A MODIFIED SCALED MEMORYLESS BFGS PRECONDITIONED CONJUGATE GRADIENT ALGORITHM FOR NONSMOOTH CONVEX OPTIMIZATION

YIGUI OU* AND XIN ZHOU
Department of Applied Mathematics
Hainan University, Haikou 570228, China

(Communicated by Wenxun Xing)

ABSTRACT. This paper presents a nonmonotone scaled memoryless BFGS preconditioned conjugate gradient algorithm for solving nonsmooth convex optimization problems, which combines the idea of scaled memoryless BFGS preconditioned conjugate gradient method with the nonmonotone technique and the Moreau-Yosida regularization. The proposed method makes use of approximate function and gradient values of the Moreau-Yosida regularization instead of the corresponding exact values. Under mild conditions, the global convergence of the proposed method is established. Preliminary numerical results and related comparisons show that the proposed method can be applied to solve large scale nonsmooth convex optimization problems.

1. Introduction. In this paper, we are concerned with the following unconstrained optimization problem

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

(1)

where $f : \mathbb{R}^n \to \mathbb{R}$ is a possibly nondifferentiable convex function. Associated with this problem is the following problem

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

(2)

where $F : \mathbb{R}^n \to \mathbb{R}$ is the so-called Moreau-Yosida regularization of $f$ [16], which is defined by

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

(3)

where $\lambda$ is a positive parameter. Throughout the paper, the value of parameter $\lambda$ is fixed.

It is a well known fact that problems (1) and (2) are equivalent in the sense that an $x \in \mathbb{R}^n$ solves (1) if and only if it solves (2), see [16] for details. A remarkable feature of the function $F$ is that it is a differentiable convex function whose gradient is Lipschitz continuous, even though the function $f$ is nondifferentiable.

2010 Mathematics Subject Classification. Primary: 90C25, 90C30; Secondary: 65K05.

Key words and phrases. Nonsmooth convex optimization, memoryless BFGS method, scaled conjugate gradient method, nonmonotone technique, Moreau-Yosida regularization, convergence analysis.

The work is supported by NNSF of China (No. 11261015) and NSF of Hainan Province (No. 2016CXTD004; No. 111001; No. 117011).

* Corresponding author: Yigui Ou.

785
Furthermore, it is also shown that the gradient function of $F$ is semismooth under some reasonable conditions [24]. These mentioned features motivate us to develop an efficient algorithm for solving problem (2) through the Moreau-Yosida regularization, particularly when $f$ is nondifferentiable.

Earlier methods attempt to solve problem (2) by combining bundle methods and quasi-Newton ideas, but are conceptual in the sense that they make use of the exact function and gradient values of the Moreau-Yosida regularization, see [22, 17, 7] for instance. In practice, however, it is difficult or even impossible in general to evaluate exactly the function value $F(x)$ and its gradient at any given point $x$. To overcome this difficulty, Fukushima and Qi [12] explored the possibility of using approximations of those values instead of their exact values, and proposed a globally and superlinearly convergent proximal Newton method for solving problem (2). Subsequently, Rauf and Fukushima [25] developed a quasi-Newton method that extended the algorithm proposed by Fukushima and Qi [12]. Implementable algorithms for solving the nonsmooth problem (1) that combine quasi-Newton approaches with Moreau-Yosida regularization are also studied by other scholars, see [8, 9, 28] for instance. More recently, some other implementable methods have been developed for solving nonsmooth convex optimization (1), such as the trust region method, the Barzilai and Borwein (BB) gradient method, etc, see [26, 20, 23, 14, 32, 34] for instance.

Conjugate gradient (CG) methods are efficient approaches for solving large scale unconstrained optimization problems, due to their simple computation and low memory requirement, see [29, 15] for instance. Recently, Andrei [1] proposed a new scaled conjugate gradient (SCALCG) algorithm for solving large scale unconstrained optimization problems using a hybridization of the memoryless BFGS preconditioned CG method suggested by Shanno [27] and the spectral CG method suggested by Birgin and Martínez [6], based on the standard secant equation (see [29] for details). Numerical comparisons show that the SCALCG algorithm [1] outperforms several well known CG algorithms, such as the spectral CG algorithm [6], the PRP-CG algorithm (see [29] for instance) and the CG-Descent algorithm (see [15] for instance). Subsequently, some modified versions of the SCALCG algorithm have been proposed, see [3, 4] for instance. It should be pointed out that the usage of CG-based methods mentioned above are mainly restricted to solving smooth optimization problems. More recently, Li [18] and Yuan et al. [33, 31, 30] extend some CG approaches to solve nonsmooth convex optimization problems by way of the Moreau-Yosida regularization. Under some assumptions, the global convergence properties of these algorithms [18, 33, 31, 30] are established. To the best of our knowledge, however, there are very few studies on nonmonotone scaled memoryless BFGS preconditioned conjugate gradient method for minimizing the Moreau-Yosida regularization so far.

Motivated by the above observations, we propose a nonmonotone scaled memoryless BFGS preconditioned CG algorithm for solving nonsmooth convex optimization problems, which combines the idea of scaled memoryless BFGS preconditioned conjugate gradient method with the nonmonotone technique and the Moreau-Yosida regularization. The proposed method makes use of approximate function and gradient values of the Moreau-Yosida regularization instead of the corresponding exact values. Under some reasonable conditions, we establish the global convergence of the proposed method. Preliminary numerical results show that the proposed method can be applied to solve large scale nonsmooth convex optimization problems.
The paper is organized as follows. In next section, we review some basic results in convex analysis and nonsmooth analysis, and outline briefly the SCALCG algorithm. In Section 3, we propose a new scaled memoryless BFGS preconditioned CG method which is based on the Moreau-Yosida regularization and a modified nonmonotone line search scheme. Section 4 is devoted to analyzing its convergence properties. Numerical results and related comparisons are reported in Section 5. Finally, some concluding remarks are given in Section 6.

Our notation is standard. For any points \(x, y \in \mathbb{R}^n\), \(\langle x, y \rangle = x^T y\) stands for the Euclidean inner product, and \(\| \cdot \|\) for the associated norm. Given a convex function \(f\), we denote its subdifferential at the point \(x\) by \(\partial f(x) = \{\xi : f(y) \geq f(x) + \xi^T (y - x), \forall y\}\) [16].

2. Preliminaries. In this section, we review some basic results and the SCALCG algorithm, which are useful in the subsequent discussions.

By Theorem 15.4.1.4 in [16], the Moreau-Yosida regularization \(F\) defined by (3) has the following basic properties.

**Proposition 2.1.** The function \(F\) is finite-valued, convex and everywhere differentiable with gradient

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

where \(p(x)\) is the unique minimizer in (3), i.e.,

\[
p(x) = \arg \min_{z \in \mathbb{R}^n} \{f(z) + \frac{1}{2\lambda} \|z - x\|^2\}.
\]

Moreover, we have for all \(x, y \in \mathbb{R}^n\),

\[
\|g(x) - g(y)\|^2 \leq \frac{(g(x) - g(y))^T(x - y)}{\lambda},
\]

and

\[
\|g(x) - g(y)\| \leq \frac{\|x - y\|}{\lambda}.
\]

This proposition shows that the mapping \(g : \mathbb{R}^n \to \mathbb{R}^n\) is globally Lipschitz continuous. The next proposition (see Theorem 15.4.1.7 in [16]) states the equivalence between problem (1) and problem (2).

