Using gradient directions to get global convergence of Newton-type methods

Daniela di Serafino[a], Gerardo Toraldo[b], Marco Viola[a]

[a] *Dipartimento di Matematica e Fisica, Università degli Studi della Campania "Luigi Vanvitelli", viale A. Lincoln 5, 81100 Caserta, Italy*
[b] *Dipartimento di Matematica e Applicazioni, Università degli Studi di Napoli Federico II, Cupa Nuova Cintia 21, 80126 Napoli, Italy*

Abstract

The renewed interest in Steepest Descent (SD) methods following the work of Barzilai and Borwein [2] has driven us to consider a globalization strategy based on SD, which is applicable to any line-search method. In particular, we combine Newton-type directions with scaled SD steps to have suitable descent directions. Scaling the SD directions with a suitable step length makes a significant difference with respect to similar globalization approaches, in terms of both theoretical features and computational behavior. We apply our strategy to Newton's method and the BFGS method, with computational results that appear interesting compared with the results of well-established globalization strategies devised ad hoc for those methods.

AMS subject classification: 65K05, 90C30, 49M15.

Keywords: Newton-type methods, globalization strategies, steepest descent step.

1. Introduction

We are concerned with the following optimization problem:

$$\text{minimize } f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. Hereafter $\mathbf{g}(\mathbf{x})$ and $H(\mathbf{x})$ denote the gradient and the Hessian of f, respectively.

A strictly monotone line-search method for solving (1) generates a sequence $\{\mathbf{x}_k\}$ as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{2}$$

Email addresses: daniela.diserafino@unicampania.it (Daniela di Serafino), toraldo@unina.it (Gerardo Toraldo), marco.viola@unicampania.it (Marco Viola)

where $\mathbf{d}_k$ is a descent direction, $\alpha_k > 0$ is a step length and $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$. For simplicity of notation, we define $f_k = f(\mathbf{x}_k)$, $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$ and $H_k = H(\mathbf{x}_k)$. The direction $\mathbf{d}_k$ must be of strict descent, i.e.,

$$\mathbf{g}_k^\top \mathbf{d}_k < 0. \tag{3}$$

However, condition (3) alone does not ensure convergence, and $\mathbf{d}_k$ must satisfy, e.g., the angle criterion

$$\cos\langle -\mathbf{g}_k, \mathbf{d}_k \rangle = \frac{-\mathbf{g}_k^\top \mathbf{d}_k}{\|\mathbf{g}_k\| \, \|\mathbf{d}_k\|} \geq \varepsilon_k, \quad \varepsilon_k > 0, \tag{4}$$

where the sequence $\{\varepsilon_k\}$ is bounded away from 0, which means that the angle between the search direction and the Steepest Descent (SD) direction must be bounded away from the right angle.

The step length must usually satisfy the Armijo condition

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) - f(\mathbf{x}_k) \leq \sigma_1 \alpha_k \mathbf{g}_k^\top \mathbf{d}_k, \quad \sigma_1 \in (0, 1), \tag{5}$$

or the Wolfe conditions, i.e., (5) and

$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^\top \mathbf{d}_k \geq \sigma_2 \alpha_k \mathbf{g}_k^\top \mathbf{d}_k, \quad \sigma_2 \in (\sigma_1, 1). \tag{6}$$

We note that (5) forces a *sufficient decrease* in the objective function, while the *curvature* condition (6) prevents the method from taking too small steps, which is not guaranteed by condition (5) alone. This drawback can be avoided by choosing α_k with a suitable backtracking procedure [25, page 37].

In a Newton-type (NT) method, the search direction $\mathbf{d}_k^{\mathrm{NT}}$ is computed as the solution of the linear system

$$S_k \mathbf{d} = -\mathbf{g}_k \tag{7}$$

where S_k is some symmetric matrix, possibly positive definite so that (3) automatically holds. With the choice $S_k = I$, where I is the identity matrix, the NT method reduces to the classical SD method, which is globally convergent with at most linear rate. Recently, several attempts have been made to get more efficient SD methods. In particular, starting from the seminal work by Barzilai and Borwein (BB) [2], it has been observed that appropriate choices of α_k can, to some extent, remedy the slow convergence of the SD method, even for the solution of constrained problems by gradient projection strategies. This led to effective algorithms [6, 10, 12, 27, 28], which have been successfully used in several applications [1, 8, 9, 14, 31].

With the inclusion of second-order information through S_k we expect a better rate of convergence. However, the search direction $\mathbf{d}_k^{\mathrm{NT}}$ does not guarantee global convergence, even if S_k is positive definite for each k. An example is provided by the classical Newton's method, where $S_k = H(\mathbf{x}_k)$. If $\mathbf{x}^*$ solves (1), with $H(\mathbf{x}^*)$ positive definite, and $H(\mathbf{x})$ is Lipschitz continuous around $\mathbf{x}^*$, Newton's method has local, but not global, convergence with quadratic rate [16].

However, a suitable reduction of the Newton step allows global convergence in the convex case (see, e.g., [24, page 34]). In the nonconvex case the Newton direction may not be a descent direction. Therefore, modifications of Newton's method have been developed that replace $H(\mathbf{x}_k)$ by $\widetilde{H}_k = H(\mathbf{x}_k)+E_k$, where E_k is a symmetric matrix such that $\widetilde{H}_k$ is positive definite [22] and the solution $\mathbf{d}_k^{\mathrm{MN}}$ of $\widetilde{H}_k\mathbf{d} = -\mathbf{g}_k$ is a descent direction at $\mathbf{x}_k$. We will refer to these methods as Modified Newton's (MN) methods. This approach can be extended to the general framework of Newton-type methods, in which, given an approximation S_k of $H(\mathbf{x}_k)$, one may consider a matrix E_k such that $\widetilde{S}_k = S_k + E_k$ is "sufficiently positive definite" and $\|E_k\|$ is not much larger than $\inf\{\|E\| \,:\, S_k + E \succ 0\}$ for some norm [15]. If the eigenvalues of $\widetilde{S}_k$ are bounded away from zero independently of k and the Armijo condition is satisfied by backtracking, then all limit points of the method using the directions obtained by solving (7) with $S_k = \widetilde{S}_k$ are stationary for (1)[3, 23]. The most successful and well-established algorithms for the computation of E_k are based on modified Cholesky factorizations of the matrix $H(\mathbf{x}_k)$ [15]. Another possibility is setting $E_k = \lambda_k I$, where $\lambda_k > 0$ is a suitable constant, so that the search direction is

$$\mathbf{d}_k = -\left(S_k + \lambda_k I\right)^{-1}\mathbf{g}_k. \tag{8}$$

For quasi-Newton methods ad-hoc globalization strategies have been proposed which avoid matrix factorizations. Next, we briefly describe some of them for the BFGS method [16]. In this case

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k, \tag{9}$$

with $\mathbf{d}_k$ solution of the system

$$B_k\mathbf{d} = -\mathbf{g}_k, \tag{10}$$

where the matrix $B_k \in \mathbb{R}^{n\times n}$ is updated by the formula

$$B_{k+1} = B_k - \frac{B_k\mathbf{s}_k\mathbf{s}_k^\top B_k}{\mathbf{s}_k^\top B_k\mathbf{s}_k} + \frac{\mathbf{y}_k\mathbf{y}_k^\top}{\mathbf{y}_k^\top\mathbf{s}_k}, \tag{11}$$

with $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ and $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$.

For convex optimization problems it can be shown that, under suitable hypotheses, the BFGS method with a line search satisfying the Wolfe conditions is globally convergent and locally superlinearly convergent [4]. For nonconvex functions Dai [7] showed with an example that the BFGS with Wolfe line search may fail. Later on Mascharenhas [20] showed that the BFGS method, as well as other methods in the Broyden class, may fail for nonconvex objective functions when an exact line search is used.

For nonconvex minimization problems Li and Fukushima [18] proposed a modified version of BFGS, called MBFGS, using an Armijio line search or a Wolfe one, and based on an update formula for the matrix in (10), which is equal to (11) with $\mathbf{y}_k$ replaced by

$$\overline{\mathbf{y}}_k = \mathbf{g}_{k+1} - \mathbf{g}_k + \gamma_k\|\mathbf{g}_k\|\mathbf{s}_k, \tag{12}$$

where

$$\gamma_k = 1 + \max\left\{-\frac{\mathbf{y}_k^\top \mathbf{s}_k}{\|\mathbf{s}_k\|^2}, 0\right\}. \tag{13}$$

This update formula guarantees that [18, Section 5]

$$\overline{\mathbf{y}}_k^\top \mathbf{s}_k > \|\mathbf{g}_k\| \, \|\mathbf{s}_k\|^2,$$

and therefore B_{k+1} is positive definite, provided B_k is positive definite, thus ensuring that the descent condition $\mathbf{g}_{k+1}^\top \mathbf{d}_{k+1} < 0$ holds. The update formula (13) was inspired by the MN method with search direction

$$\mathbf{d}_k^{\mathrm{MN}} = -(H_k + \lambda_k I)^{-1} \mathbf{g}_k. \tag{14}$$

where λ_k is a regularization parameter. The MBFGS method with Armijo or Wolfe line search is globally convergent even for nonconvex problems [18].

