因此，我们可以设计如下的算法：首先利用定理 5.19 产生零元素。然后对给定矩阵找到最大的独立零元素组。当最大的独立零元素组的零元素数目不够时增加独立零元素的数目。循环迭代直到找到足够的独立零元素。这其中的关键是如何寻找最大的独立零元素组。

我们用节点 α_i 表示系数矩阵 $\mathbf{C}$ 的第 i 行，结点 β_j 表示系数矩阵 $\mathbf{C}$ 的第 j 列，用边表示零元素的位置，可得一个二分图，所有边端点分属两个点集。而下述定理保证了求系数矩阵 $\mathbf{C}$ 最大独立零元素组则等价于求这个二分图的最大匹配数。

定义 5.35. 二分图 $G(L,R,E)$ 的对集是图中部分或全部边所组成的集合 $M \subseteq E$。如果 M 中任意两条边都没有公共端点，则称 M 是图 G 的一个匹配。最大匹配是所含边数最多的匹配集合。

M -饱和点指的是和匹配 M 的边关联的点。M -非饱和点指的是和匹配 M 的边不关联的点。

M -交错路指的是由属于和不属于 M 的边交错形成的路。

M -可增广路指的是起点和终点都是 M -非饱和点的交错路。

匈牙利树指的是起点是 M -非饱和点但终点不是的交错路。

一个顶点覆盖 $K \subseteq V$ 指的是 E 中每条边都有端点属于 K 的边的集合。最小顶点覆盖指的是所含端点数最少的覆盖。

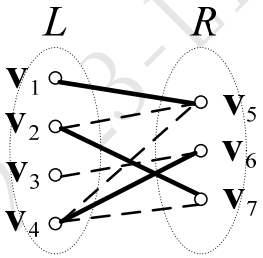


图 5.13 二分图示意图，其中 $\{(v_1,v_5),(v_2,v_7),(v_4,v_6)\}$ 构成一个匹配 M，同时也是该二分图的一个最大匹配，因为 R 中的所有顶点都有匹配边。$\{v_3\}$ 为 M -非饱和点，其余顶点都是 M -饱和点。因为 M 是最大匹配，所以该图中没有 M -可增广路，而 $(v_1,v_5)\to(v_5,v_2)\to(v_2,v_7)$ 为一条 M -交错路，$(v_3,v_6)\to(v_6,v_4)\to(v_4,v_5)\to(v_5,v_1)$ 为一棵匈牙利树。$\{v_5,v_6,v_7\}$ 是该二分图的一个最小覆盖。

定理 5.21. 二分图 $G(L,R,E)$ 中任意一个匹配 M 的边数都不会大于其覆盖的点数。M 是二分图 $G(L,R,E)$ 最大匹配的充要条件是 G 中没有 M -可增广路。

证明：注意到任意一个匹配中每条边的两个端点至少有一个在其覆盖中，则第一条结论显然。

我们用反证法证明第二条结论。如果我们在图G中找到了关于最大匹配M的一条M-可增广路，则其第一条边不属于M，第二条边属于M，如此交替。我们可以反过来，令其第一条边属于M，第二条边不属于M，如此交替。保持匹配M中与这条M-可增广路不相邻的边不变，则可以得到一个总边数增加1的新的匹配。这与M是二分图$G(L,R,E)$最大匹配的假设矛盾。因此，结论成立。□

定理 5.22. [Kőnig 定理] 二分图$G(L,R,E)$的最小顶点覆盖的端点数等于最大匹配的边数。

证明：设二部图$G(L,R,E)$中L和R是顶点集V的两个部分。假设M是图G的一个最大匹配，边数为$|M|$。因为M是匹配，则图G的任意一个顶点覆盖中都不可能存在一个顶点可以覆盖M的一条以上边的情况。所以，图G的最小顶点覆盖的端点数大于等于$|M|$。如果我们可以从匹配M构造出一个有$|M|$个顶点的顶点覆盖，那么它一定是一个最小覆盖。

以下给出从最大匹配构造最小顶点覆盖的方法。

设U是顶点集L中未被匹配的顶点的集合，或者等价的，顶点集L中所有M-非饱和点的集合。如果L中的所有顶点都在匹配M中，则U是空集。

设顶点集Z是顶点集U中的顶点通过M-交错路相连点的集合。

则顶点集$(L-Z)$覆盖所有不属于匈牙利树的边，顶点集$(R\cap Z)$覆盖所有属于匈牙利树的边。而且$(L-Z)$和$(R\cap Z)$的交集是空集。

我们接着证明顶点集$K=(L-Z)\cup(R\cap Z)$是一个顶点覆盖。

考虑边集E中的任意边e。此时有两种情况。

第一种情况，边e是一条匹配边。

因为$(L-Z)$中的每个顶点都是匹配的，所以K中的每个顶点都是一条匹配边的端点。而$(R\cap Z)$中的每个顶点也必须是匹配的，因为如果存在一条与未匹配的顶点交替的M-交错路，那么通过删除该路径上匹配的边，在原有位置上添加未匹配的边来改变匹配，会得到一个边数更大匹配，这和我们假设M是图G的一个最大匹配矛盾。进一步，任何匹配M中的边都不能存在两个端点都在K中。$|K|$等于所有匹配边的数目。因此，K覆盖了所有匹配边。

第二种情况，边e不是一条匹配边。

如果边e同时也不在M-交错路中，那么显然边e的左端点不能在M-交错路中，否则这条M-交错路可以通过添加边e变成M-可增广路，根据定理5.21这会导致矛盾。因此，边e的左端点一定属于$(L-Z)$，也属于K。

如果边e在M-交错路中，则边e的右端点一定属于$(R\cap Z)$，也属于K。

因此，K覆盖了所有非匹配边。

综合上述两种情况，K 是一个顶点覆盖。

又因为，$|M|=|K|$，K 就是我们寻找的一个最小顶点覆盖。□

据此，对于给定的匹配 M，我们可以在图 G 中寻找 M -可增广路来试图寻找更大的匹配。实际计算的时候，我们可以直接对系数矩阵进行操作。

算法 5.15. 指派问题的匈牙利算法

Step 1. 变换效率矩阵 $\mathbf{C}$，使每行每列至少有一个零元素。若某行/列已有零元素，则不用减。

其中行变换为：找出每行所有元素的最小值，该行各元素减去这个最小值；

列变换为：找出每列所有元素的最小值，该列各元素减去这个最小值.

变换后的矩阵记为 $\mathbf{B}$，进入 Step 2。

Step 2. 寻找零元素的最小覆盖：从含零元素最少的行或者列开始，圈出一个零元素，用○表示，然后划去该○所在的行和列中的其余零元素，用×表示，依次类推，若能得到 n 个○，则已得最优解，停止。否则，进入 Step 3。

Step 3. 对没有○的行打“√”；在已打“√”的行中，对×所在列打“√”；重复这两个步骤，直到再也找不到可以打“√”的行/列为止。

对没有打“√”的行画横线表示去掉这一行，对打“√”的列画横线表示去掉这一列，这样就得到能覆盖所有零元素的最小横线，进入 Step 4。

Step 4. 在未被直线覆盖的所有元素中找到其中最小值；然后在打“√”的所有行中减去这个最小值；而在打“√”的所有列中加上这个最小值，以保持原来的零元素不变，并避免出现负元素；得到新的系数矩阵 $\mathbf{B}'$。

如果已经找到 n 个独立零元素，则已得最优解，停止。否则，返回 Step 2。

如果目标函数是求最大的指派问题

$$\max_{x_{ij}\in\mathbb{N}} \quad \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij}=1,\ i=1,2,\cdots,n$$

$$\sum_{i=1}^{n} x_{ij}=1,\ j=1,2,\cdots,n$$

$$x_{ij}\in\{0,1\},\ \forall i,j$$

我们可以取一个足够大常数 $a>0$（譬如 $a=\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}$），考虑等价目标函数

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij} \Leftrightarrow \min\left(na - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij} \right)$$

$$= \min\left(a\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_{ij} \right) = \min \sum_{i=1}^{n}\sum_{j=1}^{n} \left(a - w_{ij} \right) x_{ij}$$

即可得到标准指派问题。

如果遇到人数和事情不等的指派问题，我们可以增加虚拟的人或事，并令相应成本系数为 0，转化为标准指派问题。如果遇到某人可能不能做某些事的指派问题，则可以令其人其事的对应边的成本系数为很大的正数，使得这些任务实际上不会被指派。

例 5.11. 如果 5 个工人安排分别完成 5 项工作，相对的收益矩阵为

$$\mathbf{W} = \begin{bmatrix} 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{bmatrix}$$

请用匈牙利算法求出总收益最大的指派计划。

解：首先，我们取 $a = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} = 49$，将其转化为如下系数矩阵 $\mathbf{C}$ 的标准指派问题

$$\mathbf{C} = \begin{bmatrix} 46 & 44 & 44 & 45 & 48 \\ 47 & 47 & 49 & 47 & 47 \\ 47 & 45 & 45 & 48 & 49 \\ 49 & 48 & 48 & 49 & 49 \\ 48 & 47 & 48 & 46 & 46 \end{bmatrix}$$

Step 1. 我们减去每行最小值可以得到

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & 1 & 4 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 3 & 4 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 1 & 2 & 0 & 0 \end{bmatrix}$$

Step 2. 我们从第一列开始，逐步得到

$$\begin{bmatrix} 2 & 0 & 0 & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & 0 & 0 & 3 & 4 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 1 & 2 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 0 & 0 & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & 0 & 0 & 3 & 4 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix} \rightarrow \begin{bmatrix} 2 & o & \times & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & \times & 0 & 3 & 4 \\ 1 & \times & 0 & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} 2 & o & \times & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & \times & 0 & 3 & 4 \\ 1 & \times & \times & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix}$$

Step 3. 我们依次推出

$$\begin{bmatrix} 2 & o & \times & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & \times & 0 & 3 & 4 \\ 1 & \times & \times & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix} \rightarrow \begin{bmatrix} 2 & o & \times & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & \times & 0 & 3 & 4 \\ 1 & \times & \times & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix} \rightarrow \begin{bmatrix} 2 & o & \times & 1 & 4 \\ o & \times & 2 & \times & \times \\ 2 & \times & 0 & 3 & 4 \\ 1 & \times & \times & 1 & 1 \\ 2 & 1 & 2 & o & \times \end{bmatrix}$$

$\sqrt{}$ 标记：第 5 行；$\sqrt{}$ 标记：第 2、3 列；$\sqrt{}$ 标记：第 1、4、5 行；$\sqrt{}$ 标记：第 2、3 列

得到最小覆盖：第二列和第三列竖线，第二行和第五行横线。

Step 3. 未被直线覆盖的所有元素中最小值为1，加减操作后得到新系数矩阵

$$\mathbf{B}' = \begin{bmatrix} 1 & o & \times & 0 & 3 \\ o & 1 & 3 & \times & \times \\ 1 & \times & 0 & 2 & 3 \\ 0 & \times & \times & 0 & 0 \\ 2 & 2 & 3 & o & \times \end{bmatrix}$$

显然其中不止一组最优解。其中一组最优解为 $x_{12} = x_{21} = x_{33} = x_{44} = x_{55} = 1$，原问题的最优收益为5+2+4+0+3=14 。

5.8　幺模矩阵

定义 5.35. 如果 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是整数矩阵，且 $\det(\mathbf{A}) = \pm 1$，则称矩阵 $\mathbf{A}$ 为幺模矩阵（Unimodular Matrix）。

定义 5.37. 如果 $\mathbf{A}\in\mathbb{R}^{m\times n}$ 是整数矩阵，且矩阵 $\mathbf{A}$ 的任意 $r\times r$ 非奇异子矩阵都是幺模矩阵，$1\le r\le \min\{m,n\}$，则称矩阵 $\mathbf{A}$ 为全幺模矩阵（Totally Unimodular Matrix）。显然，当我们取 $r=1$ 时，意味着全幺模矩阵 $\mathbf{A}$ 的每个元素值只能是 0，1 或 -1。

定理 5.23. 若 $\mathbf{A}\in\mathbb{R}^{m\times n}$ 是全幺模矩阵，对于任何整数向量 $\mathbf{b}\in\mathbb{R}^m$，线性不等式组 $\mathbf{Ax}\le\mathbf{b}$ 的所有基本解 $\mathbf{x}\in\mathbb{R}^n$ 都是整数向量。

证明：线性不等式组 $\mathbf{Ax}\le\mathbf{b}$ 的任一基本解 $\mathbf{x}_1$ 一定可以写为 $\mathbf{x}_1=\begin{bmatrix}\mathbf{A}_1^{-1}\mathbf{b}\\ \mathbf{0}\end{bmatrix}$ 的形式，其中 $\mathbf{A}_1\in\mathbb{R}^{m\times m}$ 是矩阵 $\mathbf{A}$ 中对应 $\mathbf{x}_1$ 的基矩阵。而 $\mathbf{A}_1^{-1}=\dfrac{\mathbf{A}_1^*}{\det(\mathbf{A}_1)}$，其中 $\mathbf{A}_1^*$ 是 $\mathbf{A}_1$ 的伴随矩阵。

如果矩阵 $\mathbf{A}$ 为全幺模矩阵，$\det(\mathbf{A}_1)=\pm1$，$\mathbf{A}_1^*$ 是整数矩阵，则基本解 $\mathbf{x}_1$ 必然是整数解。□

