Nonlinear Conjugate Gradient Algorithms with Modified Secant Condition

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.
Combining the Newton direction with the direction of conjugate gradient algorithms, and using the modified secant equation of Zhang, Deng and Chen [12] and Zhang and Xu [13] we obtain a class of conjugate gradient algorithms. Under common assumptions, these algorithms are globally convergent. The numerical results and comparisons with classical conjugate gradient algorithms show that our conjugate gradient algorithms are more effective than the classical variants.

Keywords: Unconstrained optimization, conjugate gradient method, modified secant condition, line search, numerical comparisons
AMS subject classification: 49M07, 49M10, 90C06, 65K

1. Introduction

Consider the following unconstrained optimization problem

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

where $f: R^n \to R$ is continuously differentiable and its gradient is available. Conjugate gradient methods for solving this problem are iterative methods that generates a sequence x_k of approximations to the minimum x^* of f, of the following form:

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

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3}$$

where $g_k = \nabla f(x_k)$, α_k is selected to minimize $f(x)$ along the search direction d_k, and β_k is a scalar parameter. The iterative process is initialized with an initial point x_0 and $d_0 = -g_0$. Lots of versions of conjugate gradient methods, corresponding to the selection procedure of parameter β_k, are already known. In this paper we consider a way to get a conjugate gradient algorithm by combining the Newton direction with the direction of conjugate gradient algorithms (3) and using the modified secant equation of Zhang, Deng and Chen [12] and Zhang and Xu [13]. In section 2 we present the method and the corresponding algorithm. In section 3 the global convergence analysis is given together with some variants of the main algorithm. Section 4 presents some numerical results and comparisons among conjugate gradient algorithms, including the comparison with CONMIN [10].

2. The method

For solving (1) we consider the iterative process (2), where for $k = 0,1,\ldots$ the step size α_k is positive and the directions d_k are generated by:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k, \tag{4}$$

in which θ_{k+1} and β_k are parameters which follow to be determined and $s_k = x_{k+1} - x_k$.

To determine β_k consider the following procedure. As we know, the Newton direction for solving (1) is given by $d_{k+1} = -\nabla^2 f(x_{k+1})^{-1} g_{k+1}$. Therefore, from the equality $-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k$, we get:

$$\beta_k = \frac{\theta_{k+1} s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k}. \tag{5}$$

Using the Taylor development we have: $s_k^T \nabla^2 f(x_{k+1}) = y_k^T$, where $y_k = g_{k+1} - g_k$.

For quasi-Newton methods, an approximation matrix B_k to the Hessian $\nabla^2 f(x_k)$ is updated so that a new matrix B_{k+1} satisfies the classical secant equation $B_{k+1} s_k = y_k$. Zhang, Deng and Chen [12] and Zhang and Xu [13] extended this condition and obtained a class of *modified secant condition* which use both the available gradient and function value information in two successive points. Their class of modified secant condition is in the form: $B_{k+1} s_k = z_k$, where:

$$z_k = y_k + \frac{\omega_k}{s_k^T u_k} u_k, \qquad \omega_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k$$

and $u_k \in R^n$ is any vector such that $s_k^T u_k \neq 0$. Zhang, Deng and Chen [12] proved that if $\|s_k\|$ is sufficiently small, then for any vector u_k with $s_k^T u_k \neq 0$,

$$s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T y_k = O\left(\|s_k\|^3\right), \tag{6}$$

$$s_k^T \nabla^2 f(x_{k+1}) s_k - s_k^T z_k = O\left(\|s_k\|^4\right) \tag{7}$$

hold. The above equations show that the quantity $s_k^T z_k$ given by the modified secant updating matrix approximates the second-order curvature $s_k^T \nabla^2 f(x_{k+1}) s_k$ with a higher accuracy than the quantity $s_k^T y_k$ does. Therefore, it seems reasonable to use in (5) the higher accuracy modified secant condition result, i.e. to consider:

$$\beta_k = \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{s_k^T z_k}, \qquad \text{where} \qquad z_k = y_k + \rho_k \frac{\omega_k}{s_k^T u_k} u_k$$

and ρ_k is a scalar parameter which follows to be determined. With these, the direction can be computed as

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \frac{(\theta_{k+1} y_k - s_k)^T g_{k+1}}{s_k^T y_k + \rho_k \omega_k} s_k. \tag{8}$$

The following particularizations are obvious:

1) If $\rho_k = 0$ and $\theta_{k+1} = 1$, then (8) is *the direction considered by Perry* [9]. At the same time (8) is the direction obtained by Dai and Liao [4] by using the conjugacy condition $d_{k+1}^T y_k = -t g_{k+1}^T s_k$, for $t = 1$. If $\theta_{k+1} \neq 0$, then (8) is the *scaled Perry direction*.