Later on, the same authors proposed the following BFGS formula with *cautious update rule* [19]:

$$B_{k+1} = \begin{cases} B_{k+1} = B_k - \frac{B_k \mathbf{s}_k \mathbf{s}_k^\top B_k}{\mathbf{s}_k^\top B_k \mathbf{s}_k} + \frac{\mathbf{y}_k \mathbf{y}_k^\top}{\mathbf{y}_k^\top \mathbf{s}_k}, & \text{if } \frac{\mathbf{y}_k^\top \mathbf{s}_k}{\|\mathbf{s}_k\|^2} > \chi \|\mathbf{g}_k\|^\upsilon, \\ B_k, & \text{otherwise,} \end{cases} \tag{15}$$

where χ and υ are positive constants, and for which global convergence was proved without convexity assumptions.

In this paper we consider a globalization approach applicable to any Newton-type method. The basic idea consists of linearly combining the NT and SD directions. The goal is to bring the iterates sufficiently close to a solution through the globally convergent SD method, so that once the iterates are in the basin of attraction of the NT method, it can lead to faster convergence than SD. Although this approach is not new (see, e.g., [29, 30, 17] and [3, Section 1.4.4]) and much simpler than those based on trust regions and incomplete factorizations, it has been little utilized, likely because of little confidence in the SD method. Here we show that a hybrid strategy which combines SD and NT directions can give very interesting numerical results. In our opinion, a suitable scaling of the SD direction is a key issue in making this approach effective. We also find that, besides fostering global convergence, this strategy can be effective in speeding up NT methods.

This article is organized as follows. In Section 2 we present our globalization strategy, and how the coefficient governing the linear combination can be computed. Section 3 deals with the convergence of the resulting algorithm, in particular for Newton's method. In Section 4 we discuss results of numerical experiments carried out with our algorithm using either Newton's or the BFGS method, including a comparison with some benchmarks algorithms. We conclude in Section 5.

Algorithm 1 SD Globalized (SDG) line-search method

choose $\mathbf{x}_0 \in \mathbb{R}^n$, $\{\varepsilon_k\}_{k\in\mathbb{N}} \subset (0,\,1)$, $\sigma_1 \in (0, 1/2)$;
for $k = 0, 1, 2, \ldots$ **do**
    compute S_k;
    $\mathbf{d}_k^{\mathrm{NT}} = -S_k^{-1}\mathbf{g}_k$;
    if $\cos\left\langle \mathbf{d}_k^{\mathrm{NT}}, -\mathbf{g}_k \right\rangle \geq \varepsilon_k$ **then**
        $\mathbf{d}_k = \mathbf{d}_k^{\mathrm{NT}}$;
    else
        compute the step length ξ_k;
        find β_k such that $\cos\left\langle \beta_k \mathbf{d}_k^{\mathrm{NT}} - (1-\beta_k)\xi_k\,\mathbf{g}_k, -\mathbf{g}_k \right\rangle \geq \varepsilon_k$;
        $\mathbf{d}_k = \beta_k\,\mathbf{d}_k^{NT} - (1-\beta_k)\,\xi_k\,\mathbf{g}_k$;
    end if
    select α_k satisfying (5) by backtracking, starting from $\alpha_k = 1$;
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$;
end for

2. Globalization strategy

We propose a line-search method of the form (2), where $\mathbf{d}_k$ is the NT direction if the angle criterion (4) is satisfied, otherwise

$$\mathbf{d}_k = \beta_k\,\mathbf{d}_k^{\mathrm{NT}} - (1-\beta_k)\,\xi_k\,\mathbf{g}_k, \tag{16}$$

with $0 \leq \beta_k \leq 1$ and $\xi_k > 0$. A sketch of this method, which we call *SD Globalized (SDG) line-search method*, is provided in Algorithm 1.

Regarding the NT direction (7), which can be, e.g., a Newton, quasi-Newton or inexact Newton direction, we assume that it is well-scaled, so that we scale only the SD direction through ξ_k. The search direction (16) is closely related to the one proposed by Shi in [29, 30], which is defined as

$$\mathbf{d}_k = \beta_k\,\mathbf{d}_k^{\mathrm{NT}} - (1-\beta_k)\,\mathbf{g}_k. \tag{17}$$

Notice that, unlike (17), the search direction (16) is invariant to the scaling of the objective function, as long as $\xi_k\,\mathbf{g}_k$ is invariant (e.g., when ξ_k is a BB step length and $\xi_0 = 1/\|\mathbf{g}_0\|$).

The next theorem shows how to compute values of β_k guaranteeing that (16) satisfies (4). We note that the first part slightly generalizes Lemma 2.1 in [29].

Theorem 1. *Let us consider any $\mathbf{x}_k \in \mathbb{R}^n$ and $\varepsilon_k \in (0,1)$, and assume that*

$$-\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}} < \varepsilon_k \|\mathbf{g}_k\|\,\|\mathbf{d}_k^{\mathrm{NT}}\|, \tag{18}$$

with $\mathbf{d}_k^{\mathrm{NT}}$ solution of system (7) where S_k is any symmetric matrix not multiple of I.

i) Let β_k^ε be the smallest root in $(0,1)$ of the polynomial

$$P_k(\beta) = A_k\beta^2 + B_k\beta + C_k, \tag{19}$$

where

$$\begin{aligned} A_k &= \left(\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}\right)^2 - \varepsilon_k^2 \|\mathbf{g}_k\|^2 \, \|\mathbf{d}_k^{\mathrm{NT}}\|^2 - B_k - C_k, \\ B_k &= -2\left(1-\varepsilon_k^2\right) \xi_k \, \|\mathbf{g}_k\|^2 \left(\xi_k \, \|\mathbf{g}_k\|^2 + \mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}\right), \\ C_k &= \left(1-\varepsilon_k^2\right) \xi_k^2 \, \|\mathbf{g}_k\|^4 . \end{aligned}$$

If $\mathbf{d}_k$ *is defined according to* (16) *with* $\beta_k = \beta_k^\varepsilon$*, then*

$$-\mathbf{g}_k^\top \mathbf{d}_k = \varepsilon_k \|\mathbf{g}_k\| \, \|\mathbf{d}_k\|.$$

ii) Let $\mathbf{d}_k$ *be defined according to* (16)*. Then*

$$-\mathbf{g}_k^\top \mathbf{d}_k \geq \varepsilon_k \|\mathbf{g}_k\| \, \|\mathbf{d}_k\| \tag{20}$$

if and only if $\beta_k \leq \beta_k^\varepsilon$*.*

iii) A lower bound for β_k^ε *is provided by the value* $\hat{\beta}_k$ *defined as*

$$\hat{\beta}_k = \frac{\rho_k}{\rho_k + \pi_k}, \tag{21}$$

where

$$\rho_k = \xi_k(1-\varepsilon_k) \quad \text{and} \quad \pi_k = \frac{\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\|^2} + \varepsilon_k \frac{\|\mathbf{d}_k^{\mathrm{NT}}\|}{\|\mathbf{g}_k\|}. \tag{22}$$

Proof. We first prove that β_k^ε is well defined. Let us consider

$$\Phi_k(\beta) = \frac{-\mathbf{g}_k^\top \left(\beta \, \mathbf{d}_k^{\mathrm{NT}} - (1-\beta) \, \xi_k \, \mathbf{g}_k\right)}{\|\mathbf{g}_k\| \, \|\beta \, \mathbf{d}_k^{\mathrm{NT}} - (1-\beta) \, \xi_k \, \mathbf{g}_k\|},$$

which is a continuous function of β. Note that $\Phi_k(\beta)$ is the cosine of the angle between the antigradient $-\mathbf{g}_k$ and the direction

$$\mathbf{d}(\beta) = \beta \, \mathbf{d}_k^{\mathrm{NT}} - (1-\beta)\xi_k \, \mathbf{g}_k,$$

which spans continuously the cone between $-\mathbf{g}_k$, corresponding to $\mathbf{d}(0)$, and the vector $\mathbf{d}_k^{\mathrm{NT}}$, corresponding to $\mathbf{d}(1)$. From its definition it is clear that $\Phi_k(\beta)$ is a monotonically decreasing function in the interval (0, 1). Since

$$\Phi_k(0) = 1 \;\text{ and }\; \Phi_k(1) = \frac{-\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\| \, \|\mathbf{d}_k^{\mathrm{NT}}\|} < \varepsilon_k,$$

