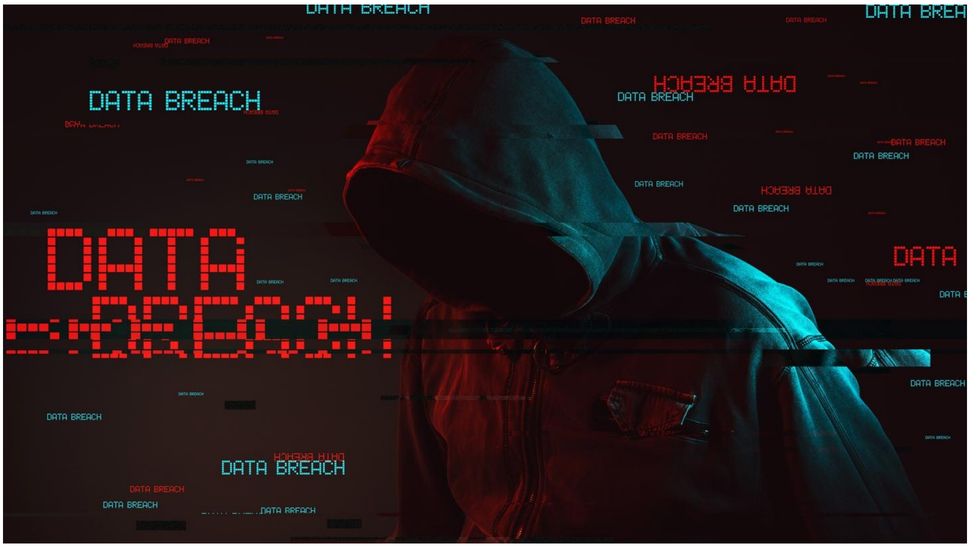

最优化抄书笔记：序列二次规划

吟雪千夏
USTC数学博士在读 高精度CFD算法

创作声明: 内容包含医疗建议

还是那句老话：七夕只有一天，而单身是每一天，不如在家卷。

Reference：

[1] 李董辉，童小娇，万中，数值最优化算法与理论.

[2] 马昌凤，最优化方法及其 Matlab 程序设计.

前言：本篇笔记主要介绍序列二次规划(SQP)方法。SQP的全称是Sequential Quadratic Programming，即通过解“Quadratic Programming”来生成一个序列(Sequence)。

一、Newton-Lagrange方法

我们第一部分的目标是学习序列二次规划算法(SQP)。作为铺垫，在此之前我们需要先简单了解Newton-Lagrange方法。

先考虑只有等式约束的优化问题：

$$\begin{aligned}&\min f(x)\\&s.t.\ h_j(x)=0,\ \forall j\in J\end{aligned}\qquad\cdots(1)$$

记 $h(x)=(h_1(x),\cdots,h_l(x))^T$ ，则 (1) 的Lagrange函数为

$$L(x,\mu)=f(x)-\mu^Th(x)=f(x)-\sum_{j\in J}\mu_jh_j(x)$$

根据KT条件，最优解满足如下的方程组：

$$\begin{bmatrix} \nabla_x L(x,\mu) \\ h(x) \end{bmatrix} = \begin{bmatrix} \nabla f(x) - \nabla h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

其中 h 的梯度 $\nabla h(x) = (\nabla h_1(x), \cdots, \nabla h_l(x))$。因此，求解优化问题 (1) 相当于解上面的这个非线性方程组，我们可以用Newton迭代法进行求解：记

$F(x,\mu) := \begin{bmatrix} \nabla f(x) - \nabla h(x)^T \mu \\ h(x) \end{bmatrix}$ ，则它的Jacobi阵为

$$J_F(x,\mu) := \begin{bmatrix} \nabla^2 f(x) - \nabla^2 h(x)^T \mu & -\nabla h(x)^T \\ -\nabla h(x) & 0 \end{bmatrix} = \begin{bmatrix} \nabla_x^2 L(x,\mu) & -\nabla h(x)^T \\ -\nabla h(x) & 0 \end{bmatrix}$$

