第11卷 第2期 数学研究与评论 Vol.11 No.2
1991年5月 JOURNAL OF MATHEMATICAL RESEARCH AND EXPOSITION May, 1991

基于目标流线的曲线方向寻优*

隋允康 覃新川 王希诚
（大连理工大学工程力学研究所）

摘 要

本文对于无约束问题提出了沿曲线寻优的思想，推导了确定寻优曲线的常微分方程组。对该常微分方程组研制了近似解析和数值的实用算法．借助对偶规划，这一方法由无约束问题推广到约束问题．

一、引 言

迄今为止，数学规划论研究的寻优途径为：

$$X^{(k+1)}=X^{(k)}+a^{(k)}p^{(k)} \tag{1}$$

式中新设计点$X^{(k+1)}$系旧设计点$X^{(k)}$沿方向$p^{(k)}$走了若干远，步长$a^{(k)}$由一维搜索确定．

各种优化方法的不同在于确定一维搜索方向$p^{(k)}$的作法各异．然而，方法千变万化，都是在直线方向上进行一维搜索，如图1中$\overrightarrow{AB}$和$\overrightarrow{AC}$皆可作为方向$p^{(k)}$．而$\overrightarrow{OA}$为$X^{(k)}$，则$\overrightarrow{OB}$或$\overrightarrow{OC}$为$X^{(k+1)}$．能否沿曲线方向（例如图1中由A到C的与目标等值线皆正交的曲线AC）进行一维搜索呢？如果能通过一种方法借助不多的计算量得到一条与目标等值线正交或近似正交的曲线（亦即我们所说的流线）

$$x_i=x_i(s)\quad(i=1,2,\cdots,n) \tag{2}$$

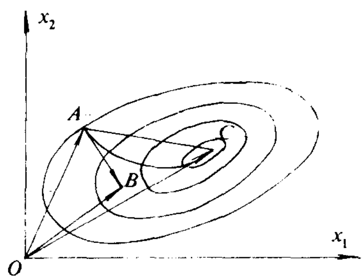

图 1

式中s为流线的弧长坐标，其坐标原点可以取在某一初始设计点X^0上，那么，我们就可以在该曲线上进行一维搜索了．这将为数学规划提供一条新颖的寻优途径．此时，(1)式依然成立，不过这里的$p^{(k)}$不再是一个常矢量，而是变矢量$p^{(k)}(s)$：

$$p^{(k)T}(s)=\{x_1(s),\cdots,x_n(s)\} \tag{3}$$

二、无约束问题的曲线寻优总体策略

设无约束优化问题为

$$\left.\begin{array}{ll}\text{求} & X\in E^n\\ \text{使} & f(X)\to\min\end{array}\right\} \tag{4}$$

* 1989年6月7月收到．

这里提出的曲线寻优策略是：1．求出过任意给定初始设计点的流线； 2．在流线上进行一维搜索．

显然，上述流线是与目标函数等值超曲面正交的一族法曲线，其中通过初始点的流线是这里待求的．设流线族的方程如（2）所示，则不难得到：

$$-\frac{dx_1}{ds}\Big/\frac{\partial f(X)}{\partial x_1}=\cdots=-\frac{dx_n}{ds}\Big/\frac{\partial f(X)}{\partial x_n} \tag{5}$$

设

$$-\frac{dx_i}{ds}\Big/\frac{\partial f(X)}{\partial x_i}=k \tag{6}$$

则

$$k=\left[\sum_{i=1}^{n}\left(\frac{dx_i}{ds}\right)^2\right]^{1/2}\Big/\left[\sum_{i=1}^{n}\left(\frac{\partial f(X)}{\partial x_i}\right)^2\right]^{1/2}=\frac{1}{\|\nabla f\|} \tag{7}$$

于是

$$\frac{dx_i}{ds}=-\frac{\partial f(X)}{\partial x_i}\Big/\|\nabla f\| \qquad (i=1,2,\cdots,n) \tag{8}$$

解常微分方程组（8）可得流线族 $x_i=x_i(s)$，由初始条件 $X|_{s=0}=X^0$， 可以流线族中定出过初始点的一条流线．将此流线代入目标函数中进行一维搜索：