**Proposition 2.2.** The following statements are equivalent:

1. \(x\) minimizes \(f\) on \(\mathbb{R}^n\);
2. \(x = p(x)\);
3. \(g(x) = 0\);
4. \(x\) minimizes \(F\) on \(\mathbb{R}^n\).

The SCALCG algorithm for solving problem (2) is an iterative algorithm [1], which consists of iteration of the form

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

where \(\alpha_k > 0\) is a stepsize satisfying the standard Wolfe conditions (see [29] for details), and \(d_k\) is a search direction defined by

\[
d_0 = -g_0,
\]

\[
d_{k+1} = -Q_{k+1} g_{k+1}, k = 0, 1, 2, \cdots,
\]

(7)
with the matrix $Q_{k+1} \in \mathbb{R}^{n \times n}$ defined by
\begin{equation}
Q_{k+1} = \theta_{k+1} I - \theta_{k+1} \frac{y_k s_k^T + s_k y_k^T}{y_k^T s_k} + \left[ 1 + \theta_{k+1} \frac{y_k^T y_k}{y_k^T s_k} \right] \frac{s_k s_k^T}{y_k^T s_k},
\end{equation}
where $s_k = x_{k+1} - x_k$, $g_k = \nabla F(x_k)$, $y_k = g_{k+1} - g_k$, and $\theta_{k+1}$ is a scaling parameter determined based on a two-point approximation of the standard secant equation [5], i.e.,
\begin{equation}
\theta_{k+1} = \frac{s_k^T s_k}{y_k^T s_k}.
\end{equation}

Since it is impossible in general to evaluate exactly the function $F$ defined by (3) and its gradient $g$ at an arbitrary point $x$, we shall make use of approximate values of the gradient and the function itself. Fortunately, for each $x \in \mathbb{R}^n$ and any $\epsilon > 0$, there exists an approximate vector $p^\alpha(x, \epsilon) \in \mathbb{R}^n$ such that
\begin{equation}
f(p^\alpha(x, \epsilon)) + \frac{1}{2\lambda} \|p^\alpha(x, \epsilon) - x\|^2 \leq F(x) + \epsilon.
\end{equation}

Hence, we can use $p^\alpha(x, \epsilon)$ to define the approximations of $F(x)$ and $g(x)$ by
\begin{equation}
F^\alpha(x, \epsilon) = f(p^\alpha(x, \epsilon)) + \frac{1}{2\lambda} \|p^\alpha(x, \epsilon) - x\|^2,
\end{equation}
and
\begin{equation}
g^\alpha(x, \epsilon) = \frac{x - p^\alpha(x, \epsilon)}{\lambda},
\end{equation}
respectively. Some implementable procedures for computing such an approximate minimizer $p^\alpha(x, \epsilon)$ can be found in [11, 2]. A remarkable feature of $F^\alpha(x, \epsilon)$ and $g^\alpha(x, \epsilon)$ is given by the following proposition [12].

**Proposition 2.3.** Let $p^\alpha(x, \epsilon)$ be a vector satisfying (10), and $F^\alpha(x, \epsilon)$ and $g^\alpha(x, \epsilon)$ be defined by (11) and (12), respectively. Then we have
\begin{align}
(1) \quad & F(x) \leq F^\alpha(x, \epsilon) \leq F(x) + \epsilon, \\
(2) \quad & \|p^\alpha(x, \epsilon) - p(x)\| \leq \sqrt{2\lambda \epsilon}, \\
(3) \quad & \|g^\alpha(x, \epsilon) - g(x)\| \leq \sqrt{\frac{2\epsilon}{\lambda}}.
\end{align}

The above proposition shows that the approximations $F^\alpha(x, \epsilon)$ and $g^\alpha(x, \epsilon)$ can be made arbitrarily close to the exact values $F(x)$ and $g(x)$, respectively, by choosing the parameter $\epsilon$ small enough. With these approximations, we redefine the search direction $d_k$ as
\begin{align}
d_0 &= -g^\alpha(x_0, \epsilon_0), \\
d_{k+1} &= -Q_{k+1} g^\alpha(x_{k+1}, \epsilon_{k+1}), k = 0, 1, 2, \ldots,
\end{align}
where $\epsilon_k$ is an appropriately chosen positive number, and the matrix $\tilde{Q}_{k+1} \in \mathbb{R}^{n \times n}$ is defined by
\begin{equation}
\tilde{Q}_{k+1} = \tilde{\theta}_{k+1} I - \tilde{\theta}_{k+1} \frac{w_k s_k^T + s_k w_k^T}{w_k^T s_k} + \left[ 1 + \tilde{\theta}_{k+1} \frac{w_k^T w_k}{w_k^T s_k} \right] \frac{s_k s_k^T}{w_k^T s_k},
\end{equation}
with
\begin{equation}
\tilde{\theta}_{k+1} = \frac{s_k^T s_k}{w_k^T s_k},
\end{equation}
where
\begin{align}
s_k &= x_{k+1} - x_k, \quad w_k = \tilde{y}_k + t_k s_k,
\end{align}
with
\[ \tilde{y}_k = g^\alpha(x_{k+1}, \epsilon_{k+1}) - g^\alpha(x_k, \epsilon_k), \quad t_k = t + \max\{-\frac{s_k^T \tilde{y}_k}{\|s_k\|^2}, 0\}(t > 0). \quad (20) \]

**Remark 2.1.** It is remarkable that if \( \tilde{y}_k^T s_k < 0 \), then we have
\[ w_k^T s_k = t\|s_k\|^2 > 0, \quad (21) \]
otherwise,
\[ w_k^T s_k = \tilde{y}_k^T s_k + t\|s_k\|^2 \geq t\|s_k\|^2 > 0. \quad (22) \]

Note that the matrix \( \bar{Q}_{k+1} \) defined by (17) is precisely the self-scaling BFGS update in which the approximation of the inverse Hessian is restarted as \( \tilde{\theta}_{k+1} I \). Since it follows from (21) and (22) that \( w_k^T s_k > 0 \), the matrix \( \bar{Q}_{k+1} \) is symmetric positive definite (see [29] for details). Furthermore, using Sherman-Morrison Theorem (see [29] for details), it can be shown that the matrix \( \tilde{P}_{k+1} \) defined by
\[ \tilde{P}_{k+1} = \frac{1}{\tilde{\theta}_{k+1}} I - \frac{1}{\tilde{\theta}_{k+1} s_k^T s_k} s_k s_k^T + \frac{w_k w_k^T}{w_k^T s_k} \quad (23) \]
is the inverse of \( \bar{Q}_{k+1} \). Thus, \( \tilde{P}_{k+1} \) is also a symmetric positive definite matrix.

3. **New method.** In this section, we first introduce a modified nonmonotone line search scheme, and then describe the new algorithm for solving problem (2). We also give some remarks about the proposed method.

**Modified line search scheme** Given \( \sigma \in (0, 1), \ \beta \in (0, 1), \ \gamma \in (0, 1) \), and a sequence \( \{\epsilon_k\} \) satisfying \( 0 < \epsilon_{k+1} < \epsilon_k \) and \( \sum_{i=0}^{\infty} \epsilon_k < \infty \). Let the stepsize \( \alpha_k \) be the largest \( \alpha \) in \( \{1, \beta, \beta^2, \cdots\} \) such that
\[ F^\alpha(x_k + \alpha_k d_k, \gamma \epsilon_k) \leq C_k - \sigma \alpha_k^2 \|d_k\|^2 + \epsilon_k, \quad (24) \]
where \( C_0 = F^\alpha(x_0, 0) \), and \( C_k \) is updated by the following rules:
\[ \begin{align*}
E_{k+1} &= \eta_k E_k + 1, \\
C_{k+1} &= \frac{\eta_k E_k (C_k + \epsilon_k) + F^\alpha(x_{k+1}, \gamma \epsilon_k)}{E_{k+1}},
\end{align*} \quad (25) \]
with \( E_0 = 1 \) and \( 0 \leq \eta_k \leq \eta_{\text{max}} < 1 \).

