第 6 章　约束非线性优化

第 2 节　制约函数法

2.1　外点惩罚函数法

2.2　内点障碍函数法

制约函数法，将约束非线性规划转化为一系列无约束极值问题，称为序列无约束极小化技术，（sequential unconstrained minimization technique，SUMT）。

常用的制约函数有两类：

✧ 惩罚函数（penalty function)——外点法

✧ 障碍函数（barrier function)——内点法

制约函数法一般能得到局部极小，不能保证全局最优；但若原问题是凸规划，则能求得全局极小。

2.1 外点惩罚函数法

一、外点法基本原理

（一）惩罚函数的结构

考虑非线性规划问题

$$\begin{aligned} &\min f(\mathbf{x}) \\ &s.t.\ g_j(\mathbf{x}) \geq 0,\ j = 1,2,\dots,l \end{aligned} \qquad (1)$$

记可行域： $\boldsymbol{R} = \{\mathbf{x} | g_j(\mathbf{x}) \geq 0, j = 1,2,\dots,l\}$。

定义函数：

$$\psi(t) = \begin{cases} 0, & if\ t \geq 0 \\ \infty, & if\ t < 0 \end{cases} \qquad (2)$$

如果把$g_j(\mathbf{x})$视为自变量 t，则：

当$\mathbf{x} \in \boldsymbol{R}$时，$g_j(\mathbf{x}) \geq 0$，有：$\psi(g_j(\mathbf{x})) = 0$；

当$\mathbf{x} \notin \boldsymbol{R}$时，$g_j(\mathbf{x}) < 0$，有：$\psi(g_j(\mathbf{x})) = \infty$。

再构造函数

$$\phi(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^{l} \psi(g_j(\mathbf{x})) \tag{3}$$

求解无约束问题：

$$\min \phi(\mathbf{x}) \tag{4}$$

若问题(4)有最优解$\mathbf{x}^*$，且$\psi(g_j(\mathbf{x}^*)) = 0$（意味着$\mathbf{x}^* \in \boldsymbol{R}$）。则$\mathbf{x}^*$不仅是问题（4）的极小点，也是原问题（1）的极小点。

用上述方法构造的函数$\psi(t)$在$t=0$处不连续，更没有导数，无法做数值计算；为此，将函数$\psi(t)$修改为

$$\psi(t)=[\min(0,t)]^2=\begin{cases}0, & if\ t\geq 0\\ t^2, & if\ t<0\end{cases} \tag{5}$$

修改后的函数$\psi(t)$满足：

当$\mathbf{x}\in \boldsymbol{R}$时，$g_j(\mathbf{x})\geq 0\Rightarrow \psi(g_j(\mathbf{x}))=0$

当$\mathbf{x}\notin \boldsymbol{R}$时，$g_j(\mathbf{x})<0\Rightarrow 0<\psi(g_j(\mathbf{x}))=[g_j(\mathbf{x})]^2<\infty$。

按照新的策略：$\psi(g_j(\mathbf{x}))=\{\min[0,g_j(\mathbf{x})]\}^2$，则得到的问题（4）并不完全等价于原问题（1）。

考虑做一个修正，取充分大的正数M，将$\phi(\mathbf{x})$改写为

$$P(\mathbf{x}, M) = f(\mathbf{x}) + M \sum_{j=1}^{l} \psi(g_j(\mathbf{x})) \qquad （6）$$

即，

$$P(\mathbf{x}, M) = f(\mathbf{x}) + M \sum_{j=1}^{l} \{\min[0, g_j(\mathbf{x})]\}^2 \qquad （7）$$

式（7）定义的函数$P(\mathbf{x}, M)$称为惩罚函数或罚函数；其中，$M \sum_{j=1}^{l} \{\min[0, g_j(\mathbf{x})]\}^2$称为惩罚项，$M$为罚因子。

设$\mathbf{x}(M)$为无约束极小化问题 $\min P(\mathbf{x}, M)$ 的极小点。

给定一个M，通过无约束极小化技术，即可得到一个极小点$\mathbf{x}(M)$。

命题 1　若罚函数$P(\mathbf{x},M)$的极小点$\mathbf{x}(M)$是原问题（1）的可行解，即$\mathbf{x}(M)\in \boldsymbol{R}$，则$\mathbf{x}(M)$就是问题（1）的极小点。

证明，因为$\mathbf{x}(M)$是函数$P(\mathbf{x},M)$的极小点，所以$P(\mathbf{x},M)\geq P(\mathbf{x}(M),M)$，即

$$f(\mathbf{x})+M\sum_{j=1}^{l}\{\min[0,g_j(\mathbf{x})]\}^2 \geq f(\mathbf{x}(M))+M\sum_{j=1}^{l}\{\min[0,g_j(\mathbf{x}(M))]\}^2$$

若$\mathbf{x}(M)\in \boldsymbol{R}$，则$\min[0,g_j(\mathbf{x}(M))]=0$，而对所有可行域内的$\mathbf{x}\in \boldsymbol{R}$，有$\min[0,g_j(\mathbf{x})]=0$，此时

$$f(\mathbf{x})\geq f(\mathbf{x}(M))$$

显然，$\mathbf{x}(M)$就是原问题的极小点。

注，在罚函数中，$g_j(\mathbf{x}) \geq 0$形式的约束，其二次惩罚项为$M\sum_{j=1}^{l}\{\min[0, g_j(\mathbf{x})]\}^2$，而不是$M\sum_{j=1}^{l}[g_j(\mathbf{x})]^2$。

例，

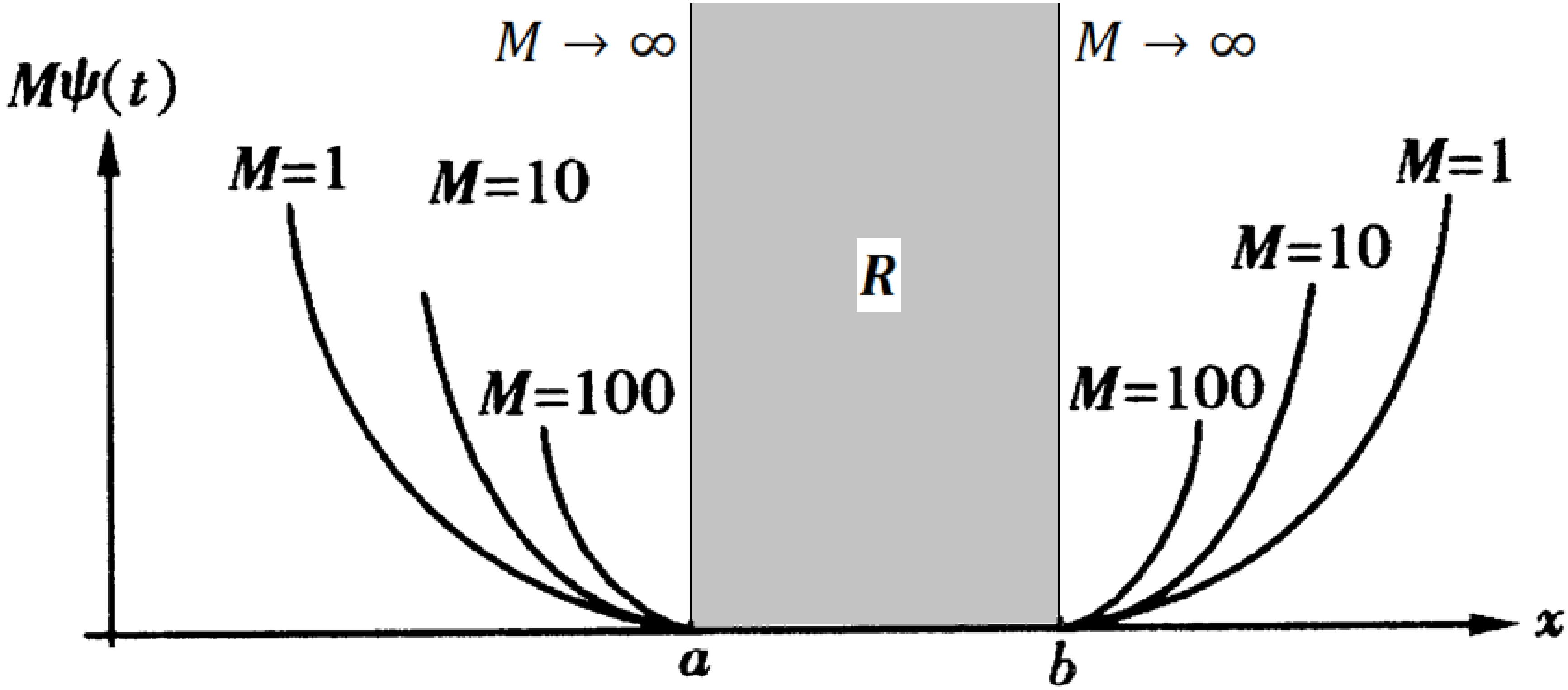
约束$g(x) = x - a \geq 0$对应的惩罚项为

$$M \cdot [\min(0, x - a)]^2$$

约束$g(x) = x - b \leq 0$对应的惩罚项为

$$M \cdot [\min(0, b - x)]^2$$

当$M \to \infty$，上述惩罚项之和就等价于原约束 $a \leq x \leq b$：



左半部表示$g(x) = x - a \geq 0$：$M \cdot [\min(0, x - a)]^2$

右半部表示$g(x) = x - b \leq 0$：$M \cdot [\min(0, b - x)]^2$

在求解中，若对某个罚因子M_1，罚函数$P(\mathbf{x}, M_1)$的极小点$\mathbf{x}(M_1) \notin \boldsymbol{R}$，即$\mathbf{x}(M_1)$是不可行解，那么就加大罚因子$M$的值，继续求解问题$P(\mathbf{x}, M)$的极小点。

