齐次自对偶算法

邓泽晓（译）

中山大学航空航天学院，深圳，中国

2024 年 12 月 17 日

摘要

本文主要关注齐次自对偶在线性规划 (LP) 上的嵌入使用，主要是为了解决内点算法初始点需为可行点的限制问题。常用内点算法初始化可以参考使用的方法有大 M 罚函数法，阶段一阶段二法和齐次自对偶法。齐次自对偶算法是路径跟踪内点算法在使用操作上的实现，它已被证明在算法复杂性上要优于阶段一阶段二法。本文只是关于此方法的一个学习笔记，主要参考文献 [1]。

关键词: 齐次自对偶法；内点算法；算法初始化

1 引文

本文主要分为 3 个部分。第 1 部分是介绍齐次自对偶的构造与嵌入；第 2 部分是关于给出算法的完整流程；而第 3 部分则是数值算例验证。

2 齐次自对偶嵌入

这里先从标准的线性规划问题 (LP)，然后转化到其自对偶的形式。下面先给出标准 LP 问题的及其对偶 LP 问题，

$$\begin{aligned} \text{maxmize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{2.1}$$

它的对偶 LP 问题为，

$$\begin{aligned} \text{minimize} \quad & b^T y \\ \text{subject to} \quad & A^T y \geq b \\ & y \geq 0 \end{aligned} \tag{2.2}$$

下面给出原始 LP 问题和对偶 LP 问题的组合形式。显然地，这个组合问题的解即为原始 LP 问题和对偶 LP 问题的解。其组合形式为，

$$\begin{array}{lrcrcrcl} \text{maximize} & & & & & 0 & & \\ \text{subject to} & & - & A^T y & + & c\phi & \leq & 0, \\ & Ax & & & - & b\phi & \leq & 0, \\ & -c^T x & + & b^T y & & & \leq & 0, \\ & & & x, & y, & \phi & \geq & 0. \end{array} \tag{2.3}$$

上式有以下几个特点：(1) 引入了一个新的变量 ϕ 和一个新的约束；(2) 总变量和总约束的个数均为 $n+m+1$；(3) 目标函数和所有约束的右手边均为零，对应问题称为齐次的 LP 问题 (Homogeneous LP)；(4) 约束矩阵是反对称的。拥有反对称约束矩阵的齐次 LP 问题称是自对偶的 (Self-Dual)，问题称为齐次自对偶线性规划问题 (Homogeneous Self-Dual Linear Programming, HSDP)。

从式 (2.3) 可以看出，当 $\phi>0$ 时，式 (2.3) 的解可以转换成式 (2.1) 和式 (2.2) 的解。下面假设 $\left(\bar{x},\bar{y},\bar{\phi}\right)$ 为问题 (2.3) 的最优解。下面分两种情况分析，一是假设 $\bar{\phi}>0$(后面会证明如此构造，无论什么时候式 (2.1) 和式 (2.2) 始终都有最优解)，以及令 $x^*=\bar{x}/\bar{\phi}$，$y^*=\bar{y}/\bar{\phi}$，代入式 (2.3) 的约束可得，

$$\begin{array}{rcrcrcl} & - & A^Ty^* & + & c & \leq & 0 \\ Ax^* & & & - & b & \leq & 0 \\ -c^Tx^* & + & b^Ty^* & & & \leq & 0 \end{array}$$

由上式第 3 个式子可知，弱对偶定理始终是满足。而当 $c^Tx^*=b^Ty^*$ 成立时，x^* 即为原始问题 (2.1) 的最优解，y^* 为对偶问题 (2.2) 的最优解。二是 $\bar{\phi}=0$ 时，对应即为问题不可行的情况，并且与原始问题不可行，对偶问题不可行或原始对偶问题皆不可行三种情况。

考虑原始问题 (2.1) 和对偶问题 (2.2)，若 $m=n$，$A=-A^T$ 和 $b=-c$，则称这样的线性规划问题是自对偶的。而当目标函数和约束的右手边都为零时，称这样的线性规划为齐次的。下面的分析都是基于齐次自对偶线性规划问题展开的，并且这里只分析原始问题的情况，因为对偶问题在齐次自对偶假设的条件下分析结果是一样的。设 z 为原始松弛变量，原始问题改写成，

$$\begin{array}{lcccccl} \text{maxmize} & & & 0 & & \\ \text{subject to} & Ax & + & z & = & 0 \\ & x, & & z & \geq & 0 \end{array} \tag{2.4}$$

对齐次自对偶线性规划，即式 (2.4) 的一些性质定理展开分析：

定理 2.1 *对于齐次自对偶问题 (2.4)，有下列性质：*
(1) 问题存在可行解且每一个可行解都是最优解；

(2) 可行解集没有内点。事实上，如果 (x,z) 是可行解，则 $z^Tx=0$。

证明 *(1) 平凡解 $(x,z)=(0,0)$ 显然是可行解。目标函数值是零，故所有可行解都是最优解。*

(2) 假设 (x,z) 是问题 (2.4) 的可行解。而 A 是反对称的，意味着对于任意一个 ξ 有 $\xi^TA\xi=0$ 成立。在式 $Ax+z=0$ 两边左乘 x^T，得 $x^Tz=0$。证毕。

