خوارزمية التدرج المترافق ذواتَ الحدود الثلاثة أستناداً الى طريقتي

Dai-Liao وPowell Symmetric

أ.د. خليل خضر عبو   اينور جودت نامق

قسم الرياضيات

كلية علوم الحاسوب والرياضيات

جامعة الموصل

الملخص

تم تطوير خوارزمية التدرج Powell symmetric وDai-Liao   استناداً الى خوارزميات المترافق ذوات الحدود الثلاثة لحل مسائل الامثلية غير المقيدة ذات القياس العالي. الطريقة المقترحة تحقق كلاً من شرط الآنحدار وشرط الترافق. وبفرض بعض الفرضيات القياسية على الدالة المحدبة بانتظام تم أثبات التقارب المطلق للخوارزمية. وأخيراً تم إعطاء بعض النتائج العددية للطريقة المقترحة.

كلمات دالة: الامثلية غير المقيدة, طرق الآنحدار، طرق التدرج المترافق.

Three-terms conjugate gradient algorithm based on the Dai-Liao and the Powell symmetric methods

Prof. Khalil K. Abbo   **Aynur J. Namik**

Department of Mathematics

College of computers sciences and mathematics

Mosul University

kh_196538@yahoo.com   aynoor.Javdet@gmail.com

Abstract:

Based on the Dai-Laio and Powell symmetric methods, we developed a new three – term conjugate gradient method for solving large-scale unconstrained optimization problem. The suggested method satisfies both the descent condition and the conjugacy condition. For uniformly convex function, under standard assumption the global convergence of the algorithm is proved. Finally, some numerical results of the proposed method are given.

Keywords: Unconstrained optimization, descent methods, Conjugate gradient methods.

1. Introduction

Conjugate Gradient (CG) method comprise a class of unconstrained optimization algorithms characterized by low memory requirements and strong global convergence properties [3] which made them popular for engineers and mathematicians engaged in solving large-scale problems in the following form:

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

Where $f : R^n \to R$ is a smooth nonlinear function and its gradient is available. The iterative formula of a CG method is given by

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \qquad k = 1,2,..., \tag{2}$$

in which α_k is a step-length to be computed by a line search procedure and d_k is the search direction defined by

$$d_1 = -g_1, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \qquad k = 1,2,..., \tag{3}$$

where $g_k = \nabla f(x_k)$ and β_k is a parameter called the conjugacy condition . The step-length α_k is usually chosen to satisfy certain line search conditions [15].

For general nonlinear functions, different choices of β_k lead to different conjugate gradient methods. Well-known formulas for β_k are

called the Fletcher-Reeves (FR) [7], Hestenes -Stiefel (HS) [8], and Polak-Ribiere (PR) [12]. are given by

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \qquad \beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \qquad \beta_k^{PR} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}$$

where $y_k = g_{k+1} - g_k$ and $\| \, . \, \|$ denotes to ℓ_2 norm.

The line search in conjugate gradient algorithms is often based on the standard Wolfe Conditions (WC) [16]:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho \alpha_k g_k^T d_k \ , \qquad (4)$$
$$g_{k+1}^T d_k \ge \sigma g_k^T d_k , \qquad (5)$$

Where d_k is a descent direction and $0 < \rho \le \sigma < 1$. However, for some conjugate gradient algorithms, a Stronger version of the Wolfe line search Conditions (SWC) given by (4) and

$$\left| g_{k+1}^T d_k \right| \le -\sigma g_k^T d_k \ , \qquad (6)$$

Is needed to ensure the convergence and to enhance the stability.

