A Scaled Conjugate Gradient Method Based on New BFGS Secant Equation with Modified Nonmonotone Line Search

Tsegay Giday Woldu¹, Haibin Zhang¹, Yemane Hailu Fissuh²

¹Department of Applied Mathematics, Beijing University of Technology, Beijing, China
²Department of Statistics, Aksum University, Aksum, Ethiopia
Email: tsegikuhil@gmail.com, zhanghaibn@bjut.edu.cn, yemaoroba@gmail.com

Abstract
In this paper, we provide and analyze a new scaled conjugate gradient method and its performance, based on the modified secant equation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method and on a new modified nonmonotone line search technique. The method incorporates the modified BFGS secant equation in an effort to include the second order information of the objective function. The new secant equation has both gradient and function value information, and its update formula inherits the positive definiteness of Hessian approximation for general convex function. In order to improve the likelihood of finding a global optimal solution, we introduce a new modified nonmonotone line search technique. It is shown that, for nonsmooth convex problems, the proposed algorithm is globally convergent. Numerical results show that this new scaled conjugate gradient algorithm is promising and efficient for solving not only convex but also some large scale nonsmooth nonconvex problems in the sense of the Dolan-Moré performance profiles.

Keywords
Conjugate Gradient Method, BFGS Method, Modified Secant Equation, Nonmonotone Line Search, Nonsmooth Optimization

1. Introduction
The conjugate gradient method (CG) and Quasi-Newton method are two major popular iterative methods for solving smooth unconstrained optimization problems, and Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is one of the most
efficient quasi-Newton methods for solving small and medium sized unconstrained optimization problems [1] [2] [3] [4]. The iterative method is computed by

\[ x_{k+1} = x_k + \alpha_k d_k, \]  

(1)

where \( \alpha_k \) is a step size and \( d_k \) is a search direction. For continuously differentiable function \( h : \mathbb{R}^n \to \mathbb{R} \), the minimization problem:

\[ \min_{x \in \mathbb{R}^n} h(x) \]  

(2)

has been well studied for several decades. Conjugate gradient method is among the preferable methods for solving problem (2) with search direction \( d_k \) given by

\[ d_k = \begin{cases} -\nabla h_k + \beta_k d_{k-1} & \text{if } k \geq 1, \\ -\nabla h_k & \text{if } k = 0, \end{cases} \]  

(3)

where \( \nabla h_k \) is the gradient of an objective function \( h(x) \) at \( k \) iterate and \( \beta_k \) is a scalar describing the attributes of the CG methods.

Some well-known formulas for the scalar \( \beta_k \) are the Hestenes-Stiefel (HS) [5], Fletcher-Reeves (FR) [6], Polak-Ribière and Polak (PRP) [7], and Dai-Yuan (DY) [8] given by

\[ \beta_k^{\text{HS}} = \frac{\nabla h_k^T y_k}{d_{k-1}^T y_k}, \quad \beta_k^{\text{PRP}} = \frac{\nabla h_k^T y_k}{\|\nabla h_k\|}, \]  

\[ \beta_k^{\text{FR}} = \frac{\|\nabla h_k\|^2}{\|\nabla h_{k-1}\|^2}, \quad \beta_k^{\text{DY}} = \frac{\|\nabla h_k\|^2}{d_{k-1}^T y_k}, \]  

where \( y_k = \nabla h(x_k) - \nabla h(x_{k-1}) \) and \( \| \| \) denotes the Euclidean norm. Due to their simplicity and low memory requirement, CG methods are more effective and desirable for large scale unconstrained smooth problems [9] [10]. The global convergence properties of nonlinear CG methods have been analyzed under the weak Wolfe line search:

\[ \begin{cases} h(x_k + \alpha_k d_k) \leq h(x_k) + \zeta \alpha_k \nabla h_k^T d_k, \\ \nabla h(x_k + \alpha_k d_k)^T d_k \geq \rho \nabla h_k^T d_k, \end{cases} \]  

(4)

and the strong Wolfe line search:

\[ \begin{cases} h(x_k + \alpha_k d_k) \leq h(x_k) + \zeta \alpha_k \nabla h_k^T d_k, \\ \|\nabla h(x_k + \alpha_k d_k)^T d_k\| \leq \rho \|\nabla h_k^T d_k\|, \end{cases} \]  

(5)

where \( 0 < \zeta < \rho < 1 \). CG methods use relatively little memory for large scale problems and require no numerical linear algebra, so each step is quite fast. However, they do not have second order information of the objective function, and typically converge much more slowly than Newton or quasi-Newton methods.

The quasi-Newton method is an iterative method with second order information of the objective function, and BFGS is the effective quasi-Newton method.

DOI: 10.4236/ajcm.2020.101001
with the search direction
\[ d_k = -B_k \nabla h_k, \]  
(6)

where \( B_k \) is an approximation of the Hessian matrix of \( h \) at \( x_k \). The update formula for \( B_k \) is defined by
\[ B_{k+1} = B_k - \frac{B_k s_k y_k^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \]  
(7)

where \( s_k \) is defined as \( s_k = x_{k+1} - x_k \), and the Hessian approximation \( B_{k+1} \) of (7) satisfies the standard secant equation
\[ B_{k+1} s_k = y_k, \]  
(8)

