DOI: 10.13656/j.cnki.gxkx.2005.02.002
广西科学 Guangxi Sciences 2005, 12(2): 85~88

# 无约束优化中新的 Rosenbrock型算法*

# A New Type of the Rosenbrock's Algorithm for Unconstrained Optimization

罗 雁[1],简 金宝[2],韩道兰[2]
Luo Yan[1], Jian Jinbao[2], Han Daolan[2]

(1.广西钦州师范高等专科学校数学与信息科学系,广西钦州 535000; 2.广西大学数学与信息科学学院,广西南宁 530004)
(1. Dept. of Math.& Comp. Sci., Qinzhou Teachers' Coll., Qinzhou, Guangxi, 535000, China; 2. Coll. of Math.& Info. Sci., Guangxi Univ., Nanning, Guangxi, 530004, China)

摘要: 在 Rosenbrock 原始算法的基础上,提出一种新的构造正交方向的方法,并由此产生一种新的 Rosenbrock 型算法,新算法具有全局收敛性.数值试验表明,新的 Rosenbrock 型算法切实可行,且就某些算法而言要优于原始的 Rosenbrock算法.

关键词: 无约束优化 Rosenbrock型算法 Gram-Schimidt正交化
中图法分类号: O211.2 文献标识码: A 文章编号: 1005-9164(2005)02-0085-04

**Abstract** Based on the Rosenbrock's algorithm, we propose a new method of contructing orthogonal directions and producting a new type of the Rosenbrock's algorithm which has global covergence. Numerical comparision shows that the new algorithm is viable and more effective than the original one for some examples.

**Key words** unconstrained optimization, Rosenbrock's algorithm, Gram-Schimidt orthogonalization

本文考虑如下无约束问题:

$$\min f(x),$$
$$\text{s.t. } x\in E^n.$$

Rosenbrock算法是无约束优化中不使用导数的多维搜索. Rosenbrock[1]在 1960年提出的算法是沿搜索方向取离散步来实现的,后来 Davies, Swann和 Campey[2]提出了采用线搜索的算法步骤. Rosenbrock算法又称为转轴法,它设法顺着"山谷"求函数的极小点.文献[3]中构造搜索方向如下: 设 $d^1,\cdots,d^n$ 是单位正交方向,从当前迭代点 $x^k$ 开始,目标函数 $f$ 沿每个方向逐次极小化导出点 $x^{k+1}$.特别地,$x^{k+1}-x^k=\sum_{j=1}^{n}\lambda_j d^j$,其中 $\lambda_j$ 是沿方向 $d^j$ 移动的步长.新的一组方向 $\bar d^1,\cdots,\bar d^n$ 通过 Gram-Schimidt过程产生如下:

$$a^j=\begin{cases} d^j, & \lambda_j=0;\\ \sum_{i=j}^{n}\lambda_i d^i, & \lambda_j\neq 0,\end{cases}$$
$$b^j=\begin{cases} a^j, & j=1;\\ a^j-\sum_{i=1}^{j-1}((a^j)^T\bar d^i)\bar d^i, & j\geqslant 2,\end{cases} \qquad (0.1)$$
$$\bar d^j=\frac{b^j}{\|b^j\|}.$$

令 $D=(d^1,\cdots,d^n)$, $A=(a^1,\cdots,a^n)$, $\lambda=(\lambda_{ij})_{n\times n}$,

其中,$\lambda_{ij}=\begin{cases}1, i=j;\\ 0, i\neq j,\end{cases}\lambda_j=0;\lambda_{ij}=\begin{cases}\lambda_i, i\leqslant j;\\ 0, i>j,\end{cases}\lambda_j\neq 0,$

