Optim Lett
DOI 10.1007/s11590-009-0132-y

ORIGINAL PAPER

A nonmonotone truncated Newton–Krylov method exploiting negative curvature directions, for large scale unconstrained optimization

Giovanni Fasano · Stefano Lucidi

Received: 1 September 2008 / Accepted: 4 June 2009

Abstract We propose a new truncated Newton method for large scale unconstrained optimization, where a Conjugate Gradient (CG)-based technique is adopted to solve Newton's equation. In the current iteration, the Krylov method computes a pair of search directions: the first approximates the Newton step of the quadratic convex model, while the second is a suitable negative curvature direction. A test based on the quadratic model of the objective function is used to select the most promising between the two search directions. Both the latter selection rule and the CG stopping criterion for approximately solving Newton's equation, strongly rely on conjugacy conditions. An appropriate linesearch technique is adopted for each search direction: a nonmonotone stabilization is used with the approximate Newton step, while an Armijo type linesearch is used for the negative curvature direction. The proposed algorithm is both globally and superlinearly convergent to stationary points satisfying second order necessary conditions. We carry out a significant numerical experience in order to test our proposal.

Keywords Truncated Newton methods · Conjugate directions · Negative curvatures · Nonmonotone stabilization technique · Second order necessary conditions

G. Fasano (✉)
Dipartimento di Matematica Applicata, Università Ca'Foscari di Venezia,
Dorsoduro 3825/e, 30123 Venice, Italy
e-mail: fasano@unive.it

S. Lucidi
Dipartimento di Informatica e Sistemistica A. Ruberti, Sapienza Università di Roma,
via Ariosto 25, 00185 Rome, Italy
e-mail: lucidi@dis.uniroma1.it

Published online: 26 June 2009

1 Introduction

We consider the solution of the unconstrained minimization problem

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

where $f(x)$ is twice continuously differentiable on $\mathbb{R}^n$ and n is large. Several appealing algorithms have already been proposed in the literature to solve (1.1) [1,5,7,8,11,12,14,18,21], however the definition of both robust and efficient methods for large scale unconstrained problems is still a challenging task. In particular, we observe that the state-of-the-art Newton-type methods are based on the idea of exploiting the local information on the function $f(x)$, obtained by investigating the second order derivatives. In the context of large scale problems, the latter task is pursued by computing at the outer iteration k the pair (d_k, s_k) of promising search directions [5,7,8,14,17,20], by means of efficient iterative techniques. Roughly speaking, d_k summarizes the local convexity of $f(x)$ at the current iterate, while s_k takes into account the local non-convexity of the objective function.

In several earlier papers [5,7,14,18] the latter directions were suitably combined in a curvilinear framework, so that the new iterate is laid on the two dimensional manifold identified by the search directions. On the other hand, in [8] a couple of search directions is computed, too. Then, a suitable test attempts to determine if either the first or the second direction is more promising. Furthermore, according with the chosen direction, a proper monotone linesearch technique is applied in order to provide the new iterate. The rational behind using a different linesearch technique for each direction, is the possibility of capturing possible differences between the two directions. In [20] there is an attempt to match both the approaches above, in order to yield an efficient algorithm for small scale problems, adopting a monotone stabilization technique.

In this paper we draw our inspiration from [8], whose results are suitably extended and partially generalized. In particular, we extend the approach in [8] by introducing the following effective ingredients.

(a) We propose an effective use of conjugate directions computed via a Conjugate Gradient (CG)-based method, for both the computation and the comparison between the search directions d_k and s_k.
(b) We use a new stabilization technique which includes a nonmonotone linesearch technique along the direction d_k, and a monotone one along the negative curvature direction s_k.

As regards (a), the use of CG-based methods has a twofold importance. On one hand, they are often the methods of choice to inexpensively and reliably compute a satisfactory approximation of Newton's direction. On the other hand they provide, as a by product, a set of conjugate directions, containing relevant local information of the objective function on an independent set [6]. As a consequence, the conjugate directions can be suitably combined into the pair of search directions d_k and s_k, in order to separately summarize the local information on the convexity and non-convexity of $f(x)$. Moreover, the conjugate directions are "similarly scaled" and so are d_k, s_k. The latter property may be considerably helpful to select the most promising direction in the pair.

As regards (b), the role of nonmonotonicity within Newton-type methods was largely investigated in [5,12–14]. The significant numerical experience reported in [14,15] suggests that over highly nonlinear and ill-conditioned problems, a nonmonotone stabilization can be very effective when combined with a Newton-type direction.

This paper is organized as follows. In Sect. 2, we describe the use of the CG method to both generate the search directions and satisfy specific conditions for the convergence. In Sect. 3, we describe our Adaptive Linesearch Algorithm (`ALA`) for the solution of problem (1.1), along with the convergence properties. We provide sufficient conditions so that the algorithm `ALA` is globally and superlinearly convergent to stationary points, which satisfy both the first and the second order necessary optimality conditions. Finally, Sect. 4 reports a detailed numerical experience of algorithm `ALA`, over a significant set of large scale problems of `CUTEr` [9], selected from [11].

We use the symbol $A \succeq 0$ to denote the positive semidefinite matrix A, and $\|\cdot\|$ represents the Euclidean norm of either a vector or a matrix. With H_k and g_k we respectively indicate the Hessian $\nabla^2 f(x_k)$ and the gradient $\nabla f(x_k)$ at the current iterate x_k.

