Numerical Algorithms **27:** 377–385, 2001.

Adaptive two-point stepsize gradient algorithm *

Yu-Hong Dai and Hongchao Zhang

State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, People's Republic of China
E-mail: {dyh,zhc}@lsec.cc.ac.cn

Received 13 April 2001; revised 24 October 2001
Communicated by C. Brezinski

Combined with the nonmonotone line search, the two-point stepsize gradient method has successfully been applied for large-scale unconstrained optimization. However, the numerical performances of the algorithm heavily depend on M, one of the parameters in the nonmonotone line search, even for ill-conditioned problems. This paper proposes an adaptive nonmonotone line search. The two-point stepsize gradient method is shown to be globally convergent with this adaptive nonmonotone line search. Numerical results show that the adaptive nonmonotone line search is specially suitable for the two-point stepsize gradient method.

Keywords: unconstrained optimization, steepest descent method, two-point stepsize gradient method, nonmonotone line search

AMS subject classification: 65k, 90c

1. Introduction

To minimize a smooth function f without constraints,

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{1.1}$$

we consider the two-point stepsize gradient method:

$$x_{k+1} = x_k - \alpha_k g_k, \tag{1.2}$$

where $g_k = \nabla f(x_k)$ and α_k is a stepsize decided by the information obtained at the points x_{k-1} and x_k. One efficient choice for α_k is presented in [2], and is given by

$$\alpha_k = \frac{s_{k-1}^{\mathrm{T}} s_{k-1}}{s_{k-1}^{\mathrm{T}} y_{k-1}}, \tag{1.3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

The two-point stepsize gradient method is preferable over the classical steepest descent method [6] both in theory and in real computations. It is known that

* Research was partly supported by the Chinese NSF grants 19801013, 10171104, and the Innovation Fund of Chinese Academy of Sciences.

the classical steepest descend method converges linearly and is badly affected by ill-conditioning (see [1,9]). However, the two-point stepsize gradient method is shown to be R-superlinearly convergent for two-dimensional strictly convex quadratics and the convergence rate is faster as its Hessian is more ill-conditioned [2]. For any dimensional convex quadratics, the method is shown to be R-linearly convergent [8]. For the nonquadratic case [16], incorporated a globalization scheme of the two-point stepsize gradient method using the technique of nonmonotone line search in [12]. The numerical results reported in [16] showed that the resulted algorithm is competitive to several famous conjugate gradient algorithms for large-scale unconstrained optimization. Due to its simplicity and numerical efficiency, the two-point stepsize gradient method has now received many studies, for example see [3–5,8,10,13,15,16].

Suppose that a descent search direction d_k has been computed at the kth iteration. The nonmonotone line search in [12] consists in computing a stepsize α_k such that

$$f(x_k + \alpha_k d_k) \leqslant \max_{0 \leqslant i \leqslant m(k)} f(x_{k-i}) + \delta \alpha_k g_k^{\mathrm{T}} d_k, \tag{1.4}$$

where $m(k) = \min(k, M-1)$, M is some fixed integer and δ is a small number. As discussed in his paper, the numerical performances of the two-point stepsize gradient algorithm by Raydan [16] heavily depend on the choice of M even for ill-conditioned problems. Such a state motivates us to propose an adaptive nonmonotone line search. Our initial idea in designing such a line search comes from [17], where Ph. L. Toint presented an adaptive nonmonotone technique for trust region method. As will be seen, however, some special techniques besides Toint's idea for trust region method are adopted into our nonmonotone line search. Thus, we propose an adaptive nonmonotone line search. Combined with this line search, the two-point stepsize gradient method is shown to be globally convergent. Numerical experiments show that the new nonmonotone line search is specially suitable for the two-point stepsize gradient method.

2. Adaptive nonmonotone line search

We denote by $f_{\min}$ the current best value of the objective function over all past iterates, that is, at the kth iteration

$$f_{\min} = \min_{0 \leqslant i \leqslant k} f(x_i). \tag{2.1}$$

The number of iterations since the value of $f_{\min}$ was actually obtained will be denoted by l. If l reaches a preset positive integer constant L, we reset the reference iteration and its associated function value, denoted by f_{r}. The value f_{r} will be used in the Armijo-like condition to accept the first trial step. When $l = L$, Toint [17] suggests for his trust region method resetting f_{r} to be f_{c}, the maximal value of the objective function since the last overall best value was found. As was observed, a direct application of Toint's technique to the two-point stepsize gradient method led to more computations than the use of the nonmonotone line search in [12] for many test problems (see the discussion in section 5). Thus, for a line search method, we suggest resetting f_{r} to be different candidate reference values if $l = L$, so that the first trial stepsize is accepted

