Nonconvex optimization using negative curvature within a modified linesearch

Alberto Olivares [a,*], Javier M. Moguerza [a], Francisco J. Prieto [b]

[a] *School of Engineering, University Rey Juan Carlos, C/Tulipan s/n, 28933 Mostoles, Madrid, Spain*
[b] *Department of Statistics, University Carlos III, C/Madrid 126, 28903 Getafe, Madrid, Spain*

Abstract

This paper describes a new algorithm for the solution of nonconvex unconstrained optimization problems, with the property of converging to points satisfying second order necessary optimality conditions. The algorithm is based on a procedure which, from two descent directions, a Newton type direction and a direction of negative curvature, selects in each iteration the linesearch model best adapted to the properties of these directions. The paper also presents results of numerical experiments that illustrate its practical efficiency.

Keywords: Newton's method; Unconstrained optimization; Negative curvature

1. Introduction

The goal of this work is the efficient solution of optimization problems having the form

$$\min_x f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function.

Many algorithms have been proposed in the literature to solve such problems, but only a few attempt to use second-order information available in the Hessian matrix of f; this information may play a very significant role in the design of efficient algorithms. Unconstrained optimization problems are usually solved by applying algorithms based on Newton's method. These methods, when properly implemented, have well-known convergence properties and, in particular, they can be shown to be globally convergent to first-order critical points. There are two broad classes of procedures to ensure these convergence properties: linesearch methods and trust-region methods. The method described in this paper belongs to the class of linesearch procedures;

[*] Corresponding author. Tel.: +34 91 488 8125; fax: +34 91 488 7626.
E mail addresses: alberto.olivares@urjc.es (A. Olivares), javier.moguerza@urjc.es (J.M. Moguerza), franciscojavier.prieto@uc3m.es (F.J. Prieto).

in particular, we describe a linesearch that uses second-order information in an efficient manner. This information is introduced through the computation of a negative curvature direction in each iteration: along these negative curvature directions the quadratic model is unbounded below and, in some cases, this property offers the potential for a larger reduction in the value of the objective function.

We shall introduce a methodology for the linesearch that exploits any nonconvexity that the objective function may present locally. To that end, in each iteration we compute a pair of directions (s_k, d_k). The first one, s_k, is a modified Newton direction that ensures fast convergence under convexity assumptions. The second one, d_k, is a negative curvature direction that allows the algorithm to move in an efficient manner away from local nonconvex regions. The approach proposed in this paper will be applied to the small dimensional setting, although the underlying ideas can also be adapted to large dimensional problems with limited modifications.

One of the first proposals to take into account second-order information in order to ensure convergence towards second-order critical points is that of Fiacco and McCormick [8]. More recently, the use of this information has been studied by Fletcher and Freeman [9], Gill and Murray [11], or Mukai and Polak [22], among others. In a linesearch context the work of McCormick [18] is particularly relevant. In this work, Armijo's rule [1] for the termination of the linesearch is adapted and modified to include negative curvature information. Moré and Sorensen [21] and Goldfarb [13] follow a similar approach, their work being the first one to use explicitly negative curvature directions in the solution of specific instances of unconstrained problems. More recently, Moguerza and Prieto [19] have extended the methodology of Moré and Sorensen to constrained problems within an interior point framework. Both in Moré and Sorensen [21] and Moguerza and Prieto [19], the next iterate is obtained through a backtracking procedure along a second-order curve combining the directions. In Moguerza and Prieto [20], the search is conducted on a curve obtained from the approximate solution of an ordinary differential equation related to the problem. Other approaches to ensure convergence to second-order critical points for the constrained case are described in Auslender [2], Gay [10], Bannert [3] and Bonnans [4].

Although out of the scope of this paper, there are linesearch procedures based on the conjugate gradient methodology for unconstrained optimization which are useful for solving large scale problems. These methods have well-known convergence properties (see [16], and references therein), and have been used in some practical engineering applications (see, for instance [25]). Other works also use directions of negative curvature within conjugate gradient schemes (see [23] and references therein). For instance, in Gould et al. [14], in each iteration the best direction is chosen and a standard linesearch is conducted. Another method based in the selection of directions is suggested by Sanmatías and Roma [24]. The relation between our proposal and Sanmatías and Roma [24] will be discussed below.

The main aim of this work is to identify a criterion such that in each iteration the best search procedure is chosen. Depending on the results from the application of this criterion, the algorithm will apply either a curvilinear search similar to that proposed by Moré and Sorensen [21], combining both search directions, or a standard linesearch similar to that proposed by Gould et al. [14], using just one of the two directions.

The structure of the paper is as follows: In Section 2, we introduce some basic definitions as well as a method to compute the descent directions. Section 3 describes the procedure to select the steplength and discusses its convergence properties. Section 4 introduces a condition to identify the search method to be used in a given iteration. In Section 5, we discuss implementation details and present a general scheme for the algorithm. Section 6 shows the results from the computational experiments. Finally, Section 7 presents some conclusions.

2. Basic definitions and search direction computations

For the remainder of the paper we will assume that the following regularity properties hold for problem (1) and the initial point of the algorithm, x_0.

A1. The level set $\mathscr{L}(x_0) = \{x : f(x) \leqslant f(x_0)\}$ is compact.

A2. The objective function f has second-order derivatives that are Lipschitz-continuous on an open set that contains $\mathscr{L}(x_0)$.

Assuming that x is an indefinite point (see [21], for a precise definition), a pair of vectors (s, d) is said to be a descent pair if $g(x)^{\mathrm{T}}s \leqslant 0$, $g(x)^{\mathrm{T}}d \leqslant 0$ and $d^{\mathrm{T}}H(x)d < 0$, where $g(x) \equiv \nabla f(x)$ and $H(x) \equiv \nabla^2 f(x)$. A possible choice for a descent pair would be $s = -g(x)$ and $d = \pm u_n$, where u_{n} denotes an eigenvector associated to a negative eigenvalue of $H(x)$. The sign for u_n is chosen to ensure that $g(x)^{\mathrm{T}}d \leqslant 0$. If x is not indefinite and $g(x) \neq 0$, then the second-order information is not relevant (and $d = 0$). From a more practical point of view, in a given iterate x_k a direction s_k is assumed to be a sufficient descent direction if there exist constants c_1 and c_2, independent of k, such that

$$s_k^{\mathrm{T}}g(x_k) \leqslant -c_1\|g(x_k)\|^2 \tag{2}$$

and

$$\|s_k\| \leqslant c_2\|g(x_k)\|. \tag{3}$$

Analogously, we will consider that d_k is a sufficient direction of negative curvature at x_k if there exists a constant c_3 independent of k such that

$$\begin{aligned} &d_k^{\mathrm{T}}g(x_k) \leqslant 0, \\ &d_k^{\mathrm{T}}H(x_k)d_k \leqslant \min(0, \theta\lambda_{H_k}), \\ &\|d_k\| \leqslant c_3. \end{aligned} \tag{4}$$

where $\theta \in (0, 1)$ is independent of k and λ_{H_k} denotes the smallest eigenvalue of the Hessian matrix $H(x_k)$ if x_k is indefinite, and zero otherwise. These conditions are similar to those introduced by Lucidi et al. [17].

Conditions (2) and (3) are the standard ones for Newton-type directions. Condition (4) ensures that d_k contains information related to the smallest eigenvalue of the Hessian matrix.