则有 $A=D\lambda$,而且 $a^r=\sum_{i=r}^{n}\lambda_i d^i$ 为加速方向,其中 $r$ 使得 $\lambda_r$ 是 $\lambda_1,\cdots,\lambda_n$中第一个非零的数.通过上述的构造可以看出,进行转轴的关键在于: 构造非奇异矩阵 $\lambda$,保证 $A=D\lambda$ 仍是非奇异的,而且为了加快收敛速度,应使得有一个 $a^r=x^{k+1}-x^k$.在 Roesnbrock算法中,构造的 $\lambda$ 是下三角阵.本文从构造上三角阵 $\lambda$ 出发建立,并产生新的且含有加速方向的新方法,由此产生了一种新的 Rosenbrock 型算法.通过数值试验得出此方法的可行性结论.与原始的 Rosenbrock 方法比

收稿日期: 2004-08-30
修回日期: 2004-10-11
作者简介: 罗 雁(1969-),女,广西钦州市人,讲师,主要从事数学分析与最优化基础.
* 国家自然科学基金(10261001)和广西科学基金(0236001, 0249003)联合资助项目.

较表明,在某些情况下要优于原始的 Rosenbrock型算法.

## 1 新的 Rosenbrock型算法

受 Rosenbrock算法关于正交方向构造技术的启发,本文给出了另一种构造方法:

$$a^j=\begin{cases} d^j, & \lambda_j=0;\\ \sum_{i=1}^{j}\lambda_i d^i, & \lambda_j\neq 0,\end{cases}$$

$$b^j=\begin{cases} a^j, & j=n;\\ a^j-\sum_{i=j+1}^{n}((a^j)^T\bar d^i)\bar d^i, & j\leqslant n-1,\end{cases} \qquad (1.1)$$

$$\bar d^j=\frac{b^j}{\|b^j\|}.$$

下面给出此方法构造新方向的重要性质.

定理 **1** 假设向量 $d^1,\cdots,d^n$ 是线性无关和彼此正交的,则对任意一组 $\lambda_1,\lambda_2,\cdots,\lambda_n$,由 (1.1) 定义的方法有如下性质:

(1) 向量 $a^1,\cdots,a^n$ 线性无关,且 $a^t=x^{k+1}-x^k$,即 $a^t$ 是加速方向,其中 $t$ 使得 $\lambda_t$ 是 $\lambda_1,\cdots,\lambda_n$ 中最后一个非零的数.

(2) $\bar d^1,\cdots,\bar d^n$ 是线性无关且正交的单位向量,如果 $\lambda_j=0$,则 $\bar d^j=d^j$.

证明 (1) 首先证明 $a^1,\cdots,a^n$ 线性无关.

由 (1.1) 式可知,

$$a^j=\begin{cases} d^j=(d^1,\cdots,d^n)(0,\cdots,0,1(jth),0,\cdots,0)^T, & \lambda_j=0;\\ \sum_{i=1}^{j}\lambda_i d^i=(d^1,\cdots,d^n)(\lambda_1,\cdots,\lambda_j,0,\cdots,0)^T, & \lambda_j\neq 0.\end{cases}$$

令 $D=(d^1,\cdots,d^n)$, $A=(a^1,\cdots,a^n)$, $\lambda'=(\lambda'_{ij})_{n\times n}$,其中,

$$\lambda'_{ij}=\begin{cases} 1, i=j;\ \lambda_j=0;\\ 0, i\neq j,\end{cases}\qquad \lambda'_{ij}=\begin{cases} \lambda_i, i\leqslant j;\ \lambda_j\neq 0,\\ 0, i>j,\end{cases}$$

则 $A=D\lambda'$. 由于 $|A|=|D||\lambda'|$,且 $|D|\neq 0$, $|\lambda'|=\lambda'_{11}\lambda'_{22},\cdots,\lambda'_{nn}\neq 0$. 故 $|A|\neq 0$,即 $a^1,\cdots,a^n$ 线性无关.