2) If $\rho_k = 0$ and $\theta_{k+1} \neq 0$, then (8) is the *spectral conjugate gradient direction* of Birgin and Martínez [2], or the direction considered by Andrei [1] in his scaled memoryless BFGS preconditioned conjugate gradient algorithms.

3) If $\rho_k = 0$ and $s_j^T g_{k+1} = 0$ for $j = 0,1,\ldots,k$, then from (8) we get

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \frac{\theta_{k+1} y_k^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k} s_k, \tag{9}$$

which is the direction corresponding to a *generalization of the Polak and Ribière* formula. Of course, if $\theta_{k+1} = \theta_k = 1$ in (9), we get the *classical Polak and Ribière* formula.

4) If $s_j^T g_{k+1} = 0$, $j = 0,1,\ldots,k$, and the successive gradients are orthogonal, then from (8)

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \frac{\theta_{k+1} g_{k+1}^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k} s_k, \tag{10}$$

which is the direction corresponding to a *generalization of the Fletcher and Reeves* formula. If $\theta_{k+1} = \theta_k = 1$, we get the *classical Fletcher and Reeves* formula.

5) If $\theta_{k+1} = 1$ and we set $y_k = z_k$, then we get for β_k the *formula considered by Yabe and Takano* [11, p.207] for $t = 1$. Parameter $t \geq 0$ is coming from the conjugacy condition of Dai and Liao [4] which is conserved by Yabe and Takano in their algorithm.

6) Observe that if the function f is a strictly convex quadratic function and the step length is obtained by the exact line search, then $\omega_k = 0$ and $s_k^T g_{k+1} = 0.$ In this case $\beta_k = \theta_{k+1} y_k^T g_{k+1} / s_k^T y_k$ which correspond to *the scaled Hestenes and Stiefel* formula within the framework of the linear conjugate gradient methods.

To ensure the convergence of the algorithm (2), with d_{k+1} given by (8), we need to constrain the choice of α_k. We consider line searches that satisfy the strong Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \sigma_1 \alpha_k g_k^T d_k, \tag{11}$$

$$\left|\nabla f(x_k + \alpha_k d_k)^T d_k\right| \le -\sigma_2 g_k^T d_k, \tag{12}$$

where $0 < \sigma_1 \le \sigma_2 < 1.$

θ_{k+1} **selection.** Form (4) we see that θ_{k+1} can be considered as a scalar approximation to the inverse Hessian. According to the procedures for a scalar estimation to the inverse Hessian we get a family of scaled conjugate gradient algorithms. The following procedures can be considered [1]:

θ_{k+1} *spectral.* This is given as the inverse of the Rayleigh quotient: $\theta_{k+1} = s_k^T s_k / y_k^T s_k.$

θ_{k+1} *anticipative.* This is only a half step from the spectral procedure. The value for parameter θ_{k+1} is selected as: $\theta_{k+1} = 1/\gamma_{k+1}$ where:

$$\gamma_{k+1} = \frac{2}{d_k^T d_k} \frac{1}{\alpha_k^2} \left[f(x_{k+1}) - f(x_k) - \alpha_k g_k^T d_k \right]. \tag{13}$$

Observe that for convex functions $\gamma_{k+1} > 0.$ It is worth saying that in the two-point step size gradient method Dai, Yuan and Yuan [6] interpret the choice for the step size from the angle interpolation and arrive at the same formula as given in (13). With these the corresponding algorithm can be described as follows:

Algorithm CGMSE (Conjugate Gradient with Modified Secant Equation)

Step 1.	Select the initial point $x_0 \in R^n$, set $d_0 = -g_0$ and $k = 0.$
Step 2.	Compute $\alpha_k > 0$ satisfying the Wolfe line search conditions.
Step 3.	Compute the parameters θ_{k+1} (spectral or anticipative) and β_k .
Step 4.	Direction computation. Define: $d_{k+1} = -\theta_{k+1} g_{k+1} + \beta_k s_k$. If (a restart criterion is satisfied, for example the Powell criterion) $\left\|g_{k+1}^T g_k\right\| > 0.2 \left\|g_{k+1}\right\|^2$, (14) then restart the iterations by $d_{k+1} = -\theta_{k+1} g_{k+1}$.
Step 5.	Set $x_{k+1} = x_k + \alpha_k d_k$. If a criterion for stopping the iterations is satisfied, then stop, otherwise set $k = k + 1$ and go to step 2.

