一种基于内点算法的三重目标过滤器优化算法的研究与仿真

宋 翌,阳彩霞,魏妮妮

武汉生物工程学院计算机与信息系,武汉 430415

摘要 大规模非线性最优化一直是规划中的研究热点。内点算法是一种有效的求解大规模不等式约束问题的算法,然而大多数过滤内点算法仅考虑了可行性和稳定性,忽略了辅助性对算法性能的影响,为此本文在综合过滤器法和内点算法特点的基础之上,提出了一种新的适用于大规模非线性优化的基于内点算法的三重目标过滤器法。新算法依据内点算法的卡罗需-库恩-塔克(KKT)条件,以可行性、辅助性和稳定性作为搜索步长的目标,将等式约束违反量,障碍目标函数和辅助条件作为过滤器选项计算搜索步长。通过搭建计算机仿真环境进行数值测试,从迭代次数、函数估计次数和运行时间三方面与基本过滤器法相比。测试结果表明,相同条件下三重目标过滤器法可以获得更大的搜索步长,实现快速收敛的目的。该算法具有良好的全局收敛性、鲁棒性和有效性。

关键词 最优化;过滤器法;内点算法;搜索步长

中图分类号 O224 **文献标志码** A **doi** 10.3981/j.issn.1000-7857.2013.01.010

Research and Simulation on the Triple-objective Filter Optimization Algorithm Based on Interior Point Algorithm

SONG Yi, YANG Caixia, WEI Nini

Department of Computer & Information Engineering, Wuhan Institute of Bioengineering, Wuhan 430415, China

Abstract The large scale nonlinear optimization has become a research focus in the planning, the interior-point algorithm is an effective method for solving large-scale inequality constraints, however most of the filter interior-point algorithm only consider the feasibility and stability, ignoring the adjuvant on the performance of algorithm, so that in this paper, in the light of the Karush-Kuhn-Tucker (KKT) conditions of the interior-point algorithm, a new algorithm, with feasibility, auxiliary and stability as the objective of the search step, use the amount of the violation of equality constraints, the obstacle objective function and auxiliary conditions as a filter option to calculate the search step and build a computer simulation environment for the numerical test, compared with the basic filter method from the number of iterations, function estimated times and run time. The test results show that under the same conditions the new algorithm, compared with the basic filter method, can get more search steps, and achieve fast convergence, having good global convergence, robustness and effectiveness.

Keywords optimization; filter; interior point algorithm; search step

0 引言

大规模非线性最优化一直是规划中的研究热点。众多学者对其作了大量的研究工作,并提出了很多算法,主要有牛顿、内点法、线性规划法、二次规划法和过滤器法[1]等。

内点算法是序列二次规化(SQP)方法的有效替代,是一种有效的求解大规模不等式约束问题的算法,已经成为优化问题的一个研究热点。它相对于SQP方法最大的区别在于它将不等式约束通过障碍函数的方法合并到目标函数中,从而避免了积极约束的选取。由于特殊的不等式处理方式,内点算法的卡罗需-库恩-塔克(KKT)条件相对于SQP算法的KKT条件增加了辅助性条件。由于内点算法的有效性和快速性,因此将过滤器法引入到了内点算法中。Benson等[1]提出了

收稿日期:2012-09-15;修回日期:2012-10-16

基金项目:武汉市教育局资助项目(2011083)

作者简介:宋翌,讲师,研究方向为运筹学等,电子信箱:jznu_song@yahoo.com.cn

障碍过滤器法和 Markov 过滤器法，并将其应用到了内点算法,并指出了过滤器法在内点算法中相对于价值函数方法的有效性。Wachter 等[2]提出了线性搜索过滤器法构架,并将其应用到了 SQP 算法和内点算法,但是他们提出的过滤器法是一种通用方法,并未结合内点算法的特点。

