DOI: 10.13718/j.cnki.xdzk.2018.02.009

一类修正的 FR 型谱共轭梯度法①

王森森[1]，　张俊容[2]，　韩　信[3]

1. 和田师范专科学校 数学与信息学院，新疆 和田 848000；2. 西南大学 数学与统计学院，重庆 400715；
3. 四川文理学院 数学学院，四川 达州 635000

摘要：提出了一类 WFR 型谱共轭梯度法，并且该算法在任何线搜索下都具有充分下降性. 在标准 Wolfe 线搜索下，证明了新算法具有全局收敛性. 数值实验结果表明新算法优于 VFR 法.

关　键　词：无约束优化；谱共轭梯度法；全局收敛；Wolfe 线搜素

中图分类号：O224　　**文献标志码**：A　　**文章编号**：1673-9868(2018)02-0049-07

共轭梯度法是求解大规模非线性无约束优化问题 $\min\{f(x)\mid x\in\mathbb{R}^n\}$ 的一种十分有效的方法，这里目标函数 $f(x)$ 是一阶连续可微函数，经典共轭梯度法的一般形式为：

$$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k \tag{1}$$

$$\boldsymbol{d}_k=\begin{cases}\boldsymbol{g}_k & k=1\\ \boldsymbol{g}_k+\beta_k\boldsymbol{d}_{k-1} & k\geqslant 2\end{cases} \tag{2}$$

其中：梯度向量 $\nabla\boldsymbol{f}(\boldsymbol{x})$ 记为 $\boldsymbol{g}(\boldsymbol{x})$，$\alpha_k$ 是由某种线搜索确定的步长因子，β_k 为共轭参数. 关于 β_k 的选择有许多著名的公式[1-5]，如：

$$\beta_k^{\mathrm{HS}}=\frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{y}_{k-1}}{\boldsymbol{d}_{k-1}{}^{\mathrm{T}}\boldsymbol{y}_{k-1}}\qquad \beta_k^{\mathrm{FR}}=\frac{\|\boldsymbol{g}_k\|^2}{\|\boldsymbol{g}_{k-1}\|^2}$$
$$\beta_k^{\mathrm{PRP}}=\frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{y}_{k-1}}{\|\boldsymbol{g}_{k-1}\|^2}\qquad \beta_k^{\mathrm{DY}}=\frac{\|\boldsymbol{g}_k\|^2}{\boldsymbol{d}_{k-1}{}^{\mathrm{T}}\boldsymbol{y}_{k-1}} \tag{3}$$

其中 $\boldsymbol{y}_{k-1}=\boldsymbol{g}_k-\boldsymbol{g}_{k-1}$. 以上 4 个公式分别对应 4 种不同的共轭梯度法，并且每种方法的收敛性和数值表现不尽相同：FR 和 DY 法有很好的收敛性，而 PRP 和 HS 法有很好的数值表现. 为寻求既保证收敛性又有良好的数值表现的算法，许多学者对上述公式进行了修改[6-10]，例如文献[9] 给出了 FR 法的一个修正公式

$$\beta_k^{\mathrm{VFR}}=\frac{\|\boldsymbol{g}_k\|\ |\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_{k-1}|}{\|\boldsymbol{g}_{k-1}\|^3} \tag{4}$$

2001 年 Birgin 和 Martinez 提出了一种谱共轭梯度法[10]，其搜索方向定义如下：

$$\boldsymbol{d}_k=\begin{cases}\boldsymbol{g}_k & k=1\\ -\theta_k\boldsymbol{g}_k+\beta_k\boldsymbol{d}_{k-1} & k\geqslant 2\end{cases} \tag{5}$$

