运筹学通论I

胡晓东

应用数学研究所

中国科学院数学与系统科学研究院

Http://www.amt.ac.cn/member/huxiaodong/

提纲

20世纪数学的五大指导理论　　Five Golden Rules

叶其孝、刘宝光　　Great Theories of 20th Century Math
上海教育出版社，2000　　-and Why They Matter

1. 线性规划 对偶定理
2. 博弈论 极大极小定理
3. 非线性规划 K-K-T 定理
4. 计算/算法理论 停机定理, 库克定理
   拓扑学 不动点定理
   奇点理论 莫尔斯定理
5. 组合最优化 算法设计技巧

运筹学

- 模型
- 理论
- 算法

参考书目

Nonlinear Programming - Theory and Algorithms
Mokhtar S. Bazaraa, C. M. Shetty
John Wiley & Sons, Inc. 1979 (2nd Edition, 1993)

Linear and Nonlinear Programming
David G. Luenberger
Addison-Wesley Publishing Company, 2nd Edition, 1984/2003..

Convex Analysis **
R. T. Rockafellar
Princeton Landmarks in Mathematics and Physics, 1996.

Optimization and Nonsmooth Analysis **
Frank H. Clarke
SIAM, 1990.

3. 带约束算法 - 罚函数方法

为了引入**罚函数**，我们考虑如下优化问题：

求最小值 $f(x)$
满足条件 $h(x) = 0$

假定我们用如下无约束优化问题代替上述优化问题：

求最小值 $f(x) + \mu h^2(x)$
满足条件 $x \in E^n$

直观地理解，上述新问题的最优解一定会使得 $h^2(x)$ 接近 0，否则就会产生很大的**惩罚项** $\mu h^2(x)$ 。

3. 带约束算法 - 罚函数方法（续一）

下面再考虑另外一个优化问题：

求最小值 $f(x)$

满足条件 $g(x) \le 0$

和一个无约束优化问题：

求最小值 $f(x) + \mu \max\{0, g(x)\}$

满足条件 $x \in E^n$

易知，若$g(x) \le 0$，则 $\max\{0, g(x)\}=0$，此时不会产生惩罚项。而另一方面，若 $g(x) > 0$，则 $\max\{0, g(x)\} > 0$，就会产生惩罚项 $\mu g(x)$。

一般来讲，合理的罚函数在不可行解处应产生正的惩罚项，在可行解处不应产生惩罚项。

3. 带约束算法 - 罚函数方法（续二）

现考虑如下问题

原始问题：

$$\begin{aligned} &\text{求最小值} \quad f(x) \\ &\text{满足条件} \quad g_i(x) \le 0,\ i = 1, 2, \ldots, m; \\ &\qquad\qquad\ \ h_j(x) = 0,\ j = 1, 2, \ldots, k; \\ &\qquad\qquad\ \ x \in S \end{aligned}$$

罚函数法是求解如下**罚问题**：

$$\begin{aligned} &\text{求最大值} \quad \theta(\mu) = \inf \{ f(x) + \mu \phi(x) \mid x \in S \} \\ &\text{满足条件} \quad \mu \ge 0 \end{aligned}$$

其中，$\phi(x)$ 是一个连续罚函数。通过计算μ充分大时的 $\theta(\mu)$，我们可以以任意的精度计算出原始问题最优值的近似值。

3. 带约束算法 - 罚函数方法（续三）

引理 9. 设对每一个 i 和 j，$f(x)$，$g_i(x)$ 和 $h_j(x)$ 都是 E^n 上的连续函数，$\phi(x)$ 是一个连续罚函数

$$\phi(x)=\sum_{i=1}^{m}[\max\{0, g_i(x)\}]^p + \sum_{j=1}^{l}|h_i(x)|^p$$

另设对每一个 μ，都存在一个 $x_\mu \in S$ 使得 $\theta(\mu)= f(x_\mu)+\mu\,\phi(x_\mu)$。

则 (i) $\inf\{ f(x) \mid g(x) \le 0, h(x)=0, x\in S \} \ge \sup_{\mu\ge 0}\theta(\mu)$，

