文章编号:1672 - 058X(2011)01 - 0008 - 03

一类二次罚函数矫正算法

刘 芳[1],单 锐[2]
(1. 忻州师范学院 数学系,山西 忻州 034000;2. 燕山大学 西校区理学院,河北 秦皇岛 066004)

摘 要:以优化理论为基础,对求解一般约束优化问题提出一种算法. 它对于惩罚因子可以通过算法自我矫正逼近最优项,在一定条件下证明了算法的收敛性. 最后给出算例,结合 MATLAB 数值试验结果验证了这一算法的有效性.

关键词:约束最优化;罚函数法;收敛性
中图分类号:O174 **文献标志码**:A

罚函数法是解决约束优化问题[1-3]的重要方法,它的基本思想是把约束优化问题转化成求解一系列的无约束极小化问题,通过有关的无约束问题来研究约束极值问题. 经常采用的方法之一是在原来的目标函数上加上由约束函数组成的一个"惩罚项"来迫使迭代点逼近可行域,这种方法称为罚函数法. 如何选取罚函数,以加速迭代算法的收敛速度,一直是约束优化问题研究[4]的热点问题.

1 二次罚函数算法

定义1 对一般约束最优化问题:

$$\min f(x) \tag{1}$$

$$\text{s.t}\quad c_i(x) = 0; i = 1,2,\cdots,m \tag{2}$$

$$c_i(x) \geqslant 0; i = m+1, m+2,\cdots,n \tag{3}$$

通常使用的外函数形式为:$L(x,\sigma) = f(x) + P_\sigma(x) = f(x) + \sigma Q(x)$. 其中罚项为: $Q(x) = \sum_{i=1}^{m} |c_i(x)|^\beta + \sum_{i=m+1}^{n} |\min\{0, c_i(x)\}|^\alpha$, $\alpha \geqslant 1, \beta \geqslant 1$, σ 为参数,若取 $\alpha = \beta = 2$,称上述罚函数 $L(x,\sigma)$ 为二次罚函数.

问题(1)-(3)的可行域 R 为:

$$R = \{x \mid c_i(x) = 0, i = 1,2,\cdots,m; c_i(x) \geqslant 0, i = m+1, m+2,\cdots,n\}$$

显然,当 x 为可行点时,$Q(x) = 0$;当 x 不是可行点时,$Q(x) > 0$,而且 x 离可行域越远,$Q(x)$ 的值越大. 文献[1]指出,它的优点是允许从可行域的外部逐步逼近最优点. 但按上述定义的罚函数的缺点是:需要罚因子趋于无穷大,才可能使求解罚函数的极小和求解原问题等价,且随着初始值选择的不同和参数的选择不同,计算过程中产生的算法速度不能保证.

一般二次罚函数算法:1) 选定初始点为 x_0,选取初始惩罚因子 $\sigma_1 > 0$(可取 $\sigma_1 = 1$),惩罚因子的放大系数 $c > 1$(可取 $c = 10$),置 $k = 1$; 2) 以 x_{k-1} 为初始点,求解无约束问题 $\min\limits_{x \in R^n} L(x, \sigma_k)$;其中 $L(x,\sigma_k) = f(x) + P_{\sigma_k}(x) = f(x) + \sigma_k Q(x)$ 为二次罚函数,设其极小点为 x_k; 3) 若 $\sigma_k Q(x) < \varepsilon$,则 x_k 就是所要求的最优解,停止;否则转下一步; 4) 置 $\sigma_{k+1} = c\sigma_k$, $k = k+1$,转至步2).

收稿日期:2010 - 04 - 28;**修回日期**:2010 - 05 - 25.

作者简介:刘芳(1982 -),女,助教,山西忻州人,硕士,从事算法研究.

2　新的二次罚函数算法

为了有效改善罚函数,试图构造一种能够加速迭代算法收敛的外罚函数法. 此处在二次罚函数法的基础上提出一种用不同惩罚因子放大系数的罚函数,不仅证明了该罚函数和算法的合理性及迭代点列的收敛性,而且做了数值实验. 下面给出关于罚参数 σ_k 的矫正方法.