The pure conjugacy condition is represented by[11] the form

$$d_{k+1}^T y_k = 0 \qquad (7)$$

for nonlinear conjugate gradient methods. The extension of the conjugacy condition was studied by Perry [11]. He tried to accelerate the conjugate gradient method by incorporating the second-order information into it. Specifically, he used the secant condition

$$H_{k+1} y_k = s_k \qquad (8)$$

of quasi-Newton methods, where a symmetric matrix H_{k+1} is an approximation to the inverse Hessian. For quasi-Newton methods, the search direction d_{k+1} can be calculated in the form

$$d_{k+1} = -H_{k+1}g_{k+1} \qquad (9)$$

By (8) and (9), the relation

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

Holds. By taking this relation into account, Perry replaced the conjugacy condition (7) by the condition

$$d_{k+1}^T y_k = -g_{k+1}^T s_k. \qquad (10)$$

Dai and Liao [5] generalized the condition (10) to the following

$$d_{k+1}^T y_k = -tg_{k+1}^T s_k, \qquad (11)$$

where $t \geq 0$ is a scalar. The case $t = 0$, (11) reduces to the usual conjugacy condition (7). On the other hand, the case $t = 1$, (11) becomes Perry's condition (10). To ensure that the search direction d_k satisfies condition (11), by substituting $d_{k+1} = -g_{k+1} + \beta_k d_k$ into (11), they had

$$-g_{k+1}^T y_k + \beta_{k+1} d_k^T y_k = -tg_{k+1}^T s_k.$$

This gives the Dai-Liao formula

$$\beta_k^{DL} = \frac{g_{k+1}^T (y_k - ts_k)}{d_k^T y_k}. \qquad (12)$$

We note that the case $t = 1$ reduces to the Perry formula

$$\beta_k^P = \frac{g_{k+1}^T (y_k - s_k)}{d_k^T y_k}. \qquad (13)$$

Furthermore, if $t = 0$, then β^{DL} reduces to the β^{HS}. The approach of Dai and Liao (DL) has been paid special attention to by many researches. In

several efforts, modified secant equations have been applied to make modifications on the DL method. It is remarkable that numerical performance of the DL method is very dependent on the parameter *t* for which there is no any optimal choice [2] .

This paper is organized as follows. In section 2 we briefly review the Three-terms conjugate gradient methods. In section 3, the proposed algorithm is stated. The properties and convergent results of the new method are given in in Section 4. Numerical results and conclusion are presented in Section 5 and in Section 6, respectively.

2. Three-terms conjugate gradient (CG) methods

Recently many researchers have been studied three- term conjugate gradient methods. For example Narushima, Yab and Ford [10] have proposed a wider class of three term conjugate gradient methods (called 3TCG) which always satisfy the sufficient descent condition. Shanno in [14] used the well-known BFGS quasi-Newton method to obtain the following three-term CG method.

$$d_{k+1} = -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \left(1 + \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{g_{k+1}^T s_k}{s_k^T y_k} \right] s_k + \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k \qquad (14)$$

Furthermore, Liu and Xu in [9] was generalized the Perry conjugate gradient algorithm (13), the search directions were formulated as follows

$$d_{k+1}^{ps} = -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} \right) \frac{g_{k+1}^T s_k}{s_k^T y_k} \right] s_k + \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k \qquad (15)$$

Where τ_k is parameter, which is symmetric Perry three-terms conjugate gradient methods. When $\tau_k s_k^T y_k > 0$, the search directions defined by (15) satisfy the descent property

$$d_{k+1}^T g_{k+1} < 0$$

Or the sufficient descent property

$$d_{k+1}^T g_{k+1} \leq -c_0 \|g_{k+1}\|^2 , \qquad c_0 > 0 \tag{16}$$

Notice that if $\tau_k = 1$, then (15) reduces to the (14). It is remarkable that there is no any optimal choice for τ_k, However different values used for τ_k in [4], for example

$$\tau_k = 1, \quad \tau_k = c_1 \frac{y_k^T y_k}{s_k^T s_k}$$

3. A modifying three-terms conjugate gradient (CG) method

The aim of this section is to develop a modified three-terms conjugate gradient method named (AKTCG say) by using Powell Symmetric (PS) method (15) and Dai and Liao (DL) CG method (3) and (12). consider the search direction given by Dai and Liao

$$d_{k+1}^{DL} = -g_{k+1} + \frac{y_k^T g_{k+1}}{s_k^T y_k} s_k - t \frac{s_k^T g_{k+1}}{s_k^T y_k} s_k , \tag{17}$$