随着罚因子M的值增大，惩罚函数中的惩罚项所起的作用也随之增大，$P(\mathbf{x}, M)$的极小点$\mathbf{x}(M)$与约束集$\boldsymbol{R}$的“距离”就越来越近。

当$M_1 < M_2 < \cdots < M_k < \cdots \to$无穷大，序列$\{\mathbf{x}(M_k)\}$就从可行域外部趋向于原问题（1）的极小点$\mathbf{x}^*$。

（二）外点罚函数方法的计算步骤

（1）取$M_1 > 0$，允许约束违反的误差$\varepsilon > 0$，令$k = 1$。

（2）构造罚函数，求无约束极值问题的最优解：

$$\mathbf{x}(M_k) = \arg\min_{\mathbf{x}} P(\mathbf{x}, M_k)$$

其中，$P(\mathbf{x}, M_k) = f(\mathbf{x}) + M_k \sum_{j=1}^{l} \{\min[0, g_j(\mathbf{x})]\}^2$。

（3）若对某约束函数$g_j(\mathbf{x})$，有$g_j(\mathbf{x}(M_k)) \leq -\varepsilon$，则增大罚因子：取$M_{k+1} > M_k$（例如，$M_{k+1} = 5M_k \ or \ 10M_k$），令$k = k + 1$，转第（2）步；

否则，若所有约束都满足：$g_j(\mathbf{x}(M_k)) > -\varepsilon$，则停止迭代，得到$\mathbf{x}^* \approx \mathbf{x}(M_k)$。

例 1，求解非线性规划

$$\begin{cases} \min & f(\mathbf{x}) = x_1 + x_2 \\ s.t. & g_1(\mathbf{x}) = -x_1^2 + x_2 \geq 0 \\ & g_2(\mathbf{x}) = x_1 \geq 0 \end{cases}$$

解，构造**罚函数**

$$P(\mathbf{x}, M) = \\ x_1 + x_2 + M\{[\min(0, -x_1^2 + x_2)]^2 + [\min(0, x_1)]^2\}$$

假设有一个违反所有约束的“外点” $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$，则有

$$-x_1^2 + x_2 < 0 \text{和} x_1 < 0$$

此时，罚函数为：

$$P(\mathbf{x},M)=x_1+x_2+M(-x_1^2+x_2)^2+Mx_1^2$$

令

$$\begin{cases}\dfrac{\partial P}{\partial x_1}=1+2M(-x_1^2+x_2)(-2x_1)+2Mx_1=0\\ \dfrac{\partial P}{\partial x_2}=1+2M(-x_1^2+x_2)=0\end{cases}$$

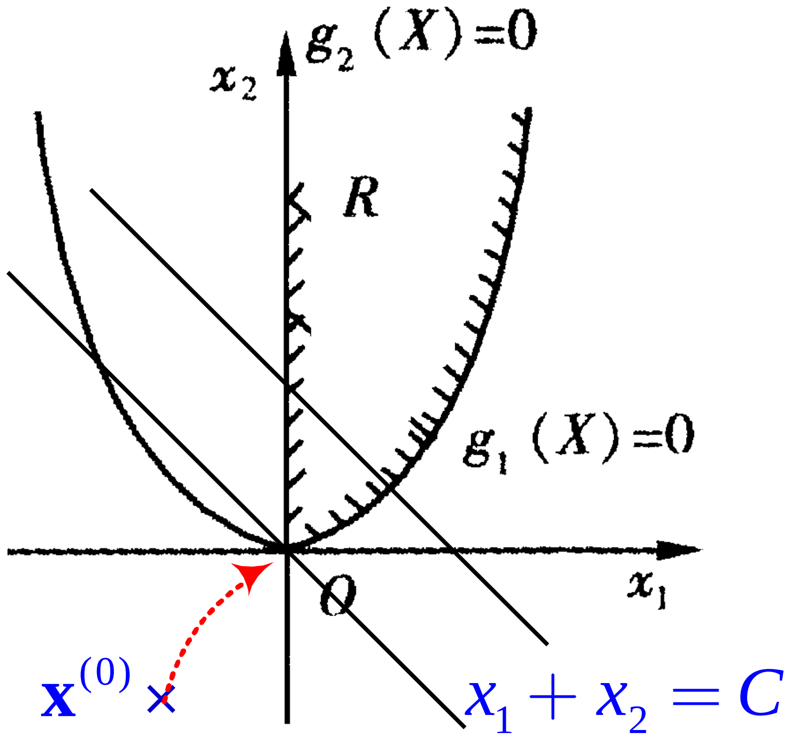
解得，$\mathbf{x}(M)=(-\frac{1}{2(M+1)},\frac{1}{4(M+1)^2}-\frac{1}{2M})^{\mathrm{T}}$；当$M\to\infty$，有

$$\mathbf{x}^*=\lim_{M\to\infty}\mathbf{x}(M)=(0,0)^{\mathrm{T}}$$

在实际工程应用中，迭代点$\mathbf{x}(M)$的解析表达式并不容易得到，一般要逐步增大罚因子M而实现优化：

M	$\mathbf{x}(M)$
1	$(-1/4,-7/16)^{\mathrm{T}}$
2	$(-1/6,-2/9)^{\mathrm{T}}$
3	$(-1/8,-29/192)^{\mathrm{T}}$
4	$(-1/10,-23/200)^{\mathrm{T}}$

当$M \to \infty$，$\mathbf{x}(M)$从可行域$\boldsymbol{R}$的外步逐步逼近边界点$\mathbf{x}^* = (0,0)^{\mathrm{T}}$。



由于$\mathbf{x}^* = (0,0)^{\mathrm{T}}$是原问题的可行解，由命题 **1**，它也就是原问题的最优解。

注：使用罚函数时，变量的非负约束也应考虑进去，如上例$x_1 \geq 0$，就要使用一个惩罚项$M[\min(0, x_1)]^2$。

外点法的特点：函数$P(\mathbf{x}, M)$是在整个$\mathbb{E}^n$空间内进行优化，初始点可任意选择，给计算带来了很大方便。

注：外点法也同样适用于同时含有等式和不等式约束的问题；含等式约束$h_i(\mathbf{x}) = 0$的罚函数可取为

$$P(\mathbf{x}, M_k) = f(\mathbf{x}) + M_k \sum_{j=1}^{l} \{\min[0, g_j(\mathbf{x})]\}^2 + M_k \sum_{j=1}^{l} [h_i(\mathbf{x})]^2$$

（三）罚因子的取值问题

根据命题 1，若足够大的罚因子M_k使得罚函数的最优解是原问题的可行解，则就得到了原问题的最优解。

但一般情况下，罚因子M_k不宜取得过大。

对于非常大的罚因子，在边界上函数变得非常“陡”，导致罚函数的性质很差，其 Hesse 矩阵接近奇异——行列式值趋于 0，这会导致很多理论上可行的极小化技术因为计算稳定性问题而变得不可用。

实际计算中，应该先使用较小的罚因子，然后逐步增大，当达到计算精度就停止——无限增大罚因子来无穷逼近真正最优解是不现实的。

若在很大的罚因子下求得的罚函数的极小点仍然不是原问题的可行解，那么就只能得到原问题的近似最优解——该近似解对某些约束有很小的违反（在容许误差$\varepsilon > 0$内）。

