计算复杂性

- 算法性能：
- （**1**）在最坏情况下算法所表现出来的性能；**-----------------------**最坏性能
- （**2**）在各种情况可能出现时，算法所表现出来的期望性能。**----------**平均性能

旅行商(TSP)问题

- 一个商人欲到n个城市推销商品，每两个城市i和j之间的距离为d_{ij}，如何选择一条道路，使得商人每个城市走过一遍后回到起点且所走路径最短。
- 解：设x_{ij}=1若商人行走的路线中包含从城市i到j的路径，否则x_{ij}=0。

$$
\begin{cases}
\min \sum\limits_{i \neq j} d_{ij} x_{ij} \\
s.t. \quad \sum\limits_{j=1}^{n} x_{ij} = 1, \ i = 1,2,\cdots,n \\
\qquad \sum_{i=1}^{n} x_{ij} = 1, \ j = 1,2,\cdots,n \\
\qquad \sum\limits_{i,j \in S} x_{ij} \leq |S| - 1, \ 2 \leq |S| \leq n-1 \ S \subset \{1,\cdots,n\} \\
\qquad x_{ij} = 1 \ or \ 0
\end{cases}
$$

- 可行解：用n个城市的一个排列表示商人按这个排列次序推销并返回起点。
- 使用枚举法求解，需要$(n-1)!$次枚举。
- 以计算机1秒可以完成24个城市所有路径枚举为单位。
- 城市数 24 25 26 27 30
- 计算时间 1秒 24秒 10分 4.3小时 10.8年

- **1. 问题与实例**
- 问题**(problem):**需要回答的一种提问，通常包含一些参数和取值未定的自由变量，可以从两个方面加以描述：
- （**1**）对所有参数的一般描述；
- （**2**）对回答（也称为解）所需要满足的特性的描述。
- 实例**(instance)**：当对一个问题中的参数赋予特定的数值时，如何寻找相应的回答（解），这种提问称为该问题的一个实例。
- 问题是对许多具体事例构成集合的一种抽象表述，而实例就是相应问题的一种具体表现形式。

- 例：线性规划问题与实例
- 一个线性规划问题的实例是指矩阵和向量组(*A*, ***b***, ***c***)的某一特定取值，这些参数按照如下的结构关联在一起，描述了问题（解）所需要满足的特性。

$$
\begin{aligned}
\min \quad & cx \\
s.t. \quad & Ax = b \\
& x \geq 0
\end{aligned}
$$

线性规划问题是对具有上述结构的所有实例的一种抽象描述。

算法

- 算法：是一组含义明确的简单指令。
- 一个问题是算法可解的**(solvable)**：存在一个求解该问题的算法，只要让算法运行足够长的时间，并且保证满足算法在运行过程中所需要的存储空间，它就能求解该问题的任何一个实例。
- 停机问题：不可能构造出一个程序来确定任意给出的程序是否会陷入无限循环。

算法复杂性

- 算法复杂性(**algorithm complexity**): 描述算法的存储要求和运行时间要求，分为算法的空间复杂性和算法的时间复杂性。
- **-----**利用算法需要的初等运算次数来表示算法的时间复杂性。

多项式时间算法与指数时间算法

- 输入规模**(input size)**：表示一个实例所需要的字符串长度。
- 一般的，使用 $\lceil 1+\log_2 r \rceil$ 位二进制就可以表示任意整数***r***。
- 线性规划的输入规模为：

$$L = \lceil 1+\log_2 m \rceil + \lceil 1+\log_2 n \rceil + \sum_{j=1}^{n} \{\lceil 1+\log_2 |c_j| \rceil\}$$

$$+ \sum_{i=1}^{m}\sum_{j=1}^{n} \{\lceil 1+\log_2 |a_{ij}| \rceil\} + \sum_{i=1}^{m} \{\lceil 1+\log_2 |b_j| \rceil\}$$

$$\leq mn + m + n + 2 + \log_2 |P| \, (P\text{为}A,b,c\text{中所有非零数的乘积})$$

- 对应TSP，枚举算法的基本计算总次数为 $[(n\text{-}1)!]n=n!$
- 实例的二进制输入长度总量不超过
- $$L=n(n-1)+\log_2|P|$$
- 其中P为所有非零数d_{ij}的乘积。
- 假设 $S=\left\{d_{ij}\,|\,1\le i,j\le n, i\ne j\right\}$ 中每个数据都有上界K，则有

$$L\le n(n-1)\left(1+\log_2 K\right)$$

- 一个求解实例I的算法的基本计算总次数$C(I)$同实例I的计算机二进制输入长度$d(I)$的关系常用符号$C(I)=f(d(I))=O(g(d(I)))$表示，它的含义：求解实例I的算法的基本计算总次数$C(I)$是实例输入长度$d(I)$的一个函数，该函数被另一个函数$g(x)$控制，即存在一个函数$g(x)$和一个常数a，使得

$$C(I) \le ag(d(I))$$

多项式时间算法与指数时间算法

- 定义：假设问题和解决该问题的一个算法已经给定，若给定该问题的一个实例***I***，存在多项式函数***g*(*x*)**，使得

$$C(I) \le ag(d(I))$$

- 成立，则称该算法对实例***I***是多项式时间算法；若存在***g*(*x*)**为多项式函数且对该问题任意一个实例***I***，都有上式成立，则称该算法为解决该问题的多项式时间算法。
- 当***g*(*x*)**为指数函数时，称相应的算法为指数时间算法。

复杂度与问题规模

算法复杂度	当前	计算机提速100倍	计算机提速10000倍
$O(n)$	n	100n	10000n
$O(n\log(n))$	n	～100n	～10000n
$O(n^2)$	n	10n	100n
$O(n^5)$	n	2.5n	6.3n
$O(2^n)$	n	n＋6.64	n＋13.29
$O(3^n)$	n	n＋4.19	n＋8.38

- 多项式时间算法的优点：
- （**1**）随着问题输入规模的增加，算法的计算量（即算法复杂性）呈多项式增长；
- （**2**）一个多项式时间算法利用另一个多项式时间算法作为其“子程序”，构造一个新的复合型算法，则新算法仍是多项式时间算法。