Letting $t = \frac{s_k^T s_k}{s_k^T y_k}$ in equation (17) we get

$$d_{k+1} = -g_{k+1} + \frac{y_k^T g_{k+1}}{s_k^T y_k} s_k - \frac{s_k^T s_k}{s_k^T y_k} \frac{s_k^T g_{k+1}}{s_k^T y_k} s_k \tag{18}$$

Now equating the equations (15) and (18) i.e

$$d_{k+1} = d_{k+1}^{PS} \ .$$

With simple algebra and with the change signal of the last term in d_{k+1}^{PS} we get

$$\tau_k = \frac{\|s_k\|^2 - \|y_k\|^2}{s_k^T y_k} \tag{19}$$

Substitute (19) in the equation (15) to obtain the new search direction

$$d_{k+1}^{AKTCG} = -g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \frac{\|s_k\|^2}{\left(s_k^T y_k\right)^2} \frac{s_k^T g_{k+1}}{} \right] s_k - \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k \tag{20}$$

Note that, if line search is exact i.e $g_{k+1}^T s_k = 0$ then the search direction d^{AKTCG} reduces to the well-known Hestenes and Stiefel β^{HS}, furthermore if $g_{k+1}^T s_k = 0$ and successive gradients are orthogonal i.e $g_{k+1}^T g_k = 0$ then d^{AKTCG} reduces to the CD-Fletcher method defined by $\beta_k^{CD} = \frac{g_{k+1}^T g_{k+1}}{s_k^T g_k}$

In the following we summarize the our AKTCG algorithm.

Algorithm (AKTCG)

Step (1): Select a starting point $x_1 \in dom\ f$ and $\varepsilon > 0$, compute $f_1 = f(x_1)$ and $g_1 = \nabla f(x_1)$. Select some positive values for $0 < \rho < \sigma < 1$. Set number of iteration $k = 1$ and $d_k = -g_k$.

Step (2): Test for convergence. If $\|g_k\|_\infty \leq \varepsilon$, then stop with x_k is optimal; otherwise go to step (3).

Step (3): Determine the step length α_k, by using the Wolfe line search conditions (4)-(5).

Step (4): 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 (5): Compute the search direction as: If $y_k^T s_k \neq 0$ then $d_{k+1} = d_{k+1}^{AKTCG}$ else $d_{k+1} = -g_{k+1}$.

Step (6): Set $k = k + 1$ and go to step 2.

4. Convergence analysis

Assume the following.

1. The level set $S = \{x \in R^n : f(x) \le f(x_0)\}$ is bounded, i.e. there exists positive constant $B > 0$ such that, for all $x \in S$, $\|x\| \le B$.

2. 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 $\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|$, for all $x, y \in N$.

Under these assumptions on f, there exists a constant $\Gamma \ge 0$ such that $\|\nabla f(x)\| \le \Gamma$, for all $x \in S$. Observe that the assumption that the function f is bounded below is weaker than the usual assumption that the level set is bounded. Although the search directions generated by (20) 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. Based on the above assumptions we shall show that our method satisfies the conjugacy condition, the sufficient descent condition, and globally convergent with Wolfe line search conditions. In the following (theorems 1,2) we will prove that our algorithm satisfies the sufficient descent condition and conjugacy condition.

Theorem(1): Suppose that the step-size α_k satisfies the standard Wolfe conditions, consider the search directions d_k generated from (20) then the search directions d_{k+1} are conjugate for all k that is .

$$d_{k+1}^T y_k = -c_0 g_{k+1}^T s_k$$

Where c_0 positive constant.

Proof:

$$y_k^T d_{k+1}^{AKTCG} = -y_k^T g_{k+1} + \left[\frac{g_{k+1}^T y_k}{s_k^T y_k} - \frac{\|s_k\|^2}{(s_k^T y_k)^2} g_{k+1}^T s_k \right] y_k^T s_k - \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k^T y_k$$

$$= -y_k^T g_{k+1} + y_k^T g_{k+1} - \frac{\|s_k\|^2}{s_k^T y_k} s_k^T g_{k+1} - \frac{g_{k+1}^T s_k}{s_k^T y_k} y_k^T y_k$$

$$= -\left(\frac{\|s_k\|^2 + \|y_k\|^2}{s_k^T y_k} \right) g_{k+1}^T s_k$$

By Lipschtz condition we have

$$\frac{\|s_k\|^2 + \|y_k\|^2}{s_k^T y_k} \le \frac{(1+L)\|s\|^2}{s_k^T y_k} = c_0$$

Therefore $d_{k+1}^T y_k = -c_0 g_{k+1}^T s_k$.