2.1. Computing a modified Newton direction

Under assumptions A1 and A2 it is possible to construct a local quadratic model for the objective function from the corresponding Taylor series expansion at all iterates $x_k \in \mathscr{L}(x_0)$ as

$$f(x_k + p) \simeq f(x_k) + g(x_k)^{\mathrm{T}}p + \frac{1}{2}p^{\mathrm{T}}H(x_k)p. \tag{5}$$

We introduce the notation

$$\Phi_k(s) \equiv g(x_k)^{\mathrm{T}}s + \frac{1}{2}s^{\mathrm{T}}H(x_k)s \tag{6}$$

for the right-hand side of (5), omitting the constant term $f(x_k)$. As shown, for example, in Gill et al. [12], s_k is a stationary point of Φ_k if it satisfies the linear system of equations

$$H(x_k)s_k = -g(x_k). \tag{7}$$

The direction s_k, obtained as a solution of (7), is known as the Newton direction. An efficient method to compute this direction uses the modified Cholesky factorization proposed by Gill and Murray [11].

2.2. Computing a direction of negative curvature

For an iterate $x_k \in \mathscr{L}(x_0)$, a direction of negative curvature d_k will be useful if it satisfies condition (4). Note that from (4), it follows that

$$d_k^{\mathrm{T}}H(x_k)d_k \to 0 \Rightarrow \min(0, \lambda_{H_k}) \to 0. \tag{8}$$

The motivation underlying (4) is to ensure that the iterates move towards regions of local convexity for f.

If the complete spectral decomposition of the Hessian matrix $H(x_k)$ were available, it would be easy to obtain directions d_k satisfying condition (4). Unfortunately, the determination of a complete system of eigenvectors and eigenvalues for this matrix can be computationally very expensive.

Alternatively, the modified Cholesky factorization of Gill and Murray [11] allows the computation of a negative curvature direction simultaneously with the determination of a modified Newton direction; in this manner, the computational cost to obtain a descent pair is reasonable. In Moré and Sorensen [21] it is shown that this factorization may either provide unbounded negative curvature directions or arbitrarily small negative curvature, compared to that available in the quadratic model at the iteration. We have chosen to impose the condition $\|d\| \leqslant c_3$ in (4). An alternative factorization which can be used, while avoiding this problem, is the one proposed by Bunch and Parlett [6]. In computational experiments using small-size problems we have obtained similar results for both factorizations.

3. The curvilinear search

In this section, we present some convergence results for a search based on the use of descent pairs, as defined in Section 2. Many of these results are adapted from those in Moré and Sorensen [21].

When using a search method to ensure global convergence, in each iteration a parametric curve $x(\alpha)$ is built from the directions computed in that iteration. The goal is to find a value of the parameter $\alpha > 0$ such that

$$f(x(\alpha)) < f(x). \tag{9}$$

In our case, in each iteration k we compute a descent pair (s_k, d_k), and, from it, we define a general parametric curve as

$$\mathscr{C} = \{x_k(\alpha) \equiv x_k + \phi_1(\alpha)s_k + \phi_2(\alpha)d_k,\ \alpha \geqslant 0\}, \tag{10}$$

where $\phi_1(\alpha)$ and $\phi_2(\alpha)$ are appropriate (nonnegative) weight functions for the linear combination. We build the curve using both descent directions, in order to use simultaneously the information available in them.

We now analyze the properties of this search and reasonable choices for the functions ϕ_1 and ϕ_2. We start by studying the problem of finding an appropriate value for α, ensuring the convergence of the algorithm, from the analysis of the univariate function

$$\psi_k(\alpha) \equiv f(x_k(\alpha)). \tag{11}$$

The following Lemma shows that there exists a value of $\alpha > 0$ such that a condition related to (9) and ensuring sufficient descent is satisfied, as long as either $\psi_k'(0) < 0$ or $\psi_k'(0) \leqslant 0$ and $\psi_k''(0) < 0$ hold. The Lemma extends the results in Moré and Sorensen [21], as it considers general functions ϕ_1 and ϕ_2 in (10) with the only assumption that they have Lipschitz-continuous second derivatives.

Lemma 1. *Let $\psi(\alpha) : \mathbb{R} \to \mathbb{R}$, defined as in (11), be a function with Lipschitz-continuous second derivatives on an open interval I containing $[0, 1]$, and $\gamma_1 \in (0, 1)$ be a given scalar. If either $\psi'(0) < 0$ or $\psi'(0) \leqslant 0$ and $\psi''(0) < 0$ then there exists a scalar $\hat{\eta} > 0$ in I such that*

$$\psi(\alpha) \leqslant \psi(0) + \gamma_1\left(\psi'(0)\alpha + \frac{1}{2}\psi''(0)\alpha^2\right)$$

holds for all $\alpha \in [0, \hat{\eta}]$, where $\hat{\eta}$ is strictly positive.

Furthermore, let γ_2 be another given scalar such that $1 > \gamma_2 > \gamma_1 > 0$. Either $\psi(1) \leqslant \psi(0)+\gamma_1(\psi'(0) + \frac{1}{2}\psi'(0))$ holds or there exists a scalar $\eta > 0$ satisfying $\eta \leqslant \hat{\eta}$, such that

$$\psi(\alpha) \leqslant \psi(0) + \gamma_1\left(\psi'(0)\alpha + \frac{1}{2}\psi''(0)\alpha^2\right), \tag{12}$$

$$\psi'(\alpha) \geqslant \gamma_2 \min\left(\psi'(0) + \psi''(0)\alpha, 0\right) \tag{13}$$

hold simultaneously for all $\alpha \in [\eta, \hat{\eta}]$, and η is strictly positive.

Proof. Let

$$\Psi(\alpha) \equiv \psi(\alpha) - \psi(0) - \gamma_1\left(\alpha\psi'(0) + \frac{1}{2}\alpha^2\psi''(0)\right).$$

Note that, from the properties of ψ and this definition, $\Psi(0) = 0$, $\Psi'(0) = (1-\gamma_1)\psi'(0) \leqslant 0$ and $\Psi''(0) = (1-\gamma_1)\psi''(0) < 0$ if $\Psi'(0) = 0$. Also, Ψ has Lipschitz-continuous second derivatives on I.

(a) Consider the case when $\psi'(0) < 0$. From the Taylor series expansion for Ψ around zero and $\alpha \in I$ we have

$$\Psi(\alpha) = \Psi'(0)\alpha + \frac{1}{2}\Psi''(\zeta_1)\alpha^2$$

for some $\zeta_1 \in [0,\alpha]$. Let $\bar{I}$ denote a closed subinterval of I that contains zero. Then ψ and its first and second derivatives are bounded on $\bar{I}$. Let K_1 be a bound for $\psi''(\alpha)$ on $\bar{I}$ and $\beta_1 = -2\psi'(0)/K_1$. For all $\alpha \in (0,\beta_1)$, it holds that $\Psi(\alpha) < 0$ and, as a consequence of the definition of $\hat{\eta}$, it also holds that $\hat{\eta} \geqslant -2\psi'(0)/K_1$.

If $\psi'(0) < 0$ but $\psi(1) \leqslant \psi(0) + \gamma_1(\psi'(0) + \frac{1}{2}\psi''(0))$ does not hold, we have that $\Psi(1) > 0$ and, by continuity, there exists a positive value $\beta_2 < 1$ such that $\Psi(\beta_2) = 0$ and β_2 is the first such value. From the mean-value theorem, there exists a positive value $\beta_3 < \beta_2$, the smallest positive zero of Ψ' so that $\Psi'(\beta_3) = 0$. Note that (12) holds for all $\alpha \in [0,\beta_2]$.