回顾：求一元非线性函数的根的Newton迭代法为

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

(推导可以参考 吟雪千夏：数值分析笔记Chapter1 方程求解)

在这里，我们不能直接把相当于 $f'(x_k)$ 地位的 J_F 除下去。但对应地，我们可以求它的逆：

$$\begin{pmatrix} x_{k+1} \\ \mu_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \mu_k \end{pmatrix} - J_F{}^{-1}(x_k,\mu_k) F(x_k,\mu_k)$$

这里注意： $A^{-1}B$ 实际上是 $Ax = B$ 的解，因此我们不用真的求逆，而只需要解

$$J_F(x_k,\mu_k)\, d_k = F(x_k,\mu_k)$$

即

$$\begin{bmatrix} \nabla_x^2 L(x_k,\mu_k) & -\nabla h(x_k)^T \\ -\nabla h(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^\mu \end{bmatrix} = \begin{bmatrix} \nabla f(x_k) - \nabla h(x_k)^T \mu_k \\ h(x_k) \end{bmatrix}$$

$$= \begin{bmatrix} \nabla_x L(x_k,\mu_k) \\ h(x_k) \end{bmatrix} \quad \cdots (2)$$

解出的这个 d_k 就是增量。所以我们的格式就是：

$$\begin{pmatrix} x_{k+1} \\ \mu_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \mu_k \end{pmatrix} - d_k$$

由于这个方法实际上是对Lagrange函数使用了Newton迭代法，故称之为Lagrange-Newton方法。因为Newton方法是二阶收敛的，故当 x^* 的Hesse阵正定(满足二阶充分条件)时，我们也得到Lagrange-Newton方法的二阶收敛性。

注意：Newton-Lagrange方法的初值无需满足 $x_0 \in D$ (可行域)。

二、局部SQP

目前来看，上一节里讨论的Lagrange-Newton方法还不适合求解含有不等式约束的问题。为了让这种方法能够求解一般的约束优化问题，我们可以将Newton迭代步换一种形式来写。注意到，解 (2) 相当于求约束优化子问题

$$
\begin{aligned}
&\min q_k(d)=\frac{1}{2}d^T\nabla_x^2L(x_k,\mu_k)d+\nabla_xL(x_k,\mu_k)^Td\\
&s.t.\ \nabla h(x_k)d+h(x_k)=0
\end{aligned}
$$

的解(也即KT点 (d_k,μ_{k+1}))，这是因为上述问题的KT条件完全就是 (2) 。

再利用约束条件(蓝色部分)，注意到对于问题的可行点，有

$$
\nabla_xL(x_k,\mu_k)^Td=\nabla f(x_k)^Td-{\mu_k}^T\nabla h(x_k)d=\nabla f(x_k)^Td+{\mu_k}^Th(x_k)
$$

而量 ${\mu_k}^Th(x_k)$ 是已经确定的量，因此上述问题可以转化为求

$$
\begin{aligned}
&\min q_k(d)=\frac{1}{2}d^T\nabla_x^2L(x_k,\mu_k)d+\nabla f(x_k)^Td\\
&s.t.\ \nabla h(x_k)d+h(x_k)=0
\end{aligned}
\quad\cdots(3)
$$

的KT点。

(我们暂时不考虑 (3) 的求解，具体的求解将在下一小节介绍)

由此可发现，如果是考虑一般的约束优化问题

$$
\begin{aligned}
&\min f(x)\\
&s.t.\begin{cases} g_i(x)\geq 0,\forall i\in I\\ h_j(x)=0,\forall j\in J\end{cases}
\end{aligned}
\quad\cdots(1')
$$

定义 $g(x)=(g_1(x),\cdots,g_{l'}(x))^T$ ，那么每个Newton迭代步可转化为求

$$
\begin{aligned}
&\min q_k(d)=\frac{1}{2}d^T\nabla_x^2L(x_k,\lambda_k,\mu_k)d+\nabla f(x_k)^Td\\
&s.t.\begin{cases}\nabla g(x_k)d+g(x_k)\geq 0\\ \nabla h(x_k)d+h(x_k)=0\end{cases}
\end{aligned}
$$