as frequently as possible. As was just mentioned, one candidate reference value is f_c, that is the maximal value of the objective function since the last overall best value was found. The other, denoted by $f_{\max}$, is the maximal function value in recent M iterations, that is, at the kth iteration

$$f_{\max} = \max_{0 \leqslant i \leqslant \min\{k, M-1\}} f(x_{k-i}). \tag{2.2}$$

If $f_{\max}$ is too large, we select f_c as the new f_r; otherwise, we think that $f_{\max}$ is one better choice for the new f_r. More exactly, given a constant $\gamma_1 \geqslant 1$, we choose the new f_r as follows:

$$f_r = \begin{cases} f_c, & \text{if } \dfrac{f_{\max} - f_{\min}}{f_c - f_{\min}} > \gamma_1; \\ f_{\max}, & \text{otherwise.} \end{cases} \tag{2.3}$$

The reference value f_r should also be adjusted if the first trial stepsize is suddenly not accepted by the line search after many successful first trial stepsizes. Suppose that the first trial stepsize at the kth iteration is $\alpha_k^{(1)}$, and that p is the largest integer such that $\{\alpha_{k-i}^{(1)}\colon\ i = 1, \ldots, p\}$ are accepted but $\alpha_{k-p-1}^{(1)}$ not. If p is large, it is much possible that the reference value f_r is not changed and hence much greater than the current function value $f(x_k)$. In such a case, we use $f_{\max}$ as the reference value. More exactly, letting P be some integer and $\gamma_2 \geqslant 1$ some constant, we adjust f_r as follows if $p > P$:

$$f_r = \begin{cases} f_{\max}, & \text{if } f_{\max} > f(x_k) \text{ and } \dfrac{f_r - f(x_k)}{f_{\max} - f(x_k)} \geqslant \gamma_2; \\ f_r, & \text{otherwise.} \end{cases} \tag{2.4}$$

As discussed in [7], the first trial stepsize $\alpha_k^{(1)}$ usually includes more second-order information of $f(x)$ than the second candidate stepsize $\alpha_k^{(2)}$. Thus our nonmonotone line search is designed such that the stepsize $\alpha_k^{(1)}$ is accepted more easily than $\alpha_k^{(2)}$. In our line search, we require $f(x_k + \alpha_k^{(1)} d_k)$ and $f(x_k + \alpha_k^{(2)} d_k)$ to decrease f_r and $\min\{f_{\max}, f_r\}$, respectively. One could see this from the relations (2.5) and (2.6).

Suppose the initial parameters L, M, P, γ_1, γ_2, $0 < \sigma_1 < \sigma_2 < 1$ are given. Suppose that the search direction d_k and the first trial stepsize $\alpha_k^{(1)}$ are given at the kth iteration. We now describe our adaptive nonmonotone line search as follows.

Algorithm 2.1 (An adaptive nonmonotone line search).

Step 1 (Possibly reset the reference value).
(i) If $l = L$, update f_r by (2.3) and set $l := 0$;
(ii) If $p > P$, compute f_r by (2.4).

Step 2 (Test the first trial stepsize $\alpha_k^{(1)}$). If

$$f\left(x_k + \alpha_k^{(1)} d_k\right) \leqslant f_r + \delta \alpha_k^{(1)} g_k^{\mathrm{T}} d_k, \tag{2.5}$$

let $\alpha_k = \alpha_k^{(1)}$ and $p := p + 1$, go to Step 4; otherwise, $p := 0$.

Step 3 (Test other trial stepsizes till some stepsize is satisfactory).
(i) $\alpha_{\text{old}} = \alpha_k^{(1)}$;
(ii) Compute a stepsize α_{new} in $[\sigma_1\alpha_{\text{old}}, \sigma_2\alpha_{\text{old}}]$. If

$$f(x_k + \alpha_{\text{new}} d_k) \leqslant \min\{f_{\max}, f_{\text{r}}\} + \delta\alpha_{\text{new}} g_k^{\text{T}} d_k, \tag{2.6}$$

let $\alpha_k = \alpha_{\text{new}}$ and go to Step 4; otherwise, $\alpha_{\text{old}} = \alpha_{\text{new}}$, repeat (ii).

