J. Appl. Math. Comput. (2016) 51:397–412
DOI 10.1007/s12190-015-0912-8

ORIGINAL ARTICLE

A modified PRP conjugate gradient algorithm with nonmonotone line search for nonsmooth convex optimization problems

Gonglin Yuan[1] · **Zengxin Wei**[1]

Received: 29 December 2010 / Published online: 5 August 2015

Abstract It is well-known that nonlinear conjugate gradient (CG) methods are preferred to solve large-scale smooth optimization problems due to their simplicity and low storage. However, the CG methods for nonsmooth optimization have not been studied. In this paper, a modified Polak–Ribière–Polyak CG algorithm which combines with a nonmonotone line search technique is proposed for nonsmooth convex minimization. The search direction of the given method not only possesses the sufficiently descent property but also belongs to a trust region. Moreover, the search direction has not only the gradients information but also the functions information. The global convergence of the presented algorithm is established under suitable conditions. Numerical results show that the given method is competitive to other three methods.

Keywords Nonsmooth convex minimization · Conjugate gradient · Nonmonotone technique · Global convergence

Mathematics Subject Classification 65K05 · 90C30

✉ Zengxin Wei
zxwei@gxu.edu.cn

Gonglin Yuan
glyuan@gxu.edu.cn

[1] The Guangxi Colleges and Universities Key Laboratory of Mathematics and Its Applications, Department of Mathematics and Information Science, Guangxi University, Nanning 530004, Guangxi, People's Republic of China

1 Introduction

Consider

$$\min_{x\in\Re^n} f(x), \tag{1.1}$$

where $f:\Re^n\to\Re$ is a possibly nonsmooth convex function. The problem (1.1) is equivalent to the following problem

$$\min_{x\in\Re^n} F(x), \tag{1.2}$$

where $F:\Re^n\to\Re$ is the so-called Moreau-Yosida regularization of f and defined by

$$F(x)=\min_{z\in\Re^n}\left\{f(z)+\frac{1}{2\lambda}\|z-x\|^2\right\}, \tag{1.3}$$

where $\lambda>0$ is a parameter and $\|\cdot\|$ denotes the Euclidean norm. A remarkable feature of problem (1.2) is that the objective function F is a differentiable convex function, even when the function f is nondifferentiable. Furthermore, F has a Lipschitz continuous gradient. But, in general, F is not twice differentiable [23]. However, it is shown that, under some reasonable conditions, the gradient function of F can be proved to be semismooth [9,32]. Based on these features, many algorithms have been proposed for (1.2) (see [3,9,32] etc.). The proximal methods have been proved to be effective in dealing with the difficulty of evaluating the function value of $F(x)$ and its gradient $\nabla F(x)$ at a given point x (see [5,7,19]). Lemaréchal [21] and Wolfe [39] initiated a giant stride forward in nonsmooth optimization by the bundle concept, which can handle convex and nonconvex f. All bundle methods carry two distinctive features: (i) They make use at the iterate x_k of the bundle of information $(f(x_k),\ g(x_k)),\ (f(x_{k-1}), g(x_{k-1})),\ldots$ collected so far to build up a model of f; (ii) If, due to the kinky structure of f, this model is not yet an adequate one, then they mobilize even more subgradient information close to x_k (or see Lemaréchal [22] and Zowe [48] in detail), where x_k the kth iteration and $g(x_k)\in\partial f(x_k)$ ($\partial f(x)$ is the subdifferential of f at x). Some further results can be found (see [18,20,33,34] etc.). Recently, Yuan et al. [43] gave the conjugate gradient algorithm for large-scale nonsmooth problems and get some good results. In this paper, we will provide a new way to solve (1.2). The idea is motivated by the differentiability of F. We all know that there are many efficient methods for smooth optimization problems, where the nonlinear CG method is one of them. However the CG method for nonsmooth problems has not been studied. Considering the differentiability of $F(x)$ and the same solution set of (1.1) and (1.2), we present a CG method for (1.2) instead of solving $f(x)$.

The nonlinear conjugate gradient method is one of the most effective line search methods for smooth unconstrained optimization problem

$$\min_{v\in\Re^n} h(v), \tag{1.4}$$

where $h : \Re^n \to \Re$ is continuously differentiable. The CG method has simplicity, low storage, practical computation efficiency and nice convergence properties. The PRP (β^{PRP}, see [27,28]) method is one of the most efficient CG methods. It has been further studied by many authors (see [8,10,12,29,30] etc.). The sufficiently descent condition is often used to analyze the global convergence of the nonlinear conjugate gradient method. Many authors hinted that the sufficiently descent condition may be crucial for conjugate gradient methods [1,2]. There are many modified nonlinear conjugate gradient formulas which possesses the sufficiently descent property without any line search (see [14,15,36,41,46] etc.). At present, the CG methods are only used to solve smooth optimization problems. Whether can the CG method be extent to solve nonsmooth problem (1.2)? We answer this question positively. Another effective method for unconstrained optimization (1.4) is quasi-Newton secant methods which obey the recursive formula