综上所述,大多数过滤内点算法仅考虑了可行性和稳定性[1~3],而忽略了辅助性对算法性能的影响。本文提出了基于内点算法的三重目标过滤器算法。该方法充分考虑内点算法的特点,从内点算法的 KKT 条件出发,分别以可行性、稳定性和辅助性作为选择步长的目标,以障碍目标函数、等式约束违反和辅助条件违反作为过滤项，从而达到快速收敛的目的。除此以外,Maratos 效应是约束优化在转换为无约束优化时出现的超线性收敛步长不可接受现象,因此应当在保证收敛的前提下尽可能地接受步长为 1 的步长因子[4]。三重目标过滤器算法相对于基本过滤器法，对于步长的接受条件更弱,更有利于避免 Maratos 效应,从而达到快速收敛的目的。

1　原–对偶内点算法

本文研究的问题可以描述为

$$\begin{aligned}&\min_x f(x)\\&\text{subject to } c(x)=0\\&x\geqslant 0\end{aligned}\qquad(1)$$

其中,$f:R^n\rightarrow R$ 和 $c:R^n\rightarrow R^m$ 是在开集 $\Omega\subset\Re^n$ 上的二次连续可导的函数。引入障碍项后,有

$$\begin{aligned}&\min_x B(x)=f(x)-\mu\sum_{i=1}^{n}\ln x_i\\&\text{subject to } c(x)=0\end{aligned}\qquad(2)$$

其中,μ 为障碍参数,正数。原–对偶内点法是基于将牛顿法应用于扰动一阶最优性条件(KKT 条件)的方法,因此原问题的 KKT 条件可以写为

可行性　$\boldsymbol{c}(\boldsymbol{x})=0$　(3)

稳定性　$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{Z}=\boldsymbol{g}$　(4)

辅助性　$\boldsymbol{xz}=0$　(5)

$\boldsymbol{x}\geqslant 0,\ \boldsymbol{y}\geqslant 0$　(6)

其中,$\boldsymbol{y}$, $\boldsymbol{z}=\boldsymbol{\mu X}^{-1}e$ 为拉格朗日乘子,$\boldsymbol{g}(x)$和 $\boldsymbol{A}$ 分别为 $f(x)$的梯度和 $c(x)$的雅克比矩阵,$\boldsymbol{X}$ 为以 x_i 为对角量的对角矩阵。

将 KKT 方程的式(5)进行扰动,可以得到:

$$\begin{cases}\boldsymbol{c}(\boldsymbol{x})=0 & x\geqslant 0\\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{z}=\boldsymbol{g} & z\geqslant 0\\ \boldsymbol{XZ}e=\hat{\boldsymbol{\mu}}\,e\end{cases}\qquad(7)$$

其中,$\hat{\boldsymbol{\mu}}>0$,e 为常数。在算法过程中，取 $\hat{\boldsymbol{\mu}}=\sigma\boldsymbol{\mu}$，其中 $\sigma\in(0,1)$为中心参数,且

$$\boldsymbol{\mu}=\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z}}{n}\qquad(8)$$

本文算法中,根据 KKT 条件定义可行性、稳定性和辅助性如下:

可行性量度　$\theta(x)=\|\boldsymbol{c}(\boldsymbol{x})\|$　(9)

稳定性量度　$\theta_s=\|\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{z}\|$　(10)

辅助性量度　$\theta_c=\|\boldsymbol{XZ}\|$　(11)

以上各量度分别作为过滤器的各项。

对于 KKT 条件(7),通过牛顿法可以得到满足条件的点(x,y,z)。相对应的牛顿系统可以写为

$$\begin{bmatrix}-\boldsymbol{w}_k & \boldsymbol{A}^{\mathrm{T}} & \boldsymbol{I}\\ \boldsymbol{A} & 0 & 0\\ \boldsymbol{Z} & 0 & \boldsymbol{X}\end{bmatrix}\begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix}=\begin{bmatrix}\boldsymbol{g}(\boldsymbol{x})-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}-\boldsymbol{z}\\ \boldsymbol{c}(\boldsymbol{x})\\ \boldsymbol{XZ}e-\boldsymbol{\mu}e\end{bmatrix}\qquad(12)$$