if \( y_k^T s_k > 0 \), which is known as the curvature condition. The BFGS method has very interesting properties and remains one of the most respectable quasi-Newton methods for unconstrained optimization [11]. The theory of BFGS method and its global convergence have been established by many researchers (see [12]). For convex objective function, using some special inexact line search, it has been proved that the BFGS method is globally convergent (see [13] [14] [15]). However, when the objective function is nonconvex, the BFGS method under exact line search may fail to converge [16]. Moreover, Dai [17] proved that the BFGS method may fail for nonconvex functions with Wolfe line search techniques given in (4) and (5) [18]. Wolfe line search technique is the most common monotone line search technique, and it may leads to small steps without making significant progress to the minimum when the contours of the objective functions are a family of curves with large curvature (see [19] [20] [21] [22]). In order to overcome this drawback, the first nonmonotone line search technique was proposed by Grippo et al. [19] for Newton’s method. With this initiative, many nonmonotone line search techniques have been proposed in recent years [23]. Yuan et al. [24] developed a modified limited memory BFGS method with the update formula that has a higher order approximation to exact Hessian, and its convergence property is analyzed under the nonmonotone line search type. However, the method converges for only uniformly convex functions. Li et al. [25] proposed a new BFGS algorithm with modified secant equation which achieves both global and superlinear convergence for generally convex functions under the nonmonotone line search of [19]. Su and Rong [26] introduced and established a new spectral CG method and its implementation under a modified nonmonotone line search technique. They introduced a new spectral conjugate gradient direction
\[ d_k = \begin{cases} -\theta_k \nabla h_k + \beta_k d_{k-1} & \text{if } k \geq 1, \\ -\nabla h_0 & \text{if } k = 0, \end{cases} \]  
(9)

where
\[ \theta_k = 1 + \frac{\beta_k d_k^T \nabla h_k}{\| \nabla h_k \|^2}, \]  
(10)
\[ \beta_k = \frac{\nabla h_k^T y_{k-1}}{(1-\tau)\|
abla h_{k-1}\|^2 + \tau d_k^T y_{k-1}}, \tag{11} \]

and \( \tau \in [0, 1] \). It is not difficult to notice that the denominator of (11) is the convex combination of the denominator of the conjugate parameters in HS and PRP conjugate gradient methods. The choice of spectral parameter given (10) ensures the sufficient descent property of the search direction without dependence of line search. The convergence property of their method analyzed under a new modified nonmonotone line search with some mild conditions. However, this spectral CG method has only first order information, and excludes second order information. When the number of dimension is large, the CG methods are more effective compared to the BFGS methods in term of the CPU-time but in term of the number of iterations and the number of function evaluations, the BFGS methods are better. In order to incorporate the remarkable properties of the CG and BFGS methods and to overcome their drawbacks, many hybrid of CG and BFGS methods are introduced for unconstrained smooth optimization [27] [28] [29]. However, the usage of these methods is mainly restricted to solve smooth optimization problems. Recently, Yuan et al. [30] [31] [32] [33] introduced some CG approaches to solve nonsmooth convex large scale problems using the smoothing regularization, and under some assumptions, the global convergence properties of these approaches are analyzed. Yuan and Wei [34] proposed the Barzilai and Borwein (BB) gradient method with nonmonotone line search to solve nonsmooth convex optimization problems. Some implementable quasi-Newton methods are also introduced for solving the same problem (see [35] [36] [37] [38]). More recently, Ou and Zhou [39] introduced a modified scaled BFGS preconditioned CG algorithm, and under appropriate assumptions, the method is proven to possess global convergence for nonsmooth convex functions.

Motivated by the work of Ou and Zhou [39], in this paper, we propose a hybrid approach of the a scaled CG method and a modified BFGS method to combine the simplicity of CG method and the Hessian approximation of BFGS method. Our work is mainly focused in developing the scaled conjugate search direction that includes the second order information of the objective function by incorporating the modified secant equation of BFGS method. Opposing the work of Ou and Zhou [39], our method has both the function and gradient value information of the objective function. Moreover, our method leads to better descent direction than the CG methods proposed so far. To the best of our knowledge, this is the first work to incorporate the scaled CG algorithm with the BFGS secant equation which contains both the function and gradient value information of the objective function for solving large scale nonsmooth optimization. Under the new modified nonmonotone line search technique, the global convergence of the algorithm is analyzed for nonsmooth convex problems.

The paper is organized as follows. In the next section, we consider a nonsmooth convex problem and review their basic results. In Section 3, we propose
a new scaled CG algorithm that incorporates the BFGS secant equation which has both function value and gradient information of the objective function via the smoothing regularization. Using the new modified nonmonotone line search technique, we prove the global convergence of our new algorithm for nonsmooth convex problems. Numerical results and related comparisons are reported in Section 4. Finally, Section 5 concludes our work.

2. Nonsmooth Convex Problems and Their Basic Results

In this section, we consider the unconstrained optimization problem

\[
\min_{x \in \mathbb{R}^n} f(x),
\]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is a possibly nonsmooth convex function. This problem is equivalent to the following problem

\[
\min_{x \in \mathbb{R}^n} F(x),
\]

where \( F : \mathbb{R}^n \rightarrow \mathbb{R} \) is the Moreau-Yosida regularization of \( f \) [40], which is defined by

\[
F(x) = \min_{z \in \mathbb{R}^n} \left\{ f(z) + \frac{1}{2\lambda} \| z - x \|^2 \right\},
\]

where \( \lambda \) is a positive parameter. The function \( F \) is a finite-valued, continuously differentiable convex function even though the function \( f \) is nondifferentiable (see [40]). Let \( p(x) \) be the unique solution of (14). In what follows, we can express \( F(x) \) as

\[
F(x) = f(p(x)) + \frac{1}{2\lambda} \| p(x) - x \|^2.
\]

Moreover, the gradient of \( F \) is globally Lipschitz continuous, i.e.,

\[
\| g(x) - g(y) \| \leq \frac{1}{\lambda} \| x - y \|, \quad \forall x, y \in \mathbb{R}^n,
\]

where

\[
g(x) = \nabla F(x) = \frac{x - p(x)}{\lambda}.
\]

The point \( x \in \mathbb{R}^n \) is an optimal solution to (12) if and only if \( g(x) = 0 \) (see [40]). Furthermore, under reasonable conditions the gradient of \( F \) is semismooth and some of its remarkable properties are given in [41] [42].

