ISSN : 0974 - 7435 Volume 10 Issue 20

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(20), 2014 [12415-12422]

A Nonmonotone Line Search combination algorithm for unconstrained optimization problems

Ping Hu*, Xiaocai Wang
Faculty of Mathematics and Physics, Huaiyin Institute of Technology,
Huai'an 223003, (P.R.CHINA)
E-mail : hyhuping@163.com

ABSTRACT

In this paper we propose a nonmonotone line search combination rule for unconstrained optimization problems and show that it possesses the global convergence property. We establish the corresponding algorithm and illustrate its effectiveness by virtue of some numerical tests. Simulation results indicate that the proposed method is very effective.

KEYWORDS

Unconstrained optimization problems; Nonmonotone line search algorithm; Global convergence; Combination rule; BFGS algorithm.

INTRODUCTION

Unconstrained optimization problems have been paid considerable attention by the researchers, because of comprehensive practical application background. There are many authors have made great efforts on the study of optimization algorithms. Consider a general unconstrained optimization problem denoted by

$$\min_{x\in R^n} f(x) \tag{1}$$

Where $f(x)$ is a continuously differentiable function from R^n to R. In the literature, it is customary to use iterative methods to solve this problem. At current iteration x_k, if $g_k = \nabla f(x_k) \neq 0$, we can find a step-length α_k by carrying some line search along the direction d_k, and then obtain the next iteration as

$$x_{k+1} = x_k + \alpha_k d_k \tag{2}$$

where d_k is the search direction, which can be determined by many methods. The literature[1] points out that the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is the best quasi-Newton method and gave the associated procedures. For convenience, we will employ the BFGS algorithm in this paper to determine d_k, namely, let

$$d_k = \begin{cases} -H_k g_k, & if\ g_k^T d_k < 0 \\ -g_k, & if\ g_k^T d_k \geq 0 \end{cases} \tag{3}$$

Where $g_k = \nabla f(x_k) \neq 0$, H_k is generated by the BFGS correction formula. Moreover, (3) guarantee that the condition $-g_k^T d_k > 0$ holds.

There are already some well-known rules used to determine the step-length α_k in (2). Among them, the Arimijo rule, the Goldstein rule, and the Wolfe rule are popular and widely used by many authors[1]. However, in obtaining the step length α_k, traditional line searches require the function value to decrease monotonically at every iteration, namely

$$f(x_{k+1}) \leq f(x_k) \tag{4}$$

Consequently, they are in general called to be *monotone line search technique*, which is resultful in some situations. However, subsequent studies showed that the convergence rate of monotone line search technique may reduce considerably when the iteration locates in a narrow curved valley[2,3,4]. To overcome this problem, Grippo et al. introduced a highly innovative method[2] called the *nonmonotone line search technique*and illustrated its effectiveness by means of some numerical tests. This method has been developed by many authors. For those nonmonotone line search rules[2-13], the inequality $f(x_{k+1}) > f(x_k)$ may hold for any k, and therefore, it can play a nonmonotone search role in the above three rules. However, the nonmonotone line search rules[2-9,11-13] for problems (1) essentially required the approximation sequence $\{x_k\}$ satisfies $f(x_k) \leq f(x_0)$ for any $k \geq 1$. Under this condition, the approximation sequence $\{x_k\}$ will be trapped and can not escape from the valley bottom if the initial point x_0 locates near a valley bottom. Motivated by this problem, we propose a new rule called to be ***a nonmonotone line search combination rule*** in the following. Moreover, we show that it possesses the global convergence property by virtue of[3] and[4]. With the help of numerical experiments it is shown that the proposed method is very effective for above problem.

Nonmonotone line search combination rule

In this section, we will put forward a nonmonotone line search combination rule. First, the following assumption is the necessary.

Assumption 1 *Throughout this paper, we assume that the function* $f: R^n \to R$ *is bounded and differentiable on the level set* $\Omega = \left\{x \middle| x \in R^n : f(x) \leq c \left| f(x_0) \right| \right\}$ *for a given constant* $c \geq 1$.