Theorem(2): Suppose that the step-size α_k satisfies the standard Wolfe conditions (WC), consider the search directions d_k generated from (20) then the search directions d_{k+1} satisfies the sufficient descent condition $d_k^T g_k \le -c\|g_k\|^2$, for all k .

Proof: The proof is by induction.

If $k = 1 \implies d_1 = -g_1$, $\therefore d_1^T g_1 = -\|g_1\|^2$
know let $s_k^T g_k < -c\|g_k\|$ to proof for $k+1$, multiply (20) by g_{k+1}^T to get

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

$$= -\|g_{k+1}\|^2 - \frac{\|s_k\|^2 (g_{k+1}^T s_k)^2}{(s_k^T y_k)^2} * \frac{\|g_{k+1}\|^2}{\|g_{k+1}\|^2}$$

$$= -\left(1 + \frac{\|s_k\|^2}{\|g_{k+1}\|^2} \frac{(g_{k+1}^T s_k)^2}{(s_k^T y_k)^2} \right) \|g_{k+1}\|^2$$

By Couchy-Shwartiz inequality and Lipschitz condition we get

$$\frac{\|s_k\|^2 (s_k^T g_{k+1})^2}{\|g_{k+1}\|^2 (s_k^T y_k)^2} \le \frac{\|s_k\|^4 \|g_{k+1}\|^2}{\|g_{k+1}\|^2 (s_k^T y_k)^2} = \frac{\|s_k\|^4}{(s_k^T y_k)^2} \le \frac{1}{L^2}$$

Therefore $d_{k+1}^T g_{k+1} = -c \|g_{k+1}\|^2$ Where $c = 1 + \frac{1}{L^2} > 0$.

Proposition 1 ([16,17]). Suppose that d_k is a descent direction and that the gradient ∇f satisfies the Lipschitz condition $\|\nabla f(x) - \nabla f(x_k)\| \le L \|x - x_k\|$ 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 Wolfe conditions (4) and (5), then

$$\alpha_k \ge \frac{(1-\sigma)|g_k^T d_k|}{L \|d_k\|^2} . \tag{21}$$

Proposition 2 ([13]). Suppose that assumptions **1** and **2** hold. Consider the algorithm (2) and (20), where d_k is a descent direction and α_k is computed by the general Wolfe line search (4) and (5). Then

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty . \tag{22}$$

.

Proposition 3 ([17]). Suppose that assumptions **1** and **2** hold, and consider any conjugate gradient algorithm (2), where d_k is a descent direction and α_k is obtained by the strong Wolfe line search (4) and (6).If

$$\sum_{k \ge 1} \frac{1}{\|d_k\|^2} = \infty , \tag{23}$$

Then $\lim\limits_{k\to\infty}\inf\|g_k\|=0$. (24)

For uniformly convex functions, we can prove that our (AKTCG) is globally convergent (theorem 3) .

Theorem (3): Suppose that assumptions **1** and **2** hold, and consider the algorithm (2) and (20), where d_k is a descent direction and α_k is computed by the strong Wolfe line search (4) and (6). Suppose that f is a uniformly convex function on S, i.e. there exists a constant $\mu>0$ such that

$$\left(\nabla f(x)-\nabla f(y)\right)^T\left(x-y\right)\ge\mu\|x-y\|^2 \quad \text{for all} \quad x,y\in N;\ \text{then}\ \lim\limits_{k\to\infty}\|g_k\|=0.$$

Proof: The prove is by Contradiction.