其中 $\theta(\mu) = \inf\{ f(x) + \mu\phi(x) \mid x\in S \}$。

(ii) $f(x_\mu)$ 是一个非降函数，$\mu \ge 0$，$\theta(\mu)$ 是一个非降函数

而 $\phi(x)$ 是一个非增函数。

证明: (i) 注意，当 $x\in S$ 且满足 $g(x) \le 0, h(x)=0$ 时，有 $\phi(x)=0$

则 $f(x)=f(x) + \mu\,\phi(x) \ge \inf\{ f(y) + \mu\,\phi(y) \mid y\in S \}= \theta(\mu)$。

3. 带约束算法 - 罚函数方法（续四）

证明: (ii) 设 $\lambda < \mu$。由 $\theta(\mu)$ 的定义可得

$$f(x_\mu) + \lambda\phi(x_\mu) \geq f(x_\lambda) + \lambda\phi(x_\lambda) = \theta(\lambda)。$$

$$f(x_\lambda) + \mu\phi(x_\lambda) \geq f(x_\mu) + \mu\phi(x_\mu) = \theta(\mu)。$$

将上两式相加，即得

$$(\mu - \lambda)[\phi(x_\lambda) - \phi(x_\mu)] \geq 0。$$

又因为 $\lambda < \mu$， $\phi(x_\lambda) \geq \phi(x_\mu)$，因此由第一个不等式可知，当 $\lambda > 0$ 时，有 $f(x_\mu) \geq f(x_\lambda)$。现将第一个不等式左侧，同时加、减 $\mu\,\phi(x_\mu)$ 可得

$$f(x_\mu) + \mu\,\phi(x_\mu) + (\lambda - \mu)\,\phi(x_\mu) = \theta(\mu) + (\lambda - \mu)\,\phi(x_\mu) \geq \theta(\lambda)。$$

又因为 $\lambda < \mu$ 和 $\phi(x_\mu) \geq 0$， 可得 $\theta(\mu) \geq \theta(\lambda)$。

3. 带约束算法 - 罚函数方法（续五）

定理 39. 设对每一个 i 和 j，$f(x)$，$g_i(x)$ 和 $h_j(x)$ 都是 E^n 上的连续函数。另设原始问题有一个可行解，$\phi(x)$ 是一个连续罚函数。若对每一个 μ，问题 $\min\{ f(x) + \mu\,\phi(x) \mid x \in S\}$ 都存在一个解 $x_\mu \in S$，且序列 $\{x_\mu\}$ 含于 S 的一个紧子集中。则

$$\inf\{ f(x) \mid g(x) \le 0, h(x)=0, x \in S \} = \sup_{\mu \ge 0} \theta(\mu) = \lim_{\mu \to \infty} \theta(\mu)$$

其中 $\theta(\mu) = \inf\{ f(x) + \mu\,\phi(x) \mid x \in S \} = f(x_\mu) + \mu\,\phi(x_\mu)$。而且，序列 $\{x_\mu\}$ 的任意一个聚点都是原始问题的一个最优解，且当 $\mu \to \infty$ 时，$\mu\,\phi(x_\mu) \to 0$。

证明: 根据**引理 9** 可知，$\theta(\mu)$ 是单调函数，因而有

$$\sup \theta(\mu) = \lim \theta(\mu)。$$

3. 带约束算法 - 罚函数方法（续六）

证明 (继续). 我们首先证明，当$\mu \to \infty$ 时， $\phi(x_\mu) \to 0$。设 y 为一个可行解， x_1 是问题 $\theta(1)$ 的最优解。由 **引理 9 可知，**

若 $\mu \geq 2+|f(y)-f(x)|/\varepsilon$，则 $f(x_\mu) \geq f(x_1)$。现证明 $\phi(x_\mu) \leq \varepsilon$。用反证法，假设$\phi(x_\mu) > \varepsilon$。仍由 **引理 9** 可得

$$\inf\{f(x) \mid g(x)\leq 0, h(x)=0, x\in S\} \geq \theta(\mu)=f(x_\mu)+\mu\phi(x_\mu) \geq f(x_1)+\mu\phi(x_\mu)$$
$$\geq f(x_1)+|f(y)-f(x)|+2\varepsilon > f(y),$$