单纯形算法的复杂性

$$\max \quad x_0 = \sum_{i=1}^{n} 10^{n-i} x_i$$

$$s.t. \quad x_i + 2\sum_{j<i} 10^{i-j} x_j \le 10^{2i-2}, i = 2, \cdots, n$$

$$x_i \ge 0, \quad i = 1, 2, \cdots, n$$

二进制输入大小$d(I) \le 6n^2 + 2n + 3n + \log_2 |P|$,

其中P为所有非零系数的乘积。

单纯形算法需要2^n-1次迭代。

$$\max \quad \sum_{i=1}^{n} 2^{n-i} x_i$$

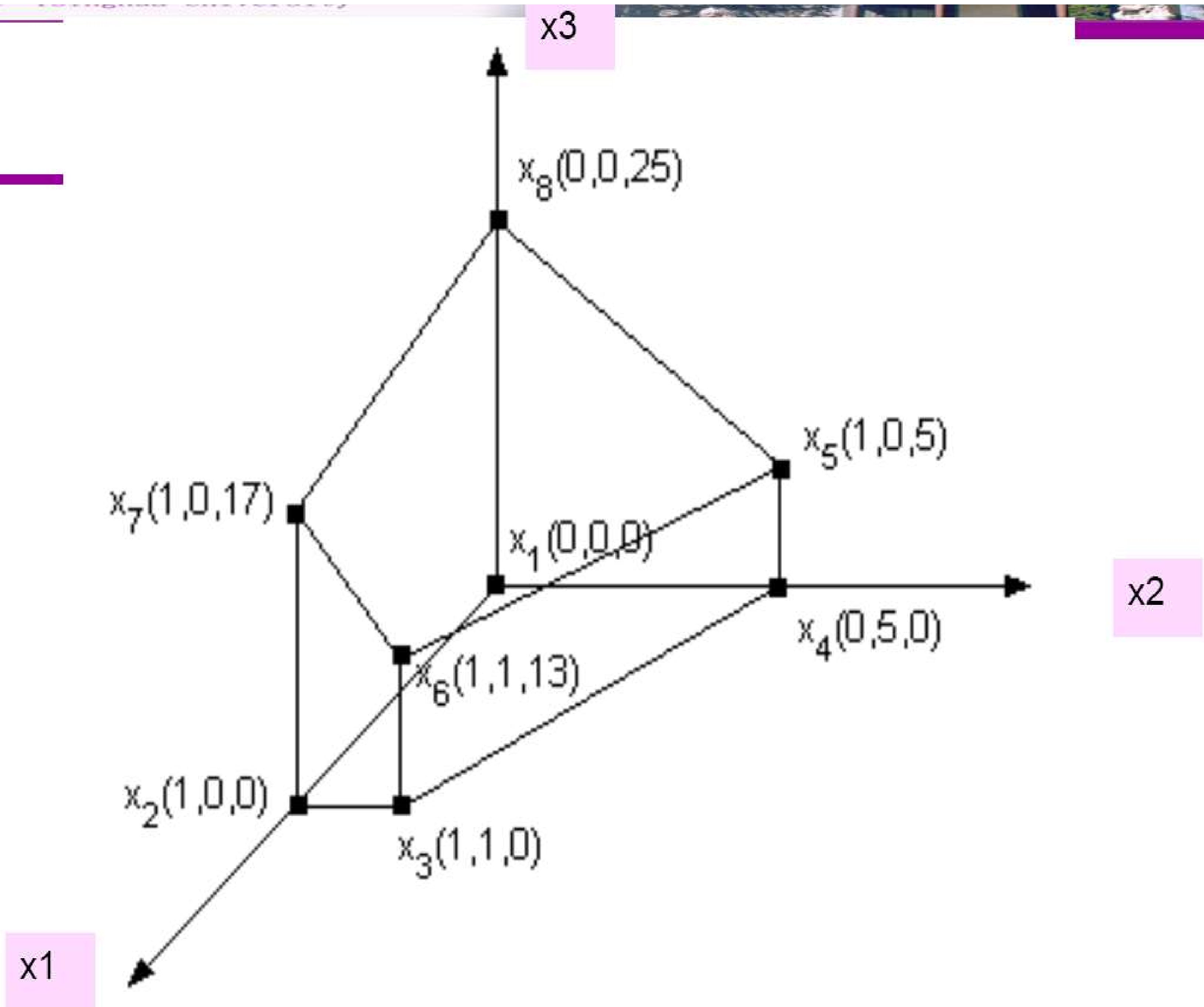
$$s.t. \quad x_i + 2\sum_{j=1}^{i-1} 2^{i-j} x_j \le a^{i-1}, i = 2, \cdots, n$$

$$x_i \ge 0, \quad i = 1, 2, \cdots, n$$

定理：当a>2时，用单纯形算法求解上述问题时需要 2^n-1次迭代。

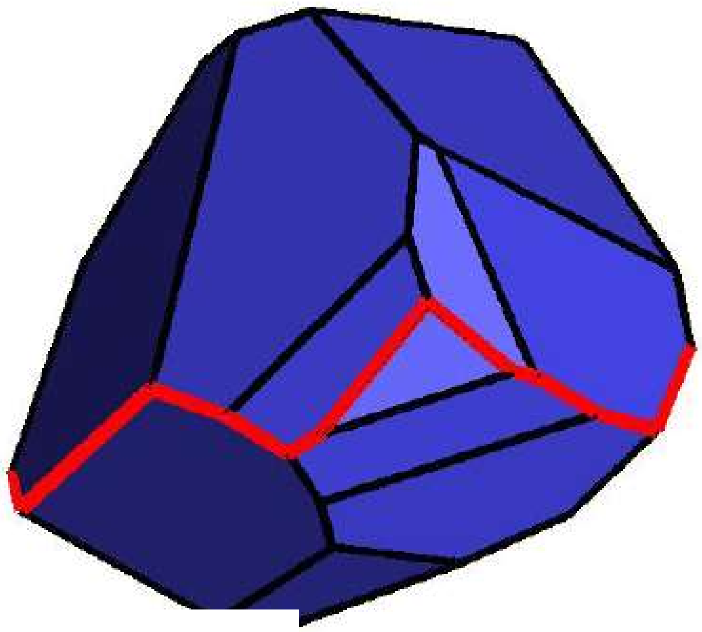
Example: n=3, a=5

$$\begin{aligned} \max \quad & 4x_1 + 2x_2 + x_3 \\ s.t. \quad & x_1 \leq 1 \\ & 4x_1 + x_2 \leq 5 \\ & 8x_1 + 4x_2 + x_3 \leq 25 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$


x3
$x_8(0,0,25)$
$x_5(1,0,5)$
$x_7(1,0,17)$
$x_1(0,0,0)$
x2
$x_4(0,5,0)$
$x_6(1,1,13)$
$x_2(1,0,0)$
$x_3(1,1,0)$
x1

Simplex Method for Linear Programming

$$\begin{aligned} \min \quad & cx \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$



- Worst-Case: exponential
- Average-Case: polynomial
- Widely used in practice

History of Linear Programming

- Simplex Method (Dantzig, '47)
  Exponential Worst-Case (Klee-Minty '72)
  Avg-Case Analysis (Borgwardt '77, Smale '82, Haimovich, Adler, Megiddo, Shamir, Karp, Todd)

- Ellipsoid Method (Khaciyan, '79) $O(n^4 L)$
- Interior-Point Method (Karmarkar, '84) $(n^{3.5} L)$