换而言之，多面体 $P=\{\mathbf{x}\mid\mathbf{Ax}\ge\mathbf{b},\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$ 的所有极点都是整数点。加上 $\mathbf{x}\ge\mathbf{0}$ 条件得到的多面体 $P=\{\mathbf{x}\mid\mathbf{Ax}\ge\mathbf{b},\mathbf{x}\ge\mathbf{0},\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{b}\in\mathbb{R}^m\}$ 的所有极点显然也都是整数点。

目前尚没有一个简单的充要条件来确定一个整数矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$ 是否全幺模矩阵。但我们有一些好用的全幺模矩阵充分性判定条件。

定理 5.24. 如果一个整数矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$ 的每个元素值只能是 0，1 或 -1，矩阵 $\mathbf{A}$ 的每列（或者每行）正好有一个元素为 1，另一个元素为 -1，则 $\mathbf{A}$ 是全幺模矩阵。因此，最大流问题的约束矩阵和运输问题的约束矩阵是全幺模矩阵。

证明：我们只证明每列的情况，每行可以完全类似证明。让我们考虑矩阵 $\mathbf{A}$ 的任意 $r\times r$ 子矩阵 $\mathbf{B}\in\mathbb{R}^{r\times r}$，$1\le r\le\min\{m,n\}$。此时存在三种情况。

第一种情况，如果矩阵 $\mathbf{B}$ 中存在元素全 0 的列，则 $\det(\mathbf{B})=0$，该子矩阵奇异。

第二种情况，如果矩阵 $\mathbf{B}$ 中每列都同时含有值为 1 和值为 -1 的两个元素，则 $\mathbf{1}^T\mathbf{B}=\mathbf{0}$，同样有 $\det(\mathbf{B})=0$，该子矩阵奇异。

第三种情况，矩阵 $\mathbf{B}$ 中至少有一列只含有一个非零元素。不妨记为第 j 列的第 i 个元素，则有

$$\det(\mathbf{B})=(-1)^{i+j}\mathbf{B}_{ij}\det(\mathbf{C})=(-1)^{i+j}(\pm1)\det(\mathbf{C})=\pm\det(\mathbf{C})$$

其中矩阵 $\mathbf{C}\in\mathbb{R}^{(r-1)\times(r-1)}$ 为矩阵 $\mathbf{B}$ 的 $(r-1)\times(r-1)$ 子矩阵。此时，我们可以用有限递降的方式证明 $\det(\mathbf{C})$ 只能是 0，1 或 -1。则 $\det(\mathbf{B})$ 只能是 0，1 或 -1。

综合上述情况，$\mathbf{A}$ 是全幺模矩阵。□

定理 5.25. 如果一个整数矩阵 $\mathbf{A}\in\mathbb{R}^{m\times n}$ 的每个元素值只能是 0 或 1，矩阵 $\mathbf{A}$ 的每列（或者每行）正好有两个元素为 1，则 $\mathbf{A}$ 是全幺模矩阵。因此，对于二分图矩阵的关联矩阵是全幺模矩阵。

证明：我们只证明每列的情况，每行可以完全类似证明。我们的证明非常类似定理 5.24 的证明。这次我们用数学归纳法的角度来描述。

首先最简单 $m=2$，$m=1$ 的情况很容易验证。假设对于矩阵 $\mathbf{A}$ 的任意 $(r-1)\times(r-1)$ 子矩阵 $\mathbf{C}\in\mathbb{R}^{(r-1)\times(r-1)}$，我们已经证明 $\det(\mathbf{C})$ 只能是 0，1 或 -1。接下来只需证明矩阵 $\mathbf{A}$ 的任意 $r\times r$ 子矩阵 $\mathbf{B}\in\mathbb{R}^{r\times r}$，满足 $\det(\mathbf{B})$ 只能是 0，1 或 -1。此时存在三种情况。

第一种情况，如果矩阵 $\mathbf{B}$ 中存在元素全 0 的列，则 $\det(\mathbf{B})=0$，该子矩阵奇异。

第二种情况，矩阵 $\mathbf{B}$ 中至少有一列只含有一个非零元素。不妨记为第 j 列的第 i 个元素，则有

$$\det(\mathbf{B})=(-1)^{i+j}\mathbf{B}_{ij}\det(\mathbf{C})=\pm\det(\mathbf{C})$$

此时，$\det(\mathbf{B})$ 只能是 0，1 或 -1。

第三种情况，如果矩阵 $\mathbf{B}$ 中每列都同时含有两个值为 1 的两个元素。由于对应的是二分图，我们可以将矩阵 $\mathbf{B}$ 中的所有行分别划归为两个集合 R_1 和 R_2；使得对于矩阵 $\mathbf{B}$ 的每一列，在 R_1 和 R_2 中正好各有一个值为 1 的元素。然后，我们可以在第一个属于集合 R_1 的行 r_1，加上所有其它属于集合 R_1 的行，再减去所有属于集合 R_2 的行，从而使得行 r_1 中所有元素变成 0，得到矩阵 $\overline{\mathbf{B}}$。显然，矩阵进行这样的操作不改变行列式的值，所以 $\det(\mathbf{B})=\det(\overline{\mathbf{B}})=0$。

综合上述情况，根据数学归纳法，$\mathbf{A}$ 是全幺模矩阵。□

定理 5.25. 用全幺模矩阵的性质证明 Kőnig 定理。

证明：我们可以将二分图 $G(L,R,E)$ 的最大匹配建模为如下的整数规划问题

$$\begin{aligned}\max_{x_i\in\mathbb{N}}\quad & \sum_{i=1}^{|E|}x_i\\ \text{s.t.}\quad & \sum_{i\sim v}x_i\le 1,\ \ \forall v\in V\\ & x_i\in\{0,1\},\ \ i=1,...,|E|\end{aligned}\tag{5.25}$$

其中 $x_i=1$ 表示第 i 条边是一条匹配边，$x_i=0$ 表示第 i 条边不是一条匹配边，不等式表示对所有和点 v 相邻的边中最多只能有一条匹配边。

显然，根据定理 5.26，该问题的约束矩阵为全幺模矩阵，我们可以将松弛为线性规划问题而找到的最优解一致。进一步将该线性规划问题简写成矩阵形式

$$\begin{aligned} \max_{x_i\in\mathbb{N}} \quad & \sum_{i=1}^{|E|} x_i \\ \text{s.t.} \quad & \mathbf{Ax}\le\mathbf{1} \\ & x_i\in\{0,1\},\ \ i=1,...,|E| \end{aligned} \tag{5.26}$$

我们可以得到该线性规划问题的对偶问题为

$$\begin{aligned} \min_{u_j\in\mathbb{N}} \quad & \sum_{j=1}^{|V|} u_j \\ \text{s.t.} \quad & \mathbf{A}^T\mathbf{u}\ge\mathbf{1} \\ & u_j\ge 0,\ \ i=1,...,|V| \end{aligned} \tag{5.27}$$

这个对偶问题的约束矩阵仍然为全幺模矩阵，其最优解显然和二分图 $G(L,R,E)$ 的最小覆盖建模成的整数规划问题的最优解一致

$$\begin{aligned} \min_{u_j\in\mathbb{N}} \quad & \sum_{j=1}^{|V|} u_j \\ \text{s.t.} \quad & u_k+u_j\ge 1,\ \text{if } (k,j)\in E \\ & u_j\in\{0,1\},\ \ i=1,...,|V| \end{aligned} \tag{5.28}$$

这里 $u_j=1$ 表示第 j 个顶点属于覆盖，$u_j=0$ 表示第 j 个顶点不属于覆盖。

根据强对偶性，原问题和对偶问题的最优值相等，最小顶点覆盖的端点数等于最大匹配的边数。□

基于全幺模矩阵的性质，我们可以将算法 5.13 修改为如下形式。

算法 5.15. 使用 Dijkstra 最短路算法的最小费用流算法

Step 1. 令 $\{x_{ij}^{(k)}=0\}$ 为初始可行流，$\{z_i^{(k)}=0\}$ 为初始对偶变量，$w^{(k)}=0$ 为初始总流量。$k:=1$，进入 Step 2。

Step 2. 利用当前流量和简化成本构造长度网络：保留所有可增边；把所有可减边反向；各边长度取简约成本 $\sigma_{ij}(\mathbf{z})=d_{ij}+z_i-z_j$ 的绝对值。用 Dijkstra 算法找到长度网络中，从源点 v_s 到各顶点的最短路并记住最短路长度 $\rho_i^{(k)}$，$i=1,...,|V|$。如果到某个顶点没有通路，令对应的 $\rho_i^{(k)}=+\infty$。

如果源点 v_s 到汇点的最短路长度 $\rho_t^{(k)}=+\infty$，原问题没有可行解，停止。

否则，对所有 v_i，更新其对应的对偶变量

$$z_i^{(k)}=z_i^{(k)}+\min\{\rho_i^{(k)},\rho_t^{(k)}\}$$

进入 Step 3。

Step 3. 根据（5.10）式求出从源点 v_s 到汇点 v_t 的最短路 $\mu^{(k)}$ 可以运输的货物

$\delta^{(k)}$，更新可行流

$$x_{ij}^{(k+1)} = x_{ij}^{(k)} + \delta^{(k)}, \forall \left(v_i, v_j\right) \in \mu^{(k)+}$$
$$x_{ij}^{(k+1)} = x_{ij}^{(k)} - \delta^{(k)}, \forall \left(v_i, v_j\right) \in \mu^{(k)-}$$
$$x_{ij}^{(k+1)} = x_{ij}^{(k)}, \forall \left(v_i, v_j\right) \in E - \mu^{(k)+} - \mu^{(k)-}$$

更新总流量

$$w^{(k+1)} = w^{(k)} + \delta^{(k)}$$

如果 $w^{(k+1)} = w$，已经达到需求流量，停止。

否则，$k := k+1$，返回 Step 2。

定理 5.27. 算法 5.16 和算法 5.13 每轮迭代时找到的从源点 v_s 到汇点的路 $\mu^{(k)}$ 是一样的。

证明：算法 5.16 每次构造的长度路网中的最短路问题可以写为

$$\min_{y_{ij} \in \mathbb{R}} \quad \sum_{\left(v_i, v_j\right) \in E} \sigma_{ij}^{(k)}(\mathbf{z}) y_{ij}$$
$$\text{s.t.} \quad \sum_{\left(v_i, v_j\right) \in E} y_{ij} - \sum_{\left(v_j, v_i\right) \in E} y_{ji} = \begin{cases} 1 & \text{if} \quad i = s \\ 0 & \text{if} \quad i \notin \{s, t\} \\ -1 & \text{if} \quad i = t \end{cases} \tag{5.29}$$
$$y_{ij} \in \{0,1\}, \forall \left(v_i, v_j\right) \in E$$

该问题的约束矩阵显然是全幺模矩阵，则其的最优解和下述问题的最优解是一样的。

$$\min_{y_{ij} \in \mathbb{R}} \quad \sum_{\left(v_i, v_j\right) \in E} \sigma_{ij}^{(k)}(\mathbf{z}) y_{ij}$$
$$\text{s.t.} \quad \sum_{\left(v_i, v_j\right) \in E} y_{ij} - \sum_{\left(v_j, v_i\right) \in E} y_{ji} = \begin{cases} 1 & \text{if} \quad i = s \\ 0 & \text{if} \quad i \notin \{s, t\} \\ -1 & \text{if} \quad i = t \end{cases} \tag{5.30}$$
$$0 \le y_{ij} \le 1, \forall \left(v_i, v_j\right) \in E$$

不难看出，使用算法 5.9 或 5.10 的网络单源最短路 Dijkstra 算法求解整数规划问题（5.29）的求解过程中，每次必然会有一个顶点到起点的最短距离被确定，并在后面的迭代中不会被改变。这些顶点被依次确定顺序，和算法 5.13 的内层循环寻找可增广路的过程中，使用原问题的可行解 $\left\{x_{ij}^{(k)}\right\}$ 和对偶问题的可行解 $\left\{z_i^{(k)}\right\}$ 交替迭代更新求解线性规划问题（5.30）时，所确定的顶点加入到可增广路的顺序是一样的。□

算法 5.17. 使用 Bellman-Ford-Moore 最短路算法的最小费用流算法

Step 1. 令 $\left\{x_{ij}^{(k)}=0\right\}$ 为初始可行流， $w^{(k)}=0$ 为初始总流量。 $k:=1$，进入 Step 2。

Step 2. 利用当前流量和简化成本构造费用网络：保留所有可增边；把所有可减边反向；用费用 d_{ij} 作为可增边的长度，用费用的负数 $-d_{ij}$ 作为可减边的长度。用 Bellman-Ford-Moore 算法找到费用网络中，从源点 v_s 到各顶点的最短路并记住最短路长度 $\rho_i^{(k)}$，$i=1,...,|V|$。如果到某个顶点没有通路，令对应的 $\rho_i^{(k)}=+\infty$。

如果源点 v_s 到汇点的最短路长度 $\rho_t^{(k)}=+\infty$，原问题没有可行解，停止。

否则，进入 Step 3。

Step 3. 根据（5.10）式求出从源点 v_s 到汇点 v_t 的最短路 $\mu^{(k)}$ 可以运输的货物 $\delta^{(k)}$，更新可行流

$$x_{ij}^{(k+1)}=x_{ij}^{(k)}+\delta^{(k)},\forall\left(v_i,v_j\right)\in\mu^{(k)+}$$

$$x_{ij}^{(k+1)}=x_{ij}^{(k)}-\delta^{(k)},\forall\left(v_i,v_j\right)\in\mu^{(k)-}$$

$$x_{ij}^{(k+1)}=x_{ij}^{(k)},\forall\left(v_i,v_j\right)\in E-\mu^{(k)+}-\mu^{(k)-}$$