这是不可能的因为 y 是一个可行解。又因ε 可取任意值，故当 $\mu \to \infty$ 时， $\phi(x_\mu) \to 0$。 现设 $\{x_{\mu_k}\} \subseteq \{x_\mu\} \to x^*$。则有

$$\sup\theta(\mu) \geq \theta(\mu_k) = f(x_{\mu_k}) + \mu_k\phi(x_{\mu_k}) \geq f(x_{\mu_k})。$$

由此可得 $\sup\theta(\mu) \geq f(x^*)$。再因当$\mu \to \infty$ 时， $\phi(x_\mu) \to 0$， 故 x^* 是一个可行解，由**定理 9**(i)可知，它也是最优解。因此可得 $\sup\theta(\mu) = f(x^*)$。而且有 $\mu\,\phi(x_\mu) = \theta(\mu) - f(x_\mu)$。当$\mu \to \infty$ 时, $\theta(\mu)$ 和 $f(x_\mu)$ 都趋于 $f(x^*)$。 故 $\mu\,\phi(x_\mu)$ 趋于**0**。

3. 带约束算法 - 罚函数方法（续七）

由上述定理可知，我们只要选取足够大的 μ，就可以让问题 $\min\{ f(x) + \mu\phi(x) \mid x \in S\}$ 的最优解 x_μ 任意接近可行区域。而且，选取足够大的 μ，可使得 $f(x_\mu) + \mu\,\phi(x_\mu)$ 任意接近原始问题的目标函数的最优值。

求解罚问题的一个常用方案是，求解一系列如下形式的问题：

求最小值 $f(x) + \mu\,\phi(x)$
满足条件 $x \in S$

其中选取 μ 为单调增的罚参数。 一般来讲，最优解序列 $\{x_\mu\}$ 是不可行解，然而当罚参数变大时，序列中的解从可行区域外面接近最优解。因而，这个方案称为**外罚函数法**。

3. 带约束算法 - 罚函数方法（续八）

步 0. 设 $\varepsilon > 0$ 为终止参数。选取初始解 x_1 和罚参数 $\mu_1 > 0$，及 $\beta > 1$。置 $k:=1$，转到 **步 1**。

步 1. 求问题 $\min\{ f(x) + \mu_k \phi(x) \mid x \in S \}$ 的最优解， 并置为 x_{k+1}。

步 2. 若 $\mu_k \phi(x_{k+1}) < \varepsilon$，则终止算法；

否则置 $\mu_{k+1}:=\beta \mu_k$，$k:= k+1$，返回到 **步 1**。

除了连续性，上述算法并不需要问题中涉及的函数，还有什么其他性质。然而，这并不意味着它可以有效地处理任意带约束的优化问题。实际上，它只能用于求解这样的带约束优化问题：存在有效求解**步1**中最小值问题的算法。

3. 带约束算法 - 罚函数方法（续九）

正如我们前面提到的，选取充分大的 μ，就可以使得罚问题的解充分接近原问题的最优解。然而，当选取的μ 非常大，且要求解相应的罚问题时，就会遇到病态所产生的计算困难。当μ 很大时，更加强调解的可行，大多数求解无约束优化问题的方法可很快趋于一个可行解。尽管这个可行解可能与最优解相距甚远，但是算法可能会提前终止。

为了说明这一点，假设在优化过程中我们找到了一个可行解，它满足 $\phi(x) = 0$。特别是处理非线性等式约束时， 从点 x 沿这任意一个方向 d 移动，都有可能导致走到一个非可行解或者一个具有很大目标函数值的可行解。

3. 带约束算法 - 障碍函数方法

与罚函数类似，**障碍函数**也可以用来将一个约束优化问题转化为一个无约束优化问题，或者一系列无约束优化问题。 这些函数在可行区域设立了一个障碍，阻止可行解离开可行区域。如果最优解在可行区域的边界，那么求解过程从可行区域的内部点移向可行区域的边界。

障碍函数是一个定义于区域 $\{x \mid g_i(x) < 0$，每一个 $i\}$ 上的非负连续函数，且当从区域 $\{x \mid g_i(x) \leq 0,$ 每一个 $i\}$ 的内部接近其边界时，函数值就会趋于无穷。如下就是一个典型的障碍函数的例子：

$$B(x) = \sum_{i=1}^{m} \frac{-1}{g_i(x)}$$

3. 带约束算法 - 障碍函数方法（续一）

直观上讲，我们希望障碍函数 $B(x)$ 在区域

$$\{x \mid g_i(x) < 0，\text{每一个 } i\}$$

上取值为 0，而在区域的边界上取值 ∞。这样就能保证，只要我们从求最小值问题的一个内部点开始搜索，最后就不会离开区域 $\{x \mid g_i(x) \le 0$，每一个 $i\}$。