定理2.1(2) 告诉我们齐次自对偶问题是没有中心路径的。

2.1 搜索方向及其步长

这里先定义两个函数描述当前解的不可行性 (Infeasibility) 和非互补性 (noncomplementarity)。解 (x,z) 对于求解问题的不可行程度用，

$$\rho\left(x,z\right)=Ax+z.$$

来度量；而 x 和 z 之间的非互补程度用，

$$\mu\left(x,z\right)=\frac{1}{n}x^Tz.$$

来度量。简写 $\rho(x,z)$ 和 $\mu(x,z)$ 为 ρ 和 μ。

而步长搜索方向 $(\Delta x, \Delta z)$ 选择为降低不可行性和非互补性的方向。转化此思想为数学表达式，这里需要引入一个转化因子 δ，$0 \le \delta \le 1$。需要每步迭代更新解 $(x+\Delta x, z+\Delta z)$ 的不可行性和非互补性都是当前解 (x,z) 的 δ 倍，即

$$A(x+\Delta x)+(z+\Delta z)=\delta(Ax+z),$$
$$(X+\Delta X)(Z+\Delta Z)e=\delta\mu(x,z)e.$$

显然，此式子是非线性的。但第二个式子的二阶小项对结果影响较小，去掉后可以得到求解搜索方向的线性方程组，

$$\begin{aligned} A\Delta x+\Delta z &= -(1-\delta)\rho(x,z), \quad (2.5)\\ Z\Delta x+X\Delta z &= \delta\mu(x,z)-XZe. \quad (2.6)\end{aligned}$$

设步长为 θ，迭代更新得到，

$$\bar{x}=x+\theta\Delta x, \qquad \bar{z}=z+\theta\Delta z.$$

由更新点得到不可行程度和非互补性程度值记为，

$$\bar{\rho}=\rho(\bar{x},\bar{z}), \qquad \bar{\mu}=\mu(\bar{x},\bar{z}).$$

下面定理给出所建立的搜索方向和步长的一些性质。

定理 2.2 以下式子恒成立：
(1) $\Delta z^T\Delta x=0$;
(2) $\bar{\rho}=(1-\theta+\theta\delta)\rho$;
(3) $\bar{\mu}=(1-\theta+\theta\delta)\mu$;
(4) $\bar{X}\bar{Z}e-\bar{\mu}e=(1-\theta)(XZe-\mu e)+\theta^2\Delta X\Delta Ze$.

证明 *(1)* 式 *(2.5)* 两边左乘 Δx^T 得，

$$\Delta x^TA\Delta x+\Delta x^T\Delta z=-(1-\delta)\Delta x^T\rho. \quad (2.7)$$

由矩阵 A 的反对称特性可知 $\Delta x^TA\Delta x=0$。因而式的左 *(2.7)* 手边简化为，

$$\Delta x^TA\Delta x+\Delta x^T\Delta z=\Delta x^T\Delta z.$$

式 *(2.7)* 中的 ρ 用其定义替代得，

$$-(1-\delta)\Delta x^T\rho=-(1-\delta)\Delta x^T(Ax+z).$$

而利用 A 的反对称性质重新排列 Δx^TAx 如下，

$$\Delta x^TAx=(Ax)^T\Delta x=x^TA^T\Delta x=-x^TA\Delta x.$$

通过整理可以得，

$$\Delta x^T\Delta z=-(1-\delta)\left(-x^TA\Delta x+z^T\Delta x\right). \quad (2.8)$$

右手边括号部分可使用式 (2.5) 替代得,

$$-x^T A\Delta x + z^T \Delta x = (1-\delta)\, x^T \rho + x^T \Delta z + z^T \Delta x. \tag{2.9}$$

利用 A 的反对称特性, 有下列式子,

$$x^T \rho = x^T (Ax + z) = x^T z.$$

而式 (2.9) 最后两项可以用式 (2.6) 两边左乘 e^T 后替代。这里再代入 μ 的定义式得,

$$z^T \Delta x + x^T \Delta z = \delta \mu n - x^T z = (\delta - 1)\, x^T z.$$

把上式代入式 (2.9) 得,

$$-x^T A\Delta x + z^T \Delta x = (1-\delta)\, x^T z + (\delta - 1)\, x^T z = 0.$$

从式 (2.8) 可知, $\Delta x^T \Delta z = 0$。定理2.2(1) 得证。(2) 从 $\bar{x}$ 和 $\bar{z}$ 的定义可知,

$$\begin{aligned} \rho &= A(x + \theta\Delta x) + (z + \theta \Delta z) \\ &= Ax + z + \theta (A\Delta x + \Delta z) \\ &= (1 - \theta + \theta\delta)\, \rho. \end{aligned}$$

(3) 同理可得,

$$\begin{aligned} \bar{x}^T \bar{z} &= (x + \theta\Delta x)^T (z + \Delta z) \\ &= x^T z + \theta \left(z^T \Delta x + x^T \Delta z \right) + \theta^2 \Delta z^T \Delta x. \end{aligned}$$

从 (1) 和式 (2.6), 我们可得,

$$\bar{x}^T \bar{z} = x^T z + \theta \left(\delta \mu n - x^T z \right).$$

因此,

$$m\bar{u} = \frac{1}{n} \bar{x}^T \bar{z} = (1 - \theta)\, \mu + \theta \delta \mu.$$

(4) 根据 $\bar{x}$ 和 $\bar{z}$ 的定义和定理2.2(3) 可得,

$$\begin{aligned} \bar{X}\bar{Z}e - \bar{\mu} e &= (X + \theta \Delta X)(Z + \theta\Delta Z)\, e - (1 - \theta + \theta\delta)\, \mu e \\ &= XZe + \theta (Z\Delta x + X \Delta z) + \theta^2 \Delta X \Delta Z e - (1 - \theta + \theta\delta)\, \mu e. \end{aligned}$$

而上式右边第二项代入式 (2.6) 即可得到 (4) 的形式。证毕。

2.2 预估校正算法

在前面铺垫好理论基础之后，这里准备介绍一个算法。令,

$$\mathcal{N}(\beta) = \{(x, z) > 0 : \| XZe - \mu(x, z)\, e \| \leq \beta \mu (x, z)\}.$$

