系 统 科 学 与 数 学
J. Sys. Sci. & Math. Scis.
27(6) (2007, 12), 923–934

一种修改的非单调线搜索 SQP 算法*

薛 文 娟

(同济大学应用数学系， 上海 200092)

沈 春 根

(上海金融学院应用数学系， 上海 201209)

摘要 提出了一种解非线性规划问题的修改的非单调线搜索算法，并给出了它的全局收敛性证明．不需要用罚函数作为价值函数，也不用滤子和可行性恢复阶段．该算法是基于多目标优化的思想：一个迭代点被接受当且仅当目标函数值或是约束违反度函数值有充分的下降．数值结果与 LANCELOT 作了比较，表明该算法是可靠的．

关键词 非线性规划，全局收敛性，线搜索， SQP.

MR(2000) 主题分类号 90C30

1 引 言

这篇文章给出了一个修改的非单调线搜索算法，以找出如下非线性规划问题 (NLP)

$$
\mathrm{P}\begin{cases} \min & f(x) \\ \text{s.t.} & c_i(x)=0, \quad i\in E, \\ & c_i(x)\le 0, \quad i\in I \end{cases} \tag{1}
$$

的局部最优解．其中 $f(x):R^n\to R, c(x)(=(c_1(x),c_2(x),\cdots,c_m(x))^{\mathrm{T}}):R^n\to R^m$ 是二次连续可微的， $E=\{1,2,\cdots,m_e\}, I=\{m_e+1,\cdots,m\}$.

最近几年， Fletcher and Leyffer[1] 提出了一种滤子的方法来解决非线性规划问题，用来代替传统的价值函数的方法，避免了罚因子的选取，而通常这个罚因子适当的选取是有难度的，这个方法关键的地方就是一个迭代点 (通过解一个信赖域序列二次规划 (SQP) 子问题得到) 被接受当且仅当目标函数值或者约束违反度函数值有充分的下降．另外，该方法的计算结果非常好，所以随后信赖域 SQP 子问题的全局收敛性由 Fletcher, Gould 和 Leyffer[2]; Fletcher, Leyffer 和 Toint[3] 证明．除了信赖域的方法，还有其它许多方法应用滤子的技巧，比如线搜索的方法 [4,5]; 内点法 [6−9]; 捆集法 [10] 和模式搜索算法 [11]. 以上的方法都是应用了

* 国家自然科学基金资助 (10371089, 10571137) 和上海 05 优青项目 (800008).
收稿日期：2004-06-24.

滤子的方法，本质上都基于多目标优化的思想．但是，很多方法都要用到可行性恢复阶段，这样会增加很大的计算量．所以， Ulbrich. M 和 Ulbrich. S[12] 提出了一种求解等式约束问题的非单调的信赖域 SQP 方法，它不需要滤子，无需可行性恢复阶段，也不需要罚函数．非单调的方法早在 1986 年由 Grippo,Lampariello 和 Lucidi[13] 提出，后来被应用到约束优化问题和信赖域方法上，见 [14−24]．这些方法都是结合了信赖域的方法．

在本文算法中，将非单调的思想和线搜索的方法相结合，其主要思想仍是将非线性等式约束和不等式约束问题理解为双目标优化问题．一个迭代点被接受，当且仅当目标函数值或者约束违反度函数值有充分的下降．但是，受 Ulbrich.M 和 Ulbrich.S[12] 的启发，不需要滤子的方法，而是应用非单调的线搜索方法，允许约束违反度函数值非单调的下降，使得在接受步长方面，比滤子的方法要灵活得多．该算法不需要罚函数或者增广的拉格朗日函数来检验迭代步的可接受性，避免了罚参数的选取．另外，不需要可行性恢复阶段，会节省大量的计算．最后，文章给出了该算法的数值计算的结果．通过和其它算法结果的比较，可以看出这个算法是非常有效的．

在这篇文章的第二部分，将详细介绍该算法的具体过程．在第三部分，将对这个修改的线搜索 SQP 问题的全局收敛性作出证明．在第四部分，将给出修改的算法的数值结果．

在给出这个算法和全局收敛性证明之前，先做出一些定义．记函数 f(x) 的梯度为 ∇f(x), ∇^{T}c(x) 为约束函数 $c(x)$ 的雅克比阵 (Jacobian), $L(x,\lambda)=f(x)+\lambda^{\mathrm{T}}c(x)$ 为 Lagrange 函数，λ 是 Lagrange 乘子， $B_k\in R^{n\times n}$ 是 Lagrange 函数的海色阵 (Hessian)$\nabla^2_{xx}L(x,\lambda)$ 的近似，下标 k 指的是求解 NLP 问题迭代的次数． $E=\{1,2,\cdots,m_e\}$, $I=\{m_e+1,m_e+2,\cdots,m\}$, $g_k=\nabla f(x_k)$, $I(x)=\{i|c_i(x)\le 0,i\in I\}$, $A_k=\nabla^{\mathrm{T}}c(x_k),\lambda_k$ 是 $QP(x_k)$ 子问题的 KKT 乘子，U 是单位矩阵．

