可行集的几何特性

现在让我们来考虑一下在标准形式下，线性规划可行集的几何特性，并引出正确的求解思路。首先我们约定(13.1)中的A是行满秩的。

$$\min_{x\in\mathbb{R}^n} f(x), \quad \text{s.t. } Ax=b, x\geq 0. \qquad (13.12)$$

由于A是$m\times n$矩阵，如果我们在A中确定m列，在指标$\{1,2,\cdots,n\}$中选择m个指标构成子集$\mathcal{B}$，那么我们相当于从A中切割出一个方阵：

$$B_{m\times m}=[A_i]_{i\in\mathcal{B}},$$

A_i是A的第i列。若B非奇异，则显然可以唯一确定一个点

$$x_B=B^{-1}b,$$

这里x_B相当于只对$i\in\mathcal{B}$中的分量给了赋值，对于那些没有被$\mathcal{B}$选中的分量，我们记为

$$\mathcal{N}=\{1,2,\cdots,n\}\backslash\mathcal{B},$$

它同样对应A的一些列，并可以组成一个矩阵

$$N_{m\times(n-m)}=[A_i]_{i\in\mathcal{N}},$$

以及对应的x分量组成的向量x_N。（$i\in\mathcal{N}$的x的分量组成，相当于现在x都可以分成x_B和x_N两部分。）对于x_N，我们令它的全部分量都为零，从而满足约束$x\geq 0$。但对于x_B，我们并不清楚它是否可行（是否各分量都大于等于零）。

（以上记号对A和x都按指标$\mathcal{B}$和$\mathcal{N}$做了一个划分，$\mathcal{B}\cup\mathcal{N}=\{1,2,\cdots,n\}$。）

现在我们规定：

定义 称(13.1)的可行点x是一个基础可行点(basic feasible point)，若存在指标集$\{1,2,\cdots,n\}$的子集$\mathcal{B}$满足：

1. $\mathcal{B}$包含m个指标；
2. $i\notin\mathcal{B}$，$x_i=0$；
3. $B_{m\times m}=[A_i]_{i\in\mathcal{B}}$非奇异。

而$\mathcal{B}$称为(13.1)的一个基(basis)，B称为基矩阵。

（$\mathcal{N}$有时称为“非基”，N有时称为“非基矩阵”。）

这里有一个严重的问题是，这样的可行点的存在性。下面这个定理回答了这个问题：

定理 13.2

1. 若(13.1)的可行域非空，则它至少有一个基础可行点；
2. 若(13.1)有解，则至少一个解是基础可行点；
3. 若(13.1)可行且有界，则它必有解。

这里我们不再重复讲义上的证明，而是直接讨论一下，从基础可行点的几何意义究竟是什么。实际上，在

$$Ax=b$$

中对A选取m列的操作我们是熟悉的，通常对于一个欠定线性系统，我们都会这么去做。因为系统欠定，我们会有多个解，事实上全部的解集构成一个线性空间，它的维数就是$n-m$。现在结合不等值约束$x \geq 0$，于是可行域Ω就是这个解空间落在$\mathbb{R}^n$中的"第一象限"的部分。大家可以想象一下，一个空间被一个n维的第一象限切割下的部分，实际上是n维空间的一个"凸多面体"（在x的正方向可以开放，但如果封闭，那么整个可行域就是一个凸多面体）。现在对于全部$i \in \mathcal{N}$，我们令$x_i = 0$，从而构成了基础可行点，这时我们发现：

1. 基础可行点是确定的单点，而不是空间。因为它的每一个分量都是唯一确定的；
2. 它是x_B空间和x_N空间的交点，而且是一个单点，如果我们将Ω视为$\mathbb{R}^n$空间中落在第一象限内的一个凸多面体，那么基础可行点实际上就是这个凸多面体和x_B空间相交的顶点。不同的基础可行点代表了不同的顶点；
3. 从上述几何意义出发，定理13.2的结论是显然的。同时，提示我们可以通过某种方式遍历全部的基础可行点（顶点）来寻找问题(13.1)的全局最优解（如果有，如果Ω在x正方向是开放的，同时f沿x的正方向是下降的，那么显然最优值是$-\infty$）。

单纯形方法(the simplex method)