这里只处理 $\mathcal{N}(1/4)$ 和 $\mathcal{N}(1/2)$ 的情况。从定义可得当 $\beta < \beta'$ 时有 $\mathcal{N}(\beta) \subseteq \mathcal{N}\left(\beta'\right)$。而 $\mathcal{N}(0)$ 就是标准中心路径点 (x, z) 构成的集合。

算法可以转化成两种类型的迭代步。首先在奇数迭代步中执行的是预估步。假设,

$$(x, z) \in \mathcal{N}(1/4).$$

在预估步中搜索方向参数选择为 $\delta=0$，步长参数选择不能使点 (x,z) 超出 $\mathcal{N}(1/2)$，故选择步长为，

$$\theta=\max\{t:(x+t\Delta x,z+t\Delta z)\in\mathcal{N}(1/2)\}. \tag{2.10}$$

而在偶数迭代步中执行的是校正步。假设，

$$(x,z)\in\mathcal{N}(1/2).$$

(保证预估步步长参数的选择)。在校正步中搜索方向参数选择为 $\delta=1$，步长参数选择为 $\theta=1$。

接下来的定理说明在算法的每执行一步预估步的时候 μ 值都会减少，而在执行校正步的时候保持不变。

定理 2.3 以下结论恒成立：
(1) 在每一步预估步执行完之后，$(\bar{x},\bar{z})\in\mathcal{N}(1/2)$ 并且 $\bar{\mu}=(1-\theta)\mu$；
(2) 在每一步校正步执行完之后，$(\bar{x},\bar{z})\in\mathcal{N}(1/4)$ 并且 $\bar{\mu}=\mu$；

证明 *(1)* 由定理2.2(3) 得，在执行预估步时 $\delta=0$，$\bar{\mu}$ 值等式成立的结果是显然的。而 $(\bar{x},\bar{z})\in\mathcal{N}(1/2)$ 则可由预估步的步长参数 θ 的选择确定。

(2) 在证明之前，我们要引入几个变量推导一些结果，

$$\begin{aligned}
p &= X^{-1/2}Z^{1/2}\Delta x,\\
q &= X^{1/2}Z^{-1/2}\Delta z,\\
r &= p+q\\
&= X^{-1/2}Z^{-1/2}(Z\Delta x+X\Delta z)\\
&= X^{-1/2}Z^{-1/2}(\delta\mu e-XZe).
\end{aligned} \tag{2.11}$$

通过上面定义的变量可以推出以下引理。

引理 2.1 以下结论恒成立：
(1) $\| PQe \|\leq\frac{1}{2}\| r \|^2$；
(2) 如果 $\delta=0$，则 $\| r \|^2=n\mu$；
(3) 如果 $\delta=1$ 并且 $(x,z)\in\mathcal{N}(\beta)$，则 $\| r \|^2\leq\beta^2\mu/(1-\beta)$

证明 *(1)* 从定理2.2(1) 知 $p^Tq=\Delta x^T\Delta z=0$，因此，

$$\| r \|^2=\| p+q \|^2=p^Tp+2p^Tq+q^Tq=\sum_j\left(p_j^2+q_j^2\right).$$

进而，

$$\begin{aligned}
\| r \|^4 &= \left(\sum_j\left(p_j^2+q_j^2\right)\right)^2\\
&\geq\sum_j\left(p_j^2+q_j^2\right)^2\\
&=\sum_j\left(\left(p_j^2-q_j^2\right)^2+4p_j^2q_j^2\right)\\
&\geq 4\sum_j p_j^2q_j^2\\
&=4\| PQe \|^2.
\end{aligned}$$

上式开方后引理2.1(1) 得证。

(2) 把 $\delta=0$ 代入式 (2.11) 得 $r=-X^{1/2}Z^{1/2}e$，故有 $\| r \|^2 = z^T x = n\mu$。

(3) 假设 $(x,z)\in\mathcal{N}(\beta)$。由 $\mathcal{N}(\beta)$ 的定义式可推出 $|x_j z_j-\mu|\le\beta\mu$。此不等式等效于，

$$(1-\beta)\,\mu \le x_j z_j\,(1+\beta)\,\mu. \tag{2.12}$$

现在把 $\delta=1$ 代入式 (2.11) 得，

$$\| r \|^2 = \sum_j \frac{(x_j z_j-\mu)^2}{x_j z_j}.$$

因此，代入式 (2.12) 可得，

$$\| r \|^2 \le \frac{1}{(1-\beta)\,\mu}\sum_j (x_j z_j-\mu)^2 .$$

因为 $(x,z)\in\mathcal{N}(\beta)$，所以上式求和的上界即为 $\beta^2\mu^2$，代入上式，引理2.1(3) 得证。

下面回到证明定理2.3(2)。因为校正步中，$\theta=1$，代入定理2.2(4) 可得 $\bar{X}\bar{Z}e-\bar{\mu}e=\Delta X\Delta Ze=PQe$。使用引理2.1的 (1) 和 (3) 得，

$$\begin{aligned}\| \bar{X}\bar{Z}e-\bar{\mu}e \| &= \| PQe \| \\ &\le \frac{1}{2}\| r \|^2 \\ &\le \frac{1}{2}\frac{(1/2)^2}{1-1/2}\mu \\ &= \frac{1}{4}\mu.\end{aligned} \tag{2.13}$$

这里还需要证实 $(\bar{x},\bar{z})>0$。对于 $0\le t\le 1$，令，

$$x(t)=x+t\Delta x,\quad z(t)=z+t\Delta z,\quad \mu(t)=\mu(x(t),z(t)).$$

由定理2.2(4) 可得，

$$X(t)\,Z(t)-\mu(t)\,e=(1-t)\,(XZe-\mu e)+t^2\Delta X\Delta Ze.$$

观察发现上式右手边是两个向量的和。由三角不等式知，两个向量和的长度小于两个向量长度的和，即，

$$\| X(t)\,Z(t)-\mu(t)\,e \| \le (1-t)\,\| XZe-\mu e \| + t^2\,\| \Delta X\Delta Ze \| . \tag{2.14}$$