$\Phi_k(\beta) - \varepsilon_k$ has a unique zero β_k^ε in $(0,1)$. The solutions of the equation

$$\left[-\mathbf{g}_k^\top \left(\beta \, \mathbf{d}_k^{\mathrm{NT}} - (1-\beta) \, \xi_k \, \mathbf{g}_k\right)\right]^2 = \left(\varepsilon_k \|\mathbf{g}_k\| \, \|\beta \, \mathbf{d}_k^{\mathrm{NT}} - (1-\beta) \, \xi_k \, \mathbf{g}_k\|\right)^2 \tag{23}$$

are the solutions of $\Phi_k(\beta) = \pm\varepsilon_k$. By simple computations, it is easy to verify that the solutions of (23) are the roots of the polynomial (19). Now we observe that $P_k(0) = C_k > 0$. To conclude the proof of item i) we need to analyze the two possible cases about the sign of

$$P_k(1) = \left(\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}\right)^2 - \varepsilon_k^2 \|\mathbf{g}_k\|^2 \, \|\mathbf{d}_k^{\mathrm{NT}}\|^2.$$

a) If $-\varepsilon_k < \frac{-\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\|\,\|\mathbf{d}_k^{\mathrm{NT}}\|} < \varepsilon_k$, then $P_k(1) < 0$ and β_k^ε is the only root of (19) in $(0, 1)$;

b) if $\frac{-\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\|\,\|\mathbf{d}_k^{\mathrm{NT}}\|} < -\varepsilon_k$, then $P_k(1) > 0$ and β_k^ε is the smallest of the two roots of (19) in $(0, 1)$.

Item ii) of the theorem comes from the observation that $\Phi_k(\beta)$ is a monotonically decreasing function in (0, 1), which implies that $\Phi_k(\beta) \geq \Phi_k(\beta_k^\varepsilon) = \varepsilon_k$ for all $\beta \leq \beta_k^\varepsilon$.

To prove item iii), we note that the search direction (16) satisfies (20) if and only if

$$\frac{-\beta_k\, \mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}} + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2}{\|\mathbf{g}_k\|\, \|\beta_k \mathbf{d}_k^{\mathrm{NT}} - (1-\beta_k)\, \xi_k\, \mathbf{g}_k\|} \geq \varepsilon_k. \tag{24}$$

Since

$$\frac{-\beta_k \mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}} + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2}{\|\mathbf{g}_k\|\, \|\beta_k \mathbf{d}_k^{\mathrm{NT}} - (1-\beta_k)\, \xi_k\, \mathbf{g}_k\|} \geq \frac{-\beta_k\, \mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}} + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2}{\beta_k \|\mathbf{g}_k\|\, \|\mathbf{d}_k^{\mathrm{NT}}\| + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2},$$

a sufficient condition for (24) to hold is that

$$\frac{-\beta_k\, \mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}} + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2}{\beta_k \|\mathbf{g}_k\|\, \|\mathbf{d}_k^{\mathrm{NT}}\| + (1-\beta_k)\, \xi_k \|\mathbf{g}_k\|^2} \geq \varepsilon_k. \tag{25}$$

This condition gives us a way to compute a lower bound $\hat{\beta}_k$ for β_k^ε. By straightforward computations one can check that (25) is equivalent to

$$\beta_k \left(\xi_k(1-\varepsilon_k) + \frac{\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\|^2} + \varepsilon_k \frac{\|\mathbf{d}_k^{\mathrm{NT}}\|}{\|\mathbf{g}_k\|} \right) \leq \xi_k(1-\varepsilon_k),$$

i.e.,

$$\beta_k\, (\rho_k + \pi_k) \leq \rho_k \tag{26}$$

where ρ_k and π_k are defined in (22). We observe that (18) implies $\pi_k > 0$ and $\rho_k > 0$ comes from the definition of ξ_k and ε_k. Therefore, we have that (26) holds if and only if $\beta_k \leq \hat{\beta}_k = \frac{\rho_k}{\rho_k + \pi_k} < 1$. Item ii) implies $\hat{\beta}_k \leq \beta_k^\varepsilon$. □

Remark 1. *Items i) and iii) of the previous theorem suggest two choices for the coefficient β_k in* (16)*, namely β_k^ε and $\hat{\beta}_k$. Note that β_k^ε is the largest value of β_k such that the angle criterion* (20) *is satisfied. Moreover, by looking at the definition of π_k in* (22) *we can easily find a relation between the "quality" of the NT direction and the value of $\hat{\beta}_k$. We can indeed write π_k as*

$$\pi_k = \frac{\|\mathbf{d}_k^{\mathrm{NT}}\|}{\|\mathbf{g}_k\|} \left(\frac{\mathbf{g}_k^\top \mathbf{d}_k^{\mathrm{NT}}}{\|\mathbf{g}_k\|\, \|\mathbf{d}_k^{\mathrm{NT}}\|} + \varepsilon_k \right) = -\frac{\|\mathbf{d}_k^{\mathrm{NT}}\|}{\|\mathbf{g}_k\|} \left(\cos\left\langle -\mathbf{g}_k,\, \mathbf{d}_k^{\mathrm{NT}} \right\rangle - \varepsilon_k \right),$$

i.e., π_k provides a measure of the violation of the angle criterion (20)*. If $\cos\left\langle -\mathbf{g}_k,\, \mathbf{d}_k^{\mathrm{NT}} \right\rangle$ approaches ε_k, we have that π_k tends to zero and $\hat{\beta}_k$ tends*

to 1*, allowing us to take a direction close to the NT one. Conversely, if* $\cos\langle -\mathbf{g}_k, \mathbf{d}_k^{\mathrm{NT}}\rangle$ *approaches* -1*, the value of* π_k *may increase, implying a decrease of* $\hat{\beta}_k$ *and thus fostering the descent direction to be close to the SD direction.*

Going back to the classical globalization strategies mentioned in the previous section, we note that the search direction

$$\mathbf{d}_k = -\widetilde{S}_k^{-1}\mathbf{g}_k, \tag{27}$$

where $\widetilde{S}_k = S_k + E_k$, is based on the following quadratic approximation of f at $\mathbf{x}_k$:

$$\psi_k(\mathbf{x}) = f(\mathbf{x}_k) + \mathbf{g}_k^\top(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^\top \widetilde{S}_k(\mathbf{x}-\mathbf{x}_k), \tag{28}$$

in which the role of E_k is to guarantee that the model be "sufficiently" convex. Our approach is based on a different second-order model, namely

$$\phi_k(\mathbf{x}) = f(\mathbf{x}_k) + \mathbf{g}_k^\top(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^\top W_k(\mathbf{x}-\mathbf{x}_k),$$

where

$$W_k = \left(\beta_k S_k^{-1} + (1-\beta_k)\xi_k I\right)^{-1}. \tag{29}$$

Even when W_k is not positive definite, the choice of β_k guarantees that (20) holds for $\mathbf{d}_k = -W_k^{-1}\,\mathbf{g}_k$. A simple computation shows that we only require convexity for the univariate function

$$\theta(\alpha) = \phi_k\left(\mathbf{x}_k - \alpha W_k^{-1}\mathbf{g}_k\right), \tag{30}$$

which attains its minimum at $\alpha = 1$. On the contrary, a globalization strategy like the one based on a shifted linear system of the form (8) forces the overall quadratic model (28) to be convex, potentially leading to a model insufficiently faithful to f.