$$v_{k+1} = v_k - B_k^{-1}\nabla h(v_k),$$

where B_k is an approximation Hessian of h at v_k. The sequence of matrix $\{B_k\}$ satisfies the secant equation

$$B_{k+1}S_k = \delta_k, \tag{1.5}$$

where $S_k = v_{k+1} - v_k$ and $\delta_k = \nabla h(v_{k+1}) - \nabla h(v_k)$. Obviously, only two gradients are exploited in the secant equation (1.5), while the function values available are neglected. Hence, techniques using gradients values as well as function values have been studied by several authors. An efficient attempt is due to Zhang et al. [44]. They developed a new secant equation which used both gradient values and function values. This equation is defined by

$$B_{k+1}S_k = \delta_k^1, \tag{1.6}$$

where $\delta_k^1 = \delta_k + \gamma_k^1 S_k$ and $\gamma_k^1 = \frac{3(\nabla h(v_{k+1})+\nabla h(v_k))^T S_k + 6(h(v_k)-h(v_{k+1}))}{\|S_k\|^2}$. The new secant equation is superior to the usual one (1.5) in the sense δ_k^1 better approximates $\nabla^2 h(v_{k+1})S_k$ than δ_k. Consequently, the matrix which is obtained from the modified quasi-Newton update better approximates the Hessian matrix (see [44] in detail). Another significant attempt is due to Wei et al. (see [37]) and the equation is defined by

$$B_{k+1}S_k = \delta_k^2, \tag{1.7}$$

where $\delta_k^2 = \delta_k + \gamma_k^2 S_k$ and $\gamma_k^2 = \frac{(\nabla h(v_{k+1})+\nabla h(v_k))^T S_k + 3(h(v_k)-h(v_{k+1}))}{\|S_k\|^2}$. A remarkable property of this secant equation (1.7) is that, if h is twice continuously differentiable and B_{k+1} is updated by the BFGS method, then the equality

$$h(v_k) = h(v_{k+1}) + \nabla h(v_{k+1})^T S_k + \frac{1}{2}S_k^T B_{k+1} S_k$$

holds for all k. Moreover this property is independent of any convexity assumption on the objective function. Furthermore, this equality does not hold for any update formula

which is based on the usual secant condition (1.5), even for the new one (1.6). Additionally, comparing with the secant equation (1.6), one concludes that $\gamma_k^2 = \frac{1}{3}\gamma_k^1$. This is a very interesting fact. The superlinear convergence theorem of the corresponding BFGS method was established in [38]. Moreover, the work of [38] was extended to deal with large-scale problems in a limited memory scheme in [40]. The reported numerical results show that this extension is beneficial to the performance of the algorithm. However, the method based on (1.6) does not possess the global convergence and the superlinear convergence for general convex functions. In order to overcome this drawback, Yuan and Wei [42] presented the following secant equation

$$B_{k+1}S_k = \delta_k^3, \tag{1.8}$$

where $\delta_k^3 = \delta_k + \gamma_k^3 S_k$ and $\gamma_k^3 = \max\left\{0, \frac{(\nabla h(v_{k+1})+\nabla h(v_k))^T S_k + 3(h(v_k)-h(v_{k+1}))}{\|S_k\|^2}\right\}$. Numerical results show that this method is competitive to other quasi-Newton methods (see [42]). Can these modified quasi-Newton methods be used in CG methods and extent to nonsmooth problems? One directive way is to replace the normal δ_k by modified δ_k^1 (δ_k^2 or δ_k^3) in CG formulas. Considering this view, we will present a modified CG method for nonsmooth problem (1.2), where the CG formula possesses not only the gradient values but also the function values.

The line search framework is often used in smooth and nonsmooth fields, where the earliest nonmonotone line search technique was developed by Grippo, Lampariello, and Lucidi in [11] for Newton's methods. Many subsequent papers have exploited nonmonotone line search techniques of this nature (see [4,16,24,47] etc.). Although these nonmonotone technique work well in many cases, there are some drawbacks. First, a good function value generated in any iteration is essentially discarded due to the max in the nonmonotone line search technique. Second, in some cases, the numerical performance is very dependent of the choice of M (see [11,35]). In order to overcome these two drawbacks, Zhang and Hager [45] presented a new nonmonotone line search technique. Numerical results show that this technique is better than the normal nonmonotone technique and the monotone technique.

Motivated by the above observations, we will present a modified PRP conjugate gradient method which combines with a nonmonotone line search technique for (1.2). The main characteristics of this method are as follows.