We study two cases: (i) If $\psi'(0) + \beta_3\psi''(0) \leqslant 0$, then from $\gamma_2 > \gamma_1$ it holds that

$$\begin{aligned} 0 &= \psi'(\beta_3) - \gamma_1(\psi'(0) + \beta_3\psi''(0)) \\ &\leqslant \psi'(\beta_3) - \gamma_2(\psi'(0) + \beta_3\psi''(0)) \\ &= \psi'(\beta_3) - \gamma_2 \min(\psi'(0) + \beta_3\psi''(0), 0) \end{aligned}$$

implying that both (12) and (13) hold for $\alpha = \beta_3$. (ii) If $\psi'(0) + \beta_3\psi''(0) > 0$, from $\Psi'(\beta_3) = 0$ it holds that

$$\psi'(\beta_3) = \gamma_2(\psi'(0) + \beta_3\psi''(0)) > \min(\psi'(0) + \beta_3\psi''(0), 0) = 0$$

and (13) also holds for $\alpha = \beta_3$.

From $\Psi'(\beta_3) = 0$ and Taylor series expansions, it follows that

$$\gamma_1(\psi'(0) + \beta_3\psi''(0)) = \psi'(\beta_3) = \psi'(0) + \beta_3\psi''(\zeta_3) \tag{14}$$

for some $\zeta_3 \in [0,\beta_3]$. From $\beta_3 < 1$ and our assumptions, there exists a value K_3 such that $|\psi''(\alpha)| \leqslant K_3$ for any $\alpha \in [0,\beta_3]$. Any α that satisfies (13) will also satisfy $\alpha \geqslant \beta_3$ and, from (14),

$$\alpha \geqslant \beta_3 \geqslant -\frac{(1-\gamma_1)\psi'(0)}{2K_3} > 0. \tag{15}$$

(b) Consider now the case when $\psi'(0) \leqslant 0$ and $\psi''(0) < 0$. Using again Taylor series expansions for $\alpha \in [0,1]$,

$$\Psi(\alpha) = \Psi'(0)\alpha + \frac{1}{2}\Psi''(\zeta_2)\alpha^2 = \Psi'(0)\alpha + \frac{1}{2}\Psi''(0)\alpha^2 + \frac{1}{2}(\Psi''(\zeta_2) - \Psi''(0))\alpha^2$$

for some $\zeta_2 \in [0,\alpha]$. Let K_2 denote the Lipschitz constant for $\Psi''(\alpha)$ on I, then $|\Psi''(\zeta_2) - \Psi''(0)| \leqslant K_2\alpha$,

$$\Psi(\alpha) \leqslant \Psi'(0)\alpha + \frac{1}{2}\Psi''(0)\alpha^2 + \frac{1}{2}K_2\alpha^3.$$

Defining $\beta_4 = -(1-\gamma_1)\psi''(0)/K_2$, for any $\alpha \in (0,\beta_4)$ it holds that $\frac{1}{2}\Psi''(0)\alpha^2 + \frac{1}{2}K_2\alpha^3 < 0$. As $\Psi'(0) \leqslant 0$, (12) holds for all $\alpha \in (0,\beta_4)$ and from the definition of $\hat{\eta}$ we have that

$$\hat{\eta} \geqslant -(1-\gamma_1)\psi''(0)/K_2. \tag{16}$$

If, in addition, $\psi(1) > \psi(0) + \gamma_1(\psi'(0) + \frac{1}{2}\psi''(0))$ holds, we apply an argument similar to the preceding case. As $\Psi(0) = 0$, $\Psi'(0) \leqslant 0$, $\Psi''(0) < 0$ and $\Psi(1) > 0$, by continuity there exists a positive value $\beta_5 < 1$ such that $\Psi(\beta_5) = 0$ and β_5 is the first such value. From the mean-value theorem, there exists a positive value $\beta_6 < \beta_5$, the smallest positive zero of Ψ', so that $\Psi'(\beta_6) = 0$. Note that (12) holds for all $\alpha \in [0,\beta_5]$.

However, under the conditions on $\psi'(0)$ and $\psi''(0)$, we have $\psi'(0) + \beta_6\psi''(0) < 0$, and from $\gamma_2 > \gamma_1$ it holds that

$$\begin{aligned} 0 &= \psi'(\beta_6) - \gamma_1(\psi'(0) + \beta_6\psi''(0)) \\ &\leqslant \psi'(\beta_6) - \gamma_2(\psi'(0) + \beta_6\psi''(0)) \\ &= \psi'(\beta_6) - \gamma_2 \min(\psi'(0) + \beta_6\psi''(0), 0) \end{aligned}$$

implying again that both (12) and (13) hold for $\alpha = \beta_6$.

Again, from $\Psi'(\beta_6) = 0$ and Taylor series expansions,

$$\gamma_1(\psi'(0) + \beta_6\psi''(0)) = \psi'(\beta_6) = \psi'(0) + \beta_6\psi''(0) + \beta_6(\psi''(\zeta_4) - \psi''(0))$$

for some $\zeta_4 \in [0, \beta_6]$. Note that $\beta_6 < 1$, and let K_4 denote the Lipschitz constant for $\psi''(\alpha)$ on $[0, \beta_6]$. Using $\psi'(0) \leqslant 0$ it follows that

$$-(1-\gamma_1)\beta_6\psi''(0) \leqslant -(1-\gamma_1)(\psi'(0) + \beta_6\psi''(0)) = \beta_6(\psi''(\zeta_4) - \psi''(0)) \leqslant K_4\beta_6^2. \tag{17}$$

Any α that satisfies (13) will also satisfy $\alpha \geqslant \beta_6$ and from (17)

$$\alpha \geqslant \beta_6 \geqslant -\frac{(1-\gamma_1)\psi''(0)}{K_4} > 0. \qquad \square \tag{18}$$

The following Corollary assumes that a backtracking search to compute α is conducted as described in Gill et al. [12], that is, for a given positive constant $\delta < 1$ we define the steplength α as the first value in the sequence $\{\delta^i\}_{i\;0}^{\infty}$ for which condition (12) is satisfied.

We also consider the case when a general search is used. We assume that this search computes a value for the steplength α_k satisfying both (12) and (13), with initial steplength equal to one.

Corollary 1. *If a backtracking search with parameter δ is used, the value of α computed from the search, $\hat{\alpha}$, satisfies*

$$\hat{\alpha} \geqslant \delta\hat{\eta}$$

and is bounded away from zero. If a general search is conducted to compute a value satisfying both (12) *and* (13), *then the computed value α satisfies*

$$\alpha \geqslant \eta.$$

Proof. This result follows for the backtracking search from $\alpha_k^{i+1} = \delta\alpha_k^i$ and α_k^i not satisfying (12) and, consequently, satisfying $\alpha_k^i > \hat{\eta}$, and the bound for $\hat{\eta}$. For the general search, the result follows directly from the bounds for η. $\square$

The preceding results establish the existence of sufficient descent given certain conditions on the derivatives of ψ. The satisfaction of these conditions depends on the properties of the functions ϕ_1 and ϕ_2, and those of the descent pair (s, d). We now analyze our requirements on these functions. From the definition of $\psi(\alpha)$ it follows that

$$\begin{aligned} \psi'(0) &= \phi_1'(0)g(x)^{\mathrm{T}}s + \phi_2'(0)g(x)^{\mathrm{T}}d, \\ \psi''(0) &= (\phi_1'(0))^2 s^{\mathrm{T}}H(x)s + (\phi_2'(0))^2 d^{\mathrm{T}}H(x)d \\ &\quad + 2\phi_1'(0)\phi_2'(0)s^{\mathrm{T}}H(x)d + \phi_1''(0)g(x)^{\mathrm{T}}s \\ &\quad + \phi_2''(0)g(x)^{\mathrm{T}}d \end{aligned}$$

and, from these equalities and the properties of the descent pair (s, d), we have

$$\begin{aligned} &\phi_1'(0) \geqslant 0, \phi_2'(0) \geqslant 0 \Rightarrow \psi'(0) \leqslant 0, \\ &\phi_1'(0) = 0,\ \phi_1''(0) \geqslant 0,\ \phi_2''(0) \geqslant 0 \Rightarrow \psi''(0) \leqslant 0. \end{aligned}$$

If, in addition to these conditions, $\phi_2'(0) > 0$ and $\phi_1''(0) > 0$, then $\psi''(0) < 0$. We also need the second derivatives to be Lipschitz-continuous on $[0, 1]$.