（四）“硬约束”问题的预处理

硬约束：不允许有丝毫违反（微小的“误差”也不允许）的约束称为“硬约束”。

设$g_k(\mathbf{x}) \geq 0$是不允许有丝毫违反的“硬约束”，令：

$$g'_k(\mathbf{x}) = g_k(\mathbf{x}) - \varepsilon \geq 0$$

其中$\varepsilon > 0$为算法的容许误差。

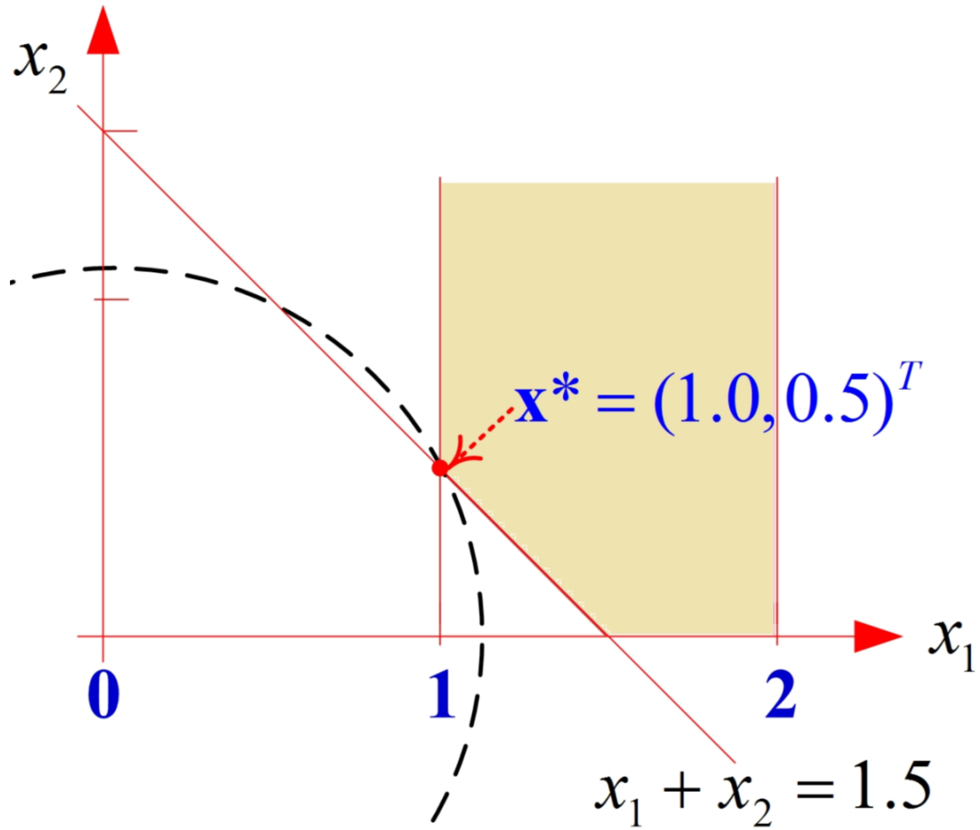
以新的约束$g'_k(\mathbf{x}) = g_k(\mathbf{x}) - \varepsilon \geq 0$构造罚函数，若近似最优解满足算法的容许误差：$g'_k(\mathbf{x}(M_k)) > -\varepsilon$，则必然有$g_k(\mathbf{x}(M_k)) \geq 0$，从而保证了“硬约束”的绝对满足。

二、外点法的数值计算

罚函数$P(\mathbf{x}, M_k)$往往很难转换为好的解析形式，此时就需要使用下降迭代算法等数值方法求解。

例 2，用外点法求解

$$\min x_1^2 + x_2^2$$
$$\text{s.t.} -x_1 - x_2 \le -1.5$$
$$1 \le x_1 \le 2$$



解，化标准形后有三个约束

$$x_1 + x_2 - 1.5 \ge 0$$
$$x_1 - 1 \ge 0$$
$$2 - x_1 \ge 0$$

仿照例 1 的做法，原始罚函数为

$$P(\mathbf{x}, M) = x_1^2 + x_2^2 + M[\min(0, x_1 + x_2 - 1.5)]^2$$
$$+M[\min(0, x_1 - 1)]^2 + M[\min(0,2 - x_1)]^2$$

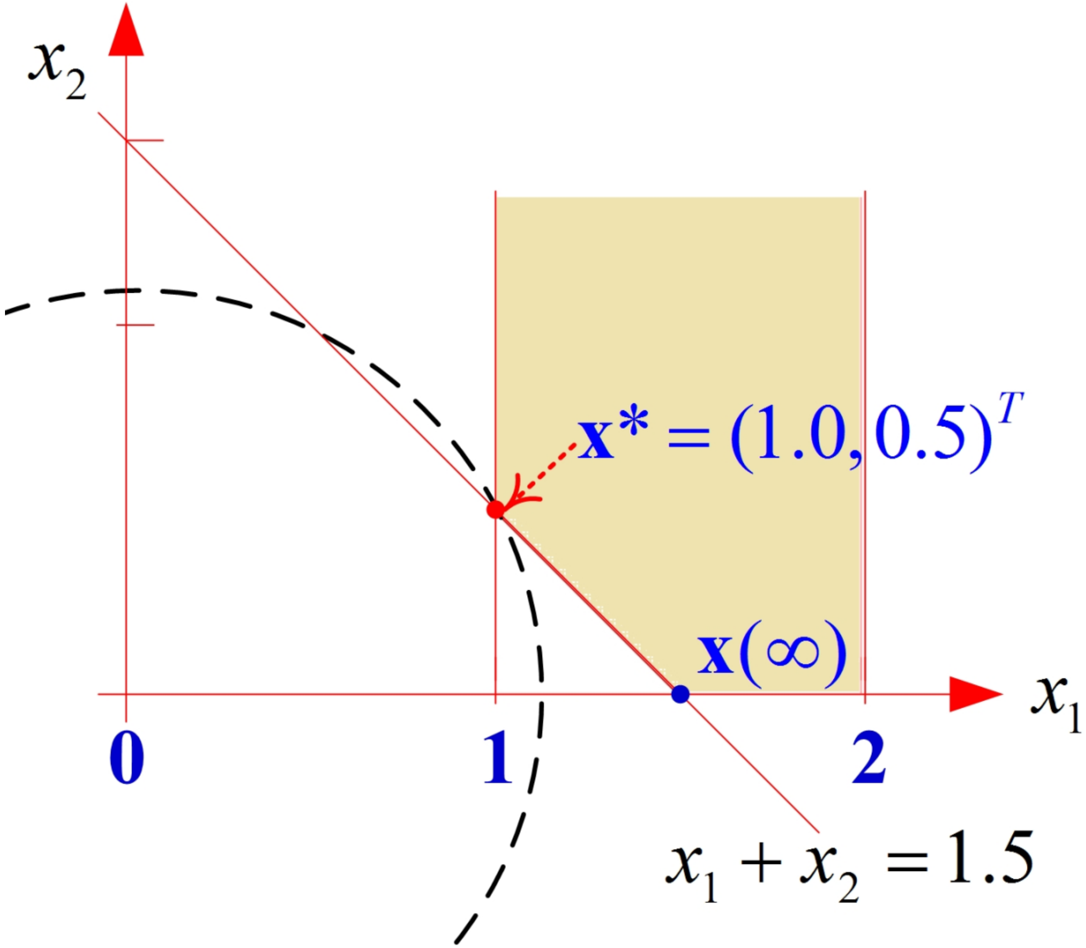
考虑 “同时违反所有约束” 的 “外点”，则罚函数可变换为

$$P'(\mathbf{x}, M) = x_1^2 + x_2^2 + M(x_1 + x_2 - 1.5)^2$$
$$+ M(x_1 - 1)^2 + M(2 - x_1)^2$$

根据一阶条件，得：

$$\mathbf{x}(M) = \begin{bmatrix} (6M^2 + 9M)/(4M^2 + 8M + 2) \\ 3M/(4M^2 + 8M + 2) \end{bmatrix} \xrightarrow{M\to\infty} \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$$

但这与真正最优解$\mathbf{x}^* = (1.0, 0.5)^{\mathrm{T}}$相距甚远……！



注意到，上述过程极小化的是函数$P'(\mathbf{x},M)$，而并非原始罚函数$P(\mathbf{x},M)$。

$$P'(\mathbf{x},M)=x_1^2+x_2^2+M(x_1+x_2-1.5)^2 \\ +M(x_1-1)^2+M(2-x_1)^2$$

$$\nparallel\ vs.$$

$$P(\mathbf{x},M)=x_1^2+x_2^2+M[\min(0,x_1+x_2-1.5)]^2 \\ +M[\min(0,x_1-1)]^2+M[\min(0,2-x_1)]^2$$

函数$P'(\mathbf{x},M)$存在的前提假设是：迭代点同时违反了所有约束，即$x_1 > 2$且$x_1 < 1$。

但同时违反这两个约束的“外点”并不存在，因此直接使用$P'(\mathbf{x},M)$作为罚函数并不正确。

正确做法：应该极小化$\boxed{\text{原始罚函数}}$，除非确实存在违反所有约束的点，否则并不能直接去掉惩罚项的所有$\min(\cdot)$函数——（例 **1** 只是运气好）。