2 The generation of search directions d_k and s_k

Our truncated Newton method generates the sequence $\{x_k\}$ according with the iterative scheme:

$$x_{k+1} = x_k + \alpha_k z_k,$$

where $z_k \in \{d_k, s_k\}$ and α_k is a suitable stepsize. Throughout this paper we consider that the following assumption holds.

Assumption 1 *The function $f(x)$ is twice continuously differentiable on $\mathbb{R}^n$, and given $x_0 \in \mathbb{R}^n$, the level set $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$ is compact.*

Let us assume that at the current iterate x_k we apply the following iterative scheme CG_gen to solve the Newton equation.[1]

CG_gen: Conjugate directions generation

Data. $x_k, g_k, H_k, \varepsilon \in (0, 2)$.
Initialization. $r_0 = -g_k$, $p_0 = -g_k$. Set $i = 0$.
Do
  If $|p_i{}^T H_k p_i| < \varepsilon \|p_i\|^2$ Stop.
  Compute $r_{i+1} = r_i - \rho_i H_k p_i$, with $\rho_i = p_i{}^T r_i / p_i{}^T H_k p_i$.
  If *a stopping criterion is satisfied* Stop (see Sect. 4).
  Compute $p_{i+1} = r_{i+1} + \beta_i p_i$, with $\beta_i = \|r_{i+1}\|^2 / \|r_i\|^2$.
  Set $i = i + 1$.
End do

[1] For the sake of simplicity we omit the dependency of p_i from the index k.

After $m+1$ steps the $m+1$ directions $p_0, \dots, p_m$ have been generated. In particular, when $m > 0$ we introduce the disjoint sets of indices

$$\begin{aligned} I_k^P &= \{i \in [0, m] \ : \ p_i^T H_k p_i \geq \varepsilon \|p_i\|^2\}, \\ I_k^N &= \{i \in [0, m] \ : \ p_i^T H_k p_i \leq -\varepsilon \|p_i\|^2\}. \end{aligned} \tag{2.1}$$

Now, we can use the set $\{p_0, \dots, p_m\}$ to generate at step k both a *negative curvature direction* s_k and a *positive curvature direction* d_k (search directions). Furthermore, we can select the most promising direction between s_k and d_k, according with the decrease of the local quadratic model $q(x_k, z)$ at iterate x_k, defined as

$$q(x_k, z) = \frac{1}{2} z^T H_k z + g_k^T z. \tag{2.2}$$

We obtain the following resulting scheme (we recall that $p_i^T r_i = -p_i^T g_k$):

<u>Scheme 1</u>

If $|p_0{}^T H_k p_0| < \varepsilon \|p_0\|^2$ **then**

$$\begin{cases} d_k = p_0 = -g_k, \\ s_k = 0. \end{cases} \tag{2.3}$$

Else

$$d_k = \begin{cases} \displaystyle\sum_{i \in I_k^P} \rho_i p_i = -\sum_{i \in I_k^P} \frac{g_k^T p_i}{p_i^T H_k p_i} p_i, & \text{if } I_k^P \neq \emptyset \\ 0 & \text{if } I_k^P = \emptyset, \end{cases} \tag{2.4}$$

$$s_k = \begin{cases} \displaystyle -\sum_{i \in I_k^N} \rho_i p_i = -\sum_{i \in I_k^N} \frac{g_k^T p_i}{|p_i^T H_k p_i|} p_i, & \text{if } I_k^N \neq \emptyset \\ 0 & \text{if } I_k^N = \emptyset. \end{cases} \tag{2.5}$$

End if

If $q(x_k, d_k) \leq q(x_k, s_k)$ choose the direction d_k.

If $q(x_k, d_k) > q(x_k, s_k)$ choose the direction s_k.

The next proposition proves that the search direction z corresponding to the largest decrease of $q(x_k, z)$ satisfies a *suitable angle condition* for an optimization framework.

Proposition 2.1 *Assume that the directions d_k and s_k are computed by Scheme 1. Then, there exist positive constants c_1 and c_2 such that*

$$\max\{\|d_k\|, \|s_k\|\} \le c_1\|g_k\|, \tag{2.6}$$

and

$$\text{if } d_k \text{ is chosen then } g_k^T d_k \le -c_2\|g_k\|^2; \tag{2.7}$$
$$\text{if } s_k \text{ is chosen then } g_k^T s_k \le -c_2\|g_k\|^2. \tag{2.8}$$

Proof First we study the case $|{p_0}^T H_k p_0| < \varepsilon\|p_0\|^2$ in Scheme 1. Observe that in the latter case $d_k = -g_k$ and $s_k = 0$, so that $q(x_k, s_k) = 0$; furthermore, $q(x_k, -g_k) \le -\|g_k\|^2 + \varepsilon/2\|g_k\|^2 = -(1-\varepsilon/2)\|g_k\|^2$. Thus, (2.6) and (2.7) hold with $c_1 = c_2 = 1$.

On the other hand, for the cases in which $|{p_0}^T H_k p_0| \ge \varepsilon\|p_0\|^2$, from [12] (see formulae (11)–(12)) the positive constants $\tilde{c}_1$ and $\tilde{c}_2$ exist such that

$$\max\{\|d_k\|, \|s_k\|\} \le \tilde{c}_1\|g_k\| \tag{2.9}$$
$$g_k^T(d_k + s_k) \le -\tilde{c}_2\|g_k\|^2. \tag{2.10}$$

Thus, relation (2.6) follows straightforwardly from (2.9) by setting $c_1 = \tilde{c}_1$.