算法 1

① 取 $\sigma_0 = 1, \sigma_1 = c_1$; ② 令 $\sigma_k = \begin{cases} c_2\sigma_{k-2}, \text{当 } k \text{ 为偶数时} \\ \sigma_2\sigma_{k-1}/c_1, \text{当 } k \text{ 为奇数时} \end{cases}$.

算法 2

① 选定初始点为 x_0,选取初始惩罚因子 $\sigma_1 > 0$(可取 $\sigma_1 = 1$),惩罚因子的放大系数 $c_1 > 1, c_2 > 1$(其中 $c_1 < c_2$,一般取 $c_1 = 5, c_2 = 10$),置 $k = 1$; ② 以 x_{k-1} 为初始点,求解无约束问题 $\min\limits_{x \in R^n} L(x, \sigma_k)$;其中 $L(x, \sigma_k) = f(x) + P_{\sigma_k}(x) = f(x) + \sigma_k Q(x)$ 为二次罚函数,设其极小点为 x_k; ③ 若 $\sigma_k Q(x) < \varepsilon$,则 x_k 就是所要求的最优解,停止,否则转下一步; ④ 调整 $\sigma_k, k = k+1$,转至步②.

3　新算法的收敛性

引理 1　设函数 Q 和 P_σ 由定义 1 定义,$\{x_k\}$ 由算法产生,且罚参数序列 $\{\sigma_k\}$ 单调递增,则:

① 由 x_k 的定义知 $\begin{cases} L_{\sigma_k}(x_k) \le L_{\sigma_k}(x_{k+1}) \\ L_{\sigma_{k-1}}(x_{k-1}) \le L_{\sigma_{k-1}}(x_k) \end{cases}$,左面的两式相加,得 $(\sigma_k - \sigma_{k-1})(Q(x_k) - Q(x_{k-1})) \le 0$,因此 $Q(x_k) \le Q(x_{k-1})$.

② 由 $L_{\sigma_{k-1}}(x_{k-1}) \le L_{\sigma_{k-1}}(x_k)$ 得 $f(x_{k-1}) \le f(x_k) + \sigma_{k-1}(Q(x_k) - Q(x_{k-1})) \le f(x_k)$.

③ 由 L_σ 以及 x_k 的定义得 $L_{\sigma_k}(x_k) \le L_{\sigma_k}(x_{k+1}) = f(x_{k+1}) + \sigma_k Q(x_{k+1}) \le f(x_{k+1}) + \sigma_{k+1} Q(x_{k+1}) = L_{\sigma_{k+1}}(x_{k+1})$.

引理 2　设函数 Q 和 P_σ 由定义 1 定义,$\{x_k\}$ 由算法产生,且罚参数序列 $\{\sigma_k\}$ 单调递增,记 $\delta_k = Q(x_k)$,则 x_k 也是约束问题:

$$\begin{aligned} &\min f(x) \\ &\text{s.t}\quad Q(x) \le \delta_k \end{aligned} \tag{4}$$

的解.

证明　设 x 是问题(4)的可行点,有:

$$0 \le \sigma_k[Q(x_k) - Q(x)] = [L_{\sigma_k}(x_k) - L_{\sigma_k}(x)] + [f(x) - f(x_k)] \le f(x) - f(x_k)$$

因此 x_k 是问题(4)的解.

定理 1　设非线性约束问题(1)-(3)的最优解存在,设 $\{x_k\}$ 由算法产生,且罚参数序列 $\{\sigma_k\}$ 单调递增且趋于 $+\infty$,则 $\{x_k\}$ 的任何极限点都是问题(1)-(3)的可行域上的最优解.

证明　设 $\{x_k\} \to \bar{x}^*$,又设 x^* 是问题(1)-(3)的最优解,由于 x_k 是无约束问题 $\min L_{\sigma_k}(x) x \in R^n$ 的解,x^* 可行,即 $Q(x^*) = 0$,故有 $L_{\sigma_k}(x_k) \le L_{\sigma_k}(x^*) = f(x^*) + \sigma_k Q(x^*) = f(x^*)$,即 $L_{\sigma_k}(x_k) = f(x_k) + \sigma_k Q(x_k) \le f(x^*)$.

