A descent Conjugate Gradient Algorithm with quasi-Newton updates

Neculai Andrei
Research Institute for Informatics, Center for Advanced Modeling and Optimization,
8-10, Averescu Avenue, Bucharest 1, Romania.
E-mail: nandrei@ici.ro

March 2, 2017

Abstract. Another conjugate gradient algorithm, based on an improvement of the Perry's method, is presented. In this algorithm the computation of the search direction is based on the quasi-Newton condition rather than the conjugacy one. The idea of Perry to compute the conjugate gradient parameter by equating the conjugate gradient direction with the quasi-Newton one is modified by an appropriate scaling of the conjugate gradient direction. The value of this scaling parameter is determined in such a way to ensure the sufficient descent condition of the search direction. The global convergence of the algorithm is proved for uniformly convex functions. Numerical experiments, using 800 unconstrained optimization test problems, prove that this algorithm is more efficient and more robust than CG-DESCENT. Using five applications from the MINPACK-2 collection with 10^6 variables, we show that the suggested conjugate gradient algorithm is top performer versus CG-DESCENT.

Keywords: Unconstrained optimization; conjugate gradient algorithms; conjugacy condition; quasi-Newton condition; sufficient descent condition; numerical comparisons.

1. Introduction

For solving large scale unconstrained optimization problem

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

where $f: R^n \to R$ is a continuously differentiable function, bounded from below, one of the most elegant, efficient and simplest methods is conjugate gradient. By modest storage requirements, this method represents a significant improvement over the steepest descent algorithms, being very well suited for solving large-scale problems. Besides the corresponding algorithms are not complicated, offering the possibility to be very easy integrated in some other complex industrial and economic applications.

Starting from an initial guess $x_0 \in R^n$, a nonlinear conjugate gradient algorithm generates a sequence $\{x_k\}$ as:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where $\alpha_k > 0$ is obtained by line search, and the directions d_k are computed as:

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad d_0 = -g_0. \tag{3}$$

In (3), β_k is known as the conjugate gradient parameter, $s_k = x_{k+1} - x_k$ and $g_k = \nabla f(x_k)$. In (2) the search direction d_k, assumed to be descent, plays the main role. On the other hand, the step size α_k guarantees the global convergence in some cases and is crucial in efficiency of the algorithm. Usually, the line search in the conjugate gradient algorithms is based on the standard Wolfe conditions [31, 32]:

$$f(x_k+\alpha_k d_k)-f(x_k)\le \rho\alpha_k g_k^T d_k, \tag{4}$$

$$g(x_k+\alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \tag{5}$$

where d_k is supposed to be a descent direction and $0<\rho\le 1/2<\sigma<1$. Also, the strong Wolfe line search conditions consisting of (4) and the following strengthened version of (5):

$$\left|g_{k+1}^T d_k\right| \le -\sigma g_k^T d_k \tag{6}$$

can be used.

Different conjugate gradient algorithms correspond to different choices for the scalar parameter β_k used to generate the search direction (3). Some conjugate gradient methods like Fletcher and Reeves (FR) [14], Dai and Yuan (DY) [11] and Conjugate descent (CD) proposed by Fletcher [13]:

$$\beta_k^{FR}=\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k},\quad \beta_k^{DY}=\frac{g_{k+1}^T g_{k+1}}{y_k^T s_k},\quad \beta_k^{CD}=\frac{g_{k+1}^T g_{k+1}}{-g_k^T s_k},$$

have strong convergence properties, but they may have modest computational performance due to jamming. On the other hand, the methods of Hestenes and Stiefel (HS) [19], Polak and Ribière [26] and Polyak [27] (PRP), or Liu and Storey (LS) [20]:

$$\beta_k^{HS}=\frac{g_{k+1}^T y_k}{y_k^T s_k},\quad \beta_k^{PRP}=\frac{g_{k+1}^T y_k}{g_k^T g_k},\quad \beta_k^{LS}=\frac{g_{k+1}^T y_k}{-g_k^T s_k},$$

may not generally be convergent, but they often have better computational performances.

If the initial direction d_0 is selected as $d_0=-g_0$ and the objective function to be minimized is a convex quadratic one:

$$f(x)=\frac{1}{2}x^T Ax+b^T x+c, \tag{7}$$

and the exact line searches are used, that is

$$\alpha_k=\arg\min_{\alpha>0} f(x_k+\alpha d_k), \tag{8}$$

then the conjugacy condition

$$d_i^T A d_j=0 \tag{9}$$

holds for all $i\ne j$. This relation is the original condition used by Hestenes and Stiefel [19] to derive the conjugate gradient algorithms, mainly for solving symmetric positive-definite systems of linear equations. Let us denote, as usual, $y_k=g_{k+1}-g_k$. Then, for general nonlinear twice continuously differentiable function f, by the mean value theorem, there exists some $\xi\in(0,1)$ such that

$$d_{k+1}^T y_k=\alpha_k d_{k+1}^T \nabla^2 f(x_k+\xi\alpha_k d_k)d_k. \tag{10}$$