Nonmonotone line search combination rule. *Let the bounded step-length* $\alpha_k \geq 0$ *along the direction* d_k *such that*

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq \sum_{r=o}^{m(k)} \lambda_{kr} \beta^{h_{kr}\, sign\, f(x_{k-r})} f(x_{k-r}) + \rho \alpha_k g_k^T d_k \tag{5}$$

Where $\beta \geq 1$, $h_{kr} \geq 0$, *and* $m(k) = \min[k, M-1]$ *with* $M \geq 1$ *is a positive integer. Let* $1 \geq \lambda_{kr} > \lambda > 0$, $\sum_{r=o}^{m(k)} \lambda_{kr} = 1$, $h_k = \sum_{r=o}^{m(k)} h_{kr}$ *and* $\sum_{k=o}^{\infty} h_k = S$ *with S is a finite constant.*

Clearly, inserting $M = 1$ and $\beta = 1$ into the representation (5) gives $f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k g_k^T d_k$, then the rule (5) reduces to the rule of Arimijo. Further, if $f(x_{k-l}) = \max_{0 \leq r \leq m(k)} \{f(x_{k-r})\}$ and $\lambda_{kl} = 1$, $\lambda_{kr} = 0\,(r \neq l)$, then (5) becomes $f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} \{f(x_{k-j})\} + \rho\alpha_k g_k^T d_k$, which is the nonmonotone line search rule proposed by Grippo et al.[2]. Therefore, the new rule (5) is a general version of some traditional rules well known in the literature.

Proof for the global convergence of the new rule

In this section, we investigate the global strong convergence properties of unconstrained optimization in conjunction with the new nonmonotone line search combination rule. First, the following some definitions and lemmas are necessary.

Definition 1 *The function* $\sigma: [0, +\infty) \to [0, +\infty)$ *is a forcing function (F-function) if for any sequence* $\{t_i\} \subset [0, +\infty)$, $\lim_{i \to \infty} \sigma(t_i) = 0$ *implies* $\lim_{i \to \infty} t_i = 0$.

The literature[3] proved that there exists a F-function $\sigma(t_i)$ such that

$$\rho\alpha_k g_k^T d_k \leq -\sigma(t_k) \text{ with } t_k = -g_k^T d_k / \|d_k\| \geq 0. \quad \textbf{(6)}$$

By the inequalities (5) and (6), the rule (5) becomes

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq \sum_{r=o}^{m(k)} \lambda_{kr} \beta^{h_{kr}\, sign\, f(x_{k-r})} f(x_{k-r}) - \sigma(t_k). \quad \textbf{(7)}$$

Lemma 3.1 *In (2), the search direction* d_k *is determined by the following BFGS algorithm. Then there is a constant* $\gamma > 0$ *satisfying*

$$-g_k^T d_k / \|d_k\| \geq \gamma \|g_k\|, k = 0, 1, 2, \cdots \quad \textbf{(8)}$$

Proof. When the $g_k \neq 0$, according (3), we easily get $g_k^T d_k < 0$. In view of $-g_k^T d_k / \|d_k\| = \|g_k\| \cos(-g_k, d_k)$, then there is a constant $\gamma > 0$ such that $\cos(-g_k, d_k) > \gamma > 0$. Therefore, (8) holds.

Lemma 3.2 *If* α_k *satisfies the rule (5) for* $k \geq 1$, *then*

$$f(x_{k+1}) \leq |f(x_0)| \beta^{\sum_{i=0}^{k} h_i} - \lambda \sum_{r=o}^{k-1} \sigma(t_r) - \sigma(t_k) \quad \textbf{(9)}$$

Proof. The principle of mathematical induction will be used to prove the conclusion.

