



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE

## FLORE

# Repository istituzionale dell'Università degli Studi di Firenze

### **Nonmonotone globalization of the finite-difference Newton-GMRES method for nonlinear equations.**

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

*Original Citation:*

Nonmonotone globalization of the finite-difference Newton-GMRES method for nonlinear equations / L. Grippo; M. Sciandrone. - In: OPTIMIZATION METHODS & SOFTWARE. - ISSN 1055-6788. - STAMPA. - 25:(2010), pp. 971-999.

*Availability:*

The webpage <https://hdl.handle.net/2158/396743> of the repository was last updated on

*Terms of use:*

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

*Publisher copyright claim:*

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)



# Nonmonotone Globalization Techniques for the Barzilai-Borwein Gradient Method

L. GRIPPO

grippe@dis.uniroma1.it

*Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Buonarroti 12,  
00185 Roma, Italy*

M. SCIANDRONE

sciandro@iasi.rm.cnr.it

*Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy*

**Abstract.** In this paper we propose new globalization strategies for the Barzilai and Borwein gradient method, based on suitable relaxations of the monotonicity requirements. In particular, we define a class of algorithms that combine nonmonotone watchdog techniques with nonmonotone linesearch rules and we prove the global convergence of these schemes. Then we perform an extensive computational study, which shows the effectiveness of the proposed approach in the solution of large dimensional unconstrained optimization problems.

**Keywords:** Barzilai-Borwein method, gradient method, steepest descent, nonmonotone techniques, unconstrained optimization

## 1. Introduction

Consider the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ & && x \in R^n \end{aligned} \tag{1}$$

where  $f : R^n \rightarrow R$  is a continuously differentiable function. The Barzilai-Borwein (BB) [1] gradient method for the solution of (1) consists essentially in a steepest descent method, where the choice of the stepsize along the negative gradient direction is derived from a two-point approximation to the secant equation underlying Quasi-Newton methods. More specifically, denoting by  $\nabla f$  the gradient of  $f$ , the BB method can be described by the iterative scheme:

$$x^{k+1} = x^k - \frac{1}{\alpha^k} \nabla f(x^k). \tag{2}$$

where the scalar  $\alpha^k$  is given either by

$$\alpha_1 = \frac{s^T y}{s^T s} \tag{3}$$

or by

$$\alpha_2 = \frac{y^T y}{s^T y} \quad (4)$$

with

$$s = x^k - x^{k-1}, \quad y = \nabla f(x^k) - \nabla f(x^{k-1}).$$

These alternative choices for  $\alpha^k$  are related to the Quasi-Newton equation

$$B^k s = y, \quad (5)$$

where  $B^k$  is a  $n \times n$  symmetric positive definite matrix approximating the Hessian matrix  $\nabla^2 f(x^k)$ . Indeed,  $\alpha_1$  can be obtained by minimizing  $\|\alpha s - y\|$ , which measures the discrepancy between the two members of (5), when  $B^k = \alpha I$ . Similarly, letting  $(B^k)^{-1} = \frac{1}{\alpha} I$ , the number  $\alpha_2$  minimizes the quantity  $\|s - \frac{1}{\alpha} y\|$ . We note also that the scalars  $\alpha_1, \alpha_2$  have been already used as scaling factors for the starting matrix in the context of limited memory Quasi-Newton algorithms (see, e.g. [12, 18, 25]) and several interpretations of them are reviewed in [12].

In [1] it has been proved that the BB method is R-superlinearly convergent in the two-dimensional quadratic case, and it has been shown, on one example, that it can be much more effective than the classical gradient method. In the strictly convex quadratic case with any number of variables, it has been demonstrated in [23] that the BB method is globally convergent and in [5] that the convergence rate is R-linear. Further results and applications concerning the quadratic case and extensions to box constrained quadratic problems have been given in several recent works (see, e.g. [8–11, 14, 15, 20]).

In the general non quadratic case, the numbers  $\alpha_1, \alpha_2$  can be unacceptably large or small (and even negative for a non convex function) and therefore we must assume that the stepsize  $\alpha^k$  computed through (3) or (4) is modified so as to satisfy a condition of the form

$$0 < \alpha_\ell \leq \alpha^k \leq \alpha_u, \quad \text{for all } k, \quad (6)$$

where  $\alpha_\ell, \alpha_u$  are prefixed numbers.

This does not ensure, in general, the convergence of algorithm (2), and hence some steplength procedure is required. As the BB stepsize does not guarantee a monotonic decrease of the objective function, a globalization strategy which accepts this stepsize as frequently as possible and retains the local properties of the method should be based on nonmonotone linesearch rules. We note also that a nonmonotone globalization technique can be useful in difficult nonlinear problems, independently of the specific local properties of the default stepsize, because of the fact that it may help escaping from steep sided valleys.

A globalization strategy based on the nonmonotone linesearch technique of [16] has been proposed and experimented in [23], and it is based on an Armijo-type linesearch on

$0 < \lambda \leq 1$  employing an acceptance condition of the form:

$$f(x^k + \lambda d^k) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} + \gamma \lambda \nabla f(x^k)^T d^k, \quad (7)$$

where  $\gamma \in (0, 1)$ ,

$$d^k = -\frac{1}{\alpha^k} \nabla f(x^k),$$

and  $\alpha^k$  is the BB stepsize, possibly modified in a way that (6) is satisfied. The computational results show that this ‘global Barzilai-Borwein’ (GGB) algorithm is competitive and sometime preferable to recent and well-known implementations of the conjugate gradient method, at least in terms of number of gradient evaluations and CPU time. However, as observed in [23], the GGB method can be inefficient in the solution of very ill-conditioned problems. In particular, we note that the acceptance condition (7) requires that at each  $k$  the point  $x^k$  must lie in the level set

$$\mathcal{L}^k = \left\{ x : f(x) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} \right\},$$

which can be too restrictive when the points  $x^{k-j}$ , for  $j = 1, \dots, \min(k, M)$  are located near the bottom of a steep valley, so that a nonmonotone line search may be not much effective in relaxing the monotonicity requirements. A consequence of this is that the behavior of the method in the solution of ill-conditioned convex problems and difficult nonlinear problems may depend critically on the choice of the starting point and on the value of  $M$ .

The need for increasing the amount of nonmonotonicity by introducing more tolerant acceptability criteria has been also pointed out in [9], through the analysis of a convex problem where the objective function contains small non quadratic terms. In the same work, with reference to the convex case, a nonmonotone line search based on the gradient norm, which does not require function evaluations, has been defined and experimented successfully. Some modifications of the GGB method, aimed at improving its efficiency and reducing the sensitivity to the linesearch parameters have been also proposed in recent works (see, e.g [6] and [19]) with promising results.

In this paper, with reference to the general nonconvex case, we define globalization schemes for descent methods where we introduce a further relaxation of the monotonicity requirements, in a way that some points can be generated out of the current level set  $\mathcal{L}^k$ , but ultimately convergence towards stationary points of  $f$  in  $\mathcal{L}^0$  is achieved and convergence towards local maxima is prevented. Nonmonotone strategies with similar objectives have been considered in [17] in the context of Newton-type methods. Here, by extending and adapting this approach, we propose new stabilization schemes that combine watchdog techniques and nonmonotone linesearch procedures. The basic idea is that of evaluating the actual reduction of the objective function by means of a ‘nonmonotone watchdog’ test that also bounds the growth of the steplength; in case of failure, we backtrack to the last accepted point and we perform a nonmonotone linesearch along a gradient related search direction. This strategy allows us to use the unmodified BB method during a finite set of

consecutive iterations and also permits the use of different formulae for the computation of the stepsize. As remarked in [10] the ideal strategy in the strictly convex quadratic case would be that of performing a sequence of iterations where  $n$  consecutive values of  $\alpha^k$  are identified with the eigenvalues of the Hessian matrix, since this would ensure termination in  $n$  steps. Although this cannot be easily realized in practice through the BB method (see, e.g., the discussion in [9]), it would seem that even in the nonconvex case the behavior of globalization techniques for the BB method is improved when a sequence of steps of the unmodified method are permitted. Thus, this feature is retained, as much as possible, in the globalization algorithms considered here.

As regards the linesearch technique, we note that, in the application to the BB method, the value of  $1/\alpha^k$  can be very small at some iterations, but an Armijo-type line search with  $\lambda \leq 1$  does not permit any increase of the stepsize. In order to overcome this limitation, we propose new nonmonotone acceptance rules that admit also occasional increases in the stepsizes, and we show that the usual convergence properties are satisfied under standard assumption.

The combination of nonmonotone watchdog techniques with nonmonotone linesearches can be realized through different algorithms; the main motivation of the present paper is that of establishing the global convergence of a few basic schemes and, at the same time, that of illustrating and evaluating some practical implementation.