Step 4 (Possibly update the best value found and the candidate value).
(i) Let $f_{k+1} = f(x_k + \alpha_k d_k)$.
(ii) If $f_{k+1} < f_{\min}$, set $f_{\text{c}} = f_{\min} = f_{k+1}$ and $l := 0$; otherwise, $l := l + 1$.
(iii) If $f_{k+1} > f_{\text{c}}$, set $f_{\text{c}} = f_{k+1}$.
(iv) Compute $f_{\max}$ by (2.2) with k replaced by $k + 1$.

In the above algorithm, we suggest that integers P, M and L are so chosen that $L \leqslant 5$ and $P \geqslant 4M \geqslant 8L$. Comparing with the nonmonotone line search in [12] (see also (1.4)), the above one has four more parameters L, P, γ_1 and γ_2. We suggest that γ_1 and γ_2 in algorithm 2.1 may be set to M/L and P/M, namely, $\gamma_1 = M/L$ and $\gamma_2 = P/M$.

3. An adaptive two-point stepsize gradient algorithm and its global convergence

Combined the two-point stepsize gradient method and our adaptive nonmonotone line search, we then obtain a new gradient algorithm for unconstrained optimization. Due to its adaptivity in choosing its reference value, we call the new algorithm as adaptive two-point stepsize gradient (ATSG) algorithm in this paper. A full description of the algorithm is given as follows.

Algorithm 3.1 (An adaptive two-point stepsize gradient algorithm).

Step 0 (Give the starting point and initialize the parameters).
(i) Given $0 < \alpha_{\min} < \alpha_{\max}$, $0 < \sigma_1 < \sigma_2 < 1$ and $\epsilon \geqslant 0$; set $k := 1$.
(ii) Given positive integers $P > M > L$ and constants $\gamma_1 \geqslant 1$, $\gamma_2 \geqslant 1$.
(iii) Pick up $x_1 \in \mathbb{R}^n$, $\alpha_1^{(1)} \in [\alpha_{\min}, \alpha_{\max}]$ and compute $d_1 = -g_1$;
(iv) Set $l := 0$, $p := 0$ and $f_{\min} = f_{\text{r}} = f_{\text{c}} := f(x_1)$.

Step 1 (Test if the stopping condition holds). If $\|g_k\|_\infty \leqslant \epsilon$, stop.

Step 2. Compute a stepsize α_k and update f_{r} and $f_{\min}$ etc. by algorithm 2.1.

Step 3 (Update the estimation and compute a new search direction).
$x_{k+1} = x_k + \alpha_k d_k$, $d_{k+1} = -g_{k+1}$.

Step 4 (Compute the first trial stepsize $\alpha_{k+1}^{(1)}$).
(i)$s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.
(ii) If $s_k^{\text{T}} y_k \leqslant 0$, $\alpha_{k+1}^{(1)} = \alpha_{\max}$; otherwise $\alpha_{k+1}^{(1)} = \max\{\alpha_{\min}, \min\{s_k^{\text{T}} s_k / s_k^{\text{T}} y_k, \alpha_{\max}\}\}$.

Step 5. $k := k + 1$ and go to Step 1.

The use of the adaptive nonmonotone line search is such that the first trial stepsize is easily accepted, thus keeping the local properties of two-point stepsize gradient method around the solution point x^*. As will be shown, such an adaptive nonmonotone line search is also such that the two-point stepsize gradient method is globally convergent in real computations, while we wonder whether the ATSG algorithm converges globally in theory. Assume that the machine precision is $\varepsilon > 0$. It is known that in real computations,

$$a < b \quad \text{means} \quad a \leqslant b - \varepsilon.$$

Theorem 3.2. Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice-continuously differentiable and its level set $\widetilde{L} = \{x \in \mathbb{R}^n : f(x) \leqslant f(x_1)\}$ is bounded. Let $\{x_k\}$ be the sequence generated by algorithm 3.1 with $\epsilon = 0$. Then we have either $g(x_k) = 0$ for some finite k or

$$\liminf_{k\to\infty} \|g_k\|_2 = 0. \tag{3.1}$$

Proof. First we denote f_r, $f_{\max}$, f_c, l at the kth iteration by f_r^k, $f_{\max}^k$, f_c^k, l^k. It can be known from algorithm 3.1 that