- A conjugate gradient method is introduced for nonsmooth problem (1.1) and (1.2).
- All search directions are sufficient descent, which shows that the function values are decreasing.
- All search directions are in a trust region, which hints that this method has good convergent results.
- The global convergence is established under suitable conditions.
- Numerical results show that this method is competitive to other three methods.

This paper is organized as follows. In the next section, we briefly review some known results of convex analysis and nonsmooth analysis. In Sect. 3, we deduce motivation of the search direction and the given algorithm. In Sect. 4, we prove the global convergence of the proposed method. Numerical results are reported in Sect. 5 and one

conclusion is given in the last section. Throughout this paper, without specification, $\|\cdot\|$ denotes the Euclidean norm of vectors or matrices.

2 Results of convex analysis and nonsmooth analysis

Some basic results in convex analysis and nonsmooth analysis, which will be used later, are reviewed in this section. Let

$$\theta(z) = f(z) + \frac{1}{2\lambda}\|z - x\|^2$$

and denote $p(x) = \text{argmin}\theta(z)$, where $\lambda > 0$ is a scalar. Then $p(x)$ is well-defined and unique since $\theta(z)$ is strongly convex. By (1.3), $F(x)$ can be expressed by

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

In what follows, we denote the gradient of F by g. Some features about $F(x)$ can be seen in [5,7,17]. The generalized Jacobian of $F(x)$ and the property of BD-regular can be found in [6,31] respectively. Some properties are given as follows.

(i) The function F is finite-valued, convex, and everywhere differentiable with

$$g(x) = \nabla F(x) = \frac{x - p(x)}{\lambda}. \tag{2.1}$$

Moreover, the gradient mapping $g : \Re^n \to \Re^n$ is globally Lipschitz continuous with modulus λ, i.e.,

$$\|g(x) - g(y)\| \le \frac{1}{\lambda}\|x - y\|, \ \forall\, x, y \in \Re^n. \tag{2.2}$$

(ii) x is an optimal solution to (1.1) if and only if $\nabla F(x) = 0$, namely, $p(x) = x$.

(iii) By the Rademacher theorem and the Lipschitzian property of ∇F, the set of generalized Jacobian matrices (see [17])

$$\partial_B g(x) = \{V \in \Re^{n\times n} : V = \lim_{x_k \to x} \nabla g(x_k), x_k \in D_g\}$$

is nonempty and compact for each $x \in \Re^n$, where $D_g = \{x \in \Re^n : \text{g is differentiable at x}\}$. Since g is a gradient mapping of the convex function F, then every $V \in \partial_B g(x)$ is a symmetric positive semidefinite matrix for each $x \in \Re^n$.

(iv) If g is BD-regular at x, namely all matrices $V \in \partial_B g(x)$ are nonsingular. Then, for all $y \in \Omega$, there exist constants $\mu_1 > 0$, $\mu_2 > 0$ and a neighborhood Ω of x such that

$$d^T V d \ge \mu_1 \|d\|^2, \ \|V^{-1}\| \le \mu_2, \ \forall\, d \in \Re^n, \ V \in \partial_B g(x).$$

It is obviously that $F(x)$ and $g(x)$ can be obtained through the optimal solution of $\text{argmin}_{z\in\Re^n}\theta(z)$. However, $p(x)$, the minimizer of $\theta(z)$, is difficult or even impossible

to exactly solve. Such makes that we can not apply the exact value of $p(x)$ to define $F(x)$ and $g(x)$. Fortunately, for each $x \in \Re^n$ and any $\varepsilon > 0$, there exists a vector $p^\alpha(x, \varepsilon) \in \Re^n$ such that

$$f(p^\alpha(x, \varepsilon)) + \frac{1}{2\lambda}\|p^\alpha(x, \varepsilon) - x\|^2 \leq F(x) + \varepsilon. \tag{2.3}$$

Thus, we can use $p^\alpha(x, \varepsilon)$ to define approximations of $F(x)$ and $g(x)$ by

$$F^\alpha(x, \varepsilon) = f(p^\alpha(x, \varepsilon)) + \frac{1}{2\lambda}\|p^\alpha(x, \varepsilon) - x\|^2 \tag{2.4}$$

and

$$g^\alpha(x, \varepsilon) = \frac{x - p^\alpha(x, \varepsilon)}{\lambda}, \tag{2.5}$$

respectively. A remarkable feature of $F^\alpha(x, \varepsilon)$ and $g^\alpha(x, \varepsilon)$ is given as follows (see [9]).