求 s

使 $f(x_1(s),\cdots,x_n(s))\to\min$． （9）

如果能求出严格的流线，那么，一维搜索一次就可求出一个局部最优解．正如下文所示，我们通常只能求出近似的目标流线，于是，求流线和一维搜索只能反复交替迭代求解了．

三、寻优曲线的实用解法

通常难以求出（8）式表达的解析解，为此我们提出下列三种解法．

1．幂级数近似曲线解法

采用Taylor展开式近似表示目标流线：

$$x_i(s)=x_i^0+\sum_{k=1}^{m}\frac{d^kx_i}{ds^k}\Big|_{s=0}\frac{s^k}{k!} \tag{10}$$

于是，问题就转化为求 $\frac{d^kx_i}{ds^k}\Big|_{s=0}$ 的问题．事实上，可以在（8）式的基础上逐阶求导即可．

具体做法如下．为表达简便，可将（8）式简记为

$$\frac{dx_i}{ds}=F_{i1}(X) \tag{8a}$$

由此进一步求导

$$\frac{d^2x_i}{ds^2}=\sum_{j=1}^{n}\frac{\partial F_{i1}(X)}{\partial x_j}F_{j1}(X)=F_{i2}(X) \tag{8b}$$

归纳可得

$$\frac{d^kx_i}{ds^k}=F_{ix}(X) \tag{8c}$$

又因 $s=0$ 时 $X=X^0$，代入（8a）－（8c）可得 $s=0$ 时各阶导数值，于是（10）式的Taylor展式就方便地得到了．

2．近似常系数微分方程组解法

将（8）式右端按Taylor展开式取一阶近似，可以得到如下常系数微分方程组

$$\frac{dx_i}{ds}=a_{i0}+\sum_{k=1}^{n}a_{ik}x_k \tag{11}$$

其中：
$$a_{i0}=\left(\frac{\partial f}{\partial x_i}\right)^0/\|\nabla f\|^0-\sum_{k=1}^{n}a_{ik}x_k^0$$

$$a_{ik}=\left(\frac{\partial^2 f}{\partial x_i\partial x_k}\right)^0/\|\nabla f\|^0-\left(\frac{\partial f}{\partial x_i}\right)^0\sum_{l=1}^{n}\left(\frac{\partial f}{\partial x_l}\right)^0\left(\frac{\partial^2 f}{\partial x_k\partial x_l}\right)^0/(\|\nabla f\|^0)^3$$

这里的上标 ^{0} 表示在初始点 x^0 处的有关值.

解方程组（11）即可得用基函数 $y_k(s)\in E^n$（$k=1,\cdots,n$）和非齐次特解表达的解 $x_i=x_i(s)$（$i=1,\cdots,n$）即：

$$x_i(s)=\sum_{k=1}^{n}a_k y_k(s)-A^{-1}a^0\quad(i=1,\cdots,n) \tag{12}$$

其中 A 与 a^0 的元素分别为 a_{ik} 与 a_{i0}，利用初始条件 $x_i(s)=x_i^0$；可以定出待定系数 $a_k(i=1,\cdots,n)$.

3．数值解法

前述两种方法皆为求目标流线的近似显式，是属于解析方法，也可以避开求目标流线而采用数值方法求解.

可以采用数值方法的原因在于：在目标流线上进行一维搜索的计算机实现总是在有限个离散点上进行的，只要给定了 s^0，就能求出相应的 $X^0\in E^n$，不一定非得求出流线的函数关系式（2）不可，而 s^0 与 X^0 的映射关系是由（8）式决定的.

由 s^0 求 X^0 是一阶微分方程组的数值解法，无论采用改进的欧拉方法，龙格－库塔方法，还是阿达姆斯方法皆可以. 为行文简便，把上述数值解法用 $s^0\mapsto X^0$ 的映射符号表示之.

数值求解的算法为：

1° $s=0$ 对应于 X^0；

2° 进行一维搜索：$f(x_1(s),\cdots,x_n(s))\to\min$，时要用到 $s\mapsto x$ 的映射算法.

一维搜索可采用多种途径，用目标函数值寻优，一阶导数寻优，或一阶与二阶导数寻优皆可以.

下面提供的一阶算法与二阶算法将用到的公式.