现在我们来具体实现上面第3点讨论的方法。我们已经根据下标$\mathcal{B}$和$\mathcal{N}$划分了B和N，以及x_B和x_N。而判定一个点是否为全局最优解的条件是(13.4)，为方便讨论起见，我们进一步将(13.4)中的s和c也做对应的划分：

$$\begin{aligned} s_B &= [s_i]_{i \in \mathcal{B}}, \quad s_N = [s_i]_{i \in \mathcal{N}}, \\ c_B &= [c_i]_{i \in \mathcal{B}}, \quad c_N = [c_i]_{i \in \mathcal{N}}. \end{aligned}$$

结合(13.4)，我们有

$$Ax = Bx_B + Nx_N = b.$$

而作为一个算法的起点，我们需要第一个基础可行点做初值，或者说我们需要选择一个$\mathcal{B}$，然后我们就有

$$x_B = B^{-1}b, x_N = 0. \tag{13.18}$$

这里显然我们应该考虑那些使B在求逆时尽可能简单的$\mathcal{B}$，这个也可以通过算法手段得到，我们后面再讨论，目前我们只在所有实际可能中选取一个"看上去最好"的$\mathcal{B}$。

例 13.1

$$\begin{aligned} \min \quad & -4x_1 - 2x_2, \\ \text{s. t.} \quad & x_1 + x_2 + x_3 = 5, \\ & 2x_1 + \frac{1}{2}x_2 + x_4 = 8, \\ & x \geq 0. \end{aligned}$$

首先我们整理一下，这里

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & \frac{1}{2} & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 8 \end{bmatrix}, \quad c = (-4, -2, 0, 0)^T.$$

这里我们选择一个"尽可能好的"$\mathcal{B} = \{3, 4\}$，于是$\mathcal{N} = \{1, 2\}$，

$$\begin{aligned} B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 1 \\ 2 & \frac{1}{2} \end{bmatrix}, \\ c_B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad c_N = \begin{bmatrix} -4 \\ -2 \end{bmatrix}. \end{aligned}$$

而我们的计算将从第一个基本可行点开始：

$$x_B = B^{-1} \cdot b = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}, \quad x_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

即$x = (0, 0, 5, 8)^T$，注意此时

$$f(x) = c^T x = 0$$

即是当前目标值。接下去迭代应该设法找到新的基础可行点来降低这个值，如果它还不是最优的话。

对一个基础可行点x判定是否最优，我们直接采用(13.4)。首先为了满足互补性条件(13.4e)，我们令$s_B = 0$，然后来确定剩下的Lagrange乘子s_N和λ。由(13.4a)，

$$A^T \lambda + s = c \Rightarrow \begin{bmatrix} B^T \\ N^T \end{bmatrix} \lambda + \begin{bmatrix} s_B \\ s_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix}$$

得

$$B^T \lambda = c_B, \quad N^T \lambda + s_N = c_N. \qquad (13.19)$$

因为B是非奇异的，因此由第一个方程可得：

$$\lambda = (B^{-1})^T c_B. \qquad (13.20)$$

再通过(13.19)的第二个方程可以确定s_N：

$$s_N = c_N - N^T \lambda = c_N - (B^{-1} N)^T c_B. \qquad (13.21)$$

现在检查s_N，若$s_N \geq 0$，则我们已经找到了一组(x, λ, s)满足KKT条件(31.4)，也即，当前的x就是问题(13.1)的全局最优解。我们继续在例13.1中演示这个算法。现在，

$$\lambda = (B^{-1})^T c_B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

而

$$s_N = c_N - (B^{-1} N)^T c_B = \begin{bmatrix} -4 \\ -2 \end{bmatrix} - (\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & \frac{1}{2} \end{bmatrix}) \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -4 \\ -2 \end{bmatrix}.$$