的KT点 $(d_k,\lambda_{k+1},\mu_{k+1})$ 。

这是合理的，回顾：**KT条件对 g 的要求是 $\lambda_Ig_I(x)=0$** 。

然而， $\nabla_x^2L(x_k,\lambda_k,\mu_k)$ 可能不是正定的。对此，我们可以用和拟Newton法一样的思路：用一个对称正定的矩阵 B_k 去近似 $\nabla_x^2L(x_k,\lambda_k,\mu_k)$ ，具体的做法我们将在下一小节介绍。

这样，上述问题就变成了求

$$
\begin{aligned}
&\min q_k(d)=\frac{1}{2}d^TB_kd+\nabla f(x_k)^Td\\
&s.t.\begin{cases}\nabla g(x_k)d+g(x_k)\geq 0\\ \nabla h(x_k)d+h(x_k)=0\end{cases}
\end{aligned}
\quad\cdots(3')
$$

的KT点。

现在我们把完整的算法总结如下：

Algorithm 1 局部SQP算法

Step 1：给定初始点 (x_0,λ_0,μ_0) ， $k=0$ ，初始对称正定阵 $B_0=I$ ；

Step 2：若 $\|\nabla_xL(x_k,\lambda_k,\mu_k)\|<\varepsilon$ ，则停止，输出 x_k ；否则转Step 3；

Step 3：求解子问题 $(3')$ 得 $(d_k, \lambda_{k+1}, \mu_{k+1})$ ，令 $x_{k+1} = x_k + d_k$ ；

Step 4：用适当的拟Newton法修正，得到 B_{k+1} ；

Step 5：更新 $k = k + 1$ ，转Step 2，反复迭代直到输出。

可以证明在一定条件下，局部SQP算法可以达到**超线性收敛**。

接下来还需讨论三个问题：

1. B_{k+1} 的更新；
2. 子问题的初始可行点的确定；
3. 子问题的求解。

三、截断BFGS

这部分讲一种 B_{k+1} 的修正方法。

回顾：Newton修正的目的是由初始的对称正定阵 B_0 ，产生矩阵序列 $\{B_k\}$ 满足：

1. B_k 对称正定；
2. B_k 是 $\nabla^2 f(x_k)$ 的某种近似；
3. B_k 容易计算。

令 $s_k = x_{k+1} - x_k, y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ ，则 B_{k+1} 应满足：

$y_k = B_{k+1} s_k$

由此，我们推导了秩1修正公式：

$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$

现在考虑秩2修正，即：令 $B_{k+1} = B_k + \alpha u_k {u_k}^T + \beta v_k {v_k}^T$ ，经过一番**我不怎么想看**的推导，可以得到秩2修正公式：

$B_{k+1} = B_k - \frac{B_k s_k {s_k}^T B_k}{{s_k}^T B_k s_k} + \frac{y_k {y_k}^T}{{y_k}^T s_k}$

这是很流行的拟Newton修正公式，称为**BFGS公式**。

可以证明：若 B_k 对称正定且 ${y_k}^T s_k > 0$ ，则 B_{k+1} 也对称正定。

还可以证明：如果配合精确线搜索或Wolfe线搜索，则 ${y_k}^T s_k > 0$ 。如果使用Armijo线搜索，则修正公式需要作一定的改动。

但这些都不是本篇笔记的重点，我们只是使用这个算法来得到 B_{k+1} 。

现在回到子问题 $(3')$ ，如果我们令

$$s_k = x_{k+1} - x_k \text{，} \quad y_k = \nabla_x L(x_{k+1}, \lambda_{k+1}, \mu_{k+1}) - \nabla_x L(x_k, \lambda_k, \mu_k)$$

则可以完全一样地定义 B_{k+1} 的BFGS修正。但这时，我们无法保证 B_{k+1} 是正定的。

为了克服这个困难，Powell提出了一种方法，用 y_k 和 $B_k s_k$ 的一个凸组合 $\bar{y}_k$ 来代替 y_k ：

$$\bar{y}_k := \begin{cases} y_k, & {y_k}^T s_k \geq 0.2 {s_k}^T B_k s_k \\ \theta y_k + (1-\theta) B_k s_k, & \text{otherwise} \end{cases}$$

