A MODIFIED DAI-YUAN CONJUGATE GRADIENT METHODS AND ITS GLOBAL CONVERGENCE

***Huda I. Ahmed , **Ghada M. Al-Naemi**
Dhuda2012@yahoo.com, ghalnaemi@yahoo.com
*Department of Operation Research and Intelligent Techniques, College of Computer Science and Mathematics, University of Mosul . Mosul -Iraq
**Department of Mathematics, College of Computer Science and Mathematics, University of Mosul . Mosul -Iraq

Abstract

Based on the conjugacy condition often which is satisfy by quasi-Newton method, the new version of DY nonlinear conjugate gradient method is proposed, which is descent methods even with inexact line searches. The search direction of the proposed method has the form $d_{k+1} = -\theta_{k+1} g_{k+1} \beta_k d_k$. When exact line search is used, the proposed method reduce to the standard DY method. Convergence properties of the proposed method is discussed. Numerical results are reported.

Key Words: Conjugate gradient algorithm, DY-Algorithm, descent direction, global convergence.

***هدى عصام أحمد، **غادة مؤيد رشيد**

*قسم بحوث العمليات والتقنيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل. الموصل– العراق

**قسم الرياضيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل. الموصل– العراق

الخلاصة

إن قاعدة شرط الترافق عادة تتحقق بواسطة أشباه نيوتن ،النسخة الجديدة المعدلة لطريقة Dai and Yuan (DY) للتدرج المترافق للدوال غير الخطية قد اقترحت في هذا البحث. هذه الطريقة تحقق خاصية الانحدار حتى إذا استخدمنا طريقة بحث غير المضبوط.

هذه الطريقة المقترحة تعرف بالشكل الآتي $d_{k+1} = -\theta_{k+1} g_{k+1} \beta_k d_k$. و عندما يكون خط البحث المستخدم مضبوط فأن الطريقة المقترحة تعود إلى الصيغة العامة لطريقة DY.

في هذا البحث قمنا بدراسة خواص التقارب الشمولي ، ووضعت المبرهنات الخاصة بها التي تعزز هذه الخواص.

1-Introduction:

We concerned with the unconstrained minimization problem

Minimize f(x), $x \in R^n$ (1)

where $f : R^n \to R$ is smooth and its gradient $g(x) = \nabla f(x)$ is available. There are several kinds of numerical methods for solving [1], which include for example, the steepest descent, the Newton method and quasi-Newton method. Among them, the conjugate gradient method is one choice for solving large scale problems, because it does not need any matrices [1,2,3]

Conjugate gradient methods are iterative methods of the form given by:

$$x_{k+1} = x_k + \alpha_k d_k , \quad (2)$$

where α_k >0 is a positive step size and d_k is a search direction. The search direction are usually defined by:

$$d_{k+1} = \begin{cases} -g_{k+1} & if\ k=0 \\ -g_{k+1}+\beta_k d_k, & if\ k\geq 1 \end{cases} \quad (3)$$

where $g_{k+1}=\nabla f(x_{k+1})$ and $\beta_k \in R$ is scalar parameter which characterizes conjugate gradient methods [4].

Usually the parameter β_k is chosen so that [2,3] reduces to linear conjugate gradient method if f(x) is strictly convex quadratic function and if α_k is calculated by the exact line search. Several kind of formulas for β_k has been proposed, for example, the Hestens-Stiefel (HS), Fletcher-Reeves (FR), Polak-Ribie're (PR), Liu-Storey (LS) and Dai-Yaun (DY) formula [5], [6], [7], [8], [9]& [10].