2 修改的非单调线搜索算法

为方便起见，在这篇文章中，记 x^* 是 P 问题的局部解．在该算法中，选择 SQP 方法作为基本的迭代方法．在当前迭代点 x_k, 该算法的 QP 子问题为

$$\mathrm{QP}(x_k)\left\{\begin{array}{lll}\min & \nabla f(x_k)^{\mathrm{T}}d+\frac{1}{2}d^{\mathrm{T}}B_kd & \\ \text{s.t.} & \nabla c_i(x_k)^{\mathrm{T}}d+c_i(x_k)=0, & i\in E, \\ & \nabla c_i(x_k)^{\mathrm{T}}d+c_i(x_k)\le 0, & i\in I.\end{array}\right. \tag{2}$$

记该问题的解 (如果存在) 为 d_k, 并把 d_k 作为搜索方向．通过简单后退法找到步长 $\alpha\in(0,1]$, 记 $x=x_k+\alpha d_k$, 若 x 满足迭代要求，则记 $x_{k+1}=x,\alpha_k=\alpha$.

现在来看算法．在 NLP 问题中，有两个目标需要同时满足，即

$$\min f(x)$$

和

$$\min h(c(x)),$$

其中 $h(c(x))=\|c^{(-)}(x)\|_1=\sum\limits_{i=1}^{m}c_i^{(-)}(x)$,

$$c_i^{(-)}(x)=\left\{\begin{array}{ll}|c_i(x)| & i\in E, \\ \max\{0,c_i(x)\} & i\in I\end{array}\right. \tag{3}$$

(当然这两个目标同时满足是有难度的).

记 $\triangle f = f(x_k) - f(x_k + \alpha d_k)$ 为 $f(x_k)$ 迭代的实际下降量， $\triangle l = -\nabla f(x_k)^{\mathrm{T}} d_k$ 为 $f(x_k)$ 的线性下降量，那么 $f(x_k)$ 的充分下降性条件为

$$\triangle f \geq \sigma\alpha\triangle l, \sigma \in (0,1).$$

对于 $h(c(x))$ 下降方面，用类似 Ulbrich 在 [12] 中的技巧，使得 $h(c(x))$ 非单调地下降. 记 $Kt_k = \|A_k\lambda_k + g_k\|_1$. 为了使 $h(c_k)$ 和 Kt_k 均衡地下降，引入松弛量 R_k. 当 Kt_k 比 $h(c_k)$ 大很多时，继续对 $h(c_k)$ 的下降量进行松弛，具体做法如下.

如果 $h(c_k) << Kt_k$, 则取一个比 $h(c_k)$ 更大的数 R_k 来松弛其下降量，但为了使 $h(c_k)$ 最终趋于 0, 即保持 x_k 最终趋于可行点，需选取一个控制 R_k 的递减数列 a_{j_k}. a_{j_k} 的下降速度很慢，以达到使 $h(c_k)$ 的下降量松弛的目的. 取 $a_0 = \min\{0.1\max\{1, h(c_k)\}, Kt_k + h(c_k)\}, a_j = \frac{a_0}{j+1}, j \geq 1$. 那么当 $j \to +\infty$ 时， $a_j \to 0$, 且有 $\frac{1}{2} \leq \frac{a_{i+1}}{a_i} < 1$. 首先给出更新 R_k 的算法.

算法 A(更新 R_k)

设 $0 < \eta_1, \eta_2 < \frac{1}{2}$, 如果

$$h(c_k) < \min\{\eta_1 a_{j_k}, \eta_2 Kt_k\}, \tag{4}$$

则置

$$R_k = \min\{a_{j_k}, Kt_k\}, \tag{5}$$

如果 $g_k^{\mathrm{T}} d_k \geq -\frac{1}{2} d_k^{\mathrm{T}} B_k d_k$ 且 $R_k \geq Mhl_k$, 则置 $j_{k+1} = j_k + 1$; 否则置 $j_{k+1} = j_k$;
否则若 (1) 不成立，则置 $R_k = h(c_k), j_{k+1} = j_k$.

下面设 l 是一个大于 1 的正整数，记

$$Mhl_k = \max_{k-l+1 \leq i \leq k-1} h(c_i), \tag{6}$$

$$Reah_k = \max\{R_k, Mhl_k\}, \tag{7}$$

要求每个迭代步满足

$$Reah_k - h(c_{k+1}) \geq \alpha\eta Reah_k. \tag{8}$$

下面给出这个修改的非单调线搜索算法.