其中：$\theta_k=\dfrac{\boldsymbol{s}_{k-1}{}^{\mathrm{T}}\boldsymbol{s}_{k-1}}{\boldsymbol{s}_{k-1}{}^{\mathrm{T}}\boldsymbol{y}_{k-1}}$ 为谱系数，$\boldsymbol{s}_{k-1}=\boldsymbol{x}_k-\boldsymbol{x}_{k-1}=\alpha_{k-1}\boldsymbol{d}_{k-1}$，$\beta_k=\dfrac{(\theta_k\boldsymbol{y}_{k-1}-\boldsymbol{s}_{k-1})^{\mathrm{T}}\boldsymbol{y}_{k-1}}{\boldsymbol{d}_{k-1}{}^{\mathrm{T}}\boldsymbol{y}_{k-1}}$. 但 Birgin 和 Martinez

① 收稿日期：2016-11-14
基金项目：国家自然科学基金项目(11401487).
作者简介：王森森(1990-)，男，河南辉县人，硕士，主要从事最优化理论，算法及应用研究的研究.
通信作者：张俊容，博士，副教授.

提出的谱共轭梯度法的搜索方向 $\boldsymbol{d}_k$ 不满足下降性. 而 Lu 等构造出了具有充分下降性的 VFR 型谱共轭梯度法，其搜索方向 $\boldsymbol{d}_k$ 中 $\theta_k = \dfrac{|\boldsymbol{d}_{k-1}{}^{\mathrm{T}}\boldsymbol{g}_k| - \boldsymbol{d}_{k-1}{}^{\mathrm{T}}\boldsymbol{g}_{k-1}}{\|\boldsymbol{g}_{k-1}\|^2}$，$\beta_k = \beta_k^{\mathrm{VFR}}$. 此方法满足充分下降条件

$$\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k \leqslant -c\|\boldsymbol{g}_k\|^2 \tag{6}$$

其中 $c > 0$ 为常数.

本文受上述文献的启发，提出了一类具有充分下降性的谱共轭梯度法，参数 β_k 为

$$\beta_k^{\mathrm{WFR}} = \frac{\|\boldsymbol{g}_k\| \, |\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{g}_{k-1}|}{\|\boldsymbol{g}_{k-1}\|^3 + \mu |\boldsymbol{d}_{k-1}^{\mathrm{T}}\boldsymbol{g}_k|} \tag{7}$$

谱系数 θ_k 为

$$\theta_k = t + \beta_k^{\mathrm{WFR}} \frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_{k-1}}{\|\boldsymbol{g}_k\|^2} \tag{8}$$

其中参数满足 $\mu > 0$，$t > 0$，然后基于(7)式和(8)式给出 WFR 型谱共轭梯度法的算法框架，证明了该算法在标准的 Wolfe 线搜索下满足充分下降性和全局收敛性，并与文献[9]中的算法进行数值比较.

1　WFR 型谱共轭梯度法的算法及其性质

本节中，首先给出 WFR 型谱共轭梯度法算法 1，然后说明它所具有的一些性质.

算法 1　WFR 型谱共轭梯度法

步骤 1　给定初始点 $\boldsymbol{x}_1$ 及精度 ε，计算 $\boldsymbol{g}_1$，若 $\|\boldsymbol{g}_1\| \leqslant \varepsilon$，停止. 否则，转步骤 2.

步骤 2　计算搜索方向 $\boldsymbol{d}_k$

$$\boldsymbol{d}_k = \begin{cases} \boldsymbol{g}_k & k = 1 \\ -\theta_k\boldsymbol{g}_k + \beta_k^{\mathrm{WFR}}\boldsymbol{d}_{k-1} & k \geqslant 2 \end{cases} \tag{9}$$

$$\theta_k = t + \beta_k^{\mathrm{WFR}} \frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_{k-1}}{\|\boldsymbol{g}_k\|^2} \tag{10}$$

步骤 3　由下式计算步长因子 α_k

$$\begin{gathered} (\boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k) \leqslant f(\boldsymbol{x}_k) + \delta\alpha_k\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k \\ \boldsymbol{g}(\boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k)^{\mathrm{T}}\boldsymbol{d}_k \geqslant \sigma\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k \end{gathered} \tag{11}$$

步骤 4　迭代计算 $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k$，$\boldsymbol{g}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_{k+1})$. 若 $\|\boldsymbol{g}_{k+1}\| \leqslant \varepsilon$，则停止.