3. Convergence analysis

The strong Wolfe conditions (11), (12) and the Beale-Powell restart criterion (14) are sufficient to prove the global convergence of the algorithm under reasonable assumptions. More exactly, the convergence of the algorithm CGMSE follows immediately from the analysis given by Yabe and Takano [11], and Dai and Liao [4]. Under basic standard assumptions: 1) The level set $\left\{x \in R^n : f(x) \le f(x_0)\right\}$ at x_0 is bounded. 2) The function f is continuously differentiable and its gradient is Lipschitz continuous, i.e. there exists a constant $L > 0$ such that $\left\|\nabla f(x) - \nabla f(y)\right\| \le L \left\|x - y\right\|,$ for all $x, y \in L,$ and supposing that f is a uniformly convex function, i.e. there exists a constant $\mu > 0$ such that:

$$f(x) \ge f(y) + \nabla f(y)^T (x - y) + \frac{\mu}{2} \left\|x - y\right\|^2, \text{ for all } x, y \in L,$$

the following theorem can be proved:

Theorem 1. (Yabe and Takano [11]) *Suppose that the function f satisfy the above basic assumptions and it is a uniformly convex function. Consider the conjugate gradient algorithm (2)-(3) where d_k is given by (8) and α_k satisfies the strong Wolfe line search conditions. If $L = \mu$, then for any $\rho_k \geq 0$, $\liminf_{k\to\infty} \|g_k\| = 0$. If $L > \mu$, then $\lim_{k\to\infty} \|g_k\| = 0$, for $0 \leq \rho_k < \frac{L}{3(L-\mu)}$.*

For general functions based on ideas of Gilbert and Nocedal [8] and Dai and Liao [4], the following theorem can also be proved:

Theorem 2 (Yabe and Takano [11]) *Suppose that the function f satisfy the above basic assumptions and consider the conjugate gradient algorithm (2)-(3) where*

$$\beta_k = max\left\{\frac{\theta_{k+1} y_k^T g_{k+1}}{s_k^T y_k + \rho_k \omega_k}, 0\right\} - \frac{s_k^T g_{k+1}}{s_k^T y_k + \rho_k \omega_k},$$

d_k satisfies the sufficient descent condition $g_k^T d_k \leq -c\|g_k\|^2$, where $c > 0$ is a constant, and α_k satisfies the strong Wolfe line search conditions. If $0 \leq \rho_k < \frac{1-\sigma_2}{3(1+\sigma_2-2\sigma_1)}$, then $\liminf_{k\to\infty} \|g_k\| = 0$.

The above theorems give us some insights how to choose the parameter ρ_k along the iterations of the algorithm CGMSE. The following procedures can be considered. Each of them is specifying a variant of the general CGMSE algorithm.

1. Algorithm CGMSE-UC1. From theorem 1 we see that if $L > \mu$, then $\frac{L}{3(L-\mu)} > \frac{1}{3}$. Therefore, theorem 1 shows that there exists a constant $\rho^* > 1/3$ such that if $0 \leq \rho_k \leq \rho^*$, then $\lim_{k\to\infty} \|g_k\| = 0$. Generally, the positive constants L and μ, which are intrinsic characteristics of the function f, are not evaluated. In this algorithm we select:

$$\rho_k = \frac{L_e}{3(L_e - \mu_e)}, \tag{15}$$

where we suggest the following estimations:

$$L_e = \|y_k\| / \|s_k\| \quad \text{and} \quad \mu_e = 2(f_k - f_{k+1} + g_{k+1}^T s_k) / \|s_k\|^2. \tag{16}$$

Observe that if $g_{k+1}^T d_{k+1} > 0$, i.e. the direction d_{k+1} is not a descent one, then a restart direction $d_{k+1} = -\theta_{k+1} g_{k+1}$ is considered in the algorithm. Therefore, for every $k \geq 1$ the algorithm generates descent directions. Now, using the strong Wolfe line search conditions we have:

$$\mu_e = \frac{2(f_k - f_{k+1} + g_{k+1}^T s_k)}{\|s_k\|^2} \geq \frac{2\alpha_k(\sigma_2 - \sigma_1)}{\|s_k\|^2} g_k^T d_k.$$

