求解 Minimax 优化问题的 Newton 型算法[*1)]

薛 毅

(北京工业大学应用数理学院 北京 100022)

A NEWTON LIKE ALGORITHM FOR SOLVING MINIMAX OPTIMIZATION PROBLEM

Xue Yi

(*College of Applied Science, Beijing University of Technology, Beijing* 100022)

Abstract

In this paper, a Newton like method for solving minimax optimization problems was proposed. The method belong to sequential quadratic programming method, the Hessian of quadratic programming subproblem is a convex combination of Hessian of objective functions. When Hessian of quadratic programming subproblem is not positive definite, the strategy to force matrix positive definite is used, so that there are good numerical solution for quadratic programming subproblem. The paper prove that the algorithm has global convergence and q-superlinear convergence properties. In order to show the new algorithm having good results, our preliminary numerical experiments are also reported.

Key words: Minimax Optimization Problem, Newton Method, SQP Method

§1. 引 言

Newton 法是求解无约束优化问题的最有效的算法，但由于需要计算目标函数的 Hesse 矩阵计算量大，因此人们大多采用拟 Newton(变度量法) 求解无约束问题. 近些年来，由于自动微分 (Automatic Differentiation)[1] 技术的提出和计算机速度与内存的不断提高，Newton 法又重新受到人们的重视. 此外，修正 Newton 法的一些技术，如负曲率方向的选择，强迫 Hesse 矩阵正定等，使 Newton 法有了良好的收敛性质.

考虑 minimax 优化问题

$$\min F(x) = \max_{1\le i\le m} f_i(x), \tag{1.1}$$

其中 $f_i(x), i=1,2,\cdots,m$ 是非线性函数，具有连续的二阶以上的偏导数.

* 2002 年 11 月 19 日收到.

1) 本文由国家自然科学基金 (19971008), 北京市教委基金资助.

Minimax 优化问题 (1.1) 可以转化为约束优化问题

$$\min \quad z, \tag{1.2}$$

$$\text{s.t.} \quad f_i(x) - z \le 0, \quad i = 1, 2, \cdots, m, \tag{1.3}$$

所以, 可以用求解约束问题的方法来求解问题 (1.2)–(1.3), 从而得到 minimax 问题 (1.1) 的解, 但这样做会失去 minimax 问题本身固有的特点. [2] 提出了求解 minimax 优化问题的 SQP 方法, 该方法本质上相当于求解无约束问题的变度量方法, 其二次规划子问题中目标函数的 Hesse 矩阵是基于拟 Newton 方程的修正公式得到的. 由于变度量法是求解无约束优化问题中有效的方法, 因此该方法对于 minimax 优化问题的求解有良好的性质.

本文在 [2] 的基础上, 提出一种新的求解方法, 其二次规划目标函数的 Hesse 矩阵是由 m 个函数的 Hesse 矩阵 $\nabla^2 f_i(x)$ 的凸组合构成, 因此称为 Newton 型算法. 当 $\nabla^2 f_i(x)$ 的凸组合构成的二次规划子问题的 Hesse 矩阵不正定时, 采取强迫矩阵正定策略, 从而保证了其二次规划子问题是一个凸规划问题, 使算法具有良好的收敛性与数值稳定性.

本文第二节给出相应的算法, 第三节给出相应的收敛性定理, 第四节给出算法的数值实验.

§2. 算　　法

算法 2.1

(1) 选取初始点 $x^{(1)} \in R^n$ 和 $\sigma \in \left(0, \frac{1}{2}\right)$, $\delta \ge 0$, $\varepsilon \ge 0$, 置初始乘子, 令

$$\lambda_i = \begin{cases} 1, & \text{当 } f_i(x^{(1)}) = F(x^{(1)}), \\ 0, & \text{否则}, \end{cases} \qquad i = 1, 2, \cdots, m$$

置 $\lambda^{(1)} = \lambda \Big/ \sum\limits_{i=1}^{m} \lambda_i$, 置 $k = 1$.