步骤 5　$k := k + 1$，转步骤 2.

命题 1　由算法 1 产生的序列$\{\boldsymbol{g}_k\}$和$\{\boldsymbol{d}_k\}$满足充分下降性，即对任意 $k \geqslant 1$，都有 $\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k < 0$.

证　由算法 1 得，

$$\begin{aligned} \boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k &= \boldsymbol{g}_k^{\mathrm{T}}(-\theta_k\boldsymbol{g}_k + \beta_k^{\mathrm{WFR}}\boldsymbol{d}_{k-1}) = \\ &-\theta_k\|\boldsymbol{g}_k\|^2 + \beta_k^{\mathrm{WFR}}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_{k-1} = \\ &-\left(t + \beta_k^{\mathrm{WFR}}\frac{\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_{k-1}}{\|\boldsymbol{g}_k\|^2}\right)\|\boldsymbol{g}_k\|^2 + \beta_k^{\mathrm{WFR}}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_{k-1} = \\ &-t\|\boldsymbol{g}_k\|^2 < 0 \end{aligned}$$

注　由算法 1 的谱系数的选取可知，可通过选取不同的参数 t 来优化算法 1 的数值效果.

2　算法收敛性

为了研究算法的收敛性，需要给出两个基本假设：

(A1) 设目标函数 $f(\boldsymbol{x})$ 在水平集 $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}_1)\}$ 上有下界，其中 $\boldsymbol{x}_1$ 为初始迭代点.

(A2) 存在 Ω 的某个邻域 Λ，使得目标函数 $f(\boldsymbol{x})$ 在该邻域上连续可微且梯度函数 $g(\boldsymbol{x})$ 满足 Lipschitz 条件，即存在常数 $L > 0$，使得

$$\| g(\boldsymbol{x})-g(\boldsymbol{y})\| \leqslant L \|\boldsymbol{x}-\boldsymbol{y}\| \qquad \forall x, y \in \Lambda \tag{12}$$

引理 1[12]　假设 A1,A2 成立，如果搜索方向 $\boldsymbol{d}_k$ 满足下降性，步长因子 α_k 由 Wolfe 非精确的一维线搜索确定，那么 $\sum \frac{(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}{\|\boldsymbol{d}_k\|^2}<\infty$.

定理 1　如果目标函数满足假设 A1,A2，则算法 1 产生的序列 $\{\boldsymbol{g}_k\}$ 有

$$\liminf_{k\to\infty}\|\boldsymbol{g}_k\|=0 \tag{13}$$

证　假设结论不成立，则存在常数 $\gamma>0$ 使得

$$\|\boldsymbol{g}_k\|\geqslant\gamma \qquad \forall k\geqslant 1 \tag{14}$$

由(9) 式可得

$$\boldsymbol{d}_k+\theta_k\boldsymbol{g}_k=\beta_k^{\mathrm{WFR}}\boldsymbol{d}_{k-1} \tag{15}$$

从而有

$$(\boldsymbol{d}_k+\theta_k\boldsymbol{g}_k)^{\mathrm{T}}(\boldsymbol{d}_k+\theta_k\boldsymbol{g}_k)=(\beta_k^{\mathrm{WFR}})^2\|\boldsymbol{d}_{k-1}\|^2 \tag{16}$$

将(16) 式展开得

$$\|\boldsymbol{d}_k\|^2+2\theta_k\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k+\theta_k{}^2\|\boldsymbol{g}_k\|^2=(\beta_k^{\mathrm{WFR}})^2\|\boldsymbol{d}_{k-1}\|^2 \tag{17}$$

移项可得

$$\|\boldsymbol{d}_k\|^2=2t\theta_k\|\boldsymbol{g}_k\|^2-\theta_k{}^2\|\boldsymbol{g}_k\|^2+(\beta_k^{\mathrm{WFR}})^2\|\boldsymbol{d}_{k-1}\|^2 \tag{18}$$