原始问题： 求最小值 $f(x)$

满足约束 $g(x) \le 0$,

$x \in S$

障碍问题： 求最小值 $\theta(\mu) = \inf \{ f(x) + \mu B(x) \mid g(x) < 0, x \in S\}$

满足约束 $\mu \ge 0$

3. 带约束算法 - 障碍函数方法（续二）

定理 40. 设 $f(x)$ 和每一个 $g_i(x)$ 都是 E^n 上的连续函数。另设 S 为 E^n 上的一个非空闭集。假设集合 $\{x \mid g(x) < 0\}$ 非空，原始问题有最优解 x^* 满足如下性质：给定 x^* 的任意一个邻域 $N(x)$，存在一个 $x \in N(x) \cap S$ 使得 $g(x) < 0$。则

$$\text{Min}\{ f(x) \mid g(x) \le 0, x \in S \} = \lim_{\mu \to 0+} \theta(\mu) = \inf_{\mu > 0} \theta(\mu)$$

设 $\theta(\mu) = f(x_\mu) + \mu B(x_\mu)$，其中 $x_\mu \in S$ 且 $g(x_\mu) < 0$，$\{x_\mu\}$ 的任意一个聚点都是原始问题的最优解，且当 $\mu \to 0$ 时，$\mu B(x_\mu) \to 0$。

对每一个 μ，算法生成的序列 $\{x_\mu\}$ 属于区域 $\{x \mid g(x) \le 0\}$ 的内部。因此这个算法称为**内罚函数法**。

3. 带约束算法 - 障碍函数方法（续三）

步 0. 设$\varepsilon > 0$ 是终止参数。选取初始解 x_1 满足 $g(x_1) < 0$。
取罚参数 $\mu_1 > 0$，和参数 $\beta \in (0, 1)$。
置 $k := 1$，转到 **步 1**。

步 1. 求最小值问题 $\min\{ f(x)+\mu_k B(x) \mid g(x)<0, x \in S\}$ 的最优解，
并置 x_{k+1} 为最优解。

步 2. 若 $\mu_k B(x_{k+1}) < \varepsilon$，则终止；
否则置 $\mu_{k+1} := \beta \mu_k$，$k := k+1$，然后转到 **步 1**。

由于大多数的线搜索方法都采用离散步长，当我们接近区域的边界时，再走一个步长就可能走到可行区域的外部了，此处罚函数 $B(x)$ 的值是一个非常大的负数。因此，只有当可行性很容易检查时，才可能将此问题当作无约束优化问题来处理。