**Remark 3.1.** The line search scheme (24) can be viewed as a modified version of the nonmonotone line search in [35] and the monotone line search in [36, 19]. In fact, if \( \epsilon_k \to 0 \), then the update rule of \( C_k \) is close to that in [35]; If \( \eta_k \equiv 0 \) for all \( k \) and \( \epsilon_k \to 0 \), then the line search scheme (24) behaves like the line search scheme in [36, 19]. Furthermore, we can deduce from Lemma 4.2 below that the modified line search scheme (24) is well defined.

Now we describe our new scaled memoryless BFGS preconditioned conjugate gradient algorithm for solving (2) as follows.

**Algorithm 3.1**

**Step 0.** Given \( \epsilon > 0, \ \sigma \in (0, 1), \ \beta \in (0, 1), \ \gamma \in (0, 1), \ \lambda > 0, \ \nu > 0, \ t > 0, \ \epsilon_{-1} > 0, \) and an initial point \( x_0 \in \mathbb{R}^n \). Set \( \epsilon_0 = \gamma \epsilon_{-1} \) and \( k := 0 \).

**Step 1.** Compute an approximation \( p^\alpha(x_k, \epsilon_k) \), then evaluate
\[ F^\alpha(x_k, \epsilon_k) = f(p^\alpha(x_k, \epsilon_k)) + \frac{1}{2 \lambda} \|p^\alpha(x_k, \epsilon_k) - x_k\|^2, \quad (26) \]
and
\[ g^\alpha(x_k, \epsilon_k) = \frac{x_k - p^\alpha(x_k, \epsilon_k)}{\lambda}. \] (27)

If \( \|g^\alpha(x_k, \epsilon_k)\| \leq \varepsilon \), then stop.

**Step 2.** Compute the direction \( d_k \) by using (16)-(20).

**Step 3.** Determine a stepsize \( \alpha_k \) satisfying (24). Set \( x_{k+1} = x_k + \alpha_k d_k \).

**Step 4.** If \( \epsilon_k \leq \nu \alpha_k^2 \|d_k\|^2 \), then go to Step 5; otherwise, set \( \epsilon_k := \nu \alpha_k^2 \|d_k\|^2 \).

**Step 5.** Set \( \epsilon_{k+1} = \gamma \epsilon_k \). Let \( k := k + 1 \) and go to Step 1.

**Remark 3.2.** From Step 1, we can see that computing the approximation \( p^\alpha(x_k, \epsilon_k) \) plays a key role in implementing Algorithm 3.1. In practical computation, we employ an implementable procedure [11] to obtain an approximation \( \epsilon_k \) from Step 1, we can see that computing the approximation \( p^\alpha(x_k, \epsilon_k) \) to the unique minimizer \( p(x_k) \) in (3) at the \( k \)-th iteration. The procedure is described as follows.

**Procedure for calculating \( p^\alpha(x_k, \epsilon_k) \) [11]:**

(a) Set \( z^0 = x_k \). Choose \( s^0 \in \partial f(z^0) \) and set \( i := 1 \).

(b) Solve the following quadratic programming to obtain \( z^i \) and the optimal value \( F_k^i \) of the objective function:

\[
\begin{align*}
\min & \left[ \frac{1}{2} \|z - x_k\|^2 + \eta \right], \quad (z, \eta) \in \mathbb{R}^n \times \mathbb{R} \\
\text{s.t.} & \quad f(z^j) + (s^j)^T(z - z^j) \leq \eta, \quad j = 0, 1, \ldots, i - 1. \\
\end{align*}
\] (28)

(c) Compute \( F_k(z^i) \), where \( F_k(z) = \frac{1}{2} \|z - x_k\|^2 + f(z) \).

If
\[
F_k(z^i) - F_k^i \leq \epsilon_k,
\] (29)
then terminate with \( p^\alpha(x_k, \epsilon_k) := z^i \). Otherwise, choose \( s^i \in \partial f(z^i) \). Set \( i := i + 1 \) and go to (b).

It should be mentioned that, in Step (b), \( z^i \) may be calculated practically by solving the dual problem associated with (28), that is,
\[
z^i = x_k - \lambda \sum_{j=0}^{i-1} \alpha_j s^j,
\] (30)
where \( \alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{i-1})^T \in \mathbb{R}^i \) solves
\[
\begin{align*}
\min & \left[ \frac{1}{2} \|\sum_{j=0}^{i-1} \alpha_j s^j\|^2 + \frac{1}{\lambda} \sum_{j=0}^{i-1} \alpha_j e^j \right] \\
\text{s.t.} & \quad \alpha_j \geq 0, \quad \text{and} \quad \sum_{j=0}^{i-1} \alpha_j = 1, \\
\end{align*}
\] (31)
with
\[
e^j := e(x_k, z^j, s^j) = f(x_k) - f(z^j) - (s^j)^T(x_k - z^j), \quad j = 0, 1, \ldots, i - 1.
\]

For more details about (30) and (31), see Lemma 15.2.4.1 in [16].

**Remark 3.3.** Note that, by construction of Step 4 in Algorithm 3.1, we always have the inequality \( \epsilon_k \leq \nu \alpha_k^2 \|d_k\|^2 \) for all \( k \), which is an essential condition for guaranteeing the global convergence of Algorithm 3.1 (see next section for details).
4. Convergence analysis. In this section, we turn our attention to the convergence properties of Algorithm 3.1. To this end, we need the following assumption.

A. The level set $L_0$ defined by

$$L_0 = \{ x \in \mathbb{R}^n | F(x) \leq F^\alpha(x_0, \epsilon_0) + \sum_{i=0}^{+\infty} \epsilon_i \}$$

(32)

is bounded, where $x_0$ is an available initial point.

Remark 4.1. As mentioned in [18], Assumption A is a weaker condition than the strong convexity of $f$ as required in [25, 26].

Lemma 4.1. Let $\{x_k\}$ be the sequence generated by Algorithm 3.1. Then we have

$$F^\alpha(x_k, \gamma \epsilon_{k-1}) \leq C_k, \ \forall k.$$  

(33)

Proof. From (25), it follows that

$$F^\alpha(x_{k+1}, \gamma \epsilon_k) - C_{k+1} = \frac{\eta_k E_k}{E_{k+1}}(F^\alpha(x_{k+1}, \gamma \epsilon_k) - C_k) - \frac{\eta_k E_k}{E_{k+1}} \epsilon_k, \ \forall k.$$  

(34)

We proceed the proof by induction. For $k = 0$, (33) holds, due to the facts that $C_0 = F^\alpha(x_0, \epsilon_0)$ and $\epsilon_0 = \gamma \epsilon_{-1}$. Next, assume that (33) holds for some $k \geq 1$. If we can show the following assertion:

$$F^\alpha(x_{k+1}, \gamma \epsilon_k) \leq C_k + \epsilon_k,$$  

(35)

then it follows from (34) and (35) that

$$F^\alpha(x_{k+1}, \gamma \epsilon_k) \leq C_{k+1},$$

which show that (33) holds for $k + 1$ as well.