设 $\lambda_t$ 是 $\lambda_1,\cdots,\lambda_n$ 中最后一个非零数,则有

$$a^t=\sum_{i=1}^{t}\lambda_i d^i=\sum_{i=1}^{n}\lambda_i d^i=x^{k+1}-x^k.$$

(2) 下面用归纳法证明 $b^1,\cdots,b^n$ 也线性无关.

由于 $b^n=d^n\neq 0$ 显然成立,只需证明,如果 $b^n,\cdots,b^k$ 线性无关,则 $b^n,\cdots,b^k,b^{k-1}$ 也线性无关.

假设 $\sum_{j=k-1}^{n}T_j b^j=0$,由 (1.1) 式中 $b^{k-1}$ 的定义,有

$$0=\sum_{j=k}^{n}T_j b^j+T_{k-1}b^{k-1}=\sum_{j=k}^{n}\left[T_j-\frac{T_{k-1}((a^{k-1})^T\bar d^j)}{\|b^j\|}\right]b^j+T_{k-1}a^{k-1}. \qquad (1.2)$$

由 (1.1) 式知,每个向量 $b^j$ 是 $a^j,\cdots,a^n$ 的线性组合,又由于 $a^{k-1},a^k,\cdots,a^n$ 线性无关,由 (1.2) 式知, $T_{k-1}=0$. 又由归纳假设知, $b^n,\cdots,b^k$ 线性无关,从而由 (1.2) 式有

$$T_j-\frac{T_{k-1}((a^{k-1})^T\bar d^j)}{\|b^j\|}=0, j=k,\cdots,n.$$

由 $T_{k-1}=0$ 知, $T_j=0, j=k,\cdots,n$,故 $b^{k-1},\cdots,b^n$ 线性无关. 又由 $\bar d^j$ 的定义立知, $\bar d^1,\cdots,\bar d^n$ 也线性无关.

现在证明 $b^1,\cdots,b^n$ 的正交性,即 $\bar d^1,\cdots,\bar d^n$ 的正交性. 由 (1.1) 式可知,

$$(b^n)^Tb^{n-1}=(d^n)^T[a^{n-1}-(a^{n-1})^T\bar d^n\bar d^n]=(a^n)^T[a^{n-1}-(d^{n-1})^T\frac{a^n}{\|d^n\|^2}d^n]=0.$$

因此只需证明,若 $b^n,\cdots,b^k$ 相互正交,则 $b^n,\cdots,b^k,b^{k-1}$ 也相互正交. 由归纳假设可知, $(b^j)^T\bar d^j=0, i\neq j, i,j=n,n-1,\cdots,k$,于是由 (1.2) 式有

$$(b^j)^Tb^{k-1}=(b^j)^T[d^{k-1}-\sum_{i=k}^{n}[(a^{k-1})^T\bar d^i]\bar d^i]=(b^j)^Ta^{k-1}-((a^{k-1})^T\bar d^j)(b^j)^T\bar d^j=0, j=k,\cdots,n. \qquad (1.3)$$

于是, $b^n,\cdots,b^k,b^{k-1}$ 相互正交.

最后证明如果 $\lambda_j=0$,有 $\bar d^j=d^j$. 由 (1.1) 式知,如 $\lambda_j=0$,有

$$b^j=d^j-\sum_{i=j+1}^{n}\frac{1}{\|b^i\|}((d^j)^Tb^i)\bar d^j. \qquad (1.4)$$

注意到 $b^j$ 是 $a^j,\cdots,a^n$ 的线性组合,可设 $b^j=\sum_{r=j}^{n}U_{jr}d^r$. 由 (1.1) 式有

$$b^j=\sum_{r\in R}U_{jr}d^r+\sum_{r\in\bar R}U_{jr}(\sum_{s=1}^{r}\lambda_s d^s), \qquad (1.5)$$

其中, $R=\{r: r\geqslant j,\lambda_r=0\}$, $\bar R=\{r: r\geqslant j,\lambda_r\neq 0\}$. 注意到 (1.4) 及 (1.5) 式中的指标 $i,j,r$ 满足 $j+1\leqslant i\leqslant r$,故由 $d^1,\cdots,d^n$ 的正交性有

$$(d^j)^Td^r=0, r\in R;\ (d^j)^T(\sum_{s=1}^{r}\lambda_s d^s)=\lambda_j(d^j)^Td^j=\lambda_j=0, r\in\bar R.$$

于是用 $(d^j)^T$ 乘 (1.5),得 $(d^j)^Tb^j=0, i=j+1,\cdots,n$. 故由 (1.4) 式立知 $b^j=d^j$,进而, $\bar d^j=d^j$. 定理证毕.