例 2 的正确做法：

第一步，任意选定初始点，不妨选 $\mathbf{x}^{(0)} = (2.5, 0.5)^{\mathrm{T}}$，罚因子$M_0 = 1$，极小化原始罚函数$P(\mathbf{x}, M) = P(\mathbf{x}, 1)$：

$$P(\mathbf{x}, 1) = x_1^2 + x_2^2 + 1 \cdot [\min(0, x_1 + x_2 - 1.5)]^2 + 1 \cdot [\min(0, x_1 - 1)]^2 + 1 \cdot [\min(0, 2 - x_1)]^2$$

不好直接使用一阶条件，所以考虑下降迭代算法。

令$\mathbf{y}^{(0)} = \mathbf{x}^{(0)} = (2.5, 0.5)^{\mathrm{T}}$，在$\mathbf{y}^{(0)} = (2.5, 0.5)^{\mathrm{T}}$附近，罚函数变为：$P'(\mathbf{x}, 1) = x_1^2 + x_2^2 + (2 - x_1)^2$，于是原始罚函数$P(\mathbf{x}, 1)$在$\mathbf{y}^{(0)} = (2.5, 0.5)^{\mathrm{T}}$处的梯度为

$$\nabla P(\mathbf{x},1)|_{\mathbf{y}^{(0)}=(2.5,0.5)^{\mathrm{T}}} = (4x_1-4,2x_2)^{\mathrm{T}} = (6,1)^{\mathrm{T}}$$

运用最速下降法，令

$$\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \lambda\nabla P(\mathbf{y}^{(0)},1) = (2.5-6\lambda, 0.5-\lambda)^{\mathrm{T}}$$

代入原始罚函数$P(\mathbf{x},1)$，有

$$\begin{aligned}\phi(\lambda) = P(\mathbf{y}^{(1)},1) &= x_1^2+x_2^2+1\cdot[\min(0,x_1+x_2-1.5)]^2 \\ &+1\cdot[\min(0,x_1-1)]^2+1\cdot[\min(0,2-x_1)]^2 \\ &=(2.5-6\lambda)^2+(0.5-\lambda)^2+[\min(0,1.5-7\lambda)]^2 \\ &+[\min(0,1.5-6\lambda)]^2+[\min(0,-0.5+6\lambda)]^2\end{aligned}$$

需要求解子问题$\min \phi(\lambda)$，以获得下降迭代的最佳步长。

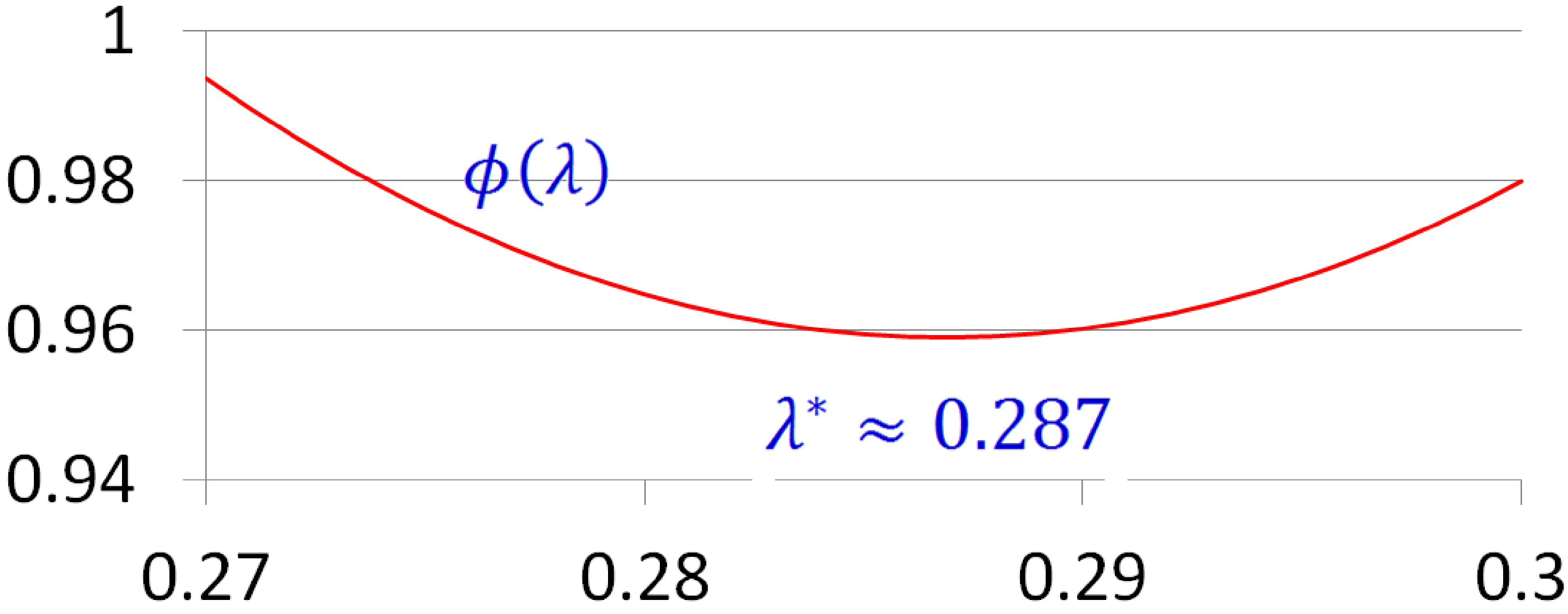
求解子问题$\min\ \phi(\lambda)$的方法：

运用一维搜索求$\phi(\lambda)$的极小点，先试探三个点：

$$\phi(\lambda_1 = 0.27) = 0.9938$$
$$\phi(\lambda_2 = 0.28) = 0.9648$$
$$\phi(\lambda_3 = 0.30) = 0.9800$$

可见$\phi(0.27) > \phi(0.28) < \phi(0.30)$，因此$\lambda$的最优值$\lambda^* \in [0.27, 0.30]$。

运用 **0.618** 法做一维搜索，得：$\lambda^* \approx 0.287$。



下降迭代算法下降到新点：

$$\mathbf{y}^{(1)} = (2.5 - 6\lambda^*, 0.5 - \lambda^*)^{\mathrm{T}} = (0.778, 0.213)^{\mathrm{T}}$$

在点$\mathbf{y}^{(1)} = (0.778, 0.213)^{\mathrm{T}}$附近，原始罚函数$P(\mathbf{x}, 1)$变换为：

$$P'(\mathbf{x},1) = x_1^2 + x_2^2 + (x_1 + x_2 - 1.5)^2 + (x_1 - 1)^2$$

于是：$\nabla P(\mathbf{y}^{(1)},1) = \begin{bmatrix} 6x_1 + 2x_2 - 5 \\ 2x_1 + 4x_2 - 3 \end{bmatrix} = \begin{bmatrix} 0.094 \\ -0.592 \end{bmatrix}$

继续迭代，令

$$\mathbf{y}^{(2)} = \mathbf{y}^{(1)} - \lambda \nabla P(\mathbf{y}^{(1)},1)$$
$$= \begin{bmatrix} 0.778 - 0.094\lambda \\ 0.213 + 0.592\lambda \end{bmatrix}$$