当通过式(12)求得搜索方向(p_x,p_y,p_z)时,为了获得下一个迭代点

$$\begin{aligned}x_{k+1}&=x_k+\alpha p_x\\ y_{k+1}&=y_k+\alpha p_y\\ z_{k+1}&=z_k+\alpha p_z\end{aligned}\qquad(13)$$

步长 $\alpha\in(0,1]$必须确定。三重目标过滤器法正是求取迭代步长的方法。

2　基于内点算法的三重目标过滤器算法

2.1　过滤器算法

对于非线性数学规划式(1)的解可以看作是由两个相互竞争的目标组成的:最小化可行性量度 $\theta(x)$和最小化目标函数$f(x)$。因此,式(1)可以转化为两目标优化问题。通过罚函数的方法组合两个目标是常用的方法,而 Fletcher 和 Leyffer[5]提出了基本过滤器法,从而将约束违反和目标函数的下降同时考虑,定义了一个过滤器来处理这两个目标,取得了很好的效果。过滤器在迭代过程中起到收集以前迭代信息的作用,同时作为选择新迭代点的标准。在基本过滤器法中,一个过滤器 F 是由点对$(\theta(x),f(x))$组成,它与点 x 对应相关,并要求在过滤器中任何项不被其他点支配。其中支配的概念是基本过滤器法的核心,起到选择存储信息和选择新点的标准的作用:如果一个点 x 加入到过滤器中,则所有的被点 x 支配的项将被移除;一个迭代点 x 可以接受,则必须不能被过滤器中任何点支配。

2.2　三重目标过滤器算法的定义

三重目标过滤器算法是以内点算法的 KKT 条件为基础,将 KKT 条件转化为步长搜索的评判标准。该方法是将障碍问题转化为一个三目标优化问题,以可行性、辅助性和稳定性作为三重目标过滤器法的过滤目标。但是简单地将 KKT 条件中的式(7)直接作为过滤项是不合适的,必须进行必要的转换。对于稳定性量度式(10),它是障碍目标函数梯度的范数,以它作为标准可以起到快速收敛的作用,但同时对于非凸问题,这种方法将影响解的全局最优性,因为梯度的范数为零点并不一定是全局最小点。因此用障碍目标函数来代替障碍目标函数梯度,从而防止产生局部极小的问题。三重目标过滤器法的各目标的量度标准重新定义为

可行性量度　$\theta(x)=\|\boldsymbol{c}(\boldsymbol{x})\|$　(14)

稳定性量度　　$\theta_s=B(x)$　　(15)

辅助性量度　　$\theta_c=\|xz\|$　　(16)

由于以上三重目标过滤器法与基本过滤器法存在着很大的不同,下面定义该方法的基本概念。

定义 1 (支配)点 x,或者与该点对应的集合$(\theta(x),\theta_s(x),\theta_c(x))$支配于点 x',或者点 x'对应的集合$(\theta(x'),\theta_s(x'),\theta_c(x'))$,即

$$\theta(x)\leqslant\theta(x')\quad \theta_s(x)\leqslant\theta_s(x')\quad \theta_c(x)\leqslant\theta_c(x')$$

或者等价于以下不满足于不等式:

$$\max\{\theta(x)-\theta(x'),\theta_s(x)-\theta_s(x'),\theta_c(x)-\theta_c(x')\}>0$$

定义 2 过滤器 F 是一个集合$\{(\theta^f,\theta_s^f,\theta_c^f)\}$,有限集合 $F\subset R^3$,其中有限集合 F 内没有集合$\{(\theta^f,\theta_s^f,\theta_c^f)\}$支配其他集合。

定义 3 如果一个点 x 需要加入过滤器 F,则应进行以下操作:

$$F\mapsto F=\{(\theta(x),\theta_s(x),\theta_c(x))\}\cup\{(\theta^f,\theta_s^f,\theta_c^f)\in F:\min\{\theta(x)-\theta(x'),\theta_s(x)-\theta_s(x'),\theta_c(x)-\theta_c(x')\}<0\}\quad(17)$$