由命题 1 的证明，(21) 式可变为

$$\|\boldsymbol{d}_k\|^2=2t\theta_k\|\boldsymbol{g}_k\|^2-\theta_k{}^2\|\boldsymbol{g}_k\|^2+(\beta_k^{\mathrm{WFR}})^2\|\boldsymbol{d}_{k-1}\|^2 \tag{19}$$

上式两边同时除以 $(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)^2$ 得

$$\frac{\|\boldsymbol{d}_k\|^2}{(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}=\frac{\|\boldsymbol{d}_k\|^2}{t^2\|\boldsymbol{g}_k\|^4}=\frac{2\theta_k}{t\|\boldsymbol{g}_k\|^2}-\frac{\theta_k{}^2}{t^2\|\boldsymbol{g}_k\|^2}+\frac{(\beta_k^{\mathrm{WFR}})^2}{t^2\|\boldsymbol{g}_k\|^4}\|\boldsymbol{d}_{k-1}\|^2 \tag{20}$$

进而有

$$\begin{aligned}\frac{\|\boldsymbol{d}_k\|^2}{(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}\leqslant&\left(\frac{\|\boldsymbol{g}_k\|^2}{\|\boldsymbol{g}_{k-1}\|^2}\right)^2\frac{\|\boldsymbol{d}_{k-1}\|^2}{t^2\|\boldsymbol{g}_k\|^4}-\frac{1}{t^2\|\boldsymbol{g}_k\|^2}(\theta_k{}^2-2t\theta_k+t^2-t^2)=\\&\frac{\|\boldsymbol{d}_{k-1}\|^2}{t^2\|\boldsymbol{g}_{k-1}\|^4}+\frac{1}{\|\boldsymbol{g}_k\|^2}-\frac{(\theta_k-t)^2}{t^2\|\boldsymbol{g}_k\|^2}\leqslant\\&\frac{\|\boldsymbol{d}_{k-1}\|^2}{t^2\|\boldsymbol{g}_{k-1}\|^4}+\frac{1}{\|\boldsymbol{g}_k\|^2}\leqslant\\&\sum_{i=1}^{k}\frac{1}{\|\boldsymbol{g}_k\|^2}+\frac{1}{t^2\|\boldsymbol{g}_1\|^2}\leqslant\frac{k}{\gamma^2}+s\left(s=\frac{1}{t^2\|\boldsymbol{g}_1\|^2}\right)\end{aligned} \tag{21}$$

综上所述可得

$$\sum\frac{(\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d}_k)^2}{\|\boldsymbol{d}_k\|^2}\geqslant\gamma^2\sum\frac{1}{k+\gamma^2 s}\to+\infty \tag{22}$$

与引理 1 矛盾.

3　数值实验

为了比较算法 1 与算法 VFR 的数值表现，我们对文献[13]中的部分测试函数进行模拟实验. 实验在 PC 机上完成，PC 的配置如下：AMD A4-3300M CPU@ 1.90 GHz，2.00 GB 内存. 程序用 Matlab 编写，运行环境为 Matlab R2010a. 算法中参数的选取为：$\delta=0.001$，$\sigma=0.9$. 终止条件为：$\|\boldsymbol{g}_k\|\leqslant 10^{-6}$ 或者算法的迭代总时间超过 1 200 s. 对于算法 1 中参数 μ 和 t 的选取，选取 $\mu=0.5$ 和 $t=0.09$，将算法 1 与 VFR 共轭梯度法进行比较，实验结果见表 1 和表 2.

在表 1 中 Problem 代表函数名称，Dim 为维数，Iter 为迭代的总次数，Tcpu 为测试问题消耗 CPU 时间，NF 与 NG 分别为目标函数值及目标函数梯度值的总计算次数.