In what follows, we verify the validity of the assertion (35). It suffices to show that there exists an $\bar{\alpha}_k > 0$ such that

$$F^\alpha(x_{k+1}, \gamma \epsilon_k) \leq C_k - \sigma \alpha^2 \|d_k\|^2 + \epsilon_k,$$  

(36)

for all $\alpha \in (0, \bar{\alpha}_k)$.

In fact, using (13) and the induction hypothesis, we have that for all $\alpha > 0$,

$$F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) \leq F(x_k + \alpha d_k) + \gamma \epsilon_k,$$

$$F(x_k) \leq F^\alpha(x_k, \gamma \epsilon_{k-1}) \leq C_k.$$  

Combining the above two inequalities, we obtain

$$F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) \leq C_k + F(x_k + \alpha d_k) + \gamma \epsilon_k - F(x_k)$$

$$= C_k - \sigma \alpha^2 \|d_k\|^2 + \gamma \epsilon_k + F(x_k + \alpha d_k) - F(x_k) + \sigma \alpha^2 \|d_k\|^2.$$  

(37)

Define

$$\varphi(\alpha) = F(x_k + \alpha d_k) - F(x_k) + \sigma \alpha^2 \|d_k\|^2.$$  

(38)

Since $F$ is continuous, we have

$$\lim_{\alpha \to 0^+} \varphi(\alpha) = 0 < \epsilon_k - \gamma \epsilon_k,$$

which, together with (38), implies that there exists an $\bar{\alpha}_k > 0$ such that

$$\gamma \epsilon_k + F(x_k + \alpha d_k) - F(x_k) + \sigma \alpha^2 \|d_k\|^2 < \epsilon_k, \ \forall \alpha \in (0, \bar{\alpha}_k).$$  

(39)
Combining (39) with (37) yields
\[
F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) \leq C_k - \sigma \alpha^2 \|d_k\|^2 + \epsilon_k,
\]
for all \( \alpha \in (0, \tilde{\alpha}_k) \), where \( \tilde{\alpha}_k = \min\{1, \tilde{\alpha}_k\} \). Hence, the desired conclusion (36) follows directly from (40). This completes the proof.

From this proof process of Lemma 4.1, we can conclude that the following conclusion holds.

**Lemma 4.2.** The modified line search scheme (24) is well defined, i.e., there exists an \( \alpha_k \) satisfying the scheme (24).

**Lemma 4.3.** Let \( \{x_k\} \) be the sequence generated by Algorithm 3.1. Then we have
\[
C_{k+1} \leq C_k + \epsilon_k,
\]
(41)
and \( x_k \in \mathcal{L}_0 \) for all \( k \).

**Proof.** From the line search scheme (24) and (25), it follows that
\[
C_{k+1} = \frac{\eta_k E_k(C_k + \epsilon_k) + F^\alpha(x_{k+1}, \gamma \epsilon_k)}{E_k + \tilde{\epsilon}_k + E_{k+1}} \leq C_k + \epsilon_k, \quad \forall k,
\]
which implies that the first assertion (41) is true.

Now we prove the second assertion by induction. Clearly, \( x_0 \in \mathcal{L}_0 \), due to (13). Assume that \( x_k \in \mathcal{L}_0 \). Then it follows from (13), (33) and (41) that
\[
F(x_{k+1}) \leq F^\alpha(x_{k+1}, \gamma \epsilon_k) \leq C_{k+1} \leq C_k + \epsilon_k \leq \cdots \leq C_0 + \sum_{i=0}^k \epsilon_i \leq C_0 + \sum_{i=0}^{+\infty} \epsilon_i.
\]
Thus \( x_{k+1} \in \mathcal{L}_0 \), since \( C_0 = F^\alpha(x_0, \epsilon_0) \). This completes the proof.

**Lemma 4.4.** There exist two positive constants \( m \) and \( M \) such that
\[
\frac{w_k^T s_k}{\|s_k\|^2} \geq m, \quad \forall k,
\]
(42)
\[
\frac{\|w_k\|^2}{w_k^T s_k} \leq M, \quad \forall k.
\]
(43)

**Proof.** Clearly, it follows from Remark 2.1 and (19) that
\[
\frac{w_k^T s_k}{\|s_k\|^2} \geq \frac{t \|s_k\|^2}{\|s_k\|^2} = t,
\]
(44)
and
\[
\|w_k\| \leq \|\tilde{y}_k\| + t \|s_k\| + \frac{\|\tilde{y}_k^T s_k\|}{\|s_k\|^2} \|s_k\| \leq 2\|\tilde{y}_k\| + t \|s_k\|.
\]
(45)
By Remark 3.3, we have that \( \epsilon_k \leq \nu \alpha_k^2 \|d_k\|^2 = \nu \|s_k\|^2 \) for all \( k \). This together with (5) and (15) implies that
\[
\|\tilde{y}_k\| = \|g^\alpha(x_{k+1}, \gamma \epsilon_k) - g^\alpha(x_k, \epsilon_k)\| \leq \|g^\alpha(x_{k+1}, \gamma \epsilon_k) - g_{k+1}\| + \|g^\alpha(x_k, \epsilon_k) - g_k\| + \|g_k - g_{k+1}\| \leq \sqrt{\frac{2\gamma \epsilon_k}{\lambda}} + \sqrt{\frac{2\epsilon_k}{\lambda}} + \frac{\|s_k\|}{\lambda} \leq \frac{T \|s_k\|}{\lambda},
\]
(46)
where \( T = \sqrt{\frac{2\nu}{\lambda}} + \frac{2\nu}{\lambda} + \frac{1}{\lambda} \). Then, it follows from (45) and (46) that
\[
\frac{\|w_k\|^2}{w_k^T s_k} \leq \frac{4\|\bar{g}_k\|^2 + (t\|s_k\|)^2 + 4t\|\bar{g}_k\|\|s_k\|}{t\|s_k\|^2} \leq \frac{4T^2}{t} + t + 4T. \tag{47}
\]

Letting \( m = t \) and \( M = \frac{4T^2}{t} + t + 4T \), then the desired conclusions (42) and (43) follow directly from (44) and (47), respectively. This completes the proof. \( \square \)

**Lemma 4.5.** There exist two positive constants \( c_1 \) and \( c_2 \) such that
\[
c_1\|g^\alpha(x_k, \epsilon_k)\| \leq \|d_k\| \leq c_2\|g^\alpha(x_k, \epsilon_k)\|, \tag{48}\]
and
\[
g^\alpha(x_k, \epsilon_k)^T d_k \leq -c_1\|g^\alpha(x_k, \epsilon_k)\|^2, \tag{49}\]
for all \( k \).