椭球法

第一个可以在多项式时间内解决一般线性规划问题的解法。

$$(P)\quad \begin{cases} \min cx \\ s.t. \quad Ax \geq b \\ \qquad\quad x \geq 0 \end{cases} \qquad (D)\quad \begin{cases} \max b^T w \\ s.t. \quad A^T w \leq c \\ \qquad\quad w \geq 0 \end{cases}$$

根据**(*P*)** 与**(*D*)** 的对偶关系**,** 我们可将两者的最优解以一组最优性条件联结起来**:**

$$\begin{cases} Ax \geq b, \quad x \geq 0 \\ A^T w \leq c, \quad w \geq 0 \\ cx - wb = 0 \end{cases} \qquad (*)$$

定理：存在求解LP问题的多项式时间算法的充要条件是存在求解线性不等式组$Ax \le b$的多项式时间算法。

证明：与线性不等式组$Ax \le b$相关的LP问题为

$$(*) \quad \begin{cases} \min cx \\ s.t. \quad \overline{A}\overline{x} \ge \overline{b} \\ \qquad\quad \overline{x} \ge 0 \end{cases}$$

其中$\overline{x} = \begin{pmatrix} x^{+} \\ x^{-} \end{pmatrix}, \overline{A} = \begin{pmatrix} -A & A \end{pmatrix}, \overline{b} = -b, x^{+}, x^{-} \ge 0, c$任取$\left(\text{如}c = 0\right)$

若有多项式时间的LP算法，能够判断问题$(*)$不可行，则不等式组$Ax \le b$无解；或者得到其最优解或判定问题无界，则得到不等式组$Ax \le b$的一个解，显然就以多项式时间解决了问题$Ax \le b$。

定理：存在求解LP问题的多项式时间算法的充要条件是存在求解线性不等式组$Ax \le b$的多项式时间算法。

$(*)$的对偶问题为
$$\begin{cases} \max\ wb \\ s.t.\quad w\bar{A} \le c \\ \qquad\quad w \ge 0 \end{cases}$$

所以求解LP问题可归结为求解关于变量$(\bar{x}, w)$的线性不等式组：$\bar{A}\bar{x} \ge \bar{b}, w\bar{A} \le c, c\bar{x} - w\bar{b} \le 0, \bar{x}, w \ge 0$

设有多项式时间方法求解线性不等式组。若该联立不等式组有解(x^*, w^*)，则x^*是LP问题的最优解，w^*是其对偶问题的最优解；若该联立不等式组无解，考虑不等式组

$$\bar{A}\bar{x} \ge \bar{b}, \bar{x} \ge 0$$

若它有解，则LP问题无界；否则LP问题不可行。

只要能有效的解决最优性条件的线性不等式，就能够同时的解决一个线性规划问题(***P***) 以及它的对偶问题(***D***)。椭球法正是一种专门解决线性不等式的方法。

介绍如何以椭球法來解一组线性不等式$\boldsymbol{Mu \le v}$

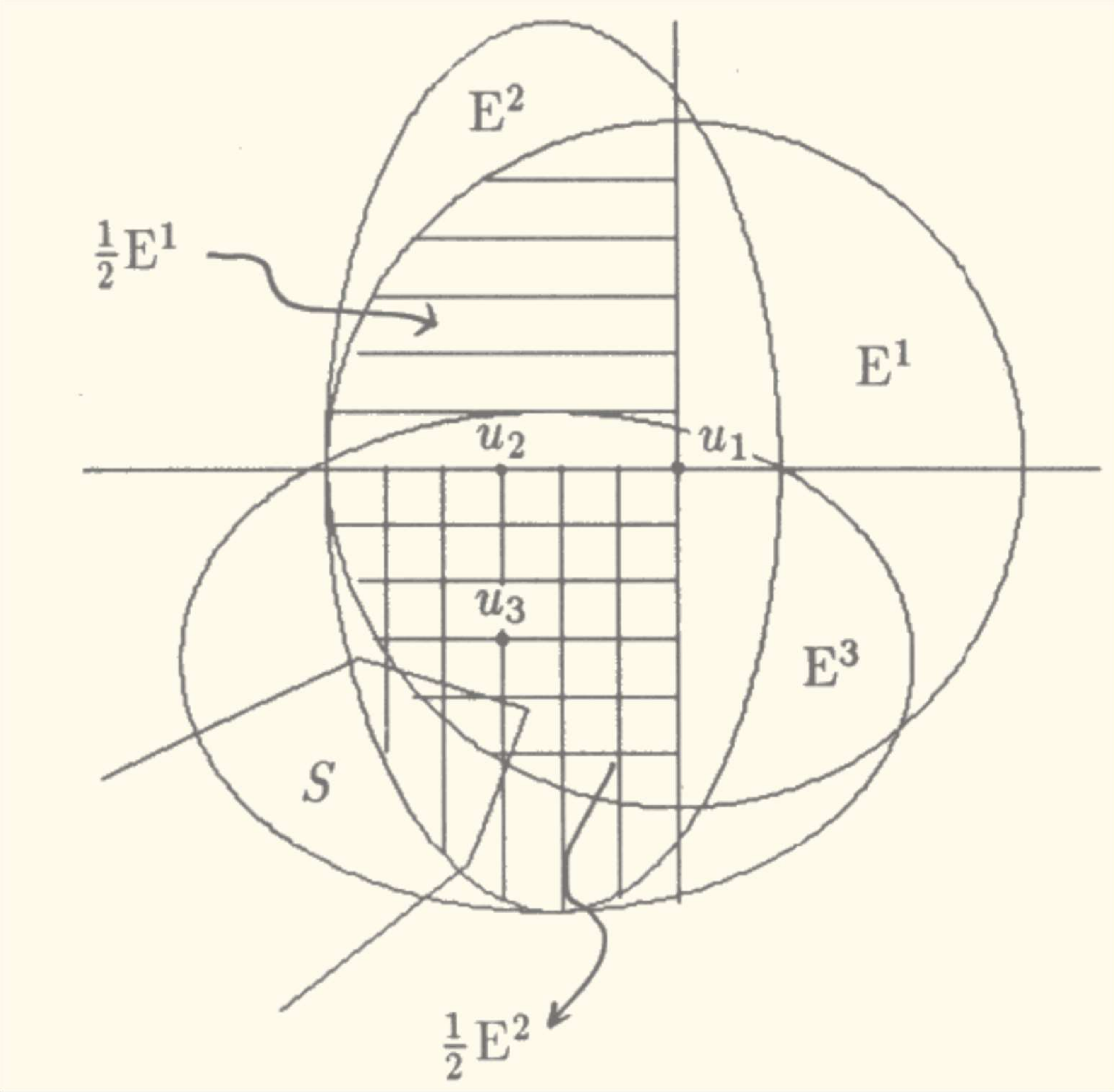
$\boldsymbol{Mu \le v}$的解集合是一个凸集： $S = \{u \mid Mu \le v\}$