表 1 数值结果

Problem	Dim	算法 VFR				算法 1			
		Iter	Tcpu/s	NF	NG	Iter	Tcpu/s	NF	NG
power	4	99	0.058	244	495	99	0.064	244	594
arwhead	50	90	0.034	234	450	90	0.033	234	540
biggsb1	100	4 527	4.803	19 550	21 171	2 268	2.358	12 636	12 946
cosine	1 000	82	0.174	214	410	82	0.238	214	492
liarwhd	6	125	0.101	316	625	125	0.095	316	750
nondia	10	5 846	6.156	26 602	27 188	1 517	1.397	8 470	8 674
nondquar	20	8 510	13.227	37 570	39 690	2 611	4.139	15 114	14 854
diagonal1	20	96	0.032	242	480	96	0.031	242	576
diagonal1	50	159	0.055	498	765	156	0.123	590	910
diagonal2	200	416	0.482	1 662	1 940	260	0.343	1 384	1 484
diagonal2	500	895	1.637	3 672	4 187	493	0.931	2 656	2 834
diagonal2	1 000	1 286	3.773	5564	6 020	655	2.122	3 378	3 760
diagonal2	5 000	3 701	40.587	16 476	17 263	1 623	20.600	9 334	9 244
diagonal2	10 000	6 868	130.020	30 616	32 056	2 211	48.301	12 514	12 602
diagonal3	20	113	0.060	302	553	109	0.038	370	638
diagonal3	50	225	0.185	732	1 079	207	0.105	784	1 202
diagonal4	10	430	0.537	1 548	2 046	320	0.302	1 446	1 844
diagonal4	50	444	0.532	16 92	2 084	283	0.258	71 134	1 644
diagonal5	60	87	0.043	212	435	87	0.092	212	522
diagonal7	60	87	0.054	212	435	87	0.068	212	522
diagonal7	80	88	0.070	214	440	88	0.038	214	528
diagonal7	300	93	0.049	228	465	93	0.049	228	558
diagonal7	600	98	0.122	240	490	98	0.109	240	588
diagonal7	500	97	0.068	238	485	97	0.125	238	582
diagonal7	1 000	96	0.191	238	480	96	0.176	238	576
diagonal7	1 200	97	0.201	240	485	97	0.242	240	582
diagonal8	40	84	0.023	204	420	84	0.023	204	504
diagonal8	200	94	0.073	230	470	94	0.076	230	564
diagonal8	500	97	0.119	238	485	97	0.105	238	582
diagonal8	1 200	101	0.218	248	505	101	0.207	248	606
diagonal8	2 200	103	0.376	254	515	103	0.339	254	618
diagonal8	4 000	104	0.597	258	520	104	0.555	258	624
dixon3dq	50	1777	1.973	7 730	8 293	1 233	1.358	7 026	7 028
dixon3dq	100	4 560	5.418	20 290	21 312	1 986	2.178	11 056	11 332
dixon3dq	200	10 002	12.247	44 536	46570	2 375	2.834	14 498	13 486
dixon3dq	500	31 881	44.434	143 018	147 729	4 509	6.089	27 002	25 630
dixon3dq	1 000	85 656	155.058	383 728	397 582	11 298	18.763	64 092	64 438
edensch	4	94	0.044	236	470	94	0.039	236	564
edensch	10	103	0.081	256	515	103	0.068	256	618
fletchcr	4	116	0.136	384	578	115	0.106	434	688
fletchcr	20	79	0.089	194	395	79	0.066	194	474
hager	50	94	0.212	232	470	94	0.147	232	564
hager	100	101	0.093	252	505	101	0.043	252	606
hager	300	113	0.124	282	565	113	0.122	282	678
himmelbg	500	77	0.348	230	385	77	0.340	230	462
himmelbg	1 000	81	0.498	242	405	81	0.476	242	486
himmelbg	5 000	86	1.426	258	430	86	1.475	258	516