**Proof.** Using the update formulas (17) and (23), we can easily obtain the trace of \( \bar{Q}_k \) and \( \bar{P}_k \) respectively as follows
\[
\text{tr}(\bar{Q}_{k+1}) = (n-2)\bar{\theta}_{k+1} + \left( 1 + \bar{\theta}_{k+1} \frac{\|w_k\|^2}{w_k^T s_k} \right) \bar{\theta}_{k+1}, \tag{50}\]
and
\[
\text{tr}(\bar{P}_{k+1}) = \frac{n-1}{\theta_{k+1}} + \frac{\|w_k\|^2}{w_k^T s_k}. \tag{51}\]
By (42) and (43), we get
\[
\frac{1}{M} \leq \frac{w_k^T s_k}{\|w_k\|^2} \leq \frac{\|s_k\|^2}{w_k^T s_k} = \bar{\theta}_{k+1} \leq \frac{1}{m}.
\]
This together with (43) and (50)-(51) implies that
\[
\text{tr}(\bar{Q}_k) \leq (n-1 + \frac{M}{m}) \frac{1}{m}, \forall k, \tag{52}\]
\[
\text{tr}(\bar{P}_k) \leq (n-1)M + M = nM, \forall k. \tag{53}\]
Let \( c_1 = \frac{1}{nM} \) and \( c_2 = \frac{m(n-1)+M}{m^2} \). Then, we have by using (16) and (52)-(53) that
\[
\|d_k\| \leq \|\bar{Q}_k\||g^\alpha(x_k, \epsilon_k)\| \leq \text{tr}(\bar{Q}_k)||g^\alpha(x_k, \epsilon_k)\| \leq c_2\|g^\alpha(x_k, \epsilon_k)\|, \forall k,
\]
\[
\|d_k\| \geq \frac{\|g^\alpha(x_k, \epsilon_k)\|}{\|\bar{P}_k\|} \geq \frac{\|g^\alpha(x_k, \epsilon_k)\|}{\text{tr}(\bar{P}_k)} \geq c_1\|g^\alpha(x_k, \epsilon_k)\|, \forall k,
\]
and
\[
-g^\alpha(x_k, \epsilon_k)^T d_k = g^\alpha(x_k, \epsilon_k)^T \bar{Q}_k g^\alpha(x_k, \epsilon_k)
\geq \frac{\|g^\alpha(x_k, \epsilon_k)\|^2}{\|\bar{P}_k\|}
\geq \frac{\|g^\alpha(x_k, \epsilon_k)\|^2}{\text{tr}(\bar{P}_k)}
\geq c_1\|g^\alpha(x_k, \epsilon_k)\|^2, \forall k.
\]
Combining the above three inequalities, we obtain the desired inequalities (48) and (49), respectively. \( \square \)

**Lemma 4.6.** There exists a constant \( \rho > 0 \) such that
\[
C_k - F^\alpha(x_{k+1}, \epsilon_k) \geq \rho\|g^\alpha(x_k, \epsilon_k)\|^2 - \epsilon_k, \forall k. \tag{54}\]
Proof. For simplicity, we define two index sets as follows.

$$K_1 = \{k|\alpha_k = 1\}, \quad K_2 = \{k|\alpha_k < 1\}.$$ 

The proof is divided into two parts.

At first, if $k \in K_1$, then, by the line search scheme (24) and (48), we have

$$C_k - F^\alpha(x_{k+1}, \gamma \epsilon_k) \geq \sigma \alpha_k^2 \|d_k\|^2 - \epsilon_k \geq \sigma \alpha_k^2 \|g^\alpha(x_k, \epsilon_k)\|^2 - \epsilon_k, \quad \forall k \in K_1. \quad (55)$$

Second, if $k \in K_2$, since $\alpha_k < 1$, we have $\frac{\alpha_k}{\beta} \in \{1, \beta, \beta^2, \ldots\}$. Let $\alpha = \frac{\alpha_k}{\beta}$. From the way $\alpha_k$ is chosen in the line search scheme (24), it follows that the stepsize $\alpha$ fails to satisfy (24), i.e.,

$$F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) > C_k - \sigma \alpha^2 \|d_k\|^2 + \epsilon_k, \quad \forall k \in K_2,$$

which, together with (33), implies that

$$F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) > F^\alpha(x_k, \gamma \epsilon_{k-1}) - \sigma \alpha^2 \|d_k\|^2 + \epsilon_k, \quad \forall k \in K_2. \quad (56)$$

By (13), we have

$$F^\alpha(x_k + \alpha d_k, \gamma \epsilon_k) - F^\alpha(x_k, \gamma \epsilon_{k-1}) \leq F(x_k + \alpha d_k) - F(x_k) + \gamma \epsilon_k. \quad (57)$$

Using the mean-value theorem, we can deduce that there exists $\theta \in [0, 1]$ such that

$$F(x_k + \alpha d_k) - F(x_k) = \alpha g(x_k + \theta \alpha d_k)^T d_k. \quad (58)$$

Combining (58) with (56)-(57) yields

$$\alpha g(x_k + \theta \alpha d_k)^T d_k > -\sigma \alpha^2 \|d_k\|^2 + \epsilon_k - \gamma \epsilon_k$$

$$> -\sigma \alpha^2 \|d_k\|^2, \quad \forall k \in K_2,$$

i.e.,

$$g(x_k + \theta \alpha d_k)^T d_k > -\sigma \|d_k\|^2, \quad \forall k \in K_2. \quad (59)$$

This together with (15) implies that

$$(g(x_k + \theta \alpha d_k) - g(x_k))^T d_k \geq -\sigma \alpha \|d_k\|^2 - g(x_k)^T d_k$$

$$= -\sigma \alpha \|d_k\|^2 - g^\alpha(x_k, \epsilon_k)^T d_k$$

$$+ (g^\alpha(x_k, \epsilon_k) - g(x_k))^T d_k$$

$$\geq -\sigma \alpha \|d_k\|^2 - g^\alpha(x_k, \epsilon_k)^T d_k$$

$$- \|g^\alpha(x_k, \epsilon_k) - g(x_k)\| \|d_k\|$$

$$\geq -\sigma \alpha \|d_k\|^2 - g^\alpha(x_k, \epsilon_k)^T d_k$$

$$- \sqrt{\frac{2\epsilon_k}{\lambda}} \|d_k\|, \quad \forall k \in K_2. \quad (60)$$

By (5) and the Cauchy-Schwarz inequality, we obtain

$$g(x_k + \theta \alpha d_k)^T d_k - g(x_k)^T d_k \leq \frac{\theta \alpha}{\lambda} \|d_k\|^2 \leq \frac{\alpha}{\lambda} \|d_k\|^2. \quad (61)$$

Then, it follows from (60) and (61) that

$$\frac{1}{\lambda} + \sigma \alpha \|d_k\|^2 \geq -g^\alpha(x_k, \epsilon_k)^T d_k - \sqrt{\frac{2\epsilon_k}{\lambda}} \|d_k\|, \quad \forall k \in K_2. \quad (62)$$

Since $\epsilon_k \leq \nu \alpha_k^2 \|d_k\|^2$ by Remark 3.3, the inequality (62) and $\alpha = \frac{\alpha_k}{\beta}$ imply