算法 B(修改的线搜索 SQP)

步 1 给定初始点 $x_0 \in R^n$, $\sigma \in (0,1), \eta \in (0,1), B_0 \in R^{n\times n}, 0 < \delta < 1, 0 < \varepsilon < 1, 0 < t < 1, k := 1$;

步 2 求解 $QP(x_k)$ 子问题，得到解 d_k 和 λ_k, 如果 $h(c(x_k)) \leq \varepsilon$ 且 $Kt_k \leq \varepsilon$, 则停止；否则置 $\alpha = 1$, 转入步 3;

步 3 更新 R_k. 如果 $Reah_k - h(c(x_k + \alpha d_k)) \geq \alpha\eta Reah_k$ 且 $g_k^{\mathrm{T}} d_k \leq -\frac{1}{2} d_k^{\mathrm{T}} B_k d_k$, 则转入步 4; 否则转入步 5;

步 4 计算 $\triangle f_k = f(x_k) - f(x_k + \alpha d_k)$, $\triangle l_k = -g_k^{\mathrm{T}} d_k$, 如果 $\triangle f_k \geq \sigma\alpha\triangle l_k$, 则转入步 7; 否则转入步 6;

步 5 如果 $Reah_k - h(c(x_k + \alpha d_k)) \geq \alpha\eta Reah_k$, 则转入步 7; 否则转入步 6;

步 6 $\alpha = t\alpha$, 转入步 3;

步 7 置 $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$, $k = k+1$; 计算 B_{k+1}; 转入步 2.

在步 7 中的 B_{k+1} 可采用 BFGS 方法求出，也可采用修改的 Broyden 方法 [25,26] 求出. 算法开始时，给定 x_0 作为初始点，在在每步的迭代中，步 2 至步 7 称为外循环，步 3 至步 6 称为内循环. 内循环实际是利用后退法求得 $\alpha \in (0,1]$, 使 x_{k+1} 满足充分下降性条件.

3 全局收敛性

在这一部分，将给出此修改的线搜索算法的全局收敛性的证明. 在给出证明之前，先做出如下的假设：

A1 设 $\{x_k\}$ 是由修改的线搜索 SQP 算法产生的迭代点列，并且假设 $\{x_k\}$, $\{x_k + \alpha d_k\}$ 包含于一个紧凸集 $S \in R^n$;

A2 $f(x)$ 和 $c_i(x)$, $i \in E \cup I$ 在 S 上二次连续可微；

A3 对任意的 k, B_k, λ_k 是一致有界的，其中 λ_k 是 QP(x_k) 的 KKT 乘子；

A4 对任意的 k, B_k 一致正定.

注 由假设 (A2)–(A4), 存在常数 $\delta, M > 0$, 使得 $\delta\|d\|^2 \leq d^{\mathrm{T}} B_k d \leq M\|d\|^2$, 对一切 k 和 $d \in R^n$ 成立，也有 $\|\lambda_k\|_\infty \leq M$ 成立.

下面考虑 QP(x_k) 子问题的 KKT 条件. 记 λ_k 为它的 Lagrange 乘子，则有

$$
\begin{aligned}
&g_k + B_k d_k = -\nabla c_i(x_k)\lambda_k, \\
&\lambda_i \left(c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} d_k\right) = 0, \quad i \in E \cup I, \\
&c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} d_k = 0, \quad i \in E, \\
&c_i(x_k) + \nabla c_i(x_k)^{\mathrm{T}} d_k \leq 0, \quad \lambda_i \geq 0, i \in I.
\end{aligned}
$$

引理 3.1 设 d_k 是 QP(x_k) 子问题的解，那么对任意 $x_k \in S, d_k \in R^n, x_k$ 到 $x_k + \alpha d_k$ 的线段包含于 S 中，且

$$
h(c(x_k + \alpha d_k)) \leq (1-\alpha) h(c(x_k)) + \frac{1}{2}\alpha^2 m M \|d_k\|^2.
$$

证 由泰勒 (Taylor) 展开定理，对所有的 $i \in E \cup I$,

$$
c_i(x_k + \alpha d_k) = c_i(x_k) + \alpha \nabla c_i(x_k)^{\mathrm{T}} d_k + \frac{1}{2}\alpha^2 d_k^{\mathrm{T}} \nabla^2 c_i(y_i) d_k,
$$

其中 y_i 在 x_k 到 $x_k + \alpha d_k$ 之间.

由 d_k 的可行性和假设 (A2) 可知，

$$
\begin{aligned}
c_i(x_k + \alpha d_k) &= (1-\alpha) c_i(x_k) + \alpha c_i(x_k) + \alpha \bigtriangledown c_i(x_k)^{\mathrm{T}} d_k + \frac{1}{2}\alpha^2 (d_k)^{\mathrm{T}} \nabla^2 c_i(y_i) d_k \\
&\leq (1-\alpha) c_i(x_k) + \frac{1}{2}\alpha^2 M \|d_k\|^2.
\end{aligned}
$$

对于 $i \in E$,

$$|c_i(x_k+\alpha d_k)| \le |(1-\alpha)c_i(x_k)| + \frac{1}{2}\alpha^2 M\|d_k\|^2 \le (1-\alpha)|c_i(x_k)| + \frac{1}{2}\alpha^2 M\|d_k\|^2.$$

同样，对于 $i \in I$ 有

$$h_i(c_i(x_k+\alpha d_k)) \le (1-\alpha)h_i(c_i(x_k)) + \frac{1}{2}\alpha^2 M\|d_k\|^2,$$

其中 $h_i(c_i(x_k)) = \max(0, c_i(x_k)), i \in I$. 因此

$$\begin{aligned} h(c(x_k+\alpha d_k)) &= \sum_{i=1}^{m_e}|c_i(x_k+\alpha d_k)| + \sum_{i=m_e+1}^{m} h_i(c_i(x_k+\alpha d_k)) \\ &\le (1-\alpha)h(c(x_k)) + \frac{\alpha^2 mM\|d_k\|^2}{2}. \end{aligned}$$

证毕.