Now we prove (2.7) and (2.8). Since the directions $\{p_i\}$ are computed by CG_gen, the following relations hold:

$$\begin{aligned} g_k^T d_k + \frac{1}{2}d_k^T H_k d_k &= g_k^T d_k + \frac{1}{2}\left(-\sum_{i\in I_k^P}\frac{g_k^T p_i}{p_i^T H_k p_i}p_i\right)^T H_k\left(-\sum_{i\in I_k^P}\frac{g_k^T p_i}{p_i^T H_k p_i}p_i\right) \\ &= \frac{1}{2}g_k^T d_k, \end{aligned} \tag{2.11}$$

$$\begin{aligned} g_k^T s_k + \frac{1}{2}s_k^T H_k s_k &= g_k^T s_k + \frac{1}{2}\left(\sum_{i\in I_k^N}\frac{g_k^T p_i}{p_i^T H_k p_i}p_i\right)^T H_k\left(\sum_{i\in I_k^N}\frac{g_k^T p_i}{p_i^T H_k p_i}p_i\right) \\ &= \frac{3}{2}g_k^T s_k. \end{aligned} \tag{2.12}$$

If $q(x_k, d_k) \le q(x_k, s_k)$, from (2.11) and (2.12)

$$g_k^T d_k \le 3g_k^T s_k;$$

then, (2.10) yields

$$4g_k^T d_k \le 3g_k^T(d_k + s_k) \le -3\tilde{c}_2\|g_k\|^2. \tag{2.13}$$

On the other hand if $q(x_k, d_k) > q(x_k, s_k)$, we have similarly from (2.11) and (2.12)

$$3g_k^T s_k < g_k^T d_k;$$

then, again (2.10) yields

$$4g_k^T s_k < g_k^T (d_k + s_k) \le -\tilde{c}_2 \|g_k\|^2. \tag{2.14}$$

Finally, relations (2.13) and (2.14) yield (2.7) and (2.8) with respectively $c_2 = 3/4\tilde{c}_2$ and $c_2 = \tilde{c}_2/4$. $\square$

In order to ensure the convergence results to critical points satisfying second order necessary conditions, we need a negative curvature direction which conveys more information on the local nonconvexity of the objective function. This can be done by adding, when needed, to the negative curvature direction produced by Scheme 1, an additional negative curvature direction $\hat{s}_k$ which satisfies the following assumption.

Assumption 2 *For any outer iteration $k \ge 0$ a bounded direction $\hat{s}_k$ exists such that*

(a) $g_k^T \hat{s}_k \le 0$;
(b) $\hat{s}_k^T H_k \hat{s}_k \le 0$;
(c) *for every $x^* \in \mathbb{R}^n$, with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \not\succeq 0$, there exist $\sigma > 0$ and $\tilde{\epsilon} > 0$ such that if $\|x_k - x^*\| \le \sigma$, then $\hat{s}_k^T H_k \hat{s}_k \le -\tilde{\epsilon}$.*

Assumption 2 generalizes the properties of the negative curvature directions proposed in literature, for defining minimization algorithms globally converging towards second order stationary points (see, for example [5,4,14,17]).

Now we can consider the following Scheme 2 including $\hat{s}_k$.

Scheme 2

Data: let $\bar{d}_k$ and $\bar{s}_k$ be directions given by (2.3)-(2.5) of Scheme 1,
let $\hat{s}_k$ be a direction satisfying Assumption 2.

Compute:

$$d_k = \bar{d}_k$$

$$s_k = \begin{cases} \bar{s}_k + \hat{s}_k, & \text{if } (\bar{s}_k + \hat{s}_k)^T H_k (\bar{s}_k + \hat{s}_k) < 0, \\ \bar{s}_k & \text{otherwise.} \end{cases} \tag{2.15}$$

If $q(x_k, d_k) \le q(x_k, s_k)$ choose the direction d_k.
If $q(x_k, d_k) > q(x_k, s_k)$ choose the direction s_k.

The following proposition describes the properties of the directions computed by Scheme 2.

Proposition 2.2 *Let us assume that the directions d_k and s_k are computed by Scheme 2. Then,*

(i) *there exist positive constants $\hat{c}_1$ and $\hat{c}_2$ such that*

$$\max\{\|d_k\|, \|s_k\|\} \le \hat{c}_1, \tag{2.16}$$

and

$$\text{if } d_k \text{ is chosen then } g_k^T d_k \le -\hat{c}_2 \|g_k\|^2, \tag{2.17}$$

$$\text{if } s_k \text{ is chosen then } g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \le -\hat{c}_2 \|g_k\|^2; \tag{2.18}$$

(ii) *for every $x^* \in \mathbb{R}^n$, with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \not\succeq 0$, there exist $\sigma > 0$ and $\tilde{\epsilon} > 0$ such that if $\|x_k - x^*\| \le \sigma$, then*

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k < -\frac{\tilde{\epsilon}}{4}.$$

Proof As regards (i), relation (2.16) follows directly from Proposition 2.1 and Assumption 2.

Now, to prove (2.17) and (2.18) we first consider the case $|{p_0}^T H_k p_0| < \varepsilon \|p_0\|^2$ in Scheme 1. In this case $d_k = -g_k$ and $\bar{s}_k = 0$, and two subcases must be considered. If $q(x_k, -g_k) \le q(x_k, s_k) \equiv q(x_k, \hat{s}_k)$ then (2.17) follows from the proof of Proposition 2.1. On the other hand if $q(x_k, -g_k) > q(x_k, \hat{s}_k)$ then we have

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k = g_k^T \hat{s}_k + \frac{1}{2} \hat{s}_k^T H_k \hat{s}_k < -\|g_k\|^2 + \frac{1}{2} g_k^T H_k g_k \le -\left(1 - \frac{1}{2}\varepsilon\right) \|g_k\|^2,$$

so that (2.18) holds with $\hat{c}_2 = 1 - \varepsilon/2$.