代入原始罚函数$P(\mathbf{x},1)$，有

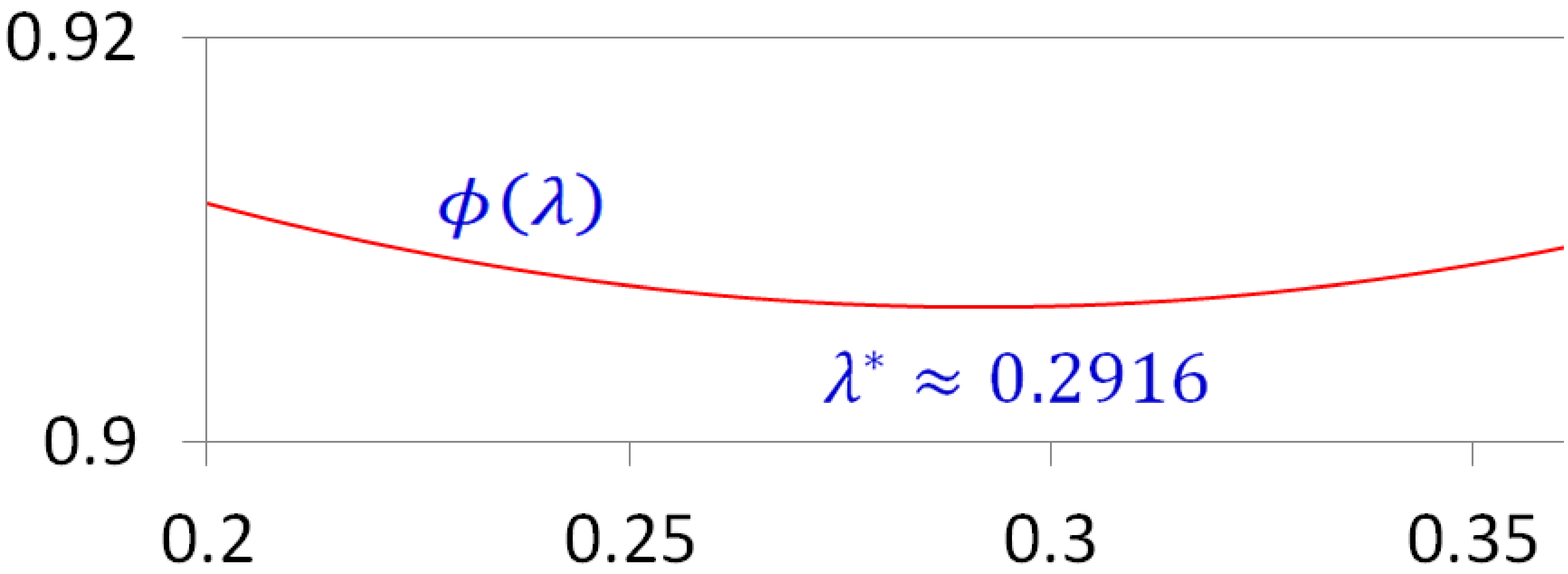
$$
\begin{aligned}
\phi(\lambda) = P(\mathbf{y}^{(2)}, 1) &= x_1^2 + x_2^2 + 1\cdot[\min(0, x_1 + x_2 - 1.5)]^2 \\
&+1\cdot[\min(0, x_1 - 1)]^2 + 1\cdot[\min(0, 2 - x_1)]^2 \\
&= (0.778 - 0.094\lambda)^2 + (0.213 + 0.592\lambda)^2 \\
&+[\min(0, -0.509 + 0.498\lambda)]^2 \\
&+[\min(0, -0.222 - 0.094\lambda)]^2 + [\min(0, 1.222 + 0.094\lambda)]^2
\end{aligned}
$$

求解$\min\phi(\lambda)$。

易知，$\phi(0.2) > \phi(0.3) < \phi(0.38)$，于是：

$$\lambda^* \in [0.2, 0.38]$$

运用黄金分割做一维搜索，得：$\lambda^* \approx 0.2916$。



下降迭代算法下降到新点：

$$\mathbf{y}^{(2)} = \begin{bmatrix} 0.778 - 0.094\lambda^* \\ 0.213 + 0.592\lambda^* \end{bmatrix} = \begin{bmatrix} 0.751 \\ 0.385 \end{bmatrix}$$

在$\mathbf{y}^{(2)}=(0.751,0.385)^{\mathrm{T}}$附近，原始罚函数$P(\mathbf{x},1)$成为：

$$P'(\mathbf{x},1)=x_1^2+x_2^2+(x_1+x_2-1.5)^2+(x_1-1)^2$$

于是：$\nabla P(\mathbf{y}^{(2)},1)=\begin{bmatrix}6x_1+2x_2-5\\2x_1+4x_2-3\end{bmatrix}=\begin{bmatrix}0.276\\0.042\end{bmatrix}$

继续迭代，令

$$\mathbf{y}^{(3)}=\mathbf{y}^{(2)}-\lambda\nabla P(\mathbf{y}^{(2)},1)=\begin{bmatrix}0.751-0.276\lambda\\0.385-0.042\lambda\end{bmatrix}$$

代入原始罚函数$P(\mathbf{x},1)$，有

$$\phi(\lambda) = P(\mathbf{y}^{(3)}, 1) = x_1^2 + x_2^2 + 1 \cdot [\min(0, x_1 + x_2 - 1.5)]^2$$
$$+1 \cdot [\min(0, x_1 - 1)]^2 + 1 \cdot [\min(0, 2 - x_1)]^2$$
$$= (0.751 - 0.276\lambda)^2 + (0.385 - 0.042\lambda)^2$$
$$+[\min(0, -0.364 - 0.318\lambda)]^2$$
$$+[\min(0, -0.249 - 0.276\lambda)]^2 + [\min(0, 1.249 + 0.276\lambda)]^2$$

运用黄金分割一维搜索，求解$\min \phi(\lambda)$，得到：

$$\lambda^* \approx 0.1527$$

下降迭代到新点：$\mathbf{y}^{(3)} = \begin{bmatrix} 0.751 - 0.276\lambda^* \\ 0.385 - 0.042\lambda^* \end{bmatrix} = \begin{bmatrix} 0.709 \\ 0.379 \end{bmatrix}$

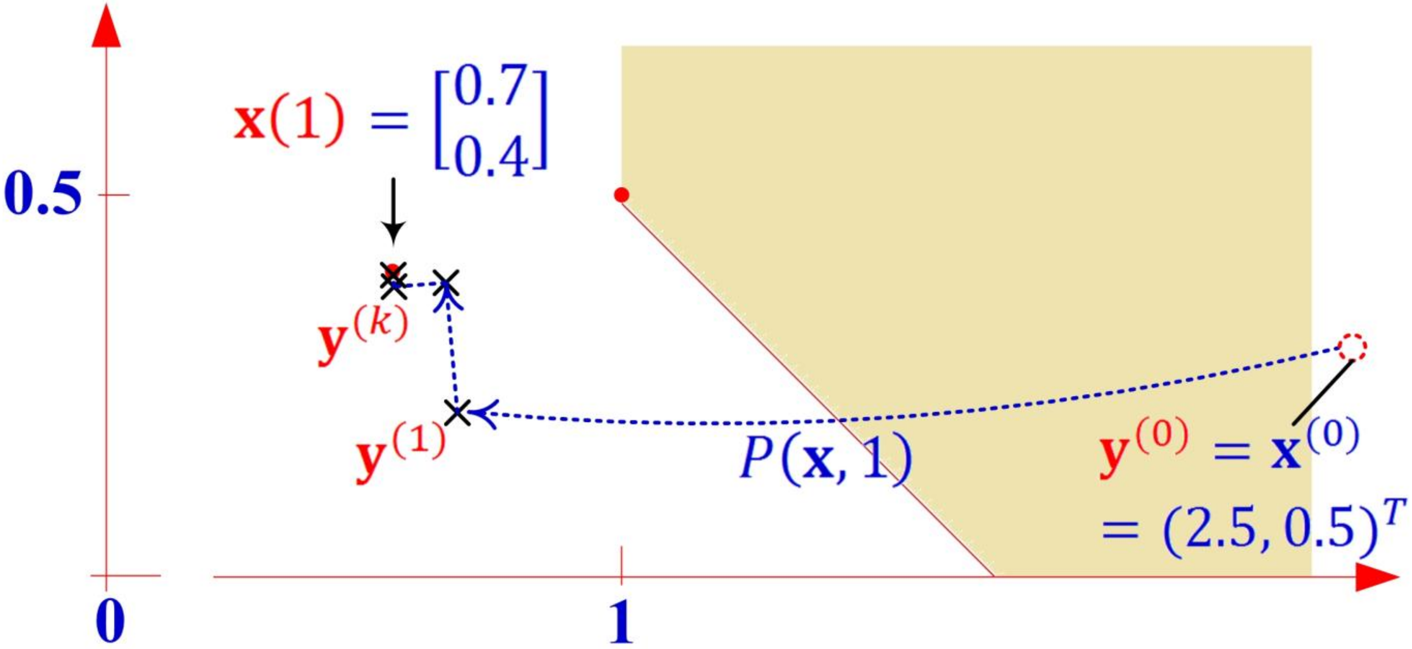
……

迭代足够多次，可知$\mathbf{y}^{(k)}$将收敛于

$$\mathbf{y}^* = \begin{bmatrix} 0.7 \\ 0.4 \end{bmatrix},\ \nabla P(\mathbf{y}^*, 1) = \begin{bmatrix} 6x_1 + 2x_2 - 5 \\ 2x_1 + 4x_2 - 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

所以，在罚因子$M_0 = 1$时，原始罚函数$P(\mathbf{x}, 1)$的无约束极小点为：

$$\mathbf{x}(1) = \mathbf{y}^* = \begin{bmatrix} 0.7 \\ 0.4 \end{bmatrix}$$



第二步，增大罚因子为$M_1 = 10M_0 = 10$，新的初始“外点”选定为：$\mathbf{x}^{(1)} = \mathbf{x}(1) = (0.7, 0.4)^{\mathrm{T}}$，原始罚函数为

$$P(\mathbf{x}, 10) = x_1^2 + x_2^2 + 10 \cdot [\min(0, x_1 + x_2 - 1.5)]^2$$
$$+10 \cdot [\min(0, x_1 - 1)]^2 + 10 \cdot [\min(0,2 - x_1)]^2$$