定理 3.1 若基本假设成立，则内循环有限步终止.

证 若 x_k 是问题 P 的 KKT 点，那么 $d_k=0$ 是 QP(x_k) 子问题的解，并且算法终止，不进入内循环. 故可假设 $d_k \ne 0$. 由引理 3.1 和 (7) 式可得

$$\begin{aligned} h(c(x_k+\alpha_k d_k)) &\le (1-\alpha_k)h(c(x_k)) + \frac{1}{2}\alpha_k^2 mM\|d_k\|^2 \\ &\le (1-\alpha_k)Reah_k + \tfrac{1}{2}\alpha_k^2 mM\|d_k\|^2 \end{aligned}.$$

所以当

$$\alpha_k \le \frac{2(1-\eta)Reah_k}{mM\|d_k\|^2}.$$

时，有

$$h(c(x_k+\alpha_k d_k)) \le (1-\alpha_k\eta)Reah_k$$

若 $g_k^{\mathrm{T}}d_k > -\dfrac{1}{2}d_k^{\mathrm{T}}B_kd_k$, 则依据算法 B 知，内循环终止. 若 $g_k^{\mathrm{T}}d_k \le -\dfrac{1}{2}d_k^{\mathrm{T}}B_kd_k$, 则

$$\begin{aligned} f(x_k) - f(x_k+\alpha_k d_k) &= -\alpha_k g_k^{\mathrm{T}}d_k + o(\|\alpha_k d_k\|) \\ &= \sigma\alpha_k\triangle l + (1-\sigma)\alpha_k\triangle l + o(\|\alpha_k d_k\|). \end{aligned}$$

因为一定可以找到一个 α_k, 使得 $(1-\sigma)\alpha_k\triangle l + o(\|\alpha_k d_k\|) > 0$, 所以

$$f(x_k) - f(x_k+\alpha_k d_k) > \sigma\alpha_k\triangle l.$$

由上可知此内循环有限步终止. 证毕.

下面记 $h(c(x_k))$ 为 $h(c_k)$.

引理 3.2 在算法 A 中，若迭代步下标 k 满足 $j_{k+1} = j_k + 1$, 则对于任意的 $k' \ge k$, 都有 $h(c_{k'}) \le a_{j_k}$ 成立. 特别地，对于任意满足 $j_k \ge 1$ 的迭代步下标 k, 都有

$$h(c_k) \le 2a_{j_k} \tag{9}$$

成立.

证 若 $j_{k+1}=j_k+1$, 则依据算法 A 及 (6) 式，可得

$$h(c_k)<\min\{\eta_1 a_{j_k},\ \eta_2 K t_k\}\le \eta_1 a_{j_k},$$

且

$$a_{j_k}\ge R_k\ge Mhl_k=\max_{k-l\le i\le k-1} h(c_i).$$

从而对于 $k'=k-l+1,\cdots,k$, 有 $h(c_{k'})\le a_{j_k}$ 成立.

下面用数学归纳法证明对于 $k'\ge k-l+1$, 有 $h(c_{k'})\le a_{j_k}$ 成立.

对于 $k'=k-l+1,\cdots,k$, 已证 $h(c_{k'})\le a_{j_k}$ 成立，现假设对于 $k-l+1,\cdots,k'(\ge k)$, 有 $h(c_{k'})\le a_{j_k}$, 只需证明对于 $k'+1$, 有 $h(c_{k'+1})\le a_{j_k}$ 成立.

依据算法 A 及 a_{j_k} 的定义知，对于 $k'\ge k$, 有

$$R_{k'}\le\max\{h(c_{k'}),a_{j_{k'}}\}\le\max\{h(c_{k'}),a_{j_k}\}.$$

由算法 A 和定理 3.1 知，存在 $\alpha>0$ 使得

$$Reah_{k'}-h(c_{k'+1})\ge\alpha\eta Reah_{k'}>0.$$

故根据归纳假设及 Mhl_k 的定义知

$$\begin{aligned}h(c_{k'+1})&\le Reah_{k'}\\&=\max\{R_{k'},Mhl_{k'}\}\\&\le\max\{h(c_{k'}),a_{j_k},Mhl_{k'}\}\le a_{j_k}.\end{aligned}$$

从而由数学归纳法知，对于任意的 $k'\ge k$, 都有 $h(c_{k'})\le a_{j_k}$ 成立，其中 k 满足 $j_{k+1}=j_k+1$.

