Dai-Yuan Conjugate Gradient Algorithm with Sufficient Descent and Conjugacy Conditions for Unconstrained Optimization

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

Abstract. A modification of the Dai-Yuan conjugate gradient algorithm is proposed. Using the exact line search, the algorithm reduces to the original version of the Dai and Yuan computational scheme. For inexact line search the algorithm satisfies both the sufficient descent and conjugacy condition. A global convergence result is proved when the Wolfe line search conditions are used. Computational results, for a set consisting of 750 unconstrained optimization test problems, show that this new conjugate gradient algorithm substantially outperforms the Dai and Yuan conjugate gradient algorithm and is close to its hybrid variants.

MSC: 49M07, 49M10, 90C06, 65K
Keywords: Unconstrained optimization, conjugate gradient method, sufficient descent condition, conjugacy condition, numerical comparisons

1. Introduction

For solving the unconstrained optimization problem

$$\min\left\{ f(x): x \in R^n \right\}, \tag{1}$$

where $f: R^n \to R$ is continuously differentiable, Dai and Yuan [7] suggested the following nonlinear conjugate gradient algorithm:

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

where the stepsize α_k is positive and the directions d_k are computed by the rule:

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

$$\beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T s_k}, \tag{4}$$

where $g_k = \nabla f(x_k)$ and $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$. Using a standard Wolfe line search, the Dai and Yuan method always generates descent directions and under Lipschitz assumption it is globally convergent. In [5] Dai established a remarkable property relating the descent directions to the sufficient descent condition, showing that if there exist constants γ_1 and γ_2 such that $\gamma_1 \le \|g_k\| \le \gamma_2$ for all k, then for any $p \in (0,1)$, there exists a constant $c > 0$ such that the sufficient descent condition $g_i^T d_i \le -c\|g_i\|^2$ holds for at least $\lfloor pk \rfloor$ indices $i \in [0,k]$, where $\lfloor j \rfloor$ denotes the largest integer $\le j$.

In this letter we present a modification of the Dai and Yuan computational scheme in order to satisfy both the sufficient descent condition and the conjugacy condition in the frame of conjugate gradient methods as:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^a s_k, \quad d_0 = -g_0, \tag{5}$$

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}}, \tag{6}$$

$$\beta_k^a = \frac{1}{y_k^T s_k}\left(g_{k+1} - \delta_k \frac{\|g_{k+1}\|^2}{y_k^T s_k} s_k \right)^T g_{k+1}, \tag{7}$$

$$\delta_k = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}}. \tag{8}$$

The method (5)-(8) is a method that belongs to the family of scaled conjugate gradient methods introduced by Birgin and Martínez [3]. Observe that if f is a quadratic function and α_k is selected to achieve the exact minimum of f in the direction d_k, then $s_k^T g_{k+1} = 0$ and the formula (7) for β_k^a reduces to the Dai and Yuan computational scheme (4). However, in this paper we consider general nonlinear functions and inexact line search.

In our algorithm the parameter β_k is selected in such a manner that the sufficient descent condition is satisfied every iteration. Besides, the parameters θ_{k+1} and δ_k are chosen that the conjugacy condition $y_k^T d_{k+1} = 0$ always holds, independent of the line search.

Theorem 1. *If* $y_k^T s_k \neq 0$ *and* $d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^a s_k$, ($d_0 = -g_0$ *), where* β_k^a *is given by (7), then*

$$g_{k+1}^T d_{k+1} \leq -\left(\theta_{k+1} - \frac{1}{4\delta_k}\right)\|g_{k+1}\|^2. \tag{9}$$

Proof. Since $d_0 = -g_0$, we have $g_0^T d_0 = -\|g_0\|^2$, which satisfy (9). Multiplying (5) by g_{k+1}^T, we have

$$g_{k+1}^T d_{k+1} = -\theta_{k+1}\|g_{k+1}\|^2 + \frac{(g_{k+1}^T g_{k+1})(g_{k+1}^T s_k)}{y_k^T s_k} - \delta_k \frac{\|g_{k+1}\|^2 (s_k^T g_{k+1})^2}{(y_k^T s_k)^2}. \tag{10}$$