Let us consider now the case $|{p_0}^T H_k p_0| \ge \varepsilon \|p_0\|^2$ in Scheme 1. If $q(x_k, d_k) \le q(x_k, s_k)$ then the Scheme 2 and the Assumption 2 yield:

$$g_k^T d_k + \frac{1}{2} d_k^T H_k d_k \le g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \le g_k^T \bar{s}_k. \tag{2.19}$$

Moreover, since the directions $\bar{d}_k$ and $\bar{s}_k$ are computed in Scheme 1, it is possible to repeat the arguments of Proposition 2.1. In particular, recalling (2.11), relation (2.19) yields

$$g_k^T d_k + \frac{1}{2} d_k^T H_k d_k = g_k^T \bar{d}_k + \frac{1}{2} \bar{d}_k^T H_k \bar{d}_k = \frac{1}{2} g_k^T \bar{d}_k \le g_k^T \bar{s}_k,$$

from which, using (2.10)

$$\frac{3}{2} g_k^T \bar{d}_k \le g_k^T (\bar{d}_k + \bar{s}_k) \le -\hat{c}_2 \|g_k\|^2, \tag{2.20}$$

so that (2.17) holds with $\hat{c}_2 = \tilde{c}_2$.

If $q(x_k, s_k) < q(x_k, d_k)$, since $d_k = \bar{d}_k$ and the direction $\bar{d}_k$ is given by (2.4), we can use again (2.11) to obtain

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k < \frac{1}{2} g_k^T \bar{d}_k; \tag{2.21}$$

then, by the definition of s_k

$$\frac{1}{2} \left[g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \right] < \frac{1}{2} g_k^T \bar{s}_k. \tag{2.22}$$

Adding term to term (2.21) and (2.22), recalling again (2.10), we obtain:

$$\frac{3}{2} \left[g_k^T s_k + \frac{1}{2} s_k^T H_k s_k \right] < \frac{1}{2} g_k^T (\bar{d}_k + \bar{s}_k) < -\frac{1}{2} \tilde{c}_2 \|g_k\|^2, \tag{2.23}$$

which yields (2.18) with $\hat{c}_2 = \tilde{c}_2/2$.

The proof of (ii) easily follows from (2.6) and Assumption 2. In fact, (2.6) ensures that for every stationary point x^* of $f(x)$, there are neighborhoods where the norm of $\bar{s}_k$ is sufficiently small. On the other hand, Assumption 2 implies that the negative scalar $\hat{s}_k^T H_k \hat{s}_k$ is bounded away from zero in a sufficiently small neighborhood of the stationary point x^*, which does not satisfy the second order necessary conditions. Therefore we can conclude that for such stationary points there exists sufficiently small $\sigma > 0$, such that if $\|x_k - x^*\| \le \sigma$, then from Assumption 2

$$g_k^T s_k + \frac{1}{2} s_k^T H_k s_k < -\frac{\tilde{\epsilon}}{4}.$$

□

3 The adaptive linesearch algorithm

In this section, we describe our new algorithmic framework, which uses the search directions d_k and s_k computed in Sect. 2. We propose at the outer iteration k an **A**daptive **L**inesearch **A**lgorithm (ALA), in which the globalization strategy is tailored on the local curvatures of $f(x)$. The relevant steps of algorithm ALA are summarized in the following points:

Computation and choice of the search direction: We compute at the current iterate x_k the pair of search directions d_k and s_k, according with either Scheme 1 or Scheme 2. The first direction d_k is given by a linear combination of conjugate vectors, which are positive curvature directions for $f(x)$ at x_k. It can be regarded as a Newton-type direction. On the other hand, the vector s_k is a negative curvature direction for $f(x)$ at x_k, which contains relevant information on the subspace of non-convexity of $f(x)$ at x_k. Then, a test based on a quadratic model of $f(x)$ is used to select either d_k or s_k. Observe that to generate d_k and s_k, the use of the Conjugate Gradient has two advantages with respect to the Lanczos process. First, the CG is slightly cheaper; then, it provides *directly* a set of conjugate directions, while the Lanczos process would require an additional computation.
Computation of the new point along the `positive` *curvature direction*: When the Newton-type direction d_k is selected by the test, we investigate whether x_k is in a region where the superlinear convergence rate holds for d_k (i.e. $\|d_k\|$ decreases at a suitable rate). In this case, the unit stepsize is desirable for d_k [1]. Thus, we adopt a nonmonotone strategy to allow the acceptance of the unit stepsize along d_k, as frequently as possible [13].
Computation of the new point along the `negative` *curvature direction*: In case the negative curvature direction s_k is selected by the test, a stepsize is computed by a monotone linesearch, which includes the negative term $s_k^T H_k s_k$ (see also [16]).
In order to compute stepsizes which take advantage of the *second order descent property* of negative curvature directions, we also include extrapolation along s_k.

For the sake of simplicity, we prefer to give here an informal description of some quantities used in Algorithm ALA. We defer the interested reader to [3], for a complete and rigorous description of both the Algorithm ALA and its theoretical properties.
Similarly to [13], in algorithm ALA the objective function is not evaluated at any iterate x_k; anyway, it is computed at least once on any N iterations. In addition, when linesearch is performed at iteration k along the search direction d_k, then the objective function values on the trial points are not compared with $f(x_k)$. Indeed, they are compared with the largest value of the objective function, over the last M iterates before k in which it was computed.
On this guideline, at iteration k we denote by ℓ the largest iteration index, not exceeding k, where f is evaluated. Moreover, we use f_ℓ^M to indicate the largest value of the objective function, over the last M iterates before k in which it was computed (see also [13]).