$$\frac{1 + \lambda \sigma}{\lambda \beta} \alpha_k \|d_k\|^2 \geq -g^\alpha(x_k, \epsilon_k)^T d_k - \sqrt{\frac{2\epsilon_k}{\lambda}} \alpha_k \|d_k\|^2, \quad \forall k \in K_2.$$
A rearrangement of the terms yields
\[
(1 + \frac{\lambda \sigma}{\lambda \beta} + \sqrt{\frac{2 \nu}{\lambda}}) \alpha_k \|d_k\|^2 \geq -g^\alpha(x_k, \epsilon_k)^T d_k, \quad \forall k \in K_2,
\]
(63)
which further implies that
\[
\alpha_k \geq \frac{\lambda \beta}{1 + \lambda \sigma + \beta \sqrt{2 \nu}} \left[ -\frac{g^\alpha(x_k, \epsilon_k)^T d_k}{\|d_k\|^2} \right], \quad \forall k \in K_2.
\]
(64)
Substituting (64) into the line search scheme (24), we obtain
\[
C_k - F^\alpha(x_{k+1}, \gamma \epsilon_k) \geq \sigma \alpha_k^2 \|d_k\|^2 - \epsilon_k
\]
\[
\geq \frac{\sigma \lambda^2 \beta^2}{(1 + \lambda \sigma + \beta \sqrt{2 \nu})^2} \left( -\frac{g^\alpha(x_k, \epsilon_k)^T d_k}{\|d_k\|^2} \right)^2 - \epsilon_k
\]
for all $k \in K_2$. From (48) and (49), it follows that there exists a constant $c_3 = \frac{\sigma \lambda^2 \beta^2}{(1 + \lambda \sigma + \beta \sqrt{2 \nu})^2} > 0$ such that
\[
-\frac{g^\alpha(x_k, \epsilon_k)^T d_k}{\|d_k\|^2} \geq c_3 \|g^\alpha(x_k, \epsilon_k)\|, \quad \forall k.
\]
This together with (65) implies that
\[
C_k - F^\alpha(x_{k+1}, \gamma \epsilon_k) \geq c_4 \|g^\alpha(x_k, \epsilon_k)\|^2 - \epsilon_k, \quad \forall k \in K_2,
\]
(66)
where $c_4 = \frac{\sigma \lambda^2 \beta^2}{(1 + \lambda \sigma + \beta \sqrt{2 \nu})^2}$. By letting $\rho = \min\{\sigma \lambda^2, c_4\}$, then the desired inequality (54) follows directly from (55) and (66). This completes the proof. \qed

Using these lemmas mentioned above, we obtain the global convergence result of Algorithm 3.1 as follows.

**Theorem 4.7.** Suppose that Assumption A holds. If Algorithm 3.1 generates an infinite sequence $\{x_k\}$, then
\[
\lim_{k \to +\infty} \|g^\alpha(x_k, \epsilon_k)\| = 0.
\]
(67)
Furthermore, any accumulation point of $\{x_k\}$ is an optimal solution of problem (1.1).

**Proof.** We first show that the assertion (67) holds. From (25) and (54), it follows that
\[
C_{k+1} = \frac{\eta_k E_k}{E_{k+1}} C_k + \frac{1}{E_{k+1}} F^\alpha(x_{k+1}, \gamma \epsilon_k) + \frac{\eta_k E_k}{E_{k+1}} \epsilon_k
\]
\[
\leq \frac{\eta_k E_k}{E_{k+1}} C_k + \frac{1}{E_{k+1}} \left( C_k - \rho \|g^\alpha(x_k, \epsilon_k)\|^2 + \epsilon_k \right) + \frac{\eta_k E_k}{E_{k+1}} \epsilon_k
\]
\[
= C_k - \frac{\rho}{E_{k+1}} \|g^\alpha(x_k, \epsilon_k)\|^2 + \epsilon_k, \quad \forall k.
\]
(68)
Note that
\[
E_{k+1} = 1 + \sum_{i=0}^k \prod_{j=i}^k \eta_{k-j} \leq 1 + \sum_{j=0}^k \eta_{\max}^{j+1} \leq \sum_{j=0}^k \eta_{\max} = \frac{1}{1 - \eta_{\max}},
\]
(69)
due to $\eta_{\max} < 1$. Consequently, it follows from (68) and (69) that
\[
\epsilon_k + C_k - C_{k+1} \geq \rho (1 - \eta_{\max}) \|g^\alpha(x_k, \epsilon_k)\|^2, \quad \forall k.
\]
(70)
Since $x_k \in L_0$, it follows from Assumption A that $F(x_k)$ is bounded from below, so is $C_k$ due to the fact that $C_k \geq F^\alpha(x_k, \gamma \epsilon_{k-1}) \geq F(x_k)$ for all $k$. Furthermore, by using $\epsilon_{i+1} = \gamma \epsilon_i$ for all $i$, we have
\[
\sum_{i=0}^{+\infty} \epsilon_i \leq \frac{\epsilon_0}{1 - \gamma} < \infty.
\]
Combining this inequality with (70) yields
$$\sum_{k=0}^{+\infty} \|g^\alpha(x_k, \epsilon_k)\|^2 < \infty,$$
which implies that the assertion (67) holds.

To prove the second part of Theorem 4.7, let $x^*$ be an arbitrary accumulation point of $\{x_k\}$. Then there exists a subsequence $\{x_k\}_{k \in \mathcal{K}}$ satisfying
$$\lim_{k \in \mathcal{K}, k \to +\infty} x_k = x^*. \tag{71}$$
By (5) and (15), we have
$$\|g^\alpha(x_k, \epsilon_k) - g(x^*)\| \leq \|g^\alpha(x_k, \epsilon_k) - g(x_k)\| + \|g(x_k) - g(x^*)\| \leq \sqrt{2\epsilon_k/\lambda} + \|x_k - x^*\|. \tag{72}$$
From (71), (72), (67) and $\lim_{k \to +\infty} \epsilon_k = 0$, it follows that
$$g(x^*) = 0.$$
Thus, $x^*$ minimizes $F$ by using Proposition 2.2. This proof of the second part is completed by noting the equivalence of problems (1) and (2).

Based on the above arguments, the desired conclusions follow. This completes the proof. □

5. **Numerical experiments.** In this section, we present some numerical experiments to evaluate the performance of the proposed method on both small scale problems and large scale problems. At the same time, we give some comparisons with the related methods.

5.1. **Numerical experiments for small scale problems.** We first test the proposed algorithm for small scale problems which are chosen from [21] and listed in Table 1, where $n$, $x_0$ and $f_{\text{op}}$ refer to the dimension of variables, initial points and optimum function values, respectively.

We implemented our algorithm with the codes written in Matlab 7.0. The testing is performed on a PC computer with HPdx2810SE Pentium(R) Dual-Core CPU E5300 @ 2.60GHZ 2.00GB. Throughout the computational experiments, the parameters used in Algorithm 3.1 are chosen as follows: $\gamma = 0.85$, $\sigma = 0.001$, $\beta = 0.6$, $\lambda = 100$, $\nu = 1$, $\epsilon_{-1} = 1$, $t = 0.01$, and $\eta_k \equiv 0.85$. We stop the algorithm when the condition $\|g^\alpha(x_k, \epsilon_k)\| \leq 10^{-6}$ is satisfied.

To validate Algorithm 3.1 from a computational point of view, we compare it with the algorithm in [20] (abbreviated LWTR), the algorithm in [32] (abbreviated YWBB), the algorithm in [26] (abbreviated SFTR), and the algorithm in [25] (abbreviated RFBFGS). For the algorithm SFTR, the related parameters used are chosen as follows: $\beta = 1$, $\gamma = 0.5$, $\eta = 0.5$, $\Delta_0 = 1$, $\mu = 0.0001$, and $\lambda = 100$; For the algorithm RFBFGS, the related parameters used are given as follows: $\rho = 0.5$, $\sigma = 0.001$, $\epsilon_{-1} = 1$, $B_0 = I$, $\gamma = 0.85$ and $\beta_k = \frac{1}{(k+1)^2}$.