But

$$\begin{aligned} \frac{(g_{k+1}^T g_{k+1})(g_{k+1}^T s_k)}{y_k^T s_k} &= \frac{\left[(y_k^T s_k) g_{k+1} / \sqrt{2\delta_k}\right]^T \left[\sqrt{2\delta_k}\,(g_{k+1}^T s_k) g_{k+1}\right]}{(y_k^T s_k)^2} \\ &\leq \frac{\frac{1}{2}\left[\frac{1}{2\delta_k}(y_k^T s_k)^2 \|g_{k+1}\|^2 + 2\delta_k (g_{k+1}^T s_k)^2 \|g_{k+1}\|^2\right]}{(y_k^T s_k)^2} \\ &= \frac{1}{4\delta_k}\|g_{k+1}\|^2 + \delta_k \frac{(g_{k+1}^T s_k)^2 \|g_{k+1}\|^2}{(y_k^T s_k)^2}. \end{aligned} \tag{11}$$

Using (11) in (10) we get (9). ■

Hence, the direction given by (5) and (7) is a descent direction. Dai and Yuan [7,8] present conjugate gradient schemes with the property that $g_k^T d_k < 0$ when $y_k^T s_k > 0$. If f is strongly convex or the line search satisfies the Wolfe conditions, then $y_k^T s_k > 0$ and the Dai and Yuan scheme yield descent. In our algorithm observe that, if for all k, θ_{k+1} is positive and $\theta_{k+1} > 1/4\delta_k$, and the line search satisfies the Wolfe conditions, then for all k the search direction (5) and (7) satisfy the sufficient descent condition. Note that in (9) we bound $g_{k+1}^T d_{k+1}$ by $-(\theta_{k+1} - 1/4\delta_k)\|g_{k+1}\|^2$, while for scheme of Dai and Yuan only the nonegativity of $g_{k+1}^T d_{k+1}$ is established.

To determine the parameters θ_{k+1} and δ_k observe that

$$d_{k+1} = -Q_{k+1} g_{k+1}, \tag{12}$$

where

$$Q_{k+1} = \theta_{k+1} I - \frac{s_k g_{k+1}^T}{y_k^T s_k} + \delta_k \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2}(s_k s_k^T). \tag{13}$$

Now, by symmetrization of Q_{k+1} as

$$\overline{Q}_{k+1} = \theta_{k+1} I - \frac{s_k g_{k+1}^T + g_{k+1} s_k^T}{y_k^T s_k} + \delta_k \frac{\|g_{k+1}\|^2}{(y_k^T s_k)^2}(s_k s_k^T) \tag{14}$$

and considering the conjugacy condition $y_k^T d_{k+1} = 0,$ i.e.

$$y_k^T \overline{Q}_{k+1} = 0, \tag{15}$$

after some algebra we get:

$$\theta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{y_k^T g_{k+1}} \text{ and } \delta_k = \frac{y_k^T g_{k+1}}{g_{k+1}^T g_{k+1}} = \frac{1}{\theta_{k+1}}. \tag{16}$$

From (16) observe that $\theta_{k+1} - 1/(4\delta_k) = (3/4)\theta_{k+1}$. Therefore, if for all k, $\theta_{k+1} \geq 0$, i.e. if $g_{k+1}^T y_k > 0$, then for all k the search direction d_{k+1} given by (5) and (7) with (16) satisfy the sufficient descent condition.

2. CGSD Algorithm

Considering the definitions of g_k, s_k and y_k we present the following Conjugate Gradient with Sufficient Descent condition:

Step 1. Initialization. Select $x_0 \in R^n$ and the parameters $0 < \sigma_1 < \sigma_2 < 1$. Compute $f(x_0)$ and g_0. Consider $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$. Set $k = 0$.

Step 2. Test for continuation of iterations. If $\|g_k\|_\infty \leq 10^{-6}$, then stop, else set $k = k+1$.