假设$\boldsymbol{S \neq \varnothing}$，以原点$\boldsymbol{u^1}$为圆心，足够大的半径做一圆$\boldsymbol{E^1}$，使得$\boldsymbol{S \cap E^1 \neq \varnothing}$。

若 $Mu^1 \le v$， 则 u^1为所求的解。

否则，因为$S\cap E^1$是一个凸集，所以经过圆心，可以切掉不含$S\cap E^1$的半个圆，而只剩下包含$S\cap E^1$的半个圆（以$1/2E^1$表示）。对$1/2E^1$而言，可以做出一个最小的椭圆E^2，使得$1/2E^1 \subseteq E^2$，椭圆的圆心记u^2。

$S\cap E^1 \subseteq E^2$，若$Mu^2 \leq v$成立，则$u^2 \in S\cap E^1$必为其解。否则经过u^2又可切去半个E^2，而使$S\cap E^1$包含在另一半椭圆$1/2E^2$之中。



E^2
$\frac{1}{2}E^1$
E^1
u_2
u_1
u_3
E^3
S
$\frac{1}{2}E^2$

在p维空间中，每次做出的椭球体积都会逐渐缩小。以$V(E^k)$及$V(E^{k+1})$来表示前后两个椭球的体积，那么可以证明$V(E^{k+1}) < e^{-1/2\,(p+1)} V(E^k)$。所以

$$V(E^{k+1}) < e^{-k/2(p+1)} V(E^1)。$$

表明在多项式时间内，新的椭球体积便可缩减至零，否则原题无解。

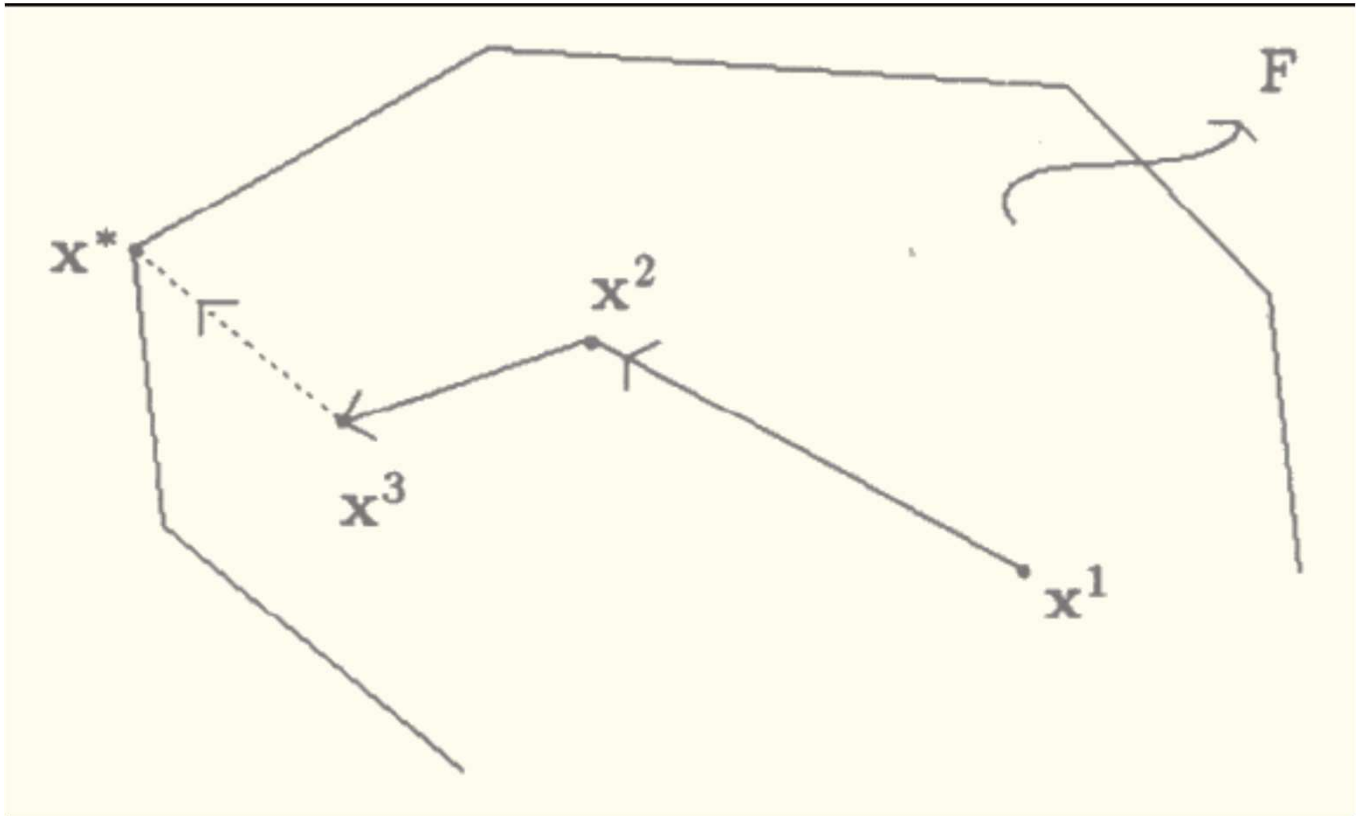
步骤:

1: 考虑最优化条件的线性不等式**(*)**。在$(x,w)\in R^{n+m}$的空间中**,** 以原点$u^1=(x^1,w^1)$为心**,** 足够大的正数2^{2L}为半径做一圆球E^1。置$k=1$。

2: 检验现有球心$u^k=(x^k,w^k)$是否满足该最优化不等式。若满足**,**则x^k是(P) 的最优解**,**w^k是(D)的最优解。若不满足**,**则依前述方法得到一个新的椭球E^{k+1}及其球心u^{k+1}。

3: 若是$V(E^{k+1})=0$,u^{k+1}仍不满足最优化不等式**,** 则(P)或(D)至少有一无最优解。否则置$k:=k+1$**,** 返回**2**。

内点法



$$
(P)\quad \begin{cases} \min c^T x \\ s.t. \quad Ax = b \\ \qquad\quad x \geq 0 \end{cases}
$$

称$x \in R^n$为一个可行内点**(interior feasible)**

如果$Ax = b, x > 0$。

可行内点解集合：$F^0 = \{x \in R^n | Ax = b, x > 0\}$。

假设F^0非空。

内点法可粗略的分为三个步骤：

步骤一：找一个可行内点$\boldsymbol{x}^1 \in \boldsymbol{F}^0$。置$\boldsymbol{k}=1$。

步骤二：决定现有解$\boldsymbol{x}^k$是否为(P) 的最优解。若是，则输出$\boldsymbol{x}^* = \boldsymbol{x}^k$。否则就寻找一个好的移动方向$d_x^k$，以及适当的步长$\alpha_k > \boldsymbol{0}$。