The paper is organized as follows. In Section 2 we define some stabilization schemes for descent methods based on the nonmonotone acceptance criteria introduced here and we prove the global convergence under suitable assumptions on the search direction and the linesearch technique employed. In Section 3 we consider nonmonotone linesearch procedures and we show that the required conditions can be satisfied through practical algorithms. In Section 4 we describe the implementation of a gradient technique incorporating the BB method and the proposed globalization strategy. In Section 5 we report the results of an extensive computational experimentation on large scale unconstrained problems and comparisons with a reduced memory Quasi-Newton method. Finally Section 6 contains some concluding remarks.

## 2. A nonmonotone stabilization strategy

In this section we define a general nonmonotone stabilization strategy, which combines a watchdog technique with a line search approach and can be viewed as a modified version of the algorithm defined in [16] in connection with Newton-type methods. The stabilization schemes proposed here are described in terms of a sequence of maior iterations where we compute some gradient related search direction and then generate a finite set of tentative points using some 'local algorithm', which will be identified in the sequel with the BB method.

We indicate by  $x^k$  the points considered at the maior iterations. At each  $x^k$  we compute a descent direction  $d^k$  satisfying suitable conditions; then, starting from  $x^k$ , we use the local algorithm for determining the tentative points  $z_i^k$ , for  $i = 1, \dots, N$ , being  $N$  a given integer. Letting

$$z_0^k = x^k, \quad p_0^k = d^k, \quad (8)$$

we suppose that the points  $z_i^k$  are generated using the iteration

$$z_{i+1}^k = z_i^k + p_i^k, \quad i = 0, \dots, N-1, \quad (9)$$

where  $p_i^k$  are suitable search directions. In the study of global convergence no specific assumption is made on these directions for  $i \geq 1$  and the only condition on the local algorithm is that a unit (tentative) step is performed along  $d^k = p_0^k$ . In the application to the BB method the vectors  $p_i^k$ , for  $i = 0, 1, \dots, N-1$ , can be defined as scaled steepest descent directions, using the BB formulae or other formulae based on past iterations for computing the scaling factors [10].

The tentative points are accepted or rejected according to a nonmonotone watchdog rule which measures the actual reduction of the objective function with respect to some reference value. When the tentative points are rejected, we backtrack to  $x^k$  and compute a stepsize  $\lambda^k$  along  $d^k$  through a (nonmonotone) linesearch technique. The convergence of this scheme depends essentially on the conditions imposed on the search direction  $d^k$  and on the properties of the linesearch algorithm that computes  $\lambda^k$ , in case of backtracking.

We suppose that  $d^k$  satisfies the following condition.

*Condition 1.* There exist positive numbers  $c_1, c_2$  such that, for all  $k$  we have:

- (i)  $\|d^k\| \leq c_1 \|\nabla f(x^k)\|$ ;
- (ii)  $\nabla f(x^k)^T d^k \leq -c_2 \|\nabla f(x^k)\|^2$ .

Condition 1 implies that  $d^k$  is a descent direction, which is uniformly gradient related to  $x^k$ . It can be easily verified that the direction  $d^k = -(1/\alpha^k)\nabla f(x^k)$  satisfies Condition 1 provided that (6) is satisfied.

We recall from [21] the following definition.

*Definition 1.* A function  $\sigma : R^+ \rightarrow R^+$  is a forcing function if for any sequence of numbers  $t^k \subset R^+$

$$\lim_{k \rightarrow \infty} \sigma(t^k) = 0 \quad \text{implies} \quad \lim_{k \rightarrow \infty} t^k = 0.$$

As regards the linesearch procedure, we suppose that the following condition holds.

*Condition 2.* Let  $\{x^k\}$  be a sequence of points and let  $\{d^k\}$  be a sequence of search directions. Assume that  $K \subseteq \{0, 1, \dots\}$  is a subset such that  $x^{k+1} = x^k + \lambda^k d^k$  for all  $k \in K$ , where  $\lambda^k \in R$  is computed through the linesearch procedure. Then:

- (i) for every  $k \in K$  we have

$$f(x^k + \lambda^k d^k) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \sigma_l(\lambda^k \|d^k\|),$$

where  $M \geq 0$  is a prefixed integer, and  $\sigma_l : R^+ \rightarrow R^+$  is a forcing function;

- (ii) if  $K$  is an infinite subset, if the sequence  $\{f(x^k)\}$  converges and the subsequence  $\{x^k\}_K$  is bounded, it follows that

$$\lim_{k \rightarrow \infty, k \in K} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0.$$

We note that Condition 2(i) represents a nonmonotone sufficient reduction criterion on  $f$ , which also bounds the stepsize in a way that, in case of convergence, we have  $\lambda^k \|d^k\| \rightarrow 0$ . Condition 2(ii) expresses the convergence properties of the linesearch procedure and implicitly requires that the linesearch algorithm can provide ‘sufficiently large’ stepsizes. Linesearch algorithms that satisfy Condition 2 will be described and analyzed in the sequel.

In relation to the general strategy described above, different stabilization models can be defined according to the watchdog rules used for accepting or rejecting the tentative points. The simplest criterion can be that of performing a nonmonotone watchdog test at the end of the local phase, by evaluating the objective function at  $z_N^k$ . This criterion is embedded in the following algorithm model, where the integer  $M$  is the same number employed in Condition 2.

#### NonMonotone Stabilization (NMS) Algorithm 1

**Data.**  $x^0 \in R^n$ , integers  $N \geq 1$ ,  $M \geq 0$ ,  $k = 0$  and a forcing function  $\sigma : R^+ \rightarrow R^+$ .

**While**  $\nabla f(x^k) \neq 0$  **do**

**Step 1.** Compute a search direction  $d^k$  satisfying Condition 1.

**Step 2.** Compute the points  $z_1^k, \dots, z_N^k$  using the local algorithm (8), (9).

**Step 3.** **If**  $f(z_N^k) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \max_{1 \leq i \leq N} \{\sigma(\|z_i^k - x^k\|)\}$  **then**

set  $x^{k+1} = z_N^k$

**else**

compute the stepsize  $\lambda^k$  along  $d^k$  by means of a linesearch algorithm ensuring Condition 2, and set  $x^{k+1} = x^k + \lambda^k d^k$ .

**end if**

**Step 4.** Set  $k = k + 1$ .

**end while**

We note that the acceptance rule at Step 3 and Condition 2(i) on the linesearch ensure the existence of a strictly decreasing subsequence of function values and, at the same time, enforce satisfaction of the limit  $\|x^{k+1} - x^k\| \rightarrow 0$ . This is at the basis of the convergence proofs for nonmonotone methods. In the nonconvex case, the existence of a strictly decreasing subsequence of  $\{f(x^k)\}$  is also a necessary requirement for ensuring that limit points are not worse than the starting point and that asymptotic convergence towards local maxima is prevented.

When  $N > 1$ , Step 2 allows us to perform some steps without imposing conditions on the local algorithm and hence permits to exploit the local properties of the method adopted for generating the tentative points  $z_i^k$ .

The convergence analysis of Algorithm NMS1 will be carried out by analyzing first the properties of the sequence  $\{x^k\}$  generated at the maior iterations and then considering the sequences  $\{z_i^k\}$  produced by the local algorithm. We suppose that the following standard assumption holds.

**Assumption 1.** The level set  $\mathcal{L}^0 = \{x : f(x) \leq f(x^0)\}$  is compact.

We state first the following Lemma.

**Lemma 1.** Let  $\{x^k\}$  be the sequence generated by Algorithm NMS1 and denote by  $x^{l(k)}$  a point such that  $k - \min(k, M) \leq l(k) \leq k$  and that

$$f(x^{l(k)}) = \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})].$$

Then:

(i) there exists a forcing function  $\tilde{\sigma} : R^+ \rightarrow R^+$  such that

$$f(x^{k+1}) \leq f(x^{l(k)}) - \tilde{\sigma}(\|x^{k+1} - x^k\|); \quad (10)$$

(ii) the sequence  $\{f(x^{l(k)})\}$  is nonincreasing

**Proof:** For each  $k \geq 0$ , let  $l(k)$  be the integer considered in the assertion. By the instructions of the algorithm we have either that

$$x^{k+1} = x^k + \lambda^k d^k, \quad (11)$$

when  $\lambda^k$  is computed by the linesearch algorithm, or that

$$x^{k+1} = z_N^k \quad (12)$$

when  $z_N^k$  satisfies the test at Step 3. In the first case, by Condition 2(i) we have:

$$f(x^{k+1}) \leq f(x^{l(k)}) - \sigma_l(\lambda^k \|d^k\|),$$

while in the latter we obtain

$$f(x^{k+1}) \leq f(x^{l(k)}) - \max_{1 \leq i \leq N} \{\sigma(\|z_i^k - x^k\|)\} \leq f(x^{l(k)}) - \sigma(\|z_N^k - x^k\|).$$

It follows that assertion (i) holds for every  $k$ , provided that we define the function

$$\tilde{\sigma}(t) = \min\{\sigma_l(t), \sigma(t)\},$$

which is a forcing function in the sense of Definition 1.