Therefore, is seems reasonable to replace the old conjugacy condition (9) from quadratic case with the following one:

$$d_{k+1}^T y_k=0. \tag{11}$$

In order to improve the convergence of the conjugate gradient algorithm, Perry [25] extended the conjugacy condition by incorporating the second-order information. In this respect he used the quasi-Newton condition also known as secant equation:

$$H_{k+1}y_k=s_k, \tag{12}$$

where H_{k+1} is a symmetric approximation to the inverse Hessian of function f. Since for the quasi-Newton method the search direction is computed as $d_{k+1}=-H_{k+1}g_{k+1}$, it follows that:

$$d_{k+1}^T y_k=-(H_{k+1}g_{k+1})^T y_k=-g_{k+1}^T(H_{k+1}y_k)=-g_{k+1}^T s_k,$$

thus obtaining a new conjugacy condition. Further on, Dai and Liao [9] extended this condition and suggested the following new one as:

$$d_{k+1}^T y_k = -u(g_{k+1}^T s_k), \tag{13}$$

where $u \geq 0$ is a scalar. Observe that if the line search is exact, then (13) reduces to the classical conjugacy condition given by (11).

Usually, conjugate gradient algorithms are based on conjugacy condition. In this paper, in order to compute the multiplier β_k in (3), our computational scheme relies on the quasi-Newton condition (12). Perry [25], considering the HS conjugate gradient algorithm, observed that the search direction (3) can be rewritten as:

$$d_{k+1} = -\left[I - \frac{s_k y_k^T}{y_k^T s_k} \right] g_{k+1} \equiv -Q_{k+1}^{HS} g_{k+1}. \tag{14}$$

Notice that Q_{k+1}^{HS} in (14) plays the role of an approximation to the inverse Hessian but is not symmetric. Besides, it is not a memoryless quasi-Newton update. However, d_{k+1} in (14) satisfies the conjugacy condition (11). In order to improve the approximation to the inverse Hessian given by (14), Perry [25] notes that under inexact line search, it is more appropriate to choose the approximation to the inverse Hessian to satisfy the quasi-Newton condition (12) rather than simply conjugacy condition. The idea of Perry was to equate $d_{k+1} = -g_{k+1} + \beta_k s_k$ to $-B_{k+1}^{-1} g_{k+1}$, where B_{k+1} is an approximation to the Hessian $\nabla^2 f(x_{k+1})$. Therefore, by the equality

$$-g_{k+1} + \beta_k s_k = -B_{k+1}^{-1} g_{k+1}, \tag{15}$$

after some simple algebraic manipulations we get the Perry's choice for β_k and the corresponding search direction as:

$$\beta_k = \frac{y_k^T g_{k+1} - s_k^T g_{k+1}}{y_k^T s_k} \tag{16}$$

$$d_{k+1} = -\left[I - \frac{s_k y_k^T}{y_k^T s_k} + \frac{s_k s_k^T}{y_k^T s_k} \right] g_{k+1} \equiv -Q_{k+1}^{P} g_{k+1}. \tag{17}$$

It is worth saying that if the exact line search is performed, than (17) is identical to the HS conjugate gradient algorithm expressed as in (14). More than this, Q_{k+1}^{P} is not symmetric and does not satisfy the true quasi-Newton (secant) condition. However, the Perry's direction (17) satisfies the Dai and Liao [9] conjugacy condition (13) with $u = 1$.

The purpose of this paper is to improve the Perry's approach. In section 2 a critical development of the Perry's approach is considered by showing its limits and suggesting a new descent conjugate gradient algorithm with quasi-Newton updates. Section 3 is devoted to prove the convergence of the corresponding algorithm for uniformly convex functions. In section 4 the numerical performances of this algorithm on 800 unconstrained optimization test problems and comparisons versus CG-DESCENT [18] are presented. By solving five applications from the MINPACK-2 collection [6] with 10^6 variables we show that our algorithm is top performer versus CG-DESCENT.

2. Descent Conjugate Gradient Algorithm with quasi-Newton updates

In order to define the algorithm, in this section, we consider a strategy based on the quasi-Newton condition rather than on the conjugacy condition. The advantage of this approach is the inclusion of the second order information, contained in the Hessian matrix, into the computational scheme, thus hopping to improve the convergence of the corresponding algorithm.

For the very beginning, observe that the quasi-Newton direction $d_{k+1} = -B_{k+1}^{-1} g_{k+1}$ is a linear combination of the columns of an approximation to the inverse Hessian B_{k+1}^{-1}, where the

coefficients in this linear combination are the negative components of the gradient g_{k+1}. On the other hand, the conjugate search direction $d_{k+1} = -g_{k+1} + \beta_k s_k$ mainly is the negative gradient g_{k+1} altered by a scaling of the previous search direction. The difference between these two search directions is significant and, as we can see, apparently a lot of information given by the inverse Hessian is not considered in the search direction of the conjugate gradient algorithm. However, in some conjugate gradient algorithms, for example the Hestenes and Stiefel [19], the conjugate parameter β_k in the search direction is obtained by requiring the search direction d_{k+1} to be B_k-conjugate to d_k, i.e. enforcing the condition $d_{k+1}^T B_k d_k = 0$. This is an important property, but this condition is involving B_k and not its inverse. Using the quasi-Newton condition improves the conjugate gradient search direction to take into consideration the information given by the inverse Hessian.