续表 1

Problem	Dim	算法 VFR				算法 1			
		Iter	Tcpu/s	NF	NG	Iter	Tcpu/s	NF	NG
himmelbg	10 000	86	2.376	262	430	86	2.391	262	516
almost perturbed quadratic	50	221	0.092	662	1 057	212	0.107	832	1 230
almost perturbed quadratic	100	380	0.263	139 2	1 800	224	0.123	1 034	1 292
almost perturbed quadratic	200	612	0.630	250 0	2 872	385	0.541	1 840	2 214
almost perturbed quadratic	500	1 356	1.774	582 8	6 344	791	0.928	3 944	4 554
qf1	100	360	0.269	1358	1688	254	0.239	1114	1460
qf1	200	604	0.562	2464	2834	384	0.368	1782	2208
qf2	50	252	0.238	786	1204	180	0.177	742	1042
qf2	100	410	0.435	1518	1940	270	0.279	1322	1540
quartc	100	105	0.129	280	525	105	0.119	280	630
quartc	500	111	0.305	298	555	111	0.296	298	666
quartc	1 000	113	0.524	306	565	113	0.542	306	678
quartc	5 000	119	1.905	324	595	119	1.891	324	714
raydan1	50	102	0.030	250	510	102	0.027	250	612
raydan2	500	99	0.115	246	495	99	0.102	246	594
raydan2	1 000	101	0.136	252	505	101	0.133	252	606
raydan2	2 000	104	0.208	260	520	104	0.209	260	624
staircase1	10	411	0.606	1 578	1 925	236	0.416	1 150	1 354
tridia	60	1 635	2.016	7 582	7 615	1 066	1.340	6 058	6 098
dixmaanae	510	779	4.055	3 180	3 651	375	2.035	1 644	2 182
dixmaanae	1 800	1 689	19.657	7 076	7 919	817	12.210	4 568	4 674
dixmaanae	2 400	1 978	28.657	8 658	9 206	923	17.050	5 024	5 284
dixmaanae	3 600	2 275	44.778	10 322	10 631	979	23.626	5 140	5 618
dixmaanae	4 800	2 639	62.625	11 678	12 343	16 55	51.951	9 466	9 436
dixmaanaf	300	566	2.472	2 180	2 660	364	1.569	1 580	2 108
dixmaanaf	1 200	1 169	13.310	5 110	5 481	790	9.857	4 296	4 512
dixmaanag	600	774	5.638	3 346	3 640	395	2.643	1 978	2 272
dixmaanag	1 200	1 210	13.178	5 046	5 682	794	10.028	4 462	4 540
dixmaanag	3 600	2 448	60.074	10 616	11 446	1 144	33.807	6 142	6 558
dixmaanag	7 200	2 867	123.513	12 990	13 345	1 254	61.479	7 012	7 188
dixmaanah	1 800	1 512	22.869	6 864	7 072	972	15.682	4 866	5 572
dixmaanah	2 400	1 461	28.900	6 620	6 807	861	17.721	4 574	4 920
dixmaanak	450	5 775	42.000	25 274	25 615	2 097	14.166	11 098	12 002
dixmaanak	1 200	2 686	32.184	11 680	12 608	1 003	12.343	5 434	5 730
dixmaanak	2 400	1 871	36.080	7 948	8 795	659	13.209	3 510	3 760
dixmaanak	3 600	1 720	46.192	7 652	8 090	1037	29.416	5 498	5 924
dixmaanak	4 500	1 841	57.803	8 040	8 543	623	21.993	3 774	3 552
dixmaanal	150	3 987	15.003	17 726	18 737	1 415	5.424	7 636	8 104
dixmaanal	1 200	2 148	26.138	9 388	10 032	1 327	16.385	7 042	7 582
dixmaanal	2 400	1 304	25.127	5 682	76 062	601	12.133	3 242	3 446
dixmaanal	3 600	1 508	38.412	6 198	7 116	1 132	33.645	6 242	6 470
dixmaanal	4 800	1 532	49.714	6 722	7 160	911	32.418	5 126	5 214
dixmaanaj	207	6 896	32.463	30 268	32 068	1 786	8.260	10 024	10 202
dixmaanaj	510	4 594	34.171	20 906	21 202	1 498	11.285	8 782	8 548
dixmaanaj	666	4 154	36.130	18 442	19 170	1 295	10.778	7 176	7 406
dixmaanaj	999	3 833	42.726	17 254	17 479	1 843	20.794	9 916	10 532
dixmaanaj	2 100	1 337	22.920	5 832	6 287	746	13.800	4 080	4 248