由定理 1可知,如果 $\lambda_j=0$,那么新方向 $\bar d^j$ 等于旧方向 $d^j$. 因此,只需对那些 $\lambda_j\neq 0$的指标计算新方向.

下面给出新 Rosenbrock型算法的算法步骤.

算法 **1**(线搜索算法)

步骤 **0** 选取 $x^1\in E^n, d^1,d^2,\cdots,d^n$ 为坐标方向,

充分小的终止精度 $\varepsilon>0$, $y^1=x^1$, $j=1$.

步骤 **1** 计算 $\min\{f(y^j+\lambda d^j),\lambda\in E^1\}$的最优解 $\lambda_j$,令 $y^{j+1}=y^j+\lambda_j d^j$.若 $j<n$,令 $j:=j+1$,重复步骤 1;否则,转入步骤 2.

步骤 **2** 令 $x^{k+1}=y^{n+1}$,如果 $\|x^{k+1}-x^k\|<\varepsilon$,停;否则,令 $y^1=x^{k+1}$, $j=1$.

步骤 **3** 按(1.1)式计算新方向 $\bar{d}^1,\cdots,\bar{d}^n$,令 $k:=k+1$,返回步骤 1.

算法 **2**(离散步算法)

步骤 **0** 选取 $x^1\in E^n$,参数 $\varepsilon>0$, $\tau>1$, $\theta\in(-1,0)$, $d^1,d^2,\cdots,d^n$ 为坐标方向,$\bar{\Delta}_1,\cdots,\bar{\Delta}_n$ 为沿这些方向的初始步长.令 $y^1=x^1$, $y^j=y^1$, $\Delta_j=\bar{\Delta}_j$, $k=j=1$.

步骤 **1** 如果 $f(y^j+\Delta_j d^j)<f(y^j)$,令 $y^{j+1}=y^j+\Delta_j d^j$, $\Delta_j=\tau\Delta_j$;如果 $f(y^j+\Delta_j d^j)\geqslant f(y^j)$,令 $y^{j+1}=y^j$, $\Delta_j=\theta\Delta_j$.若 $j<n$,令 $j:=j+1$重复步骤 1;否则,转入步骤 2.

步骤 **2** 如果 $f(y^{n+1})<f(y^1)$,令 $y^1=y^{n+1}$, $j=1$,重复步骤 1.如果 $f(y^{n+1})=f(y^1)$,若 $f(y^{n+1})<f(x^k)$,转入步骤 3;若 $f(y^{n+1})=f(x^k)$,如果 $|\Delta_j|\leqslant\varepsilon$,停;否则,令 $y^1=y^{n+1}$, $j=1$,转入步骤 1.

步骤 **3** 令 $x^{k+1}=y^{n+1}$,如果 $\|x^{k+1}-x^k\|<\varepsilon$,停;否则,由 $x^{k+1}-x^k=\sum_{j=1}^{n}\lambda_j d^j$ 计算 $\lambda_1,\cdots,\lambda_n$,按(1.1)式计算新方向 $\bar{d}^1,\cdots,\bar{d}^n$,令 $\Delta_j=\bar{\Delta}_j$, $y^1=x^{k+1}$, $k:=k+1$,返回步骤 1.

下面给出了算法 1在放宽假设的条件下,可以收敛到稳定点.