As we have seen the Perry scheme [25] is based on the quasi-Newton condition, i.e. the derivation of the β_k in (16) is determined by the equating $d_{k+1} = -g_{k+1} + \beta_k s_k$ to $-B_{k+1}^{-1} g_k$, where B_{k+1} is an approximation of the Hessian. However, if the Newton direction $-B_{k+1}^{-1} g_{k+1}$ is contained into the cone generated by $-g_{k+1}$ and s_k, then β_k cannot alone ensure the equality (15). It is clear that the above condition (15) guarantees that $-g_{k+1} + \beta_k s_k$ and the quasi-Newton direction $-B_{k+1}^{-1} g_{k+1}$ are only collinear [30]. In order to skip over this limitation we introduce an appropriate scaling of the conjugate gradient direction and consider the equality:

$$-\theta_{k+1} g_{k+1} + \theta_{k+1} \beta_k s_k = -B_{k+1}^{-1} g_{k+1}, \tag{18}$$

where $\theta_{k+1} > 0$ is a scaling parameter which follows to be determined. As above, after some simple algebraic manipulations on (18) we get a new expression for the conjugate gradient parameter β_k and the corresponding direction as:

$$\beta_k = \frac{y_k^T g_{k+1} - (1/\theta_{k+1}) s_k^T g_{k+1}}{y_k^T s_k}, \tag{19}$$

$$d_{k+1} = -\left[I - \frac{s_k y_k^T}{y_k^T s_k} + \frac{1}{\theta_{k+1}} \frac{s_k s_k^T}{y_k^T s_k} \right] g_{k+1} \equiv -P_{k+1} g_{k+1}. \tag{20}$$

Observe that with $\theta_{k+1} = 1$, (20) coincides with Perry's direction (17). On the other hand, when $\theta_{k+1} \to \infty$, then (20) coincides with HS search direction (14). Therefore, (20) provides a general frame where a continuous variation between the Hestenss and Stiefel [19] conjugate gradient algorithm and the Perry's one [25] is obtained. Besides, if the line search is exact ($s_k^T g_{k+1} = 0$), than the algorithm is indifferent to the selection of θ_{k+1}. In this case the search direction given by (20) is identical with HS strategy.

Remark 2.1. An important property of β_k given by (19) is that it is also the solution of the following one-parameter quadratic model of function f on β:

$$\min_\beta g_{k+1}^T d(\beta) + \frac{1}{2} d(\beta)^T B_{k+1} d(\beta),$$

where $d(\beta) = -g_{k+1} + \beta s_k$, the symmetrical and positive definite matrix B_{k+1} is an approximation of the Hessian $\nabla^2 f(x_{k+1})$ such that the generalized quasi-Newton equation $B_{k+1} s_k = \theta_{k+1} y_k$, with $\theta_{k+1} \neq 0$, is satisfied. Therefore, in other words, the solution of the symmetrical linear algebraic system $B_{k+1} d(\beta) = -g_{k+1}$ can be expressed as $d(\beta) = -P_{k+1} g_{k+1}$, where P_{k+1} is defined by (20) is not a symmetrical matrix. This is indeed a remarkable property (see also [21]). ■

In the following, we shall develop a procedure for θ_{k+1} computation. The idea is to find θ_{k+1} in such a way to ensure the sufficient descent condition of the search direction (20).

Proposition 2.1. *If*

$$\theta_{k+1} = \frac{y_k^T s_k}{\|y_k\|^2}, \tag{21}$$

then the search direction (20) satisfies the sufficient descent condition

$$g_{k+1}^T d_{k+1} = -\frac{3}{4}\|g_{k+1}\|^2 \le 0. \tag{22}$$

Proof. From (20) we get:

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k} - \frac{1}{\theta_{k+1}}\frac{(s_k^T g_{k+1})^2}{y_k^T s_k}. \tag{23}$$

Now, using the classical inequality $u^T v \le \frac{1}{2}\left[\|u\|^2 + \|v\|^2\right]$, where $u, v \in R^n$ are arbitrary vectors, and considering

$$u = \frac{1}{\sqrt{2}}(y_k^T s_k) g_{k+1}, \quad v = \sqrt{2}(s_k^T g_{k+1}) y_k$$

we get:

$$\frac{(y_k^T g_{k+1})(s_k^T g_{k+1})}{y_k^T s_k} = \frac{(y_k^T g_{k+1})(y_k^T s_k)(s_k^T g_{k+1})}{(y_k^T s_k)^2} = \frac{[(1/\sqrt{2})(y_k^T s_k) g_{k+1}]^T [\sqrt{2}(s_k^T g_{k+1}) y_k]}{(y_k^T s_k)^2}$$

$$\le \frac{\frac{1}{2}\left[\frac{1}{2}(y_k^T s_k)^2 \|g_{k+1}\|^2 + 2(s_k^T g_{k+1})^2 \|y_k\|^2\right]}{(y_k^T s_k)^2} = \frac{1}{4}\|g_{k+1}\|^2 + \frac{(s_k^T g_{k+1})^2}{(y_k^T s_k)^2}\|y_k\|^2.$$

Hence,

$$g_{k+1}^T d_{k+1} \le -\frac{3}{4}\|g_{k+1}\|^2 + \frac{(s_k^T g_{k+1})^2}{y_k^T s_k}\left[\frac{\|y_k\|^2}{y_k^T s_k} - \frac{1}{\theta_{k+1}}\right].$$

Obviously, if θ_{k+1} is selected as in (21), then the search direction satisfies the sufficient descent condition (22). ■