Proposition 2.1 *Let $p^\alpha(x, \varepsilon)$ be a vector satisfying (2.3), $F^\alpha(x, \varepsilon)$ and $g^\alpha(x, \varepsilon)$ are defined by (2.4) and (2.5), respectively. Then we get*

$$F(x) \leq F^\alpha(x, \varepsilon) \leq F(x) + \varepsilon, \tag{2.6}$$

$$\|p^\alpha(x, \varepsilon) - p(x)\| \leq \sqrt{2\lambda\varepsilon}, \tag{2.7}$$

and

$$\|g^\alpha(x, \varepsilon) - g(x)\| \leq \sqrt{2\varepsilon/\lambda}. \tag{2.8}$$

The above proposition says that we can approximately compute $F^\alpha(x, \varepsilon)$ and $g^\alpha(x, \varepsilon)$. By choosing parameter ε small enough, $F^\alpha(x, \varepsilon)$ and $g^\alpha(x, \varepsilon)$ may be made arbitrarily close to $F(x)$ and $g(x)$, respectively.

3 Motivation and algorithm

The following iterative formula is often used by CG method for (1.4)

$$v_{k+1} = v_k + \alpha_k q_k, \ \ k = 1, 2, \ldots \tag{3.1}$$

where v_k is the current iterate point, $\alpha_k > 0$ is a steplength, and q_k is the search direction defined by

$$q_{k+1} = \begin{cases} -\nabla h_{k+1} + \beta_k q_k, & \text{if } k \geq 1 \\ -\nabla h_{k+1}, & \text{if } \ k = 0, \end{cases} \tag{3.2}$$

where $\nabla h_{k+1} = \nabla h(v_{k+1})$ and $\beta_k \in \Re$ is a scalar which determines different conjugate gradient methods. In order to ensure that search direction is sufficiently descent, Zhang et al. [46] presented a modified PRP method with

$$q_{k+1} = \begin{cases} -\nabla h_{k+1} + \beta_k^{PRP} q_k - \vartheta_k \delta_k, & \text{if } k \geq 1 \\ -\nabla h_{k+1}, & \text{if } k = 0, \end{cases} \tag{3.3}$$

where $\vartheta_k = \frac{\nabla h_{k+1}^T q_k}{\|\nabla h_k\|^2}$, $\beta_k^{PRP} = \frac{\nabla h_{k+1}^T \delta_k}{\|\nabla h_k\|^2}$, and $\delta_k = \nabla h_{k+1} - \nabla h_k$. It is not difficult to get $q_k^T \nabla h_k = -\|\nabla h_k\|^2$. This method can be reduced to a standard PRP method if exact line search is used. Its global convergence with Armijo-type line search is obtained, but fails to weak wolfe-Powell line search. Even though this method has descent property, the direction search may not be a descent direction when v_k is far from the solution. In order to ensure the convergence of the given algorithm, the search direction should belong to a trust region. Motivated by this consideration, the method (3.3), and the observations in Sects. 1 and 2, we propose a modified PRP conjugate gradient formula for (1.2)

$$d_{k+1} = \begin{cases} -g^{\alpha}(x_{k+1}, \varepsilon_{k+1}) + \frac{g^{\alpha}(x_{k+1},\varepsilon_{k+1})^T y_k^* d_k - d_k^T g^{\alpha}(x_{k+1},\varepsilon_{k+1}) y_k^*}{\max\{2\|d_k\|\|y_k^*\|, \|g^{\alpha}(x_k,\varepsilon_k)\|^2\}}, & \text{if } k \geq 1 \\ -g^{\alpha}(x_{k+1}, \varepsilon_{k+1}), & \text{if } k = 0, \end{cases} \tag{3.4}$$

where $y_k^* = y_k + A_k s_k$, $y_k = g^{\alpha}(x_{k+1}, \varepsilon_{k+1}) - g^{\alpha}(x_k, \varepsilon_k)$, $s_k = x_{k+1} - x_k$, and

$$A_k = \max\left\{0, \frac{(g^{\alpha}(x_{k+1}, \varepsilon_{k+1}) + g^{\alpha}(x_k, \varepsilon_k))^T s_k + 3(F^{\alpha}(x_k, \varepsilon_k) - F^{\alpha}(x_{k+1}, \varepsilon_{k+1}))}{\|s_k\|^2}\right\}.$$

It is easy to see that the given method can be reduced to a standard PRP method if exact line search is used. For all k, we can easily get $d_{k+1}^T g^{\alpha}(x_{k+1}, \varepsilon_{k+1}) = -\|g^{\alpha}(x_{k+1}, \varepsilon_{k+1})\|^2$, which means that the proposed direction satisfies the sufficiently descent properties. Many authors hinted that the sufficiently descent condition may be crucial for conjugate gradient methods [1,2]. This property can ensure that the function values is decreasing. Combining with a nonmonotone line search technique, we list the steps of our algorithm as follows.

Algorithm 3.1 Nonmonotone Conjugate Gradient Algorithm.