(2) 终止准则. 如果

$$\left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \sum_{i=1}^{m} \lambda_i^{(k)} \begin{bmatrix} \nabla f_i(x^{(k)}) \\ -1 \end{bmatrix} \right\| \le \varepsilon,$$

则停止计算.

(3) 计算 Hesse 矩阵. 令 $G_k = \sum\limits_{i=1}^{m} \lambda_i^{(k)} \nabla^2 f_i(x^{(k)})$, 对 G_k 作强迫正定的 Cholesky 分解, 令

$$L_k D_k L_k^T = G_k + E_k, \qquad B_k = L_k D_k L_k^T. \tag{2.1}$$

(4) 求解二次规划子问题

$$\min \quad \frac{1}{2} d^T B_k d + \frac{\delta}{2} t^2 + t, \tag{2.2}$$

$$s.t. \quad \nabla f_i(x^{(k)})^T d - t \le F(x^{(k)}) - f_i(x^{(k)}), \quad i = 1, 2, \cdots, m. \tag{2.3}$$

设 $(d^{(k)}, t_k)$ 为子问题的最优解, $\bar{\lambda}^{(k)}$ 为最优解处的 Lagrange 乘子.

(5) 一维搜索. 取 $\beta \in (0,1)$, 令 $j=0$. 如果

$$F(x^{(k)} + \beta^j d^{(k)}) \le F(x^{(k)}) + \sigma \beta^j t_k, \tag{2.4}$$

则置 $\alpha_k = \beta^j$; 否则置 $j=j+1$ 直到条件 (2.4) 成立为止.

(6) 置 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, $\lambda^{(k+1)} = \bar{\lambda}^{(k)}$, $k=k+1$, 转 (2).

注. 在算法 2.1 中, 强迫矩阵正定的 (2.1) 的方法是由 P. E. Gill 和 W. Murray 在 1972 年提出来的, 已有许多教科书作了详细的介绍, 如 [3] 的第三章中的算法 5.

引理 2.2. 设 $(d^{(k)}, t_k)$ 是二次规划子问题 (2.2)-(2.3) 的最优解. 若 $d^{(k)}=0$, 则 $t_k=0$.

证明见 [2].

由引理 2.2 结论和证明过程, 很容易得到如下定理:

定理 2.3. 若算法 2.1 产生的 $d^{(k)}=0$, 则 $x^{(k)}$ 是 minimax 优化问题 (1.1) 的稳定点.

定理 2.4. 若算法 2.1 产生的 $d^{(k)} \ne 0$, 则 $t_k<0$ 且 $d^{(k)}$ 是 $F(x)$ 在点 $x^{(k)}$ 的下降方向.

§3. 收敛性分析

本节讨论算法 2.1 的收敛性. 首先讨论算法的全局收敛性.

引理 3.1. 设 B_k 是算法 2.1 中对矩阵 G_k 作强迫正定的 Cholesky 分解得到的, 则存在 $\eta>0$, 使得对任意的 $d \in R^n$ 有

$$d^T B_k d \ge \eta \|d\|^2. \tag{3.1}$$

证明见 [3], 第三章第四节 (81 页) 中的引理.

假设 3.2.

1. $\{x^{(k)}\}$ 和 $\{d^{(k)}\}$ 有界.
2. 向量 $\begin{bmatrix} \nabla f_i(x) \\ -1 \end{bmatrix}$, $i \in I(x)$ 线性无关, 其中 $I(x)$ 是有效函数指标集, 即

$$I(x) = \{i \mid f_i(x) = F(x), i=1,2,\cdots,m\}.$$

定理 3.3. 设 $f_i(x)$ $(i=1,2,\cdots,m)$ 二阶连续可微, $\{x^{(k)}\}$ 是由算法 2.1 产生的点列, 若假设 3.2 成立, 则 $\{x^{(k)}\}$ 的任意聚点均是 minimax 优化问题 (1.1) 稳定点.

证明. 由 $\{x^{(k)}\}$ 有界, 知 G_k 有界, 所以 B_k 有界, 并由引理 3.1, 得到 [2] 中的假设 I 成立. 由 [2] 中的引理 3.1, 引理 3.2 和定理 3.3, 命题的结论成立.