Now, noting that  $\min(k+1, M) \leq \min(k, M) + 1$ , we can write

$$\begin{aligned} f(x^{l(k+1)}) &= \max_{0 \leq j \leq \min(k+1, M)} [f(x^{k+1-j})] \leq \max_{0 \leq j \leq \min(k, M)+1} [f(x^{k+1-j})] \\ &= \max \{f(x^{l(k)}), f(x^{k+1})\} = f(x^{l(k)}), \end{aligned}$$

where the last equality follows from (10). This establishes (ii).  $\square$

The next result is essentially based on the proof given in [16] and it is reported in the Appendix for completeness.

**Lemma 2.** *Let  $\{x^k\}$  be the sequence generated by Algorithm NMS1. Then*

- (i) *the sequence  $\{x^k\}$  belongs to the compact set  $\mathcal{L}^0$ ;*
- (ii) *the sequence  $\{f(x^k)\}$  is convergent;*
- (iii)  $\lim_{k \rightarrow \infty} \|x^{k+1} - x^k\| = 0$ .

Using the preceding result we can now establish the convergence properties of the sequence  $\{x^k\}$ .

**Proposition 1.** *Let  $\{x^k\}$  be the sequence generated by Algorithm NMS1 and suppose that the algorithm does not terminate. Then, every limit point of  $\{x^k\}$  is a stationary point of  $f$ , which is not a maximum point.*

**Proof:** Let us consider any infinite subset  $K \subseteq \{0, 1, \dots\}$  such that

$$\lim_{k \rightarrow \infty, k \in K} x^k = \bar{x}. \quad (13)$$

Suppose first that there exists an infinite subset  $K_1 \subseteq K$  such that

$$x^{k+1} = x^k + \lambda^k d^k \quad \text{for all } k \in K_1, \quad (14)$$

where  $d^k$  satisfies Condition 1 and  $\lambda^k$  is computed by a linesearch algorithm satisfying Condition 2. Using Condition 1, for all  $k \in K_1$  we can write

$$\frac{|\nabla f(x^k)^T d^k|}{\|d^k\|} \geq \frac{c_2}{c_1} \|\nabla f(x^k)\|. \quad (15)$$

Moreover, recalling Lemma 2, we have that Condition 2(ii) implies

$$\lim_{k \rightarrow \infty, k \in K_1} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0,$$

from which, taking into account (13), (15) and the continuity assumption on  $\nabla f$ , we obtain

$$\nabla f(\bar{x}) = 0. \quad (16)$$

Now assume that for all sufficiently large  $k \in K$  the test at Step 3 of Algorithm NMS1 is satisfied. This implies that:

$$\begin{aligned} f(x^{k+1}) &\leq f(x^{l(k)}) - \max_{1 \leq i \leq N} \{\sigma(\|z_i^k - x^k\|)\} \leq f(x^{l(k)}) - \sigma(\|z_1^k - x^k\|) \\ &= f(x^{l(k)}) - \sigma(\|d^k\|), \end{aligned}$$

where  $l(k)$  is the index defined in Lemma 1. Taking limits for  $k \rightarrow \infty$  and  $k \in K$  and recalling assertion (ii) of Lemma 2, it follows that  $d^k \rightarrow 0$  for  $k \rightarrow \infty, k \in K_1$ . As  $d^k$  satisfies Condition 1(ii), we obtain again (16), and we can conclude that  $\bar{x}$  is a stationary point of  $f$ .

Finally, we show that  $\bar{x}$  is not a maximum point, along the same lines followed in [16]. Let us consider the subsequence  $\{x^{l(k)}\}_K$ . As  $k - l(k) \leq M$  and  $\|x^{k+1} - x^k\| \rightarrow 0$  for  $k \rightarrow \infty$ , we have that  $\{x^{l(k)}\}_K$  converges to the same limit  $\bar{x}$ . On the other hand we have

$$f(x^{l(k'')}) < f(x^{l(k')})$$

for all  $k', k'' \in K$  and such that  $k'' \geq k' + M + 1$ . Thus we can construct a subsequence  $\{x^{l(k)}\}_{K_1}$  with  $K_1 \subseteq K$  converging to  $\bar{x}$  and such that  $\{f(x^{l(k)})\}_{K_1}$  is strictly decreasing. As  $\{f(x^{l(k)})\}$  is nonincreasing and converges to  $f(\bar{x})$ , we must have  $f(x^{l(k)}) > f(\bar{x})$  for all sufficiently large  $k \in K_1$ , so that  $\bar{x}$  cannot be a local maximum point of  $f$ .  $\square$

The next proposition characterizes the convergence properties of the sequences of tentative points produced by the local algorithm.

**Proposition 2.** *Let  $\{x^k\}$  be the sequence generated by Algorithm NMS1 and suppose that the algorithm does not terminate. Let  $z_i^k$ , for  $i = 1, \dots, N$  be the points generated at Step 2 when the test at Step 3 is satisfied so that  $x^{k+1} = z_N^k$ . Then, every limit point of each sequence  $\{z_i^k\}$  is a limit point of  $\{x^k\}$  and hence a stationary point of  $f$ , which is not a local maximizer.*

**Proof:** The assertion follows from the preceding results and the test at Step 3. Indeed, by (ii) of Lemma 2 and the test at Step 3, it follows that  $\|z_i^k - x^k\| \rightarrow 0$ . Therefore, every limit point of  $\{z_i^k\}$  is a limit point of  $\{x^k\}$  and thus Proposition 2 implies that it is a stationary point of  $f$ , which is not a local maximizer.  $\square$

We have already observed that Algorithm NMS1 does not impose any restriction on the search directions produced through the local algorithm. However, especially when  $N$  is relatively large, it could be convenient to check whether the tentative points are leaving the region of interest or violate some reasonable condition. A simple technique for introducing this kind of modification, while retaining the convergence properties established above, can

be that of defining an adaptive rule for terminating prematurely the inner iterations at Step 2 on the basis of some criterion.

Formally, if we denote by  $i^k$ , with  $1 \leq i^k \leq N$ , the index of the first tentative point that violates a given condition, we can set

$$p_i^k = 0 \quad \text{for } i = i^k - 1, \dots, N - 1.$$

This implies that Step 2 will terminate at the point  $z_N^k = z_{i^k-1}^k$ , which will be accepted or rejected on the basis of the watchdog test of Step 3. Using this convention, the convergence results remain unchanged. A criterion for terminating the inner steps could be based, for instance, on the evaluation of  $\|p_i^k\|$ ; more specifically, we can terminate Step 2 whenever

$$\begin{cases} \|d^k\| \geq \mu \|p_{N-1}^{k-1}\| & (\text{or } \|d^k\| \geq \mu \|d^{k-1}\| \text{ if } z_N^{k-1} \text{ has been rejected}) & \text{for } i = 0 \\ \|p_i^k\| \geq \mu \|p_{i-1}^k\| & & \text{for } i > 0. \end{cases} \quad (17)$$

where  $\mu$  is a suitably large constant. Weaker conditions could be based on the comparison of  $\|p_i^k\|$  with the maximum value of the norm of the directions used in a finite set of previous steps. Similar conditions could be also given in terms of the gradient norm  $\|\nabla f(z_i^k)\|$ , when available. It is also evident that the convergence properties are preserved if a linesearch along  $d^k$  is performed even when this would not be required by Algorithm NMS1; in fact this corresponds to a fictitious iteration where all tentative points are rejected. Possible motivations for the use of a linesearch could be the fact that  $\|x^k - x^{k-1}\|$  is ‘small’ and we estimate that a good reduction of  $f$  could be obtained by searching along  $d^k$  with stepsizes greater than one. Note, however, that a linesearch that admits an increase in the stepsize can be useful also in case of backtracking, when  $\|d^k\|$  is small and the failure of the watchdog test is due to the last steps of the local algorithm.

For large values of  $N$ , a possible disadvantage of the stabilization schemes based on Algorithm NMS1 can be that, in case of backtracking, we reject all the tentative points  $z_1, \dots, z_N$ , while the acceptance criterion of Step 3 could have been satisfied at some of these points. As a consequence of this, we may have the need of repeating the same computations performed during several of the previous inner steps. To avoid this drawback, we can define a modified scheme by controlling the objective function values at each step of the local phase.

More specifically, we can define the following algorithm model, where the objective function is evaluated at every tentative points  $z_i$ , with  $i = 1, \dots, N$ , and the nonmonotone acceptance test is performed in correspondence to each of these points.

### NonMonotone Stabilization (NMS) Algorithm 2

**Data.**  $x^0 \in R^n$ , integers  $N \geq 1$ ,  $M \geq 0$ ,  $k = 0$  and a forcing function  $\sigma : R^+ \rightarrow R^+$ .