Step 3. Line search. Compute α_k satisfying the Wolfe line search conditions

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \sigma_1 \alpha_k g_k^T d_k, \tag{17}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g_k^T d_k, \tag{18}$$

and update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1})$, g_{k+1}, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

Step 4. Direction computation. Compute $d = -\theta_{k+1} g_{k+1} + \beta_k^a s_k$, where θ_{k+1}, δ_k and β_k^a are computed as in (16) and (7) respectively. If

$$g_{k+1}^T d \leq -10^{-3} \|d\|_2 \|g_{k+1}\|_2, \tag{19}$$

then define $d_{k+1} = d$, otherwise set $d_{k+1} = -g_{k+1}$. Compute the initial guess $\alpha_k = \alpha_{k-1}\|d_{k-1}\|/\|d_k\|$, set $k = k+1$ and continue with step 2. ■

The first trial for the steplength α_k in the line search is the same considered by Shanno and Phua [12] and Birgin and Martínez [3]. It is well known that if f is bounded along the direction d_k, there exists a stepsize α_k satisfying the Wolfe line search conditions (17) and (18). We used the same restarting procedure used by Birgin and Martínez [3], i.e. when the angle between d and $-g_{k+1}$ is not acute enough, then we restart the algorithm with the negative gradient $-g_{k+1}$. Under reasonable assumptions, conditions (17), (18), i.e. the Wolfe conditions, and (19) are sufficient to prove the global convergence of the algorithm (see for example [11]). However, we consider this aspect in the next section.

3. Convergence analysis for general nonlinear functions

Theorem 2. *Suppose that for all $k \geq 0$ there exists the positive constants ω and Ω, such that $0 < \omega \leq \theta_k \leq \Omega$. If the level set $L = \{x \in R^n : f(x) \leq f(x_0)\}$ is bounded and the Lipschitz condition $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ holds, then for the computational scheme (5)-(8) with a line search satisfying the Wolfe conditions (17) and (18) , either $g_k = 0$ for some k or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{20}$$

Proof. Suppose that $g_k \neq 0$ for all k and $\liminf_{k\to\infty} \|g_k\| > 0$. Define $\gamma = \inf\{\|g_k\| : k \geq 0\}$. Since $g_k \neq 0$ it follows that $\gamma > 0$. By the Wolfe condition we have:

$$y_k^T s_k = (g_{k+1} - g_k)^T s_k \geq (\sigma_2 - 1) g_k^T s_k = -(1-\sigma_2) g_k^T s_k. \tag{21}$$

By Theorem 1,

$$g_k^T d_k \leq -\left(\theta_k - \frac{1}{4\delta_{k-1}}\right)\|g_k\|^2 = -\frac{3}{4}\theta_k \|g_k\|^2 \leq -\frac{3}{4}\omega\|g_k\|^2. \tag{22}$$

Therefore,

$$-g_k^T d_k \geq \frac{3}{4}\omega\gamma^2. \tag{23}$$

Combining (21) with (23) we get

$$y_k^T s_k \geq \frac{3}{4}(1-\sigma_2)\omega\alpha_k\gamma^2. \tag{24}$$

Observe that $g_{k+1}^T s_k = y_k^T s_k + g_k^T s_k < y_k^T s_k$. Then from (18) we have $g_{k+1}^T s_k \geq \sigma_2 g_k^T s_k = -\sigma_2 y_k^T s_k + \sigma_2 g_{k+1}^T s_k$. Since $\sigma_2 < 1$, we obtain

$$g_{k+1}^T s_k \geq \frac{-\sigma_2}{1-\sigma_2} y_k^T s_k.$$

Therefore,

$$\left|\frac{g_{k+1}^T s_k}{y_k^T s_k}\right| \leq \max\left\{1, \frac{\sigma_2}{1-\sigma_2}\right\}. \tag{25}$$

On the other hand $\|y_k\| = \|g_{k+1} - g_k\| \leq L\|s_k\|$. Hence

$$|\delta_k| = \frac{|y_k^T g_{k+1}|}{\|g_{k+1}\|^2} \leq \frac{\|y_k\|}{\|g_{k+1}\|} \leq \frac{L\|s_k\|}{\|g_{k+1}\|}.$$