Step 0. Initialization. Given $x_0 \in \Re^n$, $E_0 \in \Re$, $\sigma \in (0, 1)$, $s > 0$, $\lambda > 0$, $\rho \in [0, 1]$, $E_0 = 1$, $J_0 = F^{\alpha}(x_0, \varepsilon_0)$, $d_0 = -g^{\alpha}(x_0, \varepsilon_0)$ and $\epsilon \in (0, 1)$. Let $k = 0$.

Step 1. Termination Criterion. Stop if x_k satisfies termination condition $\|g^{\alpha}(x_k, \varepsilon_k)\| < \epsilon$ of problem (1.2).

Step 2: Choose a scalar ε_{k+1} such that $0 < \varepsilon_{k+1} < \varepsilon_k$ and compute step size α_k by the following Armijo-type line search rule

$$F^{\alpha}(x_k + \alpha_k d_k, \varepsilon_{k+1}) - J_k \leq \sigma \alpha_k g^{\alpha}(x_k, \varepsilon_k)^T d_k, \tag{3.5}$$

where $\alpha_k = s \times 2^{-i_k}$, $i_k \in \{0, 1, 2, \ldots\}$.

Step 3: Let $x_{k+1} = x_k + \alpha_k d_k$. If $\|g^{\alpha}(x_{k+1}, \varepsilon_{k+1})\| < \epsilon$, then stop.

Step 4: Update J_k by

$$E_{k+1} = \rho E_k + 1, \ \ J_{k+1} = \frac{\rho E_k J_k + F^\alpha(x_k + \alpha_k d_k, \varepsilon_{k+1})}{E_{k+1}}. \tag{3.6}$$

Step 5: Calculate the search direction by (3.4).
Step 6: Set $k := k + 1$ and go to Step 2.

Remark It is not difficult to see that J_{k+1} is a convex combination of J_k and $F^\alpha(x_{k+1}, \varepsilon_{k+1})$. Since $J_0 = F^\alpha(x_0, \varepsilon_0)$, it follows that J_k is a convex combination of the function values $F^\alpha(x_0, \varepsilon_0)$, $F^\alpha(x_1, \varepsilon_1)$, $\ldots$, $F^\alpha(x_k, \varepsilon_k)$. The choice of ρ controls the degree of nonmonotonicity. If $\rho = 0$, then the line search is the usual monotone Armijo line search. If $\rho = 1$, then $J_k = C_k$, where

$$C_k = \frac{1}{k+1} \sum_{i=0}^{k} F^\alpha(x_i, \varepsilon_i)$$

is the average function value.

4 Properties and global convergence

In the section, we turn to the behavior of Algorithm 3.1 when it is applied to problem (1.1). In order to get the global convergence of Algorithm 3.1, the following assumptions are needed.

Assumption A (i) The sequence $\{V_k\}$ is bounded, i.e., there exists a positive constant M such that

$$\|V_k\| \le M, \ \ \forall\ k. \tag{4.1}$$

(ii) F is bounded from below.
(iii) For sufficiently large k, $\ \varepsilon_k$ converges to zero.

The following lemma shows that the conjugate gradient direction possesses the sufficiently descent property and belongs to a trust region.

Lemma 4.1 *For all $k \ge 0$, we have*

$$g^\alpha(x_k, \varepsilon_k)^T d_k = -\|g^\alpha(x_k, \varepsilon_k)\|^2 \tag{4.2}$$

and

$$\|d_k\| \le 2\|g^\alpha(x_k, \varepsilon_k)\|. \tag{4.3}$$

Proof For $k = 0$, $d_0 = -g^\alpha(x_0, \varepsilon_0)$, we get (4.2) and (4.3). For $k \ge 1$, from the definition of d_k, we obtain

$$d_{k+1}^T g^\alpha(x_{k+1}, \varepsilon_{k+1}) = -\|g^\alpha(x_{k+1}, \varepsilon_{k+1})\|^2$$

$$+\left[\frac{g^{\alpha}(x_{k+1},\varepsilon_{k+1})^T y_k^* d_k - d_k^T g^{\alpha}(x_{k+1},\varepsilon_{k+1}) y_k^*}{\max\{2\|d_k\|\|y_k^*\|, \|g^{\alpha}(x_k,\varepsilon_k)\|^2\}}\right]^T g^{\alpha}(x_{k+1},\varepsilon_{k+1})$$
$$= -\|g^{\alpha}(x_{k+1},\varepsilon_{k+1})\|^2.$$