It is worth saying that with (21) the search direction (20) is

$$d_{k+1} = -g_{k+1} + \left[\frac{y_k^T g_{k+1}}{y_k^T s_k} - \frac{\|y_k\|^2}{y_k^T s_k}\frac{s_k^T g_{k+1}}{y_k^T s_k}\right] s_k. \tag{24}$$

Remark 2.2. From the proof of Proposition 2.1 we see that if

$$\theta_{k+1} \le \frac{y_k^T s_k}{\|y_k\|^2}, \tag{25}$$

then the search direction (20) satisfies a modified sufficient descent condition. However, in our numerical experiments the value of the parameter θ_{k+1} is computed as in (21). ■

Proposition 2.2. *Suppose that the stepsize* α_k *is determined by the Wolfe line search conditions (4) and (5). Then the search direction (24) satisfies the Dai and Liao conjugacy condition* $y_k^T d_{k+1} = -v_k(s_k^T g_{k+1})$, *where* $v_k = \|y_k\|^2 / (y_k^T s_k) \geq 0$.

Proof. By direct computation from (24) we get

$$y_k^T d_{k+1} = -\frac{\|y_k\|^2}{y_k^T s_k}(s_k^T g_{k+1}) \equiv -v_k(s_k^T g_{k+1}).$$

Using the Wolfe line search (4) and (5) we have that $y_k^T s_k > 0$ showing that the Dai and Liao conjugacy condition is satisfied by the search direction (24). ■

The search direction (24) in our algorithm is not very much different by the search direction given by Hager and Zhang [17]. It is worth emphasizing that the computational scheme of Hager and Zhang is obtained by *ex abrupto* deleting a term from the search direction for the memoryless quasi-Newton scheme of Perry [24] and Shanno [29]. On the other hand, our computational scheme (2)-(24) is generated by equating a scaling of the conjugate gradient direction with the quasi-Newton direction, where the scaling parameter is determined such as the resulting search direction satisfies the sufficient descent condition.

In conjugate gradient methods the step lengths may differ from 1 in a very unpredictable way [23]. They can be larger or smaller than 1 depending on how the problem is scaled. In the following we consider an acceleration scheme, we have presented in [3] (see also [2]). Basically the acceleration scheme modifies the step length α_k in a multiplicative manner to improve the reduction of the function values along the iterations. In accelerated algorithm instead of (2) the new estimation of the minimum point is computed as

$$x_{k+1} = x_k + \xi_k \alpha_k d_k, \tag{26}$$

where

$$\xi_k = -\frac{a_k}{b_k}, \tag{27}$$

$a_k = \alpha_k g_k^T d_k$, $b_k = -\alpha_k (g_k - g_z)^T d_k$, $g_z = \nabla f(z)$ and $z = x_k + \alpha_k d_k$. Hence, if $b_k \neq 0$, then the new estimation of the solution is computed as $x_{k+1} = x_k + \xi_k \alpha_k d_k$, otherwise $x_{k+1} = x_k + \alpha_k d_k$. Using the definitions of g_k, s_k, y_k and the above acceleration scheme (26) and (27) we can present the following conjugate gradient algorithm.

Algorithm DCGQN

Step 1. Select the initial starting point $x_0 \in dom\, f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $k = 0$. Select $0 < \rho \leq 1/2$ and $1/2 < \sigma < 1$ and a value for the parameter ε.

Step 2. Test a criterion for stopping the iterations. For example, if $\|g_k\|_\infty \leq \varepsilon$, then stop; otherwise continue with step 3.

Step 3. Using the Wolfe line search conditions (4) and (5) determine the steplength α_k.

Step 4. Compute: $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_k = g_k - g_z$.

Step 5. Compute: $a_k = \alpha_k g_k^T d_k$, and $b_k = -\alpha_k y_k^T d_k$.

Step 6. If $b_k \neq 0$, then compute $\xi_k = -a_k / b_k$ and update the variables as

$x_{k+1} = x_k + \xi_k \alpha_k d_k$; otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$. Compute f_{k+1} and g_{k+1}. Compute $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.

Step 7. Compute the search direction d_{k+1} as in (24).

Step 8. Restart criterion. If the restart criterion of Powell $\left|g_{k+1}^T g_k\right| > 0.2\left\|g_{k+1}\right\|^2$ is satisfied, then set $d_{k+1} = -g_{k+1}$.

Step 9. Compute the initial guess $\alpha_k = \alpha_{k-1}\left\|d_{k-1}\right\| / \left\|d_k\right\|$, set $k = k+1$ and continue with step 2. ■

If function f is bounded along the direction d_k then there exists a stepsize α_k satisfying the Wolfe line search conditions (4) and (5). In our algorithm when the Powell restart condition [28] is satisfied, then we restart the algorithm with the negative gradient $-g_{k+1}$. Some more sophisticated reasons for restarting the conjugate gradient algorithms have been proposed in the literature [10]. However, in this paper we are interested in the performance of a conjugate gradient algorithm that uses this restart criterion of Powell associated to a direction satisfying both the descent and the conjugacy conditions. Under reasonable assumptions, the Wolfe conditions and the Powell restart criterion are sufficient to prove the global convergence of the algorithm. The first trial of the step length crucially affects the practical behavior of the algorithm. At every iteration $k \geq 1$ the starting guess for the step α_k in the line search is computed as $\alpha_{k-1}\left\|d_{k-1}\right\| / \left\|d_k\right\|$. For uniformly convex functions, we can prove the linear convergence of the acceleration scheme given by (26) and (27) [3].