Therefore, the above estimation of μ_e can be negative, and in these cases $\rho_k < 1/3$. On the other hand, by strong Wolfe conditions we have: $\omega_k \geq 3\alpha_k(1 - 2\sigma_1 + \sigma_2) g_k^T d_k$, showing again that also ω_k can be negative. However, along the iteration $g_k^T d_k$ is going to zero, and as a consequence $\omega_k \to 0$ and $\rho_k \to 1/3$.

2. Algorithm CGMSE-UC2. In this algorithm we consider the following strategy for ρ_k computation:

$$\rho_k = \frac{L_e}{3(L_e - \mu_e)} \quad \textit{and} \quad \text{if } \rho_k > 1/3, \text{ then } \rho_k = 1/3, \tag{17}$$

where L_e and μ_e are computed as above.

3. Algorithm CGMSE-GF. Following the theorem 2, in this algorithm we select

$$\rho_k = \frac{1-\sigma_2}{3(1+\sigma_2-2\sigma_1)}, \tag{18}$$

where σ_1 and σ_2 are parameters in Wolfe line search conditions. Since $0<\sigma_1\leq\sigma_2<1$ it is clear that the upper bound for ρ_k is less than $1/3$.

4. Algorithm CGMSE-CC. If we impose the conjugacy condition $y_k^T d_{k+1}=0,$ where the direction d_{k+1} is given as in (8), then we obtain the following formula for ρ_k

$$\rho_k = -\frac{(s_k^T g_{k+1})(y_k^T s_k)}{\theta_{k+1}\omega_k (y_k^T g_{k+1})}. \tag{19}$$

Introducing (19) in (8), we get $\beta_k = \theta_{k+1} y_k^T g_{k+1}/s_k^T y_k$, i.e. the conjugacy condition for nonlinear optimization yields to the *scaled Hestenes and Stiefel* method which is in the context of the linear conjugate gradient methods.

5. Algorithm CGMSE-DC. Assume that $\beta_k\geq 0$ and the current search direction d_k is a descent direction, i.e. $g_k^T d_k<0.$ In order to find a β_k that produces a descent direction d_{k+1} the following condition must be satisfied: $g_{k+1}^T d_{k+1} = -\theta_{k+1}\|g_{k+1}\|^2 + \beta_k g_{k+1}^T s_k < 0.$ Suppose that $\beta_k\geq 0$ and d_k is a descent direction. If $\beta_k\leq\theta_{k+1}\|g_{k+1}\|^2/y_k^T s_k$ then d_{k+1} is a descent direction for function f. Therefore using (8), after some algebra, the following value for ρ_k can be considered:

$$\rho_k = -\frac{((\theta_{k+1}g_k+s_k)^T g_{k+1})(s_k^T y_k)}{\theta_{k+1}\omega_k\|g_{k+1}\|^2}. \tag{20}$$

However, introducing (20) in (8) we get the *scaled Dai and Yuan* conjugate gradient method: $\beta_k = \theta_{k+1}\|g_{k+1}\|^2/s_k^T y_k$ [5].

4. Numerical results and comparisons

In this section we present the numerical performances of the above algorithms. All codes are written in 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 [3] library, along with other large-scale optimization test problems. We selected 50 large-scale unconstrained optimization test problems in extended or generalized form. For each test function we have considered 10 numerical experiments with number of variables $n=1000,2000,\ldots,10000.$

In all algorithms the Wolfe line search conditions are implemented with $\sigma_1=0.0001$ and $\sigma_2=0.9.$ In our implementation we have considered the one-dimensional line search used in CONMIN by Shanno and Phua [10] or SCG by Brigin and Martínez [2]. The initial guess of the step length at the first iteration is $\alpha_0 = 1/\|g_0\|.$ At the following iteration, in all algorithms, the starting guess for the step α_k is computed as $\alpha_{k-1}\|d_{k-1}\|_2/\|d_k\|_2.$ In all experiments we stopped the iterations whenever

$$\|g_k\|_\infty\leq\varepsilon_g \quad \text{or} \quad \alpha_k|g_k^T d_k|\leq\varepsilon_f|f(x_{k+1})|, \tag{21}$$

is satisfied, where $\|.\|_\infty$ denotes the maximum absolute component of a vector and $\varepsilon_g=10^{-6}$ and $\varepsilon_f=10^{-20}.$

Tables 1 and 2 contain the global characteristics of conjugate gradient methods. In this study we classified all these methods in three classes. The first one contains the conjugate gradient methods using the modified secant equation, the second one include the scaled Perry, the scaled Polak and Ribière and the scaled Fletcher and Reeves methods. The last one contains the classical methods. In all these methods we have considered the Powell restart criterion (14).