由 minimax 优化问题 (1.1) 与约束优化问题 (1.2)-(1.3) 的等价性, 可以得到 minimax 优化问题的一阶必要条件, 即若 x^* 是 minimax 优化问题的最优解, λ^* 是相应的 Lagrange 乘子, 则有

$$\sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) = 0, \quad \sum_{i=1}^m \lambda_i^* = 1, \quad \lambda_i^*(f_i(x^*) - z^*) = 0, \ \lambda_i^* \ge 0, \ i=1,2,\cdots,m, \tag{3.2}$$

其中

$$z^* = F(x^*) = \max_{1\le i\le m} f_i(x^*). \tag{3.3}$$

由定理 3.3, 可以得到如下定理.

定理 3.4. 在定理 3.3 的假设下，由算法 2.1 计算出的 $\lambda^{(k)}$ 的任意聚点均收敛到 λ^*.

定理 3.3 和定理 3.4 表明，算法 2.1 具有全局收敛性. 下面讨论算法的收敛速率. 在讨论收敛速率之前，先看一个引理.

引理 3.5. 设 B 是 n 阶正定对称矩阵，$\begin{bmatrix} A \\ -e^T \end{bmatrix}$ 是列满秩矩阵，其中 $A \in R^{n\times r}$, $e=(1,1,\cdots,1)^T \in R^r$, 则矩阵

$$\begin{bmatrix} B & 0 & A \\ 0 & 0 & -e^T \\ A^T & -e & 0 \end{bmatrix}$$

非奇异.

证明. 只需证明方程组

$$\begin{bmatrix} B & 0 & A \\ 0 & 0 & -e^T \\ A^T & -e & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.4}$$

只有零解.

由方程组 (3.4) 得到

$$Bu + Aw = 0, \tag{3.5}$$

$$-e^T w = 0, \tag{3.6}$$

$$A^T u - ev = 0. \tag{3.7}$$

在式 (3.5) 左乘 u^T, 在式 (3.7) 左乘 w^T, 并利用式 (3.6), 得到 $u^T Bu = 0$, 由于 B 正定，所以 $u = 0$.

将 $u = 0$ 代回式 (3.5)-(3.7), 得到 $v = 0$ 和 $\begin{bmatrix} A \\ -e^T \end{bmatrix} w = 0$. 注意到系数矩阵是列满秩的，所以 $w = 0$.

为便于收敛速率的证明，设一维搜索的步长为 1, 而算法产生的序列 $\{x^{(k)}\}$ 和 $\{\lambda^{(k)}\}$ 收敛到问题的解 x^* 和相应的 Lagrange 乘子 λ^*. 因此给出如下假设:

假设 3.6.

1. $f_i(x), i = 1,2,\cdots,m$ 是二次连续可微.

2. 算法 2.1 产生步长 $\alpha_k = 1$, 且点列 $\{x^{(k)}\}$ 收敛到问题的最优解 x^*, $\{\lambda^{(k)}\}$ 收敛到 λ^*, 这里 λ^* 是 x^* 处的 Lagrange 乘子向量，即满足 (x^*, λ^*) 满足式 (3.2).

3. 矩阵 $\sum\limits_{i=1}^{m}\lambda_i^*\nabla^2 f_i(x^*)$ 正定.

在算法收敛的假设下，当 k 充分大时，子问题 (2.2)-(2.3) 的最优解 $(d^{(k)},t_k)$ 实际上就是一个等式约束二次规划的解. 由此给出如下假设:

假设 3.7. 当 k 充分大时， $(d^{(k)},t_k)$ 是问题

$$\min \quad \frac{1}{2}d^TB_kd+\frac{\delta}{2}t^2+t, \tag{3.8}$$

$$s.t. \quad \nabla f_i(x^{(k)})^Td-t=F(x^{(k)})-f_i(x^{(k)}), \quad i\in I(x^*) \tag{3.9}$$

的解.