$$|\beta_k|=\left|\frac{g_{k+1}^T y_k}{s_k^T y_k}-\frac{\|s_k\|^2}{\left(s_k^T y_k\right)^2}g_{k+1}^T s_k\right|\le\frac{\left|g_{k+1}^T y_k\right|}{\left|s_k^T y_k\right|}+\frac{\|s_k\|^2\left|g_{k+1}^T s_k\right|}{\left|s_k^T y_k\right|^2}$$

Since f is uniformly convex then $s_k^T y_k\ge\mu\|s_k\|^2$ where $\mu>0$.

$$\therefore|\beta_k|\le\frac{\|g_{k+1}\|\ \|y_k\|}{\mu\|s_k\|^2}+\frac{\|s_k\|^2\|s_k\|\ \|g_{k+1}\|}{\mu^2\|s_k\|^4}$$

$$\le\frac{\|g_{k+1}\|\ \|y_k\|}{\mu\|s_k\|^2}+\frac{\|g_{k+1}\|}{\mu^2\|s_k\|}$$

By assumption **2** and Lipschitz continuity, we have $\|y_k\|\le L\|s_k\|$ .we get

$$\left|\beta_k\right| \le \frac{\Gamma L}{\mu \|s_k\|} + \frac{\Gamma}{\mu^2 \|s_k\|} = \frac{\Gamma}{\mu}\left(L + \frac{1}{\mu}\right)\frac{1}{\|s_k\|}$$

$$\left|\eta_k\right| = \left|\frac{g_{k+1}^T s_k}{s_k^T y_k}\right| = \frac{\left|g_{k+1}^T s_k\right|}{\left|s_k^T y_k\right|} \le \frac{\|g_{k+1}\|\ \|s_k\|}{\mu \|s_k\|^2} \le \frac{\Gamma}{\mu \|s_k\|}$$

$$\therefore \|d_{k+1}\| \le \|g_{k+1}\| + \left|\beta_k\right|\ \|s_k\| + \left|\eta_k\right|\ \|y_k\|$$

$$\le \Gamma + \Gamma\left(\frac{L}{\mu} + \frac{1}{\mu^2}\right)\frac{1}{\|s_k\|}\ \|s_k\| + \left(\frac{\Gamma}{\mu \|s_k\|}\right)\ L\|s_k\|$$

$$\le \Gamma + \Gamma\left(\frac{L}{\mu} + \frac{1}{\mu^2}\right) + \frac{\Gamma L}{\mu}\ = \frac{\Gamma\left(\mu^2 + 2L\mu + 1\right)}{\mu^2}$$

$$\|d_{k+1}\| \le \frac{\Gamma b}{\mu^2}\ \text{then}\ \frac{1}{\|d_{k+1}\|} \ge \frac{\mu^2}{\Gamma b}$$

Taking the sum for both sides and considering $\|d_1\| = \|g_1\|^2 \ge \Gamma$

$$\sum_{k=0}^{\infty}\frac{1}{\|d_{k+1}\|^2} \ge \Gamma + \sum_{k=0}^{\infty}\frac{\mu^2}{\Gamma b} = \Gamma + \frac{\mu^2}{\Gamma b}\sum_{k=0}^{\infty}1 = \infty$$

Contradiction we have $\lim_{k\to\infty}\inf\|g_k\| = 0$

5. Numerical results and comparisons

In this section, we report some numerical results obtained with an implementation of the AKTCG algorithm. The code of the AKTCG

Algorithm is written in Fortran and compiled with f77 (default compiler settings), taken from N. Andrei web page. We selected 80

Large-scale unconstrained optimization test functions in the generalized or extended form presented in [1]. For each test

function, we undertook ten numerical experiments with the number of variables increasing as n=100, 200,…, 1000.

The algorithm implements the Wolfe line search conditions with $\rho = 0.0001$, $\sigma = 0.9$ and the same stopping criterion $\left\| g_k \right\|_\infty \leq 10^{-6}$, where $\|\cdot\|_\infty$ is the maximum absolute component of a vector. In all

the algorithms we considered in this numerical study the maximum number of iterations is limited to 1000.

The comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal values 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

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

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.