The directions (8) and (16) remind us of Trust Region (TR) methods [5]. These methods compute $\mathbf{x}_{k+1}$ by minimizing a quadratic model of f near $\mathbf{x}_k$,

$$\varphi_k(\mathbf{x}) = f(\mathbf{x}_k) + \mathbf{g}_k^\top(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^\top S_k(\mathbf{x}-\mathbf{x}_k), \quad \|\mathbf{x}-\mathbf{x}_k\| \le \Delta_k, \tag{31}$$

where Δ_k is updated at each iteration to get a "good" approximation of $f(\mathbf{x})$ in the ball with center $\mathbf{x}_k$ and radius Δ_k. The point $\mathbf{x}_{k+1}$ is a minimizer of $\varphi_k(\mathbf{x})$ in this ball if and only if $\mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ is a solution of the system

$$\begin{array}{ll} (S_k + \lambda I)\,\mathbf{d} & = -\mathbf{g}_k, \\ \lambda\,(\Delta_k - \|\mathbf{d}\|) & = 0, \end{array} \tag{32}$$

for a scalar $\lambda \ge 0$ such that $S_k + \lambda I \succcurlyeq 0$. Therefore, we can regard direction (8) as a TR step. Direction (16) can be also related to the dogleg and the two-dimensional subspace minimization approaches, which provide approximate

solutions to the TR subproblem (31) by making a search in the space spanned by the SD and Newton directions (see, e.g., [25]).

As already pointed out, unlike (17), the search direction (16) is invariant to the scaling of the objective function when $\xi_k \, \mathbf{g}_k$ is invariant. In order to clarify the relevance of this issue, we show a simple numerical example. We consider the so-called "Brown badly-scaled" function [21], defined as

$$f_B(x_1, x_2) = (x_1 - 10^6)^2 + (x_2 - 2 \cdot 10^{-6})^2 + (x_1 x_2 - 2)^2. \tag{33}$$

We compare the performances of the line-search methods using as search directions respectively (17) and (16) (with ξ_k set as the BB2 step length defined in [2, equation (5)]), on scaled versions of (33),

$$f_\omega(x_1, x_2) = \omega f_B(x_1, x_2),$$

for different values of the scale factor ω. For both algorithms, $\mathbf{d}^{NT}$ is set as the Newton direction. Furthermore, we set $\varepsilon_k = 10^{-3}$ for each k and $\|\nabla f_\omega(x_1, x_2)\| < 10^{-5}\,\omega$ as stop condition. Table 1 contains the number of iterations (its) and function evaluations (evals) performed by the SDG method; as expected, when the search directions (17) are used, the performance of the line-search algorithm dramatically depends on the scale factor. On the other hand, the results in Table 1 confirm that the search directions computed using (16) are scale invariant. Notice that the Newton direction is scale invariant, whereas the gradient scales with ω, so that the larger the scale factor, the closer (17) is to the SD direction. This probably explains why in Table 1 we observe a progressive deterioration of the performance of the search direction (17) when moving from $\omega = 10^{-3}$ (close to the Newton direction) to $\omega = 10^3$ (close to the SD direction). Finally, a comparison between the two search directions for $\omega = 1$ demonstrates the importance of a suitable choice of ξ_k in the hybrid strategy (note that in (17) it is always $\xi_k = 1$).

Table 1: Comparison of line-search methods with directions (16) and (17) on the solution of Brown badly-scaled function with different scalings.

	$\mathbf{d}_k$ as in (16)		$\mathbf{d}_k$ as in (17)	
ω	its	evals	its	evals
10^{-3}	6	12	6	19
10^{-2}	6	12	6	20
10^{-1}	6	12	10	32
1	6	12	15	42
10	6	12	19	46
10^2	6	12	34	86
10^3	6	12	224	1683

3. Convergence

We now focus on the convergence properties of the SDG method. The theorems in this section are a slight modification of results presented by the authors in [11].

Theorem 2. *Let $f \in C^2(\mathbb{R}^n)$ and assume that $\{\varepsilon_k\}$ is bounded away from 0. Then, for any $\mathbf{x}_0$, the limit points of the sequence $\{\mathbf{x}_k\}$ generated by Algorithm 1 are stationary.*

Proof. Let $0 < \varepsilon_{min} = \inf\{\varepsilon_k\}$. At any iteration k, Theorem 1 guarantees the existence of a coefficient β_k for step 9 of Algorithm 1, therefore a direction $\mathbf{d}_k$ satisfying $\cos\langle \mathbf{d}_k, -\mathbf{g}_k\rangle \geq \varepsilon_k$ can be found. Since the sequence $\{\varepsilon_k\}$ is bounded from below by ε_{min} and α_k is obtained by a backtracking technique to fulfill condition (5), the thesis follows from [3, Proposition 1.2.1]. □

The next theorem shows that the SDG method has quadratic convergence rate when the direction $\mathbf{d}_k^{\mathrm{NT}}$ is the Newton direction. The proof is omitted because it is practically the same as the proof of Theorem 2 in [11].

Theorem 3. *Let $f \in C^2(\mathbb{R}^n)$ and let $\{\mathbf{x}_k\}$ be generated by the SDG method where $\mathbf{d}_k^{\mathrm{NT}}$ is the Newton direction. Let σ_1 in (5) be such that $0 < \sigma_1 < \frac{1}{2}$ and the sequence $\{\varepsilon_k\}$ be nonincreasing with limit $\varepsilon_{min} > 0$. Suppose also that there exists a limit point $\hat{\mathbf{x}}$ of $\{\mathbf{x}_k\}$ where $H(\hat{\mathbf{x}})$ is positive definite and $H(\mathbf{x})$ is Lipschitz continuous around $\hat{\mathbf{x}}$. If ε_{min} is sufficiently small, then $\{\mathbf{x}_k\}$ converges to $\hat{\mathbf{x}}$ with quadratic rate.*

4. Computational experiments

We implemented two MATLAB versions of the SDG algorithm, using Newton's method and the BFGS method, and compared them with the Modified Newton method using a modified Cholesky factorization [15] and with the CBFGS method [19], respectively. To better understand the effect of our globalization strategy, we also run Newton's method, the BFGS method, and the SD one with the BB2 Barzilai-Borwein step length used in the numerical example at the end of Section 2.

In the SDG and pure SD methods, we set $\xi_0 = 1/\|\mathbf{g}_0\|$, $\xi_k = \max\{\xi_k^{\mathrm{BB2}}, \nu_1\}$ if $\xi_k^{\mathrm{BB2}} > 0$, and $\xi_k = \min\{10\,\xi_{k-1}^{\mathrm{BB2}}, \nu_2\}$ otherwise; here ξ_k^{BB2} is the BB2 step length, $\nu_1 = 10^{-5}$ and $\nu_2 = 10^5$. We chose BB2 instead of other BB step lengths (see, e.g., [10]) because BB2 was more effective in preliminary numerical experiments. The Hessian approximations in SDG with BFGS, in BFGS and in CBFGS were initialized as explained in [25, page 143]. We applied a shrinking strategy for the selection of ε_k in (20): given $\varepsilon_0 \in (0, 1)$ and $\zeta = 0.95$, at the k-th iterate ($k > 0$) we set $\varepsilon_k = \zeta\,\varepsilon_{k-1}$ if $\beta_{k-1} < 1$, and $\varepsilon_k = \varepsilon_{k-1}$ otherwise. To prevent the sequence $\{\varepsilon_k\}$ from going toward zero, we set a threshold for ε_k equal to $\overline{\varepsilon} = 10\,\varepsilon_{mac}$, where ε_{mac} is the machine epsilon. It is worth noting that in none of the tests performed the value of ε_k reached the threshold. In all

the algorithms, the Armijo backtracking line search with $\sigma_1 = 10^{-4}$ (see (5)) and quadratic interpolation [25, Section 3.5] was performed. The methods were stopped as soon as

$$\|\mathbf{g}_k\| < \tau_g \|\mathbf{g}_0\|, \tag{34}$$

with $\tau_g = 10^{-5}$; as a safeguard we also stopped the execution when a maximum number, k_{max}, of 2000 iterations was achieved or the objective function appeared to get stuck, i.e.,

$$\frac{f(\mathbf{x}_{k-1}) - f(\mathbf{x}_k)}{|f(\mathbf{x}_{k-1})|} < \overline{\varepsilon}.$$

Algorithm 2 is a detailed version of Algorithm 1 that includes implementation details. By numerical experiments we found that the use of the pure SD direction with BB2 step length is computationally convenient when $\mathbf{d}_k = \mathbf{d}_k^{\mathrm{NT}}$ is not a descent direction (see lines 12-13). The notation SDG[NT,ε_0] highlights that the algorithm uses a selected NT method (e.g., Newton's or BFGS) and ε_0 as initial value of the sequence $\{\varepsilon_k\}$.

All the experiments were carried out using MATLAB R2018b. Comparisons were performed by using the performance profiles introduced in [13], which are briefly described next for completeness.