If $k = 1$ and $M = 1$, then we have $m(k) = 0$. By the inequality (7), we have

$$f(x_2) \leq \lambda_{10} \beta^{h_{10}\, sign\, f(x_1)} f(x_1) - \sigma(t_1) \leq \beta^{h_{10}} f(x_1) - \sigma(t_1).$$

Noting that $\beta^{h_k} \geq \beta^{h_{kr}} \geq 1 \geq h_{kr} > \lambda$ and

$$f(x_1) \leq \lambda_{00} \beta^{h_{00}\, sign\, f(x_0)} f(x_0) - \sigma(t_0) = \beta^{h_{00}} |f(x_0)| - \sigma(t_0).$$

We have

$$f(x_2) \leq \beta^{h_{10}} \left\{\beta^{h_{00}} |f(x_0)| - \sigma(t_0)\right\} - \sigma(t_1)$$
$$= \beta^{h_{00} + h_{10}} |f(x_0)| - \beta^{h_{10}} \sigma(t_0) - \sigma(t_1)$$

$$\leq \beta^{h_0+h_1}\left|f(x_0)\right| - \beta^{h_{10}}\sigma(t_0) - \sigma(t_1). \tag{10}$$

If $k=1$ and $M \geq 2$, by the inequality (7), we have

$$f(x_2) \leq \lambda_{10}\beta^{h_{10}\, sign\, f(x_1)} f(x_1) + \lambda_{11}\beta^{h_{11}\, sign\, f(x_0)} f(x_0) - \sigma(t_1)$$
$$\leq \lambda_{10}\ \beta^{h_{10}}\left\{\beta^{h_{00}}\left|f(x_0)\right| - \sigma(t_0)\right\} + \lambda_{11}\beta^{h_{11}}\left|f(x_0)\right| - \sigma(t_1)$$
$$\leq (\lambda_{10}\ \beta^{h_1+h_0} + \lambda_{11}\beta^{h_1})\left|f(x_0)\right| - \lambda_{10}\beta^{h_1}\ \sigma(t_0) - \sigma(t_1)$$
$$\leq \beta^{h_1+h_0}(\lambda_{10} + \lambda_{11})\left|f(x_0)\right| - \lambda_{10}\ \sigma(t_0) - \sigma(t_1)$$
$$\leq \beta^{h_1+h_0}\left|f(x_0)\right| - \lambda\ \sigma(t_0) - \sigma(t_1). \tag{11}$$

By (10) and (11), it follows that (9) holds for $k=1$. We now assume that (9) holds for $j=k-1$, namely,

$$f(x_k) \leq \left|f(x_0)\right|\beta^{\sum_{i=0}^{k-1}h_i} - \lambda\sum_{r=o}^{k-2}\sigma(t_r) - \sigma(t_{k-1}).$$

Then we have

$$f(x_{k+1}) \leq \sum_{r=o}^{m(k)}\lambda_{k\,r}\beta^{h_{k\,r}\, sign\, f(x_{k-r})} f(x_{k-r}) - \sigma(t_k)$$
$$\leq \sum_{r=o}^{m(k)}\lambda_{k\,r}\beta^{h_k}\ \{\left|f(x_0)\right|\beta^{\sum_{i=0}^{k-r-1}h_i} - \lambda\sum_{i=o}^{k-r-2}\sigma(t_i) - \sigma(t_{k-r-1})\} - \sigma(t_k)$$
$$\leq \sum_{r=o}^{m(k)}\lambda_{k\,r}\beta^{h_k}\ \{\left|f(x_0)\right|\beta^{\sum_{i=0}^{k-1}h_i} - \lambda\sum_{i=o}^{k-r-2}\sigma(t_i) - \sigma(t_{k-r-1})\} - \sigma(t_k)$$
$$\leq \sum_{r=o}^{m(k)}\lambda_{k\,r}\beta^{h_k}\ \{\left|f(x_0)\right|\beta^{\sum_{i=0}^{k-1}h_i} - \lambda\sum_{i=o}^{k-m(k)-2}\sigma(t_i) - \sigma(t_{k-r-1})\} - \sigma(t_k)$$
$$\leq \left|f(x_0)\right|\ (\sum_{r=o}^{m(k)}\lambda_{k\,r})\ \beta^{\sum_{i=0}^{k}h_i} - \lambda\ (\ \sum_{r=o}^{m(k)}\lambda_{k\,r}\ )\ \sum_{i=o}^{k-m(k)-2}\sigma(t_i)\ - \sum_{r=o}^{m(k)}\lambda_{k\,r}\ \sigma(t_{k-r-1}) - \sigma(t_k)$$
$$= \left|f(x_0)\right|\ \beta^{\sum_{i=0}^{k}h_i} - \lambda\ \sum_{i=o}^{k-m(k)-2}\sigma(t_i)\ - \sum_{r=o}^{m(k)}\lambda_{k\,r}\ \sigma(t_{k-r-1}) - \sigma(t_k)$$
$$\leq \left|f(x_0)\right|\ \beta^{\sum_{i=0}^{k}h_i} - \lambda\ \sum_{i=o}^{k-m(k)-2}\sigma(t_i)\ - \lambda\sum_{r=o}^{m(k)}\sigma(t_{k-r-1}) - \sigma(t_k)$$
$$= \left|f(x_0)\right|\ \beta^{\sum_{i=0}^{k}h_i} - \lambda\ \sum_{i=o}^{k-m(k)-2}\sigma(t_i)\ - \lambda\sum_{i=k-m(k)-1}^{k-1}\sigma(t_i) - \sigma(t_k)$$
$$\leq \left|f(x_0)\right|\ \beta^{\sum_{i=0}^{k}h_i} - \lambda\sum_{r=o}^{k-1}\sigma(t_r) - \sigma(t_k).$$