Then the relation (4.2) holds. Now we turn to prove that (4.3) holds too. By the definition of d_k again, we get

$$\begin{aligned}\|d_{k+1}\| &= \| - g^{\alpha}(x_{k+1},\varepsilon_{k+1}) + \frac{g^{\alpha}(x_{k+1},\varepsilon_{k+1})^T y_k^* d_k - d_k^T g^{\alpha}(x_{k+1},\varepsilon_{k+1}) y_k^*}{\max\{2\|d_k\|\|y_k^*\|, \|g^{\alpha}(x_k,\varepsilon_k)\|^2\}}\| \\ &\le \|g^{\alpha}(x_{k+1},\varepsilon_{k+1})\| \\ &\quad + \frac{\|g^{\alpha}(x_{k+1},\varepsilon_{k+1})\|\|y_k^*\|\|d_k\| + \|d_k\|\|g^{\alpha}(x_{k+1},\varepsilon_{k+1})\|\|y_k^*\|}{\max\{2\|d_k\|\|y_k^*\|, \|g^{\alpha}(x_k,\varepsilon_k)\|^2\}} \\ &\le 2\|g^{\alpha}(x_{k+1},\varepsilon_{k+1})\|,\end{aligned}$$

where the last inequality follows

$$\max\{2\|d_k\|\|y_k^*\|, \|g^{\alpha}(x_k,\varepsilon_k)\|^2\} \ge 2\|d_k\|\|y_k^*\|.$$

Then this completes the proof. □

Based on the above lemma, similar to Lemma 1.1 in [45], it is not difficult to get the following lemma. So we only state as follows but omit the proof.

Lemma 4.2 *Let Assumption A hold and the sequence* $\{x_k\}$ *be generated by* Algorithm 3.1. *Then for each k, we have* $F^{\alpha}(x_k,\varepsilon_k) \le J_k \le C_k$. *Moreover, there exists an* α_k *satisfying Armijo conditions of the line search update.*

The above lemma shows that Algorithm 3.1 is well-defined.

Lemma 4.3 *Let Assumption A hold and the sequence* $\{x_k\}$ *be generated by* Algorithm 3.1. *Suppose that* $\varepsilon_k = o(\alpha_k^2\|d_k\|^2)$ *holds. Then, for sufficiently large k, there exists a positive constant* m_0 *such that*

$$\alpha_k \ge m_0. \tag{4.4}$$

Proof Let α_k satisfy the line search (3.5). If $\alpha_k \ge 1$, the proof is complete. Otherwise we have $\alpha_k' = \frac{\alpha_k}{2}$ satisfying

$$F^{\alpha}(x_k + \alpha_k' d_k, \varepsilon_{k+1}) - J_k > \sigma\alpha_k' g^{\alpha}(x_k,\varepsilon_k)^T d_k.$$

By Lemma 4.2, we have $F^{\alpha}(x_k,\varepsilon_k) \le J_k \le C_k$, then

$$F^{\alpha}(x_k+\alpha_k' d_k,\varepsilon_{k+1}) - F^{\alpha}(x_k,\varepsilon_k) \ge F^{\alpha}(x_k+\alpha_k' d_k,\varepsilon_{k+1}) - J_k > \sigma\alpha_k' g^{\alpha}(x_k,\varepsilon_k)^T d_k \tag{4.5}$$

holds. Using (4.5), (2.6), and Taylor's formula, we get

$$\begin{aligned}\sigma\alpha_k' g^\alpha(x_k,\varepsilon_k)^T d_k &< F^\alpha(x_k+\alpha_k' d_k,\varepsilon_{k+1}) - F^\alpha(x_k,\varepsilon_k)\\ &\le F(x_k+\alpha_k' d_k) - F(x_k) + \varepsilon_{k+1}\\ &= \alpha_k' d_k^T g(x_k) + \frac{1}{2}(\alpha_k')^2 d_k^T V(\xi_k) d_k + \varepsilon_{k+1}\\ &\le \alpha_k' d_k^T g(x_k) + \frac{M}{2}(\alpha_k')^2\|d_k\|^2 + \varepsilon_{k+1}, \end{aligned} \tag{4.6}$$

where $V(\xi_k) \in \partial_B g(\xi_k)$, $\xi_k = x_k + \theta\alpha_k' d_k$, $\theta \in (0,1)$, and the last inequality follows (4.1). It follows that from (4.6)

$$\begin{aligned}\alpha_k' &> \left[\frac{(g^\alpha(x_k,\varepsilon_k) - g(x_k))^T d_k - (1-\sigma) g^\alpha(x_k,\varepsilon_k)^T d_k - \varepsilon_{k+1}/(\alpha_k')^2}{\|d_k\|^2}\right]\frac{2}{M}\\ &\ge \left[\frac{(1-\sigma)\|g^\alpha(x_k,\varepsilon_k)\|^2 - \sqrt{2\varepsilon_k/\lambda}\|d_k\| - \varepsilon_k}{\|d_k\|^2}\right]\frac{2}{M}\\ &= \left[\frac{(1-\sigma)\|g^\alpha(x_k,\varepsilon_k)\|^2}{\|d_k\|^2} - o(\alpha_k)/\sqrt{\lambda} - o(1)\right]\frac{2}{M}\\ &\ge \frac{(1-\sigma)}{2M}, \end{aligned} \tag{4.7}$$

where the second inequality follows (2.8), (4.2), and $\varepsilon_{k+1} \le \varepsilon_k$, the equality follows $\varepsilon_k = o(\alpha_k^2\|d_k\|^2)$, and the last inequality follows (4.3). Thus, we have

$$\alpha_k \ge \frac{1-\sigma}{M}.$$

Let $m_0 \in \left(0, \frac{1-\sigma}{M}\right]$, we complete the proof. □

Now we prove the global convergence of Algorithm 3.1.