3. Global convergence analysis

Assume that:

(i) *The level set* $S = \left\{x \in R^n : f(x) \leq f(x_0)\right\}$ *is bounded.*

(ii) *In a neighborhood* N *of* S *the function* f *is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant* $L > 0$ *such that* $\left\|\nabla f(x) - \nabla f(y)\right\| \leq L\left\|x - y\right\|$, *for all* $x, y \in N$.

Under these assumptions on f there exists a constant $\Gamma \geq 0$ such that $\left\|\nabla f(x)\right\| \leq \Gamma$ for all $x \in S$. Notice that the assumption that the function f is bounded below is weaker that the usual assumption that the level set is bounded.

Although the search directions generated by the algorithm are always descent directions, to ensure convergence of the algorithm we need to constrain the choice of the step-length α_k. The following proposition shows that the Wolfe line search always gives a lower bound for the step-length α_k.

Proposition 3.1. *Suppose that* d_k *is a descent direction and the gradient* ∇f *satisfies the Lipschitz condition*

$$\left\|\nabla f(x) - \nabla f(x_k)\right\| \leq L\left\|x - x_k\right\|,$$

for all x *on the line segment connecting* x_k *and* x_{k+1}, *where* L *is a positive constant. If the line search satisfies the strong Wolfe conditions (4) and (6), then*

$$\alpha_k \geq \frac{(1-\sigma)\left|g_k^T d_k\right|}{L\left\|d_k\right\|^2}.$$

Proof. Subtracting $g_k^T d_k$ from both sides of (6) and using the Lipschitz continuity we get

$$(\sigma-1)g_k^T d_k \leq (g_{k+1}-g_k)^T d_k = y_k^T d_k \leq \left\|y_k\right\|\left\|d_k\right\| \leq \alpha_k L\left\|d_k\right\|^2.$$

Since d_k is a descent direction and $\sigma<1,$ we get the conclusion of the proposition ■

For any conjugate gradient method with strong Wolfe line search the following general result holds [23].

Proposition 3.2. *Suppose that the above assumptions hold. Consider a conjugate gradient algorithm in which, for all* $k\geq 0,$ *the search direction* d_k *is a descent direction and the steplength* α_k *is determined by the Wolfe line search conditions. If*

$$\sum_{k\geq 0}\frac{1}{\left\|d_k\right\|^2}=\infty, \tag{28}$$

then the algorithm converges in the sense that

$$\liminf_{k\to\infty}\left\|g_k\right\|=0. \tag{29}$$

For *uniformly convex functions* we can prove that the norm of the direction d_{k+1} computed as in (24) is bounded above. Therefore, by proposition 3.2 we can prove the following result.

Theorem 3.1. *Suppose that the assumptions (i) and (ii) hold. Consider the algorithm DCGQN where the search direction* d_k *is given by (24). Suppose that* d_k *is a descent direction and* α_k *is computed by the Wolfe line search. Suppose that* f *is a uniformly convex function on* $S,$ *i.e. there exists a constant* $\mu>0$ *such that*

$$(\nabla f(x)-\nabla f(y))^T(x-y)\geq \mu\left\|x-y\right\|^2 \tag{30}$$

for all $x,y\in N.$ *Then*

$$\lim_{k\to\infty}\left\|g_k\right\|=0. \tag{31}$$

Proof. From Lipschitz continuity we have $\left\|y_k\right\|\leq L\left\|s_k\right\|.$ On the other hand, from uniform convexity it follows that $y_k^T s_k \geq \mu\left\|s_k\right\|^2$. Now, using (24) we have

$$\begin{aligned}\left\|d_{k+1}\right\| &\leq \left\|g_{k+1}\right\|+\frac{\left|y_k^T g_{k+1}\right|}{y_k^T s_k}\left\|s_k\right\|+\frac{\left\|y_k\right\|^2}{y_k^T s_k}\frac{\left|s_k^T g_{k+1}\right|}{y_k^T s_k}\left\|s_k\right\| \\ &\leq \Gamma+\frac{\left\|y_k\right\|\Gamma\left\|s_k\right\|}{\mu\left\|s_k\right\|^2}+\frac{\left\|y_k\right\|^2}{\mu\left\|s_k\right\|^2}\frac{\left\|s_k\right\|\Gamma\left\|s_k\right\|}{\mu\left\|s_k\right\|^2} \leq \Gamma+\frac{L\Gamma}{\mu}+\frac{L^2\Gamma}{\mu^2},\end{aligned}$$

showing that (28) is true. By proposition 3.2 it follows that (29) is true, which for uniformly convex functions is equivalent to (31) ■

For *general nonlinear functions*, having in view that the search direction (24) is close to the search direction used in the CG-DESCENT algorithm, the convergence of the algorithm follows the same procedure as that used by Hager and Zhang in [17].