更新总流量

$$w^{(k+1)}=w^{(k)}+\delta^{(k)}$$

如果 $w^{(k+1)}=w$，已经达到需求流量，停止。

否则， $k:=k+1$，返回 Step 2。

定理 5.28. 算法 5.17 和算法 5.16 每轮迭代时找到的从源点 v_s 到汇点 v_t 的可增广路 $\mu^{(k)}$ 是一样的。

证明：不难看出，长度网络中任意两点间的任意一条初级路的长度，和费用网络中任意两点间的任意一条初级路的费用，两者数值是完全一样的。如果不存在多条最短路的话，不同的最短路算法找到的最短路是同样的。即使存在多条最短路，我们也可以附加其它条件，例如类似 Bland 法则的最小下标选取方式来保证两个算法得到同样的可增广路。□

由于算法 5.17 不涉及简化成本，所以无需计算对偶变量。但其构造的费用网络含有负权值，不能用 Dijkstra 算法求最短路，寻求最短路的计算量要稍大一些。

5.9 作业题

5.1 请设计算法找出一个不联通图的各个连通分支。

5.2 请设计算法找出网络中的最大支撑树。

5.3 请设计算法找出网络中给定起点和终点之间总权最大的路。

5.4 [k -Shorted Path Problem] 假设一个网络中从给定起点到给定终点的所有路按总权值从小到大排列，请设计算法找出排名前k的k条路。

5.5 试证明 Floyd-Warshall 算法、Bellman-Ford-Moore 算法和 Dijkstra 算法的可性性。

5.6 设$\mathbf{A}\in\mathbb{R}^{m\times n}$是全幺模矩阵，请证明对矩阵$\mathbf{A}$执行如下任一操作将不改变全幺模性：

（1）行置换或列置换；

（2）转置；

（3）任一行或列乘以-1；

（4）增添只有一个元素1或-1，其它元素为0的行或列。

5.7 对于一般性的不含负权值的网络的最短路问题，请利用全幺模矩阵设计原问题可行解和对偶问题可行解交替迭代的求解算法，并证明其和 Dijkstra 算法的等价性。

5.10　参考文献

[1] A. Bondy, U.S.R. Murty, *Graph Theory*, 2nd edition, Springer, 2008.

[2] 徐俊明, 图论及其应用, 4th edition, 中国科学技术大学出版社, 2019.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 4th edition, The MIT Press, 2022.

[4] P. Erdos, A. Renyi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290-297, 1959.

[5] A.-L. Barabási, R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509-512, 1999.

[6] J. Edmonds, R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of ACM*, vol. 19, no. 2, pp. 248-264, 1972.

[7] Y. Dinitz, "Dinitz' algorithm: The original version and Even's version," *Theoretical Computer Science*, vol. 3895, pp. 218-240. Springer, Heidelberg, 2005.

[8] P. Lammich, S. R. Sefidgar, "Formalizing the Edmonds-Karp algorithm," *Proceedings of International Conference on Interactive Theorem Proving*, pp. 219-234, 2015.

[9] P. Camion, "Characterization of totally unimodular matrices," *Proceedings of the American Mathematical Society*, vol. 15, no. 5, pp. 1068-1073, 1965.

[10] A. F. Veinott, Jr., G. B. Dantzig, "Integral extreme points," *SIAM Review*, vol. 10, no. 3, pp. 371-372, 1968.

[11] F. G. Commoner, "A sufficient condition for a matrix to be totally unimodular," *Networks*, vol. 3, no. 4, pp. 351-365, 1973.

[12] A. Tamir, "On totally unimodular matrices," *Networks*, vol. 6, no. 4, pp. 373-382, 1975.

第 6 章 动态规划

6.1 引言

动态规划（Dynamic Programming，DP）是求解多阶段决策过程（Multi-Stage Decision Process）的代表性方法。20 世纪 50 年代初，Bellman 等研究者提出了多阶段决策过程的最优性原理，从而创立了动态规划 [1]-[2]。动态规划的应用极其广泛，包括经济、工业生产、军事等重要领域。最近数年，从动态规划进一步延伸开来的近似动态规划（Approximate Dynamic Programming）、强化学习（Reinforcement Learning）等方法在人工智能等重要领域发挥着越来越大的作用[3]-[9]。

本章先介绍确定性动态规划的基本概念及动态规划问题的建模和求解技巧；之后介绍动态数规划问题中广泛使用的值迭代法和策略迭代法；最后介绍 Markov 决策过程规划问题。

6.2 动态规划基础

6.2.1 多阶段决策问题和最优性原理

定义 6.1. [确定性多阶段决策问题] 假设一个针对特定系统的决策过程可以划分成若干阶段。第 t 个阶段该系统的所有可用信息被称为状态（State），用 $s_t \in S_t$ 表示，当前阶段所有可能的状态集合为 $S_t \subset S$，S 表示所有阶段所有可能的状态构成的集合。考虑当前状态 s_t 之后我们可以进行一次决策并采取行动（Action），用 $a_t = f\left(s_t\right) \in A_{s_t} \subset A$，其中 $f\left(\bullet\right): S \to A$ 表示我们的决策函数，也称策略（Policy）。状态 s_t 所有可能的动作集合为 $A_{s_t} \subset A$，A 表示所有可能的动作构成的集合。做出该行动后，系统第 $t+1$ 个阶段状态变为 $s_{t+1} = T\left(s_t, a_t\right) \in S$，

$T(\cdot,\cdot): S \times A \to S$ 表示系统转移函数（Transition Function）。我们用 $r_t = g(s_t, a_t)$ 表示在第 t 个阶段获得收益（Reward），$g(\cdot,\cdot): S \times A \to R$ 表示系统收益函数（Reward Function），R 表示所有可能的收益构成的集合。一般来说，我们希望找到一种最优策略 $f^*(\bullet)$，使得从起始状态到最终状态的长期收益最高，$\max_f \sum_t r_t$。

需要指出的是，系统转移 $s_{t+1} = T(s_t, a_t)$ 的假设隐含了 Markov 性，也称为无后效性（Memoryless）。即未来状态仅由当前状态决定，而与过去状态无关。

一些应用问题是考虑每个阶段都可能遭受某些损失，并且求解长期损失最小。我们可以将此类问题的损失取负值后记为收益，同样可以转化为求长期收益最高的问题，参加下述例 6.1-6.2，例 6.5-6.6 的求解。

还有一些应用问题是考虑每个阶段的收益连乘起来的长期收益最高，此类问题也可以对常规累加收益问题的动态规划解法稍作调整之后进行求解，这里不再赘述。

定义 6.2. 我们将一系列状态—动作—收益序列称为一个轨迹（Trajectory），通常写为 $s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \ldots$。一般来说，执行不同的策略，得到的轨迹是不同的。

实际应用时，不同问题的要素不尽相同。根据要素的差异，确定性多阶段决策问题可以进一步分成不同类型：

根据阶段数可分为：有限阶段决策问题，其阶段数为有限值；无限阶段决策问题，其阶段数为无穷大，决策过程可无限持续下去；

根据变量取值情况可分为：连续多阶段决策问题,决策变量和状态变量取连续变化的实数；离散多阶段决策问题，决策变量和状态变量取有限的数值；

根据阶段个数是否明确可分为：定期多阶段决策问题，其阶段数是明确的，不受决策的影响；不定期多阶段决策问题，其阶段数是不确定的，不同的决策下阶段数可以不同。

我们可以针对不同类型的动态规划应用问题，设计不同的求解方法。

例 6.1. 针对图 6.1 给出的节点，可选的路径和相应长度关系，利用穷举法找出从节点 A 至节点 I 的最短路径铺设输油管道。

解：不难发现第一阶段可能的状态为 $S_t = \{A\}$，第一阶段可能选择的动作为 $A_1 = \{A \to B, A \to C\}$。可能状态转移有 $B = T(A, A \to B)$，$C = T(A, A \to C)$。可能的收益包括 $g(A, A \to C) = -6$ 等。

第二阶段可能的状态为 $S_2 = \{B, C\}$，第二阶段可能选择的动作为 $A_2 = \{B \to D, B \to E, C \to E, C \to F\}$。可能状态转移包括 $D = T(B, B \to D)$，$E = T(B, B \to E)$ 等。可能的收益包括 $g(B, B \to D) = -3$ 等。

第三阶段可能的状态为 $S_3 = \{D, E, F\}$，第三阶段可能选择的动作为 $A_3 = \{D \to G, E \to G, E \to H, F \to H\}$。可能状态转移包括 $G = T(E, E \to G)$

等。可能的收益包括 $g(E,E\to H)=-2$ 等。

第四阶段可能的状态为 $S_4=\{G,H\}$，第三阶段可能选择的动作为 $A_4=\{G\to,H\to I\}$。可能状态转移包括 $I=T(G,G\to I)$ 等。可能的收益包括 $g(G,G\to I)=-1$ 等。

第五阶段可能的状态为 $S_5=\{I\}$，已经到达终态，停止。

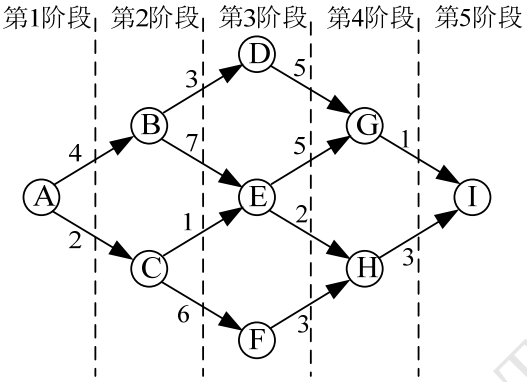


图 6.1　最短油管铺设问题示意图：节点，可选的路径和相应长度。

很显然，我们可以穷举出所有可能的 6 种策略并计算出相应的收益：

策略 $\{A\to B,B\to D,D\to G,G\to I\}$ 的总收益为 $-4-3-5-1=-13$。

策略 $\{A\to B,B\to E,E\to G,G\to I\}$ 的总收益为 $-4-7-5-1=-17$。

策略 $\{A\to B,B\to E,E\to H,H\to I\}$ 的总收益为 $-4-7-2-3=-16$。

策略 $\{A\to C,C\to E,E\to H,H\to I\}$ 的总收益为 $-2-1-2-3=-8$。

策略 $\{A\to C,C\to E,E\to G,G\to I\}$ 的总收益为 $-2-1-5-1=-9$。

策略 $\{A\to C,C\to F,F\to H,H\to I\}$ 的总收益为 $-2-6-3-3=-14$。

显然最优策略是 $\{A\to C,C\to E,E\to H,H\to I\}$。

如果我们是求从节点 C 至节点 I 的最短路径铺设输油管道，3 种可能的策略及其相应的收益为：

策略 $\{C\to E,E\to H,H\to I\}$ 的总收益为 $-1-2-3=-6$。

策略 $\{C\to E,E\to G,G\to I\}$ 的总收益为 $-1-5-1=-7$。

策略 $\{C\to F,F\to H,H\to I\}$ 的总收益为 $-6-3-3=-12$。

则最优策略是 $\{C\to E,E\to H,H\to I\}$

我们不难发现，从节点 C 至节点 I 的最短路径铺设输油管道这一子问题的最优策略是从节点 A 至节点 I 的最短路径铺设输油管道这一问题的最优策略的一部分。

类似的，如果我们是求从节点 E 至节点 I 的最短路径铺设输油管道，则该子问题的最优策略是 $\{E\to H,H\to I\}$，也是从节点 A 至节点 I 的最短路径铺设输

油管道这一问题的最优策略的一部分。

从上面的例子可以进一步扩展得出如下重要定理。

定理 6.1. [Bellman 最优性原理] “一个多阶段决策过程的最优策略满足，无论其初始状态及其初始决策如何，其今后诸策略对以第一个决策所形成的状态作为初始状态的过程而言，必须构成最优策略。”简言之，一个最优策略的子策略，对于它的初态和终态而言也必是最优的。

证明：假设该多阶段决策过程的初始状态是 s_1，终态是 s_{n+1}，执行最优策略 f^* 所采取的动作序列是 $\{a_1^*...,a_m^*,...,a_n^*\}$。若执行最优策略至第 m 个阶段的状态是 s_m，则我们可以将最优策略分为策略 f_1^* 和策略 f_2^* 两个部分，执行策略 f_1^* 所采取的动作序列是 $\{a_1^*...,a_m^*\}$，则策略 f_1^* 是初始状态是 s_1，终态是 s_m 的多阶段决策过程的最优策略。执行策略 f_2^* 所采取的动作序列是 $\{a_{m+1}^*...,a_n^*\}$，策略 f_2^* 是初始状态是 s_m，终态是 s_{n+1} 的多阶段决策过程的最优策略。

我们使用反证法证明，如果存在一个策略 q_1^* 是初始状态是 s_1，终态是 s_m 的多阶段决策过程的最优策略，在前 k 个阶段执行策略 q_1^* 的动作序列是 $\{q_1^*...,q_m^*\}$，前 k 个阶段总收益大于执行策略 f_1^* 的总收益。则我们可以将策略 q_1^* 和策略 f_2^* 组合成一个新的针对原多阶段决策过程的策略 q^*，其采取的行动序列是 $\{q_1^*...,q_m^*,a_{m+1}^*...,a_n^*\}$，而且其总收益要大于执行策略 f^* 的总收益。这与策略 f^* 是最优策略相矛盾。