**While**  $\nabla f(x^k) \neq 0$  **do**

**Step 1.** Compute a direction  $d^k$  satisfying Condition 1, set *linesearch*=true.

**Step 2.** **For**  $i = 1, N$

        Compute  $z_i^k$ , using algorithm (8), (9)

**If**  $f(z_i) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \max_{1 \leq h \leq i} \{\sigma(\|z_h^k - x^k\|)\}$  **then**  
 set  $x^{k+1} = z_i$ ,  $linsearch=false$  and exit from Step 2.  
**endif**  
**End For**  
**Step 3. If**  $linsearch=true$  **then**  
 compute the stepsize  $\lambda^k$  along  $d^k$  by means of a linesearch algorithm  
 ensuring Condition 2, and set  $x^{k+1} = x^k + \lambda^k d^k$   
**endif**  
**Step 4.** Set  $k = k + 1$ .  
**End While**

In comparison with Algorithm NMS1, we may note that we backtrack to  $x^k$  only when  $N$  consecutive tentative points  $z_i$  have been rejected.

It can be easily verified that the convergence properties of Algorithm NMS2 are exactly the same established in Propositions 1 and 2; in fact when Algorithm NMS2 is restarted from  $z_{i^k}^k$  (rather than from  $x^k$ ) in correspondence to some index  $i^k$  of the inner cycle, we can recast Algorithm NMS2 into the scheme of Algorithm NMS1 by assuming that the local algorithm generates directions  $p_i^k = 0$ , for  $i \geq i^k$ , so that  $z_N^k = z_{i^k}^k$  will be accepted by the watchdog test. As in Algorithm NMS1, we can introduce suitable criteria for terminating prematurely the inner cycle. In particular, as now the objective function values  $f(z_i^k)$  are available, we can terminate the inner cycle whenever the increase in the objective function with respect to the reference value, that is the quantity

$$f(z_i^k) - \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\},$$

is unacceptably large.

Moreover, also in the case of Algorithm NMS2 we could specify suitable criteria for employing linesearches that admit the possibility of increasing the stepsize when  $\|x^k - x^{k-1}\|$  is small and a good reduction of  $f$  along  $d^k$  can be expected.

### 3. A nonmonotone line search

In this section we define new nonmonotone linesearch rules that admit also occasional increases in the stepsize and satisfy the conditions stated in the preceding section. When  $d^k$  is a descent direction satisfying Condition 1, it can be verified that an Armijo-type linesearch based on (7) and starting with  $\lambda = 1$  allow us to satisfy Condition 2, but does not permit stepsizes greater than the starting one. To overcome this limitation, a first possibility could be that of defining a nonmonotone version of the known Armijo-Goldstein techniques for increasing, when needed, the stepsize  $\lambda$ , with the additional requirement that a constant upper bound on the stepsize is imposed. An alternative approach could be that of defining a nonmonotone version of the derivative-free linesearch technique proposed in [7], by

imposing a condition of the form:

$$f(x^k + \lambda d^k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \gamma \lambda^2 \|d^k\|^2,$$

which ensures satisfaction of Condition 2(i), without requiring a prefixed bound on  $\lambda$ .

Both approaches can be combined into a single scheme, which may have some interest in its own and allow us to avoid repetitions of similar arguments in the convergence analysis.

We suppose that  $x^k$  is an element of a sequence of points in  $R^n$ , and that  $d^k \in R^n$  is a descent direction for  $f$  at  $x^k$ . In order to simplify notation, at each  $k$  we set  $f^k = f(x^k)$  and we define

$$F^k = \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] \quad \Delta^k = \psi \left( \frac{|\nabla f(x^k)^T d^k|}{\|d^k\|} \right), \quad (18)$$

where  $M > 0$  is a prefixed integer, and  $\psi : R^+ \rightarrow R^+$  is a forcing function.

A sufficient reduction of  $f$  with respect to the reference value is imposed through a condition of the form

$$f(x^k + \lambda d^k) \leq F^k + \gamma_1 \lambda \nabla f(x^k)^T d^k - \gamma_2 \lambda^2 \|d^k\|^2, \quad (19)$$

where  $\gamma_1, \gamma_2$  are nonnegative constants that satisfy

$$0 \leq \gamma_1 < 1, \quad 0 \leq \gamma_2, \quad 0 < \gamma_1 + \gamma_2. \quad (20)$$

In the line search described below the unit stepsize is accepted, unless one of the following two situations is detected:

- (i) condition (19) is not satisfied;
- (ii) condition (19) is satisfied, but the tentative step  $\|d^k\|$  is relatively small, and, at the same time, the value of  $f(x^k + d^k)$  is smaller than  $f^k$ .

In case (i) the stepsize is reduced using any safeguarded interpolation technique until the acceptance condition is verified. In case (ii) we may attempt to obtain a further reduction of the objective function with a larger step; therefore the stepsize is increased, using some safeguarded extrapolation formula, provided that a significant reduction of  $f$  is obtained and condition (19) is not violated. These rules are defined formally in the following model.

### Nonmonotone Line Search (NLS) Algorithm

**Data.**  $F^k$  and  $\Delta^k$  defined as in (18), and parameters:

$$\gamma_1, \gamma_2 \text{ satisfying (20), } 0 < \theta_l < \theta_u < 1, \quad 1 < \sigma_l < \sigma_u.$$

**Step 0.** If  $\gamma_2 = 0$  then choose a number  $\bar{\lambda} \geq 1$ , otherwise set  $\bar{\lambda} = +\infty$ .

**Step 1.** Set  $\lambda = 1$ .

**Step 2.** While  $f(x^k + \lambda d^k) > F^k + \gamma_1 \lambda \nabla f(x^k)^T d^k - \gamma_2 \lambda^2 \|d^k\|^2$  choose  $\theta \in [\theta_l, \theta_u]$  and set  $\lambda = \theta \lambda$ .

**Step 3.** If  $\lambda < 1$  set  $\lambda^k = \lambda$  and exit.

**Step 4.** If  $\|d^k\| \geq \Delta^k$  or  $f(x^k + d^k) \geq f^k$  then set  $\lambda^k = 1$  and exit; otherwise choose  $\sigma \in [\sigma_l, \sigma_u]$ .

**Step 5.** While  $\sigma \lambda \leq \bar{\lambda}$  and

$$f(x^k + \sigma \lambda d^k) < \min \{f(x^k + \lambda d^k), f^k + \gamma_1 \sigma \lambda \nabla f(x^k)^T d^k - \gamma_2 (\sigma \lambda)^2 \|d^k\|^2\}$$

set  $\lambda = \sigma \lambda$  and choose  $\sigma \in [\sigma_l, \sigma_u]$ .

**Step 6.** Set  $\lambda^k = \lambda$  and exit.

It can be shown that Algorithm NLS is well defined and computes a ‘sufficiently large’ steplength that guarantees a ‘sufficient decrease’ of  $f$ , in a way that Condition 2 of the preceding section is satisfied. More specifically, under somewhat weaker assumptions on  $f$  and  $d^k$ , we can state the following propositions whose proofs are reported in the Appendix.

**Proposition 3.** Assume that  $f$  is bounded below on  $R^n$  and that  $\nabla f(x^k)^T d^k < 0$ .

Then, Algorithm NLS determines, in a finite number of iterations, a stepsize  $\lambda^k$  such that

$$f(x^k + \lambda^k d^k) \leq F^k + \gamma_1 \lambda^k \nabla f(x^k)^T d^k - \gamma_2 (\lambda^k)^2 \|d^k\|^2,$$

and at least one of the following conditions holds:

$$\lambda^k < 1 \text{ and } f\left(x^k + \frac{\lambda^k}{\theta^k} d^k\right) > F^k + \gamma_1 \frac{\lambda^k}{\theta^k} \nabla f(x^k)^T d^k - \gamma_2 \left(\frac{\lambda^k}{\theta^k}\right)^2 \|d^k\|^2 \quad (21)$$

$$\lambda^k = 1 \text{ and } \|d^k\| \geq \Delta^k \quad (22)$$

$$\lambda^k = 1 \text{ and } f(x^k + d^k) \geq f^k \quad (23)$$

$$\gamma_2 = 0 \text{ and } \sigma^k \lambda^k > \bar{\lambda} \quad (24)$$

$$f(x^k + \sigma^k \lambda^k d^k) \geq \min\{f(x^k + \lambda^k d^k), f^k + \gamma_1 \sigma^k \lambda^k \nabla f(x^k)^T d^k - \gamma_2 (\sigma^k \lambda^k)^2 \|d^k\|^2\} \quad (25)$$

where  $\theta^k \in (\theta_l, \theta_u)$  and  $\sigma^k \in (\sigma_l, \sigma_u)$ .