启动最速下降算法：令$\mathbf{y}^{(0)} = \mathbf{x}^{(1)} = (0.7, 0.4)^{\mathrm{T}}$，在点$\mathbf{y}^{(0)} = (0.7, 0.4)^{\mathrm{T}}$附近，原始罚函数$P(\mathbf{x}, 10)$变为

$$P'(\mathbf{x}, 10) = x_1^2 + x_2^2 + 10(x_1 + x_2 - 1.5)^2 + 10(x_1 - 1)^2$$

罚函数$P(\mathbf{x}, 10)$的梯度为：

$$\nabla P(\mathbf{x},10)|_{\mathbf{y}^{(0)}} = \begin{bmatrix} 42x_1 + 20x_2 - 50 \\ 20x_1 + 22x_2 - 30 \end{bmatrix} = \begin{bmatrix} -12.6 \\ -7.2 \end{bmatrix}$$

令：$\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \lambda\nabla P(\mathbf{y}^{(0)},10) = \begin{bmatrix} 0.7 + 12.6\lambda \\ 0.4 + 7.2\lambda \end{bmatrix}$

代入本步骤的原始罚函数$P(\mathbf{x},10)$，有

$$\begin{aligned}\phi(\lambda) = P(\mathbf{y}^{(1)},10) = (0.7 + 12.6\lambda)^2 + (0.4 + 7.2\lambda)^2 \\ +10\cdot[\min(0,-0.4 + 19.8\lambda)]^2 \\ +10\cdot[\min(0,-0.3 + 12.6\lambda)]^2 \\ +10\cdot[\min(0,1.3 - 12.6\lambda)]^2\end{aligned}$$

一维搜索求解$\min\phi(\lambda)$，得$\lambda^* \approx 0.0184$。

下降迭代到新点：$\mathbf{y}^{(1)}=\begin{bmatrix}0.7+12.6\lambda^*\\0.4+7.2\lambda^*\end{bmatrix}=\begin{bmatrix}0.932\\0.532\end{bmatrix}$

在点$\mathbf{y}^{(1)}$附近，原始罚函数$P(\mathbf{x},10)$成为：

$$P'(\mathbf{x},10)=x_1^2+x_2^2+10(x_1+x_2-1.5)^2+10(x_1-1)^2$$

罚函数$P(\mathbf{x},10)$的梯度为：

$$\nabla P(\mathbf{x},10)|_{\mathbf{y}^{(1)}}=\begin{bmatrix}42x_1+20x_2-50\\20x_1+22x_2-30\end{bmatrix}=\begin{bmatrix}-0.216\\0.344\end{bmatrix}$$

令：$\mathbf{y}^{(2)}=\mathbf{y}^{(1)}-\lambda\nabla P(\mathbf{y}^{(1)},10)=\begin{bmatrix}0.932+0.216\lambda\\0.532-0.344\lambda\end{bmatrix}$

代入本步骤的原始罚函数$P(\mathbf{x},10)$，有

$$\phi(\lambda) = (0.932 + 0.216\lambda)^2 + (0.532 - 0.344\lambda)^2$$
$$+10 \cdot [\min(0, -0.036 - 0.128\lambda)]^2$$
$$+10 \cdot [\min(0, -0.068 + 0.216\lambda)]^2$$
$$+10 \cdot [\min(0, 1.068 - 0.216\lambda)]^2$$

一维搜索求解$\min \phi(\lambda)$，得$\lambda^* \approx 0.1037$。

迭代到新点：

$$\mathbf{y}^{(2)} = \begin{bmatrix} 0.932 + 0.216\lambda^* \\ 0.532 - 0.344\lambda^* \end{bmatrix} = \begin{bmatrix} 0.9544 \\ 0.4963 \end{bmatrix}$$

…… ……

继续迭代足够多次，可知$\mathbf{y}^{(k)}$将收敛于

$$\mathbf{y}^* \approx \begin{bmatrix} 0.9542 \\ 0.4962 \end{bmatrix},\ \nabla P(\mathbf{y}^*, 1) \approx \begin{bmatrix} 0.0004 \\ 0.0004 \end{bmatrix}$$

该点就是罚因子$M_1 = 10$时，原始罚函数$P(\mathbf{x}, 10)$的无约束极小点：

$$\mathbf{x}(M) = \mathbf{x}(10) = \begin{bmatrix} 0.9542 \\ 0.4962 \end{bmatrix}$$

第三步，增大罚因子为$M_2 = 10M_1 = 100$，新的初始“外点”为：$\mathbf{x}^{(2)} = \mathbf{x}(10) = (0.9542, 0.4962)^{\mathrm{T}}$，其原始罚函数为

$$P(\mathbf{x}, 100) = x_1^2 + x_2^2 + 100 \cdot [\min(0, x_1 + x_2 - 1.5)]^2 + 100 \cdot [\min(0, x_1 - 1)]^2 + 100 \cdot [\min(0, 2 - x_1)]^2$$

启动最速下降算法，令$\mathbf{y}^{(0)} = \mathbf{x}^{(2)}$，在$\mathbf{y}^{(0)}$附近，原始罚函数$P(\mathbf{x}, 100)$成为：

$$P'(\mathbf{x}, 100) = x_1^2 + x_2^2 + 100[(x_1 + x_2 - 1.5)^2 + (x_1 - 1)^2]$$

可得原始罚函数$P(\mathbf{x}, 100)$的梯度为：

$$\nabla P(\mathbf{x},100)|_{\mathbf{y}^{(0)}} = \begin{bmatrix} 402x_1 + 200x_2 - 500 \\ 200x_1 + 202x_2 - 300 \end{bmatrix} = \begin{bmatrix} -17.17 \\ -8.9276 \end{bmatrix}$$

令：$\mathbf{y}^{(1)} = \mathbf{y}^{(0)} - \lambda \nabla P(\mathbf{y}^{(0)}, 100) = \begin{bmatrix} 0.9542 + 17.17\lambda \\ 0.4962 + 8.9276\lambda \end{bmatrix}$

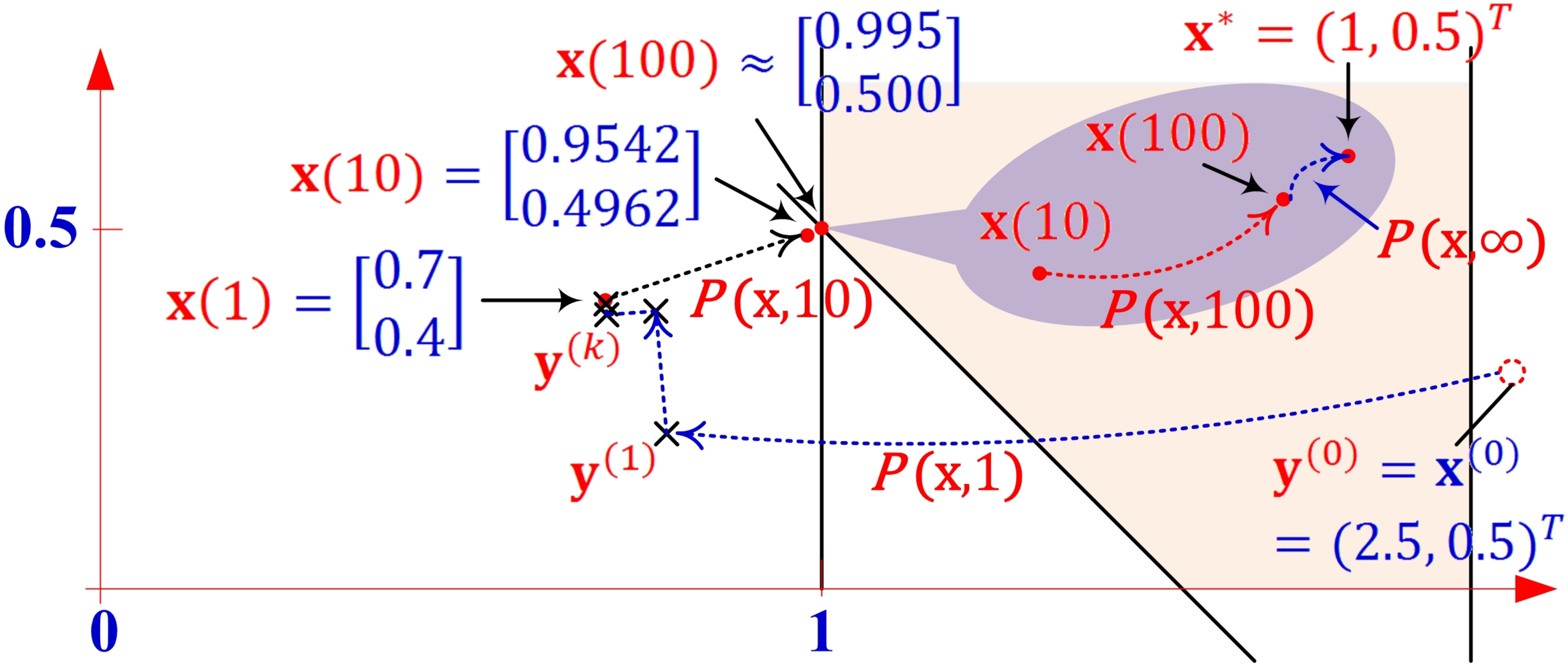
代入原始罚函数，求解$\min \phi(\lambda)$并迭代计算新点。

$\cdots \cdots$

足够多次迭代后，$P(\mathbf{x},100)$极小点收敛于

$$\mathbf{y}^* \approx (0.995, 0.500)^{\mathrm{T}} \Rightarrow \mathbf{x}(100) = (0.995, 0.500)^{\mathrm{T}}$$