（注意书上的例13.1从此处其开始有计算错误。）这里显然s_N的两个分量都是负数，也即沿这两个坐标方向，例13.1的问题有更小的值。对于一般问题而言，我们这里需要选择一个$\mathcal{B}$中的下标，和$\mathcal{N}$中的一个下标交换（这一过程有的教材称为“出基”和“入基”），从而得到一个能够取到更小值的基础可行点（新的$\mathcal{B}$和$\mathcal{N}$，新的x_B和x_N，新的x，$\cdots$等等），然后再做下一次KKT判定。到此，一个完整的迭代循环已经完成，如此循环，直到某一个基础可行点被确定为全局最优点（KKT点），或者能够判定目标函数值是可以取到负无穷的为止。现在要算法化这个循环，我们还要确定出基和入基。首先入基并没有什么特别确定的选择，理论上说，s_N的负分量的指标都可以作为入基，我们也只需要一个入基。比如目前的例13.1中，$s_1 = -4$，$s_2 = -2$，因此1或2均可以作为入基。但是为了算法统一起见，我们这里可以选择最小的一个分量，也就是$q = 1$作为入基（这样做未必是最优的，s_3更小并不意味着沿x_3方向能下降的更快，但确实有学者推荐这么做）。对一般问题，在这里有$q \in \mathcal{N}$，且$s_q < 0$。接下去继续确定出基p，也就是要从$\mathcal{B}$中选择一个指标来和出基交换。我们用x^+代表寻找中的新基础可行点，则x_B^+和x_B就只有一个分量的区别（我们暂时还不知道是哪一个），而且对x^+和x，由于约束条件，必有

$$Ax^+ = Bx_B^+ + A_q x_q^+ = Bx_B = Ax.$$

这里注意到对$i \in \mathcal{N} \backslash \{q\}$，总有$x_i^+ = 0$，$A_q$是$N$中对应入基的那一列。上式中间两式两边同乘以$B^{-1}$，有

$$x_B^+ + B^{-1} A_q x_q^+ = x_B \Rightarrow x_B^+ = x_B - B^{-1} A_q x_q^+. \qquad (13.22)$$

这个式子形象地告诉我们，新的目标点是旧的目标点沿x_q方向进行新的搜索，但要能移动这一点，我们必须在原本的$\mathcal{B}$中放弃一个约束，也就是出基p。令

$$d = B^{-1}A_q,$$

则

$$x_B^+ = x_B - x_q^+ d,$$

如果d至少存在一个正分量，那么在对应的坐标方向上，x_B的分量以$x_q d_i$的速率下降，$i \in \mathcal{B}$，$d_i > 0$。反之，若$\forall i \in \mathcal{B}$，有$d_i \leq 0$，则说明存在可行域的开放方向（到无穷远）上可以一直下降，即对应的线性规划无解。对前一种情况，考虑到x_B^+也必须满足各分量非负，即$\forall i \in \mathcal{B}$，$d_i > 0$，必须有

$$\frac{(x_B^+)_i}{d_i} = \frac{(x_B)_i}{d_i} - x_q^+ \geq 0, \quad \forall i \in \mathcal{B}, d_i > 0,$$

在上述各下降分量中，必有一个最先达到零（下降速率相同），那个就是我们选择的p。也即

$$p = \text{argmin}\{\frac{(x_B)_i}{d_i}, i \in \mathcal{B}, d_i > 0\}.$$

而同时

$$x_q^+ = \frac{(x_B)_p}{d_p},$$

以及

$$x_B^+ = x_B - x_q^+ d.$$

对例13.1，有

$$A_q = A_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \Rightarrow d = B^{-1}A_q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

d的各分量均正，而其中，

$$\frac{(x_B)_3}{d_3} = \frac{5}{1} = 5, \quad \frac{(x_B)_4}{d_4} = \frac{8}{2} = 4,$$

注意这里下标用的是$\mathcal{B}$，显然p应该选择4。以及$x_1^+ = 4$（即x_4从8降到0的同时，x_1从0升到4，从而实际同时完成了出基和入基的指标交换，与此同时，全部x_B的其他分量也会受到如下的影响）。

$$x_B^+ = x_B - d \cdot x_q^+ = \begin{bmatrix} 5 \\ 8 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot 4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

注意这里x_B^+中对应p的分量必须是零，否则必然有错误（x_3从5降到1。如果出基选3那么这一项就要降到0，从而导致x_4小于零，违背约束）

此时，新的基础可行点为

$$x^+ = (4, 0, 1, 0)^T.$$

由(13.22)，

$$c^T x^+ = c_B^T x_B^+ + c_q x_q^+ = c_B^T x_B - c_B^T B^{-1} A_q x_q^+ + c_q x_q^+. \quad (13.23)$$

以及(13.20)，$c_B^T B^{-1} = \lambda^T$。而(13.19)的第二个方程对$q \in \mathcal{N}$，有

$$A_q^T \lambda = c_q - s_q,$$

综合起来有

$$c_B^T B^{-1} A_q x_q^+ = \lambda^T A_q x_q^+ = (c_q - s_q) x_q^+,$$

代入(13.23)，得

$$c^T x^+ = c_B^T x_B - (c_q - s_q) x_q^+ + c_q x_q^+ = c^T x + s_q x_q^+. \quad (13.24)$$