**Proposition 4.** Assume that  $f$  is bounded below in  $R^n$ ; let  $\{x^k\}$  be a sequence of points in  $R^n$  and let  $K$  be an infinite index set, such that  $x^{k+1} = x^k + \lambda^k d^k$ , for all  $k \in K$  where  $d^k \in R^n$  and  $\lambda^k$  is the stepsize computed by means of Algorithm NLS. Suppose that:

– either  $\gamma_2 > 0$  and the direction  $d^k$  satisfies  $\nabla f(x^k)^T d^k < 0$ ,

– or that  $\gamma_2 = 0$  and the direction  $d^k$  satisfies Condition 1(ii).

Then:

(i) for every  $k \in K$  we have

$$f(x^{k+1}) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \eta(\lambda^k \|d^k\|)^2$$

where  $\eta$  is a positive number;

(ii) if the sequence  $\{f^k\}$  converges and the subsequence  $\{x^k\}_K$  is bounded, we have

$$\lim_{k \rightarrow \infty, k \in K} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = 0.$$

The preceding proposition ensures that Algorithm NLS satisfies Condition 2, in correspondence to a forcing function of the form  $\sigma_l(t) = \eta t^2$ , for some  $\eta > 0$ , and hence it can be used within the stabilization schemes considered there.

Algorithm NLS can be modified in different ways without affecting the convergence properties. In particular, when the steplength must be increased a more accurate search can be performed by replacing Step 5 with a standard linesearch technique based on Armijo-Goldstein or Wolfe conditions [13]. It is only required that the condition (19) is still valid. However, the computational experimentation performed on a large set of test problems seems to indicate that no significant improvement can be obtained, in the context of the stabilization strategy considered here, by employing more sophisticated line search rules.

#### 4. Implementation of a globalization strategy for the BB method

With reference to the stabilization scheme defined in Algorithm NMS1, we describe here the implementation of a nonmonotone gradient algorithm incorporating the BB method. In particular, we will specify the search directions used at Steps 1, 2 for producing the tentative points  $z_1^k, \dots, z_N^k$ , the nonmonotone watchdog test at Step 3, the nonmonotone line search procedure, and the stopping criterion. In order to simplify the notation, we omit the superscript  $k$ .

##### 4.1. Search directions

The search directions are generated according to the BB method by alternating, whenever possible, formulas (3), (4) for the computation of  $\alpha$ . More specifically, at each major iteration  $k > 0$ , and for each  $i \in \{0, \dots, N-1\}$  first we compute

$$\alpha_1 = \frac{s^T y}{s^T s} \quad \alpha_2 = \frac{y^T y}{s^T y},$$

where  $s = z_i^k - z_{i-1}^k$ ,  $y = \nabla f(z_i^k) - \nabla f(z_{i-1}^k)$ , being  $z_0^k = x^k$ , and  $z_{-1}^k = z_{N-1}^{k-1}$  when  $z_N^{k-1}$  is accepted, or  $z_{-1}^k = x^{k-1}$  in case of backtracking. Then we choose  $\alpha \in \{\alpha_1, \alpha_2\}$  in such a way that

$$\alpha_\ell \leq \alpha \leq \alpha_u, \tag{26}$$

where  $\alpha_\ell$  and  $\alpha_u$  are positive numbers defined by

$$\alpha_\ell = 10^{-5} \max \left\{ 10^{-5}, \frac{\|\nabla f(z_i)\|}{1 + \|x^0\|} \right\} \quad \alpha_u = 10^{10} \frac{\|\nabla f(x^0)\|}{1 + \|x^0\|}.$$

When in successive iterates both  $\alpha_1$  and  $\alpha_2$  satisfy (26), we alternate between  $\alpha_1$  and  $\alpha_2$  in the choice of  $\alpha$ . We set  $\alpha = \|\nabla f(z_i)\|$  whenever neither  $\alpha_1$  nor  $\alpha_2$  satisfy (26). Once  $\alpha$  has been computed, we set

$$z_{i+1} = z_i - \frac{1}{\alpha} \nabla f(z_i).$$

If both  $\alpha_1$  and  $\alpha_2$  have been rejected, then Step 2 is terminated prematurely at  $z_{i+1}$  and we set  $z_N = z_{i+1}$ , so that the watchdog test is performed at this point.

#### 4.2. Watchdog test

The watchdog test has been implemented according to the following rule

$$f(z_N) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \beta \max\{\|p_0\|, \dots, \|p_{N-1}\|\}, \quad (27)$$

where  $\beta = 10^{-4}$  and  $p_i$ , with  $i = 0, \dots, N-1$  are the search directions computed at Steps 1–2. It is easily seen that a point  $z_N$  satisfying (27) will satisfy the watchdog test of Step 3, for a suitable choice of the forcing function  $\sigma$ . Indeed, as

$$\max_{i=1, \dots, N} \{\|z_i - x^k\|\} \leq \sum_{i=0}^{N-1} \|p_i\| \leq N \max\{\|p_0\|, \dots, \|p_{N-1}\|\},$$

it follows that

$$f(z_N) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \beta \max\{\|p_0\|, \dots, \|p_{N-1}\|\}$$

implies

$$f(z_N) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x^{k-j})\} - \max_{1 \leq i \leq N} \{\sigma(\|z_i - x^k\|)\}$$

with  $\sigma(t) = \frac{\beta}{N}t$ .

#### 4.3. Nonmonotone line search procedure

We have implemented Algorithm NLS with  $\gamma_1 = 0$  and

$$\Delta^k = 10^{-2}(1 + \|x^0\|),$$

which corresponds to choosing the forcing function  $\psi$  in (18) as a constant. The parameters in the line search algorithm have been set as follows

$$M = 20 \quad \gamma_2 = 10^{-4} \quad \theta_l = 0.1 \quad \theta_u = 0.5 \quad \sigma_l = 1.5 \quad \sigma_u = 5$$

At Steps 2 and 5 the scalars  $\theta$  and  $\sigma$  are computed by setting  $\theta = \min\{\theta_u, \max\{\theta_l, \theta^*\}\}$ ,  $\sigma = \min\{\sigma_u, \max\{\sigma_l, \sigma^*\}\}$ , where  $\theta^*$  and  $\sigma^*$  are computed by means of a quadratic interpolation formula using  $f(x^k)$ ,  $\nabla f(x^k)^T d^k$ , and the function value  $f(x^k + \lambda d^k)$  at the tentative point.

#### 4.4. Stopping criterion

In correspondence to the maior iterations, we used the stopping criterion

$$\|\nabla f(x^k)\| \leq \eta(1 + |f(x^k)|). \quad (28)$$

However, at the tentative points  $z_i^k$  when  $\|\nabla f(z_i^k)\| \leq \eta(1 + |f(x^k)|)$  we compute also  $f(z_i^k)$  and we terminate the iterations when we have both  $f(z_i^k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})]$  and  $\|\nabla f(z_i^k)\| \leq \eta(1 + |f(z_i^k)|)$ . This guarantees that the termination does not occur at a point out of the current level set  $\mathcal{L}^k$ .

## 5. Numerical results

In this section we present the numerical results obtained with Algorithm NMS1 and we compare the performance of the algorithm with that of a reduced-memory quasi-Newton method (routine E04DGF of NAG library, with the default values of the parameters and with the addition of an external termination criterion based on (28)). All the runs were carried out on an IBM RISC System/6000 375 in double precision FORTRAN.

The first set of test problems is the same used in [23]. The results obtained are shown in Table 1, where we report the number of function evaluations ( $n_f$ ) and the number of gradient evaluations ( $n_g$ ) required by Algorithm NMS1 (for  $N = 2$  and  $N = 20$ ) and by the E04DGF routine to satisfy the stopping criterion (28) with  $\eta = 10^{-6}$ . We note that in Algorithm NMS1 the number  $n_g$  includes also the number of gradient evaluations performed in the inner iterations of the watchdog process. In some problems algorithm E04DGF terminates because of the internal stopping criteria without satisfying (28), and this is indicated in the table by \*.

We may note that the behavior of Algorithm NMS1 is not greatly influenced by the choice of  $N$ , and the main difference between the cases  $N = 2$  and  $N = 20$  is the fact that the number of function evaluations is much smaller for  $N = 20$ . However, this is not very relevant since in both cases we have a number of function evaluations lower than the number of gradient evaluations, so that the computational cost depends essentially on  $n_g$ . On the other hand, in some problems the choice  $N = 2$  has the effect of reducing the number of gradient evaluations. For  $N$  ranging from 10 to 100 we obtained in most of problems the same number of gradient evaluations corresponding to the choice  $N = 20$ ; small differences were detected only in three problems. The only significant difference was observed for  $N = 100$

Table 1. Complete results for the first set of test problems.