3 实际算法

下面介绍求解障碍问题式(2)的三目标线性搜索内点算法的整个步骤。

给定起点(x_0,y_0,z_0),其中 $x_0,z_0>0$;给定障碍参数初始值 $\mu_0>0$; 给定常数 $\varepsilon_{tol}>0$;$s_{max}\geqslant1$;$K_\varepsilon>0$;$K_\mu\in(0,1)$;$\theta_\mu\in(1,2)$;$\tau_{min}\in(0,1)$;$K_\Sigma>1$;$\theta_{max}\in(\theta(x_0),+\infty)$;$\theta_{min}>0$;$\gamma_f,\gamma_B,\gamma_\theta\in(0,1)$;$\delta>0$;$\gamma_\alpha\in(0,1]$;$s_\theta>1$;$s_B=2s_\theta$;$\eta_B\in(0,1/2)$;$K_{soc}\in(0,1)$;$p_{max}\in\{0,1,2,\cdots\}$。

第 1 步:初始化迭代次数 $j\leftarrow0$ 和 $k\leftarrow0$,且 $F_0=\phi$。通过式(11)计算得 τ_0。

第 2 步:判断原问题的收敛性。

第 3 步:判断障碍问题的收敛性。

第 4 步:通过式(6)和式(8)计算搜索方向(p_x,p_y,p_z)。

第 5 步:回退线性搜索,首先初始化线性搜索步长。设置 $\alpha_{k,0}=1$ 和 $l\leftarrow0$。

第 6 步:计算新的迭代点。设置 $x_k(\alpha_{k,l}):=x_k+\alpha_{k,l}p_k^x$。

第 7 步:判断对于过滤器是否可以接受。如果$(\theta(x_k(\alpha_{k,l})),\theta_s(x_k(\alpha_{k,l})),\theta_c(x_k(\alpha_{k,l})))$对于过滤器 F_k 是可接受的,跳转至第 9 步,否则跳转至第 8 步。

第 8 步:选择新的迭代步长。设置 $\alpha_{k,l+1}=\dfrac{1}{2}\alpha_{k,l}$ 和 $l=l+1$。如果试探步长太小,即 $\alpha_{k,l+1}<\alpha_{mink}$,则跳转至可行修复阶段第 12 步。否则,返回第 6 步。

第 9 步:接受试探步长。设置 $\alpha_k:=\alpha_{k,l}$,和更新估计乘子 y_{k+1} 和 z_{k+1}。

第 10 步:扩大过滤器。添加点$(\theta(x_k(\alpha_{k,l})),\theta_s(x_k(\alpha_{k,l})),\theta_c(x_k(\alpha_{k,l})))$到过滤器,并去除过滤器内被点$(\theta(x_k(\alpha_{k,l})),\theta_s(x_k(\alpha_{k,l})),\theta_c(x_k(\alpha_{k,l})))$支配的点。

第 11 步:循环迭代。增加迭代次数 $k=k+1$,返回第 2 步。

第 12 步:可行修复。根据式(17)扩大过滤器,以减小可行性量度 $\theta(x)$为目标计算新的迭代点 $x_{k+1}>0$,并将新的迭代点 x_{k+1} 加入过滤器。然后转到一般性迭代步骤第 11 步。

4 仿真实验

为了证明算法的有效性,进行了计算机仿真测试。采用约束和非约束(CUTEr)测试集合,进行数值比较。其中初始点采用 CUTEr 的原始问题的初始设定值。采用 Dolan-More 提出的性能标准作为测试标准[6]。具体测试标准描述如下。

给定包含测试问题 n_p 的测试集合 P,和不同的执行算法选项 n_s,这种性能标准对每个问题和每个选项获得的性能指标值(例如迭代次数和 CPU 时间)进行描绘,从而对不同算法的进行比较。不同问题 p 和选项 s 的性能比定义为

$$r_{p,s}:=\frac{t_{p,s}}{\min\{t_{p,s}:1\leqslant s\leqslant n_s\}}\qquad(18)$$

如果对于问题 p,选项 s 无法求解,定义 $r_{p,s}:=\infty$。然后,定义测试问题的性能比率

$$\rho_s(\tau):=\frac{1}{n_p}card\{p\in P:r_{p,s}\leqslant\tau\}\qquad(19)$$

它代表测试问题中选项 s 是求解问题的最好选项,并且选项 s 的性能比小于性能因子 $\tau\geqslant1$ 的问题所占比例。算法的性能是通过绘制选项 s 的 p_s 关于 τ 的函数进行比较的。