The preceding inequalities thus provide necessary conditions on ϕ_1 and ϕ_2 to ensure that either $\psi'(0) < 0$ or $\psi'(0) \leqslant 0$ and $\psi''(0) < 0$ hold, as required in Lemma 1. We assume in what follows that the functions ϕ_1 and ϕ_2 satisfy

C1. It holds that

$$\phi_1'(0) = 0, \quad \phi_2'(0) > 0, \quad \phi_1''(0) > 0, \quad \phi_2''(0) \geqslant 0.$$

Note that, under *C1*, for any descent pair $(s, d) \neq 0$ we have

$$\begin{aligned} &\psi'(0) = \phi_2'(0) g(x)^{\mathrm{T}} d \leqslant 0, \\ &\psi''(0) = (\phi_2'(0))^2 d^{\mathrm{T}} H(x) d + \phi_1''(0) g(x)^{\mathrm{T}} s + \phi_2''(0) g(x)^{\mathrm{T}} d < 0. \end{aligned} \tag{19}$$

Also, given x_k and a descent pair (s_k, d_k), we assume that the next iterate $x_{\mathrm{k+1}}$ is defined as

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

where α is chosen to satisfy the following condition:

C2. The steplength α_k is defined to be equal to one if $\psi(1)$ satisfies (12); otherwise, α_k is computed to satisfy both (12) and (13).

This standard condition ensures that a steplength of one is taken whenever it provides sufficient descent. Finally, we need a condition on the descent pair (s_k, d_k) to ensure that it provides either sufficient descent or sufficient negative curvature compared to that available in the current iterate x_k. We introduce the following sufficient condition to replace (2) (4):

C3. Let $\bar{c}_1$, $\bar{c}_2$, $\bar{c}_3$ and $\bar{\theta}$ be nonnegative constants. For all iterations k, the descent pair (s_k, d_k) satisfies the conditions

$$\begin{aligned} &\|s_k\| \leqslant \bar{c}_1 \|g(x_k)\|, \quad g(x_k)^{\mathrm{T}} s_k \leqslant 0, \\ &g(x_k)^{\mathrm{T}} d_k \leqslant 0, \quad \|d_k\| \leqslant \bar{c}_2 \end{aligned}$$

and defining

$$w_k = \begin{cases} s_k^{\mathrm{T}} g(x_k)/\|g(x_k)\| & \text{if } g(x_k) \neq 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$\min(w_k, d_k^{\mathrm{T}} H(x_k) d_k) \leqslant \min(-\bar{c}_3 \|g(x_k)\|, \bar{\theta} \lambda_{H_k}). \tag{20}$$

This condition ensures that either the (scaled) descent or the negative curvature available in the descent pair for a given iteration are sufficient, when compared with the largest descent and negative curvature available in the quadratic model for that iteration. Instead of testing both conditions (descent and negative curvature) separately, we have combined them into one requirement, defined to be as general as possible but also satisfied by the choice of directions used in the proposed algorithm, as shown in the following section.

The following result shows that, under these conditions and our initial assumptions, the procedure is globally convergent.

Theorem 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ in (1) satisfy assumptions A1 and A2. Consider the sequence $\{x_k\}$ generated under condition C2 using a descent pair (s_k, d_k) satisfying condition C3. Then,*

$$\lim_{k \to \infty} g(x_k) = 0 \tag{21}$$

and if the first-order critical points of f in $\mathscr{L}(x_0)$ satisfy the sufficient second-order condition,

$$\lim_{k \to \infty} \lambda_{H_k} = 0. \tag{22}$$

Proof. From C2 and (12), all iterates remain in $\mathscr{L}(x_0)$. As a consequence of assumptions A1 and A2, the sequence $\{f(x_k)\}$ is bounded below, and the sequence $\{x_k\}$ has convergent subsequences.

Consider any of these convergent subsequences, and add condition (12) for all iterations along the sequence up to a given iteration r in the subsequence, to obtain

$$f(x_r) - f(x_0) \leqslant \gamma_1 \sum_{k\ 0}^{r-1} \left(\psi_k'(0)\alpha_k + \frac{1}{2}\psi_k''(0)\alpha_k^2 \right).$$

Taking limits as $r \to \infty$, using the boundedness of f and the signs of α, ψ' and ψ'' it follows that

$$\psi_k'(0)\alpha_k \to 0 \quad \text{and} \quad \psi_k''(0)\alpha_k^2 \to 0.$$

From condition C3, if $s_k = d_k = 0$ for some iteration k, then $g(x_k) = 0$ and $\lambda_{H(x)} = 0$, implying that x_k satisfies the second-order necessary conditions. If the descent pair is not equal to zero in any iteration, from C1 we have $\psi_k'(0) \leqslant 0$ and $\psi_k''(0) < 0$ for all k. For these iterations, from Corollary 1, (16) and $\psi_k''(0) < 0$, if we conduct a backtracking, search we have that

$$\psi_k''(0)\alpha_k^2 \leqslant \frac{\delta^2(1-\gamma_1)^2(\psi_k''(0))^3}{K_2^2} \leqslant 0$$

and, if we conduct a general search, from (18),

$$\psi_k''(0)\alpha_k^2 \leqslant \frac{(1-\gamma_1)^2(\psi_k''(0))^3}{K_4^2} \leqslant 0$$

and if $\mathscr{K}_2$ is an infinite subsequence, from $\psi_k''(0)\alpha_k^2 \to 0$ and the preceding bounds it must hold along it that $\psi_k''(0) \to 0$. But from (19), this implies

$$d_k^{\mathrm{T}} H(x_k) d_k \to 0 \quad \text{and} \quad g(x_k)^{\mathrm{T}} s_k \to 0.$$