$$\beta_k^{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \quad (4)$$

$$\beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (5)$$

$$\beta_k^{PR} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} \quad (6)$$

$$\beta_k^{LS} = -\frac{g_{k+1}^T y_k}{d_k^T g_k} \quad (7)$$

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{d_k^T y_k} \quad (8)$$

where $y_k = g_{k+1}-g_k$ and $\|.\|$ denote the Euclidean norm. If f(x) is strictly convex quadratic function

$$f(x)=\frac{1}{2}x^T Gx + b^T x \quad (9)$$

where $G\in R^{nxn}$ is a symmetric positive definite matrix, and if α_k is the exact one-dimensional minimizer then the method (2) and (3) is called the linear conjugate method, within the framework of linear conjugate gradient methods, the conjugacy condition is defined by

$$d_i^T G d_j = 0,\ i\neq j \quad (10)$$

for search direction, and this condition guarantees the finite termination of the linear conjugate gradient methods. On the other hand, the method (2) and (3) is called the nonlinear conjugate gradient method for several unconstrained optimization problem. The conjugacy condition is replaced by

$$d_{k+1}^T y_k = 0 \quad (11)$$

for search direction, because the relations:

$$\begin{aligned} d_{k+1}^T G d_k &= \frac{1}{\alpha_k} d_{k+1}^T G(x_{k+1}-x_k) \\ &= \frac{1}{\alpha_k} d_{k+1}^T (g_{k+1}-g_k) \\ &= \frac{1}{\alpha_k} d_{k+1}^T y_k \end{aligned} \quad (12)$$

holds for strictly convex quadratic objective function. The extension of the conjugacy condition was studied by Perry [11] and also Shanno [15].

However, both the conjugacy condition (10) and (11) depend on exact line search. In particular computation, one normally carries on inexact line search instead of exact line search. In the case when $g_{k+1}^T d_k \neq 0$, the conjugacy condition (10) and (11) may have some disadvantages, for this season the extension of the conjugacy condition studied by Perry [11]. He tried to accelerate the conjugate gradient method by incorporating the second-order information into it, specifically, he used the quasi-Newton method, when the search direction d_k can be calculated in the form:

$$d_{k+1} = -H_{k+1} g_{k+1} \quad (13)$$

where H_{k+1} is nxn symmetric and positive definite and with quasi-Newton condition defined by:

$$H_{k+1} y_k = s_k \quad (14)$$

where $s_k = x_{k+1}-x_k = \alpha_k d_k$. For quasi-Newton method, by (13) and (14), we have that

$$\begin{aligned} d_{k+1}^T y_k &= -(H_{k+1}g_{k+1})^T y_k \\ &= -g_{k+1}^T (H_{k+1} y_k) \\ &= -g_{k+1}^T s_k \end{aligned} \quad (15)$$

equation (15) is called Perry's condition, which implies (11) hold if the line search is exact, since in this case $g_{k+1}^T s_k = 0$.

However, practical algorithms normally adopt inexact line searches instead of exact line searches. Recently Dai and Liao [13] replace the conjugacy condition (11) with the condition:

$$d_{k+1}^T y_k = -t g_{k+1}^T s_k \quad (16)$$

where $t \geq 0$ is a scalar. In the case t=0, (16) reduces to the usual conjugacy condition (11). On the other hand, in then case t=1, (16) becomes Perry's condition (15).

To establish convergence properties of any method, it is usually required that the step size α_k should satisfy the strong Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \quad (17)$$

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq -\sigma g_k^T d_k \qquad (18)$$

where $o < \delta < \sigma < 1$ [14]. On other hand, many other numerical methods (e.g. the steepest descent and quasi-Newton methods) for unconstrained optimization are proved to be convergent under the Wolfe conditions, which are weaker than the strong Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \quad (19)$$

$$g^T (x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \qquad (20)$$

thus it is an interesting to study global convergence of conjugate gradient methods under the Wolfe conditions instead of the strong Wolfe condition [15] &[16].
Besides conjugate gradient methods, the following gradient type methods

$$d_{k+1} = \begin{cases} -g_{k+1} & , if\ k = 0 \\ -\theta_{k+1} g_{k+1} + \beta_k d_k & , if\ k \geq 1 \end{cases} \quad (21)$$

have also been studied extensively by many authors, here θ_{k+1} and β_k are two parameters. Clearly, if $\theta_{k+1} = 1$, the methods (21) becomes conjugate gradient method (3). Zhang et al [17] proposed a modified FR-method where the parameter in(21) are given by:

$$\theta_{k+1} = \frac{d_k^T y_k}{\|g_k\|^2} \text{ and } \beta_k = \beta_k^{FR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}$$