类似的，我们也可以证明策略 f_2^* 是初始状态是 s_m，终态是 s_{n+1} 的多阶段决策过程的最优策略。□

根据这一性质，我们更加高效的进行多阶段决策过程的最优策略求解。

例 6.2. 针对图 4.1 给出的节点，可选的路径和相应长度关系，利用动态规划法找出从节点 A 至节点 I 的最短路径铺设输油管道。

解：我们将该问题扩展为找出除节点 I 以外的所有节点到节点 I 的最短路径，并用逆推法求解。

显然看第 4 阶段，节点 G 到节点 I 的最优策略是 $\{G \to I\}$，收益是 -1。节点 H 到节点 I 的最优策略是 $\{H \to I\}$，收益是 -3，参加图 6.2（a）。

接着看第 3 阶段，节点 D 到节点 I 的最优策略是 $\{D \to G, G \to I\}$，经过 1 次加法计算获知收益是 $-1-5=-6$。节点 F 到节点 I 的最优策略是 $\{F \to H, H \to I\}$，经过 1 次加法计算获知收益是 $-3-3=-6$。节点 E 到节点 I 的有两个可能的策略，经过 2 次加法和 1 次比较，我们找到最优策略是 $\{E \to H, H \to I\}$，收益是 $-2-3=-5$，参加图 6.2（b）。

然后看第 2 阶段，根据上述推理的结论，我们只需计算和比较：节点 B 到节

点 D 的收益加上节点 D 到节点 I 的最优策略收益，以及节点 B 到节点 E 的收益加上节点 E 到节点 I 的最优策略收益。经过 2 次加法和 1 次比较，我们找到最优策略是$\{B\to D, D\to G, G\to I\}$，收益是$-3-6=-9$。类似的，经过 2 次加法和 1 次比较，可找到节点 C 到节点 I 的最优策略是$\{C\to E, E\to H, H\to I\}$，收益是$-1-5=-6$，参加图 6.2（c）。

最后看第 2 阶段，根据上述推理的结论，我们只需计算和比较：节点 A 到节点 B 的收益加上节点 B 到节点 I 的最优策略收益，以及节点 A 到节点 C 的收益加上节点 C 到节点 I 的最优策略收益。经过 2 次加法和 1 次比较，我们找到最优策略是$\{A\to C, C\to E, E\to H, H\to I\}$，收益是$-2-6=-8$，参加图 6.2（d）。

使用穷举法解题时，我们进行了 24 次加法和 5 次比较计算。而使用动态规划解题时，我们进行了 10 次加法和 4 次比较计算，有效避免了重复计算，显然效率要更高一些。

我们也可以将该问题扩展为找出节点 A 到除节点 A 以外的所有节点到的最短路径，并用顺推法求解，结果类似，不再赘述。

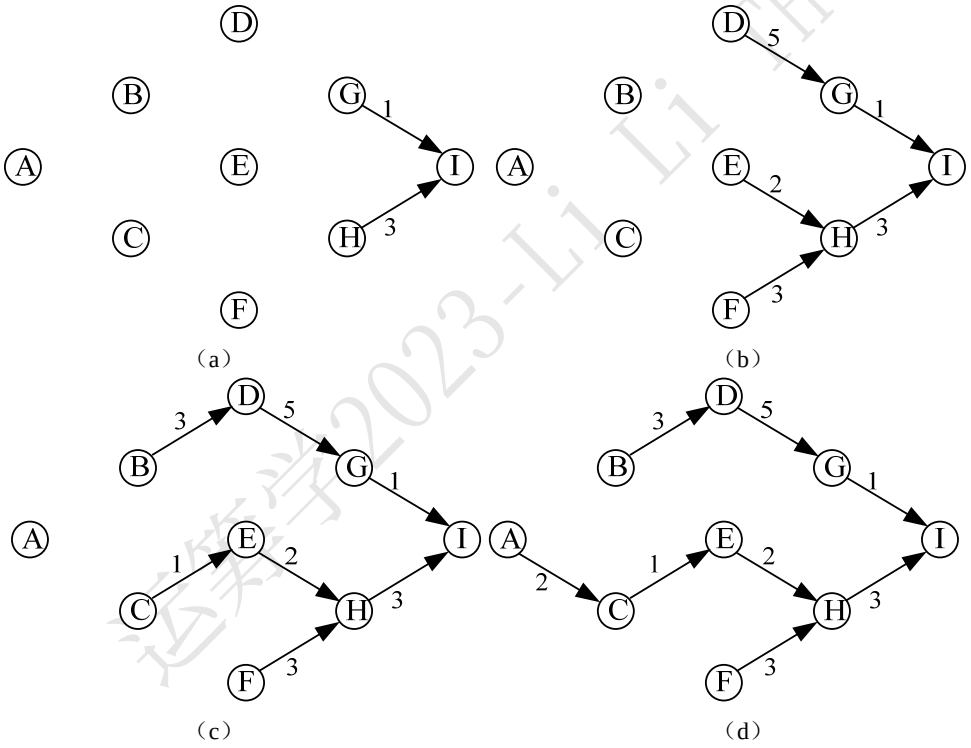


图 6.2　逆推法求解最短油管铺设问题示意图：（a）第 4 阶段到第 5 阶段；（b）第 3 阶段到第 5 阶段：（c）第 2 阶段到第 5 阶段；（d）第 1 阶段到第 5 阶段。

从上面的例子我们可以推广得到如下经常在计算机程序算法设计中使用的动态规划问题定义，特别适合定期的多阶段决策问题。

定义 6.3. [基于递推求解法的动态规划] 动态规划将一个多阶段决策问题分解成多个相互联系的单阶段决策问题，然后迭代求解每个阶段决策变量的解。这几个相互联系的单阶段问题一定是同类型的子问题，而我们要最优化的长期回报也一定是可以分解的。一定有单阶段决策问题可以直接求出最优收益，其它相关的单阶段决策问题可以迭代逐一推导出各自的最优解。因此，动态规划的核心思想就在于合理拆分子问题，尽量将每个子问题只解一遍，记住已经解决的子问题的最优解，从而减少重复计算。

我们需要注意的是，迭代不一定是动态规划。在计算机程序算法设计中，如果考虑不当，将可能造成子问题重复求解浪费时间。例如，如果我们将最优油管铺设问题设计成用顺推法求解点 A 到节点 I 的最短路径，则第一阶段需要求解

$$\max_{a_1\in\{A\to B,A\to C\}}\left\{-4+V^*(B,I),-2+V^*(C,I)\right\}$$

其中，$V^*(B,I)$代表从节点 B 铺设油管至节点 I 最优路径长度，$V^*(C,I)$代表从节点 C 铺设油管至节点 I 最优路径长度。但此时我们还不知道$V^*(B,I)$和$V^*(C,I)$应该取什么值。

因此，在最短油管铺设问题的第二阶段，我们需要求解

$$\max_{\{a_1,a_2\}\in\begin{Bmatrix}A\to B\to D,A\to B\to E\\A\to C\to E,A\to C\to F\end{Bmatrix}}\begin{Bmatrix}-4-3+V^*(D,I),-4-7+V^*(E,I)\\-2-1+V^*(E,I),-4-7+V^*(F,I)\end{Bmatrix}$$

其中，$V^*(B,I)$代表从节点 B 铺设油管至节点 I 最优路径的长度，$V^*(C,I)$代表从节点 C 铺设油管至节点 I 最优路径的长度。

……

如此迭代下去，直到我们找到可以直接求解的$V^*(G,I)$和$V^*(H,I)$，但这显然会造成一些子问题的重复求解，因此并非动态规划。

例 6.3. 试使用动态规划求解 Fibonacci 数列的第 n 项。

解：我们先看一个错误的伪代码

```
输入: n
输出: Fibonacci 数列的第 n 项

function fibonacci(n)
{if n < 2
      return n;
 else
```

```
        return fibonacci(n-1) + fibonacci (n-2);
}
```

很显然，我们随便计算一下 fibonacci(4)，就会产生子问题重复计算的问题，得到图 6.3 所示的递归调用树。

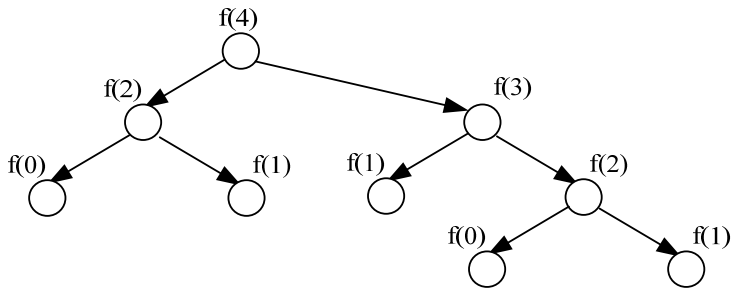


图 6.3　错误的 Fibonacci 数列求解算法递归调用示意图

正确的动态规划伪代码可以写为

```
输入: n
输出: Fibonacci 数列的第 n 项

function fibonacci(n)
{results = array(n+1); //缓存子问题的计算结果，方便复用
 for i=1:n+1 //假设这里 array 的第一个元素下标为 1
 {if i <= 2
       results[i] = 1;
  else
       results[i]=results[i-1] + results[n-2];
 }
 return results[n+1];
}
```

不难看出，虽然核心的迭代计算公式没变，但动态规划没有重复求解子问题。

6.2.2　动态规划问题的建模

动态规划建模一般包含以下过程：

①确定阶段与阶段变量

②明确状态变量和状态可能集合。

③确定决策变量和决策允许集合。

④确定状态转移方程。

⑤明确阶段目标，写出基本方程和边界条件

但最核心的关键是两点：其一，合理拆分出子问题；其二，如何尽量使得每一个子问题只求解一次。兹举数例如下。

例 6.4. 针对下图 6.4（a）给出的节点，可选的路径和响应费用关系，利用动态规划法找出从节点 A 至节点 I 的最短路径铺设输油管道。

解：我们可以考虑如图 6.4（b）的方式，引入虚拟节点 E，使得问题严格满足多阶段决策过程的建模要求，然后可以利用动态规划进行求解。

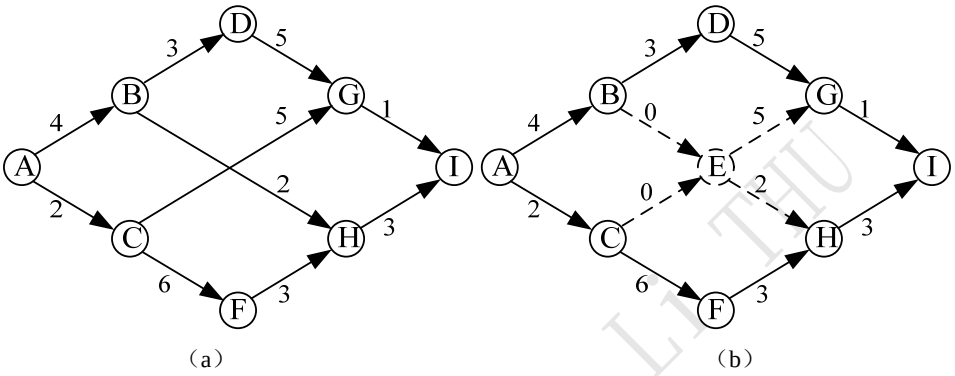


图 6.4　最短油管铺设问题示意图：节点，可选的路径和响应费用。

例 6.6. 针对图 6.1 给出的节点，可选的路径和响应费用关系，增加约束“过节点 B 铺设输油管线则必须过节点 H 铺设输油管线”，利用动态规划法找出从节点 A 至节点 I 的最短路径铺设输油管道。

解：我们可以考虑如图 6.5 的方式，引入虚拟节点 J，使得问题严格满足多阶段决策过程的建模要求，然后可以利用动态规划进行求解。

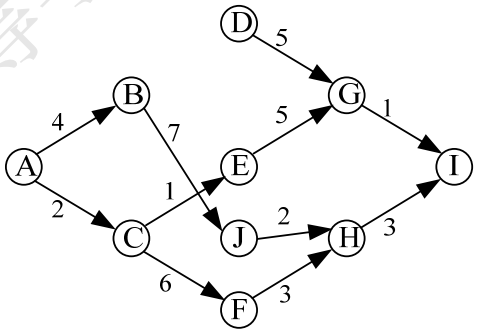


图 6.5　变形之后的最短油管铺设问题示意图：节点，可选的路径和响应费用。

例 6.6. 给定一个整数数组 nums，找到其中最长严格递增子序列长度。譬如：

输入：nums = [10,9,2,3,7,101,18]

可以找到最长递增子序列是 [2,3,7,101]，因此长度为 4

输出：4

解：我们考虑“数组 nums[i]的最长递增子序列长度是否跟子问题 nums[i-1]的最长递增子序列长度有关呢？”来分解子问题

当 nums = [10] 时，最长递增子序列是[10]，长度是 1。

当 nums = [10,9] 时，最长递增子序列是[10]或者[9]，长度是 1。

当 nums = [10,9,2] 时，最长递增子序列是[10]或者[9]或者[2]，长度是 1。

当 nums = [10,9,2,3] 时，最长递增子序列是[2,3]，长度是 2。

……

不难发现，如果新加入一个元素 nums[i]，最长递增子序列要么是以 nums[i]结尾的递增子序列，要么就是 nums[i-1] 原有的最长递增子序列。换而言之，nums[i]的最长递增子序列，就是以数组 nums[i]每个元素结尾的最长子序列集合中元素最多的那个子序列。