According with the latter notation we propose the following algorithm ALA.

ALA (Adaptive Linesearch Algorithm)

Step 0. Choose $x_0 \in \mathbb{R}^n$, $\beta \in (0,1)$, $\Delta_0 > 0$, $\delta \in (0,1)$, $N > 0$, $M \geq 0$, $\mu \in (0, \frac{1}{2})$.
Set $k = \ell = 0$, $\Delta = \Delta_0$.

Step 1. **Computation and choice of the search direction.**

If *direction* s_k *is computed and chosen then execute Step 2,* else *execute Step 3.*

Step 2. **Linesearch along the negative curvature direction.**

Step 2.1. **Check on the function.**

i. If $k \neq \ell$ *compute* $f(x_k)$;

ii. if $f(x_k) \geq f_\ell^M$, *backtrack to* x_ℓ *and go to Step 1,* else *set* $\ell = k$;

Step 2.2. **Monotone linesearch.**

i. If $f(x_k + s_k) \leq f(x_k) + \mu\left(\sigma_k g_k^T s_k + \frac{1}{2} s_k^T H_k s_k\right)$, *set* $\alpha_k = \beta^h$, *where* h *is the largest non-positive integer such that*

$$f(x_k + \alpha_k s_k) \leq f(x_k) + \mu\left(\alpha_k g_k^T s_k + \frac{1}{2}\alpha_k^2 s_k^T H_k s_k\right) \qquad (3.1)$$

$$f\left(x_k + \frac{\alpha_k}{\beta} s_k\right) > f(x_k) + \mu\left(\frac{\alpha_k}{\beta} g_k^T s_k + \frac{1}{2}\left(\frac{\alpha_k}{\beta}\right)^2 s_k^T H_k s_k\right) \qquad (3.2)$$

else *set* $\alpha_k = \beta^h$, *where* h *is the smallest positive integer such that* (3.1) *holds.*

ii. *Set* $x_{k+1} = x_k + \alpha_k s_k$, $k = k+1$, $\ell = k$ *and go to Step 1.*

Step 3. **Linesearch along the Truncated Newton direction.**

Step 3.1. **Function control every N steps.**

i. If $k = \ell + N$ *compute* $f(x_k)$;

ii. if $f(x_k) \geq f_\ell^M$, *backtrack to* x_ℓ *and go to Step 1,* else *set* $\ell = k$.

Step 3.2. **Test for acceptance.**

If $\| d_k \| \leq \Delta$, *set* $x_{k+1} = x_k + d_k$, $k = k+1$, $\Delta = \delta\Delta$ *and go to Step 1,*
else,

A. if $k \neq \ell$ *compute* $f(x_k)$;

B. if $f(x_k) \geq f_\ell^M$, *backtrack to* x_ℓ *and go to Step 1,* else $\ell = k$.

Step 3.3. **Nonmonotone linesearch.**

(a) *Set* $\alpha_k = \beta^h$ *where* h *is the smallest nonnegative integer such that*

$$f(x_k + \alpha_k d_k) \leq f_\ell^M + \alpha_k \mu g_k^T d_k, \qquad (3.3)$$

(b) *set* $x_{k+1} = x_k + \alpha_k d_k$, $k = k+1$, $\ell = k$ *and go to Step 1.*

Now we complete this section by reporting the main convergence results relative to the algorithm `ALA`.

Theorem 3.1 *Suppose the algorithm* ALA *generates the sequence* $\{x_k\}$.

(a) *If the search directions are computed by Scheme 1 then, either an integer* $h \geq 0$ *exists such that* $\nabla f(x_h) = 0$, *or the sequence* $\{x_k\}$ *is infinite, every limit point* x^* *belongs to* $\mathcal{L}_0$ *and satisfies the relation* $\nabla f(x^*) = 0$.

(b) *If the search directions are computed by Scheme 2 then, either an integer* $h \geq 0$ *exists such that* $\nabla f(x_h) = 0$ *and* $\nabla^2 f(x_h) \succeq 0$, *or the sequence* $\{x_k\}$ *is infinite and every limit point* x^* *satisfies the relations* $\nabla f(x^*) = 0$ *and* $\nabla^2 f(x^*) \succeq 0$.

Proof The main ideas in the proof of this theorem, along with some intermediate lemmas, use the reasoning in the papers [8,13]. For brevity's sake we omit the proofs and again defer the reader to [3]. □

4 Implementation issues and numerical results

We report here a numerical experience with our algorithm ALA on a set of standard test problems from the literature. We applied the algorithm ALA in order to generate the sequence $\{x_k\}$. In the practical implementation of ALA x_0 is the starting point proposed in the literature. Moreover, in ALA we set the parameters $\beta = 0.5$, $\Delta_0 = 10^3$, $\delta = 0.9$, $N = 20$, $M = 100$, $\mu = 10^{-3}$, and in the procedure CG_gen we set $\varepsilon = 10^{-8}$. We considered the test set from CUTEr collection [9] proposed in [11], discarding the test problems with too few unknowns (< 49). Anyway, we included some test functions with a number n of unknowns in the range $50 \leq n \leq 500$, since they are considered pretty difficult test problems. The list of our test problems and the relative number of unknowns is reported in Table 1.