This method satisfied $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2$. Moreover, this method convergences globally for general function with Armijo or Wolfe line search. Brigin and Martinez [4] proposed a special conjugate gradient method by combining conjugate gradient method and spectral gradient method [18] in the following way

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k d_k ,$$

where θ_{k+1} and β_k is parameter and

$$\beta_k = \frac{(\theta_{k+1} y_k - s_k)}{d_k^T y_k} \quad and \quad \theta_{k+1} = \frac{s_k^T s_k}{s_k^T y_k} ,$$

Andrei [2] proposed another conjugate gradient method where the search direction is selected as

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k^{DL} d_k ,$$

where

$$\beta_k^{DL} = \left(\frac{y_k^T g_{k+1}}{y_k^T s_k}\right) s_k - t\left(\frac{s_k^T g_{k+1}}{y_k^T s_k}\right) s_k .$$

In this paper, we are concerned with the methods (21) with the parameter $\beta_{k+1} = \beta_{k+1}^{DY}$, because Dai and Yaun method always generates descant direction and under Lipschitz assumption its globally convergent. Then we try to construct new θ_{k+1} by using the idea of DY method [10].
This paper is organized as follows. In Section 2, we present new formulas for θ_{k+1} corresponding algorithms, and prove a descent search direction. In section 3, we analyze global properties of the proposed method with inexact line searches, In Section 4, we report numerical comparison with existing conjugate gradient methods.

2-New Formula for θ_{k+1} and Algorithms

In this section we present a modified of the Dai and Yuan computational method, we describe the following two-terms DY conjugate gradient type method

$$d_{k+1} = \begin{cases} -g_{k+1} & , if\ k = 0 \\ -(1 + \theta_{k+1}^l) g_{k+1} + \beta_k^{DY} d_k & , if\ k \geq 1 \end{cases} (22)$$

where, for convenience, we write $\theta_{k+1} = 1 + \theta_{k+1}^l$, and θ_{k+1}^l is positive parameter.
In order to get the formula for θ_{k+1}^l in our method, multiply both sides of (22) by y_k.

$$d_{k+1}^T y_k = -g_{k+1}^T y_k - \theta_{k+1}^l g_{k+1}^T y_k + \beta_k^{DY} d_k^T y_k \ (23)$$

sub stituting (16) and (8) into (23), we have

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

so, we have

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

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

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

then

$$\theta_{k+1}^{l} = t\frac{g_{k+1}^{T} s_k}{g_{k+1}^{T} y_k} + \frac{\|g_{k+1}\|^2}{s_k^{T} y_k}\left(\frac{g_{k+1}^{T} s_k}{\|g_{k+1}\|^2}\right) - 1 \quad (24)$$

since, t is parameter, let t be defined by:

$$t = \frac{g_{k+1}^{T} y_k}{g_{k+1}^{T} s_k} - \rho\frac{g_{k+1}^{T} y_k}{s_k^{T} y_k} \qquad (25)$$

where ρ is constant and $\rho \in [0,1)$. Then, we have

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

since $s_k = \alpha_k d_k$

$$\theta_{k+1}^{l} = \beta_k^{DY}\frac{g_{k+1}^{T} d_k}{\|g_{k+1}\|^2} - \rho\frac{g_{k+1}^{T} d_k}{d_k^{T} y_k} \qquad (26)$$

then the new scalar θ_{k+1} is defined by

$$\theta_{k+1} = 1 + \theta_{k+1}^{l}$$
$$= 1 + \beta_k^{DY}\frac{g_{k+1}^{T} d_k}{\|g_{k+1}\|^2} - \rho\frac{g_{k+1}^{T} d_k}{d_k^{T} y_k} \quad (27)$$

For convenience, we summarize the above method as the following algorithm which we call the two-term DY method.

