不等式约束优化一个新　可行 -、|　算法*1)

韩道兰
(广西民族大学理学院, 南宁 530006)

金宝 2)
(广西大学数学与信息科学学院, 南宁 530004)

张钦锋
(广西经济　理干部学院计算机系, 南宁 530007)

摘　要

本文对非线性不等式约束优化问题提出了一个新的可行 QP-free 算法. 新算法保存了现有算法的优点, 并具有以下特性: (1) 算法每次迭代只需求解三个具有相同系数矩阵的线性　程组, 计算量小; (2) 可行下降　向只需通过求解一个线性　程组即可获得, 克服了以往分别求解两个线性　程组获得下降　向和可行　向, 然后再做凸组合的困难; (3) 迭代点均为可行点, 并不要求是严格内点; (4) 算法中采用了试探性线搜索, 可以进一步减少计算量; (5) 算法中参数很少, 数值试验表明算法具有较好的数值效果和较强的稳定性.

关键词: 约束优化; QP-free 算法; 线性　程组; 全局收敛; 超线性收敛

MR (2000) 主题分类: 90C30, 65K05

1. 引　言

本文考虑以下不等式约束优化问题

$$\begin{aligned} &\min \quad {}_0(\ ) \\ &\text{s.t.} \quad (\ ) = (\ {}_i(\ ),\quad I) \prec 0,\quad I = \ 1,2,\cdots,m\ , \end{aligned} \tag{1.1}$$

其中 $_0$, $_i$, I : $^n \to$ 是连续可微 , 其可行集记为 $=$ ^{n} $(\ ) \prec 0$. 对于 , 记积极约束集 $I(\ ) =$ I $_i(\ ) = 0$, 且 $_i(\ ) =$ $_i(\ )$, 0 I.

序列二次规划 (SQP) 算法由于其快速收敛　特性, 成为最有效　算法之一, 引起很多学者　广泛关注　研究. 然而　类算法　次迭代都需要求解一个或多个二次规划 (QP) 子问题来获得迭代方向, 计算工作量大, 目前仍只适用于中小型问题.　外, 产生迭代方向　QP 子问题不一定相容, 虽然可以通过其他措施使其可解, 必然增加算法　复杂性和计算量, 理论证明也不完善.

相对于二次规划来说, 线性方程组　求解理论要完善得多. 通过有效地利用系数矩阵　疏性和对称性等一些好　性质, 可以较快地求解一个线性方程组. 所以, 很多学者试图放弃

* 2011 年 11 月 30 日收到.

1) 基金项目: 国家自然科学基金 (71061002,11171250), 广西自然科学基金 (2011GXNSFD018022) 和广西民族大学科研基金 (2011MDYB035) 资助项目.

2) 通　作者: 简金宝 (E-mail: jianjb@gxu.edu.cn).

求解 QP 子问题, 研究 QP-free 算法, 有时也称为序列线性方程组 (SSLE) 算法. QP-free 算法最初是由 Panier 等提出 [1], 算法虽然已考虑用线性方程组来代替 QP, 步迭代除求解两个线性方程组之外, 还需要解一个 QP 子问题, 计算量仍很大. 算法还需假设稳定点 个数有限才能证明全局收敛性, 其收敛速度也 为两步超线性且假设条件较强. 自 以后, 很多学者在 方面取得了 多成果, 见 [2–9], 其中包括可行方向法 [2,4,7,8] 和不可行方向法 [3,5,6,9]. Qi 等在文 [4] 中利用 KKT 条件 非光滑方程给出了一个新 QP-free 算法, 克服了 [1] 上述缺点, 但是为了避免 Maratos 效应, 还需求解一个最小二 子问题, 其计算量同求解 QP 子问题相当. 后来文 [7] 对其进行了改进, 提出了求解一系列形如

$$\begin{cases} H_k + \sum_{i \; I_k} \lambda_i \; {}_i(\; ^k) = \; {}_0(\; ^k), \\ {}_i(\; ^k)^T \; = \; ^k_i, \qquad I_k \end{cases}$$

线性方程组来获得搜索方向, 其中 ^{k} 为当前迭代点, H_k 为对称正定阵, I_k 为近似积极约束集, 右端项 ^k_i 待定. 然而算法中 可行下 方向需要通过求解两个线性方程组分别得到下 方向和可行方向后, 再做一次 组合才能获得, 而且算法要求迭代点必须是严格内点. 文 [8] 在模松弛 QP 子问题 基础上建立了一类新型 SSLE 算法, 然而算法 次迭代中都含有一个内循环, 无形中增加了计算量.