Several methods have been proposed to solve (13) by incorporating bundle methods and quasi-Newton methods ideas [43] [44] [45], but it is burdensome to evaluate the exact value of \( p(x) \) at any given point \( x \) [46]. Luckily, for each \( x \in \mathbb{R}^n \) and any \( \varepsilon > 0 \), we can have \( p^\varepsilon(x, \varepsilon) \in \mathbb{R}^n \) such that

\[
f\left( p^\varepsilon(x, \varepsilon) \right) + \frac{1}{2\lambda} \| p^\varepsilon(x, \varepsilon) - x \|^2 \leq F(x) + \varepsilon.
\]

Therefore, we can approximate \( F(x) \) and \( g(x) \) by
\[
F^a(x,\varepsilon) = f\left(p^a(x,\varepsilon)\right) + \frac{1}{2\lambda}\|p^a(x,\varepsilon) - x\|^2, \tag{19}
\]
and
\[
g^a(x,\varepsilon) = \frac{x - p^a(x,\varepsilon)}{\lambda}, \tag{20}
\]
respectively. Implementable algorithms to define such a \( p^a(x,\varepsilon) \) for non-smooth convex model can be seen in [47]. The noticeable attributes of \( F^a(x,\varepsilon) \) and \( g^a(x,\varepsilon) \) by the following proposition [48].

Proposition 1. Let \( p^a(x,\varepsilon) \) be a vector that satisfies (18), and let \( F^a(x,\varepsilon) \) and \( g^a(x,\varepsilon) \) be defined by (19) and (20), respectively. Then we obtain
\[
F(x) \leq F^a(x,\varepsilon) \leq F(x) + \varepsilon, \tag{21}
\]
\[
\|p^a(x,\varepsilon) - p(x)\| \leq \sqrt{2\lambda\varepsilon}, \tag{22}
\]
and
\[
\|g^a(x,\varepsilon) - g(x)\| \leq \sqrt{2\varepsilon/\lambda}. \tag{23}
\]

Proposition 1 shows that the approximations of \( F^a(x,\varepsilon) \) and \( g^a(x,\varepsilon) \) can be made arbitrarily close to the exact values of \( F(x) \) and \( g(x) \) respectively.

### 3. A Scaled CG Method Based on New BFGS Secant Equation

In this section, we introduce the new scaled CG search direction that incorporates the modified BFGS secant equation, and then describe the new algorithm for solving nonsmooth problems. We make use of a modified nonmonotone line search technique introduced by [23] to compute a step size. Based on the above approximations, we redefine the search direction of CG method (3) to solve problem (13) as follows:
\[
d_{k+1} = \begin{cases} 
-g^a(x_{k+1},\varepsilon_{k+1}) + \beta_{k+1}d_k & \text{if } k \geq 1, \\
-g^a(x_{k+1},\varepsilon_{k+1}) & \text{if } k = 0,
\end{cases} \tag{24}
\]
where \( \varepsilon \) is an appropriately chosen positive number. Ou and Zhou [39] provided a search direction defined by
\[
d_{k+1} = \begin{cases} 
-\tilde{Q}_{k+1}g^a(x_{k+1},\varepsilon_{k+1}) & \text{if } k \geq 1, \\
-g^a(x_{k+1},\varepsilon_{k+1}) & \text{if } k = 0,
\end{cases} \tag{25}
\]
where \( \tilde{Q}_{k+1} \in \mathbb{R}^{m \times m} \) is defined
\[
\tilde{Q}_{k+1} = \tilde{\theta}_{k+1}I - \tilde{\theta}_{k+1}w_k s_k^T + \frac{s_k w_k^T}{w_k^T s_k} + 1 + \tilde{\theta}_{k+1} \frac{w_k w_k^T}{w_k^T s_k} \frac{s_k s_k^T}{w_k^T s_k}, \tag{26}
\]
with
\[
\tilde{\theta}_{k+1} = \frac{s_k^T s_k}{w_k^T s_k},
\]
where
The vector $k\alpha \epsilon\alpha\epsilon^*$ and $k\alpha t$ in (27) are defined as
\[ k\alpha \epsilon\epsilon\epsilon^* = g^a(x_{k+1}, \epsilon_{k+1}) - g^a(x_k, \epsilon_k) \tag{28} \]
and
\[ t_k = t + \max \left\{ \frac{s_k^T y_k}{\|s_k\|^2}, 0 \right\} \quad (t > 0). \tag{29} \]