重复上述各过程，原问题的近似最优解将逐渐逼近其真正最优点：$\mathbf{x}^* = \mathbf{x}(\infty) = (1, 0.5)^{\mathrm{T}}$。



对每一个罚因子的外点罚函数$P(\mathbf{x}, M)$进行极小化时，都嵌套了一个下降迭代算法：

（1）给定罚因子M_k，令初始“外点”$\mathbf{x}^{(k)} = \mathbf{x}(M_{k-1})$为下降迭代算法的初始点：$\mathbf{y}^{(0)} = \mathbf{x}^{(k)}$，将问题转换为关于下降迭代步长$\lambda$的单变量问题$\min\phi(\lambda)$。

（2）运用单变量优化技术（如一维搜索），确定最优步长λ，得到新的下降迭代点$\mathbf{y}^{(k)}$，直到$\mathbf{y}^{(k)}$收敛到最优点$\mathbf{y}^*$，记$\mathbf{x}(M_k) = \mathbf{y}^*$为罚函数$P(\mathbf{x}, M_k)$的极小点。

（3）初始“外点”更新为$\mathbf{x}^{(k+1)} = \mathbf{x}(M_k)$，增大罚因子$M_{k+1} = cM_k, c > 1$，确定新的罚函数$P(\mathbf{x}, M_{k+1})$，返回步骤（1）。

2.2 内点障碍函数法

一、内点法基本原理

内点法将初始点取在可行域内部，并在可行域边界设置一道“障碍”，当迭代点靠近边界时，障碍函数值迅速增大，迫使迭代点始终留在可行域内部。

障碍函数的特性：在可行域$\boldsymbol{R}$内部距离边界较远的地方，障碍函数与原目标函数$f(\mathbf{x})$尽可能相近似；而在接近可行域$\boldsymbol{R}$的边界时有任意大的值。

将非线性规划（1）转化为系列无约束极小化问题：

$$\min_{\mathbf{x}\in \boldsymbol{R}_0} \bar{P}(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^{l} 1/g_j(\mathbf{x}) \qquad (8)$$

或

$$\bar{P}(\mathbf{x}, r_k) = f(\mathbf{x}) - r_k \sum_{j=1}^{l} \log(g_j(\mathbf{x})) \qquad (9)$$

$$\boldsymbol{R}_0 = \{\mathbf{x} | g_j(\mathbf{x}) > 0, j = 1,2, \dots, l\} \qquad (10)$$

其中$r_k > 0$，为障碍因子；（8）式和（9）式右端第二项称为障碍项。

显然，在原问题可行域$\boldsymbol{R}$的边界上（即至少有一个$g_j(\mathbf{x}) = 0$），$\bar{P}(\mathbf{x}, r_k)$为正无穷大。因此从内部点开始最小化函数$\bar{P}(\mathbf{x}, r_k)$的无约束迭代过程不会跑出可行域。

从可行域内部一点$\mathbf{x}^{(0)}$出发，按无约束极小化方法求解问题（8），随着障碍因子r_k的减小，即

$$r_1 > r_2 > \cdots > r_k > \cdots > 0$$

障碍项所起的作用也越来越小。求出的$\min \bar{P}(\mathbf{x}, r_k)$的解$\mathbf{x}(r_k)$也逐步逼近原问题（1）的极小解$\mathbf{x}^*$。

若原问题极小解在可行域边界上，则随着r_k的减小，障碍作用逐步降低，所求出的障碍函数极小解不断靠近边界，直至满足某一精度要求为止 [1]。

[1]若采用障碍函数，即便真正的极小点在可行域R的边界上，所得到的结果也不会恰好在R的边界上。

内点法的迭代步骤：

（1）取$r_1>0$（如取$r_1=1$），允许误差$\varepsilon>0$。

（2）找出任一可行内点$\mathbf{x}^{(0)}\in\boldsymbol{R}_0$，令$k=1$。

（3）以倒数或对数函数做障碍项构造障碍函数。

（4）以$\mathbf{x}^{(k-1)}\in\boldsymbol{R}_0$为初始点，对障碍函数进行无约束极小化，得到下一迭代点$\mathbf{x}^{(k)}$：

$$\min_{\mathbf{x}\in\boldsymbol{R}_0}\bar{P}(\mathbf{x},r_k)=\bar{P}(\mathbf{x}^{(k)},r_k)$$

其中，障碍项会自动保证上述过程不会跑到可行域之外，即保证：$\mathbf{x}^{(k)}=\mathbf{x}(r_k)\in\boldsymbol{R}_0$。

（5）检验收敛准则（障碍项绝对值足够小）：

$$r_k \sum_{j=1}^{l} 1/g_j(\mathbf{x}^{(k)}) \le \varepsilon \text{或} \left| r_k \sum_{j=1}^{l} \log(g_j(\mathbf{x}^{(k)})) \right| \le \varepsilon$$

如满足，则以$\mathbf{x}^{(k)}$为原问题近似极小解$\mathbf{x}^*$；否则，取$r_{k+1} < r_k$（如$r_{k+1} = r_k/10$或$r_k/5$），令$k = k+1$，转第（3）步继续迭代。

注：收敛准则还可采用其他形式，如：

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\| < \varepsilon \ \text{或} \ \left| f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \right| < \varepsilon$$

类似外点法，为防止障碍函数奇异，障碍因子r_k要逐步减小，而不能直接赋予r_k过小的值。

例 3，用内点法求解

$$
\begin{aligned}
\min\ & f(\mathbf{x}) = (x_1+1)^3/3 + x_2 \\
s.t.\ & g_1(\mathbf{x}) = x_1 - 1 \ge 0 \\
& g_2(\mathbf{x}) = x_2 \ge 0
\end{aligned}
$$

解，构造障碍函数

$$\bar{P}(\mathbf{x}, r) = (x_1+1)^3/3 + x_2 + r/(x_1-1) + r/x_2$$

一阶条件：$\begin{cases} \partial \bar{P}(\mathbf{x}, r)/\partial x_1 = (x_1+1)^2 - r/(x_1-1)^2 = 0 \\ \partial \bar{P}(\mathbf{x}, r)/\partial x_2 = 1 - r/x_2^2 = 0 \end{cases}$

解方程组，得：

$$\mathbf{x}(r)=\begin{bmatrix}\sqrt{1+\sqrt{r}}\\ \sqrt{r}\end{bmatrix}$$

由于$x_1-1=\sqrt{1+\sqrt{r}}-1>0, x_2=\sqrt{r}>0$，因此 $\mathbf{x}(r)$ 满足所有约束，是一个内点，满足内点法的要求。

直接对r取极限，即可得到最优解：

$$\mathbf{x}_{\min}=\lim_{r\to 0}\begin{bmatrix}\sqrt{1+\sqrt{r}}\\ \sqrt{r}\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}$$