2.1 Algorithm of two-term DY method (modified DY method)

Step(0): Given $x_1 \in R^n$, $\varepsilon > 0$, set $d_1 = -g_1 = -\nabla f(x_1)$, if $\|g_1\| < \varepsilon$ then stop.

Step (1): Find $\alpha_k > 0$ satisfying the Wolfe condition (19) and (20)

Step(2): Let $x_{k+1} = x_k + \alpha_k d_k$ and $g_{k+1} = g(x_{k+1})$. If $\|g_{k+1}\| < \varepsilon$, then stop; Otherwise continue.

Step(3): Compute β_k, θ_{k+1} by the formula (8), (27), respectively and generate the new search direction d_{k+1} by (22).

Step(4): If k=n or $\|g_{k+1}^{T} g_k\| > 0.2\,\|g_{k+1}\|^2$ is satisfy, go to step (0), else k=k+1 and go to step (1).

Note, if exact line search is used, it is easily to see that the algorithm (2.1) reduce to the standard DY method.

2.2 The Sufficient Descent Condition.

In the global convergence analysis for many methods, the sufficient descant condition, namely for some constant c>0 (c is positive constant).

$$g_{k+1}^{T} d_{k+1} \le -c\|g_{k+1}\|^2 \qquad (28)$$

This condition has been used to analyze the global convergence of conjugate gradient Algorithms with inexact line searches [1]. The following result shows that algorithm (2.1) produces sufficient descent directions.

Lemma 2.2.1

Let $\{x_k\}$ and $\{d_k\}$ be generated by Algorithm (2.1), and let α_k be obtained by the Wolfe line search (19) and (20), if $\rho \in [0,1)$, then we have

$$\frac{g_{k+1}^{T} d_{k+1}}{\|g_{k+1}\|^2} \le -c \qquad (29)$$

where $c = [1 + \rho\sigma/(1-\sigma)]$.

Proof:

Note, that when $\rho = 0$ in (27) then (22), become:

$$d_{k+1}^{T} g_{k+1} = -\|g_{k+1}\|^2 - \beta_k^{DY} g_{k+1}^{T} d_k + \beta_k^{DY} g_{k+1}^{T} d_k$$
$$= -\|g_{k+1}\|^2.$$

For initial direction (k=1) we have :

$$d_1 = -g_1 \Rightarrow d_1^{T} g_1 = -\|g_1\|^2 < 0$$

Suppose

$$g_j^{T} d_j < 0, \quad \forall j = 1,2,\ldots,k;$$

We have from (22) and the definition of β^{DY} (8) that:

$$g_{k+1}^{T} d_{k+1} = -\|g_{k+1}\|^2 + \rho\frac{g_{k+1}^{T} d_k}{d_k^{T} y_k}\|g_{k+1}\|^2 \quad (30)$$

Which implies that:

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

$$= -(1-\rho) - \rho(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k})$$

since , $\left(1 - \frac{g_{k+1}^T d_k}{d_k^T y_k}\right) = -\frac{g_k^T d_k}{d_k^T y_k}$

$$\frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2} = -(1-\rho) + \rho \frac{g_k^T d_k}{d_k^T y_k}$$

from the Wolfe condition (20)

$$y_k^T d_k \geq -(1-\sigma) g_k^T d_k$$

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -(1-\rho) - \rho \frac{g_k^T d_k}{(1-\sigma) g_k^T d_k}$$

$$= -[1 + \rho\sigma/(1-\sigma)] \ .$$

Since $0 < \sigma < 1$ then let $c = [1 + \rho\sigma/(1-\sigma)]$ is positive constant i.e.

$$g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2 .$$

The proof is complete.

3- Convergence Analysis

For the global converge analysis of many methods, the following assumption is often needed.