于是[illegible]

设 x^k 是当前迭代点, 首先用转轴运算产生近似积极集 $I_k^0 \supseteq I(x^k)$, 使得 $N_{I_k^0}(x^k)$ 列满秩.

转轴运算

步骤 (1)　令初始参数 $\varepsilon = \varepsilon_{k-1} > 0$, 其中 $\varepsilon_{k-1} > 0$ 是上次迭代相应于 x^{k-1} 作转轴运算时产生的参数.

步骤 (2)　由下式产生 ε- 积极约束集 $I(x^k,\varepsilon)$ 和矩阵 $N_{I(x^k,\varepsilon)}(x^k)$

$$L \triangleq I(x^k,\varepsilon) = \{i \in I \mid -\varepsilon < g_i(x^k) < 0\}, \quad N_L(x^k) = (\nabla g_i(x^k), i \in L). \tag{2.2}$$

步骤 (3)　若 $L = \emptyset$ 或 $\det[N_L(x^k)^T N_L(x^k)] \geq \varepsilon$, 令 $\varepsilon_k = \varepsilon$, $I_k^0 = L$, 停; 否则, 令 $\varepsilon := \varepsilon/2$ 重复步骤 (2).

由转轴运算和引理 2 可知, 矩阵 $(\nabla g_i(x^k,\epsilon), i \in I_k^0) = (\nabla g_i(x^k) + \epsilon\rho_{I_k^0}(x^k)\nabla g_0(x^k), i \in I_k^0)$ 对于任意 $\epsilon \in [0,1]$ 列满秩, 从而矩阵 $(\nabla g_i(x^k) - \nabla g_i(x^k,\epsilon), i \in I_k^0) = (\nabla g_i(x^k) + \epsilon\rho_{I_k^0}(x^k)(\nabla g_i(x^k) - \nabla g_0(x^k)), i \in I_k^0)$ 对于任意 $\epsilon \in [0,1]$ 都是列满秩的. 为方便起见, 记

$$\rho_k = \rho_{I_k^0}(x^k), \quad N_k^0 = N_{I_k^0}(x^k), A_J(x^k,\zeta) = (\nabla g_i(x^k) + \zeta(\nabla g_i(x^k) - \nabla g_0(x^k)), i \in J), \zeta \in [0,\rho_k].$$

引理 3. 如果指标集 I_k^0 是由转轴运算产生的, 那么对于任意 $J \subseteq I_k^0$ 和任意 $\zeta \in [0,\rho_k]$, 矩阵 $A_J(x^k,\zeta)$ 都是列满秩的.

为方便讨论算法的全局收敛性, 下面给出转轴运算的一些性质.

引理 4.[8] 如果假设 (A1) 成立, 且 x^k [0

算法 A

步骤 0. (初始步): 选取参数: $\alpha \quad (0,\frac{1}{2}),\ \beta \quad (0,1),\ \tau \quad (2,3),\ \nu \quad (0,1),\ \varepsilon \ \ _1,\ \nu_1,\ \nu_2 > 0$, 适当大 正数 M. 设定初始值: 初始点 ^{0} , 对称正定阵 H_0. 令 $:= 0$.

步骤 1. (产生工作集): (1) 令参数 $\varepsilon = \varepsilon_{k \ \ 1}$, 由转轴运算产生 I_k^0 相应 终止参数 ε_k.