Which means that (9) holds for $j=k$. By the principle of mathematical induction, (9) holds for any given $k \geq 1$.

Next, we will prove the global strong convergence of the new rule.

Theorem 3.1 *Under the above assumptions 1, let the search direction* d_k *and the step-length* α_k *be determined by BFGS algorithm and (5), respectively. Assume* $\{x_k\}$ *is a sequence generated by (2) according to the search direction* d_k *and the step-length* α_k*, Then we have*

$$\{x_k\} \in \Omega,\ and\ \lim_{k\to\infty}\left\|g_k\right\| = 0.$$

Proof. According to Lemma3.2, we have

$$f(x_{k+1}) \le \left| f(x_0) \right| \beta^{\sum_{i=0}^{k} h_i} - \lambda \sum_{r=o}^{k-1} \sigma(t_r) - \sigma(t_k) \le \left| f(x_0) \right| \beta^{S} .$$

Therefore, $f(x_{k+1}) \le c \left| f(x_0) \right|$ with $c = \beta^S > 1$. Then, by the definition of Ω , $\{x_k\} \in \Omega$ follows immediately.

By (9), we have

$$f(x_{k+1}) \le \left| f(x_0) \right| \beta^{\sum_{i=0}^{k} h_i} - \lambda \sum_{r=o}^{k-1} \sigma(t_r) - \sigma(t_k)$$

$$\le \left| f(x_0) \right| \beta^{\sum_{i=0}^{k} h_i} - \lambda \sum_{r=o}^{k} \sigma(t_r)$$

$$\le c \left| f(x_0) \right| - \lambda \sum_{r=o}^{k} \sigma(t_r) .$$

Which implies

$$0 \le \lambda \sum_{r=o}^{k} \sigma(t_r) \le c \left| f(x_0) \right| - f(x_{k+1}) . \tag{12}$$

for all $k \ge 1$. By Assumption 1, $f(x_k)$ is bounded on the level set Ω . Then (12) indicates that $\lambda \sum_{r=o}^{k} \sigma(t_r) < \infty$ which implies $\lim_{k\to\infty} \sigma(t_k) = 0$. Combined with the fact that function σ is a F-function, it follows then that

$$\lim_{k\to\infty} t_k = \lim_{k\to\infty} (-g_k^T d_k \big/ \|d_k\|) = 0 .$$

By Lemma3.1, it follows that $\lim_{k\to+\infty} \|g_k\| = 0$. The proof is completed.

Numerical tests

Here, we apply our rule to some standard tests problems.

Algorithm (I)

Step 1. Initialization. Given the initial values $x_0 \in R^n$, $H_0 \in I$ and other data including an integer $M \ge 1$, a constant $\alpha_0 = 1$, $\varepsilon \ge 0$, $\rho \in (0,\ 0.5)$, as well as k=0.