If $g(x_k) = 0$ along an infinite subsequence, from C3 we have $\lambda_{H_k} \to 0$. Otherwise, assume $g(x_k) \neq 0$ for all k in the subsequence; we may have that either $g(x_k)^{\mathrm{T}} s_k / \|g(x_k)\| \to 0$ along all subsequences, in which case condition C3 implies $\min(-\bar{c}_3\|g(x_k)\|^2, \bar{\theta}\lambda_{H_k}) \to 0$ and

$$\|g(x_k)\| \to 0, \quad \lambda_{H_k} \to 0,$$

or along a subsequence there exists a positive constant K such that, for all k, in the subsequence $g(x_k)^{\mathrm{T}} s_k \leqslant -K\|g(x_k)\| \leqslant 0$, but this implies $\|g(x_k)\| \to 0$. From $\|s_k\| \leqslant \bar{c}_1\|g(x_k)\|$ we have that $0 \geqslant g(x_k)^{\mathrm{T}} s_k / \|g(x_k)\| \geqslant -\|s_k\| \geqslant -\bar{c}_1\|g(x_k)\|$ and we must also have $g(x_k)^{\mathrm{T}} s_k / \|g(x_k)\| \to 0$, contradicting our assumption. Thus, in all cases we have

$$\|g(x_k)\| \to 0, \quad \lambda_{H_k} \to 0$$

and the desired result follows from these limits and assumption A2. □

The preceding convergence results provide a justification for the algorithm of interest in this paper but, in order to have a practical algorithm, we still need to define in a precise manner both the form of the functions ϕ_1 and ϕ_2 and that of the descent pair (s_k, d_k). Regarding the functions, the simplest choice that satisfies the desired conditions C1 is $\phi_1(\alpha) \equiv \alpha^2$ and $\phi_2(\alpha) = \alpha$. We will use it in the remainder of the paper. The choice of the directions in the descent pair will be discussed in the following section.

4. The modified linesearch

As we mentioned before, in each iteration the proposed algorithm computes a descent pair (s_k, d_k), but before carrying out a search to satisfy C2, it decides what information in the descent pair will be used in the search. This approach is similar to the one discussed in Gould et al. [14], with the main difference that, while they only consider two alternatives, we consider the three following possibilities:

- To conduct a linesearch based just on the use of the modified Newton direction, using the descent pair $(s_k, 0)$.
- To conduct a linesearch based only on the negative curvature direction, using the descent pair $(0, d_k)$.
- To carry out a curvilinear search combining both directions, making use of the full descent pair (s_k, d_k).

Our choice will be made from a comparison of the descent provided by each direction on a quadratic model of the objective function. We wish to select the alternative that offers the most significant descent information. A similar proposal to this scheme was suggested in San Matías and Roma [24], where the proposed method would select one of these three different possibilities from a comparison of the descent provided on the quadratic model of the objective function. The main differences between this work and the proposal in San Matías and Roma [24] are the way in which the weights for the directions are chosen and the scaling of the direction of negative curvature.

As previously mentioned, before analyzing the descent information, it is important to consider the scaling of each of the directions. The manner in which they are computed and their properties usually imply a significant difference in their sizes. The Newton direction s_k ensures good local convergence properties for unit steplengths, while the negative curvature direction has no specific scale attached to it (the quadratic model used to define it would imply an infinite steplength). In this way, our implementation handles this problem by adjusting the scale of the direction of negative curvature so that its norm equals that of the Newton direction, that is, $\|d_k\| = \|s_k\|$. Therefore, the scaling of the negative curvature direction will be adequately related to the Newton direction (see, [19]). Some other proposals regarding this task can be consulted in Byrd et al. [7] and Moguerza and Prieto [20].

Consider now the choice of search directions. Let $\mathcal{Q}_k$ denote the local quadratic model for the objective function around x_k, defined as

$$\mathcal{Q}_k(z) = g(x_k)^{\mathrm{T}} z + \frac{1}{2} z^{\mathrm{T}} H(x_k) z. \tag{23}$$

Gould et al. [14] study both directions in the descent pair (s_k, d_k), and select the one that provides the largest descent ratio, measured against the quadratic model. Thus, the Newton direction s_k is chosen whenever the condition

$$\frac{\mathcal{Q}_k(s_k)}{\|s_k\|} \leqslant \frac{\mathcal{Q}_k(d_k)}{\|d_k\|} \tag{24}$$

is satisfied. If the Newton direction has not been modified, and the quadratic model is minimized by this direction, it holds that

$$\mathcal{Q}_k(s_k) = \frac{1}{2} g(x_k)^{\mathrm{T}} s_k. \tag{25}$$

If d_k is replaced in (23) and (24), together with (25), we have the following condition equivalent to (24): Select s_k whenever

$$\frac{g(x_k)^{\mathrm{T}} s_k}{\|s_k\|} \leqslant 2\mathcal{Q}_k(d_k) \tag{26}$$

holds for d_k such that $\|d_k\| = 1$.

Inequality (26) provides a criterion to determine which of both directions in the pair gives the largest potential for descent in the objective function. The proposal in Gould et al. [14] could be generalized to the case in which both directions offer a significant potential for descent. If inequality (26) is extended to consider this case, we have the condition

$$\tau_2 \mathcal{Q}_k(d_k) \geqslant \frac{g(x_k)^{\mathrm{T}} s_k}{\|s_k\|} \geqslant \tau_1 \mathcal{Q}_k(d_k), \tag{27}$$

where τ_1 and τ_2 are prespecified constants satisfying $0 < \tau_2 < 2 < \tau_1$. This multiple condition can be used as a criterion to select among the three alternatives introduced at the beginning of this section, in the following manner:

- The fulfilment of condition (27) implies that both directions provide relevant descent information, therefore, if (27) holds the descent pair is defined as $(\bar{s}_k, \bar{d}_k) = (s_k, d_k)$ and the new iterate is obtained from a curvilinear search as

  $$x_{k+1} = x_k + \alpha_k^2 s_k + \alpha_k d_k.$$

- Otherwise, if

  $$\frac{g(x_k)^{\mathrm{T}} s_k}{\|s_k\|} < \tau_1 \mathscr{Q}_k(d_k) \tag{28}$$

  holds, then the descent provided by the Newton direction s_k is more significant than the descent provided by the negative curvature direction. Therefore, the algorithm performs a linesearch based only on the Newton direction, the descent pair is defined as $(\bar{s}_k, \bar{d}_k) = (s_k, 0)$ and the next iterate is obtained from

  $$x_{k+1} = x_k + \alpha_k^2 s_k.$$

- Finally, if condition

  $$\tau_2 \mathscr{Q}_k(d_k) < \frac{g(x_k)^{\mathrm{T}} s_k}{\|s_k\|} \tag{29}$$

  holds, then the negative curvature direction provides significant descent and should be included in the search. In this case, we will distinguish two alternatives:

  If, in addition, condition

  $$g(x_k)^{\mathrm{T}} d_k \geqslant \tau_3 d_k^{\mathrm{T}} H(x_k) d_k \tag{30}$$

  holds simultaneously to (29) for some positive constant τ_3, then the descent direction provides worse descent information than the negative curvature direction. The procedure conducts a linesearch that considers only the direction of negative curvature, the descent pair is defined as $(\bar{s}_k, \bar{d}_k) = (0, d_k)$ and it computes the next iterate as

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

  Otherwise, if condition

  $$g(x_k)^{\mathrm{T}} d_k < \tau_3 d_k^{\mathrm{T}} H(x_k) d_k \tag{31}$$

  holds simultaneously to (29), then the direction of negative curvature also has significant descent information. In this case, the algorithm could be implemented using only the negative curvature direction, but this would require additional implementation changes. For the sake of simplicity, we have chosen to keep the information in s_k and conduct a search as in the first case, considering both directions, defining the descent pair as $(\bar{s}_k, \bar{d}_k) = (s_k, d_k)$ and computing the new iterate from a curvilinear search as

  $$x_{k+1} = x_k + \alpha_k^2 s_k + \alpha_k d_k.$$

The following result establishes that the preceding rules to select the directions in the search satisfy condition C3, and, as a consequence, that the algorithm is globally convergent.