Figures (1), (2) and (3) shows the Dolan and Moré [6] (iterations (iter), function-gradient evaluations(fg) and CPU time) performance profile of AKTCG versus Dai-Liao(DL) and Powell symmetric (PS) conjugate gradient algorithms. In a performance profile plot, the top curve corresponds to the method that solved the most problems in a(iter) or (fg) or CPU time that was within a given factor of the best((iter) or (fg) or CPU time). The percentage of the test problems for which a method is the fastest is given on the left axis(ρ-axis) of the plot. The right side(x-axis) of the plot gives the percentage of the test problems that were

successfully solved by these algorithms, respectively. The right is a measure of the robustness of an algorithm. When comparing AKTCG with the DL and PS subject (iter, fg, CPU) as in figures(1), (2) and (3) we see that AKTCG is the top performer.
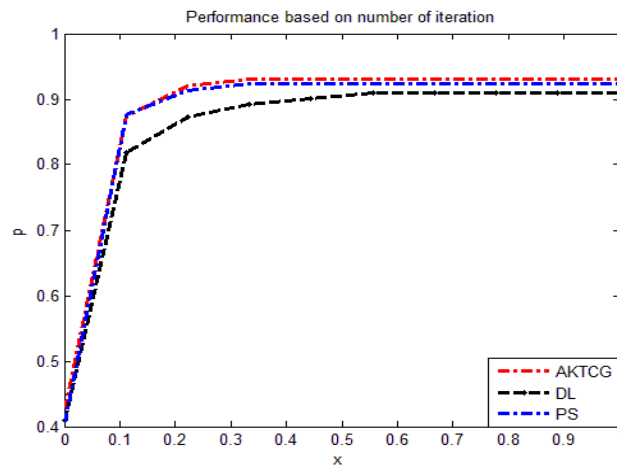

Figure (1)Performance based on iteration
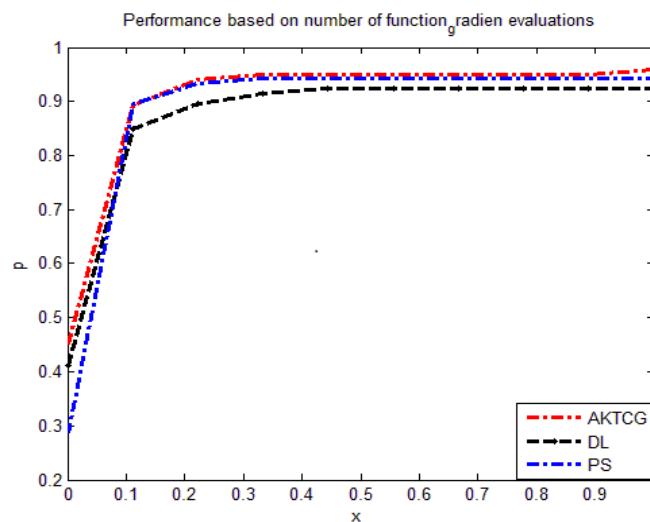

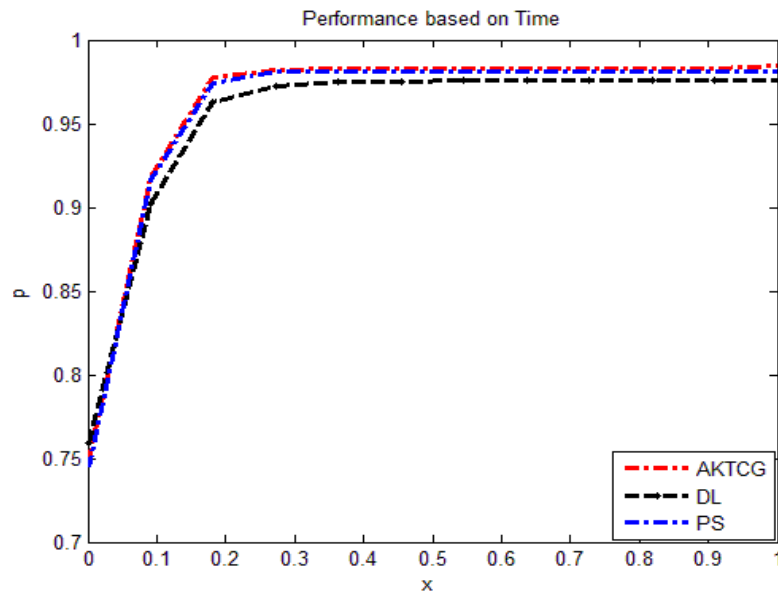Figure (2) Performance based on Function gradient evaluation