Table 1 List of our test problems from CUTEr [9]

ARGLINA	200	ARGLINB	200	ARGLINC	200	ARWHEAD	5000	BDQRTIC	5000
BROWNAL	200	BRYBND	5000	CHAINWOO	4000	CLPLATEA	9900	CLPLATEB	4970
CLPLATEC	4970	COSINE	10000	CRAGGLVY	5000	CURLY10	10000	CURLY20	10000
CURLY30	1000	DIXMAANA	9000	DIXMAANB	9000	DIXMAANC	9000	DIXMAAND	9000
DIXMAANE	9000	DIXMAANF	9000	DIXMAANG	9000	DIXMAANH	9000	DIXMAANI	9000
DIXMAANJ	9000	DIXMAANK	9000	DIXMAANL	9000	DIXON3DQ	10000	DQDRTIC	5000
DQRTIC	5000	EDENSCH	10000	EG2	1000	EIGENALS	2550	EIGENBLS	2550
EIGENCLS	2652	ENGVAL1	10000	EXTROSNB	1000	FMINSRF2	5625	FMINSURF	49
FREUROTH	5000	GENROSE	500	HYDC20LS	99	LIARWHD	5000	LMINSURF	5329
MANCINO	100	MOREBV	5000	MSQRTALS	1024	MSQRTBLS	1024	NCB20	5010
NCB20B	5000	NLMSURF	5329	NONCVXU2	5000	NONCVXUN	5000	NONDIA	5000
NONDQUAR	5000	NONMSQRT	100	ODC	4900	PENALTY1	1000	PENALTY2	200
PENALTY3	120	POWELLSG	5000	POWER	100	QUARTC	5000		
RAYBENDL	2046	RAYBENDS	2046	SBRYBND	500	SCHMVETT	5000	SCOSINE	5000
SCURLY10	100	SCURLY20	100	SCURLY30	100	SENSORS	100	SINQUAD	10000
SPARSINE	5000	SPARSQUR	10000	SPMSRTLS	4900	SROSENBR	5000	SSC	4900
TESTQUAD	5000	TOINTGSS	5000	TQUARTIC	5000	TRIDIA	5000	VARDIM	200
VAREIGVL	50	WOODS	10000						

We coded ALA in Fortran 90 (Compaq Visual Fortran) and we used a Pentium 4, 2.53 Ghz, 1 Gb RAM, to perform the computation. For all the algorithms in our numerical experience we set the following limits: 1,800s (1/2 hour) CPU-time, 100,000 outer iterations or function evaluations, 300,000 inner iterations. The stopping criterion in the iterative scheme CG-gen is the Nash-Sofer rule in [19]. However, based on our experience, we reformulated the latter rule and considered:

- the possible nonconvexity of the objective function;
- the possible conjugacy loss.

On this purpose, in order to cope with the nonconvexity of $f(x)$, the Nash-Sofer stopping criterion is modified as

$$\left| \frac{q(x_k, d_i) - q(x_k, d_{i-1})}{q(x_k, d_i)/i} \right| \leq \gamma, \quad \gamma \in (0, 1). \tag{4.1}$$

In addition, observe from (2.11) that $d_i^T H_k d_i + g_k^T d_i = 0$ and $g_k^T d_i = 2q(x_k, d_i)$, so that

$$2q(x_k, d_i) = g_k^T d_i = g_k^T d_i - 2(d_i^T H_k d_i + g_k^T d_i) = -2d_i^T H_k d_i - g_k^T d_i = -\tilde{q}(x_k, d_i).$$

Thus, the criterion (4.1) can be replaced by

$$\left| \frac{\tilde{q}(x_k, d_i) - \tilde{q}(x_k, d_{i-1})}{\tilde{q}(x_k, d_i)/i} \right| \leq \gamma, \quad \gamma \in (0, 1), \tag{4.2}$$

and since $\tilde{q}(x_k, d_i) = 2d_i^T H_k d_i + g_k^T d_i = 3/2 d_i^T H_k d_i + q(x_k, d_i)$, from (4.1) and (4.2) we have

$$\left| \frac{[q(x_k, d_i) - q(x_k, d_{i-1})] - \frac{3}{2}[g_k^T d_i - g_k^T d_{i-1}]}{q(x_k, d_i) - \frac{3}{2} g_k^T d_i} \right| i \leq \gamma, \quad \gamma \in (0, 1), \tag{4.3}$$

which is the criterion we used. We remark that (4.3) is theoretically equivalent to (4.1) in exact arithmetic. However, on our test set when conjugacy loss is experienced in practice, (4.3) performs much better. The latter result may be interpreted as follows. When i increases, the conjugacy loss may seriously affect the quadratic model used in the test (4.1). In (4.3) (see Proposition 2.2) we monitor the decrease of *both* the quadratic model and the directional derivative of the current Newton-type direction d_i, in order to deflate the conjugacy loss.

For LANCELOT B and the Lanczos-based algorithm we adopted the stopping criterion on inner iterations (i.e. the truncation rule) respectively specified in [10] and [14].