When this criterion is satisfied, then the scaled descent direction is considered: $d_{k+1} = -\theta_{k+1} g_{k+1}$, where θ_{k+1} is computed in a spectral or an anticipative manner.

Table 1. Global characteristics of the algorithms. 500 problems. θ_{k+1} spectral. Powell restart with the direction $d_{k+1} = -\theta_{k+1} g_{k+1}$.

Algorithm	# iterations	# fg	cpu time (sec)
CGMSE-UC1	**133929**	**233656**	**1825.78**
CGMSE-UC2	154521	257672	1923.14
CGMSE-GF	148493	274809	2067.38
CGMSE-CC	141419	250514	1858.20
CGMSE-DC	139085	255765	1921.00
scaled Perry	142915	268377	2046.68
scaled Polak-Ribière	136272	237097	1883.96
scaled Fletcher-Reeves	135281	254043	1923.65
Polak- Ribière	333109	427364	3551.11
Polak- Ribière +	333109	427364	3551.45
Fletcher-Reeves	421166	501361	4044.06
Hestenes-Stiefel	139066	250335	1928.44
Dai-Yuan	142292	265919	1930.56
Dai-Liao (t=1)	142684	258510	1925.06

Table 2. Global characteristics of the algorithms. 500 problems. θ_{k+1} anticipative. Powell restart with the direction $d_{k+1} = -\theta_{k+1} g_{k+1}$.

Algorithm	# iterations	# fg	cpu time (sec)
CGMSE-UC1	154354	265045	2075.68
CGMSE-UC2	133289	237544	1880.03
CGMSE-GF	177843	291372	2163.74
CGMSE-CC	137091	245724	**1817.70**
CGMSE-DC	140456	254687	1913.70
scaled Perry	147978	279341	2102.58
scaled Polak-Ribière	**124380**	**230439**	**1795.64**
scaled Fletcher-Reeves	136930	247807	1878.74
Polak- Ribière	200695	269301	2539.04
Polak- Ribière +	200695	269301	2539.11
Fletcher-Reeves	254331	336275	2868.05
Hestenes-Stiefel	137013	248784	1895.90
Dai-Yuan	137924	249193	1895.75
Dai-Liao (t=1)	140330	255277	1922.42

In Table 1 we have the computational evidence that for θ_{k+1} spectral the most efficient algorithm is CGMSE-UC1. The most inefficient was Fletcher-Reeves. In fact, subject to the number of function evaluations, CGMSE-UC1 was about 2 times more efficient than Fletcher-Reeves, and about 2.2 faster. From Table 2 we can see that for θ_{k+1} anticipative the scaled Polak-Ribière algorithm is the most efficient. However, it is not much better than CGMSE-CC, since the difference is only about 22 seconds for solving the set of 500 problems.

In the second set of numerical experiments we compare the best algorithm CGMSE-UC1 with CONMIN. The CONMIN package by Shanno and Phua [10] solved all these 500 unconstrained optimization test problems in 87611 iterations, 945363 function evaluations and 3789.50 seconds. Subject to the number of function evaluations, we see that for this set of 500 test problems, CGMSE-UC1 with θ_{k+1} spectral is about 4 times more performant. On the other hand it is about 2 times faster than CONMIN. Figures 1 and 2 show the Dolan and Moré [7] performance profile of CGMSE-UC1 versus CONMIN referring to the number of function evaluations and cpu time metrics, respectively. For each algorithm we plot the fraction of

problems for which the algorithm is within a factor of the best of the number of function evaluations and cpu time metrics, respectively. The left side of these Figures gives the percentage of the test problems, out of 500, for which an algorithm is more performant. The right side gives the percentage of the test problems that were successfully solved by each of the algorithms. Mainly, the right side represents a measure of the algorithm's robustness. The top curve corresponds to the algorithm that solved the most problems in a number of function evaluations (Figure 1) or in a cpu time (Figure 2) that was within a given factor τ of the best number of function evaluations or cpu time, respectively. Since the top curve in Figures 1 and 2 corresponds to CGMSE-UC1, this algorithm is clearly better than CONMIN.