Problem	$n$	NMS1 ( $N=2$ )		NMS1 ( $N=20$ )		E04DGF
		$n_f$	$n_g$	$n_f$	$n_g$	$n_f-n_g$
Stricly Convex 1	100	5	7	3	7	7
	1000	5	7	3	7	7
	10000	5	7	3	7	7
Stricly Convex 2	100	27	50	5	50	43
	500	40	76	6	76	96
	1000	40	77	6	77	116
Brown	100	9	13	13	13	17*
	1000	5	4	4	4	16*
Trigonometric	100	49	71	18	115	75
	1000	42	76	8	76	66
	10000	48	86	10	86	78
Broyden tridiagonal	100	18	33	4	33	96
	1000	21	39	4	39	132
	3000	20	36	4	36	136
Oren's Power	100	56	109	8	109	416*
	1000	118	216	20	300	292*
	10000	417	677	61	1046	476*
Extended Rosenbrock	100	45	45	11	90	63
	1000	57	88	5	52	92
	10000	93	120	12	42	101
Penalty I	100	24	45	5	45	67
	1000	26	46	12	46	34
	10000	25	34	10	34	83
Tridiagonal 1	100	79	155	10	155	83*
	1000	315	523	24	438	145*
Variably dimensioned	100	25	46	12	46	51
	1000	42	65	24	65	73
Extended Powell	100	109	174	11	180	116
	1000	73	142	10	142	189
Generalized Rosenbrock	100	725	989	62	996	341*
	500	2242	3299	208	3338	1436*
Extended ENGLV1	100	13	22	7	22	70
	1000	13	22	8	22	73
	10000	15	15	10	15	95
Extended Freudenstein and Roth	100	50	86	7	95	107
	1000	92	123	7	91	204
	10000	37	62	10	62	533
Wrong extended wood	100	34	65	6	65	104*
	1000	28	52	6	52	171

in Trigonometric problem with dimension  $n = 100$ , where convergence is obtained after 235 gradient evaluations.

Algorithm NMS1 appears to be comparable with algorithm E04DGF in terms of gradient evaluations, but in some difficult problems (Oren's Power, Tridiagonal, Generalized Rosenbrock) algorithm E04DGF is more efficient, although the results are not precisely comparable because of the fact that, in these cases, the termination of the algorithms is due to different criteria. In comparison with the results of the GBB method reported in [23], we observe that the number of function evaluations is greatly reduced and it never exceeds the number of gradient evaluations; moreover, also the number of gradient evaluations is significantly reduced in correspondence to the problems where the line search procedure of the GBB method is active.

A few experiments have been performed for evaluating the effect of adopting different criteria for the choice of  $\alpha$ , and it would seem that the choice of alternating between  $\alpha_1$  and  $\alpha_2$  gives consistent improvements with respect to the case where a single formula is employed.

We have considered a second set of 95 test problems taken from the CUTE collection [3], with dimension  $n$  ranking from 1000 to 10000. We used the value  $\eta = 10^{-5}$  in the stopping criterion (28), and we imposed a maximum number of gradient evaluations equal to 5000. The failures due to this bound were

- 5 for Algorithm NMS1 with  $N = 2$ ;
- 12 for Algorithm NMS1 with  $N = 20$ ;
- 4 for the E04DGF routine.

On 18 problems, the E04DGF routine terminated because of the internal stopping rules without satisfying the stopping criterion (28). The cumulative results obtained for the remaining 66 problems (where all the algorithms terminate satisfying the criterion (28) with  $n_g < 5000$ ) are shown in Table 2. In this table, the number of function evaluations represents the total number of function evaluations needed to solve all these problems, and the same is for the number of gradient evaluations. The complete results are reported in Table 3.

From Tables 2 and 3 it would appear that the implementation of the BB method proposed here is competitive with a reduced memory quasi-Newton method suitable for large scale problems. However, it can not be claimed that the algorithm is superior to the quasi-Newton method, especially when a high level of accuracy is required.

On the same set of test problems we have also evaluated the effect of the expansion step in the line search algorithm. In particular, we have tested Algorithm NMS1 (with  $N = 2$ ), where we have set  $\bar{\lambda} = 1$  in the line search procedure, so that, Step 5 (the expansion step) of Algorithm NLS is never performed.

In this case we obtained 6 failures due to the maximum number of gradient evaluations, and the cumulative results are reported in Table 4. By comparing the results of Table 2 with

Table 2. Cumulative results for a selection of CUTE problems.

	NMS1 ( $N = 2$ )	NMS1 ( $N = 20$ )	E04DGF
Function evaluations	10050	1511	19478
Gradient evaluations	14706	15824	19478

Table 3. Complete results for a selection of CUTE problems.

Problem	$n$	NMS1 ( $N=2$ )		NMS1 ( $N=20$ )		E04DGF
		$n_f$	$n_g$	$n_f$	$n_g$	$n_f-n_g$
ARWHEAD	1000	6	9	3	9	16
ARWHEAD	5000	6	9	3	9	23
BDQRTIC	1000	42	80	6	80	231
BROYDN7D	1000	260	458	28	461	425
CRAGGLVY	1000	60	110	8	121	724
CRAGGLVY	5000	145	241	20	204	1543
DIXMAANA	1500	7	11	6	11	25
DIXMAANA	3000	12	9	12	9	23
DIXMAANB	1500	8	12	6	12	28
DIXMAANB	3000	12	7	11	7	33
DIXMAANC	1500	8	13	6	13	25
DIXMAANC	3000	13	8	12	8	23
DIXMAAND	1500	9	15	7	15	47
DIXMAAND	3000	13	9	12	9	45
DIXMAANE	1500	134	234	14	236	159
DIXMAANE	3000	132	246	22	246	188
DIXMAANF	1500	118	199	15	196	159
DIXMAANF	3000	161	236	21	227	273
DIXMAANG	1500	107	181	20	226	129
DIXMAANG	3000	88	159	17	159	159
DIXMAANH	1500	101	184	20	198	408
DIXMAANH	3000	190	327	21	226	204
DIXMAANI	1500	734	1131	64	1226	904
DIXMAANI	3000	493	820	64	1099	1522
DIXMAANJ	1500	129	235	17	234	274
DIXMAANJ	3000	187	265	21	232	247
DIXMAANK	1500	144	229	18	199	246
DIXMAANK	3000	155	258	22	251	408
DIXMAANL	1500	104	190	18	200	741
DIXMAANL	3000	119	189	19	182	289
EDENSCH	2000	17	17	11	17	563
ENGVAL1	1000	12	20	10	20	61
ENGVAL1	5000	15	14	10	14	63
FLETGBV3	1000	20	24	16	40	11
FMINSURF	1024	373	508	27	487	279
FMINSURF	5625	1476	1184	77	1284	572
FREUROTH	1000	31	49	12	67	144

(Continued on next page.)

Table 3. (Continued).

Problem	$n$	NMS1 ( $N=2$ )		NMS1 ( $N=20$ )		E04DGF
		$n_f$	$n_g$	$n_f$	$n_g$	$n_f-n_g$
FREUROTH	5000	25	33	24	51	319
LIARWHD	1000	35	61	8	61	36
MOREBV	1000	95	124	11	98	509
MOREBV	5000	42	43	8	40	327
NCB20B	1000	66	114	8	113	152
NONCVXU2	1000	737	1417	79	1249	1327
NONCVXUN	1000	575	1082	85	1270	1880
NONDIA	1000	9	15	10	15	73
NONDIA	5000	11	19	15	19	71
NONDIA	10000	8	9	5	9	66
NONDQUAR	1000	2064	2742	352	3529	1765
POWELLSG	1000	60	116	8	116	153
POWELLSG	5000	97	156	16	152	244
POWELLSG	10000	85	155	16	155	160
POWER	1000	145	287	24	287	287
SCHMVETT	1000	15	27	4	27	76
SCHMVETT	5000	13	23	4	23	75
SCHMVETT	10000	10	17	3	17	60
SROSENBR	1000	13	22	11	22	58
SROSENBR	5000	13	23	12	23	32
SROSENBR	10000	17	24	8	24	40
TOINTGSS	1000	9	14	13	14	8
TOINTGSS	5000	9	5	8	5	6
TOINTGSS	10000	9	5	8	5	6
TQUARTIC	5000	69	42	28	45	27
TQUARTIC	10000	79	66	8	46	24
VAREIGVL	1000	44	84	16	84	94
WOODS	1000	23	42	9	42	174
WOODS	10000	32	49	14	49	215

those of Table 4, it appears that occasional increases of the initial stepsize during the line search may give some advantages in terms of computational cost.

Finally, a few experiments have been performed with an implementation of Algorithm NMS2 with  $N=20$ . In the first set of test problems (indicated in Table 1) the only differences observed are those reported in Table 5.

With reference to the set of CUTE problems, we observed for Algorithm NMS2 8 failures. In the 66 problems reported in Table 3, the only differences between algorithms NMS1 and NMS2 were observed for problem NONDQUAR with  $n=1000$ , where  $n_g=2515$ , and for problem TQUARTIC with  $n=5000$ , where  $n_g=149$ . On the whole, it would appear

Table 4. Cumulative results without expansion step.