4. Numerical results

The DCGQN algorithm was implemented in double precision Fortran using loop unrolling of depth 5 and compiled with f77 (default compiler settings) and run on a Workstation Intel Pentium 4 with 1.8 GHz. We selected a number of 80 large-scale unconstrained optimization test functions in generalized or extended form, of different structure and complexity, presented in [1]. For each test function we have considered 10 numerical experiments with the number of variables increasing as $n = 1000, 2000, \ldots, 10000$. The algorithm uses the Wolfe line search conditions with cubic interpolation, $\rho = 0.0001$, $\sigma = 0.8$ and the same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector.

Since, CG-DESCENT [18] is among the best nonlinear conjugate gradient algorithms proposed in the literature, but not necessarily the best, in the first set on numerical experiments we compare our algorithm DCGQN versus CG-DESCENT (version 1.4). The algorithms we compare in these numerical experiments find local solutions. Therefore, the comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2, for problem $i = 1, \ldots, 800$, respectively. We say that, in the particular problem i, the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{32}$$

and the number of iterations (#iter), or the number of function-gradient evaluations (#fg), or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively.

Figure 1 shows the Dolan and Moré's [12] performance profiles subject to CPU time metric. Form Figure 1, comparing DCGQN versus CG-DESCENT with Wolfe line search (version 1.4, Wolfe line search, default settings, $\|g_k\|_\infty \leq 10^{-6}$), subject to the number of iterations, we see that DCGQN was better in 641 problems (i.e. it achieved the minimum number of iterations for solving 641 problems), CG-DESCENT was better in 74 problems and they achieved the same number of iterations in 56 problems, etc. Out of 800 problems, we considered in this numerical study, only for 771 problems does the criterion (32) hold. Therefore, in comparison with CG-DESCENT, on average, DCGQN appears to generate the best search direction and the best step-length. We see that this computational scheme based on scaling the conjugate gradient search direction and equating it to the quasi-Newton direction lead us to a conjugate gradient algorithm which substantially outperforms the CG-DESCENT, being way more efficient and more robust.

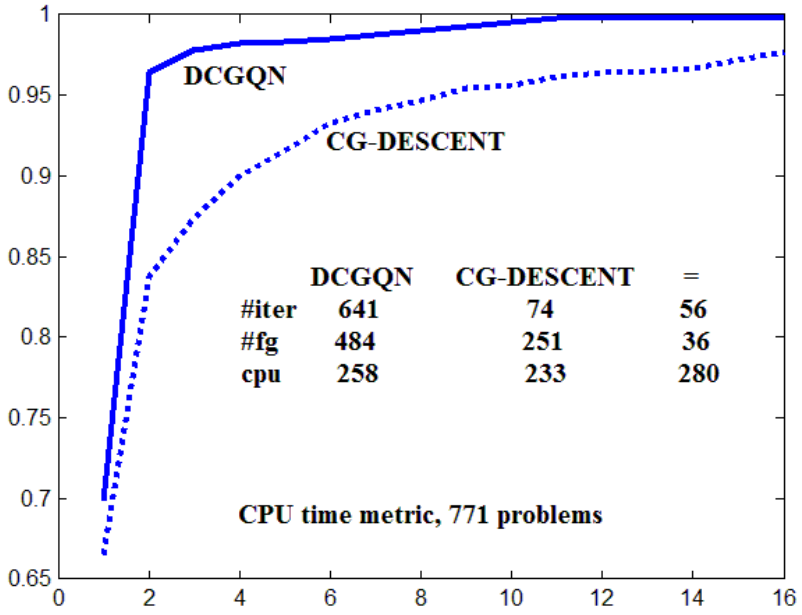

Fig.1. DCGQN versus CG-DESCENT.

Remark 4.1. A theoretical justification of this behavior of DCGQN versus CG-DESCENT is as follows. The search direction of DCGQN given by (24) can be written as:

$$d_{k+1} = -P_{k+1}^{DCGQN} g_{k+1},$$

where

$$P_{k+1}^{DCGQN} = I - \frac{s_k y_k^T}{y_k^T s_k} + \frac{\|y_k\|^2}{y_k^T s_k} \frac{s_k s_k^T}{y_k^T s_k}. \tag{33}$$

By the Wolfe line search conditions (4) and (5) we have that $y_k^T s_k > 0$. Therefore, the vectors y_k and s_k are nonzero vectors. Let V be the vector space spanned by $\{s_k, y_k\}$. Clearly, $\dim(V) \le 2$ and $\dim(V^{\perp}) \ge n-2$. Thus, there exists a set of mutually unit orthogonal vectors $\{u_k^i\}_{i=1}^{n-2} \subset V^{\perp}$ such that $s_k^T u_k^i = y_k^T u_k^i = 0, \quad i = 1,\ldots,n-2,$ which from (33) leads to $P_{k+1}^{DCGQN} u_k^i = u_k^i,$ $i = 1,\ldots,n-2.$ Therefore, the matrix P_{k+1}^{DCGQN} has $n-2$ eigenvalues equal to 1, which corresponds to $\{u_k^i\}_{i=1}^{n-2}$ as eigenvectors. Now, we are interested to find the rest of the two remaining eigenvalues, denoted as λ_{k+1}^{+} and λ_{k+1}^{-}, respectively. After a simple algebra the trace and the determinant of P_{k+1}^{DCGQN} are as follows:

$$tr(P_{k+1}^{DCGQN}) = n - 1 + a_k,$$
$$\det(P_{k+1}^{DCGQN}) = a_k,$$

where

$$a_k = \frac{\|s_k\|^2 \|y_k\|^2}{(y_k^T s_k)^2}.$$

Observe that $a_k > 1$. Therefore, the other eigenvalues of P_{k+1}^{DCGQN} are the roots of the following quadratic polynomial:

$$\lambda^2-(1+a_k)\lambda+a_k=0,$$

as $\lambda_{k+1}^{+}=a_k>1$ and $\lambda_{k+1}^{-}=1$. Therefore the eigenvalues of P_{k+1}^{DCGQN} are $\{1,1,\ldots,1,a_k\}$.
On the other hand, the CG-DESCENT search direction [17, 18] is given by:

$$d_{k+1}^{HZ}=-P_{k+1}^{HZ}g_{k+1},$$

where

$$P_{k+1}^{HZ}=\left[I-\frac{s_k y_k^T}{y_k^T s_k}+2\frac{\|y_k\|^2}{y_k^T s_k}\frac{s_k s_k^T}{y_k^T s_k}\right]. \tag{34}$$

Using similar arguments as above, the matrix P_{k+1}^{HZ} has $n-2$ eigenvalues equal to 1 and two other denoted as μ_{k+1}^{+} and μ_{k+1}^{-}. Computing the trace and the determinant of P_{k+1}^{HZ} we get:

$$tr(P_{k+1}^{HZ})=n-1+2a_k,$$
$$\det(P_{k+1}^{HZ})=2a_k.$$

Therefore, the other eigenvalues of P_{k+1}^{HZ} are the roots of the quadratic polynomial:

$$\mu^2-(1+2a_k)\mu+2a_k=0,$$

as $\mu_{k+1}^{+}=2a_k>1$ and $\mu_{k+1}^{-}=1$. Therefore the eigenvalues of P_{k+1}^{HZ} are $\{1,1,\ldots,1,2a_k\}$.
As we know, the rate of convergence of conjugate gradient depends strongly by the distribution of eigenvalues of the iterate matrix (in our case P_{k+1}^{DCGQN} or P_{k+1}^{HZ}) (see [4]). We see that the eigenvalues of P_{k+1}^{DCGQN} are better clustered around 1 than the eigenvalues of P_{k+1}^{HZ}. This is the reason why DCGQN algorithm is top performer versus CG-DESCENT. ■

In the following, in the second set of numerical experiments, we present comparisons between DCGQN and CG-DESCENT conjugate gradient algorithms for solving five applications from the MINPACK-2 test problem collection [6]. In Table 1 we present these applications, as well as the values of their parameters.

Table 1.
Applications from the MINPACK-2 collection.

A1	Elastic–plastic torsion [15, pp. 41–55], $c=5$
A2	Pressure distribution in a journal bearing [8], $b=10,\ \varepsilon=0.1$
A3	Optimal design with composite materials [16], $\lambda=0.008$
A4	Steady-state combustion [5, pp. 292–299], [7], $\lambda=5$
A5	Minimal surfaces with Enneper conditions [22, pp. 80–85]

The infinite-dimensional version of these problems is transformed into a finite element approximation by triangulation. Thus, a finite-dimensional minimization problem is obtained whose variables are the values of the piecewise linear function at the vertices of the triangulation. The discretization steps are $nx=1{,}000$ and $ny=1{,}000$, thus obtaining minimization problems with 1,000,000 variables. A comparison between DCGQN (Powell restart criterion, $\|\nabla f(x_k)\|_\infty\le 10^{-6}$, $\rho=0.0001$, $\sigma=0.8$) and CG-DESCENT (version 1.4, Wolfe line search, default settings, $\|\nabla f(x_k)\|_\infty\le 10^{-6}$) for solving these applications is given in Table 2.

Table 2.
Performance of DCGQN versus CG-DESCENT. 1,000,000 variables. CPU seconds

	DCGQN			CG-DESCENT		
	#iter	#fg	cpu	#iter	#fg	cpu
A1	1113	2257	**355.84**	1145	2291	481.40
A2	2845	5718	**1141.47**	3370	6741	1869.77
A3	4770	9636	**2814.16**	4814	9630	3979.26
A4	1413	2864	**2110.20**	1802	3605	3802.37
A5	1279	2587	**575.62**	1225	2451	756.96
TOTAL	11420	23062	**6997.29**	12356	24718	10889.76

Form Table 2, we see that, subject to the CPU time metric, the DCGQN algorithm is top performer and the difference is significant, about **3892.47 seconds** for solving all these five applications.

5. Conclusions

Plenty of conjugate gradient algorithms are known in the literature. In this paper we have presented another one based on the quasi-Newton condition. The search direction is computed by equating a scaling of the classical conjugate gradient search direction with the quasi-Newton one. The scaling parameter is determined in such a way that the resulting search direction of the algorithm satisfies the sufficient descent condition. In our algorithm the step length is computed using the classical Wolfe line search conditions. The updating formulas (2) and (24) are not complicated and we proved that it satisfies the sufficient descent condition $g_k^T d_k \leq -\frac{3}{4}\|g_k\|$, independent of the line search procedure as long as $y_k^T s_k > 0$. For uniformly convex function the convergence of the algorithm was proved under classical assumptions. As a by product, we proved once again that the rate of convergence of the conjugate gradient algorithms depends strongly by the distribution of the eigenvalues of the iterate matrix defining the search direction. In numerical experiments the algorithm DCGQN proved to be more efficient and more robust versus CG-DESCENT on a large number of unconstrained optimization test problems of different structures and complexities. For solving large-scale nonlinear engineering optimization from MINPACK-2 collection the implementation of our algorithm proves to be way more efficient than the CG-DESCENT implementation.