It is easy to observe that the (27) has only gradient value information. In order to have both gradient and function value information, we replace (27) and (29) by
\[ w_k^* = y_k^* + \max \left\{ t_k^*, 0 \right\} s_k, \tag{30} \]
and
\[ t_k^* = \frac{6 \left[ F(x_k) - F(x_k + \alpha_k d_k) \right] + 3 \left( g^a(x_k + \alpha_k d_k, \epsilon_{k+1}) + g^a(x_k, \epsilon_k) \right)^T s_k}{\|s_k\|^2}, \tag{31} \]
respectively. Thus, the BFGS method with the secant equation
\[ B_{k+1} s_k = w_k^*, \tag{32} \]
and the update formula
\[ B_{k+1} = B_k - \frac{B_k s_k s_k^T}{s_k^T B_k s_k} + \frac{w_k^* s_k^* s_k^*}{w_k^* s_k^*}, \tag{33} \]
has both gradient and function value information, and the matrix $B_{k+1}$ inherits the positive definiteness of $B_k$ for generally convex functions. Using the secant Equation (32), we propose the new search direction is defined by
\[ d_{k+1} = \begin{cases} \bar\beta_{k+1} g^a(x_{k+1}, \epsilon_{k+1}) + \beta_{k+1} d_k - \partial_{k+1} w_k^*, & \text{if } k \geq 1, \\ -g^a(x_{k+1}, \epsilon_{k+1}), & \text{if } k = 0, \end{cases} \tag{34} \]
where
\[ \bar\beta_{k+1} = 2 - \frac{d_k^T g^a(x_{k+1}, \epsilon_{k+1})}{\left\| g^a(x_{k+1}, \epsilon_{k+1}) \right\|^2} \left( \frac{g^a(x_{k+1}, \epsilon_{k+1})^T w_k^*}{\left\| d_k \right\| \left\| w_k^* \right\|} \right), \tag{35} \]
\[ \beta_{k+1} = \frac{g^a(x_{k+1}, \epsilon_{k+1})^T w_k^*}{\left\| d_k \right\| \left\| w_k^* \right\| + \left\| d_k^T y_k^* \right\|}, \tag{36} \]
and
\[ \partial_{k+1} = \frac{d_k^T g^a(x_{k+1}, \epsilon_{k+1})}{\left\| d_k \right\| \left\| w_k^* \right\|}. \tag{37} \]

Now, based on the above search direction, we describe our new scaled CG al-
Algorithm with a modified nonmonotone line search for solving problem (13) as follows.

**Algorithm 1**

**Step 0.** Given $\epsilon > 0, \beta \in (0,1), \epsilon \in (0,1), \sigma \in (0,1)$, and a point $x_0 \in \mathbb{R}^n$. Set $d_0 = -g^a(x_0, \epsilon_0)$ and $k := 0$.

**Step 1.** If $\|g^a(x_k, \epsilon_k)\| < \epsilon$, then stop, else go to the next step.

**Step 2.** Compute the search direction $d_k$ by using (34)-(37).

**Step 3.** Set trial step size $\alpha_k = 1$.

**Step 4.** Set $x_{k+1} = x_k + \alpha_k d_k$ and choose a scalar $\epsilon_{k+1}$ such that $0 < \epsilon_{k+1} < \epsilon_k$.

**Step 5.** Let $\mu \in (0,1], M \geq 1$ is a positive integer, define $m(k) = \min \{k + 1, M\}$, and choose $\mu_i \geq \mu, i = 0, 1, 2, \cdots, m(k) - 1$, $\sum_{i=0}^{m(k)-1} \mu_i = 1$.

Let $\alpha_k \geq 0$ be bounded above and satisfy:

$$F^a(x_k + \alpha_k d_k, \epsilon_{k+1})$$

$$\leq \max \left[ F^a(x_k, \epsilon_k), \sum_{i=0}^{m(k)-1} \mu_i F^a(x_{k-i}, \epsilon_{k-i}) \right] \sigma \alpha_k g^a(x_k, \epsilon_k)^\top d_k.$$  

(38)

If (38) does not holds, define $\alpha_k = \beta \alpha_k$ and go to step 5.

**Step 6.** Set $K := k + 1$ and go to step 1.

It can be observed that the line search technique in step 5 of Algorithm 1 is a nonmonotone line search technique with some modifications.

**Convergence Analysis**

In this subsection, we establish the global convergence of our method for nonsmooth convex problem (12). To prove the global convergence of Algorithm 1, the following Lemmas are needed.

**Lemma 1.** Assume that the search direction $d_k$ is generated by Algorithm 1, then for all $k \geq 0$, we have

$$g^a(x_{k+1}, \epsilon_{k+1})^\top d_{k+1} \leq -\|g^a(x_{k+1}, \epsilon_{k+1})\|^2,$$  

(39)

and

$$\|d_{k+1}\| \leq 5\|g^a(x_{k+1}, \epsilon_{k+1})\|.$$  

(40)

**Proof.** If $k = 0$, then

$$g^a(x_0, \epsilon_0)^\top d_0 = -g^a(x_0, \epsilon_0)^\top g^a(x_0, \epsilon_0) = -\|g^a(x_0, \epsilon_0)\|^2,$$

and

$$\|d_0\| = \|g^a(x_0, \epsilon_0)\| \leq \|g^a(x_0, \epsilon_0)\| \leq 5\|g^a(x_0, \epsilon_0)\|.$$  

Let $k \geq 1$, then from (34) we have
Once more, (34) yields that
\[
\|d_{k+1}\| = \left\| -\hat{\beta}_{k+1} g^a(x_{k+1}, e_{k+1}) + \beta_{k+1} d_k - \beta_{x_{k+1}} w_k^* \right\|
\]
\[
= \left\| -\hat{\beta}_{k+1} g^a(x_{k+1}, e_{k+1}) + \beta_{k+1} d_k - \beta_{x_{k+1}} g^a(x_{k+1}, e_{k+1})^T w_k^* \right\|
\]
\[
\leq \|\hat{\beta}_{k+1} g^a(x_{k+1}, e_{k+1})\| + 2 \|g^a(x_{k+1}, e_{k+1})\|
\]
\[
\leq 4 \|g^a(x_{k+1}, e_{k+1})\| + \frac{\|d_k\| \|w_k^*\|}{\|g^a(x_{k+1}, e_{k+1})\|} \left\| g^a(x_{k+1}, e_{k+1}) \right\|
\]
\[
\leq \frac{5}{L} \left\| g^a(x_{k+1}, e_{k+1}) \right\|
\]
Thus, the proof is completed.

Lemma 1 shows that the search direction \(d_k\) developed in (34)-(37) leads to the most sufficiently descent direction and it belongs to a trust region.