例 4，用内点法求解

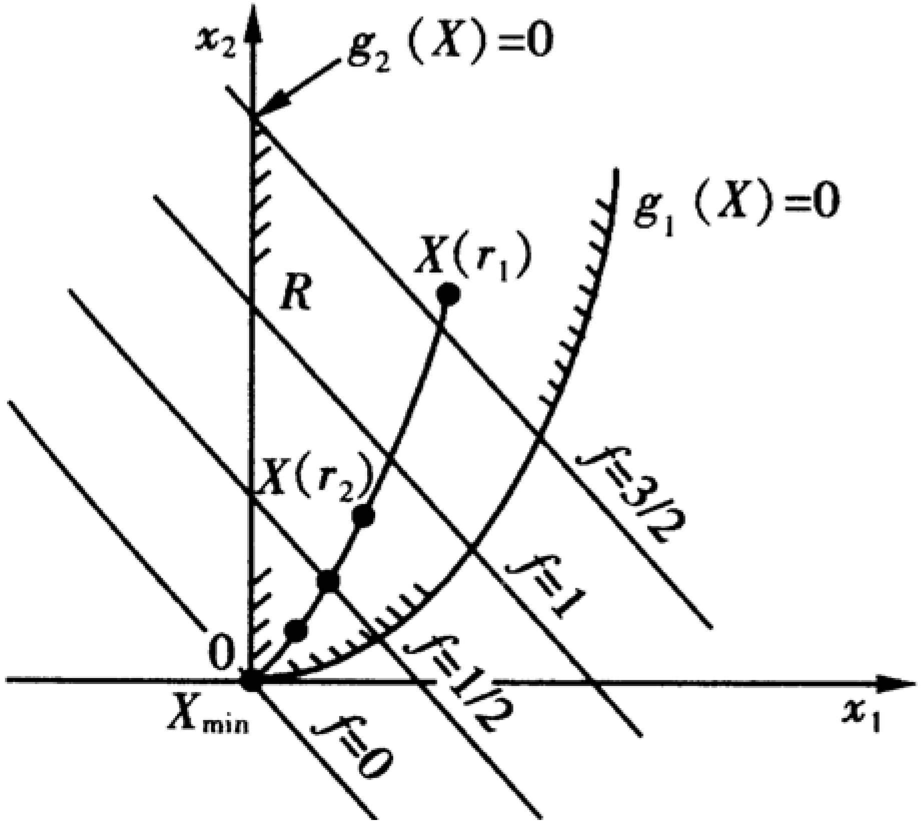
$$\min\ f(\mathbf{x}) = x_1 + x_2$$
$$s.t.\ g_1(\mathbf{x}) = -x_1^2 + x_2 \geq 0$$
$$g_2(\mathbf{x}) = x_1 \geq 0$$

解，构造障碍函数如下

$$\bar{P}(\mathbf{x}, r) = x_1 + x_2 - r\log(-x_1^2 + x_2) - r\log x_1$$

难以解析求解，以最速下降法求解不同罚因子 r 下的最优解，直到收敛。

障碍因子	r	$x_1(r)$	$x_2(r)$
r_1	1.000	0.500	1.250
r_2	0.500	0.309	0.595
r_3	0.250	0.183	0.283
r_4	0.100	0.085	0.107
r_5	0.0001	0.000	0.000



最优解收敛于：$\mathbf{x}^* = (0,0)^{\mathrm{T}}$。

二、初始内点问题

内点法需要找出某个内点作为初始点，这可以通过求解一系列内点优化问题的迭代过程实现。

首先，任意找一个点$\mathbf{x}^{(0)}$，考察满足约束的情况：

$$\boldsymbol{S}_0=\{j|g_j(\mathbf{x}^{(0)})\le 0, 1\le j\le l\}$$

$$\boldsymbol{T}_0=\{j|g_j(\mathbf{x}^{(0)})> 0, 1\le j\le l\}$$

$\boldsymbol{S}_0$表示不满足约束的下标集合（不是内点），$\boldsymbol{T}_0$表示满足约束的下标集合。如果$\boldsymbol{S}_0$为空集，则$\mathbf{x}^{(0)}$就是初始内点。

否则，若$\boldsymbol{S}_0\neq\phi$，则构造如下优化问题

$$\min \tilde{P}(\mathbf{x}, r_0) = -\sum_{j\in \boldsymbol{S}_0} g_j(\mathbf{x}) + r_0 \sum_{j\in \boldsymbol{T}_0} 1/g_j(\mathbf{x})$$

其中$r_0 > 0$；设得到最优解$\mathbf{x}^{(1)}$。

因为$\min \tilde{P}(\mathbf{x}, r_0) \Leftrightarrow \max \sum_{j\in \boldsymbol{S}_0} g_j(\mathbf{x})$，所以最优解 $\mathbf{x}^{(1)}$就有可能通过约束函数的极大化过程使原来不满足的约束从$g_j(\mathbf{x}^{(0)}) \le 0, j \in \boldsymbol{S}_0$，变为$g_j(\mathbf{x}^{(1)}) > 0$。

而且，若以$\mathbf{x}^{(0)}$作为初始点求解$\min \tilde{P}(\mathbf{x}, r_0)$，因障碍项$r_0 \sum_{j\in \boldsymbol{T}_0} 1/g_j(\mathbf{x})$的存在，最优解$\mathbf{x}^{(1)}$不会违反在$\mathbf{x}^{(0)}$已被满足的约束，即必然有$g_j(\mathbf{x}^{(1)}) > 0, \text{for } j \in \boldsymbol{T}_0$。

上述过程使新点$\mathbf{x}^{(1)}$处被满足的约束集合扩大为

$$\begin{aligned}\boldsymbol{T}_1 &= \{j | g_j(\mathbf{x}^{(1)}) > 0, 1 \le j \le l\} \\ &= \boldsymbol{T}_0 \cup \{t | g_t(\mathbf{x}^{(0)}) \le 0 \text{ and/but } g_t(\mathbf{x}^{(1)}) > 0\}\end{aligned}$$

而在新点$\mathbf{x}^{(1)}$，被违反的约束集合缩小为

$$\begin{aligned}\boldsymbol{S}_1 &= \{j | g_j(\mathbf{x}^{(1)}) \le 0, 1 \le j \le l\} \\ &= \boldsymbol{S}_0 \backslash \{t | g_t(\mathbf{x}^{(0)}) \le 0 \text{ and } g_t(\mathbf{x}^{(1)}) > 0\}\end{aligned}$$

如果$\boldsymbol{S}_1$不是空集，则继续构造新的最小化问题：

$$\min \tilde{P}(\mathbf{x}, r_1) = -\sum_{j\in \boldsymbol{S}_1} g_j(\mathbf{x}) + r_1 \sum_{j\in \boldsymbol{T}_1} 1/g_j(\mathbf{x})$$

其中$r_1 < r_0$，比如$r_1 = r_0/10$，得到最优解$\mathbf{x}^{(2)}$。

同理，在$\mathbf{x}^{(2)}$，违反约束的指标集将被缩小为

$$\begin{aligned}\boldsymbol{S}_2 &= \{j | g_j(\mathbf{x}^{(2)}) \le 0, 1 \le j \le l\} \\ &= \boldsymbol{S}_1 \backslash \{t | g_t(\mathbf{x}^{(1)}) \le 0 \text{ and } g_t(\mathbf{x}^{(2)}) > 0\}\end{aligned}$$

如此循环直到$\boldsymbol{S}_k = \text{空集}$，则对应的最优点$\mathbf{x}^{(k)}$就是原问题的内点；

如果原问题确实有可行解，则运用障碍函数不断构造内点无约束优化问题，最终就可以找到一个初始可行解。

求初始内点的迭代步骤：

（1）任取一点$\mathbf{x}^{(0)}$，$r_0 > 0$（如$r_0 = 1$），令$k = 0$。

（2）在第k步，定出指标集$\boldsymbol{S}_k$及$\boldsymbol{T}_k$

$$\boldsymbol{S}_k = \{j | g_j(\mathbf{x}^{(k)}) \leq 0, 1 \leq j \leq l\}$$

$$\boldsymbol{T}_k = \{j | g_j(\mathbf{x}^{(k)}) > 0, 1 \leq j \leq l\}$$

（3）如果$\boldsymbol{S}_k = \Phi$，为空集，则$\mathbf{x}^{(k)}$就是所需的初始内点，迭代停止。否则转第（4）步。

（4）构造函数

$$\tilde{P}(\mathbf{x}, r_k) = -\sum_{j\in \boldsymbol{S}_k} g_j(\mathbf{x}) + r_k \sum_{j\in \boldsymbol{T}_k} 1/g_j(\mathbf{x})$$

其中，$r_k < r_{k-1}$（如$r_k = r_{k-1}/10$）。以$\mathbf{x}^{(k)}$为初始点求解无约束问题$\min\tilde{P}(\mathbf{x}, r_k)$，得到最优点$\mathbf{x}^{(k+1)}$。

$\min\tilde{P}(\mathbf{x}, r_k)$可使某些$g_j(\mathbf{x}^{(k)}) \le 0$的约束在$\mathbf{x}^{(k+1)}$点变为$g_j(\mathbf{x}^{(k+1)}) > 0$；而障碍项$r_k \sum_{j\in \boldsymbol{T}_k} 1/g_j(\mathbf{x})$保证了<u>已经满足的约束不会被违反</u>，因此在新的最优点$\mathbf{x}^{(k+1)}$，不可行约束集$\boldsymbol{S}_k$将被缩小，这使$\mathbf{x}^{(k+1)}$比 $\mathbf{x}^{(k)}$更趋向内点一些。

（5）令$r_{k+1} < r_k$，$k = k + 1$，转（2），直到得到不违反所有约束的内点。