其中

$$\theta = \frac{0.8 {s_k}^T B_k s_k}{{s_k}^T B_k s_k - {y_k}^T s_k}$$

然后令

$$B_{k+1} = B_k - \frac{B_k s_k {s_k}^T B_k}{{s_k}^T B_k s_k} + \frac{\bar{y}_k {\bar{y}_k}^T}{{\bar{y}_k}^T s_k}$$

这种修正方法称为**截断BFGS修正**。

四、QP子问题的可行点

本部分讲如何来找到 $(3')$ 的一个初始可行点，这也是Powell提出的一个方法。引进辅助变量 ξ ，首先解下面的线性规划子问题：

$$\max \xi$$

$$s.t. \begin{cases} \nabla g_i(x_k)^T d + \xi g_i(x_k) \geq 0, & i \in V_k := \{i : g_i(x_k) < 0\} \\ \nabla g_i(x_k)^T d + g_i(x_k) \geq 0, & i \in S_k := \{i : g_i(x_k) \geq 0\} \\ \nabla h_j(x_k)^T d + \xi h_j(x_k) = 0, & j \in J \\ 0 \leq \xi \leq 1 \end{cases} \quad \cdots (4)$$

这个线性规划问题的可行域和原可行域的区别是加了深红色的两个部分。显然，如果问题的解 $\xi^* = 1$ ，则说明 $(3')$ 的可行域非空：这是因为存在 d 使得

$$\begin{cases} \nabla g_i(x_k)^T d + 1 g_i(x_k) \geq 0, & g_i(x_k) < 0 \\ \nabla g_i(x_k)^T d + g_i(x_k) \geq 0, & g_i(x_k) \geq 0 \\ \nabla h_j(x_k)^T d + 1 h_j(x_k) = 0, & j \in J \end{cases}$$

成立，这也就是 $(3')$ 的约束条件。另一方面， (4) 一定有解，这是因为 $\xi = 0, d = 0$ 至少是问题的解。但这时，如果 $d = 0$ 是 (4) 的唯一解，那么 $x_{k+1} = x_k + d_k = x_k$ ，算法就进入了死循环(新的子问题还是和这次一样的，依然会解出 $d = 0$)。因此，当问题的解 $\xi^* = 0$ 时，需要重新选择初始点，这是美中不足的地方。而如果 $\xi^* > 0$ ，那么我们就**用 (4) 中关于 d 的约束条件替换 $(3')$ 中的约束条件，并以 (4) 的最优解 d^* 作为初始点**，解如下的新二次规划问题(至少这样算法可以动一动)：

$$\min q_k(d)=\frac{1}{2}d^TB_kd+\nabla f(x_k)^Td$$
$$s.t.\begin{cases}\nabla g_i(x_k)^Td+\xi^*g_i(x_k)\geq 0, & g_i(x_k)<0\\ \nabla g_i(x_k)^Td+g_i(x_k)\geq 0, & g_i(x_k)\geq 0\\ \nabla h_j(x_k)^Td+\xi^*h_j(x_k)=0, & j\in J\end{cases}$$

它可以重新写成：

$$\min q(x)=\frac{1}{2}x^TQx+q^Tx$$
$$s.t.\begin{cases}{a_i}^Tx+b_i\geq 0, & i\in I\\ {a_j}^Tx+b_j=0, & j\in J\end{cases}\qquad\cdots(3'')$$

五、QP子问题的求解：有效集法

在讲子问题 $(3'')$ 的求解之前，我们考虑只有等式约束的严格凸二次规划问题

$$\min q(x)=\frac{1}{2}x^TQx+q^Tx$$
$$s.t.\,{a_j}^Tx+b_j=0,\ j\in J\qquad\cdots(5)$$