Assumption (1):

i-The level $\Psi = \{x \in R^n \ / \ f(x) \leq f(x_0)\}$ is bounded.

ii- In some neighborhood Ω of ψ, f is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant L>o, such that

$\|g(x) - g(y)\| \leq L\|x - y\|, \forall x, y \in \Omega$ (31)

Clearly Assumption (1) implies that, their exists a constant $\gamma \geq 0$, such that

$$\|g(x)\| \leq \gamma, \forall x \in \Omega \qquad (32)$$

The following Lemma, called the Zountendijk condition is often used to prove global convergence of conjugate gradient methods holds and consider It was originally given by [19],[20].

Lemma 3.1

Suppose that Assumption (1) holds. Consider any iteration method of the form (1) and (2), where d_k satisfies descent direction $g_k^T d_k < 0$ and α_k is obtained by the Wolfe conditions or the strong Wolfe conditions, then the following holds

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty \qquad (33)$$

or equivalently

$$\sum_{k=0}^{\infty} \|g_k\|^2 \cos^2 \phi_k < \infty \qquad (34)$$

where ϕ_k is the angel between the search direction d_k and the steepest descent direction $-g_k$ [21].

Theorem 3.2

Suppose that Assumption (1) holds, if $\rho \in [0,1)$, where the sequence $\{x_k\}$ generated by Algorithm (2.1) with the Wolfe line search then (19)& (20) satisfies

$$\liminf_{k \to \infty} \|g_k\| = 0 \qquad (35)$$

Proof:

We will using the contradiction for prove theorem (3.2), i.e. if the theorem is not true, then $\|g_k\| \neq 0$, then there exists $\gamma > 0$, such that

$$\|g(x)\| > \gamma, \qquad \forall k \ .$$

Then, from (30)

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

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} + 1 = \rho \frac{g_{k+1}^T d_k}{d_k^T y_k} \qquad (36)$$

Use second Wolfe condition (20) and Lipschitz condition (32) for $d_k^T y_k \leq L d_k^T s_k$. Therefore

$$\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} + 1 \geq \rho \frac{g_k^T d_k}{L d_k^T s_k}$$

$$\geq \rho \frac{\sigma g_k^T d_k}{L \alpha_k \|d_k\|^2}$$

Then

$$\frac{L\alpha_k}{\rho\sigma} \left(\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1\right) \geq \frac{g_k^T d_k}{\|d_k\|^2} \qquad (37)$$

Take the squares of both sides to (37), we get

$$\left(\frac{L\alpha_k}{\rho\sigma}\right)^2 \left(\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2} + 1\right)^2 \geq \left(\frac{g_k^T d_k}{\|d_k\|^2}\right)^2$$

Since $(g_k^T d_k)^2 = \|g_k\|^2 \|d_k\|^2 \cos^2 \phi_k$

Then

$$\eta(\frac{d_{k+1}^T g_{k+1}}{\|g_{k+1}\|^2}+1)^2 \geq \|g_k\|^2 \cos^2\phi_k \geq \gamma^2 \cos^2\phi_k$$

where, $\eta = (\frac{L\alpha_k}{\rho\sigma})^2 \|d_{k-1}\|^2$.

Taking the summation from k=1 to k=∞, we get

$$\sum_{k\geq 1}^{\infty}\eta(\frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2}+1)^2 \geq \sum_{k\geq 1}^{\infty}\frac{(g_k^T d_k)^2}{\|d_k\|^2}$$

$$\geq \sum_{k\geq 1}^{\infty}\gamma^2 \cos^2\phi_k = \infty$$

Contradiction with Zountendijk theorem. Therefore $\liminf_{k\to\infty}\|g_k\| = 0$.

4- Numerical results

We testes the HS, FR, PR and DY formulas (4), (5), (6) and (8) respectively and our new method (modified of the DY method) (22), (27). All results are obtained using Pentium 4 workstation and all programs are written in Fortran Language. Our line search subroutine compute α_k such that the Wolfe condition (19) &(20) holds with $\delta = 0.001$, and $\rho = 0.9$ the value of α_k is always compute by cubic fitting procedure which was described in details by Bundy [22].