在假设 3.7 成立时，存在 $\hat{\lambda}^{(k+1)}\in R^{|I(x^*)|}$(其分量为 $\lambda_i, i\in I(x^*)$) 使得

$$B_kd^{(k)}=-A(x^{(k)})\hat{\lambda}^{(k+1)}, \tag{3.10}$$

$$\delta t_k=-1+e^T\hat{\lambda}^{(k+1)}, \tag{3.11}$$

$$A(x^{(k)})^Td^{(k)}-t_ke=z_ke-\hat{f}(x^{(k)}), \tag{3.12}$$

其中 $\hat{f}(x)\in R^{|I(x^*)|}$, 其分量为 $f_i(x), i\in I(x^*)$, $A(x)\in R^{n\times|I(x^*)|}$, $\nabla f_i(x)(i\in I(x^*))$ 是 $A(x)$ 的列向量， $e=(1,1,\cdots,1)^T\in R^{|I(x^*)|}$, $z_k=F(x^{(k)})=\max\limits_{1\le i\le m} f_i(x^{(k)})$.

定理 3.8. 设假设 3.2, 假设 3.6 和假设 3.7 成立，在算法 2.1 中取 $\delta=0$, $\varepsilon=0$, 则

$$\left\|x^{(k+1)}-x^*\right\|=O\left(\|x^{(k)}-x^*\|h_k\right), \tag{3.13}$$

其中

$$h_k=\|x^{(k)}-x^*\|+\|\lambda^{(k)}-\lambda^*\|. \tag{3.14}$$

证明. 考虑 $\sum\limits_{i=1}^{m}\lambda_i\nabla f_i(x)$ 在 $(x^{(k)},\lambda^{(k)})$ 处展开，并注意到，由假设 3.2 知 $x^{(k)}$ 有界，从而 $\nabla^2 f_i(x^{(k)})$ 有界，得到

$$\begin{aligned}0=\sum_{i=1}^{m}\lambda_i^*\nabla f_i(x^*)&=\sum_{i=1}^{m}\lambda_i^*\left[\nabla f_i(x^{(k)})+\nabla^2 f_i(x^{(k)})(x^*-x^{(k)})+O(\|x^{(k)}-x^*\|^2)\right]\\&=\sum_{i=1}^{m}\lambda_i^*\nabla f_i(x^{(k)})+\left(\sum_{i=1}^{m}\lambda_i^{(k)}\nabla^2 f_i(x^{(k)})\right)(x^*-x^{(k)})+O(\|x^{(k)}-x^*\|h_k),\end{aligned} \tag{3.15}$$

其中 h_k 由式 (3.14) 定义， $O(\cdot)$ 对应的是向量.

由于 $\sum\limits_{i=1}^{m}\lambda_i^*\nabla^2 f_i(x^*)$ 正定，所以当 k 充分大时，有 $B_k=\sum\limits_{i=1}^{m}\lambda_i^{(k)}\nabla^2 f_i(x^{(k)})$. 由式 (3.10) 和式 (3.15) 得到

$$B_k(x^*-x^{(k+1)})+A(x^{(k)})(\hat{\lambda}^*-\hat{\lambda}^{(k+1)})+O(\|x^{(k)}-x^*\|h_k)=0, \tag{3.16}$$

其中 $\hat{\lambda}^*\in R^{|I(x^*)|}$, 其分量为 λ_i^*, $i\in I(x^*)$.

再考虑 $f_i(x)$ 在 $x^{(k)}$ 处的展开式. 当 $i \in I(x^*)$ 时，有

$$z^* = f_i(x^*) = f_i(x^{(k)}) + \nabla f_i(x^{(k)})(x^* - x^{(k)}) + O(\|x^{(k)} - x^*\|^2),$$

即

$$z^* e = \hat{f}(x^{(k)}) + A(x^{(k)})^T(x^* - x^{(k)}) + O(\|x^{(k)} - x^*\|^2), \tag{3.17}$$

这里 $O(\cdot)$ 为对应的向量. 将式 (3.12) 代入式 (3.17), 得到

$$A(x^{(k)})^T(x^* - x^{(k+1)}) - (z^* - z_k - t_k)e + O(\|x^{(k)} - x^*\|^2) = 0. \tag{3.18}$$