Let $\mathcal{S}_{\mathcal{T},\mathcal{A}} \geq 0$ be a statistic corresponding to the solution of a test problem $\mathcal{T}$ by an algorithm $\mathcal{A}$, and suppose that the smaller the statistic the better the algorithm. Furthermore, let $\mathcal{S}_{\mathcal{T}}$ be the smallest value attained on the test $\mathcal{T}$ by one of the algorithms under analysis. The performance profile of the algorithm $\mathcal{A}$ is defined as

$$\pi(\chi) = \frac{\text{number of tests such that } \mathcal{S}_{\mathcal{T},\mathcal{A}}/\mathcal{S}_{\mathcal{T}} \leq \chi}{\text{number of tests}}, \quad \chi \geq 1,$$

where the ratio $\mathcal{S}_{\mathcal{T},\mathcal{A}}/\mathcal{S}_{\mathcal{T}}$ is set to $+\infty$ if algorithm $\mathcal{A}$ fails in solving $\mathcal{T}$. In other words, $\pi(\chi)$ is the fraction of problems for which $\mathcal{S}_{\mathcal{T},\mathcal{A}}$ is within a factor χ of the smallest value $S_{\mathcal{T}}$. Thus $\pi(1)$ is the percentage of problems for which $\mathcal{A}$ is the best, while $\lim_{\chi\to+\infty} \pi(\chi)$ gives the percentage of problems that are successfully solved by $\mathcal{A}$.

The performance profiles considered in this work use as performance statistics the number of iterations and the number of function evaluations. We note that in Section 4.2.2, in comparing our SDG algorithm based on Newton's method with an MN method, we do not consider the execution time because the MN implementation exploits a C code for the modified Cholesky factorization, called via a MATLAB mex file, while the Newton systems in the SDG algorithm are solved by the MATLAB function backslash. Thus, in this case a time comparison would be unfair. In all the other experiments the algorithms compared have about the same cost per iteration, therefore using the number of iterations as performance statistic appears sensible. Furthermore, the number of objective function evaluations is related to the number of line searches performed, and provides also information on the quality of the descent direction. Nevertheless, we use a performance profile based on the execution time at the end of Section 4.2.2, to further support some results.

Algorithm 2 SDG[NT,ε_0]

choose an NT method and $\varepsilon_0 \in (0,1)$;
choose $\mathbf{x}_0 \in \mathbb{R}^n$, $\zeta \in (0,1)$, $\sigma_1 \in (0,1/2)$, $\overline{\varepsilon} = 10\,\varepsilon_{mac}$, $\tau_g \in (0,1)$, $k_{\max} \in \mathbb{N}$;
$k = 0$; *continue* = true;
while $(\|\mathbf{g}_k\| \geq \tau_g \|\mathbf{g}_0\|) \wedge$ *continue* **do**
    compute S_k according to the NT method;
    $\mathbf{d}_k^{\mathrm{NT}} = -S_k^{-1}\mathbf{g}_k$;
    if $\cos\langle \mathbf{d}_k^{\mathrm{NT}}, -\mathbf{g}_k \rangle \geq \varepsilon_k$ **then**
        $\mathbf{d}_k = \mathbf{d}_k^{\mathrm{NT}}$;
        $\varepsilon_{k+1} = \varepsilon_k$;
    else
        compute the step length ξ_k;
        if $\cos\langle \mathbf{d}_k^{\mathrm{NT}}, -\mathbf{g}_k \rangle \leq 0$ **then**
            $\mathbf{d}_k = -\xi_k \mathbf{g}_k$;
        else
            compute $\hat{\beta}_k$ as defined in (21);
            $\mathbf{d}_k = \hat{\beta}_k \mathbf{d}_k^{\mathrm{NT}} - (1-\hat{\beta}_k)\xi_k\,\mathbf{g}_k$;
        end if
        $\varepsilon_{k+1} = \max\{\overline{\varepsilon},\ \zeta\,\varepsilon_k\}$;
    end if
    select α_k satisfying (5) by backtracking with quadratic interpolation [25, Section 3.5], starting from $\alpha_k = 1$;
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$;
    $k = k + 1$;
    if $(k > k_{max}) \vee (|f_{k-1} - f_k| < \overline{\varepsilon}\,|f_{k-1}|)$ **then**
        continue = false;
    end if
end while

To better show the effectiveness of the proposed strategy, we considered two sets of test problems, described in the following section.

4.1. Test problems

4.1.1. Nonconvex problems

We considered 36 problems available from `https://people.sc.fsu.edu/~jburkardt/m_src/test_opt/test_opt.html`, including the unconstrained minimization problems from the Moré-Garbow-Hillstrom collection [21] and other problems. We set the problem size equal to 100 for all the problems where the dimension could be chosen by the user. For each problem we used 10 starting points, i.e., the point $\mathbf{x}_0$ provided with the problem and the points $\mathbf{x}_0^s$, with $s = 1, \ldots, 9$, where $(\mathbf{x}_0^s)_i = (\mathbf{x}_0)_i + \gamma_i^s$, γ_i^s was a random number in $[-\eta_s a_i, \eta_s a_i]$, $a_i = |(\mathbf{x}_0)_i|$, and the values η_s were logarithmically spaced in the interval $\left[10^{-2}, 10^{-1}\right]$. These choices resulted in a set of 360 nonconvex optimization problem instances.

4.1.2. Convex problems coming from machine learning

The second set of test problems consists in the minimization of convex functions arising from machine learning. In particular, given N pairs $(\mathbf{a}_i, b_i)$, where $\mathbf{a}_i \in \mathbb{R}^n$ and $b_i \in \{-1, 1\}$, we considered the problem of training a linear classifier by minimizing the function

$$f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x}) + \frac{\mu}{2} \|\mathbf{x}\|^2, \tag{35}$$

where $f_i(x) = \log\left(1 + e^{-b_i \, \mathbf{a}_i^\top \mathbf{x}}\right)$ and $\mu > 0$.

Table 2: Number of points and number of features for each machine learning dataset. We indicate the source by adding a superscript to the dataset name, according to the following list:
1 - `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`,
2 - `http://www.ics.uci.edu/~mlearn/MLRepository.html`,
3 - NAACCR Incidence - CiNA Public File, 1995-2015, North American Association of Central Cancer Registries,
4 - `http://yann.lecun.com/exdb/mnist`.

name	points	features
a6a[1]	11220	123
a7a[1]	16100	123
a8a[1]	22696	123
a9a[1]	32561	123
adult[2]	48842	122
cina[3]	16033	132
cod-rna[1]	59535	8
ijcnn1[1]	49990	22
mnist[4]	7603	100
mushrooms[1]	8124	112
phishing[1]	11055	68
w6a[1]	17188	300
w7a[1]	24692	300
w8a[1]	49749	300

We considered 14 datasets, whose dimensions and sources are reported in Table 2. For each dataset we considered a 10-fold cross validation setting, thus obtaining 10 different training problems of the form (35) with N approximately equal to 0.9 times the total number of points. For each problem we set $\mu = \frac{1}{N}$, which is a choice usually found in literature. This produced a total of 140 instances for training our strategy. For this set of test problems we focused on the BFGS method.

4.2. Numerical results

4.2.1. Comparison on the choice of β_k

First, we focused on the choice of β_k in (16). Theorem 1 suggests β_k^ε and $\hat{\beta}_k$ as two possible alternatives. Just to get a first picture, we considered the so-called "Gulf research and developement" function [21], defined as

$$f_{GRD}(x_1, x_2, x_3) = \sum_{i=1}^{99} \left[\exp\left(-\frac{\left| \left(-50 \log(\frac{i}{100})\right)^{\frac{2}{3}} + 25 - x_2 \right|^{x_3}}{x_1} \right) - \frac{i}{100} \right]^2,$$

with starting point $\mathbf{x}_0 = [40, 20, 1.2]^\top$. We ran SDG[Newton,0.5] with $\beta_k = \hat{\beta}_k$ and computed also β_k^ε at each iteration. The values of β_k^ε and $\hat{\beta}_k$ are shown in the top plot in Figure 1, for the iterations in which the Newton step was rejected as a search direction. For the same iterations, in the bottom plot we depicted the values of $\cos\langle \mathbf{d}_k, -\mathbf{g}_k \rangle$ for $\mathbf{d}_k$ in (16), computed with $\beta_k = \beta_k^\varepsilon$, $\beta_k = \hat{\beta}_k$ and $\beta_k = 1$ (the last one corresponds to the pure Newton's method). This example suggests that the difference between β_k^ε and $\hat{\beta}_k$ is negligible, especially when close to the solution. Conversely, the angle between $\mathbf{d}_k$ (computed with either β_k^ε or $\hat{\beta}_k$) and $\mathbf{d}_k^{\text{Newton}}$ is non-negligible, especially far from the solution.