**Lemma 2.** Let the step size \(\alpha_k\) satisfy (38), then there exist \(\beta > 0\) satisfy a
\[
\alpha_k \geq \min \left\{ \frac{1}{L}, \frac{1}{\beta} \right\}
\]

**Proof.** If \(\alpha_k = 1\) satisfies the formula (38), then the proof is completed. Otherwise, there exist \(\beta\) such that
\[ F^a \left( x_i + \frac{\alpha_k}{\beta} d_k, e_{k+1} \right) > \max \left\{ F^a \left( x_i, e_k \right), \sum_{j=0}^{m(k)-1} \mu_{ij} F^a \left( x_{i-j}, e_{k-j} \right) \right\} + \frac{\alpha_k}{\beta} g^a \left( x_i, e_k \right) \mathbf{1}^T d_k \]

Thus,
\[ F^a \left( x_i + \frac{\alpha_k}{\beta} d_k, e_{k+1} \right) > F^a \left( x_i, e_k \right) + \frac{\alpha_k}{\beta} g^a \left( x_i, e_k \right) \mathbf{1}^T d_k. \]

Using mean value theorem, we have
\[ F^a \left( x_i + \alpha d_k, e_{k+1} \right) - F^a \left( x_i, e_k \right) = \int_0^\alpha \left( g^a \left( x_i + t d_k, e_{k+1} \right) - g^a \left( x_i, e_k \right) \right) \mathbf{1}^T dt + \alpha g^a \left( x_i, e_k \right) \mathbf{1}^T d_k \]
\[ \leq \frac{1}{2} \alpha \beta \alpha \|d_k\|^2 + \alpha g^a \left( x_i, e_k \right) \mathbf{1}^T d_k. \]

Combining the above inequality with (42), we have
\[ \alpha_k \geq \min \left\{ 1, \frac{\left( 1 - \sigma \right) \beta \| g^a \left( x_i, e_k \right) \mathbf{1}^T d_k \|}{L \|d_k\|^2} \right\}. \]

Thus, the proof is completed.

**Lemma 3.** Assume that the sequence \( \{x_k\} \) is generated by Algorithm 1, then we have
\[ F^a \left( x_i, e_k \right) \leq F^a \left( x_0, e_0 \right) + \mu \xi \sum_{i=0}^{K-1} \alpha_i g^a \left( x_i, e_i \right) \mathbf{1}^T d_i + \sigma \xi \sum_{i=0}^{K-1} g^a \left( x_{i-1}, e_{k-1} \right) \mathbf{1}^T d_{k-i} \]
\[ \leq F^a \left( x_0, e_0 \right) + \mu \xi \sum_{i=0}^{K-1} \alpha_i g^a \left( x_i, e_i \right) \mathbf{1}^T d_i. \]

**Proof.** We prove this lemma by induction. For \( k = 1 \), by (38) and \( \mu \leq 1 \), we have
\[ F^a \left( x_i, e_i \right) \leq F^a \left( x_0, e_0 \right) + \sigma \xi g^a \left( x_0, e_0 \right) \mathbf{1}^T d_0 \]
\[ \leq F^a \left( x_0, e_0 \right) + \mu \xi g^a \left( x_0, e_0 \right) \mathbf{1}^T d_0 \]

Assume the equation holds for \( 1, 2, \cdots, k \), and we need to show for \( k+1 \). To show the condition, we have considered two cases.

**Case 1:**
\[ \max \left\{ F^a \left( x_i, e_k \right), \sum_{j=0}^{m(k)-1} \mu_{ij} F^a \left( x_{i-j}, e_{k-j} \right) \right\} = F^a \left( x_k, e_{k} \right). \]

Then, from (38), we have
\[ F^a \left( x_{k+1}, e_{k+1} \right) = F^a \left( x_k + \alpha d_k, e_{k+1} \right) \]
\[ \leq F^a \left( x_k, e_k \right) + \sigma \xi g^a \left( x_k, e_k \right) \mathbf{1}^T d_k \]
\[ \leq F^a \left( x_0, e_0 \right) + \mu \xi \sum_{i=0}^{k} \alpha_i g^a \left( x_i, e_i \right) \mathbf{1}^T d_i + \sigma \xi \sum_{i=0}^{k} g^a \left( x_{i-1}, e_{k-i} \right) \mathbf{1}^T d_{k-i} \]
\[ \leq F^a \left( x_0, e_0 \right) + \mu \xi \sum_{i=0}^{k} \alpha_i g^a \left( x_i, e_i \right) \mathbf{1}^T d_i. \]
Case 2:

$$\max \left[ F^\alpha (x_k, \epsilon_k), \sum_{i=0}^{m(k)-1} \mu_{k,i} F^\alpha (x_{k-i}, \epsilon_{k-i}) \right] = \sum_{i=0}^{m(k)-1} \mu_{k,i} F^\alpha (x_{k-i}, \epsilon_{k-i}),$$

let $n = \min \left[ k, m-1 \right]$. Then, again from (38),

$$F^\alpha (x_{k+1}, \epsilon_{k+1}) = F^\alpha (x_k + \alpha d_k, \epsilon_{k+1})$$

$$\leq \sum_{j=0}^{k} \mu_{j} F^\alpha (x_{k-j}, \epsilon_{k-j}) + \sigma \alpha \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k$$

$$\leq \sum_{j=0}^{k} \mu_{j} \left[ F^\alpha (x_0, \epsilon_0) + \mu \sigma \sum_{i=0}^{k-2} \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i \right.$$

$$+ \sigma \alpha \mathbf{g}^\alpha (x_{k-j-1}, \epsilon_{k-j-1})^T d_{k-j-1} \right] + \sigma \alpha \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k.$$