Lemma 2. *Assume that in each iteration k, the iterates $\{x_k\}$ remain in the set $\mathscr{L}(x_0)$, assumptions $A1$ and $A2$ hold and we are able to compute a descent pair (s_k, d_k) satisfying conditions (2) (4) and $\|d_k\| = 1$. Then, condition $C3$ holds for the descent pair $(\bar{s}_k, \bar{d}_k)$ defined as indicated in the preceding paragraphs.*

Proof. Note that under assumptions A1 and A2 and the condition that the iterates remain in $\mathscr{L}(x_0)$ there exists a constant c_4 such that $\|g(x_k)\| \leqslant c_4$ and from conditions (2) (4) we have

$$g(x_k)^{\mathrm{T}} s_k \leqslant 0, \quad \|s_k\| \leqslant c_2 \|g(x_k)\|, \quad g(x_k)^{\mathrm{T}} d_k \leqslant 0, \quad \|d_k\| \leqslant c_3. \tag{32}$$

As a consequence, it is enough that we study condition (20).

If (27) holds, the desired result follows from $\bar{s}_k = s_k$, $\bar{d}_k = d_k$ and conditions (2) and (4) implying

$$\bar{s}_k^{\mathrm{T}} g(x_k) \leqslant -c_1 \|g(x_k)\|^2 \quad \text{and} \quad \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k \leqslant \theta \lambda_{H_k},$$
$$\Rightarrow \min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k) \leqslant \min(-c_1 \|g(x_k)\|, \theta \lambda_{H_k}).$$

If (28) holds, then $\bar{s}_k = s_k$ and $\bar{d}_k = 0$, implying $\min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k) = s_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|$. From (28), (32), (4) and $\tau_1 > 2$,

$$\frac{s_k^{\mathrm{T}} g(x_k)}{\|s_k\|} \leqslant \tau_1 \left(g(x_k)^{\mathrm{T}} d_k + \frac{1}{2} d_k^{\mathrm{T}} H(x_k) d_k \right) \leqslant \tau_1 \frac{1}{2} d_k^{\mathrm{T}} H(x_k) d_k \leqslant d_k^{\mathrm{T}} H(x_k) d_k \leqslant \theta \lambda_{H_k}. \tag{33}$$

But (2) implies $-\|s_k\| \|g(x_k)\| \leqslant s_k^{\mathrm{T}} g(x_k) \leqslant -c_1 \|g(x_k)\|^2 \Rightarrow \|s_k\| \geqslant c_1 \|g(x_k)\|$ and using this bound in (33), we obtain

$$\frac{s_k^{\mathrm{T}} g(x_k)}{\|g(x_k)\|} \leqslant c_1 \frac{s_k^{\mathrm{T}} g(x_k)}{\|s_k\|} \leqslant c_1 \theta \lambda_{H_k}.$$

From this inequality and $s_k^{\mathrm{T}} g(x_k) \leqslant -c_1 \|g(x_k)\|^2$, we have that

$$\min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k) = s_k^{\mathrm{T}} g(x_k)/\|g(x_k)\| \leqslant \min(-c_1 \|g(x_k)\|, c_1 \theta \lambda_{H_k}).$$

Consider now the case in which conditions (29) and (30) hold. Now we have $\bar{s}_k = 0$ and $\bar{d}_k = d_k$, implying $\min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k) = d_k^{\mathrm{T}} H(x_k) d_k$. From (29), (30) and (3) we have that

$$\frac{s_k^{\mathrm{T}} g(x_k)}{\|g(x_k)\|} \geqslant c_2 \frac{s_k^{\mathrm{T}} g(x_k)}{\|s_k\|} \geqslant c_2 \tau_2 \left(g(x_k)^{\mathrm{T}} d_k + \frac{1}{2} d_k^{\mathrm{T}} H(x_k) d_k \right) \geqslant c_2 \tau_2 \left(\tau_3 + \frac{1}{2} \right) d_k^{\mathrm{T}} H(x_k) d_k.$$

From (2) we then have

$$d_k^{\mathrm{T}} H(x_k) d_k \leqslant -\frac{c_1}{c_2 \tau_2 (\tau_3 + \frac{1}{2})} \|g(x_k)\|$$

and using (4) and the preceding results we obtain

$$\min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k) = d_k^{\mathrm{T}} H(x_k) d_k \leqslant \min\left(-\frac{c_1}{c_2 \tau_2 (\tau_3 + \frac{1}{2})} \|g(x_k)\|, \theta \lambda_{H_k} \right).$$

Finally, if (29) and (31) hold, then using the same arguments as in the first case

$$\bar{s}_k^{\mathrm{T}} g(x_k) \leqslant -c_1 \|g(x_k)\|^2 \quad \text{and} \quad \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k \leqslant \theta \lambda_{H_k} \Rightarrow \min(\bar{s}_k^{\mathrm{T}} g(x_k)/\|g(x_k)\|, \bar{d}_k^{\mathrm{T}} H(x_k) \bar{d}_k)$$
$$\leqslant \min(-c_1 \|g(x_k)\|, \theta \lambda_{H_k}). \qquad \square$$

5. Adapted curvilinear search (ACS) algorithm

The implementation of the proposed algorithm involves several decisions concerning practical details; we describe some of them in the following paragraphs.

- In those iterations where the negative curvature direction is used, it is computationally efficient to scale this direction using the information from the Newton direction, see Moguerza and Prieto [19].
- The direction of negative curvature is computed from the modified Cholesky factorization. But in some cases, even if the factorization detects the presence of negative curvature in the Hessian matrix, it is not efficient to use it in the search. We consider a negative curvature direction d_k to be useful if it holds that

$$d_k^{\mathrm{T}} H(x_k) d_k \leqslant -\varepsilon_d \|H(x_k)\|. \tag{34}$$

If this condition is not satisfied in a given iteration, then we set $d_k = 0$.

The termination condition for the algorithm has been derived from the optimality conditions for problem (1). The procedure terminates whenever it holds that

$$\|g(x_k)\| \leqslant \varepsilon(1 + \|g(x_0)\|). \tag{35}$$

The proposed algorithm follows the scheme presented below:

```
Adapted curvilinear search (ACS) algorithm
Select an initial iterate $x_0$
Let $k = 0$
Repeat
  Compute $s_k$ as an approximate solution for the system (7)
  Compute, if it exists, a direction of negative curvature $d_k$
  Let $d_k = 0$ if (34) is not satisfied
  Select the search model from condition (27)
  Compute $\alpha_k$ to satisfy condition C2
  Update the variables according to the search model used:
    If (27) holds, $x_{k+1} = x_k + \alpha_k^2 s_k + \alpha_k d_k$
    If (28) holds, $x_{k+1} = x_k + \alpha_k^2 s_k$
    If (29) holds, $x_{k+1} = x_k + \alpha_k d_k$
  Let $k = k + 1$
until convergence
```

6. Numerical results

We have conducted numerical experiments using the proposed **ACS** algorithm on a set of test problems. Both the algorithm and the test problems have been implemented and executed using MATLAB 6.5 under Linux.

The algorithm makes use of some parameters. In our implementation we have taken $\varepsilon_d = 10^{-8}$ in (34). For the classifying condition (27), we have taken $\tau_1 = 10$ and $\tau_2 = 0.05$. Finally, in the termination condition (35) we have taken $\varepsilon = 10^{-5}$.