定理 **2** 设 $f$ 是可微函数,对于极小化问题 $\min\{f(x)\mid x\in E^n\}$,考虑如下定义的算法映射 $A$,向量 $y\in A(x)$ 意味着 $y$ 是由 $x$ 开始逐次沿单位正交方向 $d^1,\cdots,d^n$ 极小化 $f$ 而得到的.现假设下面的性质成立:

(1) $f$ 沿 $E^n$ 中任何方向的极小点唯一;

(2)初始点为 $x^1$,由算法 1产生的点列 $\{x^k\}$ 有界,则点列 $\{x^k\}$ 的任何聚点 $x$ 必满足 $f(x)=0$.

有关证明可参见文献[4]的证明方法.

## 2 对新的 Rosenbrock型算法的补充

上面构造的新方向中的确包含加速方向 $x^{k+1}-x^k$,但每次迭代中的加速方向的排序(序号)不断变化,如果希望最后一个方向 $a^n$ 为加速方向,可作如下处理:记 $\lambda_r$ 为 $\lambda_1,\cdots,\lambda_n$ 中第一个不为 0的数,将 $(\lambda_r,d^r)$ 与 $(\lambda_n,d^n)$ 交换:

$$\tilde{d}^j=d^j,\ j\neq r,n;\ \tilde{d}^r=d^n,\ \tilde{d}^n=d^r.$$

$$\tilde{\lambda}_j=\lambda_j,\ j\neq r,n;\ \tilde{\lambda}_r=\lambda_n,\ \tilde{\lambda}_n=\lambda_r.$$

则满足:(1) $x^{k+1}-x^k=\sum_{j=1}^{n}\tilde{\lambda}_j\tilde{d}^j$, $\tilde{\lambda}_n\neq 0$.(2) $\{\tilde{d}^1,\cdots,\tilde{d}^n\}$ 非零正交.

构造
$$a^j=\begin{cases}\tilde{d}^j, & \tilde{\lambda}_j=0;\\ \sum_{i=1}^{j}\tilde{\lambda}_i\tilde{d}^i, & \tilde{\lambda}_j\neq 0,\end{cases}$$

则 $\boldsymbol{A}=\tilde{\boldsymbol{D}}\tilde{\Lambda}$,其中 $\tilde{\Lambda}$ 是上三角阵, $\tilde{\boldsymbol{D}}=(\tilde{d}^1,\cdots,\tilde{d}^n)$, $\boldsymbol{A}=(a^1,\cdots,a^n)$,而且 $a^n=x^{k+1}-x^k$ 是加速方向.此外,如果希望 $a^i$ 为加速方向, $a^i=x^{k+1}-x^k$,可将 $a^i$ 与 $a^n$ 交换,如希望 $a^1$ 为加速方向,即 $a^1=x^{k+1}-x^k$,可将 $a^1$ 与 $a^n$ 交换.

采用这样的交换技术,可以使我们更清楚 $f$ 是沿着怎样的山谷下降的,对函数的下降过程更容易得到直观的判断,就某些算例而言,收敛可能更快些.在本文算例中,只讨论 $a^t$ 为加速方向的一般情况,其中 $t$ 是使得 $\lambda_t$ 为 $\lambda_1,\cdots,\lambda_n$ 中最后一个不为 0的数.

## 3 数值试验

本文对算法进行了比较性的数值试验,结果表明新 Rosenbrock型算法确实可行,而且就某些算例而言要优于原始的 Rosenbrock算法.本文从中选出 2个简单的例子加以说明,并略去其详细迭代过程.在下面 2个算例中,均选取坐标方向为初始单位正交方向.