Figure (3)Performance based on Time

Table (1) shows the comparison of the algorithms AKTCG, PS and DL with respect to the total number of iterations (iter) , toal number of function gradient evaluations (fg) and total time for solving 800 test problems.

Algorithm	iter	fg	time
AKTCG	134276	226656	2342 s
PS	134652	228412	2894 s
DL	137538	231532	3346 s

6. Conclusion

In this paper, we have proposed a three –term conjugate gradient method based on the DL and PS methods which generates sufficient descent and conjugate directions. Our method have been shown to converge globally. In numerical experiments, we have confirmed the effectiveness of the proposed method by using performance profile.

References

[1]   N. Andrei, (2008). "An Unconstrained Optimization test function collection". Adv. Model. Optimization. 10. .

[2]   N. Andrei, (2011). Open Problems in Nonlinear Conjugate Gradient Algorithms for Unconstrained Optimization. BULLETIN of the Malaysian Mathematical Sciences Society. 34, 319–330.

[3]   Y. Dai, J. Han, G. Liu, D. Sun, H.Yin, & Y. Yuan, (1999). Convergence properties of nonlinear conjugate gradient methods. SIAM Journal on Optimization, 10, 348-358.

[4]   Y.H. Dai and C.X. Kou, (2013). A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search, SIAM J. Optim., Vol. 23, No. 1, PP. 296-320.

[5]   Y. Dai and L. Liao, (2001). New Conjugacy Conditions and Related Nonlinear Conjugate Gradient Methods, Applied Mathematics and Optimization, Springer-Verlag, New York, USA, 43, PP. 87-101.

[6]   E.D. Dolan and J.J. Mor´e, "Benchmarking optimization software with performance profiles", Math. Programming, 91 (2002), pp. 201-213.

[7]   R. Fletcher and C.M. Reeves, (1964). Function Minimization by Conjugate Gradients. Computer Journal, 7, PP. 149-154.

[8]   M.R. Hestenes and E. Stiefel, (1952). Methods of Conjugate Gradients for Solving Linear Systems, Journal of Research of the National Bureau of Standards, Vol.(5), No.(49).

[9]   D. Liu and G. Xu, (2013). Symmetric Perry conjugate gradient method. Comput. Optim Appl. (56).

[10]   Y. Narushima, H. Yabe and J.A. Ford, A three-term conjugate gradient method with sufficient descent property for unconstrained optimization, SIAM Journal on Optimization, 21(2011), 212-230.

[11]   A. Perry, (1978). A Modified Conjugate Gradient Algorithms. Operations Research, 26, PP. 1073–1078.

[12]   E. Polak and G. Ribiére, (1969). Note Sur la Convergence de Directions Conjuguée. Revue Francaise Information, Recherche. Operationnelle, (16), pp.35-43.

[13]   M.J.D. Powell, (1984). Nonconvex Minimization Calculations and the Conjugate Gradient Method. in : Numerical Analysis (Dundee,1983), In Lecture Notes in mathematics, Springer-Verlag, Berlin, 1066, PP. 122-141.

[14] D.F. Shanno, (1978). Conjugate gradient methods with inexact searches, Mathematics of Operations Research 3, PP. 244-256.

[15] W. Sun and Y. Yuan, (2006). Optimization Theory and Methods, Nonlinear programming, Springer Science, Business Media, LLC., New York.

[16] M. Wolfe (1978)."Numerical Methods For Unconstrained Optimization An Introduction". New York: Van Nostraned Reinhold

[17] G. Zoutendijk, (1970). Nonlinear Programming, Computational Methods. Integer and Nonlinear Programming (J. Abadie ED.). North-Holland, Amsterdam, PP. 37-86.