续表 1

Problem	Dim	算法 VFR				算法 1			
		Iter	Tcpu/s	NF	NG	Iter	Tcpu/s	NF	NG
dixmaanaj	4 500	1 852	60.167	8 360	8 640	883	29.325	4 774	5 068
dixmaanai	99	3 638	9.349	16 320	16 964	1 040	2.789	5 596	5 936
dixmaanai	207	6 962	25.765	32 450	32 146	2 404	9.536	13 830	13 694
dixmaanai	510	8 718	50.731	38 272	40 808	3 377	20.848	19 142	19 244
dixmaanai	999	14 843	136.471	64 970	69 367	4 551	46.842	27 290	25 870
eg2	10	1 071	1.131	4 692	5 035	456	0.382	2 448	2 600
eg2	20	3 494	4.043	15 816	16 240	1 190	1.334	6 990	6 780

此外，我们还利用 Dolan 和 More[14] 提出的性能概况理论来比较算法的优劣，算法在某种度量下的性能概况能够反映它的效率性和稳定性. 由此我们分别以迭代时间、迭代次数、目标函数值迭代次数及目标函数梯度值的迭代次数为度量，绘制出算法 1 和 VFR 法在 4 种度量下的性能概况图(图 1—图 4).

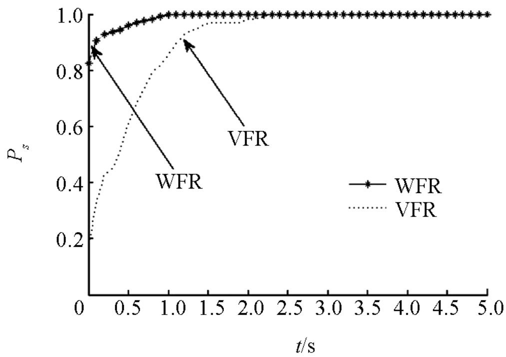

图 1 迭代时间性能概况图

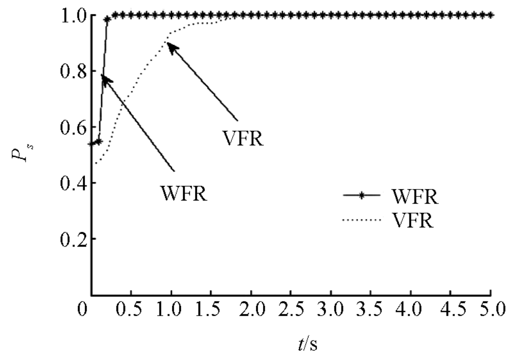

图 2 梯度迭代次数性能概况图

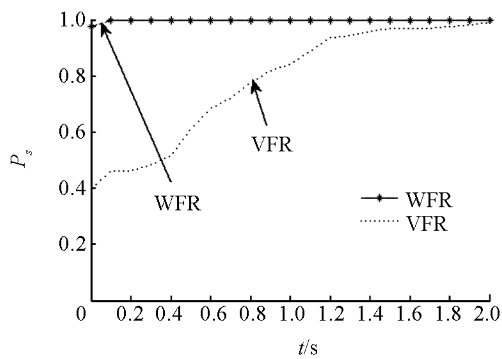

图 3 迭代次数性能概况图

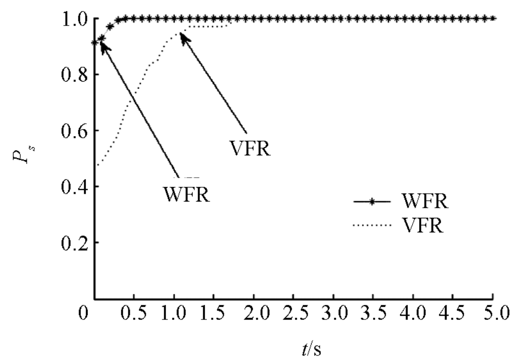

图 4 函数值迭代次数性能概况图

由性能概况图中曲线的变化趋势可以看出，无论在迭代时间、迭代次数、目标函数值迭代次数及目标函数梯度值的迭代次数方面，算法 1 的效率和稳定性都优于 VFR 共轭梯度法. 所以新算法是有效的，是 VFR 谱共轭梯度法的改进.