Table 2 shows the detailed numerical results which are given in the form of $n_i/n_f/f_k$, where $n_i$, $n_f$ and $f_k$ denote the number of iterations, the number of function evaluations and the function value $f(x_k)$ at the final iteration, respectively. It should be mentioned that the numerical results of the algorithms LWTR and YWBB can be found in [20, 32], respectively.
Table 1. Testing functions for small scale problems

| No. | Functions       | n | $x_0$          | $f_{ops}(x)$ |
|-----|-----------------|---|----------------|--------------|
| 1   | Rosenbrock      | 2 | (-1.2; 1)      | 0            |
| 2   | Crescent        | 2 | (-1.5; 2)      | 0            |
| 3   | CB2             | 2 | (1; -0.1)      | 1.9522245    |
| 4   | CB3             | 2 | (2; 2)         | 2.0          |
| 5   | DEM             | 2 | (1; 1)         | -3           |
| 6   | QL              | 2 | (-1; 5)        | 7.2          |
| 7   | LQ              | 2 | (-0.5; -0.5)   | -1.4142136   |
| 8   | Mifflin 1       | 2 | (0.8; 0.6)     | -1.0         |
| 9   | Mifflin 2       | 2 | (-1; -1)       | -1.0         |
| 10  | Wolfe           | 2 | (3; 2)         | -8           |
| 11  | Rosen-Suzuki    | 4 | (0; 0; 0; 0)   | -44          |
| 12  | Shor            | 5 | (0; 0; 0; 0; 1)| 22.600162    |

Table 2. Numerical results for five algorithms

| No. | Algorithm 3.1 | $LWTR$          | $YWBB$        | $SFTR$        | $RFBFGS$       |
|-----|----------------|-----------------|---------------|---------------|----------------|
| 1   | 3/4/2.6178e-9 | 6/13/0.2976e-7 | 54/56/3.4484e-7| 48/49/7.1545e-4| 4/5/6.2072e-10|
| 2   | 3/4/3.3026e-6 | 3/7/6.5430e-4  | 14/16/2.7450e-5| 31/32/1.6000e-3| 35/36/3.0590e-7|
| 3   | 4/5/1.9522    | 5/12/1.9522    | 13/15/1.9522  | 54/55/1.9573  | 5/6/1.9522     |
| 4   | 2/3/2.0000    | 6/13/2.0000    | 4/8/2.0000    | 55/56/2.0100  | 2/3/2.0076     |
| 5   | 3/4/-3.0000   | 8/16/-3.0000   | 4/7/-3.0000   | 5/6/-3.0000   | 3/4/-3.0000    |
| 6   | 11/12/7.2000  | 4/9/7.2000     | 22/25/7.2000  | 48/49/7.2003  | 10/11/7.2000   |
| 7   | 3/4/-1.4142   | 5/10/-1.4142   | 6/7/-1.4142   | 3/4/-1.4118   | 2/3/-1.4033    |
| 8   | 34/35/-1.0000 | 15/31/-0.9938  | 3/6/-1.0000   | 59/60/-1.0000 | 57/58/-1.0000  |
| 9   | 2/3/-1.0000   | 8/16/-1.0000   | 12/13/-0.9999 | 4/5/-0.9997   | 2/3/-0.9813    |
| 10  | 3/4/-8.0000   | 12/24/-8.0000  | 9/12/-8.0000  | 43/46/-8.0000 | 4/5/-8.0000    |
| 11  | 49/106/-43.9999| 20/40/-44.0000| 8/9/-43.9493  | 60/61/-39.9924| 25/31/-43.9982|
| 12  | 14/16/22.6019 | 14/28/22.6002  | 9/10/22.6004  | 71/72/22.6892 | 66/152/22.6017|

We adopt the performance profile introduced by Dolan and More [10] to compare the efficiency among the five different methods on the set of test problems, which has some advantages over other existing benchmarking tools. Figure 1 gives the performance profiles of the five methods for the number of iterations and function evaluations, respectively. From Figure 1, we observe that for the test problems in
Table 1, Algorithm 3.1 performs better than the algorithms LWTR, YWBB, SFTR and RFBFGS in terms of the number of iterations and function evaluations.

5.2. Numerical experiments for large scale problems. Since the conjugate gradient-type methods are particularly efficient for large scale problems, in the second set of numerical experiments, we present some numerical experiments for large scale nonsmooth problems which are chosen from [13] and listed in Table 3. Throughout the computational experiments, the values of parameters used are similar to that used in small scale problems. In order to show the performance of the proposed algorithm, we also list the testing results of paper [33] (abbreviated CG-YWL), which were implemented in Fortran 90 and could be found in [33].

Table 4 shows the detailed testing results which are given in the form of $n_i/n_f/f_k$, where their meanings are the same as those in Table 2. Efficiency comparisons are also made by using the performance profile [10]. The performance profiles in Figure 2 compare Algorithm 3.1 with CG-YWL in terms of the number of iterations and function evaluations, respectively. From Figure 2, we see that for the test problems in Table 3, Algorithm 3.1 can be competitive with CG-YWL.

While it would be unwise to draw any firm conclusions from the limited numerical
results, they indicate some promise for the new approach proposed in this paper. Further improvement is expected from more suitable implementation.

Table 3. Testing functions for large scale problems

| No. | Functions                              | Initial points $x_0$                              |
|-----|----------------------------------------|-------------------------------------------------|
| 1   | Generalization of MAXQ                 | $(1, 2, \ldots, \frac{n}{2}, -\frac{n}{2}, 1, \ldots, -n)$ |
| 2   | Generalization of MXHILB              | $(1, 1, \ldots, 1)$                              |
| 3   | Chained LQ                             | $(-0.5, -0.5, \ldots, -0.5)$                    |
| 4   | Number of active faces                 | $(1, 1, \ldots, 1)$                              |
| 5   | Nonsmooth generalization of Brown 2    | $(1, 0, 1, 0, \ldots)$                          |
| 6   | Chained Mifflin 2                      | $(-1, -1, \ldots, -1)$                          |
| 7   | Chained Crescent I                     | $(-1.5, 2, -1.5, 2, \ldots)$                    |
| 8   | Chained Crescent II                    | $(1, 1, 0, \ldots)$                             |

Table 4. Numerical results for Algorithms 3.1 and CG-YWL

| No. | n   | Algorithm 3.1 | CG – YWL |
|-----|-----|---------------|----------|
| 1   | 1000| 186/1601/2.6568e − 10 | 225/4710/6.9354e − 8 |
|     | 5000| 242/2725/1.2183e − 10 | 250/5235/6.8798e − 8 |
|     | 10000| 253/2997/3.4045e − 10 | 261/5466/6.6528e − 8 |
| 2   | 1000| 94/1301/4.0582e − 9   | 91/1482/8.2738e − 9 |
|     | 5000| 119/1499/2.6731e − 9   | 111/1938/9.7206e − 9 |
|     | 10000| 129/1901/2.1905e − 9   | 120/2127/5.8524e − 9 |
| 3   | 1000| 37/110/2.3278e − 9     | 37/114/7.2687e − 9 |
|     | 5000| 39/116/5.9987e − 9     | 39/120/9.0932e − 9 |
|     | 10000| 40/121/1.6943e − 9     | 40/123/9.0941e − 9 |
| 4   | 1000| 71/89/3.6735e − 11     | 77/1026/6.8037e − 9 |
|     | 5000| 82/93/7.9864e − 11     | 90/1281/7.8405e − 9 |
|     | 10000| 86/1208/5.7163e − 11   | 96/1401/9.9366e − 9 |
| 5   | 1000| 35/114/1.5021e − 11    | 38/117/7.2687e − 9 |
|     | 5000| 39/120/5.2352e − 11    | 40/123/9.0932e − 9 |
|     | 10000| 41/126/2.1136e − 11    | 41/125/1.8188e − 8 |
| 6   | 1000| 38/116/−5.3979e+3      | 37/114/−2.4975e+4 |
|     | 5000| 41/123/−3.2153e+4      | 39/120/−1.2498e+5 |
|     | 10000| 43/129/−2.0127e+4      | 40/123/−2.4998e+5 |
| 7   | 1000| 34/101/7.0463e − 11    | 37/114/5.4897e − 9 |
|     | 5000| 36/113/3.8145e − 11    | 39/120/6.8294e − 9 |
|     | 10000| 39/120/4.3654e − 11    | 40/123/6.8253e − 9 |
| 8   | 1000| 37/112/6.0424e − 11    | 39/120/6.8185e − 9 |
|     | 5000| 39/121/1.8205e − 11    | 41/126/8.5258e − 9 |
|     | 10000| 42/125/2.6473e − 11    | 42/129/8.5262e − 9 |