在数值测试中,选择了 4 种过滤器法,并在内点算法构架内进行了比较,这 4 种过滤器法分别是 Triple-objective Filter、IPOPT、FL Filter 和 Ulbrich Filter 算法。其中 Triple-objective Filter 是本文提出的三重目标过滤器法;IPOPT 是由 Wachter 和 Biegler[2]提出的通用过滤器法(采用 IPOPT 默认设置进行比较);FL Filter 是由 Fletcher 和 Leyffer[5]提出的基本过滤器法;Ulbrich filter 是由 Ulbrich[7]在内点算法上提出的过滤器法。所有这些算法的代码都在 IPOPT 的代码基础上实现,仅采用了不同的过滤器法,并且所有方法使用同样的停止准则和停止误差 $\varepsilon_{tol}=10^{-8}$。

在试验中测试了 CUTEr 测试集合中规模小于 1000 的 571 个测试问题中,成功率方面,三重目标过滤器法成功求解了所有问题中的 522 个,IPOPT 和 FLFilter 方法成功求解了所有问题中的 510 个,Ulbrich Filter 成功求解了 400 个。

上述测试标准下,3 种不同测试项下的结果分别如图 1、图 2 和图 3 所示。图 1 和图 2 显示在迭代次数和函数估计次数方面,Triple-objective Filter 表现出了优异的性能。图 3 显示在运行时间方面,Triple-objective Filter 略优于 IPOPT 方法,因为在过滤器法计算过程中,虽然在迭代次数方面 Triple-objective Filter 优于 IPOPT 方法,但在计算过滤时,Triple-objective Filter 方法需要比 IPOPT 方法多计算一项辅助性条件,因此在迭代次数差距不大的情况下,计算时间略多于 IPOPT 方法。

采用三重目标过滤器法可以使得算法得到更长的搜索步长。如果采用基本过滤器法,步长可能会变短,因为它可能仅减小辅助性条件而不减小稳定性条件和可行性条件。图1

显示了三重目标过滤器法在迭代次数上优于其他 3 种基本过滤器法,这表明在选取试探步长时,采用更大的步长将会更快的达到问题收敛。

三重目标过滤器法不仅有利于算法的步长搜索,而且在改进算法步长的同时,由于它比其他方法更多地考虑了内点算法的特点,并增加了辅助性标准,因此它在全局收敛性方面也优于基本过滤器法。在所有测试问题中,三重目标过滤器法相对于其他方法成功求解了更多的问题,这表明了该方法有利于算法的收敛性。

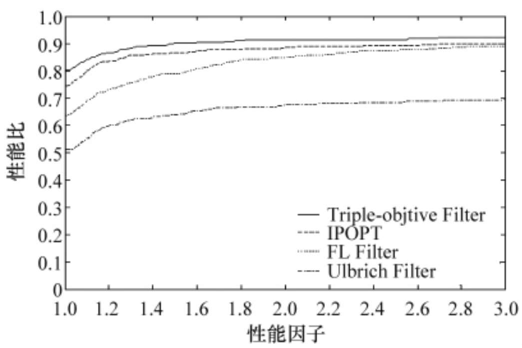

图 1　过滤器法对于迭代次数评判标准的比较

Fig. 1　Comparison of filter method with the iteration standard

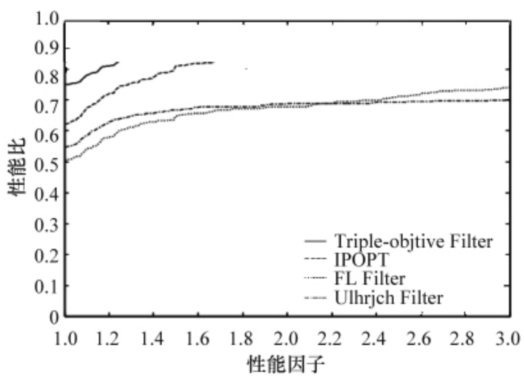

图 2　过滤器法对于函数估计次数评判标准的比较

Fig. 2　Comparison of filter method with the function estimation frequency standard