Thus, by imposing

$$(1,2,\ldots,n) \times (1,2,\ldots,k-n-2) \subset \{(j,i) : 0 \leq j \leq n, 0 \leq i \leq k-n-2\},$$

and

$$\sum_{j=0}^{n} \mu_{j} = 1, \mu_{j} \geq \mu,$$

we have

$$F^\alpha (x_{k+1}, \epsilon_{k+1})$$

$$\leq F^\alpha (x_0, \epsilon_0) + \mu \sum_{j=0}^{k-2} \left[ \sum_{j=0}^{n} \mu_{j} \right] \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i$$

$$+ \sigma \sum_{j=0}^{k} \mu_{j} \alpha \mathbf{g}^\alpha (x_{k-j-1}, \epsilon_{k-j-1})^T d_{k-j-1} + \sigma \alpha \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k$$

$$\leq F^\alpha (x_0, \epsilon_0) + \mu \sigma \sum_{i=0}^{k-2} \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i$$

$$+ \mu \sigma \sum_{i=0}^{k-1} \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i + \sigma \alpha \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k$$

$$= F^\alpha (x_0, \epsilon_0) + \mu \sigma \sum_{i=0}^{k-1} \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i + \sigma \alpha \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k$$

$$\leq F^\alpha (x_0, \epsilon_0) + \mu \sigma \sum_{i=0}^{k-1} \alpha \mathbf{g}^\alpha (x_i, \epsilon_i)^T d_i.$$

**Theorem 1.** Assume that the sequences $\{x_k\}$ and $\{d_k\}$ are generated by Algorithm 1. Let $F$ is bounded below on the level set

$$\mathcal{L}_0 = \left\{ x \in \mathbb{R}^n \mid F(x) \leq F(x_0) \right\} \text{ and }$$

$$\lim_{k \to \infty} \epsilon_k = 0.$$

Then

$$\lim_{k \to \infty} \mathbf{g}^\alpha (x_k, \epsilon_k)^T d_k = 0. \quad (43)$$

**Proof.** Suppose that (43) is not true. Then there exist constants $\gamma > 0$ and $k_0$ such that
From Lemma 3, we have
\[ F^a(x_0, \varepsilon_0) - F^a(x_k, \varepsilon_k) \geq -\mu \sigma \sum_{i=0}^{k-1} g^a(x_i, \varepsilon_i)^T d_i. \] 
(45)

By (40), (41) and (44), we have
\[ F(x_0, \varepsilon_0) - F(x_k, \varepsilon_k) \geq -\mu \sigma \sum_{i=0}^{k-1} g(x_i, \varepsilon_i)^T d_i \geq \mu \sigma \sum_{i=0}^{k-1} \alpha_i. \]
\[ \geq \mu \sigma \sum_{i=0}^{k-1} \min \left\{ 1, \frac{(1-\sigma) \beta} {L} \right\} \left\| d_i \right\|^2 \]
\[ \geq \mu \sigma \sum_{i=0}^{k-1} \min \left\{ 1, \frac{(1-\sigma) \beta} {25L} \right\}. \]

Letting \( k \to \infty \), we have
\[ \mu \sigma \sum_{i=0}^{k} \min \left\{ 1, \frac{(1-\sigma) \beta} {25L} \right\} \leq \sum_{i=0}^{k} F(x_0, \varepsilon_0) - F(x_k, \varepsilon_k), \]
and this contradicts our assumption on \( F \). Hence the theorem is proved.

**Theorem 2.** Let the conditions in Lemma 1 and Theorem 1 hold, then Algorithm 1 converges for nonsmooth problem (12).

**Proof.** From Lemma 1 and Theorem 1, we have
\[ 0 \geq \lim_{k \to \infty} \left\| g^a(x_k, \varepsilon_k) \right\| \geq \lim_{k \to \infty} g^a(x_k, \varepsilon_k)^T d_k = 0. \]
(46)

Then,
\[ \lim_{k \to \infty} \left\| g^a(x_k, \varepsilon_k) \right\| = 0. \]

Thus, (23) and convergence of sequence \( \{ \varepsilon_k \} \) yield
\[ 0 \leq \lim_{k \to \infty} \left\| g^a(x_k, \varepsilon_k) - g(x_k) \right\| \leq \lim_{k \to \infty} \sqrt{2\varepsilon_k^2} = 0. \]

Hence,
\[ \lim_{k \to \infty} \left\| g(x_k) \right\| = 0. \]
(47)

Let \( x^* \) be an accumulation point of \( \{ x_k \} \). Then there exists a subsequence \( \{ x_{k_i} \} \) satisfying
\[ \lim_{k \to \infty} x_{k_i} = x^*. \]
(48)

Thus, (17), (43) and (47) yield \( x^* = p(x^*) \). Therefore \( x^* \) is an optimal solution of nonsmooth problem (12).

### 4. Numerical Experiments for Large Scale Nonsmooth Problems

In this section, we present some numerical experiments to examine the efficien-
cy of Algorithm 1 for some large scale nonsmooth academic test problems which are introduced in [49]. The details of these large scale nonsmooth academic test problems with their initial points $x_i^{(i)}$ and the minimum values $f(x')$ are listed as follows:

**Problem 1**

$$f(x) = \max_{1 \leq i \leq n} x_i^2$$

$$x_i^{(i)} = i \text{ for } i = 1, \cdots, n/2 \text{ and }$$

$$x_i^{(i)} = -i \text{ for } i = n/2 + 1, \cdots, n$$

$$f(x') = 0.$$

**Problem 2**

$$f(x) = \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} \frac{x_j}{i+j-1} \right|$$

$$x_i^{(i)} = i \text{ for } i = 1, \cdots, n.$$  

$$f(x') = 0.$$

**Problem 3**

$$f(x) = \sum_{i=1}^{n} \max \left\{ -x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1) \right\}$$

$$x_i^{(i)} = -0.5 \text{ for } i = 1, \cdots, n;$$

$$f(x') = -\sqrt{2(n-1)}.$$

**Problem 4**

$$f(x) = \sum_{i=1}^{n} \max \left\{ x_i^4 + x_{i+1}^2, (2-x_i)^2 + (2-x_{i+1})^2, 2e^{-x_i^2+x_{i+1}} \right\}$$

$$x_i^{(i)} = 2 \text{ for } i = 1, \cdots, n;$$

$$f(x') = 2(n-1).$$

**Problem 5**

$$f(x) = \max \left\{ \sum_{i=1}^{n} (x_i^4 + x_{i+1}^2), \sum_{i=1}^{n} ((2-x_i)^2 + (2-x_{i+1})^2), \sum_{i=1}^{n} (2e^{-x_i^2+x_{i+1}}) \right\}$$

$$x_i^{(i)} = 2 \text{ for } i = 1, \cdots, n;$$

$$f(x') = 2(n-1).$$

**Problem 6**

$$f(x) = \max_{1 \leq i \leq n} \left\{ g \left( \sum_{i=1}^{n} x_i \right), g(x) \right\},$$

where $g(y) = \ln(y) + 1$;

$$x_i^{(i)} = 1 \text{ for } i = 1, \cdots, n;$$

$$f(x') = 0.$$

**Problem 7**

$$f(x) = \sum_{i=1}^{n} \left( |x_i|^{i+1} + |x_{i+1}|^{i+1} \right)$$
\[ x_i^{(1)} = -1 \quad \text{when} \quad \text{mod}(i, 2) = 1, (i = 1, \ldots, n) \quad \text{and} \\
\quad x_i^{(1)} = 1 \quad \text{when} \quad \text{mod}(i, 2) = 0, (i = 1, \ldots, n); \\
f(x^*) = 0. 
\]

**Problem 8**

\[
f(x) = \sum_{i=1}^{n} \left( -x_i + 2 \left( x_i^2 + x_{i+1}^2 - 1 \right) + 1.75 \left| x_i^2 + x_{i+1}^2 - 1 \right| \right)
\]

\[ x_i^{(1)} = -1 \quad \text{for} \quad i = 1, \ldots, n; \\
f(x^*) = \text{varies.} 
\]

**Problem 9**

\[
f(x) = \max \left\{ \sum_{i=1}^{n} \left( x_i^2 + (x_{i+1} - 1)^2 + x_i - 1 \right), \sum_{i=1}^{n} \left( -x_i^2 - (x_{i+1} - 1)^2 + x_i + 1 \right) \right\}
\]

\[ x_i^{(1)} = -1.5 \quad \text{when} \quad \text{mod}(i, 2) = 1, (i = 1, \ldots, n) \quad \text{and} \\
\quad x_i^{(1)} = 2.0 \quad \text{when} \quad \text{mod}(i, 2) = 0, (i = 1, \ldots, n); \\
f(x^*) = 0. 
\]

**Problem 10**

\[
f(x) = \sum_{i=1}^{n} \max \left\{ x_i^2 + (x_{i+1} - 1)^2 + x_i - 1, -x_i^2 - (x_{i+1} - 1)^2 + x_i + 1 \right\}
\]

\[ x_i^{(1)} = -1.5 \quad \text{when} \quad \text{mod}(i, 2) = 1, (i = 1, \ldots, n) \quad \text{and} \\
\quad x_i^{(1)} = 2.0 \quad \text{when} \quad \text{mod}(i, 2) = 0, (i = 1, \ldots, n); \\
f(x^*) = 0. 
\]

The problems 1 - 5 are convex functions, and the others are nonconvex functions. We test the above problems with the dimension of \( n = 1000, 3000, 5000, 6000, 10000, 20000, 50000, 60000 \) and \( n = 100000 \). For convenience sake, we denote Algorithm 1 by scaled conjugate gradient method based on modified secant equation of BFGS method (SCG-MBFGS), and in order to demonstrate validity of our algorithm, we also list the results of other three algorithms MPRP in [30], MHS in [31] and MSBFGS-CG in [39]. All algorithms were implemented in Fortran 90 and run on a PC with an intel(R) Core(TM)i3-3110M CPU at 2.40 GHz, 4.00 GB of RAM, and the Windows 7 operating system. We stopped the iteration when the condition \( \| g(x^*) \| \leq 10^{-10} \) was satisfied. The parameters for SCG-MBFGS were chosen as \( M = 10 \beta = 0.6, \sigma = 0.85 \lambda = \mu = 1 \). All parameters for other three methods are chosen as in [30] [31] and [39] respectively. **Table 1** shows the numerical results of SCG-MBFGS, MPRP, MHS and MSBFGS-CG on the given test problems. The columns in **Table 1** have the following meanings:

- Dim: the dimensions of problem.
- NI: the total number of iterations.
- NF: the number of function evaluations.
- TIME: the CPU time in seconds.
- \( f(x) \): the value of \( f(x) \) at the final iteration.

From the numerical results in **Table 1**, it is not difficult to see that
Table 1. Numerical results for 10 problems with given initial points and dimensions.