参考文献：

[1] HESTENES M R, STIEFEL E. Methods of Conjugate Gradients for Solving Linear Systems [J]. Journal of Research of the National Bureau of Standards, 1952, 49(6): 409-436.

[2] FLETCHER R, REEVES C M. Function Minimization by Conjugate Gradients [J]. Computer Journal, 1964, 7(2): 149-154.

[3] POLYAK B T. The Conjugate Gradient Method in Extremal Problems [J]. Ussr Computational Mathematics and Mathematical Physics, 1969, 9(69): 94-112.

[4] POLAK E，RIBIÈRE G. Note Sur La Convergence De Méthodes De Directions Conjuguées. [J]. Rev. franaise Informat. recherche Opérationnelle，1968，16(16)：35－43.

[5] DAI Y H，YUAN Y. A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property [J]. Siam Journal on Optimization，1999，10(1)：177－182.

[6] YAO S W，WEI Z X，HUANG H. A Note about WYL's Conjugate Gradient Method and Its Applications [J]. Applied Mathematics and Computation，2007，191(2)：381－388.

[7] DAI Z，WEN F. Another Improved Wei-Yao-Liu Nonlinear Conjugate Gradient Method with Sufficient Descent Property [J]. Applied Mathematics and Computation，2012，218(14)：7421－7430.

[8] YUAN G L，LU X W. A Modified PRP Conjugate Gradient Method [J]. Annals of Operations Research，2009，166(1)：73－90.

[9] LU A G，LIU H W，ZHENG X Y，et al. A Variant Spectral-Type FR Conjugate Gradient Method and Its Global Convergence [J]. Applied Mathematics and Computation，2011，217(12)：5547－5552.

[10] HUANG H，LIN S H. A Modified Wei-Yao-Liu Conjugate Gradient Method for Unconstrained Optimization [J]. Applied Mathematics and Computation，2014，231(1)：179－186.

[11] BIRGIN E G，MARTÍNEZ J M. A Spectral Conjugate Gradient Method for Unconstrained Optimization [J]. Applied Mathematics and Optimization，2001，43(2)：117－128.

[12] 戴彧虹，袁亚湘. 非线性共轭梯度法 [M]. 上海：上海科学出版社，2000：10－13.

[13] ANDREI N. An Unconstrained Optimization Test Functions Collection [J]. Environmental Science and Technology，2008，10(1)：6552－6558.

[14] DOLAN E D，MORÉ J J. Benchmarking Optimization Software with Performance Profiles [J]. Mathematical Programming，2002，91(2)：201－213.

A Modified FR Spectral Conjugate Gradient Method

WANG Sen-sen[1]，　ZHANG Jun-rong[2]，　HAN　Xin[3]

1. *School of Mathematics and Information，Hotan Teachers College，Hetian Xinjiang 848000，China*；
2. *School of Mathematics and Statistics，Southwest University，Chongqing 400715，China*；
3. *School of Mathematics，Sichuan University of Arts and Science，Dazhou Sichuan 635000，China*

Abstract：In this paper，a spectral conjugate gradient WFR method is put forward，which always possesses the sufficient descent property with any line search. It is proved under the standard Wolfe line search that the new spectral conjugate gradient method possesses global convergence. A series of numerical tests indicate that the new algorithm is superior to the VFR method.

Key words：unconstrained optimization；spectral conjugate gradient method；global convergence；Wolfe line search

责任编辑　张　枸