假设我们构建数组 dp[i]表示 i 之前包括 i 的最长上升子序列。则状态转移方程应该描述：位置 i 的最长升序子序列等于 j 从 0 到 i-1 各个位置的最长升序子序列 + 1 的最大值。也即 if (nums[i] > nums[j]) dp[i] = max(dp[i], dp[j] + 1)。

于是，我们可以得到如下的伪代码

```
输入：存放图的一维数字数组 nums[]
输出：最长子序列的长度

function lengthof(array nums)
{if (nums.size() <= 1) return nums.size();
 array dp(nums.size(), 1);
 int result = 0; //缓存当前找到的最长子序列的长度
 for (int i = 1; i < nums.size(); i++)
 {for (int j = 0; j < i; j++)
  {if (nums[i] > nums[j])
      dp[i] = max(dp[i], dp[j] + 1);
  }
  if (dp[i] > result)
       result = dp[i]; //  更新当前最长子序列的长度
 }
 return result;
}
```

例 6.7. 用动态规划法求解背包问题（例 4.1）。

解：第一步：构建 dp 数组的含义，dp[i][j] 将存放“前 i 个物品加入容量为 j 的背包里，能达到的最大总价值”。

第二步：构建状态转移方程。实际上，对于一个物品来说：要么放入背包要么不放入背包。如果选择第 i 件物品放入背包中，则有

dp[i][j] = dp[i-1][j-w[i]] + V[i]，并且需要满足约束 j>W[i]

不选择选择第 i 件物品放入背包中，则有

dp[i][j] = dp[i-1][j]

综上分析，可以得到状态转移方程为

dp[i][j] = max(dp[i-1][j],dp[i-1][j-w[i]] + V[i])

动态规划伪代码可以写为

输入：物品数量 N，背包容量 M，记录各物品价值的一维数组 V[]，记录各物品重量的一维数组 W[]

输出：二维数组 dp[i][j]代表前 i 个物品加入容量为 j 的背包里面价值总和的最大值

```
knapsack_2(int N, int M, int V[], int W[])
{int dp[N+1][M+1];   // 创建 dp 数组

 for (int i = 0; i < N; i++)   // 初始化 dp 数组
 {
     dp[i][0] = 0;   // 背包容量为 0，则不能放入任何物品，最大总价值为 0
 }

 for (int j = 0; j < N ; j++)
 {
     dp[0][j] = 0;   // 什么物品都不放入是，最大总价值为 0
 }

 //根据状态转移方程递推填满 dp 数组
 for (int i = 1; i <= N; i++)
    {for (int j = 1; j <= M; j++)
        { if(j<W[i])
            {dp[i][j] = dp[i - 1][j];   // 无法装下第 i 件物品
            }
          else
```

```
                {dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - W[i]] + V[i]); //可装第 i 件物品
              }
          }
      return dp[N][M];    // 输出
    }
```

不难得知，上述算法的时间复杂度为 O(N*M)，空间复杂度为 O(N*M)。时间复杂度无法进一步优化，但空间复杂度还可以优化。注意到在填 dp 数组的第 i+1 行数据时，只需用到 dp 数组的第 i 行数据，根本没有用到 i-1 行及更前面行的数据。也即，填 dp 数组某一行的数据的时候，只与 dp 数组的前一行有关。根据这个规律，我们可以使用一维 dp 数组来缩减空间复杂度。动态规划伪代码可以写为

输入：物品数量 N，背包容量 M，记录各物品价值的一维数组 V[]，记录各物品重量的一维数组 W[]

输出：一维数组 dp[j]代表加入物品容量不搞过 j 时，背包里面武平价值总和的最大值

```
knapsack_1(int N, int M, int V[], int W[])
{int dp[M+1];    // 创建 dp 数组
 for (int j = 0; j < M; j++)    //初始化 dp 数组
    {dp[j] = 0;    //什么物品都不放入是，最大总价值为 0
    }

 //根据状态转移方程递推填满 dp 数组
 for (int i = 1; i <= N; i++)
    {for (int j = M; j >= W[i]; j--)
        {dp[j] = max(dp[j], dp[j - W[i]] + V[i]); //可以装下第 i 件物品
        }
    }

 return dp[M];
}
```

在上述填充 dp 数组的第二层循环的时候，必须保证从后到前（也即右到左）填写数组。因为如果选择从前到后（左到右）填写数组，会导致数组前面元素的值在被引用计算前更新，导致计算错误。

6.3　值迭代法和策略迭代法

对于很多有限阶段但不定期的多阶段决策问题，我们往往不能迅速找到可以直接求出最优收益的单阶段决策问题。此时，我们常常采用评估改进的方式迭代改进策略，直到找到最优动态规划策略。其代表性算法即是值迭代法（Value Iteration Method）和策略迭代法（Policy Iteration Method）。

定义 6.4. [Bellman 最优性方程] 对于很多确定性多阶段决策问题，我们定义：采取特定策略 $f(\bullet)$ 之后，从状态 s 出发，经过多次状态转移之后，到达状态 s' 的收益为 $V_f(s,s'):S\times S\to R$，也可将 $V_f(s,s')$ 称为指标函数或者值函数。不影响理解的情况下，我们甚至直接简写为 $V(s,s')$。相关的优化问题可以定义为找到一种最优策略 $f^*(\bullet)$，使得从给定状态 s_m 到终止状态 s_n 的收益最高，$\max\limits_f V(s_m,s_n)$。结合 Bellman 最优性原理，执行最优策略 $f^*(\bullet)$ 所得到最优值函数 $V^*(s,s_n)$ 应该满足如下的 Bellman 最优性方程

$$V^*(s,s_n)=\max_{a=f(s)}\left\{g(s,a)+V^*\left(T(s,a),s_n\right)\right\},\forall s\in S \tag{6.1}$$

不难发现，如果想知道最优的策略，就需要能够准确估计值函数。然而想准确估计值函数，又需要知道最优策略。因此，我们需要采取迭代方法逐步提高对于值函数的估计精度。由此，我们设计出如下的值迭代法。

算法 6.1. 离散有限确定多阶段决策过程的值迭代法

Step 1. [确定初始值函数] 我们将一个多阶段决策问题分解成多个相互联系的单阶段决策问题，任选一个初始可行策略为 $f^{(1)}(\bullet)$，确定执行 $f^{(1)}(\bullet)$ 后不同状态 s 转移到最终状态 s_n 的值函数 $V^{(1)}(s,s_n)$。令 $k:=1$，进入 Step 2。

Step 2. [贪心改进] 基于当前对于最优值函数 $V^*(s,s_n)$ 的估计值函数 $V^{(k)}(s,s_n)$，采用贪心算法对每一种状态 s 尝试所有可能的动作 a，以期增加值函数

$$V^{(k+1)}(s,s_n)=\max_a\left\{g(s,a)+V^{(k)}\left(T(s,a),s_n\right)\right\},\forall s\in S \tag{6.2}$$

进入 Step 2。

Step 3. [判断收敛性] 比较任意状态 s 转移到最终状态 s_n 的收益 $V^{(k)}(s,s_n)$ 和 $V^{(k+1)}(s,s_n)$，如果 $\left|V^{(k)}(s,s_n)-V^{(k-1)}(s,s_n)\right|\le\varepsilon,\forall s\in S$，$\varepsilon>0$ 为预设的阈值，则值函数已经收敛，应用下述公式提取值函数收敛之后的策略，停止。

$$f^{(k+1)}(s)=\arg\max_a\left\{g(s,a)+V^{(k)}\left(T(s,a),s_n\right)\right\},\forall s\in S \tag{6.3}$$

如果值函数没有收敛，$k:=k+1$，返回 Step 2。

定理 6.2. 如果对于任意状态 s 和 s'，存在 $g(s,\to s)=0$，$V(s,s')\le M$，$M\ge 0$ 为给定的上界，则值迭代法 6.1 对于离散有限阶段的多阶段决策问题可以在有限时间内停止。

如果该问题本质上可以递推求解，也即一定有单阶段决策问题可以直接求出最优收益，其它相关的单阶段决策问题可以迭代逐一推导出各自的最优解。则值迭代法 6.1 可以找到最优解。

证明：如果对于任意状态 s，存在 $g(s,\to s)=0$，则采用公式（6.2）进行贪心改进，必然有 $V^{(k+1)}(s,s_n)\ge V^{(k)}(s,s_n)$，$\forall s\in S$，$k\in\mathbb{N}$，$s_n$ 为最终状态。

对于任意状态 s 和 s'，$V(s,s')\le M$，则 $V^{(k+1)}(s,s_n)\le M$，$\forall s\in S$，$k\in\mathbb{N}$。

因此，对于任意状态 s，单调递增有上界序列 $V^{(k+1)}(s,s_n)$ 必然收敛，故值迭代法 6.1 对于离散有限阶段的多阶段决策问题可以在有限时间内停止。

如果该问题本质上可以递推求解，则第一次迭代时，一定可以通过公式（6.2）确定最终状态 s_n 之前的状态 s_{n-1} 的最佳策略。同理，每一次迭代时，至少可以通过公式（6.2）多确定一个状态的最佳策略。这一求解过程本质上还是递推法，只不过我们不需要显式的指定从那一个状态开始倒退。因为离散有限阶段的多阶段决策问题总的状态数有限，因此一定可以最终收敛到最优解。□

例 6.8. 下图中有五个城市 v_i，$i=1,\cdots,5$，任何两个城市间均有道路直接相连，直连道路长度由图 6.6（a）所示，往返路程一样。试用值迭代法求每个城市到第五个城市 v_5 的最短路线和最短路程。

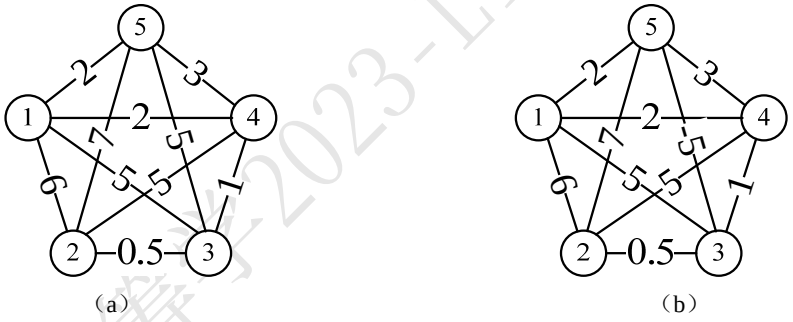


图 6.6　五个城市的直连道路长度示意图：（a）不存在路程之和小于零的环；（b）存在路程之和小于零的环。

解：我们可以构造状态集为 $S=\{v_i,i=1,\cdots,5\}$，策略为 $a=f(s),\forall s\in S$，动作集为 $A=\{\to v_i,i=1,\cdots,5\}$，状态转移函数 $T(s,a)\in S,\forall s\in S,a\in A$，收益函数 $g(s,a)$ 即为从状态 s，执行动作 a，跳到下一个城市之后，两个城市之间的直连道路长度的负值。

我们设初始策略 $f^{(0)}(\bullet)$ 为每个城市直接去城市 v_5，也即 $f^{(0)}(v_1)=\{\to v_5\}$，

$f^{(0)}(v_2)=\{\to v_5\}$，$f^{(0)}(v_3)=\{\to v_5\}$，$f^{(0)}(v_3)=\{\to v_5\}$，$f^{(0)}(v_5)=\{\to v_5\}$。相应的值函数为$V^{(0)}(v_1,v_5)=-2$，$V^{(0)}(v_2,v_5)=-7$，$V^{(0)}(v_3,v_5)=-5$，$V^{(0)}(v_4,v_5)=-3$，$V^{(0)}(v_5,v_5)=0$。

第一轮迭代，我们发现可以更新$f^{(0)}(v_2)=\{\to v_3\}$，$f^{(0)}(v_3)=\{\to v_4\}$。相应的值函数更新为$V^{(0)}(v_2,v_5)=-5.5$，$V^{(0)}(v_3,v_5)=-4$。

第二轮迭代，我们发现策略都没变，但值函数有更新，$V^{(0)}(v_2,v_5)=-4.5$。

第三轮迭代，我们发现策略和值函数都没有更新，停止。各城市到第五个城市v_5的最短路线和最短路程为：

路线$v_1\to v_5$，路程为2；

路线$v_2\to v_3\to v_4\to v_5$，路程为4.5；

路线$v_3\to v_4\to v_5$，路程为4；

路线$v_4\to v_5$，路程为3；

路线$v_5\to v_5$，路程为0。

如果五个城市之间的直连道路长度由图 6.6（b）所示，则存在路程之和小于零的环$v_3\to v_4\to v_5\to v_3$。如果反复绕该环进行状态跳转，将违背对于任意状态s_1和s_2，$V(s_1,s_2)$有上界的假定，因此无法用值迭代算法 6.1 正确求解。

实际上，在迭代过程中我们对策略和值函数的估计同时进行更新，设计如下的策略迭代法求解。

算法 6.2. 离散有限确定多阶段决策过程的策略迭代法

Step 1. [确定初始策略] 我们将一个多阶段决策问题分解成多个相互联系的单阶段决策问题，任选一个初始可行策略为$f^{(1)}(\bullet)$，该策略需要保证下述方程组（6.4）有解。令$k:=1$，进入 Step 2。