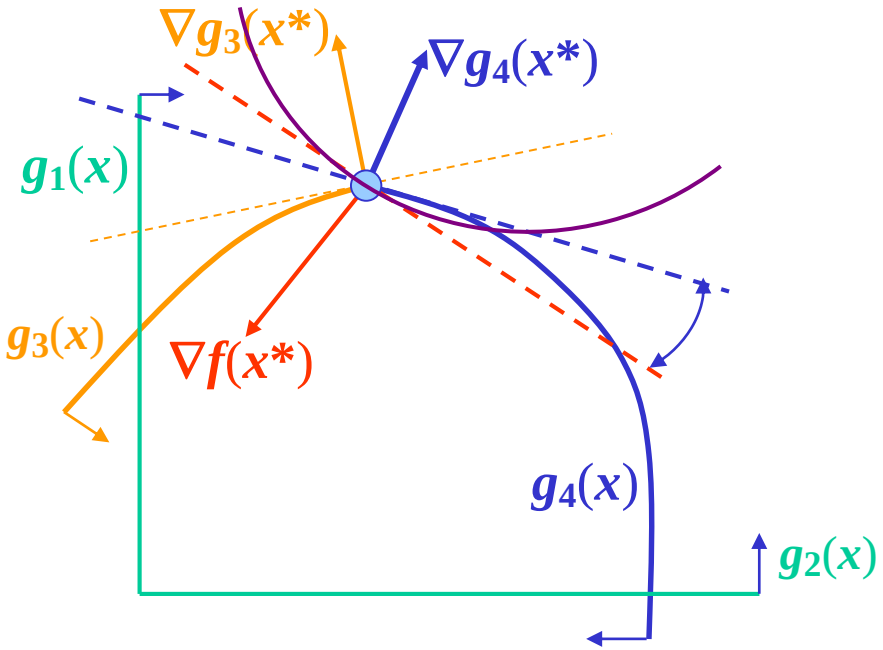
3. 带约束算法 - 可行方向法

求解带约束的非线性规划问题的**可行方向法**从一个可行解移动到一个更好的可行解。这类方法通常采用如下策略：给定一个可行解 x_k，确定一个方向 d_k，使得对足够小的 $\lambda > 0$，$x_k + \lambda d_k$ 是一个可行解，且 $f(x_k + \lambda d_k) < f(x_k)$。

当一个可行的改进方向 d_k 确定以后，需要求解一个一维优化问题，以决定沿着方向 d_k 走多远。由此可以求得一个新的解 x_{k+1}，然后算法重复此过程。注意，这个优化过程保持了原始可行性，因此称其为**原始方法**。

3. 带约束算法 - 可行方向法（续一）

定理 41.考虑求最小值问题 $\min\{f(x) \mid g(x) \le 0\}$。设 x 是一个可行解，$I=\{i \mid g_i(x)=0\}$是积极约束的指标集。另设 $f(x)$ 和$g_i(x)$，$i \in I$，在点 x 处可微，且 $g_i(x)$，$i \notin I$，在点 x 处连续。 若 $\nabla f(x)d < 0$，且$\nabla g_i(x)d < 0$，$i \in I$，则 d 是一个可行的改进方向。



目标函数的等高线

可行的改进方向

3. 带约束算法 - 可行方向法（续二）

定理 42. 考虑求最小值问题 $\min\{f(x) \mid g(x) \le 0\}$。设 x 是一个可行解，$I = \{i \mid g_i(x) = 0\}$。考虑如下**求搜索方向问题**：

求最小值 z

满足条件 $\nabla f(x)\, d \le z,$

$\nabla g_i(x)\, d \le z,$ 每一个 $i \in I,$

$-1 \le d_j \le 1,$ 每一个 $j = 1, 2, \ldots, n.$

则 x 是一个**Fritz John 点**当且仅当上述问题的最优值等于**0**。

证明. (**练习***) 目标函数的最优值等于**0**当且仅当 线性系统

$\nabla f(x) d < 0$ 和 $\nabla g_i(x)\, d < 0$，每个 $i \in I$,

无解。依据**Gordan 定理**， …… (参见第47页的**练习**)

3. 带约束算法 - 可行方向法（续三）

Zoutnedijk 算法 [1960]

步 0. 选取一个初始可行解 x_1 满足 $g(x_1) \le 0$。

置 $k:=1$，转到**步 1**。

步 1. 求出**定理39** 中求搜索方向问题的最优解，

置 (z_k, d_k) 为最优解。

若 $z_k = 0$，则终止；x_k 是一个**Fritz John 点**；

步 2. 求解下述线搜索问题

求最小值 $f(x_k + \lambda d_k)$

满足条件 $0 \le \lambda \le \lambda_{\max} \equiv \sup\{\lambda \mid g_i(x_k + \lambda d_k) \le 0,$ 每个 $i\}$

置 λ_k 为最优解。

步 3. 置 $x_{k+1} := x_k + \lambda_k d_k$，$k := k+1$，返回 **步 1**。

不过上述算法生成的点列不一定收敛到一个 **K-K-T 点**。这是因为，沿着搜索方向移动的步长有可能趋于**0**，从而在非最优点造成**堵塞现象**。

3. 带约束算法 - 可行方向法（续四）

Topkis 和**Veinott** [1967] 提出了一个修正的**Zoutnedijk算法**。在**求搜索方向问题**中，他们采用如下约束条件

$$\nabla g_i(x)\, d < z - g_i(x) \quad \text{所有 } i,$$

而不是 $\nabla g_i(x)\, d < z$ 每个 $i \in I$.