即依然假设 Q 是对称正定的，并且 $A=(a_1,\cdots,a_j)$ 行满秩，即 (5) 的可行域不是空集。那么 x^* 是 (5) 的解当且仅当它是KT点，也就是

$$Qx^*-A^T\mu^*+q=0,\ Ax^*+b=0$$

可以写成线性方程组形式

$$\begin{pmatrix}Q & -A^T\\ -A & 0\end{pmatrix}\begin{pmatrix}x^*\\ \mu^*\end{pmatrix}=\begin{pmatrix}-q\\ b\end{pmatrix}$$

在 A 行满秩的条件下，可以证明上述线性方程组有唯一解。

下面考虑一般的二次规划 $(3'')$，记 $A(x):=\{i\in I,{a_i}^Tx+b_i=0\}\cup J$。因为 Q 对称正定，故 x^* 是 $(3'')$ 的最优解当且仅当 x^* 是KT点，即：

$$Qx^*-{A_I}^T\lambda^*-{A_J}^T\mu^*=0,A_Jx^*+b_J=0,\min\{\lambda^*,A_Ix^*+b_I\}=0$$

若 $i\notin A(x^*)$，则 $\lambda_i^*=0$，记 $A_I^*x+b_I^*=0$ 是那些 $i\in A(x^*)$ 中的约束，那么上式就成为

$$Qx^*-{A_I^*}^T\lambda^*-{A_J}^T\mu^*=0,\ A_Jx^*+b_J=0,\ A_I^*x^*+b_I^*=0$$

将 A_I^* 和 A_J 整合在一起可以看出，x^* 也是如下问题的KT点：

$$\min q(x)=\frac{1}{2}x^TQx+q^Tx$$
$$s.t.\,{a_i}^Tx+b_i=0,\ i\in A(x^*)\qquad\cdots(3''')$$

但由于我们并不事先知道 x^*，所以不能够直接求解 $(3''')$。这就产生了有效集法的基本思路：

首先确定初始点 $x_0\in D$（**注意：对于SQP，这个初始点是已经有的**），由此可确定其有效集 A_0，我们就以 A_0 代替 $(3''')$ 中的 $A(x^*)$，求解得到 x_1，再确定 A_1，一般地，在第 k 步，求解

$$\min q(x) = \frac{1}{2}x^T Qx + q^T x \qquad \cdots (3'''')$$
$$s.t.\, a_i{}^T x + b_i = 0,\ i \in A_k$$

这样一直迭代下去直到确定最优解 x^* 即可。

接下来需要研究几个问题：

1. 如何确认，某个 x_k 就是最优解呢？

2. 产生的新 x_k 是 $(3'''')$ 的最优解，但如果它不是原问题 $(3'')$ 的可行解怎么办？

3. 如何更新 A_{k+1} ？

对于第一个问题，下面的定理回答了我们：

Theorem 1

设在 $(3'''')$ 中用 A_k 代替 $A(x^*)$ 后得到的解为 x_k 。如果 x_k 对应的Lagrange乘子满足 $\lambda_k(i) \geq 0, \forall i \in A_k \cap I$ ，则 x_k 就是 $(3'')$ 的最优解。

Proof

可知 x_k 满足KT条件：

$$Qx_k + q - A_I^{k\,T}\lambda_k - A_J{}^T\mu_k = 0,$$

$$A_I^k x_k + b_I^k = 0, \lambda_k(i) \geq 0, \forall i \in I \cap A_k,$$

$$A_J x_k + b_J = 0.$$

构造原问题的Lagrange乘子 $\lambda_{k'}$ ，对于 $i \in I \backslash A_k$ ，令 $\lambda_{k'}(i) = 0$ 。则

$$Qx_k + q - A_I{}^T\lambda_{k'} - A_J{}^T\mu_k = 0,$$

$$\min\{A_I x_k + b_I, \lambda_{k'}\} = 0,$$

$$A_J x_k + b_J = 0.$$

这就是原问题 $(3'')$ 的KT条件。□

这里作一个小的讨论：为什么Lagrange乘子的 $\lambda_k(i)$ 可能小于零？这是因为，实际上，我们已经把 $(3'''')$ 当成了一个**等式约束优化**问题来看待了，所以原来在原问题的 I (**不等式约束**)里的那些 λ_i ，在这里实际上是 μ_i 的地位，在解KT条件的方程组的时候是不保证它们大于等于零的。