由 $(x,z)\in\mathcal{N}$ 得 $\| XZe-\mu e \|\le\mu/2$，并且从式 (2.13) 可得 $\| \Delta X\Delta Ze \| = \| PQe \| \le \mu/4$，把它们都代入式 (2.14) 替换上界，

$$\| X(t)\,Z(t)\,e-\mu(t)\,e \| \le (1-t)\,\frac{\mu}{2}+t^2\frac{\mu}{4}\le\frac{\mu}{2}. \tag{2.15}$$

(因为 $t^2\le t$ 和 $\mu/4\le\mu/2$，右边第二个不等号显然成立。)

现在，从式 (2.15) 拆出来考虑具体的第 j 项，

$$x_j(t)\,z_j(t)-\mu(t)\ge-\frac{\mu}{2}.$$

又因为 $\delta=1$，由定理2.2(3) 得 $\mu(t)=\mu$ 对于所有的 t 都成立。故上式又可以写成，

$$x_j(t)\,z_j(t)\ge\frac{\mu}{2}>0. \tag{2.16}$$

上式对于所有 $0\le t\le 1$ 都有 $x_j(t)>0$ 和 $z_j(t)>0$。令 $t=1$，得 $\bar{x}_j(t)>0$ 和 $\bar{z}_j(t)>0$。因为 j 的取值是任意的，$(\bar{x},\bar{z})>0$。从式 (2.15) 可知，$(\bar{x},\bar{z})\in\mathcal{N}(1/4)$。

2.3 收敛性分析

前面的定理给出了预估校正算法的框架。接下来的定理给出每一步预估步的一个下边界。

定理 2.4 *在每一步预估步中，$\theta \geq \frac{1}{2\sqrt{n}}$。*

证明 预估步中，$(x, z) \in \mathcal{N}(1/4)$ 并且 $\delta = 0$，因而有，

$$\| XZe - \mu e \| \leq \frac{\mu}{4}.$$

而从引理2.1的 (1) 和 (2) 可得，

$$\| \Delta X \Delta Z e \| = \| PQe \| \leq \frac{1}{2} \| r \|^2 = \frac{1}{2} n\mu.$$

代入式 (2.14) 可得到如下式，

$$\| X(t) Z(t) e - \mu(t) e \| \leq (1-t) \frac{\mu}{4} + t^2 \frac{\mu}{2}.$$

现在，令 $t \leq (2\sqrt{n})^{-1}$，$t^2 n/2 \leq 1/8$。对于 $n \geq 2$ 时，$t \leq 1/2$ 有，

$$\begin{aligned}
\| X(t) Z(t) e - \mu(t) e \| &\leq (1-t) \frac{\mu}{4} + \frac{\mu}{8} \\
&\leq (1-t) \frac{\mu}{4} + (1-t) \frac{\mu}{4} \\
&= (1-t) \frac{\mu}{2} \\
&= \frac{\mu(t)}{2}.
\end{aligned}$$

根据邻域的定义可得 $(x(t), z(t)) \in \mathcal{N}(1/2)$。又因为任意一个 $t \leq (2\sqrt{n})^{-1}$，由式 (2.10) 可知 $\theta \geq (2\sqrt{n})^{-1}$。证毕。

令 $\left(x^{(k)}, z^{(k)}\right)$ 为第 k 次迭代的解，同时令，

$$\rho^{(k)} = \rho\left(x^{(k)}, z^{(k)}\right), \quad \mu^{(k)}\left(x^{(k)}, z^{(k)}\right).$$

我们的目标是证明算法的收敛性。即需要证明 k 在趋于无穷的时候，$\rho^{(k)}$ 和 $\mu^{(k)}$ 趋于零。这里算法初始点选择为 $x^{(0)} = z^{(0)} = e$，因而 $\mu^{(0)} = 1$。由定理2.3可知，在每一次偶数迭代步完成之后，也就是 $2k$，有下面式子成立，

$$\mu^{(2k)} \leq \left(1 - \frac{1}{2\sqrt{n}}\right)^k. \tag{2.17}$$

而校正步不会改变 μ 值，故

$$\mu^{(2k-1)} = \mu^{(2k)}.$$

故当 $k \to \infty$，有，

$$\lim_{k \to \infty} \mu^{(k)} = 0.$$

现在考虑 $\rho^{(k)}$。从定理2.2的 (2) 和 (3) 可知，我们可以使用不可行程度的下降量来跟踪非互补性程度的下降量，即，

$$\rho^{(k)} = \mu^{(k)} \rho^{(0)}.$$

因此当 $\mu^{(k)}$ 趋于零时，$\rho^{(k)}$ 同样趋于零。

事实上，还可以得到接下来的定理。

定理 2.5 极限 $x^* = \lim_{k\to\infty} x^{(k)}$ 和 $z^* = \lim_{k\to\infty} z^{(k)}$ 存在，并且 (x^*, z^*) 即是最优解。此外，向量 x^* 和 z^* 之间严格互补。意味着，对每一个 j，$x_j^* z_j^* = 0$，但可能是 $x_j^* > 0$ 或 $z_j^* > 0$。

关于上述定理的证明是相当有技术性的。它的核心观点是证明序列收敛。证明后续会在 SDP 的齐次自对偶嵌入时补充，其中证明类似。下面只是给出一个重要定理，

定理 2.6 存在正常数 $c_1, c_2, ..., c_n$，当 $(x, z) \in \mathcal{N}(\beta)$ 时，有 $x_j + z_j \geq c_j > 0$，$j = 1, 2, ..., n$。