例 **1** 取自于文献[4]第 200页.

$$\min f(x)=\left(\sum_{i=1}^{10}i^3(x_i-1)^2\right)^3.$$

例 **2** 取自于 Discrete boundary value function.

$$\min f(x)=uu^T.$$

其中, $u_1=2x_1-x_2+\frac{1}{2}h^2(x_1+t_1+1)^3$;

$u_i=2x_i-x_{i-1}+\frac{1}{2}h^2(x_i+t_i+1)^3$, $i=2,3,\cdots,12$;

$u_{13}=2x_{13}-x_{12}+\frac{1}{2}h^2(x_{13}+t_{13}+1)^3$;

$h=\frac{1}{14}$, $t_i=ih$, $i=1,2,\cdots,13$.

算法 1和算法 2及原始算法求解以上 2个问题的运算结果如表 1和表 2所示.

表 1 采用线搜索算法的比较结果

**Table 1 Numerical results on the algorithm using line searthes**

| 例题 Example | 参数 Parameter | 初始点 Initial point | 算法 Algorithm | 迭代次数 $k$ Iterations $k$ | 最优值 Optimal value |
|---|---|---|---|---|---|
| 例 1 Example 1 | X= 0.01 | $x^1$= (0.5, 0.5, … , 0.5) | 算法 1 Algorithm 1 | 5 | 2.2940e-006 |
| | | | 原算法 Original algorithm | 12 | 2.8062e-006 |
| 例 2 Example 2 | X= 0.001 | $x^1$= $(h(h-1), \cdots, 13h(13h-1))$, $h=\frac{1}{14}$ | 算法 1 Algorithm | 4 | 3.4925e-005 |
| | | | 原算法 Original algorithm | 10 | 2.8719e-004 |

表 2 采用离散步算法的比较结果

**Table 2 Numerical results on the algorithm with discrete steps**

| 例题 Example | 参数 Parameter | 初始点 Initial point | 算法 Algorithm | 迭代次数 $k$ Iterations $k$ | 最优值 Optimal value |
|---|---|---|---|---|---|
| 例 1 Example 1 | X= 0.01 | $x^1$= (0, 0, … , 0) | 算法 2 Algorithm 2 | 3 | 2.3223e-005 |
| | T= 2.2, U= 0.2 | | 原算法 Original algorithm | 6 | 2.2295e-005 |
| 例 2 Example 2 | X= 0.001 | $x^1$= $(h(h-1), \cdots, 13h(13h-1))$, | 算法 2 Algorithm 2 | 3 | 9.3408e-028 |
| | T= 2.2, U= 0.2 | $h=\frac{1}{14}$ | 原算法 Original algorithm | 5 | 1.3069e-025 |

## 4 结束语

Rosenbrock算法是在循环坐标法的基础上提出的,比循环坐标法具有更好的收敛性,但也存在很多不足的地方,本文针对原始的 Rosenbrock 算法关于方向的构造方法,提出了一类新的构造方法和一类新的 Rosenbrock型算法,并且给出了采用线搜索和取离散步的算法步骤.数值试验表明这类新的 Rosenbrock型算法对某些问题而言要优于原始的 Rosenbrock 算法.实际上,这两类 Rosenbrock 算法是互为补充的,在实际计算中,由于这两类算法相关甚微,我们可以同时采用不同算法,并且通过比较后择优而用.

参考文献:


[1] Rosenbrock H H. An automatic method for finding the greatesst or least value of a function [J]. Computer Journal, 1960, (3): 175–184.

[2] Davies, Swann W H, Campey. Report on the development of a new direct search method of optimization [M]. London: Imperical Chemical Industries Ltd, 1964. 3.

[3] Bazaraa M S, Sherali H D, Shetty C M. Nonlinear Progr–
amming: Theory and Algorithms [M]. Wiley: John Wiley & Sons Inc, 1993. 291–300.

[4] Hock W, Schittkowski K, Test Examples for Nonlinear Programming Codes: Lecture Notes in Economics and Mathematical Systems [M]. Berlin: Springer_ Verlag, 1981.

[5] 赵凤治,席少霖.最优化计算方法 [M].上海:上海科学技术出版社, 1983. 168–177.


(责任编辑: 黎贞崇)