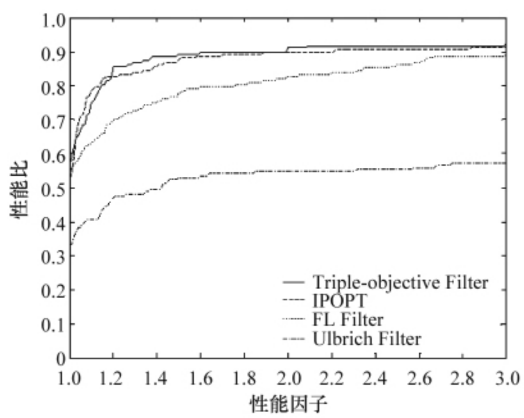

图 3　过滤器法对于运行时间评判标准的比较

Fig. 3　Comparison of filter method with the run time standard

5　结论

本文在基本过滤器法的基础上, 根据内点算法的特点, 将约束优化问题看做三目标优化问题,提出了一种新的基于内点算法的三重目标过滤器方法。该方法的本质是内点算法的 KKT 条件,从 KKT 条件出发,将可行性,稳定性和辅助性作为选择步长的目标。由于该方法相对于基本过滤方法在步长选择上更宽松,因此可以得到更长的步长,并且由于其根据内点算法的特点,在得到更长的步长的同时,也能够更快地收敛,获得更好的收敛性。仿真结果表明,基于内点算法的三重目标过滤器方法在处理大规模非线性问题方面优于基本过滤器法,具有良好的收敛性、鲁棒性和有效性。

参考文献(References)

[1] Benson H Y, Shanno D F, Vanderbei R J. Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions[J]. Computational Optimization and Applications, 2002, 23(2): 257–272.

[2] Wachter A G, Biegler L T G. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming[J]. Mathematical Programming, 2006, 106(1): 25–57.

[3] 谢亮. 基于内点理论最优潮流的算法及应用研究 [D]. 上海: 上海交通大学, 2011.
Xie Lang. Research on the algorithms and applications of optimal power flow based on interior point theory[D]. Shanghai: Shanghai Jiao Tong University, 2011.

[4] Wachter A. An interior point algorithm for large-scale nonlinear optimization with applications in process engineering[D]Pittsburgh, PE: Carnegie Mellon University, 2002.

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

[6] Dolan E D, More J J. Benchmarking optimization software with performance profiles[J]. Mathematical Programming, 2002, 91: 201–213.

[7] Ulbrich M, Ulbrich S, Vicente L N. A global convergent primal-dual interior filter method for nonconvex nonlinear programming [J]. Mathematical Programming, 2004, 100: 379–410.

[8] 王祝君, 朱德通. 一簇非线性等式约束优化问题的过滤线搜索修正正割方法[J]. 应用数学学报, 2012, 35(3): 483–502.
Wang Zhujun, Zhu Detong. Journal of Acta Mathematicae Applicatae Sinica, 2012, 35(3): 483–502.

[9] 张继明, 于宪伟. 一种改进的非线性优化问题的进化规划算法 [J]. 池州学院学报, 2010(6): 4–5.
Zhang Jiming, Yu Xianwei. Journal of Chizhou Teachers College, 2012 (6): 4–5.

[10] 孙英云, 何光宇, 梅生伟. 基于 Filter 集合的内点最优潮流新算法[J]. 电工电能新技术, 2007(2): 29–33.
SunYingyun, He Guangyu, Mei Shengwei. Journal of Advanced Technology of Electrical Engineering and Energy, 2007(2): 29–33.

(责任编辑　刘志远)