$$f_k \leqslant f_c^k \leqslant f_{\max}^k, \quad \text{for all } k. \tag{3.2}$$

From the algorithm, we also know that the iteration sequence x_k remains in the level set $\widetilde{L}$ since $f(x_k) \leqslant f(x_1)$ for all k. Thus, using the fact that $\alpha_k^{(1)} \in [\alpha_{\min}, \alpha_{\max}]$ and the technique of choosing α_k and (3.2), we can conclude that a stepsize α_k satisfying (2.5) or (2.6) can be found after finite number of trials and hence there must exist a positive constant c_1, such that $\alpha_k \geqslant c_1$ for all k (for example, one can see the proof of theorem 2.1 in [7] for its details). It follows from $\alpha_k \geqslant c_1$, (2.5) and (2.6) that

$$f(x_{k+1}) \leqslant f_r^k - \delta c_1 \|g_k\|_2^2. \tag{3.3}$$

In the following, assuming the algorithm does not stop after finite iterations, we consider two cases.

Case 1. If $l^k < L$ for all large k, we can obtain from the algorithm that there exists an infinite subsequence $\{x_{k_i}\}$ such that

$$f(x_{k_{i+1}}) < f(x_{k_i}), \quad k_{i+1} \leqslant k_i + L. \tag{3.4}$$

According to the conditions in real computation which we have stated before the theorem, (3.4) means $f(x_{k_{i+1}}) \leqslant f(x_{k_i}) - \varepsilon$. This will contradict the fact that f is bounded below on the level set $\widetilde{L}$. So case 1 will never happen in real computations.

Case 2. If $l^k = L$ for infinite times, we define the infinite index set $\widetilde{A} = \{k_i : l^{k_i} = L, k_i < k_{i+1}\}$. Then for all $k_i < j < k_{i+1}$, f_r^j is determined by the formula (2.4). Since $\gamma_2 \geqslant 1$, we can then get

$$f_r^j \leqslant f_r^{j-1}, \quad k_i < j < k_{i+1}. \tag{3.5}$$

From (3.5), (3.2) and the definition of set $\widetilde{A}$, we know that

$$f_{\mathrm{r}}^{j} \leqslant f_{\mathrm{r}}^{k_i} \leqslant f_{\max}^{k_i}, \quad k_i < j < k_{i+1}. \tag{3.6}$$

Now assuming the theorem is not true, that is $\|g_k\|_2 \geqslant c_2$ for all k, where c_2 is a constant, we can obtain by (3.3) that

$$f(x_{k+1}) \leqslant f_{\mathrm{r}}^{k} - \epsilon_1, \tag{3.7}$$

for all k with $\epsilon_1 = \delta c_1 c_2^2$. Combining (3.6) and (3.7), we get that

$$f_j \leqslant f_{\max}^{k_i} - \epsilon_1, \quad \text{for } k_i < j \leqslant k_{i+1}. \tag{3.8}$$

Further, by the definition of $f_{\max}$ and $\widetilde{A}$, we know from (3.6) and (3.8) that

$$f_{\mathrm{r}}^{k_{i+1}} \leqslant f_{\max}^{k_{i+1}} \leqslant f_{\max}^{k_i}, \tag{3.9}$$

which implies that $\{f_{\max}^{k_i}\colon\ k_i \in \widetilde{A}\}$ is a nonincreasing sequence. It follows from (3.8), (3.9) and the nonincreasing property of $\{f_{\max}^{k_i}\colon\ k_i \in \widetilde{A}\}$ that

$$f_{\max}^{k_{i_1}} \leqslant f_{\max}^{k_{i_2}} - \epsilon_1, \quad \text{if } k_{i_1} - k_{i_2} > M. \tag{3.10}$$

So there exists a subsequence of $\{f_{\max}^{k_i}\colon\ k_i \in \widetilde{A}\}$, still denoted by $\{f_{\max}^{k_i}\}$, such that

$$f_{\max}^{k_{i+1}} \leqslant f_{\max}^{k_i} - \epsilon_1. \tag{3.11}$$

Since $\widetilde{A}$ is an infinite set, we know that the above relation contradicts the lower boundedness of f. Therefore the theorem is true. □

Relation (3.1) and the boundness of the level set $\widetilde{L}$ imply that there must exist one cluster point x^* of $\{x_k\}$ for which $\nabla f(x^*) = 0$. In many existing results on gradient method, the following convergence relation can be established:

$$\lim_{k\to\infty} \|g_k\|_2 = 0, \tag{3.12}$$

which means that every cluster point of $\{x_k\}$ is a stationary point. Since algorithm 3.1 only requires the function to be decreased every L iterations, it is easy to construct an example such that there exists a subsequence $\{k_i\}$ such that $f(x_{k_i}) \equiv f(x_1)$ and $k_{i+1} - k_i = L$. Thus it is not possible to establish the convergence relation (3.12) for algorithm 3.1.