步骤三：由x^k移动到新的内点

$$x^{k+1} = x^k + \alpha_k d_x^k \in F^0.$$

置$k := k+1$，返回步骤二。

Primal Affine Scaling内点法

假想(P) 的可行解集合位于第一象限的一个球，而现行解x^k落在球心上，此球的半径是$r > 0$。

$$(P_1)\quad \begin{cases} \min c^T x \\ s.t.\quad \sum_{j=1}^{n}\left(x_j - x_j^k\right)^2 \le r^2 \end{cases}$$

最优解为：

$$x^* = x^{k+1} = x^k + r\frac{-c}{\left\|-c\right\|}$$

将**(P_1)** 变得稍微复杂一些，将球改为第一象限来考虑下列问题：

$$(P_2)\quad \begin{cases} \min c^T x \\ s.t. \quad x \geq 0 \end{cases}$$

假设$x^k > 0$，定义$n \times n$矩阵

$$D_k = \begin{bmatrix} x_1^k & & & 0 \\ & x_2^k & & \\ & & \ddots & \\ 0 & & & x_n^k \end{bmatrix} \quad D_k^{-1} = \begin{bmatrix} \frac{1}{x_1^k} & & & 0 \\ & \frac{1}{x_2^k} & & \\ & & \ddots & \\ 0 & & & \frac{1}{x_n^k} \end{bmatrix}$$

定义映射：

$$T_k: R_+^n \to R_+^n$$

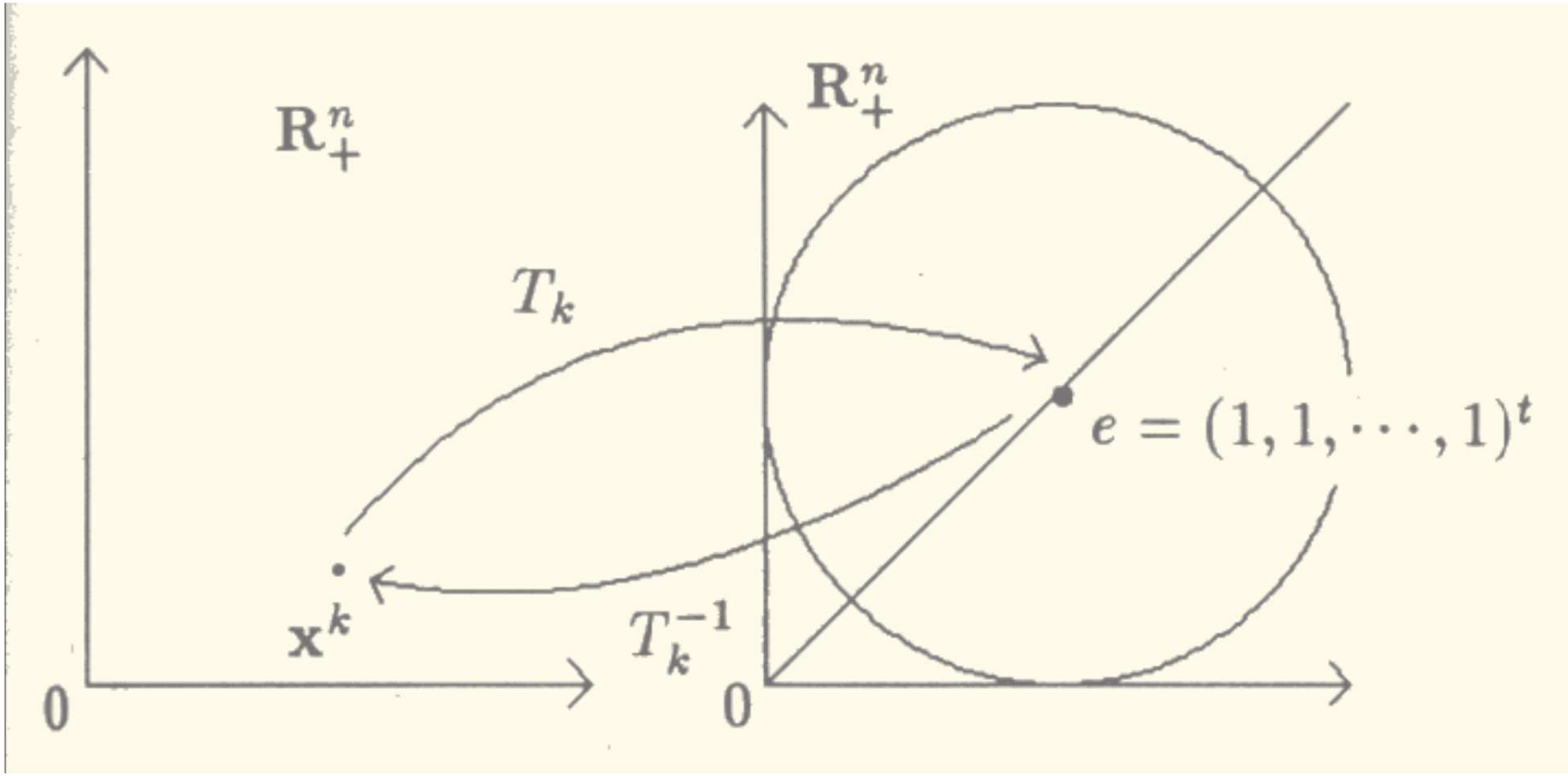
$$x \to y = T_k(x) \overset{\Delta}{=} D_k^{-1} x \qquad 则$$

$$T_k^{-1}: R_+^n \to R_+^n$$

$$y \to x = T_k^{-1}(y) = D_k y$$

$$y^k = T_k(x^k) = D_k^{-1} x^k = e$$

$$T_k^{-1}(e) = D_k e = x^k$$



$\mathbf{R}_+^n$
T_k
$\mathbf{x}^k$
T_k^{-1}
0
$\mathbf{R}_+^n$
$e = (1, 1, \cdots, 1)^t$
0

在上述转化之下，$(\mathbf{P_2})$ 的近似问题在**y**空间中就可写成：

$$(P_2')\quad \begin{cases} \min\ c^T D_k y \\ s.t.\quad \sum_{j=1}^{n}\left(y_j-1\right)^2 \le 1 \end{cases}$$

应用$(\mathbf{P_1})$ 的解答技巧，在y空间中的近似最优解为：

$$y^* = y^{k+1} = y^k + \frac{-D_k c}{\|D_k c\|} = e - \frac{D_k c}{\|D_k c\|}$$

在x空间中$(\mathbf{P_2})$的近似最优解为：

$$x^* = x^{k+1} = D_k y^{k+1} = D_k e - \frac{D_k^2 c}{\|D_k c\|} = x^k - \frac{D_k^2 c}{\|D_k c\|}$$

$$(P)\quad \begin{cases} \min c^T x \\ s.t. \quad Ax = b \\ \qquad\quad x \geq 0 \end{cases}$$