由此可得 $0 \le Q(x_k) \le \dfrac{f(x^*) - f(x_k)}{\sigma_k}$,由于 $\{x_k\} \to \bar{x}^*, \sigma_k \to \infty$. 故得 $f(\bar{x}^*) \le f(x^*)$,且 $Q(\bar{x}^*) = 0$. 即 $\bar{x}^*$ 可行,且 $f(\bar{x}^*) \le f(x^*)$. 但 x^* 是问题(1)-(3)的解,因此 $\bar{x}^*$ 也是问题(1)-(3)的解.

4　数值实验

通过数值实验来检验算法的有效性:

$$\begin{aligned} &\min f(x) = \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2 \\ &\text{s.t}\quad x_1 + x_2 - 1 = 0 \end{aligned}$$

用 matlab 编程[5]实现.

表1 $c=2$ 时的运行结果

罚因子	最优解		$\sigma_k Q(x)$	次数
1	0.222 2	0.666 7	1.230 0E-2	15
2^1	0.235 3	0.705 9	6.900 0E-3	15
2^2	0.242 4	0.727 3	3.700 0E-3	15
2^3	0.246 2	0.738 5	1.900 0E-3	15
2^4	0.248 1	0.744 2	9.614 8E-4	15
2^5	0.249 0	0.747 1	4.844 9E-4	15
2^6	0.249 5	0.748 5	2.431 9E-4	15
2^7	0.249 8	0.749 3	1.218 3E-4	15
2^8	0.249 9	0.749 6	6.097 7E-5	15
2^9	0.249 9	0.749 8	3.050 2E-5	15
2^10	0.250 0	0.749 9	1.525 3E-5	15
2^11	0.250 0	0.750 0	7.630 4E-6	15
2^12	0.250 0	0.750 0	3.814 4E-6	15
2^13	0.250 0	0.750 0	1.905 4E-6	33
2^14	0.250 0	0.750 0	9.542 8E-7	31
2^15	0.250 0	0.750 0	4.759 6E-7	32
2^16	0.250 0	0.750 0	2.381 7E-7	202
2^17	0.250 0	0.750 0	1.171 9E-7	34
2^18	0.250 0	0.750 0	6.013 4E-8	35
2^19	0.250 0	0.750 0	2.934 5E-8	49
2^20	0.250 0	0.750 0	1.469 2E-8	36
2^21	0.250 0	0.750 0	5.556 1E-9	38

表2 $c=10$ 时的运行结果

罚因子	最优解		$\sigma_k Q(x)$	次数
1	0.222 2	0.666 7	1.230 0E-2	15
10^1	0.246 9	0.740 7	1.500 0E-3	15
10^2	0.249 7	0.749 1	1.558 6E-4	15
10^3	0.250 0	0.749 9	1.562 2E-5	15
10^4	0.250 0	0.750 0	1.564 8E-6	202
10^5	0.250 0	0.750 0	1.555 4E-7	34
10^6	0.250 0	0.750 0	1.436 3E-8	36
10^7	0.250 0	0.750 0	3.998 6E-9	201

表3 $c_1=5, c_2=10$ 时的运行结果

罚因子	最优解		$\sigma_k Q(x)$	次数
1	0.222 2	0.666 7	1.230 0E-2	15
2×10^0	0.235 3	0.705 9	6.900 0E-3	15
10^1	0.246 9	0.740 7	1.500 0E-3	15
20	0.248 4	0.745 3	7.715 8E-4	15
10^2	0.249 7	0.749 1	1.558 6E-4	15
2×10^2	0.249 8	0.749 5	7.802 6E-5	15
10^3	0.250 0	0.749 9	1.562 2E-5	15
2×10^3	0.250 0	0.750 0	7.811 6E-6	15
10^4	0.250 0	0.750 0	1.564 8E-6	202
2×10^4	0.250 0	0.750 0	7.807 0E-7	32
10^5	0.250 0	0.750 0	1.555 4E-7	34
2×10^5	0.250 0	0.750 0	7.583 7E-8	35
10^6	0.250 0	0.750 0	1.436 3E-8	36
2×10^6	0.250 0	0.750 0	5.793 9E-9	38
10^7	0.250 0	0.750 0	3.998 6E-9	201