6. Concluding remarks. In this paper, we have proposed an implementable non-monotone scaled memoryless BFGS preconditioned conjugate gradient algorithm for solving the nonsmooth convex optimization problem (1), based on a modified line search technique and the Moreau-Yosida regularization. The proposed method
makes use of approximate function and gradient values of the Moreau-Yosida regularization instead of the corresponding exact values. Under some reasonable assumptions, the global convergence of the proposed algorithm is established. Preliminary numerical results and related comparisons show that the proposed method can be applied to solve large scale nonsmooth convex optimization problems.

Note that the proposed algorithm needs to compute a vector $p^\alpha(x,\epsilon)$ for given $x$ and $\epsilon$. Though some existing approaches can be used as a procedure to obtain such a vector, it tends to spend more iterations when $\epsilon$ is small. Thus, it would be interesting to study some more efficient methods of choosing $p^\alpha(x,\epsilon)$ to improve the efficiency of the proposed algorithms in the future research work.

**Acknowledgments.** The authors are very grateful to the anonymous referees and the associate editor for their valuable comments and suggestions that greatly improved this paper.

**REFERENCES**

[1] N. Andrei, Scaled conjugate gradient algorithms for unconstrained optimization, *Computational Optimization and Applications*, 38 (2007), 401–416.

[2] A. Auslender, Numerical methods for nondifferentiable convex optimization, *Mathematical Programming Study*, 30 (1987), 102–126.

[3] S. Babaie-Kafaki, A modified scaled memoryless BFGS preconditioned conjugate gradient method for unconstrained optimization, *4OR, A Quarterly Journal of Operations Research*, 11 (2013), 361–374.

[4] S. Babaie-Kafaki and R. Chambari, A class of descent four-term extension of the Dai-Liao conjugate gradient method based on the scaled memoryless BFGS update, *Journal of Industrial and Management Optimization*, 13 (2017), 649–658.

[5] J. Barzilai and J. M. Borwein, Two-point stepsize gradient methods, *IMA Journal of Numerical Analysis*, 8 (1988), 141–148.

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

[7] J. F. Bonnans, J. C. Gilbert, C. Lemarechal and C. Sagastizabal, A family of variable-metric proximal methods, *Mathemtical Programming*, 68 (1995), 15–47.

[8] J. V. Burke and M. Qian, On the superlinear convergence of the variable metric proximal point algorithm using Broyden and BFGS matrix secant updating, *Mathematical Programming*, 88 (2000), 157–181.

[9] X. Chen and M. Fukushima, Proximal quasi-Newton methods for nondifferentiable convex optimization, *Mathematical Programming*, 85 (1999), 313–334.

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

[11] M. Fukushima, A descent algorithm for nonsmooth convex optimization, *Mathematical Programming*, 30 (1984), 163–175.

[12] M. Fukushima and L. Q. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM Journal on Optimization*, 6 (1996), 1106–1120.

[13] M. Haarala, K. Miettinen and M. M. Mäkelä, New limited memory bundle method for large-scale nonsmooth optimization, *Optimization Methods and Software*, 19 (2004), 673–692.

[14] M. Haarala, K. Miettinen and M. M. Mäkelä, Globally convergent limited memory bundle method for large-scale nonsmooth optimization, *Mathematical Programming*, 109 (2007), 181–205.

[15] W. W. Hager and H. C. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific Journal of Optimization*, 2 (2006), 35–58.

[16] J. B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer, Berlin, 1993.

[17] C. Lemarechal and C. Sagastizabal, Practical aspects of the Moreau-Yosida regularization, I: Theoretical preliminaries, *SIAM Journal on Optimization*, 7 (1997), 367–385.

[18] Q. Li, Conjugate gradient type methods for the nondifferentiable convex minimization, *Optimization Letters*, 7 (2013), 533–545.
A MODIFIED SCALCG METHOD FOR NONSMOOTH OPTIMIZATION

[19] D. H. Li and M. Fukushima, A derivative-free line search and global convergence of Broyden-like method for nonlinear equations, *Optimization Methods and Software*, 13 (2000), 181–201.

[20] S. Lu, Z. X. Wei and L. Li, A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization, *Computational Optimization and Applications*, 51 (2012), 551–573.

[21] L. Lukšan and J. Vlček, Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, Technical Report No. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[22] R. Mifflin, A quasi-second-order proximal bundle algorithm, *Mathematical Programming*, 73 (1996), 51–72.

[23] Y. G. Ou and H. C. Lin, An ODE-like nonmonotone method for nonsmooth convex optimization, *Journal of Applied Mathematics and Computing*, 52 (2016), 265–285.

[24] L. Q. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research*, 18 (1993), 227–244.

[25] A. I. Rauf and M. Fukushima, A globally convergent BFGS method for nonsmooth convex optimization, *Journal of Optimization Theory and Applications*, 104 (2000), 539–558.

[26] N. Sagara and M. Fukushima, A trust region method for nonsmooth convex optimization, *Journal of Industrial and Management Optimization*, 1 (2005), 171–180.

[27] D. F. Shanno, On the convergence of a new conjugate gradient algorithm, *SIAM Journal on Numerical Analysis*, 15 (1978), 1247–1257.

[28] J. Shen, L. P. Pang and D. Li, An approximate quasi-Newton bundle-type method for nonsmooth optimization, *Abstract and Applied Analysis*, 2013, Art. ID 697474, 7 pp.

[29] W. Y. Sun and Y. X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer, New York, 2006.

[30] G. L. Yuan, Z. H. Meng and Y. Li, A modified Hestenes and Stiefel conjugate gradient algorithm for large scale nonsmooth minimizations and nonlinear equations, *Journal of Optimization Theory and Applications*, 168 (2016), 129–152.

[31] G. L. Yuan, Z. Sheng and W. J. Liu, The modified HZ conjugate gradient algorithm for large scale nonsmooth optimization, *Plos One*, 11 (2016), e0164289, 15pp.

[32] G. L. Yuan and Z. X. Wei, The Barzilai and Borwein gradient method with nonmonotone line search for nonsmooth convex optimization problems, *Mathematical Modelling and Analysis*, 17 (2012), 203–216.

[33] G. L. Yuan, Z. X. Wei and G. Y. Li, A modified Polak-Ribière-Polyak conjugate gradient algorithm for nonsmooth convex programs, *Journal of Computational and Applied mathematics*, 255 (2014), 86–96.

[34] G. L. Yuan, Z. X. Wei and Z. X. Wang, Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minization, *Computational Optimization and Applications*, 54 (2013), 45–64.

[35] H. C. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM Journal on Optimization*, 14 (2004), 1043–1056.

Received January 2017; revised March 2017.

E-mail address: ouyigui@126.com
E-mail address: 876899406@qq.com