(2) 按 (2.6),(2.5) (2.7) 分别计算 $(\ ^k),\ \omega(\ ^k,\ (\ ^k))$ 以 工作集 I_k. 若 $\omega(\ ^k,\ (\ ^k)) = 0$, 则 ^{k} 为问题 (1.1) KKT 点, 停.

步骤 2. 由 (2.1) 计算 $\rho_k = \rho_{I_k^0}(\ ^k)$, 并按下面 方式调整参数 ζ_k

$$\zeta_k = \begin{cases} \rho_0, & 若 \quad = 0; \\ \min \ \rho_k, \ ^k \ \ _{1,0} \ \ ^{\nu_1} + \ \bar{}^k \ \ _1 \ \ ^{\nu_2}, \zeta_{k \ \ 1}, & 若 \quad > 0. \end{cases} \tag{2.8}$$

步骤 3. 计算下面线性方程组 唯一解 $(\ ^{k0}, \lambda_{I_k}^{k0})$

$$\text{(LS1)} \qquad M_k \begin{pmatrix} \\ \lambda \end{pmatrix} = \begin{pmatrix} \ \ _0(\ ^k) \\ 0 \end{pmatrix}, \text{其中 } M_k = \begin{pmatrix} H_k & A_k \\ A_k^T & 0 \end{pmatrix}, A_k = A_{I_k}(\ ^k, \zeta_k).$$

步骤 4. 计算如下线性方程组 唯一解 $(\ ^k, \lambda_{I_k}^k)$

$$\text{(LS2)} \qquad M_k \begin{pmatrix} \\ \lambda \end{pmatrix} = \begin{pmatrix} \ \ _0(\ ^k) \\ \ ^k \end{pmatrix},$$

其中 $^k = (\ _i^k, \quad I_k),\ \bar{}^k = (\bar{}_i^k, \quad I_k)$,

$$_i^k = \begin{cases} \lambda_i^{k0}, & 若\ \lambda_i^{k0} < 0; \\ \ \ _i(\ ^k), & 若\ \lambda_i^{k0} \ \ _i(\ ^k) < 0; \\ 0, & 其他, \end{cases} \qquad \bar{}_i^k = \begin{cases} \lambda_i^{k0}, & 若\ \lambda_i^{k0} < 0; \\ \lambda_i^{k0} \ \ _i(\ ^k), & 若\ \lambda_i^{k0} \end{cases}$$

注 1. 算法　迭代过程中很难会出现 $\nearrow^k = 0$　情况, 如果一旦出现 $\nearrow^k = 0$, 算法将无法迭代下去,　时如果 $\mu_k < 0$, 算法将终止于一个非 KKT 点. 为了防止　一情况　出现, 我们将对 ζ_k 压缩一半, 重新计算方程组 (LS1) 与 (LS2).

注 2. 在算法　步骤 5 中引入试探性线搜索以检验步长 $_k = 1$ 是否能　接受, 若　接受, 则无需计算步骤 6 中　修正方向 $\tilde{\nearrow}^k$ 以　步骤 7 中　弧搜索, 从而进一步减少计算量.

为了分析方便, 下面给出方程组 (LS1)-(LS3)　等价形式.

$$\text{(LS1)} \quad \left\{ \begin{array}{l} H_k \nearrow^{k0} + \sum\limits_{i、I_k} \lambda_i^{k0}(\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k)) = \blacktriangledown \swarrow_0(\ ^k), \\ (\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k))^T \nearrow^{k0} = 0, \quad I_k. \end{array} \right. \tag{2.13}$$

$$\text{(LS2)} \quad \left\{ \begin{array}{l} H_k \nearrow^{k} + \sum\limits_{i、I_k} \lambda_i^{k}(\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k)) = \blacktriangledown \swarrow_0(\ ^k), \\ (\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k))^T \nearrow^{k} = \ _i^k, \quad I_k. \end{array} \right. \tag{2.14}$$

$$\text{(LS3)} \quad \left\{ \begin{array}{l} H_k \tilde{\nearrow}^{k} + \sum\limits_{i、I_k} \tilde{\lambda}_i^{k}(\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k)) = 0, \\ (\swarrow_i(\ ^k) \blacktriangledown \zeta_k \swarrow_i(\ ^k) \swarrow_0(\ ^k))^T \tilde{\nearrow}^{k} = \blacktriangledown \psi_k \blacktriangledown \vartheta_k \blacktriangledown \ _i(\ ^k + \nearrow^k), \quad I_k. \end{array} \right. \tag{2.15}$$