证明略。

2.4 算法复杂性

工程上，我们不可能执行无限次迭代，而是预先设置一个阈值，当 $\mu^{(k)}$ 小于这个阈值时即停止迭代。这个阈值记为 2^{-L}，L 是一个数。我们通常希望阈值设置为 10^{-8}，对应的 $L \approx 26$。

由式 (2.17) 可得，偶数次迭代时，

$$\mu^{(2k)} \leq \left(1 - \frac{1}{2\sqrt{n}}\right)^k \leq 2^{-L}.$$

两边取对数后得，

$$k \geq \frac{L}{-\ln\left(1 - \frac{1}{2\sqrt{n}}\right)}.$$

又因为 $-\ln(1-x) \leq x$，有，

$$k \geq 2L\sqrt{n} \geq \ln \frac{L}{-\left(1 - \frac{1}{2\sqrt{n}}\right)}.$$

故最多有 $k \geq 2L\sqrt{n}$ 次迭代。而这里使用的 k 只代表一半的迭代步数。因而至多迭代 $4L\sqrt{n}$ 步，μ 值会下降到设定阈值以下。这也说明预估矫正算法是一个多项式算法。所谓的多项式算法就是指可以通过一个关于 n 的多项式迭代步数后可以到达任意想要精度的算法 (注意：这里不是指 $4L\sqrt{n}$ 自己是一个多项式，而是指它是一个关于 n 的线性方程的边界)。

3 算法完整流程

在接下来一节，我们将开发完整算法解决齐次自对偶问题 (2.3)。

3.1 分情况讨论

回到齐次自对偶问题的标准形式，给对应约束添加的松弛变量 z，w 和 ψ，

$$\begin{array}{lrrrrrrl}
\text{maximize} & & & & & 0 & & \\
\text{subject to} & & - & A^T y & + & c\phi & + & z \leq 0, \\
 & Ax & & & - & b\phi & + & w \leq 0, \\
 & -c^T x & + & b^T y & & & + & \psi \leq 0, \\
 & x, & & y, & & \phi, & & z, \; w, \; \psi \geq 0.
\end{array} \tag{3.1}$$

由定理A.1可知，如果对于所有的 j 有 $\bar{x}_j + \bar{z}_j > 0$，对于所有的 i 有 $\bar{y}_i + \bar{w}_i > 0$，且 $\bar{\phi} + \bar{\psi} > 0$，则称这样的可行解 $(\bar{x}, \bar{y}, \bar{\phi}, \bar{z}, \bar{w}, \bar{\psi})$ 是严格互补的。

接下来的定理总结和拓展第2节的讨论内容。

定理 3.1 假设 $\left(\bar{x},\bar{y},\bar{\phi},\bar{z},\bar{w},\bar{\psi}\right)$ 是问题 *(3.1)* 的一个严格互补可行解 *(*因此也是最优解*)*，
(1) 如果 $\bar{\phi}>0$，则 $x^{*}=\bar{x}/\bar{\phi}$ 是原始问题 *(2.1)* 的最优解，且 $y^{*}=\bar{y}/\bar{\phi}$ 是对偶问题 *(2.2)* 的最优解；
(2) 如果 $\bar{\phi}=0$，则可能是 $c^{T}\bar{x}>0$ 或 $b^{T}\bar{y}<0$。
  (a) 如果 $c^{T}\bar{x}>0$，则对偶问题是不可行问题；
  (b) 如果 $b^{T}\bar{y}<0$，则原始问题是不可行问题。

证明 *(1)* 已经在第2节证明。下面着重证明 *(2)*。假设 $\bar{\phi}=0$，因为严格互补，$\bar{\psi}>0$。因此 $\bar{x}$ 和 $\bar{y}$ 满足，

$$\begin{aligned}A^{T}\bar{y}&\geq 0,\\ A\bar{x}&\leq 0,\\ b^{T}\bar{y}&<c^{T}\bar{x}.\end{aligned}\tag{3.2}$$

从最后一个不等式可以看出不可能会出现 $b^{T}\bar{y}\geq 0$ 和 $c^{T}\bar{x}\leq 0$ 的情况，意味着可能的情况是 $c^{T}\bar{x}>0,b^{T}\bar{y}<0$ 或以上两种情况同时出现。不失一般性，假设 $c^{T}\bar{x}>0$，之后我们会通过反证法证实这种条件下对偶问题是不可行问题。假设存在向量 $y^{0}\geq 0$，有

$$A^{T}y^{0}\geq c.\tag{3.3}$$

因为 $\bar{x}\geq 0$，在上式两边同时乘以 $\bar{x}^{T}$ 不改变不等式的符号，

$$\bar{x}^{T}A^{T}y^{0}\geq\bar{x}^{T}c>0.$$

现在得出上不等式左边严格为正，但从式 *(3.2)*，在非负 y^{0} 条件下，上式左边不可能能为正，即

$$\bar{x}^{T}A^{T}y^{0}=\left(A\bar{x}\right)^{T}y^{0}\leq 0.$$

结果与假设矛盾，故对偶问题必为不可行问题。同理可证定理*3.1(2b)*。证毕。

3.2 简化 KKT 系统方程

我们这里先讨论一下 KKT 系统方程式 (2.5) 和 (2.6)。在预估校正算法里，解决这个系统方程是最消耗时间的。首先解出 Δz 得，

$$\begin{aligned}\Delta z&=X^{-1}\left(-Z\Delta x+\delta\mu e\right)\\&=-X^{-1}Z\Delta x+\delta\mu X^{-1}e-z,\end{aligned}\tag{3.4}$$

把上式代入 (2.5) 得到简化的 KKT 系统 (Reduced KKT System) 方程，

$$\left(A-X^{-1}Z\right)\Delta x=-\left(1-\delta\right)\rho+z-\delta\mu X^{-1}e.$$