6.1. The test problems

We have selected the test problems from the collection CUTEr (Constrained and Unconstrained Testing Environment, revisited) proposed by Gould et al. [15], and itself an expansion of the original CUTE collection [5]. This set is considered to be a referent for the verification and comparison of nonlinear programming algorithms as the one proposed in this paper. The problems used in the numerical experiments that we have conducted have been chosen to satisfy the following criteria: they must be nonlinear unconstrained problems of dimension between 1 and 500, and having continuous second derivatives available. Only one instance has been included for those problems with selectable dimension. A total of 80 problems in CUTEr satisfy these conditions.

6.2. Analysis of the results

Our main motivation for this work has been to analyze the practical impact of using second-order information, as well as the study of efficient implementations of algorithms that make use of this information. As a consequence, we have defined the computational experiments to compare the results obtained on the test set using the proposed algorithm to those of three alternative procedures.

The first algorithm, **MN**, is based on a modified Newton method and does not use any second-order information. The second algorithm, **ALS**, carries out an adapted linesearch similar to the one described in Gould et al. [14], where in each iteration the best direction in the descent pair is selected from condition (26), and used in a standard linesearch framework. The third algorithm, **CS**, is based on the proposal by Moré and Sorensen [21]. Finally, a fourth algorithm, **ACS**, uses the adapted curvilinear search described in the preceding sections. In all cases, the initial points used have been those specified as default ones in the CUTEr environment.

Among the 80 problems in the set, the modified Cholesky factorization detected the presence of significant negative curvature in 46 instances. The comparative study centers on these 46 problems, as for the remaining cases the four algorithms provide basically equivalent results. In nine out of the 46 problems (DECONVU, DJTL, HEART6LS, HIMMELBF, HYDC20LS, MARATOSB, PFIT2LS, PFIT3LS and PFIT3LS) none of the algorithms were able to reach convergence in less than 300 iterations, and these problems were removed from the study.

Regarding the results obtained by the four algorithms on the 37 remaining problems, algorithms **MN**, **ALS** and **CS** failed to solve problem SCOSINE while algorithms **ALS** and **CS** failed to solve problem SPARSINE. The modified Newton method, **MN**, was unable to solve problem OSBORNEB, while the adapted search algorithm **ALS** failed to solve problem HEART8LS and the curvilinear search algorithm **CS** was unable to solve problem INDEF. All algorithms were stopped whenever the iteration count exceeded 300 and the number of function evaluations exceeded 2600. Problem BARD was the only one in which, though negative curvature was detected using the modified Cholesky factorization, none of the methods made use of it. Also, the adapted linesearch algorithm **ALS** detects negative curvature in one of the iterations for problem BOX3, but it is not used there. It is interesting to note the small number of problems where the norm of the gradient is larger than 10^{-8}: one problem for **MN**, two for **ALS**, two for **CS** and three for **ACS**.

We study first the correlation between the number of curvilinear searches and the performance of the algorithm. Fig. 1 displays in the x-axis the number of curvilinear searches used within each problem, while the y-axis represents the average decrease in the number of iterations when using the **ACS** algorithm instead of the **MN** algorithm for the preceding problems. In all cases, the average reduction is negative, that is, the algorithm that uses negative curvature information is more efficient than the modified Newton algorithm. It is interesting to note that an increase in the number of curvilinear searches conducted within the algorithm tends to be associated with a larger reduction in the total number of iterations. For example, the average decrease in the iteration count when the curvilinear search is used three or more times is always larger than 36. At least for the test set used in this experiment, this result seems to imply that the benefits of using negative curvature are not associated to a particular iteration, but rather have an impact in all occasions in which they are used.

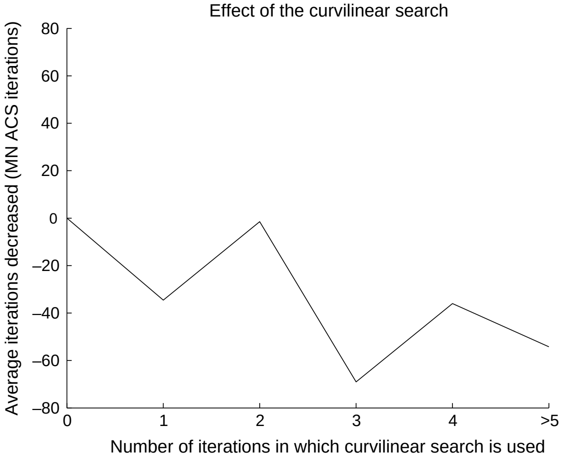

Fig. 1. Correlation between number of curvilinear searches and performance of the algorithm.

In the tables we have used the following labels:

- *pnam*: Problem name.
- *dim*: Dimension of the problem.
- *iter*: Iteration count.
- *fgeval*: Number of function and gradient evaluations.
- *Average 1*: Average numbers of iterations and function evaluations, excluding those problems where an algorithm may have failed.
- *Average 2*: Average numbers of iterations and function evaluations, including those problems where an algorithm may have failed.

From Table 1, the worst-case performance of the adapted curvilinear search algorithm **ACS**, when compared with the modified Newton algorithm **MN**, corresponds to problem HATFLDE, where the number of

Table 1
Overall comparison of iteration and function evaluation counts

pnam	dim	iter				fgeval			
		MN	ALS	CS	ACS	MN	ALS	CS	ACS
ALLINITU	2	8	8	9	9	10	8	11	11
BARD	3	9	9	11	9	9	9	14	9
BEALE	2	9	10	9	9	63	42	88	88
BIGGS6	6	42	39	71	43	69	87	112	90
BOX3	3	8	8	8	8	8	8	8	8
BROWNAL	200	16	19	52	16	91	63	1365	152
BROYDN7D	100	34	27	29	23	248	232	364	285
BRYBND	100	11	11	12	11	16	16	17	15
DENSCHND	3	48	48	48	47	70	49	68	70
DENSCHNE	3	11	12	9	11	14	12	9	11
ENGVAL2	3	17	14	14	14	23	14	16	16
ERRINROS	50	29	35	35	34	70	15	114	109
EXPFIT	2	8	11	10	10	34	17	40	40
GROWTHLS	3	20	24	27	21	32	58	44	99
GULF	3	22	26	31	14	39	41	60	27
HAIRY	2	25	24	16	20	95	125	80	158
HATFLDD	3	22	23	23	22	38	28	30	32
HATFLDE	3	14	17	26	17	17	26	44	33
HEART8LS	8	241		212	197	664		818	761
HELIX	3	16	14	19	18	21	14	36	36
HIELOW	3	8	9	6	6	10	13	9	9
HIMMELBB	2	14	12	10	12	18	12	11	13
HUMPS	2	76	112	78	40	488	1010	1119	848
INDEF	100	244	3		6	246	3		10
KOWOSB	4	10	10	11	6	12	14	15	11
LOGHAIRY	2	96	84	66	68	720	300	797	557
OSBORNEA	5	18	22	20	14	25	137	30	19
OSBORNEB	11		32	16	19		104	39	54
PFIT1LS	3	189	82	159	138	267	98	222	195
SCOSINE	100				94				672
SINQUAD	500	17	25	9	9	27	30	19	19
SNAIL	2	104	103	106	106	215	185	247	247
SPARSINE	100	275			235	1924			2520
SPMSRTLS	28	139	36	24	22	327	184	134	114
STRATEC	10	17	15	16	17	24	18	23	24
VAREIGVL	100	32	53	18	18	36	57	18	20
YFITU	3	36	38	34	34	46	48	47	47
Average 1		35.16	30.63	31.75	26.53	99.44	92.81	162.84	106.63
Average 2		67.16	51.76	57.95	37.84	303.14	229.92	374.81	200.78

Table 2
Percentage of problems with best performance

	MN	ALS	CS	ACS
% best iter	27.03	29.73	32.43	59.46
% best feval	24.32	45.96	18.92	27.03

Table 3
Comparison between **ACS** and the rest of algorithms

	ACS vs MN	ACS vs ALS	ACS vs CS
Best	21	24	21
Worst	8	7	6
Tied	8	6	10

Number of problems showing the best performance in terms of number of iterations.