即方程组 (1.2) 中　参数 $\eta_i^k = \zeta_k \swarrow_i(\ ^k)$,　I_k, 且右端项 $\ _i^k$ 在不同　方程组中定义不同. 在假设 (A1) 下, 下面　引理成立.

引理 6. 线性方程组(LS1)-(LS3)　系数矩阵 M_k 非奇异.

经简单运算, 不难发现矩阵 M_k　逆矩阵可表示为 $M_k^{\ 1} = \begin{pmatrix} C_k & B_k \\ B_k^T & \blacktriangledown D_k^{\ 1} \end{pmatrix}$, 其中

$$D_k = A_k^T H_k^{\ 1} A_k, \ B_k = H_k^{\ 1} A_k D_k^{\ 1}, \ C_k = H_k^{\ 1} \blacktriangledown H_k^{\ 1} A_k B_k^T.$$

进而, 由于 $^{k} = 0$, 由 (2.18) 可知 $(^{k0}, {}^{k}) = (0,0)$, 于是结合 (2.9) 立知 (1) 成立.

(2) 对于 $I(^{k})$, 由 $I(^{k}) \subseteq I_k$ (2.9) 可知 $_i^k < 0$. 由注 1 可知 $^{k\ 1} = 0$, 结合 (2.18) 有 $(^{k\ 1,0}, {}^{k\ 1}) = (0,0)$, 又由 (2.9) 易知 $(^{k\ 1,0}, {}^{-k\ 1}) = (0,0)$. 结合 (2.8) 有 $\zeta_k > 0$. 因 , 由 (2.14) 第二个等式以 结论 (1) 易知 $_i(^{k})^T{}^{k} = \zeta_k\ _i(^{k})\ _0(^{k})^T{}^{k} + {}_i^k <$ $\zeta_k\ _i(^{k})\ _0(^{k})^T{}^{k} < 0$, $I(^{k})$.

引理 9. 算法 A 是适定 , 即对于充分小 正数 , 弧搜索 (2.12) 式恒成立.

3. 算法的全局收敛性

若算法 A 在点 ^{k} 处终止, 则由步骤 1,4 引理 7 可知迭代点 ^{k} 是问题 (1.1) KKT 点. 为证明在适当 条件下算法产生 无穷点列 ^{k} 存在聚点是问题 (1.1) KKT 点, 需要进一步假设如下:

(A2) 算法 A 产生 点列 ^{k} 有界.

(A3) 存在两个正常数 和 使得 $^2 < {}^T H_k < {}^2$, ^{n}, .

引理 10.[8] (1) 序列 ρ_k, $(^{k})$ 和 ζ_k 有界.

(2) 序列 M^{1} ^{T} ^{T} ^{T} ^{T} ^{Tf} ^{T} ^{T} ^{Tf} ^{Tf}

即 $d^{k-1,0}\to 0$, $\bar{d}^{k-1}\to 0$, K. 根据 λ^k 有界性, 不妨假设 $\lambda^{k-1}\to\lambda^*$, K, 由引理 8(1) (2.9) 可知,

$$d_0(x^{k-1})^T d^{k-1}=(d^{k-1,0})^T H_{k-1} d^{k-1,0}\sum_{\lambda_i^{k-1,0}<0}(\lambda_i^{k-1,0})^2+\sum_{\lambda_i^{k-1,0} f_i(x^{k-1})<0}\lambda_i^{k-1,0} f_i(x^{k-1})\to 0.$$

故由引理 11 易知, x^* 是问题 (1.1) KKT 点, (λ^*) 是相应 KKT 子.