由式 (3.11), 式 (3.16) 和式 (3.18), 并注意到 $\delta = 0$, 得到

$$\begin{bmatrix} B_k & 0 & A(x^{(k)}) \\ 0 & 0 & -e^T \\ A(x^{(k)})^T & -e & 0 \end{bmatrix} \begin{bmatrix} x^* - x^{(k+1)} \\ z^* - z_k - t_k \\ \hat{\lambda}^* - \hat{\lambda}^{(k+1)} \end{bmatrix} = \begin{bmatrix} O(\|x^{(k)} - x^*\| h_k) \\ 0 \\ O(\|x^{(k)} - x^*\|^2) \end{bmatrix}. \tag{3.19}$$

由假设 3.2 和引理 3.5 知, 方程 (3.19) 的系数矩阵非奇异. 再由 $x^{(k)}$ 有界, 知逆矩阵有界, 因此有

$$\left\|x^{(k+1)} - x^*\right\| \le \left\| \begin{pmatrix} x^* - x^{(k+1)} \\ z^* - z_k - t_k \\ \hat{\lambda}^* - \hat{\lambda}^{(k+1)} \end{pmatrix} \right\| = O\left(\|x^{(k)} - x^*\| h_k\right).$$

由定理 3.8 知, 当 $\delta = 0$, $\lambda^{(k)} = \lambda^*$ 时, 算法 2.1 产生的点列 $\{x^{(k)}\}$ 是二阶收敛. 当 $\delta = 0$, $\lambda^{(k)} \ne \lambda^*$ 时, 算法 2.1 产生的点列 $\{x^{(k)}\}$ 能够达到超线性收敛.

在计算中尽量取 $\delta = 0$. 但由于 $\delta = 0$ 对应的二次规划子问题目标函数的 Hesse 矩阵半正定, 这可能会给数值求解带来一定的困难, 此时取 δ 为一个尽可能小的正数.

4. 数值结果

从文献 [4] 和 [5] 中选择了 10 个问题, 其中问题 1 ~ 问题 6 选自文献 [4], 问题 7 ~ 问题 10 选自文献 [5](文献 [5] 中同时也包含问题 1 和问题 3). 这 10 个问题的基本形式为

$$\min_x F(x) = \max_{1\le i\le m} f_i(x), \qquad x \in R^n.$$

[2] 详细列出了问题 1 ~ 问题 10 中 $f_i(x)$ 的表达式, 以及问题的标准初始点.

用 Matlab 5.3 进行计算, 在算法 2.1 中, 选取 $\beta = \dfrac{1}{2}$, $\sigma = 0.1$ 和 $\delta = 0$. (取修正 Cholesky 分解算法中的参数 $\delta = 10^{-5}$). 为了比较算法的有效性, 算法的终止准则与文献 [4] 和 [5] 相同的终止准则. 对应于文献 [4] 的终止准则为

$$\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\| + 1} < 10^{-10} \quad 或 \quad \left\|\sum_{i=1}^{m} \lambda_i^{(k)} \nabla f_i(x^{(k)})\right\| \le 10^{-6}. \tag{4.1}$$

对应于文献 [5] 的终止准则为

$$\frac{\|x^{(k)}-x^{(k-1)}\|}{\|x^{(k)}\|}<10^{-8} \text{ 或 } \frac{|F_k-F_{k-1}|}{|F_k|}<10^{-6} \text{ 或 } \left\|\sum_{i=1}^{m}\lambda_i^{(k)}\nabla f_i(x^{(k)})\right\|\le 10^{-8}. \quad (4.2)$$

计算结果见表 4.1 和表 4.2.