注意，修正前后的区别是：积极约束和非积极约束在确定搜索方向的时候都起作用。因此，在趋于当前非积极约束的边界时，不会发生方向突然改变的情况。

定理 43. 考虑求最小值问题 $\min\{ f(x) \mid g(x) \le 0\}$。设 $f(x)$ 和 $g_i(x)$ 都是连续可微的函数，且上述算法生成的序列为 $\{x_k\}$。则$\{x_k\}$ 的任意一个聚点都是**Fritz John 点**。

证明: (练习*) 类似**定理 42** 的证明。

3. 带约束算法 - 梯度投影法

回忆一下，最速下降方向是负梯度方向。当处理带约束的优化问题时， 沿着此方向进行搜索可能会找到一个不可行的解。**Rosen** [1960] 提出了一个梯度投影算法，它将负梯度做一个投影使得目标函数减少的同时还可以保证得到解是可行的。称一个($n\times n$)-矩阵P 是**投影矩阵**如果$P = P^T$ 且 $PP = P$。

引理 10. 设 P 是一个 ($n\times n$)-矩阵，则有

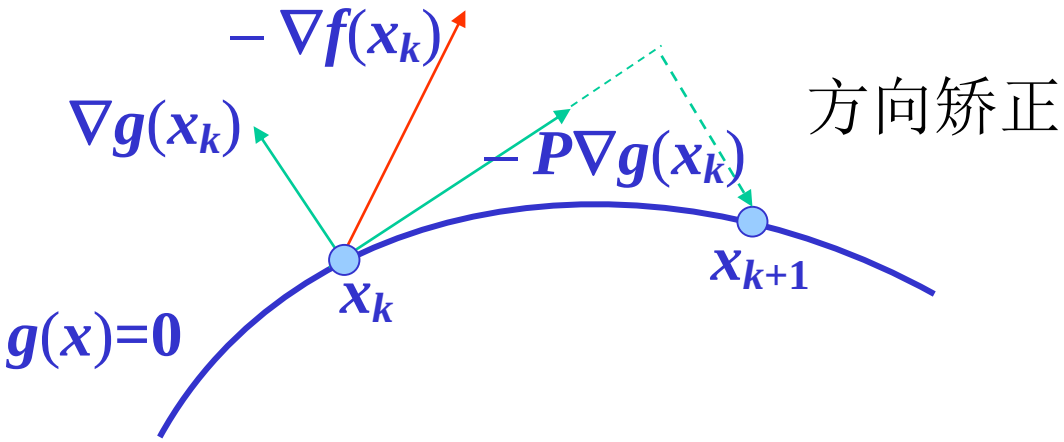
(1) 若 P 是一个投影矩阵，则 P 是半正定矩阵。

(2) P 是一个投影矩阵当且仅当 $I - P$ 是投影矩阵。

(3) 设 P 是一个投影矩阵，$Q = I - P$。则 $L=\{Px \mid x\in E^n\}$ 和 $L^{\perp}=\{Qx \mid x\in E^n\}$ 是正交线性子空间。而且，任意一个 $x\in E^n$ 都可以惟一表示为 $p + q$，其中 $p\in L$， $q\in L^{\perp}$。

3. 带约束算法 - 梯度投影法（续一）

Rosen算法将目标函数的梯度投影到（若干）积极约束的梯度构成的**零空间**上。然而，沿梯度的投影方向搜索常常不能得到可行解（除非我们仅考虑线性约束），因为它是可行区域的**切向量**。 因此，沿着梯度的投影方向搜索时，还需要做一些矫正，使得搜索方向指向可行区域。



3. 带约束算法 - 梯度投影法（续二）

Rosen 算法用于求解如下优化问题：

求最小值 $f(x)$

满足条件 $g_i(x) \le 0,\ i = 1,\ldots, m;\ h_j(x) = 0,\ j = 1,\ldots, k;$

设 x_k 是一个可行解，$I = \{i \mid g_i(x_k) = 0\}$。$M$ 是一个矩阵，其行为 $\nabla g_i(x_k)$，$i \in I$ 和 $\nabla h_j(x_k)$，$j = 1,\ldots, k$，且 $P = I - M^T(MM^T)^{-1}M$。注意，P 将一个向量投影到等式约束函数的梯度和积极的不等式约束函数的梯度构成的零空间上。