通过以上3种不同罚因子参数的实际运算,经统计后得到表4:

表4 不同罚因子参数下的运行结果

终止规则	$c=10$ 时的次数	$c=2$ 时的次数	$c_1=5, c_2=10$ 时的次数
$\varepsilon=10^{-4}$	60	135	90
$\varepsilon=10^{-5}$	262	180	120
$\varepsilon=10^{-6}$	296	259	354
$\varepsilon=10^{-7}$	332	562	423
$\varepsilon=10^{-8}$	533	685	497

5 结束语

提出了通过矫正罚因子提高算法收敛速度的算法,即利用二次函数不同罚因子的不同速度,将两种罚因子相结合使算法达到最优点.通过上述算法实验可以明显看出,所提出的罚函数算法较二次罚函数常用的两个罚因子参数有较快的收敛速度,而且在终止规则精确度要求越高的状态下,效果越好.

参考文献:

[1] 席少霖.非线性最优化方法[M].北京:高等教育出版社,1992

[2] 袁亚湘,孙文瑜.最优化理论与方法[M].上海:科学出版社,2003

[3] MENG Z Q,HU Q Y,DANG C Y. An Objective Penalty Function Method for Nonlinear Programming[J]. Applied Math Letter, 2004,17(6):683-689

[4] 陈宝林.最优化理论与算法[M].北京:清华大学出版社,2005

[5] 张可村.工程优化的算法与分析[M].西安:西安交通大学出版社,1988

(下转第13页)

$$\| u_{n+1} - u \| \leqslant (1 - a_n) \| u_n - u \| + a_n \theta \| u_n - u \| = [1 - a_n(1 - \theta)] \| u_n - u \| \quad (16)$$

在式(16)中,令 $a'_n = \| u_n - u \|$,$t'_n = a_n(1 - \theta)$,$b'_n = 0$,$c'_n = 0$,在定理 1 的假设条件下,它们显然满足引理的条件,因此由引理 1,就有 $\| u_n - u \| \to 0 (n \to \infty)$,即 $\lim\limits_{n\to\infty} u_n = u$.

注 2　定理 1 改进了文献[1]中定理 3.1 的结论,即把文献[1]的定理 3.1 中的强单调条件减弱为松弛强制条件.

参考文献:

[1] NOOR M A. Projection iterative methods for extended general variational inequalities[J]. J Appl Math Comput,2010,32:83-95

[2] NOOR M A. Some developments in general variational inequalities[J]. Appl Math Comput,2004,152:199-277

[3] NOOR M A. Extended general variational inequalities[J]. Appl Math Lett,2009,22:182-185

[4] 万波,王文惠. 求解一类广义混合变分不等式组的迭代算法[J]. 内蒙古大学学报:自然科学版,2010,41(1):41-47

[5] 姚莉. 广义混合变分不等式解的存在性与迭代算法[J]. 重庆工商大学学报:自然科学版,2009,26(4):311-315

Predictor-Corrector Algorithms for Extended General Variational Inequalities

YAO Li

(School of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing 400067, China)

Abstract: In this paper, we suggest and analyze a predictor-corrector projection iterative method for extended general variational inequalities. We also study the convergence of the new iterative method under much weaker conditions. Our results can be viewed as a novel and important extension and improvement of the previously known results .

Key words: variational inequalities; strong monotone; relaxed coercive; prediction-correction

责任编辑:李翠薇

(上接第 10 页)

A Kind of Revised Penalty Function Self-correcting Algorithm

LIU Fang[1], SHAN Rui[2]

(1. Department of Mathematics, Xinzhou Teacher's University, Shanxi Xinzhou 034000;

2. Department of Mathematics, Yanshan University, Hebei Qinhuangdao 066004, China)

Abstract: This paper gives a new algorithm for both equality and inequality constrained optimization problems based on optimization theory. Being self-correcting of the penalty factor, the algorithm is close to the optimal entry. Then it proves the astringency of the algorithm under certain condition. At last, an example is given, based on numerical test result of MATLAB, the feasibility of this algorithm is proven.

Key words: constrained optimization; penalty function methods; convergence

责任编辑:李翠薇