情形 B. 若 $\zeta_*>0$, 则当 充分大时有 $\zeta_k \geq \frac{1}{2}\zeta_*$. 下面假设 (x^*, λ^*) 是序列 $(x^k, (\lambda^k))$ 任一聚点. 考虑到 I_k^0 和 I_k 是固定有限集 I 子集以 引理 10(2), 存在无穷指标子集 K 使得

$$I_k^0\equiv I^0,\ I_k\equiv I', (x^k, (\lambda^k))\to (x^*,$$

考虑到序列 $_0(\ ^k)$ 是单调下 , 并且 $\lim_{k, K} \ _0(\ ^k) = \ _0(\ ^*)$, 于是有 $\lim_{k\to} \ _0(\ ^k) = \ _0(\ ^*)$. 因 , 令 K 且 $\to$, 对 (3.6) 取极限可得 $\alpha \ _* \quad 0$, 与 $\alpha > 0, \ _* > 0 \quad > 0$ 相矛盾. 从而 (3.1) 成立, 故由引理 11 知 * 是 KKT 点, 与假设矛盾.

最后, 假设 $\bar{\ }^*$ 是相应于 * KKT 子, 则由 $I(\ ^*) \subseteq I^0$ 可知 $\bar{\ }^*_{I \ I^0} = 0$ $_0(\ ^*) +$ $^0\bar{\ }^*_{* \ I^0} = 0$. 因 , $\bar{\ }^* =$ Tc(=0)Tj[T321801Tm1T3121(.)]Tf6.9738006.973889.88660.1801Tm()TTf.41Tf9.9626009.9626163.08656

表 1 算法 A 的数值结果

Prob	n, m	Code	niter	nf_0	nf	final-f_0	w-set
Svanberg-10	10, 30	FSLE	36	227	258	15.731517	6
		SSLE	25	193	176	15.731517	6
		ALGO A	24	187	166	15.731517	6
Svanberg-30	30, 90	FSLE	101	777	864	49.142526	22
		SSLE	54	663	610	49.142526	22
		ALGO A	51	621	578	49.142526	21
Svanberg-50	50, 150	FSLE	108	881	968	82.581912	38
		SSLE	76	1004	954	82.581912	38
		ALGO A	69	683	721	82.581912	38
Svanberg-80	80, 240	FSLE	190	1666	1835	132.749819	61
		SSLE	163	1456	1623	132.749820	61
		ALGO A	157	1371	1511	132.749820	60
Svanberg-100	100, 300	FSLE	178	1628	1782	166.197171	77
		SSLE	167	1698	1653	166.197172	77
		SSLE	152	1534	1597	166.197170	75
Svanberg-500	500, 1500	FSLE	402	4020	4407	166.197171	398
		SSLE	378	3648	3966	166.197171	396
		ALGO A	354	3376	3628	166.197170	390

本文所测试 [14–16] 中 绝大部分算例都可以经过有限次迭代找到最优解, 并且迭代次数相对较少. 由表 1 可以看出, 与 FSLE, SSLE 相比, 本文算法 ALGO A 具有更好 数值表现. 值得指出 是, 算法 次迭代只需求解三个线性方程组, 由于 三个线性方程组具有相同 系数矩阵, 从而计算量相对较小. 另外, 步骤 5 中 试探性线搜索对于减少计算量发挥了较大 作用, 在不少迭代中是成功 . 最后, 算法中 参数较少, 易于调整, 数值试验表明本文所提出 算法具有非常好 稳定性, 适用于求解中大规模 问题.

参 考 文 献

[1] Panier E R, Tits A L, Herskovits J N. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization[J]. SIAM Journal on Control and Optimization, 1988, 26(4): 788–811.

[2] Gao Z Y, He G P, Wu F. An algorithm of sequential systems of linear equation for nonlinear optimization problems, Part I Inequality constrained problem[R]. Technical Report of Institute of Applied Mathematics, Academia Sinica. 1994: 94-31.

[3] Gao Z Y, He G P, Wu F. An algorithm of sequential systems of linear equations for nonlinear optimization problems with arbitrary initial point[J]. Science in China (Series A), 1997, 40(2): 1–10.