设 $d_k = -P\nabla f(x_k)$。若 $d_k \ne 0$，则可由 x_k 出发沿着方向 d_k，并向可行区域做适当矫正，极小化函数 $f(x)$。若 $d_k = 0$，计算 $(u, v) = -\nabla f(x_k)M^T(MM^T)^{-1}$。当 $u \ge 0$ 时，算法终止，并得到一个 K-K-T 解 x_k；否则去掉矩阵 M 中的那些行，其相应的 $u_i < 0$，重复此过程。

3. 非线性规划 - 算法的比较

普适性： 人们为不同类型的非线性规划问题设计了不同的求解算法，比如无约束优化问题，带不等式约束的优化问题，带等式约束的优化问题，及带这两种约束的优化问题。对每类优化问题，不同的求解算法会对问题的结构做不同的具体假设。

精度： 注意，求解非线性规划算法的收敛性是在极限意义下的。因此好的算法应很快地找到一个能达到满意目标函数值的可行解。不好的算法往往会生成一系列不可行的解，而只是在算法终止时达到可行性解。

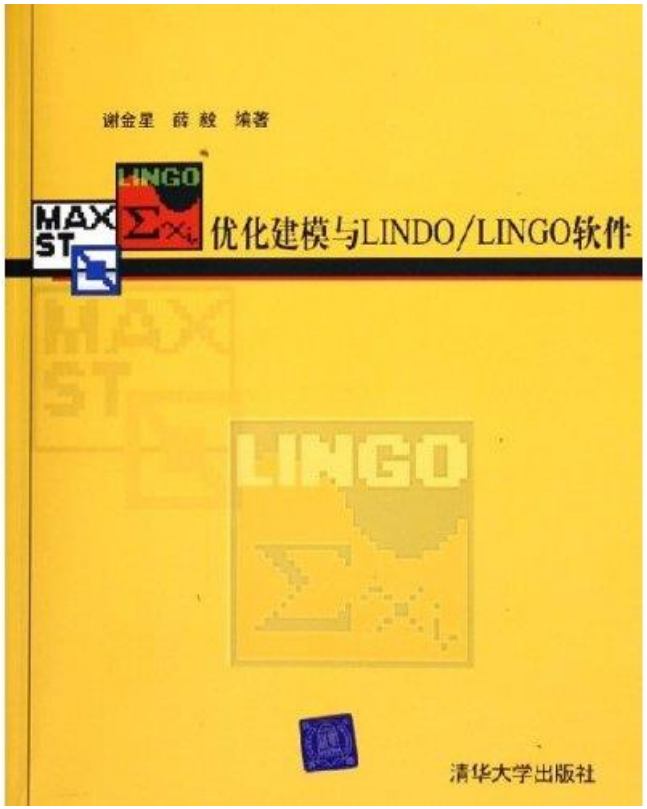
灵敏度： 在运行大多数算法时，首先需要给一些参数设定初始值，比如初始解、步长、算法终止条件。有些算法对这些参数和问题的输入值相当敏感，因此会因这些参数和值的不同，而生成不同的解或者提前终止运行。对一组固定的参数，好的算法应该对问题的范围很广的输入都是有效的。

3. 非线性规划 - 算法的比较（续一）

可靠性/鲁棒性：给定任意一个具体的算法，要找到一个问题，使得这个算法无法求出其有效解通常并不是很困难的。所以在这里，我们是指算法可以在满足一定的精度要求下，求出大多数问题的有效解；而不能忽略 一个具体算法的可靠性与问题的规模及结构之间的关系。例如，有些算法在求解规模较小的优化问题时是可靠的，然而，它们在求解大规模的优化问题是，就有可能变得不可靠了。

预处理和计算成本：有些算法需要计算一阶或二阶导数，如果原函数相当复杂，那么这些算法的计算成本就要比只需要计算函数值的算法的成本高得多。人们常用计算时间，迭代次数，或者函数值的计算量，来度量计算成本。不过，仅用单单一个量来评判一个算法的好坏并不令人满意。

3. 非线性规划 - 算法的比较（续二）



ILOG

Changing the rules of business

World's Leading Mathematical Programming Optimizers

ILOG IS NOW IBM

GUROBI OPTIMIZATION

3. 非线性规划 - 算法的比较（续三）

is a commercial optimization solver for LP, QP, QCP, MILP, MIQP, and MIQCP.

https://www.gurobi.com/