NMS1 ( $N = 2$ )	
Function evaluations	10713
Gradient evaluations	15894

Table 5. Results of Algorithm NMS1 and Algorithm NMS2.

Problem	NMS1 ( $N = 20$ )		NMS2 ( $N = 20$ )
	$n$	$n_g$	$n_g$
Trigonometric	100	104	51
Generalized Rosenbrock	100	996	943
	500	3338	3221

that Algorithm NMS2 is more robust, but this is paid with a considerable increase in the number of function evaluations, which becomes approximately equal to the number of gradient evaluations.

## 6. Concluding remarks

The main result of this paper is that of demonstrating that the global convergence of the Barzilai-Borwein method can be guaranteed through a nonmonotone globalization strategy that introduces only very limited perturbations of the local properties of the method. A useful feature of the proposed strategy is that of permitting a sequence of steps where different formulae for the computation of the stepsize can be used. The resulting algorithm can be implemented through a very simple code and appears to be comparable in many large dimensional problems with more sophisticated reduced memory quasi-Newton methods. Possible improvements of the results reported here could be obtained by adopting some of the rules suggested in [10] and [24] for the choice of the stepsize, and by performing a suitable tuning of the parameters. In particular, the definition of the bounds for accepting the BB stepsize and the choice of the stepsize when these bounds are violated may deserve some further attention.

## Appendix

### *Proof of Lemma 2*

For each  $k \geq 0$ , let  $l(k)$  be an integer such that  $k - \min(k, M) \leq l(k) \leq k$  and that

$$f(x^{l(k)}) = \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})].$$

By Lemma 1(i) there exists a forcing function  $\tilde{\sigma}$  such that

$$f(x^{k+1}) \leq f(x^{l(k)}) - \tilde{\sigma}(\|x^{k+1} - x^k\|). \quad (29)$$

As  $x^{l(0)} = x^0$ , (29) and (ii) of Lemma 1 imply that  $f^k \leq f(x^0)$  for all  $k$ , so that the sequence  $\{x^k\}$  belongs to the compact set  $\mathcal{L}^0$  and this proves assertion (i).

Thus the non increasing sequence  $\{f(x^{l(k)})\}$  admits a limit for  $k \rightarrow \infty$ . Using (29), where  $k$  is replaced by  $l(k) - 1$ , we can write

$$f(x^{l(k)}) \leq f(x^{l(k)-1}) - \tilde{\sigma}(\|x^{l(k)} - x^{l(k)-1}\|), \quad (30)$$

and hence, taking limits and recalling the definition of forcing function, it follows that

$$\lim_{k \rightarrow \infty} \|x^{l(k)} - x^{l(k)-1}\| = 0. \quad (31)$$

Now let  $\hat{l}(k) = l(k + M + 2)$ . First we show, by induction, that for every  $j \geq 1$  we have

$$\lim_{k \rightarrow \infty} \|x^{\hat{l}(k)-j+1} - x^{\hat{l}(k)-j}\| = 0 \quad (32)$$

and

$$\lim_{k \rightarrow \infty} f(x^{\hat{l}(k)-j}) = \lim_{k \rightarrow \infty} f(x^{l(k)}). \quad (33)$$

If  $j = 1$ , using (31) and the fact that  $\{\hat{l}(k)\} \subset \{l(k)\}$ , we get (32). Recalling that

$$\lim_{k \rightarrow \infty} f(x^{\hat{l}(k)}) = \lim_{k \rightarrow \infty} f(x^{l(k)}),$$

we have that also (33) is satisfied, as  $f$  is uniformly continuous on  $\mathcal{L}^0$ .

Then, assume that (32) and (33) hold for a given  $j$ . By (29) we can write

$$f(x^{\hat{l}(k)-j}) \leq f(x^{l(\hat{l}(k)-j-1)}) - \tilde{\sigma}(\|x^{\hat{l}(k)-j} - x^{l(\hat{l}(k)-j-1)}\|).$$

Taking limits for  $k \rightarrow \infty$  and recalling (33) we obtain

$$\lim_{k \rightarrow \infty} \|x^{\hat{l}(k)-j} - x^{l(\hat{l}(k)-j-1)}\| = 0.$$

The uniform continuity of  $f$  on  $\mathcal{L}^0$  and the assumptions made imply

$$\lim_{k \rightarrow \infty} f(x^{\hat{l}(k)-j-1}) = \lim_{k \rightarrow \infty} f(x^{\hat{l}(k)-j}) = \lim_{k \rightarrow \infty} f(x^{l(k)}).$$

Then, (32) and (33) holds for any given  $j \geq 1$ . On the other hand, for every  $k$  we have

$$\begin{aligned} x^{\hat{l}(k)} &= x^{k+1} + (x^{k+2} - x^{k+1}) + \dots + (x^{\hat{l}(k)} - x^{\hat{l}(k)-1}) \\ &= x^{k+1} + \sum_{j=1}^{\hat{l}(k)-k-1} x^{\hat{l}(k)-j+1} - x^{\hat{l}(k)-j}. \end{aligned} \quad (34)$$

As  $\hat{l}(k) - k - 1 = l(k + M + 2) - k - 1$  and  $l(k + M + 2) \leq k + M + 2$ , it follows that  $\hat{l}(k) - k - 1 \leq M + 1$ , so that (34) and (32) imply

$$\lim_{k \rightarrow \infty} \|x^{k+1} - x^{\hat{l}(k)}\| = 0. \quad (35)$$

As  $\{f(x^{l(k)})\}$  admits a limit, it follows from the uniform continuity of  $f$  on  $\mathcal{L}^0$  that

$$\lim_{k \rightarrow \infty} f^k = \lim_{k \rightarrow \infty} f(x^{l(k)}),$$

which proves assertion (ii). Assertion (iii) follows from (29) and assertion (ii).  $\square$

### *Proof of Proposition 3*

In order to prove that the algorithm terminates we must show that it does not cycle at Step 2 or at Step 5. Consider first the cycle at Step 2, and let  $h$  be a counter of the inner iterations of the cycle; we can write

$$\lambda_h = \prod_{j=1}^h \theta_j \leq (\theta_u)^h,$$

so that  $\lambda_h \rightarrow 0$  for  $h \rightarrow \infty$ . Let us assume, by contradiction, that the cycle does not terminate. Then, for all  $h$  we have

$$f(x^k + \lambda_h d^k) > \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + \gamma_1 \lambda_h \nabla f(x^k)^T d^k - \gamma_2 (\lambda_h)^2 \|d^k\|^2,$$

whence it follows

$$\frac{f(x^k + \lambda_h d^k) - f(x_k)}{\lambda_h} - \gamma_1 \nabla f(x^k)^T d^k > -\gamma_2 \lambda_h \|d^k\|^2. \quad (36)$$

Taking limits in (36) for  $h \rightarrow \infty$ , as  $\gamma_1 < 1$  and  $\lambda_h \rightarrow 0$ , we obtain  $\nabla f(x^k)^T d^k \geq 0$ , which contradicts the hypothesis on  $d^k$ .

Now we show that the cycle at Step 5 terminates. Let  $h$  be a counter of the inner iterations of the cycle; we can write

$$\lambda_h = \prod_{j=1}^h \sigma_j \geq (\sigma_l)^h,$$

so that  $\lambda_h \rightarrow \infty$  for  $h \rightarrow \infty$ . Reasoning again by contradiction, assume that the cycle does not terminate in a finite number of inner iterations. By the instructions of Step 6 this implies that

$$f(x^k + \lambda_h d^k) < f^k + \gamma_1 \lambda_h \nabla f(x^k)^T d^k - \gamma_2 \lambda_h^2 \|d^k\|^2 \quad \text{for all } h \quad (37)$$

and hence, as at least one of nonnegative parameters  $\gamma_1, \gamma_2$  is positive, for  $h \rightarrow \infty$  we would obtain that  $f(x^k + \lambda_h d^k) \rightarrow -\infty$ , which violates the boundedness assumption on  $f$ .