而这个定理告诉我们，如果解出来这些 λ_i 都非负，那么我们就已经解得了原问题的KT点。

对于第二个问题，如果问题的解 $\bar{x}_k$ 是原问题 $(3'')$ 的可行解，那么就令 $x_{k+1} = x_k$ ；而如果 $\bar{x}_k$ 不可行，那么我们需要把 x_{k+1} “拉回”到可行域范围里来。记 $d_k = \bar{x}_k - x_k$ ，令

$$\alpha_k = \min\left\{-\tfrac{a_i{}^T x_k + b_i}{a_i{}^T d_k} : i \notin A_k, a_i{}^T d_k < 0\right\}$$

则 α_k 是最大的使得 $x_k + \alpha d_k \in D$ 的 α 。我们就令 $x_{k+1} = x_k + \alpha_k d_k$ 。而如果 x_{k+1} 可行，那我们同样可以约定 $\alpha_k = 1$ 。故可以统一地写

$$\alpha_k = \min\left\{1, -\frac{{a_i}^T x_k + b_i}{{a_i}^T d_k} : i \notin A_k, {a_i}^T d_k < 0\right\} \quad \cdots (6)$$

对于第三个问题，我们这样来对 A_{k+1} 进行更新：

先看，如果 $(3'''')$ 的解 $\bar{x}_k$ 的Lagrange乘子 λ_k 的某个分量 $\lambda_k(i)$ 为负，则 $A_{k+1} = A_k \backslash \{i\}$ (如果全为正，则说明已经找到了原问题的最优解)。

接着，如果 $\bar{x}_k$ 不在可行域内，则设 i 是使 α_k 取到 $\min$ 的那一个 i，我们令 $A_{k+1} = A_{k+1} \cup \{i\}$。

综合上述，我们给出 $(3'')$ 完整的有效集法：

Algorithm 2 有效集法

Step 1：给定初始的可行点 x_0，令 $k = 0$；

Step 2：确定有效集 A_k，解子问题 $(3'''')$，得到解 $\bar{x}_k$ 和 λ_k；

Step 3：若 $\lambda_k(i) < 0$，则令 $A_{k+1} = A_k \backslash \{i\}$；

Step 4：利用 (6) 确定 $x_{k+1} = x_k + \alpha_k d_k$；

Step 5：若 $\alpha_k < 1$，令 $A_{k+1} = A_{k+1} \cup \{j\}$，其中

$$j = \arg\min\left\{-\frac{{a_i}^T x_k + b_i}{{a_i}^T d_k} : i \notin A_k, {a_i}^T d_k < 0\right\}$$

Step 6：令 $k = k + 1$，转Step 2，迭代直至输出。

六、线搜索-全局SQP算法

为了建立全局SQP算法，我们需要一个函数作为下降性的检测，便于调整步长或者信赖域半径。对于无约束问题，自然的选择是目标函数本身；而对于无约束问题，我们需要把约束条件的信息也包含进去。

Definition 1 l_1 精确罚函数

我们称

$$\Phi_1(x, \mu) = f(x) + \mu(\|\min\{g_I(x), 0\}\|_1 + \|h_J(x)\|_1)$$

为约束优化问题 $(1')$ 的 l_1 精确罚函数，其中 μ 称为罚因子。

这几个定理的证明都比较困难，在此不证。

Lemma 1

设 (d_k, λ_{k+1}) 是QP子问题 $(3')$ 的解，则 Φ_1 沿 d_k 的方向导数满足

$$D\left(\Phi_1\left(x_k,\mu\right);d_k\right)\leq$$
$$-{d_k}^TB_kd_k-\mu(\|\min\{g_I(x),0\}\|_1+\|h_J(x)\|_1)-{\lambda_{k+1,I}}^T\min\{g_I(x_k),0\}$$
$$-\lambda_{k+1,J}^Th_J(x_k)$$