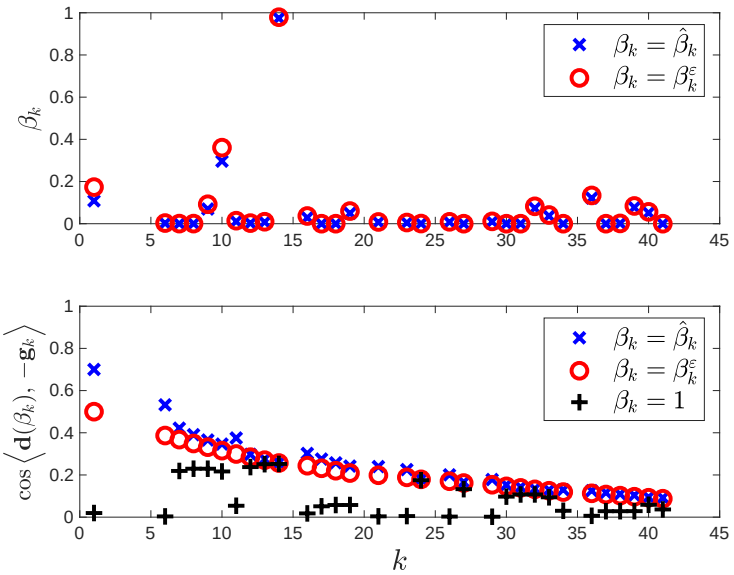

Figure 1: Test on the Gulf research and development function. Top plot: values of β_k^ε and $\hat{\beta}_k$. Bottom plot: values of $\cos\langle \mathbf{d}(\beta_k), -\mathbf{g}_k \rangle$, with $\mathbf{d}(\beta) = \beta\, \mathbf{d}_k^{\text{Newton}} - (1-\beta)\xi_k\, \mathbf{g}_k$ for $\beta_k = \beta_k^\varepsilon$, $\beta_k = \hat{\beta}_k$ and $\beta_k = 1$.

Then we ran two versions of SDG[Newton,0.5], with $\beta_k = \beta_k^\varepsilon$ and $\beta_k = \hat{\beta}_k$, on the solution of the 360 nonconvex problem instances previously described,

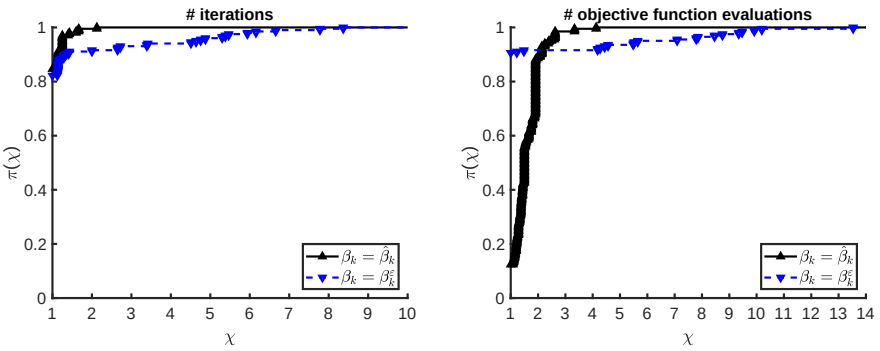

Figure 2: Performance profiles of SDG[Newton,0.5], with $\beta_k = \beta_k^\varepsilon$ and $\beta_k = \hat{\beta}_k$, on the solution of the 202 nonconvex problem instances in which the two algorithms reached the same solution.

looking for experimental evidence about the choice of β_k. Since the problems are nonconvex, different algorithms may reach different local minima starting from the same point. We noted that, out of the 360 considered problem instances, SDG went to the smallest local minimum 267 times with $\beta_k = \hat{\beta}_k$, and 295 times with $\beta_k = \beta_k^\varepsilon$. To have a fair picture, we compared the two versions of SDG on the 202 instances where they reached equal solutions (two solutions were considered equal if they coincided up to the third significant digit). The performance profiles reported in Figure 2 show a comparison in terms of number of iterations (left) and number of objective function evaluations (right), and suggest that $\beta_k = \hat{\beta}_k$ is preferable to β_k^ε. If we had to venture a guess, based on our experience, we would say that far from the solution $\hat{\beta}_k$ can be significantly smaller then β_k^ε, and this increases the SD component in (16). Far from the solution, the SD direction with a suitable step length like BB2 can even be more effective than Newton's method in decreasing the objective function, and this might explain, to some extent, the results in Figure 2.

We also compared SDG[BFGS,0.5] with β_k^ε and $\hat{\beta}_k$ on the first set of test problems. In this case, the two versions of SDG computed the same solution on 148 problem instances. As shown by the performance profiles in Figure 3, the implementation with $\beta_k = \hat{\beta}_k$ slightly outperformed the one with $\beta_k = \beta_k^\varepsilon$. A similar analysis was carried out for the convex problems from machine learning, comparing the versions SDG[BFGS,0.5] with the two different values of β_k. Again, $\beta_k = \hat{\beta}_k$ seems to provide the best results, in terms of both number of iterations and number of function evaluations, as shown in Figure 4. Therefore, we decided to set $\beta_k = \hat{\beta}_k$ in the remaining numerical experiments.

4.2.2. Comparison with other globalization strategies

To perform a comparison with other globalization strategies, we ran, on the nonconvex problems, the SDG[Newton,0.5] method and an MN method based on the modified Cholesky factorization GMW-II [15] (see `https://github.com/hrfang/mchol`). For completeness, we also ran Newton's method.

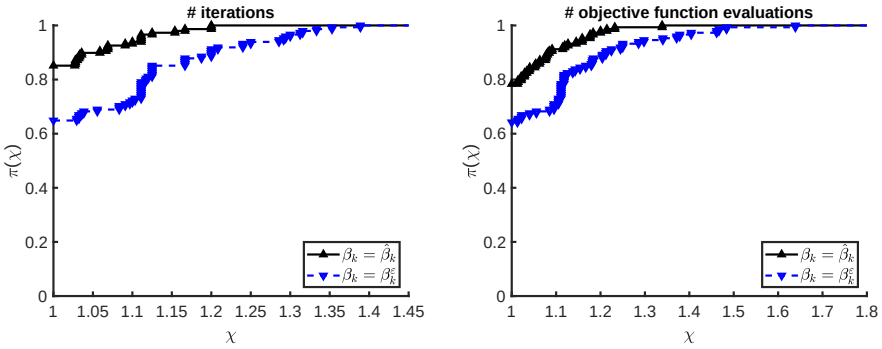

Figure 3: Performance profiles of SDG[BFGS,0.5], with $\beta_k = \beta_k^\varepsilon$ and $\beta_k = \hat{\beta}_k$, on the solution of the 148 nonconvex problem instances in which the two algorithms reached the same solution.

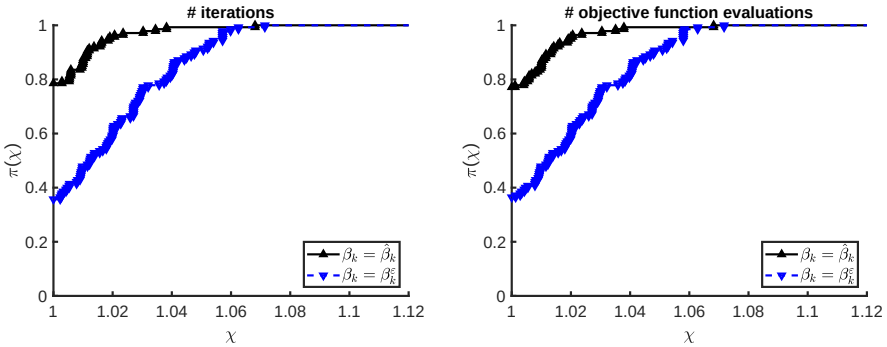

Figure 4: Performance profiles of SDG[BFGS,0.5], with $\beta_k = \beta_k^\varepsilon$ and $\beta_k = \hat{\beta}_k$, on the solution of the 140 convex problem instances.

We found that Newton's method with line search stopped without satisfying criterion (34) for 168 out of 360 problem instances. Conversely, MN failed only on 10 instances, whereas SDG[Newton,0.5] was always able to satisfy (34) within 2000 iterations. Figure 5 summarizes the results of the comparison between SDG[Newton,0.5] and the Modified Newton's method for the 134 problem instances in which the methods obtained the same solution. The profiles show that our algorithm required less function evaluations, although it was slightly less efficient in terms of iterations.