[4] Qi H D, Qi L Q. A new QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization[J]. SIAM Journal on Control and Optimization, 2000, 11: 113–132.

[5] 简金宝. 不等式约束最优化无严格互补条件下的快速收敛序列线性 程组算法 [J]. 数学学报, 2004, 47(4): 781–792.

[6] Jian J B, Cheng W X. A superlinearly convergent strongly sub-feasible SSLE-type algorithm with working set for nonlinearly constrained optimization[J]. Journal of Computational and Applied Mathematics, 2009, 225: 172–186.

[7] Yang Y F, Li D H, Qi L Q. A feasible sequential linear equation method for inequality constrained optimization[J]. SIAM Journal on Optimization, 2003, 13(4): 1222–1244.

[8] Jian J B, Han D L, Xu Q J. A new sequential systems of linear equations algorithm of feasible descent for inequality constrained optimization[J]. Acta Mathematica Sinica, English Series, 2010, 26(12): 2399–2420.

[9] Mo X D, Jian J B, Yang S M. An improved strongly sub-feasible SSLE method for optimization problems and numerical experiments[J]. Applied Mathematics and Computation, 2011, 217: 7226–7237.

[10] 堵丁柱. 非线性约束条件下的梯度投影 法 [J]. 应用数学学报, 1985, 8(1): 7–16.

[11] Facchinei F, Fischer A, Kanzow C. On the accurate identification of active constraints[J]. SIAM Journal on Optimization, 1998, 9: 14–32.

[12] Jian J B, Xu Q J, Han D L. A norm-relaxed method of feasible directions for finely discretized problems from semi-infinite programming[J]. European Journal of Operational Research, 2008; 186: 41–62

[13] Powell M J D. A fast algorithm for nonlinearly constrained optimization calculations[R]. Numerical analysis (Proc. 7th Biennial Conf., Univ. Dundee, 1977). Berlin: Springer-Verlay, 1978, 630: 144–157.

[14] Hock W, Schittkowski K. Test Examples for Nonlinear Programming Codes[A]. Lecture Notes in Economics and Mathematical Systems[C]. Berlin: Springer-Verlay, 1981, 187.

[15] Schittkowski K. More Test Examples for Nonlinear Programming Codes[A], Berlin: Springer-Verlay, 1987.

[16] Bongartz I, Conn A R, Gould N I M, Toint Ph L. CUTE: Constrained and unconstrained testing environment[J]. ACM Transactions on Mathematical Software, 1995, 21: 123–160.

A NEW TYPE FEASIBLE QP-FREE ALGORITHM FOR INEQUALITY CONSTRAINED OPTIMIZATION

Han Daolan
(*College of Science, Guangxi University for Nationalities, Nanning 530006, China*)
Jian Jinbao
(*College of Mathematics and Information Science, Guangxi University, Nanning 530004, China*)
Zhang Qinfeng
(*Computer Science Department, Guangxi Economic Management Cadre College, Nanning 530007, China*)

Abstract

In this paper, a new feasible QP-free algorithm for solving the nonlinear optimization problems with inequality constraints is presented. It reserves all the advantages of previous algorithms, and the interesting features of the algorithm are summarized as follows: (1) At each iteration, only three systems of linear equations with the same coefficient matrix need to be solved, which decreases largely the amount of computations; (2) A feasible descent direction can be obtained by solving only one system of linear equations, while the previous algorithms need to solve one linear system to get a feasible direction and another one to obtain a descent direction, and an improving direction is obtained by doing a convex combination; (3) The iteration points are all feasible without requiring to be strictly interior points; (4) The exploratory line search is introduced to the algorithm, and the computational cost can be further reduced; (5) The parameters in the proposed algorithm are few, and some numerical results illustrate that the proposed algorithm is efficient and stable.

Keywords: constrained optimization, QP-free algorithm, system of linear equations, global convergence, superlinear convergence

2000 Mathematics Subject Classification: 90C30, 65K05