With these we have:

$$\begin{aligned}|\beta_k^a| &\leq \frac{1}{|y_k^T s_k|}\left[\|g_{k+1}\|^2 + |\delta_k|\|g_{k+1}\|^2 \frac{|g_{k+1}^T s_k|}{|y_k^T s_k|}\right] \\ &\leq \frac{4}{3(1-\sigma_2)\omega\alpha_k\gamma^2}\left[\Gamma^2 + L\Gamma\|s_k\|\max\left\{1, \frac{\sigma_2}{1-\sigma_2}\right\}\right] = E + F\|s_k\| = E + FD,\end{aligned}$$

where

$$E = \frac{4\Gamma^2}{3(1-\sigma_2)\omega\alpha_k\gamma^2}, \qquad F = \frac{4L\Gamma}{3(1-\sigma_2)\omega\alpha_k\gamma^2}\max\left\{1, \frac{\sigma_2}{1-\sigma_2}\right\},$$

$D = \max\{\|y - z\| : y, z \in L\}$ is the diameter of the level set L and $\Gamma = \max_{x\in L}\|\nabla f(x)\|$. Therefore,

$$\|d_{k+1}\| \leq |\theta_{k+1}|\|g_{k+1}\| + |\beta_k^a|\|s_k\| \leq \Omega\Gamma + (E + FD)D. \tag{26}$$

Now, from Lipschitz and Wolfe conditions we can prove that

$$\alpha_k \geq \frac{1-\sigma_2}{L}\frac{|g_k^T d_k|}{\|d_k\|^2}. \tag{27}$$

Since the level set L is bounded and the function f is bounded from below, then from (17) and (27) it follows that

$$\sum_{k=0}^{\infty}\frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{28}$$

Therefore, using (22), the descent property yields:

$$\sum_{k=0}^{\infty} \frac{\gamma^4}{\|d_k\|^2} \le \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \le \sum_{k=0}^{\infty} \frac{16}{9\omega^2} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty,$$

which contradicts (26). Hence, $\gamma = \liminf_{k\to\infty} \|g_k\| = 0.$ ■

4. Numerical results and comparisons

In this section we present the computational performance of a Fortran implementation of the CGSD algorithm on a set of 750 unconstrained optimization test problems. The Fortran 77 implementation of the present method is based on the Fortran 77 implementation of the SCG method [3] provided by the authors, as well as on the Fortran 77 implementation of SCALCG algorithm presented in [2]. We compare the performance of CGSD algorithm to the Dai and Yuan conjugate gradient algorithms. Dai [6] and Dai and Yuan [7,9] studied the hybrid conjugate gradient algorithms and proposed the following two hybrid methods:

$$\beta_k^{hDY} = \max\left\{ -\frac{1-\sigma_2}{1+\sigma_2} \beta_k^{DY}, \min\left\{ \beta_k^{HS}, \beta_k^{DY} \right\} \right\}, \tag{29}$$

and

$$\beta_k^{hDYz} = \max\left\{ 0, \min\left\{ \beta_k^{HS}, \beta_k^{DY} \right\} \right\}. \tag{30}$$

Therefore, we compare CGSD to DY, hDY and hDYz. All codes are written in double precision Fortran using the same style of programming and compiled with f77 (default compiler settings) on an Intel Pentium 4, 1.8GHz workstation.

The test problems are the unconstrained problems in the CUTE [4] library, along with other large-scale optimization problems presented in [1]. We selected 75 large-scale unconstrained optimization problems in extended or generalized form. For each function we have considered 10 numerical experiments with number of variables $n = 1000, 2000, \ldots, 10000.$

All algorithms implement the Wolfe line search conditions (17)-(18) with $\sigma_1 = 0.0001$ and $\sigma_2 = 0.9$, and the same stopping criterion $\|g_k\|_\infty \le 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector.

The numerical comparison follows the lines of the experiments performed in [2] and [3]. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2, for problem $i = 1, \ldots, 750,$ respectively. We say that, in the particular problem $i,$ the performance of ALG1 was better than the performance of ALG2 if: $\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3}$ and the number of iterations, or the number of function-gradient evaluations, 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.

In Figure 1, we consider CPU time to compare the performance of CGSD to that of DY, hDY and hDYz, by using the profiles of Dolan and Moré [10]. Figure 2 presents the performance of these algorithms and CONMIN by Shanno and Phua [12]. From Figure 1 we see that the best performance, relative to the CPU time metric, was obtained by CGSD and hDYz; hDYz being slightly more robust. However, as we see in Figure 2, the top performer is CONMIN, a BFGS preconditioned conjugate gradient algorithm.