表 4.1 文 [4] 检验问题的计算结果

	本文的算法						文 [4] 的算法	
问题	初始点	NI	NF	NG	F(x)	I(x)	NI	NG
1	(1.0, −0.1)	4	9	5	1.952224495	1,2	7	13
	(100, −10)	11	23	12	1.952224494	1,2	12	23
2	(1.0, −0.1)	4	9	5	2.000000004	1,2,3	6	16
	(100, −10)	19	39	20	2.000000006	1,2,3	17	29
3	(0,0,0,0)	6	16	7	−44.00000000	1,2,4	10	26
	100(1,1,1,1)	11	23	12	−44.00000000	1,2,4	18	44
4	(3,1)	8	19	9	0.6164324362	1,3	10	28
	100(3, 1)	15	32	16	0.6164324356	1,3	11	19
5	(1,1,1)	11	23	12	3.599719300	2,5	13	25
	100(1,1,1)	24	53	25	3.599719300	2,5	15	28
6	(1,1,1)	5	11	6	0.05081632653	9,23,30	9	29
	100(1,1,1)	14	31	15	0.05081632672	9,23,30	25	56
总和		132	288	144			153	336

在表 4.1 中, NI 表示迭代次数, NF 表示目标函数的调用次数, NG 表示梯度函数的调用次数, $F(x)$ 表示终止点处的目标函数值, $I(x)$ 表示终止点处有效函数指标集. 为了与文献 [4] 的结果作比较, 将文献 [4] 中相应的迭代次数和梯度函数的调用次数 (文献 [4] 中没有列出目标函数的调用次数) 也列在表 4.1 中.

表 4.2 文 [5] 中的检验问题的计算结果

	本文的算法					文 [5] 的算法				
问题	NI	NF	NG	F(x)	P∼	NI	NF	NG	F(x)	P ∼
1	6	13	7	1.9522245	10^{-12}	10	11	11	1.9522245	10^{-10}
3	6	16	7	−44.000000	10^{-11}	11	15	12	−44.000000	10^{-10}
7	47	96	48	2.34685×10^{-6}	10^{-8}	47	54	48	1.223712×10^{-4}	10^{-7}
8	9	29	9	680.63006	10^{-8}	25	42	26	680.63006	10^{-10}
9	5	11	6	24.306209	10^{-13}	19	25	20	24.306209	10^{-8}
10	8	18	8	132.61425	10^{-7}	29	33	30	133.72825	10^{-9}
总和	81	183	85			141	180	147		

表 4.2 中的 P 表示目标函数在终止点处的相对精度, 即 $P=\dfrac{|F_k-F_{k-1}|}{|F_k|}$. 在表 4.2 中没有列出问题的初始点, 这是因为在本文算法中选取的初始点与文献 [5] 的初始点相同. 所

列的文献 [5] 的迭代次数、目标函数及梯度函数的调用次数是选该文作者认为最好的一组数据.

从表 4.2 可以看出，在用本文算法计算时，问题 7 需要的计算步骤较多，占了总计算量的一半以上，其原因是因为问题 7 的目标函数

$$f_i(x) = \frac{x_1 + x_2 t_i}{1 + x_3 t_i + x_4 t_i^2 + x_5 t_i^3} - \exp(t_i), \quad t_i = \frac{i-1}{10} - 1$$

是分式线性函数，与二次函数相差甚远，因此 Newton 法的计算效果不是很好.

综合表 4.1 和表 4.2，本文算法的计算迭代次数约是原算法的 72%，梯度函数的调用次数约是原算法的 47%，虽然 Newton 型算法需要计算 Hesse 矩阵，会多消耗一定的计算量，但如果采用自动微分技术等方法，这部分计算量会大大降低，因此可以得出这样的结论：求解 Minimax 优化问题的 Newton 型算法具有良好的计算效果.

参 考 文 献

[1] Rall, Louis B., *Automatic Differentiation: Techniques and Applications* Lectures Notes in Computer Science. Springer Verlag, Berlin, Volume 120 (1981).

[2] 薛毅，求解 Minimax 优化问题的 SQP 方法，系统科学与数学， 22:3 (2002), 355-364.

[3] 邓乃扬等，无约束最优化计算方法，北京，科学出版社， 1982.

[4] Vardi, A., *New Minimax Algorithm, Journal of Optimization Theory and Application*, 75:3 (1992), 613-634.

[5] Luksan, L., *A Compact Variable Metric Algorithm for Nonlinear Minimax Approximation*, Computing, vol. 36 (1986), 355-373.