同理，还可以得到

$$\Delta z=-X^{-1}Z\Delta x+\delta\mu X^{-1}e-z,\tag{3.5}$$

$$\Delta w=-Y^{-1}W\Delta y+\delta\mu Y^{-1}e-w,\tag{3.6}$$

$$\Delta\psi=-\frac{\psi}{\phi}\Delta\phi+\delta\mu/\phi-\psi.\tag{3.7}$$

以及对应的简化 KKT 系统方程组。而简化 KKT 系统方程组牵涉向量的不可行性和非互补性，故这里把它分成 3 部分，

$$\begin{bmatrix}\sigma\\ \rho\\ \gamma\end{bmatrix}=\begin{bmatrix} & -A^{T} & c\\ A & & -b\\ -c^{T} & b^{T} & \end{bmatrix}\begin{bmatrix}x\\ y\\ \phi\end{bmatrix}+\begin{bmatrix}z\\ w\\ \psi\end{bmatrix}=\begin{bmatrix}-A^{T}y+c\phi+z\\ Ax-b\phi+w\\ -c^{T}x+b^{T}y+\psi\end{bmatrix}.$$

给出式 (2.3) 对应的简化 KKT 系统方程组为，

$$\begin{bmatrix} -X^{-1}Z & -A^T & c \\ A & -Y^{-1}W & -b \\ -c^T & b^T & -\psi/\phi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\phi \end{bmatrix} = \begin{bmatrix} \hat{\sigma} \\ \hat{\rho} \\ \hat{\gamma} \end{bmatrix}, \tag{3.8}$$

式中，

$$\begin{bmatrix} \hat{\sigma} \\ \hat{\rho} \\ \hat{\gamma} \end{bmatrix} = \begin{bmatrix} -(1-\delta)\,\sigma + z - \delta\mu X^{-1}e \\ -(1-\delta)\,\rho + w - \delta\mu Y^{-1}e \\ -(1-\delta)\,\gamma + \psi - \delta\mu/\phi \end{bmatrix}.$$

这个系统方程组不是对称的，可以使用通用求解器进行求解，但方程的特殊结构会被忽略掉。为利用这种结构，我们分两步求解这个方程。使用前两个方程同时用 $\Delta\phi$ 表示出 Δx 和 Δy，

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -X^{-1}Z & -A^T \\ A & -Y^{-1}W \end{bmatrix}^{-1} \left(\begin{bmatrix} \hat{\sigma} \\ \hat{\rho} \end{bmatrix} - \begin{bmatrix} c \\ -b \end{bmatrix} \Delta\phi \right).$$

简写成，

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} - \begin{bmatrix} g_x \\ g_y \end{bmatrix} \Delta\phi, \tag{3.9}$$

式中，

$$f = \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \quad g = \begin{bmatrix} g_x \\ g_y \end{bmatrix}.$$

细心发现，这里需要求解两组方程组，

$$\begin{bmatrix} -X^{-1}Z & -A^T \\ A & -Y^{-1}W \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \hat{\sigma} \\ \hat{\rho} \end{bmatrix}, \qquad \begin{bmatrix} -X^{-1}Z & -A^T \\ A & -Y^{-1}W \end{bmatrix} \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} c \\ -b \end{bmatrix}.$$

使用式 (3.9) 消掉式 (3.8) 最后一个方程的 Δx 和 Δy 得，

$$\begin{bmatrix} -c^T & b^T \end{bmatrix} \left(\begin{bmatrix} f_x \\ f_y \end{bmatrix} - \begin{bmatrix} g_x \\ g_y \end{bmatrix} \Delta\phi \right) - \frac{\psi}{\phi}\Delta\phi = \hat{\gamma}.$$

可解出 $\Delta\phi$，

$$\Delta\phi = \frac{c^T f_x - b^T f_y + \hat{\gamma}}{c^T g_x - b^T g_y - \psi/\phi}.$$

我们可以发现简化 KKT 系统方程组可通过求解两个方程组得到 f 和 g。这两个方程组包含相同的系数矩阵。调整方程位置可得拟正定系统方程组 (Quasidefinite System)，如

$$\begin{bmatrix} -Y^{-1}W & A^T \\ A & X^{-1}Z \end{bmatrix} \begin{bmatrix} g_y \\ g_x \end{bmatrix} = \begin{bmatrix} -b \\ -c \end{bmatrix}.$$

这只是个简单的代数处理办法，但在这类型问题中却经常遇到。

齐次自对偶线性规划问题完整的算法流程如伪代码1所示。

Algorithm 1: 预估校正算法

Input: A, b, c, f, 线性规划参数; ϵ, 停止阈值.

Output: $(x^*, y^*, \phi, z^*, w^*, \psi^*)$, 最优解; f^*, 最优目标函数值.

Initialize , $(x, y, \phi, z, w, \psi) = (e, e, 1, e, e, 1)$;

while 不是最优 **do**

$\mu = \left(z^T x + w^T y + \psi\phi\right) / (n + m + 1)$;

$\delta = \begin{cases} 0, & \textit{on odd iterations} \\ 1, & \textit{on even iterations} \end{cases}$;

$\hat{\sigma} = -(1 - \delta)\,\sigma + z - \delta\mu X^{-1} e$;

$\hat{\rho} = -(1 - \delta)\,\rho + w - \delta\mu Y^{-1} e$;

$\hat{\gamma} = -(1 - \delta)\,\gamma + \psi - \delta\mu/\phi$;