Finally, the direction $\hat{s}_k$ in Scheme 2 may be computed as in [4] so that the theoretical assumptions on $\hat{s}_k$ are satisfied (in addition, the latter choice of $\hat{s}_k$ always satisfies the condition $(\bar{s}_k + \hat{s}_k)^T H_k (\bar{s}_k + \hat{s}_k) < 0$ in Scheme 2). However, for practical efficiency (see Fig. 1 for comparative results), we preferred to set $s_k = \alpha_N p_N$, where

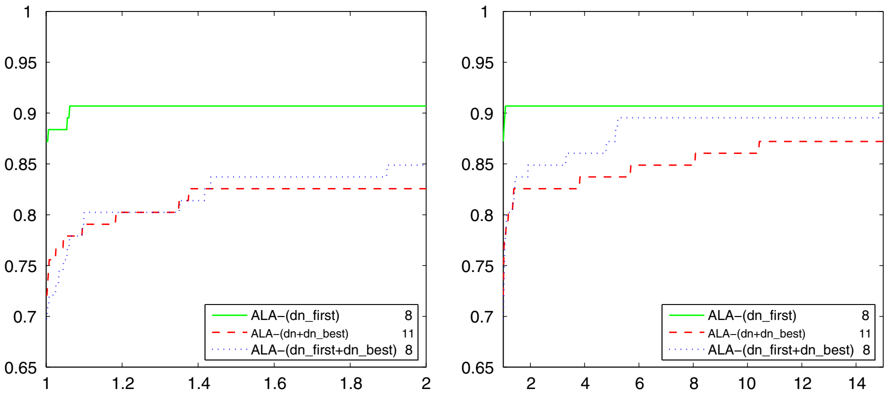

Fig. 1 (*left*) Detail of the performance profile and (*right*) complete performance profile, on the comparison among three implementations of ALA. The negative curvature direction s_k in the three implementations is given by: (dn_first) $s_k = \alpha_N p_N$, where p_N is the first conjugate direction computed by CG_gen, such that $p_N^T H_k p_N < 0$; (dn + dn_best) $s_k = \bar{s}_k + \hat{s}_k$ as in Scheme 2, where $\hat{s}_k$ is computed according with [4]; (dn_first + dn_best) $s_k = \alpha_N p_N + \hat{s}_k$, where again $\hat{s}_k$ is computed as in [4]. Despite the theoretical results in Sect. 2, a fast computation of the negative curvature direction (dn_first) is often a winning strategy. The comparison refers to inner iterations

p_N is the first conjugate direction computed by CG_gen, such that $p_N^T H_k p_N < 0$. As remarked in [6], the latter choice is partially motivated by the fact that in the early inner iterations the CG exploits the eigenspaces associated with the largest (in absolute value) eigenvalues of H_k. Thus, the first conjugate directions collect significant information on the local curvatures of the objective function.

The algorithms involved in our numerical experience stop (as in [10]) when

$$\|\nabla f(x_k)\|_\infty \leq 10^{-5}.$$

Figures 2 and 3 report the performance profiles [2] of a comparison among LANCELOT B, the curvilinear truncated Newton method in [14] and ALA. The profiles compare the number of inner iterations (Fig. 2), and the time of computation (Fig. 3). The legends in the figures also report the failures of each algorithm on the test set. All the algorithms, in the profiles of Figs. 1, 2, and 3, fail on the eight problems: HYDC20LS, RAYBENDL, RAYBENDS, SBRYBND, SCOSINE, SCURLY10, SCURLY20, SCURLY30. In addition, LANCELOT B fails on NONCVXUN, the Lanczos-based method fails on ARGLINB, ARGLINC, NONCVXUN, and in Fig. 1 ALA-(dn+dn_best) fails on the three problems EIGENCLS, MSQRTBLS, NONCVXUN.

As regards Figs. 2 and 3, LANCELOT B always fails since the number of inner iterations exceeds 300,000. The Lanczos-based method fails on the problem RAYBENDL, because a very small steplength was detected (linesearch failure), and on the problem RAYBENDS, because the time limit was exceeded. Finally, also our proposal ALA-(dn_first) fails on RAYBENDL and RAYBENDS because of a linesearch failure, while the other failures occur since the inner iterations exceed 300,000.

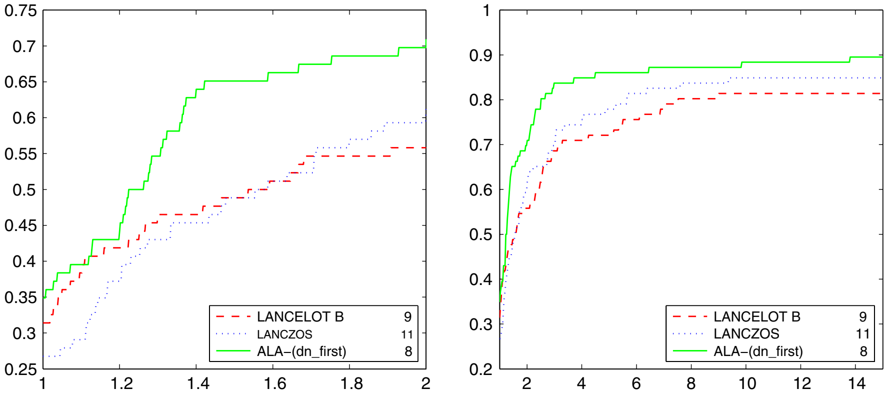

Fig. 2 (*left*) Detail of the performance profile and (*right*) complete performance profile, on the comparison among LANCELOT B, the curvilinear Lanczos-based code in [14] and ALA. Here, the negative curvature direction s_k used in ALA is given by $s_k = \alpha_N p_N$, where p_N is simply the first conjugate direction computed by CG_gen, such that $p_N^T H_k p_N < 0$. The comparison refers to inner iterations