4. Numerical comparisons

We have tested the ATSG algorithm, namely, algorithm 3.2, on an SGI Indigo workstation (Indigo 7) in double precisions with the machine error of 2×10^{-16}. Our code for the ATSG algorithm is based on those for the SPG2 algorithm in [5] and written with FORTRAN language. The initial parameters are $\delta = 10^{-4}$, $\alpha_{\min} = 10^{-30}$, $\alpha_{\max} = 10^{30}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.9$, and $\alpha_1^{(1)} = 1/\|g_1\|_\infty$. We also set L, M and P to be 3, 8 and 40,

respectively. In addition, in Step 2 of the ATSG algorithm, the new trial stepsize α_{new} is obtained by the same rule as the one in [5]. That is, at the kth iteration,

$$\alpha_{\text{new}} = \begin{cases} \bar{\alpha}, & \text{if } \alpha_{\text{old}} > 0.1\alpha_k^{(1)} \text{ and } \bar{\alpha} \in \left[0.1\alpha_k^{(1)}, 0.9\alpha_{\text{old}}\right]; \\ 0.5\alpha_{\text{old}}, & \text{otherwise,} \end{cases} \tag{4.1}$$

where α_{old} is the old trial stepsize and $\bar{\alpha}$ is the stepsize obtained by a quadratic interpolation at x_k and $x_k + \alpha_{\text{old}} d_k$. Our test problems were taken from Morè et al. [14], except "Strictly Convex 1" and "Strictly Convex 2" that are provided in [16]. The total number of the test problems are 26. The stopping condition is

$$\|g_k\|_\infty \leqslant 10^{-6}, \tag{4.2}$$

which is stronger than those normally used in real applications. The upper bound for the number of function evaluations is set to 9999.

We also compared the ATSG algorithm with the SPG2 algorithm by setting the constrained bounds in the SPG2 algorithm to infinity. We checked that both methods arrived to exactly the same solution in all problems. The numerical results are reported in table 1, where the test problems from [14] are numbered in the following way: "MGH i" means the ith problem in [14]. In addition, n denotes the dimension of the problem, and I, F are number of iterations and number of function evaluations, respectively. The number of gradient evaluations is equal to that of iterations, since no gradient evaluation is required in the line search procedure. The column R gives the number of rejected first trial stepsizes.

From table 1, we can see that for all the testing problems the ATSG algorithm requires at most the same numbers of function evaluations and gradient evaluations as the SPG2 algorithm. We can also see that, the ATSG algorithm is better than the SPG2 algorithm for 14 problems, more than a half of the testing problems. Meanwhile, the total CPU time required by the ATSG algorithm is much less than that required by the SPG2 algorithm, saving 25.5%. For some ill-conditioned problems such as "Strictly Convex 2" with $n = 10{,}000$, the ATSG algorithm is significantly better than the SPG2 algorithm. Therefore, it is safe to say that the ATSG algorithm is a very effective algorithm, which requires few more storages but behaves better than the SPG2 algorithm in solving unconstrained optimization problems.

5. Concluding remarks

In this paper, we have proposed an adaptive nonmonotone line search for the two-point stepsize gradient method. One important feature of the line search is that, the reference iteration is chosen adaptively. By setting $L = 3$, $M = 8$ and $P = 40$, we found that the ATSG algorithm performs better than the SPG2 algorithm even for ill-conditioned problems. Good numerical results were also obtained when running ATSG with $L = 3$, $M = 9$, $P = 45$. We think the good performances of this algorithm is due to accepting the prior trial steps as frequently as possible. This can be partly seen from the column R in table 1.

Table 1
Numerical Comparisons of gradient algorithms.