Step 2. Test termination conditions. Examine the stopping criterion by computing $g_k = \nabla f(x_k)$. If $\|g_k\| \le \varepsilon$, $x^* = x_k$ and the algorithm stops.

Step 3. Determine search direction. Calculate $d_k = -H_k g_k$, if $g_k^T d_k > 0$, set $d_k = -g_k$,which guarantee the condition $-g_k^T d_k > 0$ holds.

Step 4. Determine the line search step α_k . Let $m(k) = \min[k, M-1]$. If (5) holds, $\alpha_k = \alpha$. Otherwise, contract α .

Step 5. Compute the next point. Set $s_k = \alpha_k d_k$ and $x_{k+1} = x_k + s_k$, and then calculate $f(x_{k+1})$ and $g_{k+1} = \nabla f(x_{k+1})$.

Step 6. Update the iteration matrix H_{k+1} using BFGS formulae, namely, Set

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + v_k v_k^T$$

With $y_k = g_{k+1} - g_k$, $v_k = (y_k^T H_k y_k)^{\frac{1}{2}} [\frac{s_k}{s_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k}]$. Set $k =$ k+1 and then go to the step 2.

The functions in the numerical tests are the same as[5,7]. Take now the parameters involved in Algorithm (I) as follows:

$\varepsilon = 10^{-6}$, $\rho = 10^{-3}$, $H_0 = I_{n\times n}$, $\lambda_{kr} = \dfrac{1}{1+m(k)}$, $h_{kr} = \dfrac{1}{(1+\mathrm{k})^{1.2}}$.

With $r = 0,1,\cdots,m(k)$.

Clearly, Algorithm (I) reduces to the standard of the BFGS algorithm (saying **Algorithm (II)**) when M=1 and $\beta = 1$. If $M > 1$ and $\beta = 1$, algorithm (I) is similar to the nonmonotone BFGS algorithm (saying **Algorithm (III)**). The three methods are compared with the aid of Matlab, where the controllable parameter M is taken as 1 and 6 while λ as 1 and 3. The numerical results are presented in the following four TABLES, in which n_g and n_f denote the outside loop iterations and the function evaluations, respectively, while $f(x^*)$ is the function value of the approximate solution x^* .

TABLE 1 : Algorithm (I) does not dominate compared to Algorithm (II) or (III)

Problem	Dimension	Algorithm 2(M =1, β =1)		Algorithm 3(M =3, β =1)		Algorithm 1(M =3, β =6)	
		n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$

					9.0191e-		
			-----		15		
			2.8656e-		-----		
			13		7.5025e-		
Pow.Sin.			4.0157e-		13		
Ex.Ros.	4	>999	13	>999	4.6651e-	60/327	3.3350e-04
Ex.Ros.	10	95/245	3.1755e-	98/246	20	85/212	4.6259e-15
Ex.Ros.	30	190/515	13	>999	3.5514e-	175/464	6.1892e-14
Ex.W.&	40	230/636	4.6651e-	239/649	23	205/560	4.0964e-14
H.	2	32/77	20	32/77	2.5945e-	28/66	1.2043e-17
Ge.Ros.	2	39/96	1.5647e-	38/92	16	31/75	7.8019e-20
Ge.Ros.	8	>999	17	60/167	1.2228e-	66/180	1.8386e-17
Ge.Ros.	20	>999	-----	133/379	17	138/385	7.5640e-17
Penalty Ⅰ	30	136/328	-----	22/60	2.4773e-	22/60	2.4773e-04
Penalty Ⅰ	40	>999	2.4773e-	>999	04	74/160	3.3925e-04
Penalty Ⅰ	80	>999	04	>999	-----	155/351	7.1305e-04
Penalty Ⅱ	32	>999	-----	104/248	------	112/270	0.1032
Penalty Ⅱ	40	>999	------	98/253	0.1032	102/253	0.5569
Watson	10	66/154	------	68/155	0.5569	64/149	5.3212e-07
Watson	36	58/157	------	85/210	5.3203e-	56/150	3.3854e-08
Watson	150	>9999	5.3203e-	97/356	07	98/344	1.2336e-09
Per.Qua	70	>999	07	90/349	1.5352e-	90/347	5.8369e-15
Per.Qua	80	>999	3.3840e-	103/403	09	102/399	1.6813e-14
Per.Qua	90	>999	08	>999	1.2814e-	115/451	1.0610e-14
Raydan Ⅰ	80	>999	------	201/3828	09	103/217	324.00
Raydan Ⅰ	100	519/14606	-----	73/192	1.0450e-	146/318	505.00
Raydan Ⅰ	200	>999	-----	>999	14	297/752	2.0100e+03
Raydan Ⅰ			-----		2.9824e-		
			-----		15		
			505.00		------		
			-----		324.00		
					505.00		