在目标流线上，目标函数是弧长 s 的一元函数，即：

$$f(s)=f(x_1(s)\cdots,x_n(s)) \tag{13}$$

由此
$$\frac{df(s)}{ds}=\|\nabla f\| \tag{14}$$

$$\frac{d^2f(s)}{ds^2}=\sum_{i=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)\left(\frac{\partial f}{\partial x_k}\right)\left(\frac{\partial^2 f}{\partial x_i\partial x_k}\right)/\|\nabla f\|^2 \tag{15}$$

根据（14）式，可以利用一阶算法中割线性进一一维搜索：

$$s=[f'(s^{(1)})s^{(0)}-f'(s^{(0)})s^{(1)}]/[f'(s^{(1)})-f'(s^{(0)})] \tag{16}$$

根据（14）、（15）两式，可以利用二阶算式中的 Newton－Raphson方法进行一维搜索

$$s=s^{(0)}-f'(s^{(0)})/f''(s^{(0)}) \tag{17}$$

反复迭代（16）或（17）式可得到最优解 s^*. 相应的 x^* 也就找到了.

值得指出的是：在进行一维搜索时，因为要按（14）、（15）式计算 $f'(s^{(k)})$ 和 $f''(s^{(k)})$，须知点 $X^{(k)}$，所以 $s^{(k)}\mapsto X^{(k)}$ 的数值解法须多次进行，并不是最后只算一次 $s^*\mapsto X^*$.

四、约束问题的曲线寻优方法

可采用SUMT法将有约束优化问题化为序列无约束优化问题进行处理，然后采用前述方法求解．这里，我们建议采用对偶规划求解．

$$\begin{cases} 求 & X\in E^n \\ 使 & f(X)\to \min \\ \text{s.t} & h_j(X)\leqq 0 \quad (j=1,2,\cdots,m) \\ & X\in C \end{cases} \tag{18}$$

其中 $C=\{X\,|\,\underline{x_i}\leqq x_i\leqq \overline{x_i}\ ,\ i=1,2,\cdots,n\}$

（18）式对应的拉格朗日增广函数为

$$L(X,\lambda)=f(X)+\sum_{j=1}^{m}\lambda_j h_j(X),\quad X\in C \tag{19}$$

Falk定义的对偶问题为[8]

$$\varphi(\lambda)\to \max,\quad \lambda\geqq 0 \tag{20}$$

其中

$$\varphi(\lambda)=\min_x L(X,\lambda),\quad X\in C \tag{21}$$

对于凸规划，（18）、（21）式的最优解相同，即

$$\min f(X)=\max\varphi(\lambda)$$

据（18）式由Kuhn－Tucker条件

$$\left.\frac{\partial f(X)}{\partial x_i}\right|_{X=X^*}+\sum_{j=1}^{m}\lambda_j\left.\frac{\partial h_j(X)}{\partial x_i}\right|_{X=X^*}\begin{cases}\leqq 0 & x_i^*=\overline{x_i}\\ =0 & \underline{x_i}<x_i^*<\overline{x_i}\\ \geqq 0 & x_i^*=\underline{x_i}\end{cases} \tag{22}$$

（22）式表示：x^*与λ存在函数关系

$$X^*=X^*(\lambda) \tag{23}$$

将（23）式代入（21）式中得

$$\varphi(\lambda)=L(X^*(\lambda),\lambda) \tag{24}$$

对（24）式求偏导数，并利用（22）式

$$\frac{\partial\varphi(\lambda)}{\partial\lambda_j}=h_j(X^*(\lambda)) \tag{25}$$

利用（25）式，可以在对偶空间的目标流线上进行一维搜索，设对偶空间的目标$\varphi(\lambda)$的流线弧长变量为t，问题是在$\lambda=\lambda(t)$目标流线上寻优：

$$\varphi(\lambda(t))\to \max,\quad \lambda\geqq 0 \tag{26}$$

采用只用一阶信息的割线法，则需求出$\varphi(\lambda(t))$的导数：

$$\frac{d\varphi}{dt}=\|\nabla\varphi\|$$

具体算法是：选初始点$t^{(0)}$，$t^{(1)}$．则t可照搬（16）式解之．

$$t=(\varphi'(t^{(1)})t^{(0)}-\varphi'(t^{(0)})t^{(1)})/(\varphi'(t^{(1)})-\varphi'(t^{(0)})) \tag{28}$$