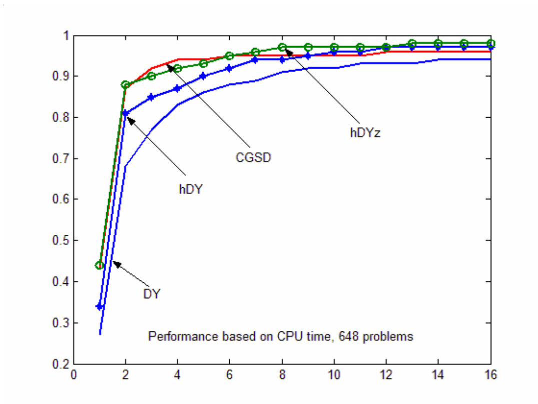

Fig. 1. Performance based on CPU time for CGSD versus DY, hDY and hDYz.

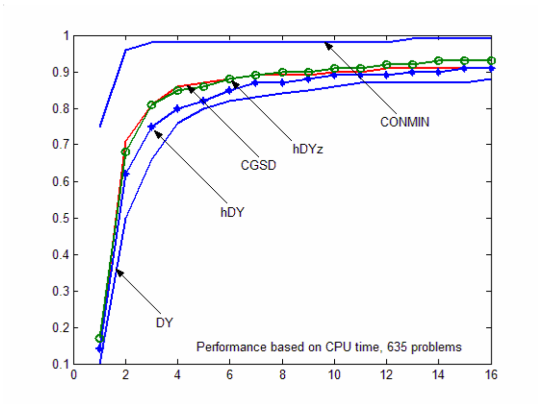

Fig. 2. Performance based on CPU time for CGSD versus DY, hDY, hDYz and CONMIN.

5. Conclusion

We have presented a new conjugate gradient algorithm for solving unconstrained optimization problems. The parameter β_k^a is a modification of the Dai and Yuan computational scheme in such a manner that the direction d_k generated by the algorithm satisfies both the sufficient descent condition and the conjugacy condition, independent of the line search. Under standard Wolfe line search conditions we proved the global convergence of the algorithm. The computational evidence showed that the performance of our algorithm CGSD was higher than those of the Dai and Yuan conjugate gradient algorithm and its hybrid variants, for a set consisting of 750 problems.

References

[1] N. Andrei, "*Test functions for unconstrained optimization*". http://www.ici.ro/camo/neculai/SCALCG/evalfg.for

[2] N. Andrei, "*A scaled nonlinear conjugate gradient algorithm for unconstrained optimization*", Optimization, to appear.

[3] F.G. Birgin and J.M. Martínez, "*A spectral conjugate gradient method for unconstrained optimization*", Applied Mathematics and Optimization 43, (2001), pp.117-128.

[4] I. Bongartz, A.R. Conn, N.I.M. Gould and P.L. Toint, "*CUTE: constrained and unconstrained testing environments*", ACM Trans. Math. Software, 21, pp.123-160, 1995.

[5] Y.H. Dai, "*New properties of a nonlinear conjugate gradient method*". Numer. Math., 89 (2001), pp.83-98.

[6] Y.H. Dai, "*A family of hybrid conjugate gradient methods for unconstrained optimization*", Mathematics of Computation, 72 (2003), 1317-1328.

[7] Y.H. Dai and Y, Yuan, "*A nonlinear conjugate gradient method with a strong global convergence property*", SIAM J. Optimization, 10 (1999), pp.177-182.

[8] Y.H. Dai and Y. Yuan, "*An efficient hybrid conjugate gradient method for unconstrained optimization*". Annals of Operations Research, 103 (2001), pp.33-47.

[9] Y.H., Dai and Y. Yuan, "*A class of globally convergent conjugate gradient methods*", Science in China (Series A), 46 (2003), pp.251-262.

[10] E.D. Dolan and J.J. Moré, "*Benchmarking optimization software with performance profiles*", Math. Programming, 91 (2002), pp. 201-213.

[11] R. Fletcher, "*A new variational result for quasi-Newton formulae*", University of Dundee Report NA/119, 1989.

[12] D.F. Shanno, and K.H. Phua, "*Algorithm 500, Minimization of unconstrained multivariate functions*", ACM Trans. On Math. Soft., 2 (1976), pp.87-94.

February 5, 2007