We have tested 14 nonlinear test functions with different dimensions n=1000,10000 and also higher dimension like n=100000, 1000000. The numerical results are given in the form of NOI and NOF (number of iterations and number of function evaluations). The stopping condition used was $|g_{k+1}| \leq 1\times 10^{-5}$.

Comparing the new method with HS, FR, PR and DY formulas we could say that the new method is more better than all especially for large dimensions.

In order to get relatively better ρ values in Algorithm (2.1), we choose 14 complex problems to test Algorithm (2.1) with different ρ values. (Table 1) lists these numerical results where NOF and NOI mean the number of function evaluations and the number of iterations.

In (Table 1), we see that Algorithm (2.1) with $\rho = 0.5$ performed best. Moreover, we also compared Algorithm (2.1) with other Algorithms in the previous sections and numerical results showed that they performed similarly. So in this section, we only listed the numerical results for Algorithms (2.1) with $\rho = 0.5$. These results are reported in (Table 2). The symbol * in (Table 1) and (table 2) means that the algorithm is unable to solve the particular problem.

Table 1: Test results for Algorithm(2.1) with different ρ values.

Test functions (n)	ρ	NOI	NOF	Test functions (n)	ρ	NOI	NOF
Powell (1000)	0.0	52	126	Wood (10000)	0.0	24	54
	0.1	30	68		0.1	56	135
	0.3	30	68		0.3	60	151
	0.5	28	63		0.5	33	96
	0.7	30	67		0.7	56	173
	0.9	28	62		0.9	40	122
	1.0	26	60		1.0	56	169
Rosen (100000)	0.0	27	69	Cubic (1000000)	0.0	16	44
	0.1	27	69		0.1	16	44
	0.3	27	69		0.3	16	44
	0.5	27	69		0.5	16	44
	0.7	29	76		0.7	16	45
	0.9	30	76		0.9	16	45
	1.0	30	76		1.0	16	45
Recipe (12000)	0.0	6	18	Shallow (1000000)	0.0	10	25
	0.1	6	18		0.1	10	25
	0.3	6	18		0.3	10	25
	0.5	6	18		0.5	10	25
	0.7	6	18		0.7	10	25
	0.9	6	18		0.9	10	25
	1.0	6	18		1.0	10	25
NOND (1000)	0.0	26	65	Strait (1000)	0.0	8	20
	0.1	26	65		0.1	8	20
	0.3	27	64		0.3	8	20
	0.5	27	64		0.5	8	20
	0.7	27	64		0.7	8	20
	0.9	27	65		0.9	8	20
	1.0	27	65		1.0	8	20
Wolfe !10000)	0.0	135	274	Sum (1000000)	0.0	80	270
	0.1	129	260		0.1	96	360
	0.3	127	256		0.3	112	394
	0.5	116	235		0.5	107	534
	0.7	118	240		0.7	102	385
	0.9	109	224		0.9	116	439
	1.0	120	242		1.0	132	470
Dixon (100000)	0.0	470	1025	Raydan (100)	0.0	45	49
	0.1	481	1044		0.1	45	49
	0.3	484	1048		0.3	45	49
	0.5	474	1017		0.5	45	49
	0.7	465	1025		0.7	45	49
	0.9	428	1046		0.9	45	49
	1.0	483	1060		1.0	45	49
Powell-3 (300000)	0.0	20	43	Quartc (1000000)	0.0	1	4
	0.1	20	43		0.1	1	4
	0.3	19	40		0.3	1	4
	0.5	18	39		0.5	1	4
	0.7	18	39		0.7	1	4
	0.9	16	35		0.9	1	4
	1.0	*	*		1.0	1	4

* the algorithm fail converge

Table 2: Comparison of different CG-algorithms with different test functions and different dimensions