<table>
<thead>
<tr>
<th>No.</th>
<th>Dim</th>
<th>Algorithm 3.1</th>
<th>MHS</th>
<th>MPRP</th>
<th>MSBFGS−CG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NI/NF/f(x)/TIME</td>
<td></td>
<td></td>
<td>NI/NF/f(x)/TIME</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>213/4351/6.917E−8/−8.0.303E+0</td>
<td>225/4710/6.9345E−8/0.4358E+0</td>
<td>225/4710/6.7102E−6/6.6947E+0</td>
<td>186/1601/2.6568E−0/9.5977E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>219/4509/1.795E−8/−8.0.2356E+0</td>
<td>238/5273/6.815E−8/1.1157E+0</td>
<td>228/5196/6.5327E−6/1.6902E+0</td>
<td>219/2310/2.2089E−0/1.3871E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>224/4597/5.773E−7/−7.1.6248E+0</td>
<td>246/5284/8.4915E−7/1.2650E+0</td>
<td>250/5253/9.018E−7/1.9357E+0</td>
<td>242/2725/1.2183E−9/1.6962E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6000</td>
<td>227/4608/5.8167E9/−8.1.1756E+0</td>
<td>235/5298/6.7261E−8/1.2323E+0</td>
<td>252/5308/8.5863E−6/2.0417E+0</td>
<td>284/2489/1.8972E−8/2.0233E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>230/4719/3.0892E−6/−6.1.9051E+0</td>
<td>264/5387/7.3580E−6/1.8361E+0</td>
<td>261/5466/3.0488E−8/8.2684E+0</td>
<td>253/2996/3.4045E−9/2.1527E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12,000</td>
<td>234/4841/8.5011E−8/−8.2.1084E+0</td>
<td>264/5529/6.5042E−8/2.4321E+0</td>
<td>266/5573/9.0925E−7/2.5337E+0</td>
<td>261/3328/4.9085E−9/2.2886E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20,000</td>
<td>243/5150/7.7681E−8/−8.2.4301E+0</td>
<td>275/5830/6.9187E−8/2.6893E+0</td>
<td>270/5769/5.6898E−8/2.8049E+0</td>
<td>275/4573/6.1823E−10/2.6981E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50,000</td>
<td>268/5539/1.0736E−8/−8.2.9990E+0</td>
<td>289/6002/7.6945E−7/2.8937E+0</td>
<td>286/5991/8.0015E−6/3.0359E+0</td>
<td>283/5597/7.0687E−8/2.9298E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60,000</td>
<td>277/5913/5.924E−8/−8.3.1618E+0</td>
<td>294/6125/6.4520E−8/2.9931E+0</td>
<td>295/6110/9.6412E−7/2.3229E+0</td>
<td>288/6018/1.7851E−8/3.1037E+0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

T. G. Woldu et al.

DOI: 10.4236/ajcm.2020.101001 15 American Journal of Computational Mathematics
| Value | Expression | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 | Data 8 | Data 9 | Data 10 | Data 11 | Data 12 | Data 13 | Data 14 | Data 15 | Data 16 | Data 17 | Data 18 | Data 19 | Data 20 | Data 21 | Data 22 | Data 23 | Data 24 | Data 25 | Data 26 | Data 27 | Data 28 | Data 29 | Data 30 | Data 31 | Data 32 | Data 33 | Data 34 | Data 35 | Data 36 | Data 37 | Data 38 | Data 39 | Data 40 | Data 41 | Data 42 | Data 43 | Data 44 | Data 45 | Data 46 | Data 47 | Data 48 | Data 49 | Data 50 | Data 51 | Data 52 | Data 53 | Data 54 | Data 55 | Data 56 | Data 57 | Data 58 | Data 59 | Data 60 | Data 61 | Data 62 | Data 63 | Data 64 | Data 65 | Data 66 | Data 67 | Data 68 | Data 69 | Data 70 | Data 71 | Data 72 | Data 73 | Data 74 | Data 75 | Data 76 | Data 77 | Data 78 | Data 79 | Data 80 | Data 81 | Data 82 | Data 83 | Data 84 | Data 85 | Data 86 | Data 87 | Data 88 | Data 89 | Data 90 | Data 91 | Data 92 | Data 93 | Data 94 | Data 95 | Data 96 | Data 97 | Data 98 | Data 99 | Data 100 |
SCG-MBFGS is superior or competitive to the other three methods in solving the given problems in terms of number of iteration, number of function evaluations and CPU time. Furthermore, to directly illustrate the performances of our method, we employed the tool provided by Dolan and Moré [50] to analyze and compare the efficiency of the method in terms of the number of iterations, number of function evaluations and CPU time. **Figures 1-3** represent the computational performance profiles of the above algorithms regarding the number of iterations, number of function evaluations, and CPU time respectively. From the 3 figures, we can observe that for the given test problems, SCG-MBFGS is competitive or superior to other three methods in terms of number of iteration, function evaluations and CPU time respectively.
From Figure 1 and Figure 2, we also notice that SCG-MBFGS performs better than the other methods do in terms of the numbers of iterations and function evaluations. Figure 3 indicates that MHS is comparable to SCG-MBFGS in terms of CPU time, and since the search direction of MHS is developed with only first order information while SCG-MBFGS, MPRP and MSBFGS-CG are with second order information, it is reasonable to need less CPU time for MHS.

Figure 1. Performance profiles of these three methods based on NI.

Figure 2. Performance profiles of these three methods based on NF.

Figure 3. Performance profiles of these three methods based on CPUTIME.
5. Conclusion

In this paper, we propose a new scaled conjugate gradient method which incorporates a modified secant equation of BFGS method. This modified secant equation contains both function value and gradient information of the objective function, and its Hessian approximation update generates positive definite matrix. Under a modified nonmonotone line search and some mild conditions, the strong global convergence of the proposed method is analyzed for nonsmooth convex problems. The search direction of our new method generates sufficiently descent condition and belongs to a trust region. Compared with existing nonsmooth CG methods, the search direction of our approach is more descent direction. Numerical results and related comparisons show that the proposed method is effective for solving large scale nonsmooth optimization problems.

Acknowledgements

The authors would like to thank the reviewers and editor for their valuable comments which greatly improve our paper. This work is supported by the National Natural Science Foundation of China [Grant No. 11771003].

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References