References

[1] Andrei, N., *An unconstrained optimization test functions collection.* Advanced Modeling and Optimization, 10 (2008), pp. 147-161.

[2] Andrei, N., *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*, Numerical Algorithms, 42 (2006), pp. 63-73.

[3] Andrei, N., *Acceleration of conjugate gradient algorithms for unconstrained optimization.* Applied Mathematics and Computation, 213 (2009), 361-369.

[4] Andrei, N., *Eigenvalues versus singular values study in conjugate gradient algorithms for large-scale unconstrained optimization.* Optimization Methods & Software. DOI: 10.1080/10556788.2016.1225211

[5] Aris, R., *The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts*, Oxford, 1975.

[6] Averick, B.M., Carter, R.G., Moré, J.J., Xue, G.L., *The MINPACK-2 test problem collection*, Mathematics and Computer Science Division, Argonne National Laboratory, Preprint MCS-P153-0692, June 1992.
[7] Bebernes, J., Eberly, D., *Mathematical Problems from Combustion Theory*, in: Applied Mathematical Sciences, vol. 83, Springer-Verlag, 1989.
[8] Cimatti, G., *On a problem of the theory of lubrication governed by a variational inequality*, Applied Mathematics and Optimization 3 (1977) 227–242.
[9] Dai, Y.H. and Liao, L.Z., *New conjugate conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim. 43, 87-101 (2001)
[10] Dai, Y.H., Liao, L.Z., Duan, Li, *On restart procedures for the conjugate gradient method.* Numerical Algorithms 35 (2004), pp. 249-260.
[11] Dai, Y.H., Yuan, Y., *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177-182.
[12] Dolan, E., Morè, J.J., *Benchmarking optimization software with performance profiles.* Mathematical Programming Ser. A, 91, 2002, pp.201-213.
[13] Fletcher, R., *Practical Methods of Optimization, vol. 1: Unconstrained Optimization*, John Wiley & Sons, New York, 1987.
[14] Fletcher, R. and Reeves, C.M., *Function minimization by conjugate gradients* Comput. J. **7**, 149-154 (1964)
[15] Glowinski, R., *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, 1984.
[16] Goodman, J., Kohn, R., Reyna, L., *Numerical study of a relaxed variational problem from optimal design,* Computer Methods in Applied Mechanics and Engineering 57 (1986) 107–127.
[17] Hager, W.W., Zhang, H., *A new conjugate gradient method with guaranteed descent and an efficient line search.* SIAM Journal on Optimization, 16, (2005) 170-192.
[18] Hager, W.W., Zhang, H., *Algorithm 851: CG-DESCENT, a conjugate gradient method with guaranteed descent.* ACM Transaction on Mathematical Software, 32 (2006) 113-137.
[19] Hestenes, M.R., Stiefel, E.L., *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp.409-436.
[20] Liu, Y., Storey, C., *Efficient generalized conjugate gradient algorithms, Part 1: Theory.* Journal of Optimization Theory and Applications, 69 (1991), pp.129-137.
[21] Liu, D., Xu, G., *A Perry descent conjugate gradient method with restricted spectrum.* Technical Report of Optimization No: 2010-11-08, Control Theory Laboratory, Department of Mathematics, University of Tianjin, 2011. [Optimization Online, March 2011.]
[22] Nitsche, J.C.C., *Lectures On Minimal Surfaces*, Vol. 1, Cambridge University Press, 1989.
[23] Nocedal, J., *Conjugate gradient methods and nonlinear optimization.* In Linear and nonlinear Conjugate Gradient related methods, L. Adams and J.L. Nazareth (eds.), SIAM, 1996, pp.9-23.
[24] Perry, A., *A class of conjugate gradient algorithms with a two step variable metric memory.* Discussion paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1977.
[25] Perry, A., *A modified conjugate gradient algorithm.* Operations Research, Technical Notes, 26 (1978) 1073-1078.
[26] Polak, E., Ribière, G., *Note sur la convergence de directions conjuguée,* Rev. Francaise Informat Recherche Operationelle, 3e Année 16 (1969) 35-43.
[27] Polyak, B.T., *The conjugate gradient method in extreme problems*. USSR Comp. Math. Math. Phys. 9, (1969), pp. 94-112.
[28] Powell, M.J.D., *Restart procedures of the conjugate gradient method.* Mathematical Programming, 2 (1977), pp.241-254.

[29] Shanno, D.F., *On the convergence of a new conjugate gradient algorithm.* SIAM J. Numer. Anal., 15 (1978), pp.1247-1257.
[30] Sherali, H.D., Ulular, O., *Conjugate gradient methods using quasi-Newton updates with inexact line search.* Journal of Mathematical Analysis and Applications, 150, (1990), pp.359-377.
[31] Wolfe, P., (1969) *Convergence conditions for ascent methods.* SIAM Review, 11, 1969, pp. 226-235.
[32] Wolfe, P., (1971) *Convergence conditions for ascent methiods. II: Some corrections.* SIAM Review, 13, 1971, pp.185-188.