Step 2. [策略评估（Policy Evaluation）] 基于当前策略，对于最优值函数$V^*(s,s_n)$进行估计，我们可以通过解一个线性方程组

$$\begin{bmatrix}V^{(k)}(s_1,s_n)\\V^{(k)}(s_2,s_n)\\\cdots\\V^{(k)}(s_n,s_n)\end{bmatrix}=\begin{bmatrix}g\left(s_1,f^{(k)}(s_1)\right)\\g\left(s_2,f^{(k)}(s_2)\right)\\\cdots\\g\left(s_n,f^{(k)}(s_n)\right)\end{bmatrix}+\begin{bmatrix}V^{(k)}\left(T\left(s_1,f^{(k)}(s_1)\right),s_n\right)\\V^{(k)}\left(T\left(s_2,f^{(k)}(s_2)\right),s_n\right)\\\cdots\\V^{(k)}\left(T\left(s_n,f^{(k)}(s_n)\right),s_n\right)\end{bmatrix}\quad(6.4)$$

求解该线性方程组，我们可以同时得到每一个状态的估计值函数$V^{(k)}(s,s_n)$。进入 Step 3。

Step 3. [策略改进（Policy Improvement）] 基于当前对于最优值函数$V^*(s,s_n)$的估计值函数$V^{(k)}(s,s_n)$，采用贪心算法对每一种状态s尝试所有可能的动作以

期改进策略

$$f^{(k+1)}(s)=\arg\max_{a=f(s)}\left\{g(s,a)+V^{(k)}\left(T(s,a),s_n\right)\right\},\forall s\in S \qquad (6.5)$$

进入 Step 4。

Step 4. [判断收敛性] 比较策略 $f^{(k)}$ 和 $f^{(k+1)}$ 对于任意状态 s 的行为差异，如果 $\left|f^{(k)}(s)-f^{(k-1)}(s)\right|\le\varepsilon,\forall s\in S$，$\varepsilon>0$ 为预设的阈值，则策略已经收敛，停止。如果策略没有收敛，$k:=k+1$，返回 Step 2。

需要注意的是，一个决策问题的最优策略可能不唯一。使用不同的初始策略，通过策略迭代最后收敛到的最优策略可以是不一样。不过这些策略对应的值函数应该是一样的。

定理 6.3. 如果对于任意状态 s 和 $s'\neq s$，存在 $g(s,\to s')\neq 0$，且有 $g(s,\to s)=0$，$V(s,s')\le M$，$M\ge 0$ 为给定的上界，则策略迭代法 6.2 对于离散有限阶段的多阶段决策问题可以在有限时间内停止。

如果该问题本质上可以递推求解，也即一定有单阶段决策问题可以直接求出最优收益，其它相关的单阶段决策问题可以迭代逐一推导出各自的最优解。则策略迭代法 6.2 可以找到最优解。

证明：由于这里考虑的是离散有限阶段的多阶段决策问题，我们可以引入矩阵 $P^{(k)}=\left[p_{ij}\right]\in\mathbb{R}^{n\times n}$ 来表示执行第 k 轮策略改进之后，状态之间的转移关系

$$p_{ij}=\begin{cases}1, if\ \ T\left(s_i,f^{(k)}(s_i)\right)=s_j\\ 0, otherwise\end{cases},\ \ i,j=1,\ldots,n \qquad (6.6)$$

则我们将第 k 轮策略评估公式（6.4）简写为

$$\begin{bmatrix}V^{(k)}(s_1,s_n)\\ V^{(k)}(s_2,s_n)\\ \cdots\\ V^{(k)}(s_n,s_n)\end{bmatrix}=\begin{bmatrix}g\left(s_1,f^{(k)}(s_1)\right)\\ g\left(s_2,f^{(k)}(s_2)\right)\\ \cdots\\ g\left(s_n,f^{(k)}(s_n)\right)\end{bmatrix}+P^{(k)}\cdot\begin{bmatrix}V^{(k)}(s_1,s_n)\\ V^{(k)}(s_2,s_n)\\ \cdots\\ V^{(k)}(s_n,s_n)\end{bmatrix} \qquad (6.7)$$

我们首先证明，如果策略迭代法 6.2 的 Step 1 选择的初始策略可行，则迭代过程中不会产生不可行的策略。具体而言，也就是迭代过程中，方程（6.7）保证有解。为了证明这一点，我们首先分析一下，方程（6.7）何时有解。

如果策略 $f^{(k)}(s)$ 不构成环，$\forall s\in S$，$k\in\mathbb{N}$，s_n 为最终状态，则方程（6.7）一定有解。因为任一一个状态只转移到自身或或者另外一个状态。从图的角度来看，如果我们将每个状态视为一个节点，每个转移视为一条边，则状态转移关系构成一个有向无环图。方程（6.7）的每个状态（节点）的值函数可以依次从图中没有离开边的那些节点从进入边反向递推得到，一定有解。

如果策略 $f^{(k)}(s)$ 构成一个或者多个状态环，不妨假设其中一个状态环定义为 $s_{(p)} \to s_{(p+1)} \to \ldots \to s_{(m)} \to s_{(p)}$， $p \neq m$，也即

$$T\left(s_{(i-1)}, f^{(k+1)}\left(s_{(i-1)}\right)\right) = s_{(i)}，\ i = p,\ldots,m，\ T\left(s_{(m)}, f^{(k+1)}\left(s_{(m)}\right)\right) = s_{(p)} \tag{6.8}$$

则必须有

$$g\left(s_{(i-1)}, f^{(k+1)}\left(s_{(i-1)}\right)\right) = g\left(s_{(i-1)}, \to s_i\right) = 0，\ i = p,\ldots,m，\ g\left(s_{(\mathrm{m})}, \to s_p\right) = 0 \tag{6.9}$$

方程（6.7）才能有解。但这与我们的假设对于任意状态 s 和 $s' \neq s$，存在 $g(s, \to s') > 0$ 矛盾。因此，可行策略不能包含任何环。

因此，想要证明策略迭代法 6.2 保持有解性，就需要证明迭代过程中，策略 $f^{(k)}(s)$ 不构成环，策略 $f^{(k+1)}(s)$ 也不可能构成环。我们还是用反证法来证明。

假设结论不成立，策略 $f^{(k)}(s)$ 不构成环，经过迭代后，策略 $f^{(k+1)}(s)$ 构成了一个或者多个状态环，不妨进一步假设其中一个状态环定义为 $s_{(p)} \to s_{(p+1)} \to \ldots \to s_{(m)} \to s_{(p)}$， $p \neq m$，此时有（6.8）成立。

在策略 $f^{(k+1)}(s)$ 下，状态 $s_{(p+i)}$， $i = 0,\ldots,m-p-1$，到达终止状态 s_n 的收益为 $V^{(k+1)}\left(s_{(p+i)}, s_n\right)$。同时满足

$$V^{(k+1)}\left(s_{(p)}, s_n\right) = g\left(s_{(p)}, f^{(k+1)}\left(s_{(p)}\right)\right) + V^{(k+1)}\left(s_{(p+1)}, s_n\right)$$

$$V^{(k+1)}\left(s_{(p+1)}, s_n\right) = g\left(s_{(p+1)}, f^{(k+1)}\left(s_{(p+1)}\right)\right) + V^{(k+1)}\left(s_{(p+2)}, s_n\right)$$

…

$$V^{(k+1)}\left(s_{(m-1)}, s_n\right) = g\left(s_{(m-1)}, f^{(k+1)}\left(s_{(m-1)}\right)\right) + V^{(k+1)}\left(s_{(m)}, s_n\right)$$

$$V^{(k+1)}\left(s_{(m)}, s_n\right) = g\left(s_{(m)}, f^{(k+1)}\left(s_{(m)}\right)\right) + V^{(k+1)}\left(s_{(p)}, s_n\right)$$

将上述这些式子全部叠加，等号左右对应的 $V^{(k+1)}\left(s_{(p+i)}, s_n\right)$，$i = 0,\ldots,m-p-1$，可以全部消去，得到

$$\sum_{i=0}^{m-p} g\left(s_{(p+i)}, f^{(k+1)}\left(s_{(p+i)}\right)\right) = 0 \tag{6.10}$$

迭代前，$f^{(k)}(s)$ 策略下，状态 $s_{(p+i)}$，$i = 0,\ldots,m-p-1$ 到达终止状态 s_n 的收益为 $V^{(k)}\left(s_{(p+i)}, s_n\right)$。按照策略迭代的改进方法（6.2），策略 $f^{(k+1)}\left(s_{(p+i)}\right)$，$i = 0,\ldots,m-p-1$，必定满足：

$$g\left(s_{(p+i)}, f^{(k+1)}\left(s_{(p+i)}\right)\right) + V^{(k)}\left(s_{(p+i+1)}, s_n\right) \geq V^{(k)}\left(s_{(p+i)}, s_n\right)$$

依次列出这些不等式，我们得到

$$g\left(s_{(p)}, f^{(k+1)}\left(s_{(p)}\right)\right)+V^{(k)}\left(s_{(p+1)}, s_n\right) \geq V^{(k)}\left(s_{(p)}, s_n\right)$$

$$g\left(s_{(p+1)}, f^{(k+1)}\left(s_{(p+1)}\right)\right)+V^{(k)}\left(s_{(p+2)}, s_n\right) \geq V^{(k)}\left(s_{(p+1)}, s_n\right)$$

…

$$g\left(s_{(m-1)}, f^{(k+1)}\left(s_{(m-1)}\right)\right)+V^{(k)}\left(s_{(m)}, s_n\right) \geq V^{(k)}\left(s_{(m-1)}, s_n\right)$$

$$g\left(s_{(m)}, f^{(k+1)}\left(s_{(m)}\right)\right)+V^{(k)}\left(s_{(p)}, s_n\right) \geq V^{(k)}\left(s_{(m)}, s_n\right)$$

将上述这些式子全部叠加，等号左右对应的 $V^{(k)}\left(s_{(p+i)}, s_n\right)$，$i=0,...,m-p-1$，可以全部消去，得到

$$\sum_{i=0}^{m-p} g\left(s_{(p+i)}, f^{(k+1)}\left(s_{(p+i)}\right)\right) \geq 0 \tag{6.11}$$

我们结合（6.10）式和（6.11）式来讨论：

第一种情况，如果（6.11）式中取等号，则（6.10）式和（6.11）式同时成立，并且说明此时迭代已经完成，策略 $f^{(k)}(s)$ 和策略 $f^{(k+1)}(s)$ 一致，但根据假设，策略 $f^{(k)}(s)$ 不构成环，策略 $f^{(k+1)}(s)$ 构成环，产生矛盾，所以假设不成立，$f^{(k+1)}(s)$ 也不构成环。

第二种情况，（6.11）式中不取等号，则（6.10）式和（6.11）式不能同时成立，产生矛盾，所以假设不成立，$f^{(k+1)}(s)$ 也不构成环。

我们接着证明，策略迭代法 6.2 的 Step 1 选择的初始策略可行时，可以收敛。类似公式（6.7），对于第 $k+1$ 轮策略评估，有

$$\begin{bmatrix} V^{(k+1)}\left(s_1, s_n\right) \\ V^{(k+1)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k+1)}\left(s_n, s_n\right) \end{bmatrix} = \begin{bmatrix} g\left(s_1, f^{(k+1)}\left(s_1\right)\right) \\ g\left(s_2, f^{(k+1)}\left(s_2\right)\right) \\ \cdots \\ g\left(s_n, f^{(k+1)}\left(s_n\right)\right) \end{bmatrix} + P^{(k+1)} \cdot \begin{bmatrix} V^{(k+1)}\left(s_1, s_n\right) \\ V^{(k+1)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k+1)}\left(s_n, s_n\right) \end{bmatrix} \tag{6.12}$$

而进行第 $k+1$ 轮策略改进之后，对于任意状态 s，有

$$g\left(s, f^{(k+1)}(s)\right)+P^{(k+1)} \cdot V^{(k)}\left(s, s_n\right) \geq g\left(s, f^{(k)}(s)\right)+P^{(k)} \cdot V^{(k)}\left(s, s_n\right)$$

代入公式（6.7），得到

$$\begin{bmatrix} g\left(s_1, f^{(k+1)}\left(s_1\right)\right) \\ g\left(s_2, f^{(k+1)}\left(s_2\right)\right) \\ \cdots \\ g\left(s_n, f^{(k+1)}\left(s_n\right)\right) \end{bmatrix} + P^{(k+1)} \cdot \begin{bmatrix} V^{(k)}\left(s_1, s_n\right) \\ V^{(k)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k)}\left(s_n, s_n\right) \end{bmatrix}$$

$$\geq \begin{bmatrix} g\left(s_1, f^{(k)}\left(s_1\right)\right) \\ g\left(s_2, f^{(k)}\left(s_2\right)\right) \\ \cdots \\ g\left(s_n, f^{(k)}\left(s_n\right)\right) \end{bmatrix} + P^{(k)} \cdot \begin{bmatrix} V^{(k)}\left(s_1, s_n\right) \\ V^{(k)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k)}\left(s_n, s_n\right) \end{bmatrix} = \begin{bmatrix} V^{(k)}\left(s_1, s_n\right) \\ V^{(k)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k)}\left(s_n, s_n\right) \end{bmatrix} \tag{6.13}$$