特别地，若 $\mu\geq\|\lambda_{k+1}\|_\infty$ 且 ${d_k}^TB_kd_k>0$，则 d_k 是 Φ_1 的严格下降方向。

由此，可建立线搜索型的全局SQP算法：

Algorithm 3 线搜索型全局SQP算法

Step 1：给定初始点 x_0，对称正定阵 $B_0=I$，令 $k=0$；

Step 2：求解QP子问题 $(3')$ 得到 (d_k,λ_{k+1})，若 $\|d_k\|<\varepsilon$，则停止计算；

Step 3：确定 α_k 使得 $\Phi_1\left(x_k+\alpha_kd_k,\mu\right)\leq\min\limits_{0\leq\alpha\leq\delta}\Phi\left(x_k+\alpha d_k,\mu\right)+\varepsilon_k$，其中 $\sum\limits_{k=1}^{\infty}\varepsilon_k<+\infty$；

Step 4：令 $x_{k+1}=x_k+\alpha_kd_k$；

Step 5：用截断BFGS修正得到 B_{k+1}；

Step 6：令 $k=k+1$，转Step 2，迭代直到输出。

Theorem 2 线搜索型全局SQP算法的收敛性

假设

1. f,g_I,h_J 都连续可微；

2. 存在常数 m,M 使得

$$m\|d\|^2\leq d^TB_kd\leq M\|d\|^2$$

对一切 $k\geq 0,d\in\mathbb{R}^n$ 都成立；

3. 子问题 $(3')$ 总有解；

4. $\mu\geq\|\lambda_k\|_\infty$ 对所有的 $k\geq 0$ 都成立；

那么由Algorithm 3产生的点列或者终止于 $(1')$ 的KT点，或者其聚点是 $(1')$ 的KT点。

最后对Algorithm 3再作两点说明：

1. 我们很难选取 μ 使得 $\mu\geq\|\lambda_k\|_\infty$ 总成立，故可以在每一步更改 $\mu=\mu_k$，并且把它取成向量，把 Φ_1 改成

$$\Phi\left(x,\mu\right)=f\left(x\right)+\sum_{i\in I}\mu_i\left|g_i\left(x\right)\right|+\sum_{j\in J}\mu_j\left|h_j\left(x\right)\right|$$

每一步用下式来确定 $\mu_{k,p}$ ($p\in I\cup J$ 是分量)的值：

$$\mu_{k,p}=\max\left\{\left|\lambda_{k,p}\right|,\tfrac{1}{2}(\mu_{k-1,p}+\left|\lambda_{k,p}\right|)\right\},\ p\in I\cup J$$

2. Step 3中的精确线搜索通常计算量较大，可以改用非精确线搜索，找 α_k 使得

$$\Phi_1\left(x_k+\alpha_k d_k, \mu_k\right) \leq \Phi\left(x_k\right)+\beta \alpha_k D\left(\Phi_1\left(x_k, \mu_k\right) ; d_k\right)$$

编辑于 2023-05-24 11:38・IP 属地辽宁

最优化 计算数学 凸优化

▲ 赞同 47 ▼ 8 条评论 分享 喜欢 收藏 申请转载 ···

写下你的评论...

8 条评论 默认 最新

MathHub
我很喜欢抄书
2022-08-06 · IP 属地浙江 回复 4

吟雪千夏 作者
抄下来确实印象深刻一些
2022-08-06 · IP 属地辽宁 回复 喜欢

简朴人生
大佬您好呀，SQP算法复杂度怎么分析
08-28 · IP 属地安徽 回复 喜欢

呀呀呀
大佬您好，想请教一下如果性能函数f(x)是一个隐式函数的话，梯度那一项该怎么处理呢。或者您知道有相关资料可以参考的吗
05-14 · IP 属地浙江 回复 喜欢

魔法红酒烧肉
想知道 这些个算法的具体过程真的是可以记得住的吗...
2022-08-05 · IP 属地广东 回复 喜欢

搞飞机的阿十木
如果不考试的话不用记，理解原理就行，调用现成的库就行了
2022-09-07 · IP 属地陕西 回复 喜欢

吟雪千夏 作者 ▸ **搞飞机的阿十木**
是的
2022-09-07 · IP 属地安徽 回复 喜欢

展开其他 1 条回复 >

文章被以下专栏收录

推荐阅读