We also compared SDG[BFGS,0.5] with CBFGS using $\upsilon = 1$ (see (15)) and with BFGS. The performance profiles in Figure 6 show how SDG[BFGS,0.5] compares with CBFGS and BFGS in the solution of the 221 problem instances in which the algorithms get the same solution. Note that CBFGS and BFGS overlap extensively. In other words, BFGS does not seem to really need a globalization strategy, and the cautious update rule (15) is likely to reduce to the standard BFGS update rule almost always. Figure 6 also shows that our globalization strategy can slightly improve the performance of the BFGS method.

With the aim of better understanding the behavior of SDG[BFGS,0.5], in

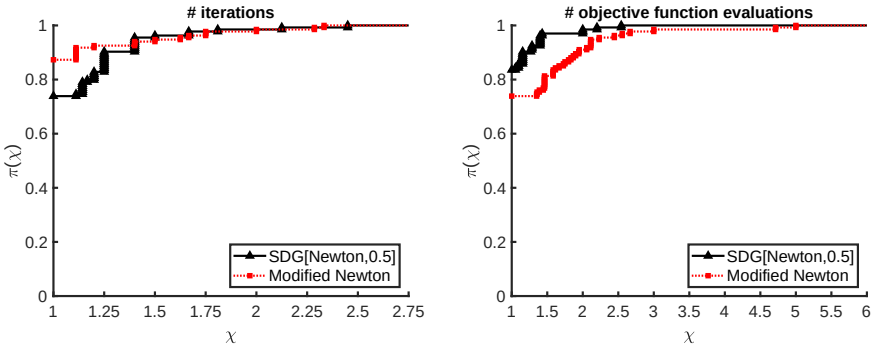

Figure 5: Performance profiles of SDG[Newton,0.5] and the Modified Newton's method on the solution of the 134 nonconvex problem instances in which they computed the same solution.

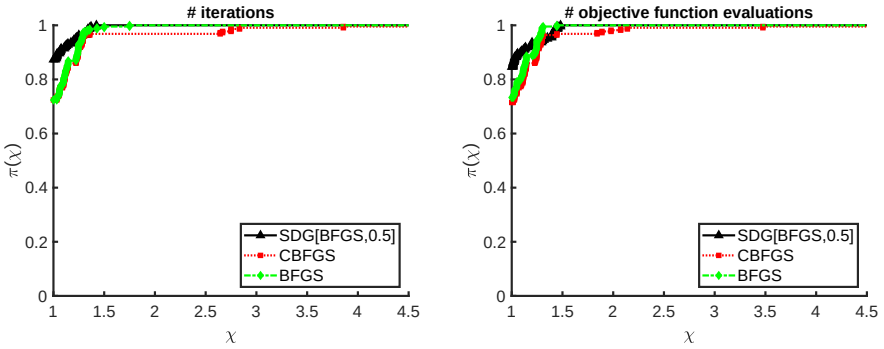

Figure 6: Performance profiles of SDG[BFGS,0.5], CBFGS and BFGS in the solution of the 221 nonconvex problem instances in which the algorithms computed the same solution.

Figure 7 we plotted the sequence $\{\beta_k\}$ for two representative instances of nonconvex problems. We set $\beta_k = 1$ when the BFGS direction was accepted (see lines 7-8 of Algorithm 2), and $\beta_k = 0$ when the SD direction scaled by the BB2 step length was selected (see lines 12-13 of Algorithm 2). In the top plot, concerning a very easy problem (the Goldstein-Price polynomial), we see that the method practically switches from SD to BFGS at the third iteration. The bottom plot, concerning the extended Rosenbrock parabolic valley function, shows that in the very first iterations it is likely that β_k is close to 0, and the SD component is dominating the search direction (16). As the number of iterations increases, the SD component in (16) becomes smaller and smaller, and eventually the method reduces to BFGS.

Concerning the convex problems, we considered BFGS only, i.e., we compared SDG[BFGS,0.5], CBFGS and BFGS. Figure 8 confirms the trend already observed in the nonconvex case: the CBFGS and BFGS methods behave the same way, and SDG[BFGS,0.5] outperforms both of them.

The results in the convex case suggest that a suitable linear combination of an NT direction with the SD one can have a beneficial effect in speeding up the convergence, in addition to providing global convergence. To further investigate

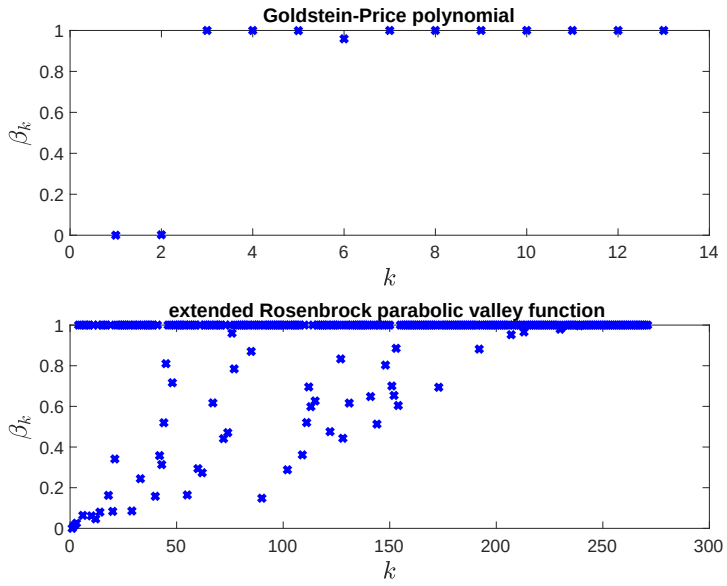

Figure 7: Values of β_k used by SDG[BFGS,0.5] in the solution of two selected nonconvex problems.

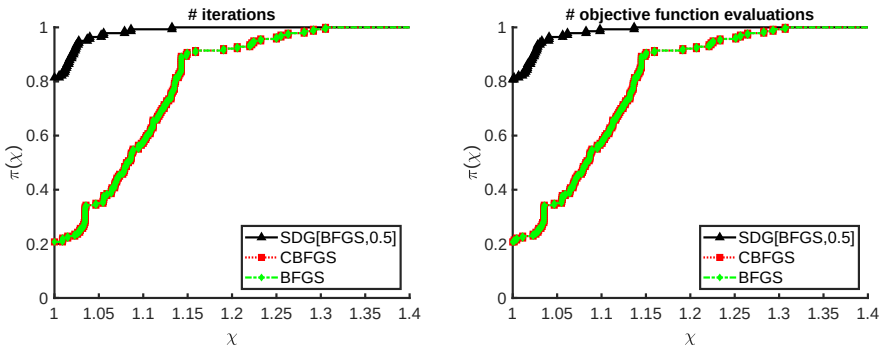

Figure 8: Performance profiles of SDG[BFGS,0.5], CBFGS and BFGS in the solution of the 140 convex problem instances.

this issue, we also made computational experiments with SDG[BFGS,0.9] on the convex test problems. Of course, the choice $\varepsilon_0 = 0.9$ favors the SD component in the search direction, and we cannot suggest it as a safe choice in general. However, the comparison with SDG[BFGS,0.5] in Figure 9 shows that for the selected problems SDG[BFGS, 0.9] is more efficient than SDG[BFGS,0.5],and hence than the standard BFGS.

Finally, the performance profiles in Figure 10 show that SDG[BFGS,0.9] generally outperforms the SD method with BB2 step length. This suggests that the good behavior of SDG[BFGS,0.9] does not depend only on the use of SD directions with effective step lengths, but also on the efficient combination

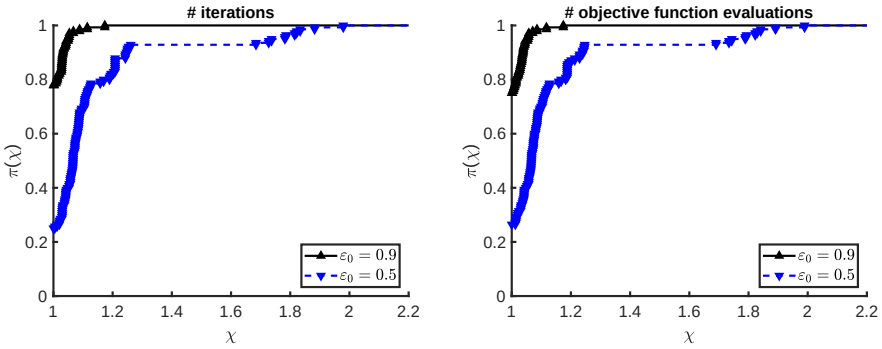

Figure 9: Performance profiles of SDG[BFGS,0.9] and SDG[BFGS,0.5] in the solution of the 140 convex problem instances.