结合公示（6.12）和（6.13）可以得到

$$\left(I - P^{(k+1)}\right) \cdot \left(\begin{bmatrix} V^{(k)}\left(s_1, s_n\right) \\ V^{(k)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k)}\left(s_n, s_n\right) \end{bmatrix} - \begin{bmatrix} V^{(k+1)}\left(s_1, s_n\right) \\ V^{(k+1)}\left(s_2, s_n\right) \\ \cdots \\ V^{(k+1)}\left(s_n, s_n\right) \end{bmatrix} \right) \leq \mathbf{0} \tag{6.14}$$

其中 $I \in \mathbb{R}^{n \times n}$ 为单位阵。

以下，我们用反证法证明，必然有 $V^{(k+1)}\left(s, s_n\right) \geq V^{(k)}\left(s, s_n\right)$，$\forall s \in S$，$k \in \mathbb{N}$，$s_n$ 为最终状态。

假设存在某状态 $s_{(1)} \in S$，有 $V^{(k+1)}\left(s_{(1)}, s_n\right) < V^{(k)}\left(s_{(1)}, s_n\right)$，根据公示（6.14），发生这一情况，只可能存在某状态 $s_{(2)} \in S$，使得

$$T\left(s_{(1)}, f^{(k+1)}\left(s_{(1)}\right)\right) = s_{(2)} \in S, \ V^{(k+1)}\left(s_{(2)}, s_n\right) < V^{(k)}\left(s_{(2)}, s_n\right)$$

因为 $V^{(k+1)}\left(s_{(2)}, s_n\right) < V^{(k)}\left(s_{(2)}, s_n\right)$，可推出必然存在一个状态 $s_{(3)} \in S$ 满足

$$V^{(k+1)}\left(s_{(3)}, s_n\right) < V^{(k)}\left(s_{(3)}, s_n\right), \ T\left(s_{(1)}, f^{(k+1)}\left(s_{(1)}\right)\right) = s_{(3)}$$

……

因为离散有限阶段的多阶段决策问题总的状态数有限，依次类推，最终一定可以得到一个状态环 $s_{(p)} \to s_{(p+1)} \to \ldots \to s_{(m)} \to s_{(p)}$（该环不一定包含 $s_{(1)}$），均满足公式（6.8）及

$$V^{(k+1)}\left(s_{(i)}, s_n\right) < V^{(k)}\left(s_{(i)}, s_n\right), \ i = p, \ldots, m \tag{6.15}$$

但公式（6.15）代入公式（6.14），必然无解，发生矛盾。

因此，必然有 $V^{(k+1)}\left(s, s_n\right) \geq V^{(k)}\left(s, s_n\right)$，$\forall s \in S$，$k \in \mathbb{N}$，$s_n$ 为最终状态。

对于任意状态 s 和 s'，$V\left(s, s'\right) \leq M$，则 $V^{(k+1)}\left(s, s_n\right) \leq M$，$\forall s \in S$，$k \in \mathbb{N}$。

因此，对于任意状态 s，单调递增有上界序列 $V^{(k+1)}\left(s, s_n\right)$ 必然收敛，故策略

迭代法 6.2 对于离散有限阶段的多阶段决策问题可以在有限时间内停止。

如果该问题本质上可以递推求解，则经过有限次迭代，一定可以通过公式（6.5）首先确定最终状态 s_n 之前的状态 s_{n-1} 的最佳策略。类似的，我们可以逐渐依次确定各个状态的最佳策略。这一求解过程本质上还是递推法，只不过我们不需要显式的指定从那一个状态开始倒退。因为离散有限阶段的多阶段决策问题总的状态数有限，因此一定可以最终收敛到最优解。□

例 6.9. 试用策略迭代法求图 6.6（a）中每个城市到第五个城市 v_5 的最短路线和最短路程。

解：我们取初始策略为 $f(v_1)=\{\rightarrow v_4\}$，$f(v_2)=\{\rightarrow v_4\}$，$f(v_3)=\{\rightarrow v_4\}$，$f(v_4)=\{\rightarrow v_5\}$，$f(v_5)=\{\rightarrow v_5\}$。得到第一个方程组：

$$\begin{bmatrix} V^{(1)}(s_1) \\ V^{(1)}(s_2) \\ V^{(1)}(s_3) \\ V^{(1)}(s_4) \\ V^{(1)}(s_5) \end{bmatrix} = \begin{bmatrix} -2 \\ -5 \\ -1 \\ -3 \\ 0 \end{bmatrix} + \begin{bmatrix} V^{(1)}(s_4) \\ V^{(1)}(s_4) \\ V^{(1)}(s_4) \\ V^{(1)}(s_5) \\ V^{(1)}(s_5) \end{bmatrix}$$

注意到 $V^{(1)}(s_5)=0$，我们可以解出 $V^{(1)}(s_1)=-5$，$V^{(1)}(s_2)=-8$，$V^{(1)}(s_3)=-4$，$V^{(1)}(s_4)=-3$。

进行改进之后，我们得到 $f(v_1)=\{\rightarrow v_5\}$，$f(v_2)=\{\rightarrow v_3\}$，$f(v_3)=\{\rightarrow v_4\}$，$f(v_4)=\{\rightarrow v_5\}$，$f(v_5)=\{\rightarrow v_5\}$。进而得到第二个方程组：

$$\begin{bmatrix} V^{(1)}(s_1) \\ V^{(1)}(s_2) \\ V^{(1)}(s_3) \\ V^{(1)}(s_4) \\ V^{(1)}(s_5) \end{bmatrix} = \begin{bmatrix} -2 \\ -0.5 \\ -1 \\ -3 \\ 0 \end{bmatrix} + \begin{bmatrix} V^{(1)}(s_4) \\ V^{(1)}(s_4) \\ V^{(1)}(s_4) \\ V^{(1)}(s_5) \\ V^{(1)}(s_5) \end{bmatrix}$$

注意到 $V^{(1)}(s_5)=0$，我们可以解出 $V^{(1)}(s_1)=-2$，$V^{(1)}(s_2)=-4.5$，$V^{(1)}(s_3)=-4$，$V^{(1)}(s_4)=-3$。

进行改进之后，我们得到 $f(v_1)=\{\rightarrow v_5\}$，$f(v_2)=\{\rightarrow v_3\}$，$f(v_3)=\{\rightarrow v_4\}$，$f(v_4)=\{\rightarrow v_5\}$，$f(v_5)=\{\rightarrow v_5\}$。发现策略已经收敛，停止。结果同值迭代法结果一样。

如果五个城市之间的直连道路长度由图 6.5（b）所示，容易验证，同样无法用策略迭代算法 6.1 正确求解。

需要注意的是，对于该问题，我们选取的初始策略必须是无环的，如果有环，则得到的线性方程组可能无解，无法进一步计算。例如，如果我们取初始策略为

$f(v_1)=\{\to v_4\}$，$f(v_2)=\{\to v_1\}$，$f(v_3)=\{\to v_2\}$，$f(v_4)=\{\to v_3\}$，$f(v_5)=\{\to v_5\}$，则得到的方程组无解。

有的应用问题策略迭代快，有的应用问题值迭代快。选择哪种算法解决问题比较复杂，我们不在本书讨论。

6.4 Markov 决策过程

不确定性多阶段决策过程的建模方法有很多，其中最为重要的是 Markov 决策过程（Markov Decision Process）。

定义 6.6. [Markov 多阶段决策过程] 假设一个针对特定系统的决策过程可以划分成若干阶段。第t个阶段该系统的所有可用信息被称为状态，用$s_t \in S_t$表示，当前阶段所有可能的状态集合为$S_t \subset S$，S 表示所有阶段所有可能的状态构成的集合。考虑当前状态s_t之后我们可以进行一次决策并采取行动（Action），用策略函数$\pi(s_t, a_t) \to [0,1]$，表示我们根据状态s_t选择动作$a_t \in A$的概率，A表示所有可能的动作构成的集合。做出该行动后，系统从第t个阶段状态s到第$t+1$个阶段状态成为s'的转移概率（Transition Probability）为$\Pr_a(s,s') = \Pr(s_{t+1}=s' \mid s_t=s, a_t=a)$。我们在第$t$个阶段获得收益（Reward）用$r_t = g(s_t, a_t, s_{t+1})$表示，$g(\cdot,\cdot,\cdot): S\times A\times S \to R$表示系统收益函数（Reward Function），R表示所有可能的收益构成的集合。一般来说，我们希望找到一种最优策略$\pi^*(\cdot,\cdot)$，使得从起始状态到最终状态的长期收益期望最高，$\max\limits_{\pi} \mathbf{E}\left(\sum\limits_t r_t\right)$，这里$\mathbf{E}(\cdot)$表示期望函数。考虑到$\max\limits_{\pi} \mathbf{E}\left(\sum\limits_t r_t\right)$一般趋向于正无穷，因此我们常引入折扣因子$\gamma \in (0,1)$，转而寻找一种最优策略$\pi^*(\cdot,\cdot)$，使得从起始状态到最终状态的长期折扣收益期望最高，$\max\limits_{\pi} \mathbf{E}\left(\sum\limits_t \gamma^{t-1} r_t\right)$。

需要指出的是，系统转移概率$\Pr_a(s,s') = \Pr(s_{t+1}=s' \mid s_t=s, a_t=a)$的假设隐含了 Markov 性，也称为无后效性。即未来状态仅由当前状态决定，而与过去状态无关。而且，这里的系统转移概率不随时间变化，具有时间同质性（Time-Homogeneous Stationary Transition Function）。

和确定性多阶段决策过程不同，我们这里采取的策略 $\pi(s_t, a_t)$ 对于同样的状态 s_t，可以依概率采取不同行动 a_t。也就是我们的策略也可以是随机的。

定义 6.6. [Bellman 最优性方程] 对于 Markov 多阶段决策过程，我们定义：采取特定策略 $f(\bullet)$ 之后，从状态 s 出发到终止状态获得的长期收益为值函数 $V_f(s)$ 称为指标函数或者值函数。注意到给定策略 $\pi(\cdot,\cdot)$ 之后，我们常将每一阶段的收益 $r_t = g(s_t, a_t, s_{t+1})$ 进一步简化为 $r_t = g(s_t, s_{t+1})$。此时，$V(s)$ 可以视为两个部分组成：其一当前收益 $r_t = g(s_t, \cdot)$，其二是转移到所有可能的状态，并进一步从这些状态迭代到终态的期望回报的概率加权和。进一步考虑时间同质性，我们可以根据 Bellman 最优性原理，定义 Markov 多阶段决策过程的 Bellman 最优性方程为

$$\begin{aligned} V_\pi^*(s) &= \max_{a\in A}\left\{ g(s,\cdot) + \sum_{s'\in S} \Pr_a(s,s')\cdot\gamma\cdot V_\pi^*(s') \right\} \\ &= \max_{a\in A}\left\{ \sum_{s'\in S} \Pr_a(s,s')\left[g(s,s') + \gamma\cdot V_\pi^*(s') \right] \right\} \end{aligned} \tag{6.16}$$

同时，还成立

$$\pi^*(\mathrm{s}) = \arg\max_{a\in A}\left\{ \sum_{s'\in S} \Pr_a(s,s')\left[g(s,s') + \gamma\cdot V_\pi^*(s') \right] \right\} \tag{6.17}$$

类似于确定性多阶段决策问题，我们可以得到如下的 Markov 多阶段决策过程的值迭代法和策略迭代法。

算法 6.3. Markov 多阶段决策过程的值迭代法

Step 1. [确定初始值函数] 我们将一个多阶段决策问题分解成多个相互联系的单阶段决策问题，对于所有状态 s，将状态 s 转移到最终状态的值函数 $V^{(1)}(s)$ 置为 0 或者任意值。令 $k := 1$，进入 Step 2。

Step 2. [贪心改进] 基于当前对于最优值函数 $V^*(s)$ 的估计值函数 $V^{(k)}(s)$，采用贪心算法对每一种状态 s 尝试所有可能的动作 a，以期增加值函数

$$V^{(k+1)}(s) = \max_{a}\left\{ \sum_{s'\in S} \Pr_a(s,s')\left[g(s,s') + \gamma\cdot V^{(k)}(s') \right] \right\}, \forall s\in S \tag{6.18}$$

进入 Step 2。

Step 3. [判断收敛性] 比较任意状态 s 转移到最终状态的收益 $V^{(k)}(s)$ 和 $V^{(k+1)}(s)$，如果 $\left|V^{(k)}(s) - V^{(k-1)}(s)\right| \le \varepsilon, \forall s\in S$，$\varepsilon > 0$ 为预设的阈值，则值函

数已经收敛，应用下述公式提取值函数收敛之后的策略，停止。

$$\pi^{(k+1)}(s)=\arg\max_{a}\left\{\sum_{s'\in S}\Pr_a(s,s')\left[g(s,s')+\gamma\cdot V^{(k)}(s')\right]\right\},\forall s\in S \tag{6.19}$$

如果值函数没有收敛，$k:=k+1$，返回 Step 2。

引理 6.1. 对于函数 $f(\mathbf{x}):\Omega\to R$ 和 $g(\mathbf{x}):\Omega\to R$，$\mathbf{x}\in\Omega$，存在

$$\left|\max_{\mathbf{x}\in\Omega}f(\mathbf{x})-\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\right|\le\max_{\mathbf{z}\in\Omega}\left|f(\mathbf{z})-g(\mathbf{z})\right| \tag{6.20}$$