即目标函数值只需做针对调整即可（不用整体重算）。对例13.1，即新目标函数值为

$$c^T x^+ = c^T x + s_q x_q^+ = 0 + s_q x_q^+ = -4 \cdot 4 = -16.$$

我们继续用此算法完成例13.1。目前$\mathcal{B} = \{3, 1\}$，即

$$B = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix},$$

注意为了和之前的次序及交换一致，这里第一列和第二列分别对应指标3和1。于是

$$B^T \lambda = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \lambda_3 \\ \lambda_1 \end{bmatrix} = c_B = \begin{bmatrix} 0 \\ -4 \end{bmatrix} \Rightarrow \lambda = \begin{bmatrix} 0 \\ -2 \end{bmatrix}.$$

而

$$s_N = \begin{bmatrix} s_4 \\ s_2 \end{bmatrix} = c_N - N^T \lambda = \begin{bmatrix} 0 \\ -2 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}.$$

仍然有负分量。于是继续选择$q = 2$，则

$$A_q = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}, Bd = A_q = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} d = \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix} \Rightarrow d = \begin{bmatrix} \frac{3}{4} \\ \frac{1}{4} \end{bmatrix},$$

于是

$$x_B./d = \begin{bmatrix} \frac{4}{3} \\ 16 \end{bmatrix},$$

这里借用一下Matlab的点运算，确定出基是$p = 3$（排第一位的是3）。即$\mathcal{B}$更新为$\{2, 1\}$。再由

$$x_B^+ = \begin{bmatrix} 1 \\ 4 \end{bmatrix} - \begin{bmatrix} \frac{3}{4} \\ \frac{1}{4} \end{bmatrix} \cdot \frac{4}{3} = \begin{bmatrix} 0 \\ \frac{11}{3} \end{bmatrix},$$

即x更新为

$$x = (\frac{11}{3}, \frac{4}{3}, 0, 0)^T.$$

此时目标函数值更新为：

$$f(x) = -16 + -1 \cdot \frac{4}{3} = -\frac{52}{3}.$$

再次检查

$$\begin{bmatrix} 1 & \frac{1}{2} \\ 1 & 2 \end{bmatrix} \lambda = \begin{bmatrix} -2 \\ -4 \end{bmatrix} \Rightarrow \lambda = \begin{bmatrix} -\frac{4}{3} \\ -\frac{4}{3} \end{bmatrix},$$

以及

$$s_N = c_N - N^T\lambda = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{4}{3} \\ -\frac{4}{3} \end{bmatrix} = \begin{bmatrix} \frac{4}{3} \\ \frac{4}{3} \end{bmatrix} \geq 0,$$

满足(13.4)，即$x^* = (\frac{11}{3}, \frac{4}{3}, 0, 0)^T$，$f(x^*) = -\frac{52}{3}$。

单纯形方法尽管繁琐，但是是一个确定性算法，也即我们可以用计算机程序实现全部过程。从而使得线性规划是可计算的。然而稍加分析我们就能发现，凸多面体的顶点个数，关于维数n是指数增加的，因此在最坏可能性下，单纯形方法是指数时间的（我们可以举这样的反例）。不过对于大多数实际问题，或者说，在平均可能性下，单纯形方法的复杂性期望仍然是多项式的。因此它在实际计算中，特别是n不是特别大的时候，也经常被采用。

第一个基础可行点

我们现在讨论一个遗留问题：如何确定单纯形方法的初值，也即第一个基础可行点。实际上，寻找第一个初值本身也是一个线性规划。对标准形式的Ω：

$$\left\{ \begin{array}{rcl} Ax & = & b, \\ x & \geq & 0, \end{array} \right.$$

这里$x = (x_1, x_2, \cdots, x_n)^T$。引入松弛变量$x_{n+1}, x_{n+2}, \cdots, x_{n+m}$，并将原依赖修改为：

$$\left\{ \begin{array}{rcl} [A \quad I]x & = & b, \\ x & \geq & 0, \end{array} \right.$$

现在$x = (x_1, x_2, \cdots, x_{n+m})^T$。于是$x$限制在$n$维是原问题的一个基础可行点当且仅当

$$\left\{ \begin{array}{ll} \min & f(x) = \sum_{i=n+1}^{n+m} x_i, \\ \mathrm{s.\,t.} & [A \quad I]x = b. \end{array} \right.$$