在y空間中**(P)** 变成下列的近似问题**:**

$$(P')\quad \begin{cases} \min \;\; c^T D_k y \\ s.t. \quad AD_k y = b \\ \qquad \sum_{j=1}^{n} \left(y_j - 1\right)^2 \leq 1 \end{cases}$$

与(P'_2) 相比较，(P') 多了一些等式约束条件 $AD_k y = b$。为了保持这些等式的继续成立，原来在(P'_2) 中的移动方向$-D_k c$便需要投影在(AD_k)这个矩阵的零空间(null space) 之中，即经过投影后的方向应是$P_k(-D_k c)$，其中P_k是一个投影矩阵，定义为：

$$P_k = \left[I - D_k A^T \left(A D_k^2 A^T \right)^{-1} A D_k \right]$$

所以**(P')** 的近似最优解是 $y^* = y^{k+1} = e + \dfrac{P_k\left(-D_k c\right)}{\left\| P_k\left(-D_k c\right) \right\|}$

(P) 的近似最优解是 $x^* = x^{k+1} = D_k y^{k+1} = x^k - \dfrac{D_k P_k D_k c}{\left\| P_k D_k c \right\|}$

定理：若存在$j, j=1,2,\cdots,n$使得$x_j^{k+1}=0$，则x^{k+1}是(P)的最优解。

Primal Affine Scaling内点法的重要步骤:

1. 找一个可行内点$x^1 \in F^0$，置$k:=1$.

2. 计算移动方向$d_x^k=-D_k P_k D_k c$，步长$\alpha_k=\dfrac{1}{\|P_k D_k c\|}$，

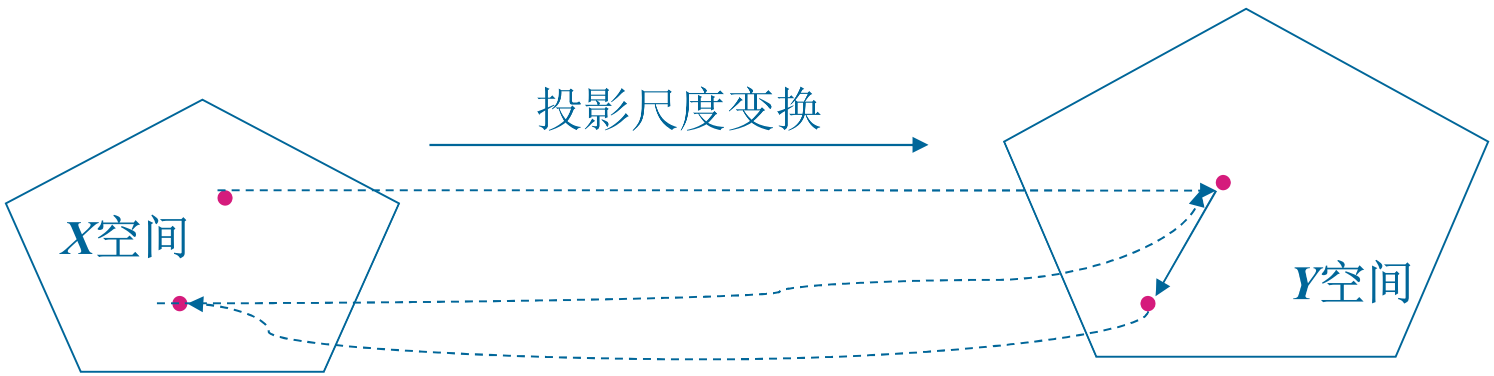
新内点$x^{k+1}=x^k+\alpha_k d_x^k$.

3. 检验是否有$x_j^{k+1}=0$，$j=1,2,\cdots,n$，若成立，则输出$x^*=x^{k+1}$；否则置$k:=k+1$，返回2.

Karmarkar投影尺度算法

- 两个基本事实：
- （**1**）如果一个内点居于多面体的中心，那么目标函数的负梯度方向是比较好的可行方向；
- （**2**）在不改变问题基本特性的条件下，存在一个适当的变换，能够将可行域中给定的内点置于变换后的可行域的中心。

- 基本思想：
- （**1**）选定一个内点解作为迭代过程的初始点，利用可行域的投影尺度变换，将当前的内点解置于变换后的可行域的中心；
- （**2**）在变换后的可行域中沿着目标函数最速下降方向的正交投影移动，获得新的可行内点，并通过投影尺度逆变换将新的可行内点映射到原来的可行域，作为新的迭代点。
- （**3**）重复这一过程，直至求出满足一定精度的近优解。



Karmarkar标准形

$$(*)\quad \begin{cases} \min & c^T x \\ s.t. & Ax = 0 \\ & x_1 + x_2 + \cdots + x_n = 1 \\ & x \geq 0 \end{cases}$$

其中A是$m \times n$满秩矩阵，$c, x \in R^n$.

基本假设: (1) 问题(*)是可行的，单纯形$S = \left\{ x \mid \sum_{j=1}^{n} x_j = 1,\ x \geq 0 \right\}$

的中心$a^{(0)} = \dfrac{1}{n} e$是可行点，其中$e = (1, \cdots, 1)^T$.

(2) 对每个可行点x，有$c^T x \geq 0$。

(3) 终止参数q给定，目的是求一个可行点x，使得

$$\frac{c^T x}{c^T a^{(0)}} \leq 2^{-q}.$$

单纯形

定义集合$S=\left\{x \mid e^T x=1, x \geq 0\right\}$为$n-1$维单纯形，

其顶点是$e_j=(0,\cdots,0,1,0,\cdots,0)^T, j=1,\cdots,n.$

S的中心是$a^{(0)}=\frac{1}{n}e.$

单纯形S的外接球半径

$$R=\sqrt{\left(1-\frac{1}{n}\right)^2+(n-1)\left(0-\frac{1}{n}\right)^2}=\sqrt{\frac{n-1}{n}}.$$

单纯形S的内接球半径

$$r=\sqrt{\left(0-\frac{1}{n}\right)^2+(n-1)\left(\frac{1}{n}-\frac{1}{n-1}\right)^2}=\frac{1}{\sqrt{n(n-1)}}.$$

外接球与内接球半径之比$R/r=n\text{-}1.$

单纯形S的投影尺度变换

定义 $T(x)=\dfrac{D^{-1}x}{e^{T}D^{-1}x}$,

其中$D=diag(a_1,\cdots,a_n),\ a_i>0,\ i=1,\cdots,n.$

记$T(x)=y$, 则

$$y_i=\frac{x_i/a_i}{\sum_j\left(x_j/a_j\right)},\ i=1,\cdots,n.$$

$$\sum_{i=1}^{n}y_i=1$$

变换$\boldsymbol{T}$的性质

1. T是单纯形S上的一一对应，其逆变换为

$$x = T^{-1}(y) = \frac{Dy}{e^T Dy} \quad (*) \quad \left(x_i = \frac{a_i y_i}{\sum_j a_j y_j} \right).$$

S的每个顶点$e_j = (0,\cdots,0,1,0,\cdots,0)^T$的像还是这个顶点。

2. 约束$Ax = 0$在变换$(*)$下等价于$ADy = 0$.