Test functions	n	New algorithm $\rho = 0.5$ NOI(NOF)	FR algorithm NOI(NOF)	HS algorithm NOI(NOF)	DY algorithm NOI(NOF)	PR algorithm NOI(NOF)
Powell	1000	33 (96)	31 (92)	41 (109)	48 (138)	54 (164)
	10000	33 (96)	36 (110)	41 (109)	56 (169)	56 (168)
	100000	33 (96)	36 (110)	41 (109)	56 (196)	62 (203)
	1000000	33 (96)	36 (124)	41 (109)	63 (210)	68 (242
Wood	1000	28 (36)	27 (61)	30 (67)	27 (60)	29 (67)
	10000	28 (63)	29 (66)	33 (73)	26 (60)	29 (67)
	100000	29 (65)	29 (66)	33 (73)	27 (62)	29 (67)
	1000000	29 (65)	29 (66)	34 (75)	27 (62)	30 (69)
Rosen	1000	27 (69)	29 (76)	30 (76)	30 (76)	29 (76)
	10000	27 (69)	29 (76)	30 (76)	30 (76)	29 (76)
	100000	27 (69)	30 (78)	30 (76)	30 (76)	30 (78)
	1000000	27 (69)	30 (78)	30 (76)	30 (76)	30 (78)
Cubic	1000	16 (44)	15 (43)	16 (44)	15 (43)	16 (44)
	10000	16 (44)	16 (45)	16 (44)	15 (45)	16 (44)
	100000	16 (44)	16 (45)	16 (44)	16 (45)	16 (44)
	1000000	16 (44)	16 (45)	16 (44)	16 (45)	16 (44)
Recipe	1000	5 (16)	5 (16)	5 (16)	5 (16)	5 (16)
	10000	6 (18)	6 (18)	6 (18)	6 (18)	6 (18)
	100000	6 (18)	6 (18)	6 (18)	6 (18)	6 (18)
	1000000	6 (18)	6 (18)	6 (18)	6 (18)	6 (18)
NOND	1000	26 (64)	30 (78)	27 (65)	27 (65)	30 (78)
	10000	27 (64)	30 (78)	27 (65)	27 (65)	30 (78)
	100000	27 (64)	30 (78)	27 (65)	27 (65)	30 (78)
	1000000	27 (64)	31 (80)	29 (69)	27 (65)	33 (84)
Wolfe	1000	52 (105)	52 (105)	70 (141)	52 (105)	64 (129)
	10000	116(235)	114(232)	98 (200)	120(242)	118 (238)
	100000	123(250)	113(234)	108(220)	111(226)	111 (227)
	1000000	122(249)	121(250)	97 (197)	108(220)	106 (215)
Dixon	1000	455(998)	252(994)	4214(8433)	479(1034)	1038(2086)
	10000	481(1062)	521(994)	6597(13197)	485(1068)	6353(12709)
	100000	482(1046)	521(1128)	4214(8433)	483(1060)	481(1065)
	1000000	437(961)	486(8433)	522(1127)	471(1022)	524(1140)
Shallow	1000	10 (25)	10 (25)	10 (25)	10 (25)	10 (25)
	10000	10 (25)	10 (25)	10 (25)	10 (25)	10 (25)
	100000	10 (25)	10 (25)	10 (25)	10 (25)	10 (25)
	1000000	10 (25)	10 (25)	10 (25)	10 (25)	10 (25)
Strait	1000	7 (18)	6 (15)	6 (15)	7 (18)	6 (14)
	10000	7 (18)	6 (15)	6 (15)	7 (18)	6 (14)
	100000	7 (18)	7 (18)	6 (15)	8 (20)	6 (14)
	1000000	8 (20)	7 (18)	6 (15)	8 (20)	6 (14)
Sum	1000	27 (115)	21(106)	18 (82)	25 (91)	21(110)
	10000	41 (161)	23(102)	30 (107)	37 (175)	32(161)
	100000	73 (336)	63(262)	68 (321)	61 (307)	71(315)
	1000000	107(534)	98(380)	136(537)	132(470)	123(552)
Quartc	1000	1 (4)	1 (4)	1 (4)	1 (4)	1 (4)
	10000	1 (4)	1 (4)	1 (4)	1 (4)	1 (4)
	100000	1 (4)	1 (4)	1 (4)	1 (4)	1 (4)
	1000000	1 (4)	1 (4)	1 (4)	1 (4)	1 (4)
Powell-3	3000	17 (37)	20 (43)	14 (31)	17 (36)	21 (46)
	30000	18 (39)	20 (43)	*	*	*
	300000	18 (39)	21 (46)	*	*	*
Raydan	100	13 (39)	45 (91)	46 (93)	13 (39)	45 (91)