求解两个 $(n + m) \times (n + m)$ 的拟正定系统方程组：

$\begin{bmatrix} -Y^{-1}W & A \\ A^T & X^{-1}Z \end{bmatrix} \begin{bmatrix} f_y \\ f_x \end{bmatrix} = \begin{bmatrix} \hat{\rho} \\ -\hat{\sigma} \end{bmatrix}, \begin{bmatrix} -Y^{-1}W & A \\ A^T & X^{-1}Z \end{bmatrix} \begin{bmatrix} g_y \\ g_x \end{bmatrix} = \begin{bmatrix} -b \\ -c \end{bmatrix}$;

$\Delta\phi = \left(c^T f_x - b^T f_y + \hat{\gamma}\right) / \left(c^T g_x - b^T g_y - \psi/\phi\right)$;

$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} - \begin{bmatrix} g_x \\ g_y \end{bmatrix} \Delta\phi$;

$\Delta z = -X^{-1} Z \Delta x + \delta\mu X^{-1} e - z$;

$\Delta w = -Y^{-1} W \Delta y + \delta\mu Y^{-1} e - w$;

$\Delta\psi = -\frac{\psi}{\phi}\Delta\phi + \delta\mu/\phi - \psi$;

$\theta = \begin{cases} \max\left\{t : (x(t), ..., \psi(t)) \in \mathcal{N}\left(1/2\right)\right\}, & \textit{on odd iterations} \\ 1, & \textit{on even iterations} \end{cases}$;

$x \leftarrow x + \theta\Delta x, \quad z \leftarrow z + \theta\Delta z$;

$y \leftarrow y + \theta\Delta y, \quad w \leftarrow w + \theta\Delta w$;

$\phi \leftarrow \phi + \theta\Delta\phi, \quad \psi \leftarrow \psi + \theta\Delta\psi$;

end

4 数值算例

下面给出几个算例进行验证

$$\begin{array}{rrrrrcl} \text{maximize} & x_1 & -2x_2 & +x_3 & & & \\ \text{subject to} & x_1 & +x_2 & -2x_3 & +x_4 & = & 0, \\ & 2x_1 & -x_2 & +4x_3 & & \leq & 0, \\ & -x_1 & +2x_2 & -4x_3 & & \leq & 0, \\ & x_j & \geq 0, & j = 1, & \cdots, & 4. & \end{array} \tag{4.1}$$

上式 LP 问题的最优解为 $(x_1, x_2, x_3, x_4) = (0, 12, 5, 8)$，目标函数值为 $f_{min} = -19$。

输入命令：

```
A = [1, 1, -2, 1; -1, -1, 2, -1; 2, -1, 4, 0; -1, 2, -4, 0];
b = [10, -10, 8, 4]';
c = [1, -2, 1, 0]';
c = -c;
[x, y, phi, z, w, psi, fval, status] = hsd(A, b, c);
x
f = -fval
```

```
HSD method
----------------------------------------------------------------------------
|                 PRIMAL      |                  DUAL      |          |
iter    |      Obj Value      Infeas |      Obj Value      Infeas |       mu |
----------------------------------------------------------------------------
1    |  0.0000000e+00     1.4e+01 |  1.2000000e+01     2.4e+00 |  1.0e+00 |
2    |  2.1375661e+00     9.0e+00 |  8.8492063e+00     1.6e+00 |  3.3e-01 |
3    |  2.1375661e+00     9.0e+00 |  8.8492063e+00     1.6e+00 |  3.3e-01 |
4    |  5.9525764e+00     5.5e+00 |  9.5098523e+00     9.5e-01 |  1.5e-01 |
5    |  5.9525764e+00     5.5e+00 |  9.5098523e+00     9.5e-01 |  1.5e-01 |
6    |  1.1902398e+01     2.8e+00 |  1.3107519e+01     4.7e-01 |  5.4e-02 |
7    |  1.1902398e+01     2.8e+00 |  1.3107519e+01     4.7e-01 |  5.4e-02 |
8    |  1.7308765e+01     7.5e-01 |  1.7366377e+01     1.3e-01 |  1.2e-02 |
9    |  1.7308765e+01     7.5e-01 |  1.7366377e+01     1.3e-01 |  1.2e-02 |
10    |  1.8823847e+01     8.8e-02 |  1.8808974e+01     1.5e-02 |  1.4e-03 |
11    |  1.8823847e+01     8.8e-02 |  1.8808974e+01     1.5e-02 |  1.4e-03 |
12    |  1.8990025e+01     5.1e-03 |  1.8988860e+01     8.8e-04 |  8.0e-05 |
13    |  1.8990025e+01     5.1e-03 |  1.8988860e+01     8.8e-04 |  8.0e-05 |
14    |  1.8999497e+01     2.6e-04 |  1.8999438e+01     4.4e-05 |  4.0e-06 |
15    |  1.8999497e+01     2.6e-04 |  1.8999438e+01     4.4e-05 |  4.0e-06 |
16    |  1.8999975e+01     1.3e-05 |  1.8999972e+01     2.2e-06 |  2.0e-07 |
17    |  1.8999975e+01     1.3e-05 |  1.8999972e+01     2.2e-06 |  2.0e-07 |
18    |  1.8999999e+01     6.4e-07 |  1.8999999e+01     1.1e-07 |  1.0e-08 |
19    |  1.8999999e+01     6.4e-07 |  1.8999999e+01     1.1e-07 |  1.0e-08 |
20    |  1.9000000e+01     3.2e-08 |  1.9000000e+01     5.5e-09 |  5.1e-10 |
21    |  1.9000000e+01     3.2e-08 |  1.9000000e+01     5.5e-09 |  5.1e-10 |
22    |  1.9000000e+01     1.6e-09 |  1.9000000e+01     2.8e-10 |  2.5e-11 |
23    |  1.9000000e+01     1.6e-09 |  1.9000000e+01     2.8e-10 |  2.5e-11 |
24    |  1.9000000e+01     8.1e-11 |  1.9000000e+01     1.4e-11 |  1.3e-12 |
25    |  1.9000000e+01     8.1e-11 |  1.9000000e+01     1.4e-11 |  1.3e-12 |

x =
```

```
0.0000
12.0000
5.0000
8.0000


f =

-19.0000
```