3. 点$a = (a_1,\cdots,a_n)^T$的像是单纯形S的中心$\frac{1}{n}e = (a^{(0)})$。

4. T把由$x_j = 0$给出的S的每个面映射成对应的面$y_j = 0$。

5. 若$a \in \{x \mid Ax = 0\}$，则S的中心$a^{(0)} \in \{y \mid ADy = 0\}$.

势函数

对每个线性函数$l(x)$，定义与它相联系的**势函数**为：

$$f(x)=\sum_{j=1}^{n}\ln\frac{l(x)}{x_j}+k \quad (\text{其中}k\text{为常数})$$

性质：

1．射影变换T把势函数变换成具有相同形式的函数.

2．目标函数值所期望的下降量可通过势函数值的充分小来达到.

3．优化函数$f(x)$可用优化线性函数来近似.

求解方法

$$(*)\quad \begin{cases} \min \quad c^T x \\ s.t. \quad Ax = 0 \\ \qquad\quad x_1 + x_2 + \cdots + x_n = 1 \\ \qquad\quad x \geq 0 \end{cases}$$

其中A是$m \times n$满秩矩阵，$c, x \in R^n$.

$$(*)\quad \begin{cases} \min \quad c^T x \\ s.t. \quad x \in S \bigcap \Omega \end{cases}$$

其中$\Omega = \{x \mid Ax = 0\}, S = \left\{ x \mid \sum_{j=1}^{n} x_j = 1, x \geq 0 \right\}$

运用变换$x=T^{-1}(y)=\dfrac{Dy}{e^T Dy}$把$\Omega$变为$\Omega'=\{y \mid ADy=0\}$，$S$变为$S$，即把可行域$S\cap\Omega$变为$S\cap\Omega'$，同时把单纯形上的可行点$a=(a_1,\cdots a_n)^T>0$变成单纯形$S$的中心$a^{(0)}=\dfrac{1}{n}e$，则$a^{(0)}\in S\cap\Omega'$。

$$c^T x \quad\Rightarrow\quad \frac{c^T Dy}{e^T Dy}$$

$$f(x)=\sum_{j=1}^{n}\ln\frac{c^T x}{x_j} \quad\Rightarrow\quad \tilde{f}(y)=\sum_{j=1}^{n}\ln\frac{c'^T y}{y_j}-\sum_{j=1}^{n}\ln a_j$$

$$\begin{cases} \min & c^T x \\ s.t. & x \in S \bigcap \Omega \end{cases} \Rightarrow \begin{cases} \min & \tilde{f}(y) \\ s.t. & y \in S \bigcap \Omega' \end{cases}$$

用包含在S内、以$a^{(0)}$为球心，αr为半径的球$B(a^{(0)}, \alpha r)$取代S，其中$r = \dfrac{1}{\sqrt{n(n-1)}}$，$\alpha \in (0,1)$。记

$$B'(a^{(0)}, \alpha r) = B(a^{(0)}, \alpha r) \bigcap \Omega'$$

则对$\forall y \in B'(a^{(0)}, \alpha r)$，有

$$ADy = 0 \qquad\qquad 且$$

$$\sum_{j=1}^{n} y_j = 1，即 e^T y = 1。$$

令 $B = \begin{bmatrix} AD \\ e^T \end{bmatrix}$ $e_{m+1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$\begin{aligned} ADy &= 0 \\ e^T y &= 1 \end{aligned} \quad \Longleftrightarrow \quad By = e_{m+1}$$

令 $\Omega'' = \{ y \mid By = e_{m+1} \}$

$$\begin{cases} \min & \tilde{f}(y) \\ s.t. & y \in S \bigcap \Omega' \end{cases} \Rightarrow \begin{cases} \min & \tilde{f}(y) \\ s.t. & y \in B(a^{(0)}, \alpha r) \bigcap \Omega'' \end{cases}$$

$$\begin{cases} \min & \tilde{f}(y) \\ s.t. & y \in B(a^{(0)}, \alpha r) \bigcap \Omega'' \end{cases} \Rightarrow \begin{cases} \min & c'^T y \\ s.t. & y \in B(a^{(0)}, \alpha r) \bigcap \Omega'' \end{cases}$$

其中$c' = Dc$。

记 $c_p=\left(I-B^T\left(BB^T\right)^{-1}B\right)Dc$

$c'^T y$在Ω''上下降最快的方向是

$$d^{(0)}=-\frac{c_p}{\left\|c_p\right\|}$$

$c'^T y$在$B(a^{(0)},\alpha r)$上的最小点是

$$b'=a^{(0)}-\frac{\alpha r}{\left\|c_p\right\|}c_p$$

Karmarkar投影尺度算法

1. 置$k=0,\ x^{(0)}=\dfrac{1}{n}e,\ r=\sqrt{1/(n-1)n}$; 参数$\alpha\in(0,1)$，$L$是充分大的正整数.

2. 若$c^Tx^{(k)}\le 2^{-L}c^Tx^{(0)}$，则停止计算，得到近优解$x^{(k)}$; 否则转3。

3. 令$X^{(k)}=diag\left(x^{(k)}\right)$，$B_k=\begin{pmatrix} AX^{(k)} \\ e^T \end{pmatrix}$,

$$d_y^k=-\left(I-B_k^T\left(B_kB_k^T\right)^{-1}B_k\right)X^{(k)}c,$$

$$y^{k+1}=\frac{1}{n}e+\alpha r\frac{d_y^k}{\left\|d_y^k\right\|},\quad x^{(k+1)}=\frac{X^{(k)}y^{k+1}}{e^TX^{(k)}y^{k+1}}.$$

4. 计算势函数值$f\left(x^{(k)}\right)$和$f\left(x^{(k+1)}\right)$，若$f\left(x^{(k)}\right)-f\left(x^{(k+1)}\right)<\delta$，则停止计算，得出*Karmarkar*标准问题的极小值大于0的结论，这种情形对应原来的线性规划问题不可行或无下界；若$f\left(x^{(k)}\right)-f\left(x^{(k+1)}\right)\geq 0$，则转5。