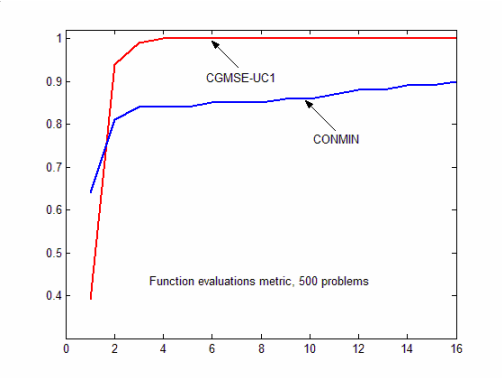

Fig. 1. CGMSE-UC1 with θ_{k+1} spectral versus CONMIN. Function evaluations metric.

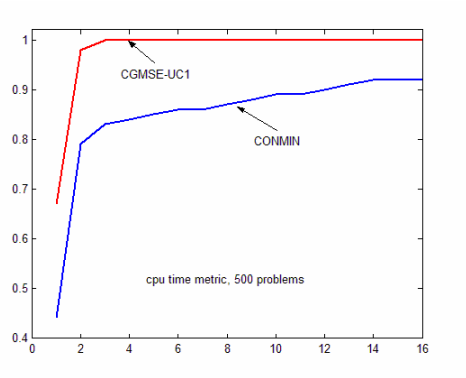

Fig. 2. CGMSE-UC1 with θ_{k+1} spectral versus CONMIN. cpu time metric.

5. Conclusions

In this paper we have presented a class of conjugate gradient algorithms by combining the Newton direction with the classical conjugate gradient direction and by using the modified secant condition given by Zhang, Deng and Chen [12] and Zhang and Xu [13]. Under common assumptions the algorithms are globally convergent. The modified secant condition in the frame of conjugate gradient algorithms has been also considered by Yabe and Takano [11], but using the conjugacy condition of Dai and Liao [4]. The algorithm of Yabe and Takano depends on two parameters ρ and t. For different choices of these parameters, the performance of their algorithm, as illustrated on 4 unconstrained optimization problems, can be quite different. In this paper we use the modified secant condition in a more natural context and suggest some formulas for parameter ρ computation. The numerical study indicates that our conjugate gradient algorithms with modified secant condition and spectral selection of scaling parameter θ_{k+1} are more performant and more robust than the classical conjugate gradient algorithms.

References

[1] N. Andrei, Scaled conjugate gradient algorithms for unconstrained optimization. ICI Technical Report, September 2005.

[2] E. Birgin and J.M. Martínez, A spectral conjugate gradient method for unconstrained optimization. *Applied Math. and Optimization*, 43, 117-128, (2001).

[3] I. Bongartz, A.R. Con, N.I.M. Gould and P.L. Toint, CUTE: a constrained and unconstrained testing environments. *ACM Trans. Math. Software, 21, 123-160, (1995).*

[4] Y.-H. Dai and L.-Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and Optimization*, 43, 87-101, (2001).

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

[6] Y.H. Dai, J.Y. Yuan and Y-X. Yuan, Modified two-point step size gradient methods for unconstrained optimization. *Computational Optimization and Applications,* 22, 103-109, (2002).

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

[8] J.C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for

optimization. *SIAM J. Optim.* 2, 21-42, (1992).

[9] J.M. Perry, 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.

[10] D.F. Shanno and K.H. Phua, Algorithm 500, minimization of unconstrained multivariate functions. *ACM Trans. on Math. Soft.,* 2, 87-94, (1976)

[11] H. Yabe and M. Takano, Global convergence properties of nonlinear conjugate gradient methods with modified secant condition, *Computational Optimization and Applications*, 28, 203-225, (2004).

[12] J.Z. Zhang, N.Y. Deng, and L.H. Chen, New quasi-Newton equation and related methods for unconmstrained optimization. *Journal of Optimization Theory and Applications*, 102, 147-167, (1999).

[13] J.Z. Zhang and C.X. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. *Journal of Computational and Applied Mathematics*, 137, 269-278, (2001).

November 22, 2005