Problem	n	SPG2 ($M = 10$)			ATSG		
		I	F	R	I	F	R
MGH11	3	949	2507	222	478	1097	59
MGH14	4	163	329	18	119	239	5
MGH18	6	1091	2042	238	390	810	56
MGH22	16	466	776	80	158	232	11
MGH24	20	708	1939	220	277	437	26
	40	258	527	47	229	323	21
MGH28	20	907	923	8	907	923	8
	50	6967	7018	24	6967	7018	24
MGH30	50	38	39	0	38	39	0
	500	36	37	0	36	37	0
MGH31	50	30	31	0	30	31	0
	500	29	30	0	29	30	0
MGH22	100	272	468	36	189	324	18
	500	425	755	73	157	229	11
MGH25	100	1	2	0	1	2	0
	1000	1	2	0	1	2	0
MGH21	1000	53	279	8	53	278	7
	10000	53	279	8	53	278	7
MGH23	1000	56	251	2	51	53	1
	10000	64	163	2	62	64	1
MGH26	1000	89	205	9	75	90	4
	10000	83	107	9	78	94	2
Strictly	1000	5	6	0	5	6	0
Convex 1	10000	5	6	0	5	6	0
Strictly	1000	533	786	83	451	620	46
Convex 2	10000	2091	3205	329	1516	2278	193
Total CPU time (s)			166.30			l23.88	

Under the same testing conditions as in section 4, we also experimented the two-point stepsize gradient method with a direct extension of Toint's technique in [17]. In this case, relatively good numerical results were obtained with $L = 3$ (the choice for L in Toint [17] is 5). However, they are still much worse than the ones by the SPG2 algorithm for many test problems such as MGH11 with $n = 3$, MGH28 with $n = 20$, and MGH21 with $n = 1000, 10000$. For example, for problem MGH11 with $n = 3$, the total numbers of iterations and function evaluations required by the former are 1767 and 3503, respectively.

As many authors have pointed out, the most important characteristics of algorithms based on the two-point stepsize gradient method are their simplicity, efficiency and extremely low memory requirements. These properties make them very attractive for large-scale problems. In this paper, we have given some useful considerations about the problem how to combine the two-point stepsize gradient method with proper global

strategies. Nevertheless, this problem is a very important problem in real computations and we think that it still merits further studies.

Acknowledgements

The authors would like to thank the two anonymous referees for their useful suggestions and comments on this paper.

References

[1] H. Akaike, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method, Ann. Inst. Statist. Math. Tokyo 11 (1959) 1–17.
[2] J. Barzilai and J.M. Borwein, Two point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141–148.
[3] E.G. Birgin, I. Chambouleyron and J.M. Martínez, Estimation of the optical constants and the thickness of thin films using unconstrained optimization, J. Comput. Phys. 151 (1999) 862–880.
[4] E.G. Birgin and Y.G. Evtushenko, Automatic differentiation and spectral projected gradient methods for optimal control problems, Optim. Methods Softw. 10 (1998) 125–146.
[5] E.G. Birgin, J.M. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods for convex sets, SIAM J. Optim. 10(4) (2000) 1196–1211.
[6] A. Cauchy, Méthode générale pour la résolution des systèms d'equations simultanées, Comp. Rend. Sci. Paris 25 (1847) 46–89.
[7] Y.H. Dai, On the nonmonotone line search, 2000 (accepted by JOTA).
[8] Y.H. Dai and L.Z. Liao, *R*-linear convergence of the Barzilai and Borwein gradient method (1999), accepted by IMA J. Numer. Anal.
[9] G.E. Forsythe, On the asymptotic directions of the *s*-dimensional optimum gradient method, Numer. Math. 11 (1968) 57–76.
[10] A. Friedlander, J.M. Martínez, B. Molina and M. Raydan, Gradient method with retards and generalizations, SIAM J. Numer. Anal. 36 (1999) 275–289.
[11] W. Glunt, T.L. Hayden and M. Raydan, Molecular conformations from distance matrices, J. Comput. Chem. 14 (1993) 114–120.
[12] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.
[13] W.B. Liu and Y.H. Dai, Minimization algorithms based on supervisor and searcher co-operation: I – faster and robust gradient algorithms for minimization problems with stronger noises (1999), accepted by JOTA.
[14] J.J. Morè, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software 7 (1981) 17–41.
[15] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, IMA J. Numer. Anal. 13 (1993) 321–326.
[16] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM J. Optim. 7(1) (1997) 26–33.
[17] Ph.L. Toint, A nonmonotone trust region algorithm for nonlinear optimization subject to convex constraints, Math. Prog. 77 (1997) 69–94.