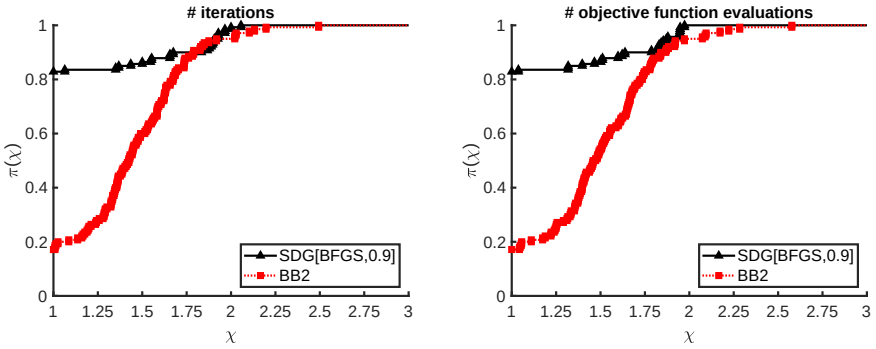

Figure 10: Performance profiles of SDG[BFGS,0.9] and SD with BB2 step length on the solution of the 140 convex problem instances.

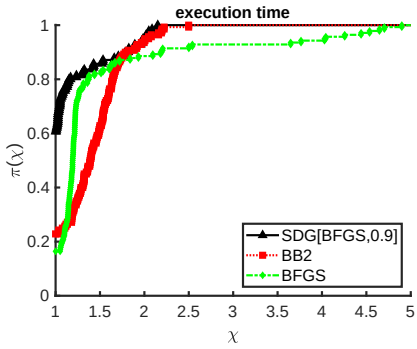

Figure 11: Performance profiles (execution time) of SDG[BFGS,0.9], BFGS and SD with BB2 step length on the solution of the 140 convex problem instances.

of these directions with BFGS ones. This is confirmed by Figure 11, where SDG[BFGS,0.9], BFGS and the SD method with BB2 step length are compared in terms of execution time, showing that the proposed algorithm is more efficient than the others.

5. Conclusions

We proposed a globalization strategy to be used with any NT method, which is based on a linear combination of the NT and SD search directions. Our approach, which generalizes the one proposed in [29, 30], looks easier and more flexible than globalization strategies which have been devised ad hoc for specific methods [15, 19]. We believe that a key issue in our strategy is to take the SD direction with a suitable step length. The reason is twofold: first, from the theoretical point of view, it allows us to have search directions that are invariant to the scaling of the objective function; second, it allows us to inject in the globalized method the proven effectiveness of gradient methods based on particular step-length rules [10]. Our computational experiments suggest that the use of a line search along a suitable linear combination of NT and SD directions can improve numerical performance with respect to the NT method, in addition to providing global convergence. In particular, the SD component with the BB2 step length showed a beneficial effect especially when far from the solution.

Acknowledgments

This work was supported by GNCS-INdAM, Italy, and by the V:ALERE Program of the University of Campania "L. Vanvitelli" under the VAIN-HOPES Project.

References

[1] L. Antonelli, V. De Simone, and D. di Serafino, *On the application of the spectral projected gradient method in image segmentation*, Journal of Mathematical Imaging and Vision, 54 (2016), pp. 106–116.

[2] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 8 (1988), pp. 141–148.

[3] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, second ed., 1999.

[4] R. H. Byrd, J. Nocedal, and Y.-X. Yuan, *Global convergence of a class of Quasi-Newton methods on convex problems*, SIAM Journal on Numerical Analysis, 24 (1987), pp. 1171–1190.

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-region methods*, MPS-SIAM Series on Optimization, SIAM, 2000.

[6] S. Crisci, V. Ruggiero, and L. Zanni, *Steplength selection in gradient projection methods for box-constrained quadratic programs*, Applied Mathematics and Computation, 356 (2019), pp. 312–327.

[7] Y.-H. Dai, *Convergence properties of the BFGS algoritm*, SIAM Journal on Optimization, 13 (2002), pp. 693–701.

[8] R. De Asmundis, D. di Serafino, and G. Landi, *On the regularizing behavior of the SDA and SDC gradient methods in the solution of linear ill-posed problems*, Journal of Computational and Applied Mathematics, 302 (2016), pp. 81–93.

[9] D. di Serafino, G. Landi, and M. Viola, *ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration*, Applied Mathematics and Computation, 364 (2020), p. 124678.

[10] D. di Serafino, V. Ruggiero, G. Toraldo, and L. Zanni, *On the steplength selection in gradient methods for unconstrained optimization*, Applied Mathematics and Computation, 318 (2018), pp. 176–195.

[11] D. di Serafino, G. Toraldo, and M. Viola, *A gradient-based globalization strategy for the Newton method*, in Numerical Computations: Theory and Algorithms. NUMTA 2019, Y. D. Sergeyev and D. E. Kvasov, eds., vol. 11973 of Lecture Notes in Computer Science, Springer, 2020, pp. 177–185.

[12] D. di Serafino, G. Toraldo, M. Viola, and J. Barlow, *A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables*, SIAM Journal on Optimization, 28 (2018), pp. 2809–2838.

[13] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, Series B, 91 (2002), pp. 201–213.

[14] Z. Dostál, G. Toraldo, M. Viola, and O. Vlach, *Proportionality-based gradient methods with applications in contact mechanics*, in High Performance Computing in Science and Engineering. HPCSE 2017, T. Kozubek, M. Čermák, P. Tichý, R. Blaheta, J. Šístek, D. Lukáš, and J. Jaroš, eds., vol. 11087 of Lecture Notes in Computer Science, Springer, 2018, pp. 47–58.

[15] H. Fang and D. P. O'Leary, *Modified Cholesky algorithms: a catalog with new approaches*, Mathematical Programming, 115 (2008), pp. 319–349.

[16] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, second ed., 2000.

[17] L. Han and M. Neumann, *Combining quasi-Newton and Cauchy directions*, International Journal of Applied Mathematics, 12 (2003), pp. 167–191.

[18] D.-H. Li and M. Fukushima, *A modified BFGS method and its global convergence in nonconvex minimization*, Journal of Computational and Applied Mathematics, 129 (2001), pp. 15–35.

[19] ——, *On the global convergence of the BFGS method for nonconvex unconstrained optimization problems*, SIAM Journal on Optimization, 11 (2001), pp. 1054–1064.

[20] W. F. Mascarenhas, *The BFGS method with exact line searches fails for non-convex objective functions*, Mathematical Programming, Series B, 99 (2004), pp. 49–61.

[21] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software, 7 (1981), pp. 17–41.

[22] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Numerical Analysis, G. Golub, ed., The Mathematical Association of America, Providence, RI, 1984, pp. 29–82.

[23] W. Murray, *Newton-type methods*, in Wiley Encyclopedia of Operations Research and Management Science, Wiley, 2011.

[24] Y. Nesterov, *Introductory lectures on convex optimization. A basic course*, vol. 87 of Applied Optimization, Springer Science+Business Media, 2004.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, second ed., 2006.

[26] E. Polak, *Optimization. Algorithms and consistent approximations*, vol. 124 of Applied Mathematical Sciences, Springer, 1997.

[27] F. Porta, M. Prato, and L. Zanni, *A new steplength selection for scaled gradient methods with application to image deblurring*, Journal of Scientific Computing, 65 (2015), pp. 895–919.

[28] L. Pospíšil and Z. Dostál, *The projected Barzilai–Borwein method with fall-back for strictly convex QCQP problems with separable constraints*, Mathematics and Computers in Simulation, 145 (2018), pp. 79–89.

[29] Y. Shi, *A globalization procedure for solving nonlinear systems of equations*, Numerical Algorithms, 12 (1996), pp. 273–286.

[30] ——, *Globally convergent algorithms for unconstrained optimization*, Computational Optimization and Applications, 16 (2000), pp. 295–308.

[31] R. Zanella, P. Boccacci, L. Zanni, and M. Bertero, *Efficient gradient projection methods for edge-preserving removal of Poisson noise*, Inverse Problems, 25 (2009), p. 045010.