下面证明 (9) 也成立. 若 k 满足 $j_{k+1}=j_k+1$, 则直接可从算法 A 得到

$$h(c_k)\le 2a_{j_k},\quad j_k\ge 1.$$

现假设任意的 k' 满足 $j_{k'}=j_{k+1},k'>k+1$, 则再由 a_{j_k} 的定义知

$$h(c_{k'})\le a_{j_k}=a_{j_{k'}-1}\le 2a_{j_{k'}},\quad j_{k'}\ge 1.$$

因此 (9) 对于任意 $k(j_k\ge 1)$ 都成立.

引理 3.3 若有无限多次迭代，使得 $Reah_k\ne\max\{h(c_k),Mhl_k\}$ 和 $g_k^{\mathrm T}d_k\ge-\frac12 d_k^{\mathrm T}B_kd_k$ 成立，则 $j_k\to+\infty$ 且 $h(c_k)\to 0$.

证 由引理条件知，存在无限子列 $\{k'\}$, 使得 $R_{k'}=\min\{a_{j_{k'}},Kt_{k'}\}>h(c_{k'})$ 且 $R_{k'}>Mhl_{k'}$. 由 R_k 更新算法知， $j_{k'+1}=j_{k'}+1$, 对于任意的 k'. 因此有 $j_k\to\infty$. 再由引理 3.2 知，

$$h(c_k)\le 2a_{j_k},$$

于是 $a_{j_k}=\dfrac{a_0}{j_k+1}\to 0,j_k\to+\infty$. 从而 $h(c_k)<2a_{j_k}\to 0$. 证毕.

引理 3.4 若算法 B 不能有限终止，则 $\lim\limits_{k\to+\infty}h(c_k)=0$.

证 反证法. 若 $h(c_k)$ 不趋于 0, 由引理 3.3 知，存在充分大的整数 $k_1>0$, 使得当 $k\ge k_1$ 时，有

$$Reah_k=\max\{h(c_k),Mhl_k\}.\tag{10}$$

又由 (6) 和 (10) 可知

$$Reah_k = \max\{h(c_k), Mhl_k\} = \max_{k-l+1\leq i\leq k} h(c_i). \tag{11}$$

对于 $k \geq k_1$, 由 (8), (11) 和算法 A 知，存在 $\alpha_{\min} > 0$ 使得

$$\begin{cases} h(c_{k+1}) < \ (1-\alpha_{\min}\eta) \max\limits_{k-l+1\leq i\leq k} h(c_i), \\ h(c_{k+2}) < (1-\alpha_{\min}\eta) \max\limits_{k-l+2\leq i\leq k+1} h(c_i), \\ \quad \vdots \\ h(c_{k+l}) < \ (1-\alpha_{\min}\eta) \max\limits_{k\leq i\leq k+l-1} h(c_i), \end{cases} \tag{12}$$

且

$$\max_{k-l+1\leq i\leq k} h(c_i) \geq \max_{k-l+2\leq i\leq k+1} h(c_i) \geq \cdots \geq \max_{k\leq i\leq k+l-1} h(c_i). \tag{13}$$

从 (12) 和 (13) 可得

$$\max_{k\leq i\leq k+l-1} h(c_i) \leq (1-\alpha_{\min}\eta)^{\lfloor\frac{k-k_1}{l}\rfloor} \max_{k_1-l+1\leq i\leq k_1} h(c_i), \tag{14}$$

其中 $\lfloor\ \rfloor$ 表示向下取整. 所以当 $k \to +\infty$ 时,

$$\lim_{k\to+\infty} \max_{k\leq i\leq k+l-1} h(c_i) = 0,$$

故有

$$\lim_{k\to+\infty} h(c_k) = 0.$$

与假设 $h(c_k)$ 不趋向于 0 相矛盾，所以 $\lim\limits_{k\to+\infty} h(c_k) = 0$. 证毕.

定理 3.2 若基本假设成立，则算法产生的结果有如下两种情况：

1) 迭代到问题 P 的 KKT 点;

2) 至少存在一个聚点是 KKT 点.

证 只需证明算法的结果是第二种情况即可. 由定理 3.1, 内循环是有限步终止的，只要考虑外循环是无限循环即可.

当外循环是无限时，即有无限个迭代步，由基本假设知，$\{x_k\} \subset S$ 是有界的，所以必存在一个聚点 x^*. 不妨设 $\lim\limits_{k\in K_0,\ k\to+\infty} x_k = x^*$, 其中指标集 K_0 是无限集. 由引理 3.4 可得 $h(c_k) \to 0, k \to +\infty, k \subset K_0$, 故可得 x^* 是可行点.