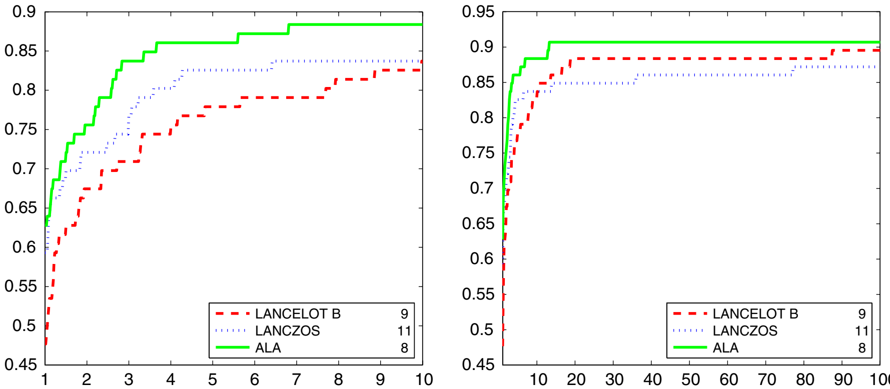

Fig. 3 (*left*) Detail of the performance profile and (*right*) complete performance profile, on the comparison among LANCELOT B, the curvilinear Lanczos-based code in [14] and ALA. As in Fig. 2, the negative curvature direction s_k used in ALA is given by $s_k = \alpha_N p_N$, where p_N is simply the first conjugate direction computed by CG_gen, such that $p_N^T H_k p_N < 0$. The comparison refers to the time of computation

The results seem to suggest that, on the test set adopted, the curvilinear stabilization performs less efficiently than the schemes which adopt a strategy of alternating directions.

Acknowledgments G. Fasano wishes to thank the INSEAN research program "VISIR", and Programma PRIN 20079PLLN7 "Nonlinear Optimization, Variational Inequalities, and Equilibrium Problems". S. Lucidi wishes to thank Programma PRIN 20079PLLN7 "Nonlinear Optimization, Variational Inequalities, and Equilibrium Problems".

References

1. Dembo, R.S., Steihaug, T.: Truncated-Newton algorithms for large-scale unconstrained optimization. Math. Program. **26**, 190–212 (1983)
2. Dolan, E.D., More, J.J.: Benchmarking Optimization Software with Performance Profiles. Mathematics and Computer Science Division. Preprint ANL/MCS-P861-1200
3. Fasano, G., Lucidi, S.: A nonmonotone truncated Newton–Krylov method exploiting negative curvature directions, for large scale unconstrained optimization: complete results. Technical Report INSEAN 2008-035/rt, 2008
4. Fasano, G., Roma, M.: Iterative computation of negative curvature directions in large scale optimization. Comput. Optim. Appl. **38**(1), 81–104 (2007)
5. Ferris, M.C., Lucidi, S., Roma, M.: Nonmonotone curvilinear linesearch methods for unconstrained optimization. Comput. Optim. Appl. **6**, 117–136 (1996)
6. Forsgren A.: On the Behavior of the Conjugate-Gradient Method on Ill-conditioned Problems. Technical Report TRITA-MAT-2006-OS1, Department of Mathematics, Royal Institute of Technology
7. Goldfarb, D.: Curvilinear path steplength algorithms for minimization which use directions of negative curvature. Math. Program. **18**, 31–40 (1980)
8. Gould, N.I.M., Lucidi, S., Roma, M., Toint, Ph.L.: Exploiting negative curvature directions in linesearch methods for unconstrained optimization. Optim. Methods Softw. **14**, 75–98 (2000)
9. Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEr: Constrained and unconstrained testing environment, revised. Trans. ACM Math. Softw. **29**(4), 373–394 (2003)
10. Gould, N.I.M., Orban, D., Toint, Ph.L.: GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization. ACM Trans. Math. Softw. **29**(4), 353–372 (2003)
11. Gould, N.I.M., Sainvitu, C., Toint, Ph.L.: A Filter-Trust-Region method for unconstrained optimization. SIAM J. Optim. **16**(2), 341–357 (2006)
12. Grippo, L., Lampariello, F., Lucidi, S.: A truncated Newton method with nonmonotone linesearch for unconstrained optimization. J. Optim. Theory Appl. **60**, 401–419 (1989)
13. Grippo, L., Lampariello, F., Lucidi, S.: A class of nonmonotone stabilization methods in unconstrained optimization. Numer. Math. **59**, 779–805 (1991)
14. Lucidi, S., Rochetich, F., Roma, M.: Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization. SIAM J. Optim. **8**, 916–939 (1998)
15. Lucidi, S., Roma, M.: Numerical experiences with new truncated Newton methods in large scale unconstrained optimization. Comput. Optim. Appl. **7**, 71–87 (1997)
16. McCormick, G.P.: A modification of Armijo's stepsize rule for negative curvature. Math. Program. **13**, 111–115 (1977)
17. More', J.J., Sorensen, D.C.: On the use of directions of negative curvature in a modified Newton method. Math. Program. **16**, 1–20 (1979)
18. Mukai, H., Polak, E.: A second-order method for unconstrained optimization. J. Optim. Theory Appl. **26**, 501–513 (1978)
19. Nash, S.G., Sofer, A.: Assessing a search direction within a truncated Newton method. Oper. Res. Lett. **9**, 219–221 (1990)
20. Olivares, A., Moguerza, J.M., Prieto, F.J.: Nonconvex optimization using negative curvature within a modified linesearch. Euro. J. Oper. Res. **189**, 706–722 (2008)
21. Shultz, G.A., Schnabel, R.B., Byrd, R.H.: A family of trust-region-based algorithms for unconstrained minimization. SIAM J. Numer. Anal. **22**, 47–67 (1985)