取到全局最优点$f(x^*) = 0$。此时x^*中必有$x_i = 0$，$i = n+1, n+2, \cdots, n+m$。而此问题，是一个自带一个初始基本可行解的线性规划问题，不难用单纯形法求解。有时，这样的单纯形法又称为两阶段单纯形方法。

主-对偶方法

对于达到一定规模的线性问题，单纯形方法的效率较低（事实上单纯形方法曾用于手算）。此时，直接采用迭代法有更好的效果。主要的思路是如果我们能够找到一个可行域内的初值，并用之前学习的求解无约束优化的连续迭代方法去求解，同时如果还能保证迭代过程不脱离可行域，那么就可以确保最后迭代到最优解。这种思路的关键是确保迭代序列在可行域内部，故又称“内点法(interior-point methods)”。

对线性规划问题的标准形式：

$$\min c^T x, \quad \mathrm{s.\,t.} Ax = b, x \geq 0, \qquad (14.1)$$

其中c和x是$\mathbb{R}^n$中的向量，b是$\mathbb{R}^m$中向量，且$m \times n$矩阵A是行满秩的，那么其对偶问题为：

$$\max b^T \lambda, \quad \mathrm{s.t.} A^T\lambda + s = c, s \geq 0. \qquad (14.2)$$

其中λ是$\mathbb{R}^m$中向量，而s是$\mathbb{R}^m$中向量。根据KKT条件(13.4)，上述两个问题的最优解都必须满足：

$$\begin{aligned} A^T\lambda + s &= c, && (14.3a) \\ Ax &= b, && (14.3b) \\ x_i s_i &= 0, \quad i = 1, 2, \cdots, n, && (14.3c) \\ (x, s) &\geq 0. && (14.3d) \end{aligned}$$

因此，接下去我们设法去寻找(x^*, λ^*, s^*)满足上述条件。这里我们将上述问题改写成如下方程形式：

$$F(x,\lambda,s) = \begin{bmatrix} A^T\lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0, \qquad (14.4a)$$

$$(x,s) \geq 0, \qquad (14.4b)$$

其中

$$X = \text{diag}(x_1, x_2, \cdots, x_n), \ S = \text{diag}(s_1, s_2, \cdots, s_n), \ (14.5)$$

以及$e = (1, 1, \cdots, 1)^T$。注意这里F是一个$\mathbb{R}^{2n+m}$到$\mathbb{R}^{2n+m}$维的非线性函数，我们可以用Newton迭代求解，而对于(14.4b)，我们要确保在迭代求解的过程中，有$x^k > 0$和$s^k > 0$（严格成立）。这里我们提出一个监察指标（对偶尺度，measure）

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{x^T s}{n}, \qquad (14.6)$$

显然，这个μ在迭代过程中越小越好，它提供了一个搜索方向或者步长的依据。现在我们来考虑如何迭代。首先对非线性方程组(14.4a)，其Newton迭代步为

$$J(x,\lambda,s)\begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -F(x,\lambda,s),$$

其中J是F的Jacobi矩阵。我们定义

$$r_b = Ax - b, \quad r_c = A^T\lambda + s - c, \qquad (14.7)$$

则整个Newton方程组为：

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}. \qquad (14.8)$$

于是一个完整的Newton迭代步为：

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha(\Delta x^k, \Delta\lambda^k, \Delta s^k).$$

这里$\alpha \in (0, 1]$。因为(x^k, λ^k, s^k)是可行点，根据连续性，只要α足够小，我们总能确保$(x^{k+1}, \lambda^{k+1}, s^{k+1})$可行。但这里如果我们想摆脱这种猥琐发育的算法，使得单步步长尽可能大，我们就需要对Newton方向做一个调整。比如，我们在考虑方向的时候就同时在非线性残量中加入μ的因素：

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe + \sigma\mu e \end{bmatrix}. \ (14.9)$$

这里$\sigma \in [0, 1]$称为中心化参数(centering parameter)。当$\sigma > 0$时，上述方程组得到的方向相比Newton方向一般可取到更大的步长。我们这里只是引入这样一个算法，接下去不再详细讨论σ和α的实际取法，以及其他可以考虑的模型方程。有兴趣的同学可以自己阅读参考书P396以后的第十四章部分，而Matlab的线性规划求解器中也提供了相应的内点法求解程序。