若 x^* 满足约束规范条件，但不是 KKT 点，由反证法可知不成立. 分两种情况讨论：令

$$K_1 = \{k | g_k^{\mathrm{T}} d_k > -\frac{1}{2} d_k^{\mathrm{T}} B_k d_k, k \in K_0\}.$$

第一种情况： K_1 是无限指标集. 若存在 $K_2 \subset K_1$ 使得 $\lim\limits_{k\in K_2,\ k\to+\infty} \|d_k\| = 0$, 则可得 x^* 是 KKT 点，矛盾！所以假设存在 $\varepsilon > 0$, 使得对任意 $k \in K_1$, 有 $\|d_k\| > \varepsilon$, 由引理 4 知

$$\lim_{k\to+\infty} h(c_k) = 0.$$

故存在 $j_0 > 0$, 使得当 $k > j_0, k \in K_1$ 时，有

$$h(c_k) \le \frac{\varepsilon^2 \delta}{2M} \le \frac{\delta \|d_k\|^2}{2M} \le \frac{d_k^{\mathrm{T}} B_k d_k}{2M}. \tag{15}$$

而从 QP(x_k) 子问题的 KKT 条件知 $g_k + A_k \lambda_k + B_k d_k = 0$. 于是结合 (15), 可知对于 $k \in K_1, k > j_0$, 有

$$\begin{aligned} g_k^{\mathrm{T}} d_k &= -d^{\mathrm{T}} A_k \lambda_k - d_k^{\mathrm{T}} B_k d_k \\ &= \lambda_k^{\mathrm{T}} c_k - d_k^{\mathrm{T}} B_k d_k \\ &\le \|\lambda_k\|_\infty h(c_k) - d_k^{\mathrm{T}} B_k d_k \\ &\le M h(c_k) - d_k^{\mathrm{T}} B_k d_k \\ &\le -\frac{1}{2} d_k^{\mathrm{T}} B_k d_k. \end{aligned}$$

与 K_1 定义相矛盾，故 x^* 是 KKT 点.

第二种情况： K_1 是有限集，则存在 $j_0 > 0$, 使得当 $k > j_0$ 时，有 $g_k^{\mathrm{T}} d_k \le -\frac{1}{2} d_k^{\mathrm{T}} B_k d_k$. 由算法 A 知， a_{j_k} 不再下降，再结合引理 3.1 知，存在 $\alpha_{\min} > 0$, 使得 (8) 和 $\triangle f \ge \sigma\alpha\triangle l$ 成立且 $\alpha_k \ge \alpha_{\min}$. 当 $k > j_0$ 时，由定理 3.1 知存在 α_k 使得

$$f(x_k) - f(x_k + \alpha d_k) \ge \sigma\alpha_k \Delta l > \sigma\alpha_{\min}\Delta l \ge \frac{\sigma\alpha_{\min} d_k^{\mathrm{T}} B_k d_k}{2} \ge \frac{\sigma\alpha_{\min}\delta\|d_k\|^2}{2}.$$

两边对 k 从 $j_0 + 1$ 到无穷大求和，得

$$\begin{aligned} \sum_{k=j_0+1}^{\infty} (f(x_k) - f(x_k + \alpha_k d_k)) = f(x_{j_0+1}) - f(x^*), \\ \ge \sum_{k=j_0+1}^{\infty} \frac{\sigma\alpha_{\min}\delta\|d_k\|^2}{2}, \\ \ge \frac{\alpha_{\min}\delta\sigma}{2} \sum_{k=j_0+1}^{\infty} \|d_k\|^2. \end{aligned}$$

因为左边是有限数，所以

$$\sum_{k=j_0+1}^{\infty} \|d_k\|^2 < +\infty.$$

于是有 $\|d_k\| \to 0, k \to +\infty$. 从而得到 x^* 也是 KKT 点. 证毕.

4 数值结果分析

该算法是通过让 $h(c(x))$ 或 $f(x)$ 有充分下降，以使其达到全局收敛性，选取 Matlab 编程以实现该算法. 在这一节，主要讨论该算法的实现，具体给出算法中需要的参数，并对数据结果进行分析.

在算法的实现过程中，以下几点需要说明一下：

a) 终止条件：如果 $h(c(x_k)) \le \varepsilon$ 且 $Kt_k \le \varepsilon$, 则算法终止.

b) BFGS 迭代：令开始的 $B_0 = U$, 并且利用下面的 BFGS 公式进行迭代得到 H_k, 公式如下

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

其中

$$y_k = \begin{cases} \widehat{y}, & \widehat{y}_k^{\mathrm{T}} s_k \ge 0.2 s_k^{\mathrm{T}} B_k s_k, \\ \theta^k \widehat{y}_k + (1-\theta^k) B_k s_k, & \text{其他}, \end{cases}$$