Form TABLE 1, Algorithm (I) performs the best for the given initial point. As see form TABLE 2, the efficiency of Algorithm (I) is not worse then Algorithm (II) and Algorithm (III).

Extended Freudenstein & Roth (EFR) function is non-convex with two minimum points, one of which is globally minimum. Form TABLE 3, it is concluded that Algorithm (I) performs the best in finding the globally optimal solution and is very suitable for solving high-dimension problem. In specification, in the case of the dimension equal to 2, both Algorithms (II) and (III) give the minimum 48.9843 while Algorithm (I) gives the globally minimum 2.0835e-019.

The results for Brown and Dennis Function are presented in TABLE 4. It is easy to see that the iterations of Algorithm (I) is decreasing as $ becomes larger, which indicates the efficiency the new algorithm.

TABLE 2 : Algorithm (I) does not dominate compared to Algorithm (II) or (III)

Problem	Dimension	Algorithm 2(M =1, β =1)		Algorithm 3(M =3, β =1)		Algorithm 1(M =3, β =6)	
		n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$
Beale							
Ex.W.&				2/28		2/28	
H.	2	12/28	9.6837e-18	89/225	9.6837e-18	83/206	9.6837e-18
Ex.W.&	10	85/219	6.5620e-16	126/34	3.7749e-16	121/32	1.8803e-14
H.	20	123/341	1.4192e-14	1	4.7872e-14	4	6.2400e-15
Ex.W.&	30	156/448	3.6872e-14	154/44	6.5794e-14	153/43	1.7713e-14
H.	1000	22/45	1.2011e-06	0	1.2011e-06	2	1.2011e-06
BDEXP	5000	23/47	2.8855e-06	22/45	2.8855e-06	22/45	2.8855e-06
BDEXP	2	8/20	8.0664e-07	23/47	8.0664e-07	23/47	8.0664e-07
Penalty II	10	18/51	4.5920e-14	9/21	3.9012e-14	9/21	6.1450e-14
Per.Qua	100	11/65	100	19/51	100	19/50	100
Raydan II	1000	11/65	1000	16/78	1000	17/78	1000
Raydan II	50	45/93	5.4517e-06	16/78	5.4517e-06	17/78	5.4517e-06
Trigo.	500	53/109	4.4742e-07	44/89	4.4742e-07	44/89	4.4742e-07
Trigo.	1000	12/36	883.1941	52/107	883.1941	52/107	883.1941
Ex.Penalty	100	75/219	1.1267e-08	16/45	1.1268e-08	21/55	1.1307e-08
Watson				76/217		76/216	

TABLE 3 : Results for Extended Freudenstein and Roth function (using the initial point x_0 =(0.5,−2,0.5,−2,···,0.5,−2))

Dimension	Algorithm 2(M =1, β =1)		Algorithm 3(M =3, β =1)		Algorithm 1(M =3, β =6)	
	n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$	n_g/n_f	$f(x^*)$
2	10/32	48.9843	11/33	48.9843	15/42	2.0835e-19
6	45/759	146.9528	22/65	146.9528	39/158	1.1415e-15
10	27/95	244.9213	30/95	244.9213	46/144	1.3625e-16
18	53/174	440.8583	50/167	440.8583	62/217	2.8598e-16
22	52/219	538.8268	60/198	538.8268	75/259	1.7857e-16
24	46/504	587.8110	62/206	587.8110	80/282	1.6609e-16

TABLE 4 : Results for Brown and Dennis function (using m = 10 and the initial point x_0 =(0.5,−2,0.5,−2,···,0.5,−2))