* the algorithm fail converge

References

1. Andrei, N. **2006**. Conjugate gradient algorithms for mdecular formation Under pair wise potential minimization. *Mathematical Modeling of Environmental and life Sciences problems*:7-26.
2. Andrei, N. **2010**. Another accelerated conjugate gradient algorithm with guranteed descent and conjugacy condition for large-scale unconstrained optimization, Submitted.
3. Mamat, M., Mohd, I., and Fauzi, M., **2010**.A new conjugate gradient coefficient for unconstrained optimization, *Int. J. Contemp. Sciences*, **5**(29): 1429-1437.
4. Birgn, E. G., and Martinez, J. M. **2001**. A spectral conjugate gradient method for unconstrained optimization. *Appl. Math. Optim.*,**43** :.117-128.
5. Hestenes, M. R., and Stiefel, E. L. **1952**. Methods of conjugate gradient for solving linear systems, *J. Research Nat. Bur. Standards*, 49: 409-436
6. Fletcher, R., and Reeves, C. **1964**. Function minimization by conjugate gradient, *Computer J*., **7:**149-154.
7. Polat, E., and Ribe're, G. R.**1969**. Note sur la convergence de direction conjuge'cs, Rev. *Francaise Informat conjugate gradient methods Research Operational*, 3e Anne's, **16:** 35-43.
8. Polyak, B. T.**1969**. The conjugate gradient method in extreme problems, *USSR Comp. math. And Math. Phys*., **9**:94-112.
9. Liu, Y. and Story.**1991**. Efficient generalized conjugate gradient algorithm", Part I: *Theory J. Optim. Theory Appl*., **69:**129-137.
10. Dai, Y. H., and Yuan, Y. L.,**1999**. A nonlinear conjugate gradient methods with a strong global convergence property, *SIAM J. Optimization*, **10:**177-182.
11. Perry, A. **1978**. A modified conjugate gradient algorithm, *Operation Research*, **26:**1073-1078.
12. Shanno, F. D. **1978**. Conjugate gradient methods with inexact searches, *Mathematics of Operations Research*, **3**:244-256.
13. Dai, Y. H., and Liao, L. Z. **2001**. New conjugacy conditions and related nonlinear conjugate gradient methods, *Applied Mathematics Optimization*, **43:**87-101
14. Wolfe, P. **1969**. Converge condition for ascent methods. II: Some corrections, *SIAM Review*, **13:**185-188.
15. Yabe,H., Sakaiwa, N., and Takano, M**.** **2004**.Global convergence properties of new nonlinear conjugate gradient methods for unconstrained optimization. *Computational Optimization and Application*, **28:**203-225.
16. Glibart, J. C., and Noceded, J**.** **1999**. Global convergence properties of conjugate gradient methods for optimization, *SIAM J., Optimization*, **2:** 21-42.
17. Zhang, L., Zhou, W. and Li, D**.** **2006**. Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line-search, *Numer. Math*., **104:**561-572.
18. Raydan, M., The Barzilai and Borwein**.** **1997**. Gradient method for the large unconstrained minimization problem, *SIAM J. Optim*.,**7:**26-33.
19. Wolfe, P. **1969**. Convergence condition, for ascent methods, *SIAM Rev.*, **11:** 226-235.
20. Zoutendijk, G. **1970**. Non linear programming computational methods, *J. Abadie (Ed.), Integer and Nonlinear programming, North-Holland, Amesterdam*: 37-86.
21. Yab, Y., and Sakaiwa, N. **2005**. A new nonlinear conjugate gradient methods for unconstrained optimization, *J. Operations Research*, **48**(4):284-296.