在搜索过程中，涉及到映射量时解$t^0\mapsto\lambda^0\mapsto X^0$，因

$$\frac{d\lambda_j}{dt}=\frac{\partial\varphi}{\partial\lambda_j}\Big/\|\nabla\varphi\|=h_j(X^*(\lambda))\Big/\Big[\sum_{j=1}^{m}h_j^2(X^*(\lambda))\Big]^{\frac{1}{2}} \tag{29}$$

如前按(29)式由t^0求出λ^0是微分方程组的数值解法，而$\lambda^0 \mapsto X^0$则利用(23)式.由于有$\lambda \geqq 0$ 的约束，所以若算出$\lambda < 0$，则令$\lambda = 0$ 即可.

有些对偶目标的二阶导数易于求出显式，也可以采用Newton-Raphson方法求解对偶目标函数的一维搜索问题.

也可以用求解析解的方法解(29)式，得到近似解$\lambda_j = \lambda_j(t)$，$(j = 1, 2, \cdots, n)$避免数值解法.

五、算　例

作为本文工作的一个总结，我们对于无约束优化问题的幂级数近似曲线解法编制了计算程序，程序是在VAX-Ⅱ超级微型机上用PASCAL语言实现的. 在程序中,我们考虑了三种情况，即在(10)式中 分别取$m = 1, 2, 3$ 来近似表示流线方程，当$m = 1$时,实质上就是一般的最速下降法，我们把它作为算法比较的标准.

例 1　　$(x_2 - x_1^2)^2 + (1 - x_1)^2 \to \min$

理论最优点为$x_1^* = 1.0$，$x_2^* = 1.0$；函数值收敛标准为10^{-5}，导数值收敛标准为10^{-4}；由初始点$x_1^0 = -1.2$，$x_2^0 = 1.0$,出发寻优，一阶方法(最速下降法)经16次迭代得最优点，二阶方法经10次迭代、三阶方法经 8 次迭代分别得最优点.

例 2　　$100(x_3 - (x_1 + x_2)^2/4)^2 + (1 - x_1)^2 + (1 - x_2)^2 \to \min$

理论最优点为$x_1^* = x_2^* = x_3^* = 1.0$；函数值收敛标准为$10^{-5}$，导数值收敛标准为$10^{-2}$，由初始点$x_1^0 = -1.2$，$x_2^0 = 2.0$，$x_3^0 = 0.0$ 出发寻优，一阶、二阶、三阶方法分别经78次、41次和36次迭代后收敛到最优点.

例 3　　$(x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \to \min$

理论最优点为$x_1^* = x_2^* = x_3^* = x_4^* = 0.0$，函数值收敛标准为$10^{-4}$，导数值收敛标准为$10^{-2}$；由初始点$x_1^0 = -3.0$，$x_2^0 = -1.0$，$x_3^0 = 0.0$，$x_4^0 = 1.0$出发寻优，一阶、二阶、三阶方法分别经84次、45次、43次迭代后收敛到最优点.

已有的工作表明曲线寻优是很有意义的，然而大量的工作还有待于进一步开拓.

参 考 文 献

[1] D. M. 希梅尔布劳. 实用非线性规划. 科学出版社. 1979.6.
[2] 詹汉生. 微分流形导引. 北京大学出版社. 1987. 11.
[3] C. Fleury, Structural Weight Optimization by Dual Method of Convex Programming, Int. J. Nm. Eng., Vol. 14, 1761—1783 (1979).

An Optimization Method of Curve Direction Based on Stream-Line of Objective Function

Sui Yunkang, Qin Xinchuan, Wang Xicheng
(Research Institute of Engineering Mehanics, Dalian University of Technology)

Abstract

An idea of searching along curve lines for the optimum point of an unconstrained problem is proposed and ordinary differential equations determining search curves are derived in this paper. Both approximately analytical and numerical solutions of the ordinary differential equations are developed to be used in practical algorithms. Then our method is extended from unconstrained problems to constrained ones by means of dual programming.