Algorithm2 (M =1, β =1)	Algorithm 3 (M =3, β =1)	Algorithm 1(M =3)					
		β =6	β =10	β =20	β =30	β =40	β =50
n_g/n_f	n_g/n_f	n_g/n_f	n_g/n_f	n_g/n_f	n_g/n_f	n_g/n_f	n_g/n_f
-----	-----	134/609	78/380	37/154	37/150	35/145	31/127

CONCLUSIONS

By (12), we get $c\left|f(x_0)\right| - f(x_k) \ge 0$. Then it follows that $f(x_k) \le c\left|f(x_0)\right|$, where $c \ge 1$, and therefore it is possible to ensure $f(x_k) > f(x_0)$. Hence, we can reach the goal of slackness and let the iteration point escape from the valley near x_0 and search a better solution. This idea is a breakthrough of this paper.

We mention here that the rule (5) can be easily achieved. In general, there are many selections for λ_{kr}, h_{kr} which is involved in (5). Take $h_{kr} = \frac{1}{(1+\mathrm{k})^{p}}$ with the constants $p > 1$ for example. In this case, the series p-series $\sum_{k=0}^{\infty}\frac{1}{(k+1)^p}$ converge on a limited number *S*.

ACKNOWLEDGEMENT

Research supported by the Natural Science Foundation of Jiangsu Province (BK20141409). The second author is supported by the National Natural Science Foundation of China (No. 11226131) and Qing Lan Project.

REFERENCES

[1] Ni Qin; Optimization methods and programming, Science Press, Beijing **(2009)**.

[2] L.Grippo, F.Lampariello, S.Lucidi.; A nonmonotone line search technique for Newton's thods, SIAM J.Numer.Anal, **23**, 707-716 **(1986)**.

[3] Sun Wen-Yu, Han Ji-Ye, Sun Jie; Global convergence of nonmonotone descent methods for unconstrained optimization problems, Journal of Computational and Applied Mathematics, **146**, 89 -98 **(2002)**.

[4] Yu Zhen-Sheng, Pu Ding-Gou; A new nonmonotone line search technique for unconstrained optimization, Journal of Computational and Applied Mathematics, **219(1)**, 134-144 **(2008)**.

[5] Neculai Andrei; An Unconstrained Optimization Test Functions Collection, Adv.Model.Optim, **10**, 147-161 **(2008)**.

[6] M.A.Diniz-Ehrhardt, J.M.Martinez, M.Raydan; Aderivative-free nonmonotone line search technique for unconstrained optimization, Journal of Computational and Applied Mathematics, **219**, 383-397 **(2008)**.

[7] Shi Zhen-Jun, Shen Jie; New Inexact Line Search Method for Unconstrained Optimization, J. Optim. Theory Appl, **127**, 425-446 **(2005)**.

[8] Xiao Yun-Hai, Sun Hui-Juan, Wang Zhi-Guo; A globally convergent BFGS method nonmonotone line search for non-convex minimization. Journal of Computational and Applied Mathematics, **230(1)**, 95-106 **(2008)**.

[9] Zhou Qun-Yan, Sun Wen-Yu; Adaptive nonmonotone line search method for unconstrained optimization, Front. Math. China, **3**, 133-148 **(2008)**.

[10] Hu Ping, Liu Xu-Qing; A Nonmonotone Line Search Slackness Technique for Unconstrained Optimization, J.Optim.Theory Appl, **158**, 773-786 **(2013)**.

[11] Liu Jing-Hui, Ma Chang-Feng; A nonmonotone trust region method with new inexact line search for unconstrained optimization, Numerical Algorithms, **64**, 1-20 **(2012)**.

[12] Shi Zhen-Jun, Wang Sheng-Quan; Modified nonmonotone Armijo line search for descent method, Numerical Algorithms, **57**, 1-25 **(2010)**.

[13] Keyvan Amini, Masoud Ahookhosh, Hadi Nosratipour; An inexact line search approach using modified nonmonotone strategy for unconstrained optimization, Numerical Algorithms, **66**, 49-78 **(2014)**.