Table 4
Comparison between **ACS** and the rest of algorithms

	ACS vs MN	ACS vs ALS	ACS vs CS
Best	17	15	18
Worst	16	20	9
Tied	4	2	10

Number of problems showing the best performance in terms of function evaluations.

iterations required for convergence is increased by 3 (17.65%), while the largest improvement corresponds to problem INDEF with a decrease in the number of iterations of 238 (97.54%). The worst case for the **ACS** algorithm when compared to **ALS** implies an increase of three iterations (50%) in problem INDEF, while the largest improvement corresponds to a decrease of 206 iterations (68.67%) in problem SCOSINE. Regarding the curvilinear search algorithm **CS**, the worst performances for **ACS** correspond to problems HAIRY and HIMMELBB, where iteration counts are increased by 4 and 2 (20%) respectively, while the largest improvement, 206 iterations (68.67%), corresponds to problem SCOSINE.

Table 1 also presents a summary of iteration and function evaluation counts for each one of the algorithms, both including and excluding those problems were the algorithms may have failed. From these tables it is interesting to note that the lowest average number of iterations corresponds to the proposed algorithm **ACS**; the lower numbers of function evaluations for algorithms **MN** and **ALS** is explained by the complexity of a search that uses two directions. This effect can be reduced by using a specialized quadratic search procedure, see for example Gill et al. [12]. Nevertheless, the number of function evaluations for **ACS** is lower than the one for **CS**.

Similar conclusions can be reached from Table 2. This table provides the proportion of cases in which each algorithm has been the most efficient one (including ties), both regarding iteration and function evaluation counts.

Finally, Tables 3 and 4 show a comparison of the number of cases in which there was an improvement, a worsening or no change regarding iteration counts and function evaluation counts, respectively, when comparing the proposed algorithm **ACS** to the other methods. In all cases, the number of improved cases associated to the **ACS** algorithm is significant.

7. Conclusions

We have described an efficient procedure that uses directions of negative curvature for the solution of nonconvex unconstrained problems, selecting the most appropriate search to obtain local solutions for these problems. The algorithm is based on a modified Newton model to compute the search directions, and a new proposal of a modified curvilinear search to combine these directions. Particular attention has been paid regarding the conditions to impose on directions of negative curvature before they are considered as a part of the search process.

The computational results illustrate the impact of an appropriate use of negative curvature information: in more than 50% of the problems negative curvature information was used. Due to the low computational cost to obtain these directions, it seems highly advisable to integrate them within algorithms for unconstrained optimization. Furthermore, the results seem to show that using an adaptive search model increases the efficiency of Newton based algorithms.

An important issue raised by the results in this work is to determine the impact of the quality of the negative curvature direction on the computational efficiency of the overall algorithm, that is, if an improvement of the directions based on iterative methods, for example, would yield even better computational results for the algorithm.

Another interesting and promising task would be to verify the performance of the proposed approach within numerical procedures based on the use of approximate directions such as conjugate gradient or quasi-Newton methods.

Acknowledgement

This work has been partially supported by Spanish grants MCYT TIC2003-05892-C05-05, URJC-CM-2006-CET-0391, MEC MTM2004-02334 and MEC MTM2006-14961-C05-05.

References

[1] L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, Pacific Journal of Mathematics 16 (1966) 1 3.

[2] A. Auslender, em Penalty methods for computing points that satisfy second order necessary conditions, Mathematical Programming 17 (1979) 229 238.

[3] T. Bannert, A trust region algorithm for nonsmooth optimization, Mathematical Programming 67 (1994) 247 264.

[4] J.F. Bonnans, G. Launay, Sequential quadratic programming with penalization of the displacement, SIAM Journal on Optimization 5 (1995) 782 812.

[5] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, CUTE: Constrained and unconstrained testing environment, ACM Transactions on Mathematical Software 21 (1995) 123 160.

[6] J.R. Bunch, B.N. Parlett, Direct methods for solving symmetric indefinite systems of linear equations, SIAM Journal on Numerical Analysis 8 (1971) 639 655.

[7] R.H. Byrd, R.B. Schnabel, G.A. Shultz, Approximate solution of the trust region problem by minimization over two dimensional subspaces, Mathematical Programming 40 (1988) 247 263.

[8] A.V. Fiacco, G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968.

[9] R. Fletcher, T.L. Freeman, A modified Newton method for minimization, Journal of Optimization Theory and Applications 23 (1977) 357 372.

[10] D.M. Gay, A trust region approach to linearly constrained optimization, in: Numerical Analysis Proceedings, Dundee 1983, Lecture Notes in Mathematics, Springer Verlag, Heidelberg, 1984, pp. 72 105.

[11] P.E. Gill, W. Murray, Newton type methods for unconstrained and linearly constrained optimization, Mathematical Programming 7 (1974) 311 350.

[12] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, Academic Press, London/New York, 1981.

[13] D. Goldfarb, Curvilinear path steplength algorithms for minimization which use directions of negative curvature, Mathematical Programming 18 (1980) 31 40.

[14] N.I.M. Gould, S. Lucidi, M. Roma, Ph.L. Toint, Exploiting negative curvature directions in linesearch methods for unconstrained optimization, Optimization Methods and Software 14 (2000) 75 98.

[15] N.I.M. Gould, D. Orban, Ph.L. Toint, CUTEr (and SifDec), A Constrained and Unconstrained Testing Environment, revisited, Technical Report TR/PA, 2004.

[16] W.W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and efficient line search, SIAM Journal on Optimization 16 (2005) 170 192.

[17] S. Lucidi, F. Rochetich, M. Roma, Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization, SIAM Journal on Optimization 8 (1998) 916 939.

[18] G. McCormick, A modification of Armijo's step size rule for negative curvature, Mathematical Programming 13 (1977) 111 115.

[19] J.M. Moguerza, F.J. Prieto, An augmented Lagrangian interior point method using directions of negative curvature, Mathematical Programming 95 (2003) 573 616.

[20] J.M. Moguerza, F.J. Prieto, Combining search directions using gradient flows, Mathematical Programming 96 (2003) 529 559.

[21] J.J. Moré, D.C. Sorensen, On the use of directions of negative curvature in a modified Newton method, Mathematical Programming 16 (1979) 1 20.

[22] H. Mukai, E. Polak, A second order algorithm for the general nonlinear programming problem, Journal of Optimization Theory and Applications 4 26 (1978).

[23] S. Sanmatías, E. Vercher, A generalized conjugate gradient algorithm, Journal of Optimization Theory and Applications 98 (1998) 489 502.

[24] S. Sanmatías, M. Roma, Un método de búsqueda lineal con direcciones combinadas para optimización irrestringida, Actas del XXVI Congreso Nacional de Estadística e Investigación Operativa, Úbeda, Spain, 2001.

[25] J. Sun, X. Yang, X. Chen, Quadratic cost flow and the conjugate gradient method, European Journal of Operational Research 164 (2005) 104 114.