和

$$\begin{cases} s_k = x_{k+1} - x_k, \\ \widehat{y}_k = \nabla f(x_{k+1}) - \nabla f(x_k) + \big(\ A_{k+1} - A_k\ \big)\lambda_k, \\ \theta^k = 0.8\ \dfrac{s_k^{\mathrm{T}} H_k s_k}{(s_k^{\mathrm{T}} H_k s_k - s_k^{\mathrm{T}} \widehat{y}_k)}. \end{cases}$$

c) 计算 R_k. 当 $h(c_k)$ 比 Kt_k 小很多时，取 $R_k = \min\{a_{j_k}, Kt_k\}$,

$$a_0 = \min\{0.1 \max\{1, h(c_0)\}, Kt_k + h(c_0)\},\ a_j = \frac{a_0}{j+1}, j \ge 1;$$

否则就取 $R_k = h(c_k)$. 这样就会使得 $h(c_k)$ 和 Kt_k 均衡下降.

d) 非单调下降. 并不要求 $h(c(x))$ 在每一步都下降，只要它能在 $l(l>1)$ 内保持下降即可，即 $Reah_k - h(c_{k+1}) \ge \alpha\eta Reah_k$.

e) 测试函数和参数的设定. 下述测试函数均选自 [27]. 在算法 A 和算法 B 中用到的参数设定如下： $\sigma = 0.1$, $B_0 = U$, $l = 5$, $\eta = 0.1$, $\varepsilon = 10^{-6}$, $t = 0.6$, $\eta_1 = \eta_2 = 0.2$, $\alpha_{\min} = 10^{-16}$, 其中 I 是单位矩阵.

LANCELOT [28] 是一种采用信赖域 – 罚函数法求解约束优化问题的软件，在这里，我们用该软件求解 [27] 中的测试问题，与本文算法结果相比较. 对该软件的参数设置如下： 1) 终止精度为 10^{-5}(约束和梯度的精度); 2) 采用预条件的共轭梯度法解 2– 范数的信赖域子问题； 3) 最大迭代次数为 1000 次.

对下表中的符号作如下说明：

Problem: 问题的编号.

NIT: 迭代的次数.

NF: 目标函数的计算次数.

NG: 目标函数梯度的计算次数.

NC: 约束函数的计算次数.

NA: 约束函数梯度的计算次数.

表 1 算法 A 与 LANCELOT 的数值比较

	算法 A					LANCELOT		
Problem	NIT	NF	NG	NC	NA	NIT	$\frac{NF}{NC}$	$\frac{NG}{NA}$
HS6	9	14	10	14	10	49	49	43
HS7	11	14	12	14	12	27	27	20
HS8	4	6	5	6	5	10	10	9
HS10	11	12	12	12	12	27	27	22
HS11	7	8	8	8	8	14	14	15
HS12	9	11	10	11	10	103	103	101
HS14	5	6	6	6	6	87	87	68
HS22	1	2	2	2	2	12	12	13
HS26	34	55	35	55	35	30	30	29
HS27	25	34	26	34	26	25	25	25
HS28	8	10	9	10	9	3	3	4
HS29	13	18	14	18	14	79	79	68
HS39	12	13	13	13	13	37	37	29
HS40	6	7	7	7	7	21	21	17
HS42	8	12	9	12	9	11	11	12
HS43	13	18	14	18	14	172	172	166
HS46	34	39	35	39	35	20	20	20
HS47	23	37	24	37	24	24	24	24
HS77	14	19	15	19	15	28	28	24
HS78	7	8	8	8	8	29	29	19
HS79	9	12	10	12	10	14	14	14
HS100	18	44	19	44	19	271	271	271
HS109	27	28	28	28	28		失败	
HS111	50	51	51	51	51	165	165	141
HS113	14	22	15	22	15		失败	
HS119	9	16	10	16	10	43	43	42

从上述的数据结果可以看出该算法能够有效地解决问题.

5 小 结

前几部分已经给出了该算法的详细步骤以及它的理论证明，并且编程分析了它的实际运算结果. 结果表明该算法还是比较令人满意的.

同时在这个算法中，还有几个问题需要注意. 首先，在求解 QP 子问题时，尚未考虑不相容的情形，对于这个问题，可采用文 [29] 中克服不相容的方法来解决；其次，对于算法 B 中求得满足条件 α 是用简单后退方法来寻找的，类似线搜索中运用的插值技巧，可以考虑用二次或三次插值来寻找 α_k; 再次，对于该算法的超线性收敛性的证明，该工作正在进行之中.

参 考 文 献

[1] Fletcher R and Leyffer S. Nonlinear programming without a penalty function. *Mathematical Programming*, 2002, **91**(2): 239–269.

[2] Fletcher R and Leyffer S and Toint P L. On the global convergence of a trust-region SQP-filter algorithm. *SIAM Journal on Optimization*, 2002, **13**: 44–59.

[3] Fletcher R, Gould N I M, Leyffer S, Toint Ph L and Wächter A. Global convergence of a trust region SQP-filter algorithms for general nonlinear programming. *SIAM Journal on Optimization*, 2002, **13**(3): 635–659.

[4] Wätcher A and Biegler L T. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 2005, **16**(1): 1–31.

[5] Wätcher A and Biegler L T. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 2005, **16**(1): 32–48.