Then, condition (21) follows from the stopping criterion of Step 3. The stopping criterion of Step 4 implies that at least one of conditions (22) (23) is satisfied in case of termination. Finally, when the algorithm terminates at Step 6, it follows that at least one of conditions (24) (25) holds.  $\square$

*Proof of Proposition 4*

Assume first  $\gamma_2 > 0$ ; then Proposition 3 implies assertion (i) with  $\eta = \gamma_2$ . Let us consider the case  $\gamma_2 = 0$ ; by Proposition 3, taking into account the assumptions on  $d^k$  we have for every  $k \in K$

$$\begin{aligned} f(x^{k+1}) &\leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \gamma_1 \frac{c_2}{c_1^2} \lambda^k \|d^k\|^2 \\ &= \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \gamma_1 \frac{c_2}{c_1^2} \bar{\lambda} \frac{\lambda^k}{\bar{\lambda}} \|d^k\|^2, \end{aligned} \quad (38)$$

where  $\bar{\lambda}$  is the positive number chosen at Step 0. The instructions of algorithm NLS imply that  $\lambda^k \leq \bar{\lambda}$ , so that we have  $(\lambda^k / \bar{\lambda})^2 \leq \lambda^k / \bar{\lambda} \leq 1$ . Then, from (38) we obtain

$$f(x^{k+1}) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \frac{\gamma_1}{\bar{\lambda}} \frac{c_2}{c_1^2} (\lambda^k \|d^k\|)^2,$$

and this proves assertion (i).

By (i) we have, for all  $k \in K$ ,

$$f(x^{k+1}) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] - \eta \|x^{k+1} - x^k\|^2,$$

so that, as the sequence  $\{f^k\}$  is convergent, we get

$$\lim_{k \rightarrow \infty, k \in K} \|x^{k+1} - x^k\| = 0. \quad (39)$$

In order to prove assertion (ii), let us assume, by contradiction, that there exists an infinite subset  $K_1 \subseteq K$  such that

$$\lim_{k \rightarrow \infty, k \in K_1} x^k = \bar{x}$$

and

$$\lim_{k \rightarrow \infty, k \in K_1} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = \nabla f(\bar{x})^T \bar{d} < 0. \quad (40)$$

The boundedness of  $\{x^k\}_K$  and the continuity assumption on  $\nabla f$  ensure the existence of a subsequence that yields the preceding limits.

From (39) and (40), recalling the definition of  $\Delta^k$  given in (18), it follows that for  $k \in K_1$  and  $k$  sufficiently large condition (22) does not hold, and hence at least one of the conditions (21), (23)–(25) is satisfied. By redefining the subset  $K_1$  if necessary, assume first that condition (21) holds for  $k \in K_1$ . Then, we have

$$\begin{aligned} f\left(x^k + \frac{\lambda^k}{\theta^k} d^k\right) &> \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] + \gamma_1 \frac{\lambda^k}{\theta^k} \nabla f(x^k)^T d^k - \gamma_2 \left(\frac{\lambda^k}{\theta^k}\right)^2 \|d^k\|^2 \\ &\geq f^k + \gamma_1 \frac{\lambda^k}{\theta^k} \nabla f(x^k)^T d^k - \gamma_2 \left(\frac{\lambda^k}{\theta^k}\right)^2 \|d^k\|^2 \end{aligned}$$

where  $\theta^k \in [\theta_l, \theta_u]$ . By the Mean Value Theorem, there exists a point  $u^k = x^k + \beta^k \frac{\lambda^k}{\theta^k} d^k$ , with  $\beta^k \in (0, 1)$ , such that

$$\nabla f(u^k)^T d^k \geq \gamma_1 \nabla f(x^k)^T d^k - \gamma_2 \frac{\lambda^k}{\theta^k} \|d^k\|^2. \quad (41)$$

By (39) we have that  $u^k \rightarrow \bar{x}$  for  $k \rightarrow \infty$  and  $k \in K_1$ , and hence, dividing both members of (41) by  $\|d^k\|$ , taking limits and recalling that  $\gamma_1 < 1$  we obtain

$$\lim_{k \rightarrow \infty, k \in K_1} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = \nabla f(\bar{x})^T \bar{d} \geq 0, \quad (42)$$

which contradicts (40).

Assuming that condition (23) or (25) is satisfied for  $k \in K_1$ , we can repeat similar reasonings and we obtain again (42), and hence a contradiction to (40).

Finally, let us suppose that condition (24) is satisfied for  $k \in K_1$ ; by Proposition 3 we have

$$f(x^k + \lambda^k d^k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x^{k-j})] + \gamma_1 \lambda^k \nabla f(x^k)^T d^k,$$

so that, the convergence of the sequence  $\{f^k\}$  and condition (24) imply

$$\lim_{k \rightarrow \infty, k \in K_1} \nabla f(x^k)^T d^k = 0.$$

On the other hand, recalling Condition 1 on  $d^k$ , we have

$$|\nabla f(x^k)^T d^k| \geq c_2 \|\nabla f(x^k)\|^2,$$

from which it follows that  $\|\nabla f(x^k)\| \rightarrow 0$ , for  $k \rightarrow \infty$  and  $k \in K_1$ . Therefore, we can write

$$\lim_{k \rightarrow \infty, k \in K_1} \frac{\nabla f(x^k)^T d^k}{\|d^k\|} = \nabla f(\bar{x})^T \bar{d} = 0,$$

which contradicts again (40). □

## Acknowledgments

The authors are indebted to an anonymous referee for the useful suggestions.

## References

1. J. Barzilai and J.M. Borwein, "Two point step size gradient method," *IMA J. Numer. Anal.*, vol. 8, pp. 141–148, 1988.
2. D.P. Bertsekas, *Nonlinear Programming*, 2nd edn., Athena Scientific, 1999.
3. I. Bongartz, A. Conn, N. Gould, and P. Toint, "CUTE: constrained and unconstrained testing Environments," *ACM Transaction on Math. Software*, vol. 21, pp. 123–160, 1995.
4. R.M. Chamberlain, M.J.D. Powell, C. Lemarechal, and H.C. Pedersen, "The watchdog technique for forcing convergence in algorithms for constrained optimization," *Math. Programming*, vol. 16, pp. 1–17, 1982.
5. Y.H. Dai and L.Z. Liao, "R-Linear Convergence of the Barzilai and Borwein gradient method," *Research Report*, 1999. Also in *IMA J. Numer. Anal.*, vol. 22, pp. 1–10, 2002.
6. Y.H. Dai and H. Zhang, "An adaptive two-point stepsize gradient algorithm," *Research report*, Chinese Academy of Sciences, 2000.
7. R. De Leone, M. Gaudioso, and L. Grippo, "Stopping criteria for linesearch methods without derivatives," *Math. Programming*, vol. 30, pp. 285–300, 1984.
8. R. Fletcher, "Low storage methods for unconstrained optimization," *Lectures in Applied Mathematics (AMS)*, vol. 26, pp. 165–179, 1990.
9. R. Fletcher, "On the Barzilai-Borwein method," *Numerical Analysis Report NA/207*, 2001.
10. A. Friedlander, J.M. Martinez, B. Molina, and M. Raydan, "Gradient method with retards and generalizations," *SIAM J. Numer. Anal.*, vol. 36, pp. 275–289, 1999.
11. A. Friedlander, J.M. Martinez, and M. Raydan, "A new method for large-scale box constrained convex quadratic minimization problems," *Optimization Methods and Software*, vol. 5, pp. 55–74, 1995.
12. J. Gilbert and C. Lemaréchal, "Some numerical experiments with variable-storage quasi-Newton algorithms," *Math. Programming, Series B*, vol. 45, pp. 407–435, 1989.
13. P. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, Academic Press: San Diego, 1981.
14. W. Glunt, T.L. Hayden, and M. Raydan, "Molecular conformations from distances matrices," *J. Comput. Chem.*, vol. 14, pp. 114–120, 1993.
15. W. Glunt, T.L. Hayden, and M. Raydan, "Preconditioners for distance matrix algorithms," *J. Comput. Chem.*, vol. 15, pp. 227–232, 1994.
16. L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM J. Numer. Anal.*, vol. 23, pp. 707–716, 1986.
17. L. Grippo, F. Lampariello, and S. Lucidi, "A class of nonmonotone stabilization methods in unconstrained optimization," *Numer. Math.*, vol. 59, pp. 779–805, 1991.
18. D.C. Liu and J. Nocedal, "On the limited-memory BFGS method for large scale optimization," *Math. Programming*, vol. 45, pp. 503–528, 1989.
19. W. Liu and Y. H. Dai, "Minimization algorithms based on supervisor and searcher cooperation," *J. Optimization Theory and Applications*, vol. 111, pp. 359–379, 1989.
20. B. Molina and M. Raydan, "Preconditioned Barzilai-Borwein method for the numerical solution of partial differential equations," *Numerical Algorithms*, vol. 13, pp. 45–60, 1996.
21. J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press: San Diego, 1970.
22. M. Raydan, "On the Barzilai and Borwein choice of the steplength for the gradient method," *IMA J. Numer. Anal.*, vol. 13, pp. 618–622, 1993.

23. M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, pp. 26–33, 1997.
24. M. Raydan and B.F. Svaiter, "Relaxed steepest descent and Chauchy-Barzilai-Borwein method," *Computational Optimization and Applications*, vol. 21, pp. 155–167, 2002.
25. D.F. Shanno and K.H. Phua, "Matrix conditioning and nonlinear optimization," *Math. Programming*, vol. 14, pp. 149–160, 1978.