5. 置$k:=k+1$，返回2。

路径跟踪法

$$(L)\quad \begin{cases} \min\ c^T x \\ s.t. \quad Ax = b \\ \qquad\quad x \geq 0 \end{cases} \qquad (D)\quad \begin{cases} \max\ b^T y \\ s.t. \quad A^T y + w = c \\ \qquad\quad w \geq 0 \end{cases}$$

$$c_{n\times 1},\ x_{n\times 1},\ b_{m\times 1},\ y_{m\times 1},\ A_{m\times n},\ r(A) = m$$

$$S_L = \{x \mid Ax = b,\ x \geq 0\},\quad S_D = \left\{ \begin{pmatrix} y \\ w \end{pmatrix} \mid A^T y + w = c,\ w \geq 0 \right\}$$

$$S_L^+ = \{x \mid Ax = b,\ x > 0\},\quad S_D^+ = \left\{ \begin{pmatrix} y \\ w \end{pmatrix} \mid A^T y + w = c,\ w > 0 \right\}$$

x, y, w为最优解的充要条件是：

$$\begin{cases} Ax = b, \quad x \geq 0 \\ A^T y + w = c, \quad w \geq 0 \\ XWe = 0 \end{cases}$$

Karush-Kuhn-Tucker条件

其中$X = diag(x_1, x_2, \cdots, x_n)$, x_i为x的第i个分量；

$W = diag(w_1, w_2, \cdots, w_n)$, w_i为w的第i个分量

$$\begin{cases} Ax = b, \quad x \geq 0 \\ A^T y + w = c, \quad w \geq 0 \\ XWe = \mu e \end{cases}$$

松弛KKT条件

定理1　设(L)可行域有界且内部S_L^+非空，则对每个正数μ，松弛KKT条件存在唯一内点解.

定义原始-对偶可行集

$$S=\{(x,y,w)\mid Ax=b,\ A^T y+w=c,\ (x,w)\ge 0\}$$

$$S^+=\{(x,y,w)\mid Ax=b,\ A^T y+w=c,\ (x,w)>0\}$$

$$\{(x(\mu),\ y(\mu),\ w(\mu))\mid \mu>0\}$$

原始-对偶中心路径

定理2 在中心路径上，当μ减少时，原问题的目标值单调减少且趋于最优值，对偶问题的目标值单调增加且趋于最优值，对于中心路径参数μ，对偶间隙$c^T x(\mu)-b^T y(\mu)=n\mu.$

移动方向的计算

设任取一点(x, y, w)，其中$x > 0$，$w > 0$，求一个方向$(\Delta x, \Delta y, \Delta w)$，使迭代产生的点$(x+\Delta x, y+\Delta y, w+\Delta w)$位于原始－对偶中心路径上。

$$A(x+\Delta x)=b$$

$$A^T(y+\Delta y)+(w+\Delta w)=c$$

$$(X+\Delta X)(W+\Delta W)e=\mu e$$

$$A\Delta x=b-Ax$$

$$A^T\Delta y+\Delta w=c-A^Ty-w$$

$$W\Delta x+X\Delta w+\Delta X\Delta We=\mu e-XWe$$

记$b-Ax=\rho, c-A^Ty-w=\sigma$,忽略二次项$\Delta X\Delta We$,

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ W & 0 & X \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta w \end{bmatrix}=\begin{bmatrix} \rho \\ \sigma \\ \mu e-XWe \end{bmatrix}$$

步长的确定

λ的取值应满足：　$x+\lambda\Delta x>0 \quad w+\lambda\Delta w>0$

$$x_j+\lambda\Delta x_j>0,\quad j=1,2,\cdots,n$$

$$w_j+\lambda\Delta w_j>0,\quad j=1,2,\cdots,n$$

$\because x_j>0, w_j>0, \lambda>0$

$$\therefore\quad \frac{1}{\lambda}>-\frac{\Delta x_j}{x_j},\quad \frac{1}{\lambda}>-\frac{\Delta w_j}{w_j}, j=1,2,\cdots,n$$

取　$\lambda=\min\left\{p\left[\max\limits_{i,j}\left(-\frac{\Delta x_j}{x_j},-\frac{\Delta w_j}{w_j}\right)\right]^{-1},1\right\}$

计算步骤：

1. 给定初点$\left(x^{(1)}, y^{(1)}, w^{(1)}\right)$, 其中$x^{(1)} > 0,\ w^{(1)} > 0$，取小于1且接近1的数$p$，精度要求$\varepsilon > 0$，正数$M < \infty$，置$k := 1.$

2. 计算$\rho = b - Ax^{(k)}, \sigma = c - A^T y^{(k)} - w^{(k)}, \gamma = x^{(k)^T} w^{(k)},$

$\mu = \delta \dfrac{\gamma}{n}$，其中$\delta$是小于1的正数，一般取$\delta = 0.1.$

3. 若$\|\rho\|_1 < \varepsilon, \|\sigma\|_1 < \varepsilon$，$\gamma < \varepsilon$同时成立，则停止计算，得到最优解$\left(x^{(k)}, y^{(k)}, w^{(k)}\right)$；若$\left\|x^{(k)}\right\|_\infty > M$或$\left\|y^{(k)}\right\|_\infty > M$，则停止计算，原问题或对偶问题无界；否则进行4。

4. 解方程 $\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ W & 0 & X \end{bmatrix}\begin{bmatrix} \Delta x^{(k)} \\ \Delta y^{(k)} \\ \Delta w^{(k)} \end{bmatrix} = \begin{bmatrix} \rho \\ \sigma \\ \mu e - XWe \end{bmatrix}$

其中$X = diag\left(x_1^{(k)}, x_2^{(k)}, \cdots, x_n^{(k)}\right)$，$W = diag\left(w_1^{(k)}, w_2^{(k)}, \cdots, w_n^{(k)}\right)$，

得解$\left(\Delta x^{(k)}, \Delta y^{(k)}, \Delta w^{(k)}\right)$，置

$$\lambda = \min\left\{ p\left[\max_{i,j}\left(-\frac{\Delta x_j^{(k)}}{x_j^{(k)}}, -\frac{\Delta w_i^{(k)}}{w_i^{(k)}}\right)\right]^{-1}, 1\right\}.$$

5. 令$x^{(k+1)} = x^{(k)} + \lambda \Delta x^{(k)}$，$y^{(k+1)} = y^{(k)} + \lambda \Delta y^{(k)}$，

$w^{(k+1)} = w^{(k)} + \lambda \Delta w^{(k)}$，置$k := k+1$，返回2。

三种解法比较

单纯形法最老牌,研究的最为透彻, 商业化的软件程序也最成熟.

椭球法像昙花一現, 虽然在理论上证明了线性规划问题可在多项式时间内求解, 但在实际应用上反而不如单纯形法来得有效便捷。

内点法是最新的设计, 理论上它比椭球法还要有效,实际应用上它也可与单纯形法相抗衡,不少的商业化软件已经上市, 前景甚佳。