证明：根据 max 函数的性质，容易得到

$$\max_{\mathbf{z}\in\Omega}\left[f(\mathbf{z})-g(\mathbf{z})\right]+\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\ge\max_{\mathbf{x}\in\Omega}\left[f(\mathbf{x})-g(\mathbf{x})+g(\mathbf{x})\right]\ge\max_{\mathbf{x}\in\Omega}f(\mathbf{x})$$

因此有

$$\max_{\mathbf{x}\in\Omega}f(\mathbf{x})-\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\le\max_{\mathbf{z}\in\Omega}\left[f(\mathbf{z})-g(\mathbf{z})\right]$$

类似的，我们有

$$\max_{\mathbf{y}\in\Omega}g(\mathbf{y})-\max_{\mathbf{x}\in\Omega}f(\mathbf{x})\le\max_{\mathbf{z}\in\Omega}\left[g(\mathbf{z})-f(\mathbf{z})\right]$$

接下来，我们分两种情况讨论。

第一种情况，$\max_{\mathbf{x}\in\Omega}f(\mathbf{x})\ge\max_{\mathbf{y}\in\Omega}g(\mathbf{y})$，则有

$$0\le\max_{\mathbf{x}\in\Omega}f(\mathbf{x})-\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\le\max_{\mathbf{z}\in\Omega}\left[f(\mathbf{z})-g(\mathbf{z})\right]$$

因此有

$$\left|\max_{\mathbf{x}\in\Omega}f(\mathbf{x})-\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\right|\le\left|\max_{\mathbf{z}\in\Omega}\left[f(\mathbf{z})-g(\mathbf{z})\right]\right|$$
$$\le\max_{\mathbf{z}\in\Omega}\left|f(\mathbf{z})-g(\mathbf{z})\right|$$

第二种情况，$\max_{\mathbf{x}\in\Omega}f(\mathbf{x})<\max_{\mathbf{y}\in\Omega}g(\mathbf{y})$，则有

$$0\le\max_{\mathbf{y}\in\Omega}g(\mathbf{y})-\max_{\mathbf{x}\in\Omega}f(\mathbf{x})\le\max_{\mathbf{z}\in\Omega}\left[g(\mathbf{z})-f(\mathbf{z})\right]$$

因此有

$$\left|\max_{\mathbf{y}\in\Omega}g(\mathbf{y})-\max_{\mathbf{x}\in\Omega}f(\mathbf{x})\right|\le\left|\max_{\mathbf{z}\in\Omega}\left[g(\mathbf{z})-f(\mathbf{z})\right]\right|$$
$$\le\max_{\mathbf{z}\in\Omega}\left|g(\mathbf{z})-f(\mathbf{z})\right|$$

综合上述两种情况，我们有

$$\left|\max_{\mathbf{x}\in\Omega}f(\mathbf{y})-\max_{\mathbf{y}\in\Omega}g(\mathbf{y})\right|\le\max_{\mathbf{z}\in\Omega}\left|f(\mathbf{z})-g(\mathbf{z})\right|$$

引理得证。□

定理 6.4. 值迭代法 6.3 对于 Markov 多阶段决策过程问题可以在有限时间内停止。

证明：如果对于任意状态 s，满足 $\delta>0$，$\left|V^{(k)}(s)-V^{(k-1)}(s)\right|<\delta$，或者写成向量 ∞ 范数的形式

$$\left\|\begin{bmatrix}V^{(k+1)}(s_1)\\V^{(k+1)}(s_2)\\\cdots\\V^{(k+1)}(s_n)\end{bmatrix}-\begin{bmatrix}V^{(k)}(s_1)\\V^{(k)}(s_2)\\\cdots\\V^{(k)}(s_n)\end{bmatrix}\right\|_{\infty}<\gamma\delta$$

则根据引理 6.1，我们可以得到

$$\begin{aligned}&\left|V^{(k+1)}(s)-V^{(k)}(s)\right|\\&=\left|\begin{aligned}&\max_a\left\{\sum_{s\in S}\Pr_a(s,s')\left|g(s_1,s')+\gamma\cdot V^{(k+1)}(s')\right|\right\}\\&-\max_a\left\{\sum_{s\in S}\Pr_a(s,s')\left[g(s_1,s')+\gamma\cdot V^{(k)}(s')\right]\right\}\end{aligned}\right|\\&\le\max_a\left\{\left|\gamma\cdot\sum_{s\in S}\Pr_a(s,s')\left[V^{(k)}(s)-V^{(k)}(s)\right]\right|\right\}\\&\le\max_a\left\{\gamma\cdot\sum_{s\in S}\Pr_a(s,s')\left|V^{(k)}(s)-V^{(k)}(s)\right|\right\}\\&\le\max_a\left\{\gamma\cdot\sum_{s\in S}\Pr_a(s,s')\delta\right\}\\&=\gamma\delta\end{aligned}$$

进一步可知 $\left|V^{(k+m)}(s)-V^{(k+m-1)}(s)\right|<\gamma^m\delta$，由于 $\gamma\in(0,1)$，值迭代法 6.3 所形成的映射是压缩映射，根据巴拿赫不动点定理（Banach Fixed-Point Theorem），对于完备度量空间上的压缩映射，有且仅有一个不动点。因此迭代算法肯定收敛到唯一的策略

$$V^*(s)=\lim_{m\to+\infty}V^{(m)}(s),\quad\forall s\in S$$

假设值迭代法 6.3 经过 k 次迭代后满足停止条件 $\left|V^{(k)}(s)-V^{(k-1)}(s)\right|\le\varepsilon$，则

$$\left|V^{(k+m)}(s)-V^{(k)}(s)\right|=\left|\sum_{i=1}^{m}\left[V^{(k+i)}(s)-V^{(k+i-1)}(s)\right]\right|$$

$$\leq\sum_{i=1}^{m}\left|\left[V^{(k+i)}(s)-V^{(k+i-1)}(s)\right]\right|$$

$$<\sum_{i=1}^{m}\gamma^{i}\varepsilon=\gamma\frac{1-\gamma^{m}}{1-\gamma}\varepsilon$$

对上述不等式两边同时对于m取极限之后，不难得知

$$\left|V^{*}(s)-V^{(k)}(s)\right|\leq\frac{\gamma}{1-\gamma}\varepsilon$$

则$V^{(k)}(s)$距离策略$V^{*}(s)$的差距最多为$\frac{\gamma}{1-\gamma}\varepsilon$。□

算法 6.4. Markov 多阶段决策过程的的策略迭代法

Step 1. [确定初始策略] 我们将一个多阶段决策问题分解成多个相互联系的单阶段决策问题，任选一个初始可行策略为$\pi^{(1)}(\bullet)$。令$k:=1$，进入 Step 2。

Step 2. [策略评估（Policy Evaluation）] 基于当前策略，对于最优值函数$V^{*}(s)$进行估计，得到每一个状态s的估计值函数$V^{(k)}(s)$

$$V^{(k)}(s)=\sum_{s'\in S}\Pr_{a}(s,s')\left[g(s,s')+\gamma\cdot V^{(k)}(s')\right] \tag{6.21}$$

如果是离散有限状态的情况，我们可以通过解一个线性方程组

$$\begin{bmatrix}V^{(k)}(s_1)\\V^{(k)}(s_2)\\\cdots\\V^{(k)}(s_n)\end{bmatrix}=\begin{bmatrix}\sum_{s'\in S}\Pr_{a}(s_1,s')\left[g(s_1,s')+\gamma\cdot V^{(k)}(s')\right]\\\sum_{s'\in S}\Pr_{a}(s_2,s')\left[g(s_2,s')+\gamma\cdot V^{(k)}(s')\right]\\\cdots\\\sum_{s'\in S}\Pr_{a}(s_n,s')\left[g(s_n,s')+\gamma\cdot V^{(k)}(s')\right]\end{bmatrix} \tag{6. 22}$$

求解该线性方程组，我们可以同时得到每一个状态s的估计值函数$V^{(k)}(s)$。进入 Step 3。

Step 3. [策略改进（Policy Improvement）] 基于当前对于最优值函数$V^{*}(s)$的估计值函数$V^{(k)}(s)$，采用贪心算法对每一种状态s尝试所有可能的动作以期改进策略

$$\pi^{(k+1)}(s)=\arg\max_{a}\left\{\sum_{s'\in S}\Pr_{a}(s,s')\left[g(s,s')+\gamma\cdot V^{(k)}(s')\right]\right\},\forall s\in S \tag{6.23}$$

进入 Step 4。

Step 4. [判断收敛性] 比较策略 $\pi^{(k)}$ 和 $\pi^{(k+1)}$ 对于任意状态 s 的行为差异，如果 $\left|\pi^{(k)}(s)-\pi^{(k-1)}(s)\right|\le\varepsilon,\forall s\in S$，$\varepsilon>0$ 为预设的阈值，则策略已经收敛，停止。如果策略没有收敛，$k:=k+1$，返回 Step 2。

定理 6.6. 策略迭代法 6.4 对于 Markov 多阶段决策过程问题可以在有限时间内停止。

证明：如果对于任意状态 s，满足 $\left|V^{(k)}(s)-V^{(k-1)}(s)\right|<\delta$，根据策略评估和策略改进公式，我们得到

$$
\begin{aligned}
&\left|V^{(k+1)}(s)-V^{(k)}(s)\right| \\
&=\left|\begin{array}{l}\sum\limits_{s'\in S}\Pr_a(s_1,s')\left[g(s_1,s')+\gamma\cdot V^{(k)}(s')\right] \\ -\sum\limits_{s'\in S}\Pr_a(s_1,s')\left[g(s_1,s')+\gamma\cdot V^{(k-1)}(s')\right]\end{array}\right| \\
&\le\left|\gamma\cdot\sum_{s'\in S}\Pr_a(s_1,s')\left[V^{(k)}(s')-V^{(k-1)}(s')\right]\right| \\
&\le\gamma\cdot\sum_{s'\in S}\Pr_a(s_1,s')\left|V^{(k)}(s')-V^{(k-1)}(s')\right| \\
&\le\gamma\cdot\sum_{s'\in S}\Pr_a(s_1,s')\delta \\
&=\gamma\delta
\end{aligned}
$$

进一步可知 $\left|V^{(k+m)}(s)-V^{(k+m-1)}(s)\right|<\gamma^m\delta$，由于 $\gamma\in(0,1)$，策略迭代法 6.4 所形成的映射是压缩映射，根据巴拿赫不动点定理，对于完备度量空间上的压缩映射，有且仅有一个不动点。因此迭代算法肯定收敛到唯一的策略 $V^*(s)$。其余分析同定理 6.4 证明。□

6.5　作业题

6.1 一只青蛙一次可以跳上 1 级台阶，也可以跳上 2 级台阶。求该青蛙跳上一个 10 级的台阶总共有多少种跳法。

6.2 假设我们使用 L、O、A 三种字符组成长度为 30 的字符串，这样的字符串不能出现连续 3 个或以上的 A，最多出现一个 L，请问有多少种这样的字符串？

6.3 给出两个单词 word1 和 word2，请计算出将 word1 转换成 word2 所使用的最少操作数。允许可以对一个单词进行如下三种操作：（1）插入一个字符；（2）删除一个字符；（3）替换一个字符。

示例：

输入：word1 = "horse", word2 = "rose"

输出：2

解释：

horse -> rorse（将'h'替换为'r'）

rorse -> rose（删除'r'）

6.4 你将获得 K 个鸡蛋，并可以使用一栋 N 层楼的建筑，楼层编号为 1 到 N。每个蛋的功能都是一样的，如果一个蛋碎了，你就不能再把它掉下去。你知道存在楼层 F，F 在 1 到 N 之间，任何从高于 F 的楼层落下的鸡蛋都会破碎，从 F 楼层或比它低的楼层落下的鸡蛋都不会破碎。

你可以进行如下操作：如果你有未破的鸡蛋，则可以把它从任一 1 到 N 之间的楼层 X 扔下，回到 1 楼检查该鸡蛋是否破碎。如果破碎，就取一个新的鸡蛋。

请问如何设计操作，使得我们可以用最少数量的鸡蛋确定 F 的值？

6.5 如果有 N 种物品和一个容量为 M 的背包，每种物品都就可以选择任意多个，每件第 i 种物品的价值为 V[i]，重量是 W[i]。请用动态规划求解：选哪些物品放入背包，可使得这些物品的总价值最大，并且体积总和不超过背包容量。

6.6 如果有 N 种物品和一个容量为 M 的背包。第 i 种物品最多有 n[i]件可用，每件第 i 种物品的价值为 V[i]，重量是 W[i]。请用动态规划求解：选哪些物品放入背包，可使得这些物品的总价值最大，并且体积总和不超过背包容量。

6.6 参考文献

[1] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503-516, 1954.

[2] R. Bellman, *Eye of the Hurricane*, World Scientific Publishing Co, 1984.

[3] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., 1994.

[4] A. Lew, H. Mauch, *Dynamic Programming: A Computational Tool*, Springer, 2007.

[5] R. E. Bellman, *Dynamic Programming*, Princeton University Press, 2010.

[6] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd edition, John Wiley & Sons, Inc., 2011.

[7] 《运筹学》教材编写组, *运筹学*, 第4版本科版, 清华大学出版社, 2013.

[8] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd edition, The MIT Press, 2018.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 4th edition, The MIT Press, 2022.