[6] Ulbrich M, Ulbrich S and Vicente L N. A globally convergent primal–dual interior filter method for nonconvex nonlinear programming. *Mathematical Programming*, 2004, **100**: 379–410.

[7] Benson H Y, Shanno D F and Vanderbei R. Interion-point methods for nonconvex nonlinear programming jamming and numercical test. *Mathematical Programming*, 2004, **99**(1): 35–48.

[8] Liu X W and Sun J. Global convergence analysis of Line search interior-point methods for nonlinear programming without regularity assumptions. *Journal of Optimization Teory and Appications*, 2005, **125**: 609–628.

[9] Liu X W and Sun J. A robut primal-dual interior-point algorithm for nonlinear proguams. *SIAM Journal on Optimization*, 2004, **14**: 1163–1186.

[10] Fletcher R and Leyffer S. A Bundle Filter Method for Nonsmooth Nonlinear Optimization. Technical Report NA/195, Department of Mathematics, University of Dundee, Scotland, December 1999.

[11] Audet C and Dennis J E. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 2004, **14**(4): 980–1010.

[12] Ulbrich M and Ulbrich S. Nonmonotone Trust–Region methods for nonlinear equality constrained optimization without a penalty function. *Mathematical Programming*, 2003, **Ser.B**: 103–113.

[13] Grippo L, Lampariello F and Ludidi S. A nonmontone line search technique for Newtons's method. *SIAM Journal on Numerial Analysis*, 1986, **23**: 707–716.

[14] Bonnans J F, Panier E, Tits A and Zhou J L. Avoiding the msratos effect by means of a nonmonotone line search, II:Inequality constrained problems-feasible iterates. *SIAM Journal on Numerical Analysis*, 1992, **29**: 1187–1202.

[15] Deng N Y, Xiao Y and Zhou F J. Nonmonotonic trut-region algorithm. *Journal of Optimization Theory and Application*, 1993, **26**: 259–285.

[16] Grippo L, Lampariello F and Ludidi S. A truncated Newton method with nonmonotone line search for unconstrained optimization. *Journal of Optimization Theory and Applications*, 1989, **60**: 401–419.

[17] Grippo L, Lampariello F and Ludidi S. A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Methematik*, 1991, **59**: 779–805.

[18] Ke X and Han J. A nonmonotone trust region algorithm for equality constrained optimization. *Science in China*, 1995, **38A**: 683–695.

[19] Ke X, Liu G and Xu D. A nonmonotone trust-region algorithm for unconstrained optimization, *Chinese Science Bulletin*, 1996, **41**: 197–201.

[20] Lucidi S, Rochetih F and Roma M. Curcilinear stabilization techniques for trunated Newton methods in large–scale unconstrained optimization. *SIAM Journal on Optimization*, 1998, **8**: 916–939.

[21] Panier E and Tits A. Avoiding the Marats effect by means of a nonmonotone line search, I: General constrained problems. *SIAM Journal on Numerical Analysis*, 1991, **28**: 1183–1195,.

[22] Raydan M. The Barzilai and Borwein gradient methos for the large–scale unconstrained minimization problem. *SIAM Journal on Optimization*, 1997, **7**: 26–33.

[23] Toint P L. An assessment of nonmonotone line search techniques for unconstrained optimization. *SIAM Journal on Scientific Computing*, 1996, **17**: 725–739.

[24] Toint P L. A nonmonotone trust–region algorithm for nonlinear optimization subject to convex constraints. *Mathematical Programming*, 1997, **77**: 69–94 .

[25] Pu D and Tian W. A class of modified Broyden algorithms. *Journal of Computational Mathematics*, 1994, **8**: 366–379.

[26] Pu D and Tian W. A class of modified Broyden algorithms without exact line search. *Applied Mathematics*, 1995, **10**: 313–322.

[27] Schittkowski K. Test Examples for Nonlinear Programming Codes. Springer–Verlag, 1983.

[28] Conn A R, Gould N I M and Ph L Toint. LANCELOT. A Fortran Package for Large-Scale Nonlinear Optimization, in Numerical Optimization (Release A). Springer Ser. Comput. Math., Springer-Verlag, Berlin, 1992.

[29] 袁亚湘，孙文瑜．最优化理论与方法．北京：科学出版社， 2001.

AN SQP METHOD WITH A REVISED NON–MONOTONE LINE SEARCH

Xue Wenjuan

(*Department of Mathematics, Tongji University, Shanghai* 200092)

Shen Chungen

(*Department of Mathematics, Shanghai Finance University, Shanghai* 201209)

Abstract In the paper, a kind of revised non–monotone line search SQP method for constrained NLP problems is introduced, and the global convergence of this method is proved without using a penalty function as a merit function, a filter or the feasibility restoration phase. This method is based on the concept of multi–objective optimization: a trial point can be accepted if and only if either object function value decreases or the measure of violation constraints decreases. Numerical results, compared with LANCELOT, show that the approach is effective.

Key words NLP, global convergence, line search, SQP.