A 附录

定理 A.1 如果原始和对偶 LP 问题都有可行解，则原始问题存在可行解 $(\bar{x}, \bar{w})$ 和对偶问题存在可行解 $(\bar{y}, \bar{z})$，使得 $\bar{x} + \bar{z} > 0$，$\bar{y} + \bar{w} > 0$。

证明略。

定理 A.2 (严格互补松弛定理) 如果线性规划问题有最优解，则原始问题最优解 (x^*, w^*) 和对偶问题最优解 (y^*, z^*) 使得 $x^* + z^* > 0$，$y^* + w^* > 0$。

证明略。

B 源代码

```
function [x, y, phi, z, w, psi, fval, status] = hsd(A, b, c)
% function : Implementation of the Homogeneous Self-Dual Method
%
% primal problem:
%   max   c'x
%   s.t.  Ax <= b
%         x >= 0
%
% dual problem:
%   min   b'y
%   s.t.  A'y >= c
%           y >= 0
%
% dx, dy, dz : step direction
% fx, fy, gx, gy :
% phi, psi, dphi, dpsi :
```

```
% D, E : diagonal matrices
% gamma, beta, delta, mu, theta : parameters
%

EPS = 1.0e-12;
MAX_ITER = 600;

status = 5;

% Allocate memory for arrays.
n = size(c(:), 1);
m = size(b(:), 1);

% Initialization
x = ones(n, 1);
z = ones(n, 1);
y = ones(m, 1);
w = ones(m, 1);

phi = 1.0;
psi = 1.0;

fprintf('HSD method\n');
fprintf('-------------------------------------------------------------------------------\n');
fprintf('%4s   | %22s     | %22s     | %8s |\n', '', 'PRIMAL', 'DUAL', '');
fprintf('%4s   | %14s    %8s | %14s    %8s | %8s |\n', 'iter', 'Obj Value',...
    'Infeas', 'Obj Value', 'Infeas', 'mu');
fprintf('-------------------------------------------------------------------------------\n');

% Iteration.
% beta = 0.80;
% delta = 2 * (1 - beta);

for iter = 1: MAX_ITER,
    % STEP 1: Compute mu.
    mu = (z'*x + w'*y + psi*phi) / (n + m + 1);
    if (mod(iter, 2) ~= 0)
        delta = 0;
    else
        delta = 1.0;
```

```
end
% STEP 1: Compute primal and dual objective function values.
primal_obj = c'*x;
dual_obj = b'*y;
% STEP 2: Check stopping rule.
if (mu < EPS),
    if (phi > EPS), status = 0; break;            % optimal
    elseif (dual_obj < 0.0),  status = 2; break;  % primal infeasible
    elseif (primal_obj > 0.0), status = 4; break; % dual infeasible
    else status = 7; break;                       % numerical problem
    end
end
% STEP 3: Compute infeasibilities.
rho = A*x - b*phi + w;
normr = norm(rho, 2)/phi;
rho_hat = -(1 - delta)*rho + w - delta*mu./y;
sigma = -A'*y + c*phi + z;
norms = norm(sigma, 2)/phi;
sigma_hat = -(1 - delta)*sigma + z - delta*mu./x;
gamma_hat = -(1 - delta)*(dual_obj - primal_obj + psi) + psi - delta*mu/phi;

fprintf('%4d   | %14.7e    %8.1e | %14.7e    %8.1e | %8.1e |\r',...
    iter, primal_obj/phi, normr, dual_obj/phi, norms, mu);

% STEP 4: Compute step directions.
X = diag(x); Y = diag(y); Z = diag(z); W = diag(w);
D = X^(-1)*Z;
E = Y^(-1)*W;
quasiD = [-E, A; A', D];
fyfx = quasiD\[rho_hat; -sigma_hat;];
gygx = quasiD\[-b; -c];

fy = fyfx(1: m, 1);
fx = fyfx(m + 1: end, 1);
gy = gygx(1: m, 1);
gx = gygx(m + 1: end, 1);

dphi = (c'*fx - b'*fy + gamma_hat)/(c'*gx - b'*gy - psi/phi);

dx = fx - gx*dphi;
```

```
    dy = fy - gy*dphi;
    dz = delta*mu./x - z - D*dx;
    dw = delta*mu./y - w - E*dy;
    dpsi = delta*mu/phi - psi - (psi/phi)*dphi;

    % STEP 5: Compute step length.
    if (mod(iter,2) == 0),
        theta = 1.0;
        continue;
    else
        theta = 0.0;
        % Attention!!!!!
        theta = max([theta; -dx./x; -dy./y; -dz./z; -dw./w; -dphi/phi; -dpsi/psi]);
        theta = min(0.95/theta, 1.0);
    end

%     if (theta < 4*beta/(n+m+1)),
%         printf('ratio = %10.3e \n', theta*(n+m+1)/(4*beta));
%         status = 7;
%         break;
%     end
%     if (theta < 1.0), theta = theta * 0.9999; end

    % STEP 6: Update the iteration.
    x = x + theta*dx;
    y = y + theta*dy;
    z = z + theta*dz;
    w = w + theta*dw;
    phi = phi + theta*dphi;
    psi = psi + theta*dpsi;

end
x = x/phi;
z = z/phi;
y = y/phi;
w = w/phi;
fval = primal_obj/phi;
end
% End of solver
```

参考文献

[1] Robert J. Vanderbei. *Linear Programming Foundations and Extensions.* Springer New York Heidelberg Dordrecht London, Princeton, New Jersey, USA, 4nd edition, 2014.