Theorem 4.1 *Let the conditions in* Lemma 4.3 *hold. Then,* $\lim_{k\to\infty}\|g(x_k)\| = 0$ *and any accumulation point of* x_k *is an optimal solution of* (1.1).

Proof We first show that

$$\lim_{k\to\infty}\|g^\alpha(x_k,\varepsilon_k)\| = 0. \tag{4.8}$$

Suppose that (4.8) is not true. Then there exist $\epsilon_0 > 0$ and $k_0 > 0$ satisfying

$$\|g^\alpha(x_k,\varepsilon_k)\| \ge \epsilon_0, \ \forall\, k > k_0. \tag{4.9}$$

By (3.5), (4.2), (4.4), and (4.9), we get

$$F^{\alpha}(x_{k+1},\varepsilon_{k+1}) - J_k \le \sigma\alpha_k g^{\alpha}(x_k,\varepsilon_k)^T d_k$$
$$= -\sigma\alpha_k \|g^{\alpha}(x_k,\varepsilon_k)\|^2 \le -\sigma m_0\epsilon_0,\ \ \forall\ k > k_0.$$

By the definition of J_{k+1}, we have

$$\begin{aligned} J_{k+1} &= \frac{\rho E_k J_k + F^{\alpha}(x_k+\alpha_k d_k,\varepsilon_{k+1})}{E_{k+1}} \\ &\le \frac{\rho E_k J_k + J_k - \sigma m_0\epsilon_0}{E_{k+1}} \\ &= J_k - \frac{\sigma m_0\epsilon_0}{E_{k+1}}. \end{aligned} \tag{4.10}$$

Since $F^{\alpha}(x,\varepsilon)$ is bounded from below and $F^{\alpha}(x_k,\varepsilon_k) \le J_k$ holds for all k, we conclude that J_k is bounded from below. It follows that from (4.10) that

$$\sum_{k=k_0}^{\infty} \frac{\sigma m_0\epsilon_0}{E_{k+1}} < \infty. \tag{4.11}$$

By the definition of E_{k+1}, we get $E_{k+1} \le k+2$, then (4.8) holds. By (2.8), we have

$$\|g^{\alpha}(x_k,\varepsilon_k) - g(x_k)\| \le \sqrt{\frac{2\varepsilon_k}{\lambda}}.$$

Together with Assumption A(iii), this means that

$$\lim_{k\to\infty} \|g(x_k)\| = 0 \tag{4.12}$$

holds. Let x^* be an accumulation point of $\{x_k\}$, without loss of generality, there exists a subsequence $\{x_k\}_K$ satisfying

$$\lim_{k\in K,\, k\to\infty} x_k = x^*. \tag{4.13}$$

From properties of $F(x)$, we have $g(x_k) = (x_k - p(x_k))/\lambda$. Then by (4.12) and (4.13), $x^* = p(x^*)$ holds. Therefore x^* is an optimal solution of (1.1). □

5 Numerical results

5.1 Small-scale problems

The nonsmooth problems of Table 1 can be found in [26]. Table 1 contains problem dimensions and optimum function values.

The algorithm is implemented by Matlab 7.6, experiments are run on a PC with CPU Intel Pentium Dual E7500 2.93GHz, 2G bytes of SDRAM memory, and Windows XP operating system. The parameters were chosen as $s = \lambda = 1,\ \rho = 0.5,\ \sigma = 0.8,$

Table 1 Small-scale nonsmooth problems

No.	Problems	Dim	$f_{ops}(x)$
1	Rosenbrock	2	0
2	Crescent	2	0
3	CB2	2	1.9522245
4	CB3	2	2.0
5	DEM	2	−3
6	QL	2	7.20
7	LQ	2	−1.4142136
8	Mifflin 1	2	−1.0
9	Mifflin 2	2	−1.0
10	Rosen-Suzuki	4	−44
11	Shor	5	22.600162
12	Colville	5	−32.348679
13	Maxq	20	0
14	Maxl	20	0
15	MXHILB	30	0
16	L1HILB	30	0
17	Shell Dual	15	32.348679
18	Maxquadl	10	−0.84140833

and $\varepsilon_k = 1/(NI + 2)^2$ (NI is the iteration number). For problem min $\theta(x)$, we use the function $fminsearch$ of Matlab to get the solution $p(x)$. We stopped the iteration when the condition $\|g^{\alpha}(x, \varepsilon)\| \leq 10^{-10}$ was satisfied. In order to show the performance of the given algorithm, we also list the results of paper [25] (proximal bundle method, PBL) and the paper [34] (trust region concept, BT). The numerical results of PBL and BT can be found in [25]. The columns of Table 2 have the following meanings:

Problem: the name of the test problem.
NI: the total number of iterations.
NF: the iteration number of the function evaluations.
$f(x)$: the function evaluations at the final iteration.
$f_{ops}(x)$: the optimization function evaluation.

From the numerical results in Table 2, for most of the test problems, it is not difficult to see that Algorithm 3.1 performs best among these three methods. The function value of the PBL and the BT more approximate to the optimization evaluation than Algorithm 3.1 does. Overall we think that the method provide a valid approach for solving nonsmooth problems.

5.2 Large-scale problems

The following problems of Table 3 can be found in [13]. The numbers of variables used were 1000. The values of parameters were similar to the small-scale problems.

Table 2 Test results

No.	Algorithm 3.1 NI/NF/$f(x)$	PBL NI/NF/$f(x)$	BT NI/NF/$f(x)$	$f_{ops}(x)$
1	$53/55/3.248983 \times 10^{-7}$	$42/45/0.381 \times 10^{-6}$	$79/88/0.130 \times 10^{-11}$	0
2	$13/15/2.851814 \times 10^{-5}$	$18/20/0.679 \times 10^{-6}$	$24/27/ \times 10^{-6}$	0
3	12/14/1.952225	32/34/1.9522245	13/16/1.952225	1.9522245
4	3/7/2.000008	14/16/2.0	13/21/2.0	2.0
5	$4/6/-2.999928$	$17/19/-3.0$	9/13/−3.0	−3
6	19/21/7.20	13/15/7.2000015	12/17/−7.20	7.20
7	5/6/−1.414214	11/12/−1.4142136	10/11/−1.414214	−1.4142136
8	3/5/−0.9922728	66/68/−0.99999941	49/74/−1.0	−1.0
9	11/12/−0.9999205	13/15/−1.0	6/13/−1.0	−1.0
10	8/9/−43.99761	43/45/−43.999999	22/32/−43.99998	−44
11	7/8/22.60030	27/29/22.600162	29/30/−22.60016	22.600162
12	7/8/−32.32184	62/64/−32.348679	45/45/−32.3486	−32.348679
13	$158/160/1.446425 \times 10^{-5}$	$161/162/0.166 \times 10^{-6}$	125/128/0	0
14	$14/18/3.451419 \times 10^{-4}$	$39/40/ \times 10^{-12}$	74/84/0	0
15	$34/39/2.200567 \times 10^{-5}$	$19/20/0.424 \times 10^{-8}$	$15/15/0.13 \times 10^{-7}$	0
16	$63/65/1.614436 \times 10^{-5}$	$19/20/0.99 \times 10^{-12}$	$16/16/0.77 \times 10^{-8}$	0
17	20/23/34.95971	1410/1501/32.349129	191/600/32.3538	32.348679
18	14/17/−0.8814104	80/81/−0.84140833	45/56/−0.8414083	−0.84140833

Table 3 Large-scale nonsmooth problems

No.	Problems	Dim	x_0
1	Chained LQ	1000	$(-0.5, -0.5, \ldots)$
2	Generalization of MXHILB	1000	$(1, 1, \ldots)$
3	Nonsmooth generalization of Brown function 2	1000	$(1, 0, \ldots)$
4	Chained Mifflin 2	1000	$(-1, -1, \ldots)$
5	Chained Crescent I	1000	$(-1.5, 2, \ldots)$
6	Chained Crescent II	1000	$(1, 0, \ldots)$

The following experiments were implemented in Fortran 90. In order to show the performance of the given algorithm, we compared it with the method (LMBM) of paper [13]. The stop rule and parameters are the same as [13].

LMBM [13]. New limited memory bundle method for large-scale nonsmooth optimization. The fortran codes are contributed by Haarala, Miettinen, and Mäkelä, which are available at
http://napsu.karmitsa.fi/lmbm/.

For these six large-scale problems, the iteration number of Algorithm 3.1 is competitive to those of the LMBM method. The final gradient value of the LMBM is

Table 4 Test results

No.	Dim	Algorithm 3.1 NI/NF/$\|g(x_k)\|^2$	LMBM NI/NF/$\|g(x_k)\|^2$
1	1000	$112/1076/0.3994 \times 10^4$	$300/1824/0.288433737885047 \times 10^4$
2	1000	$5296/275346/0.10197820736616 \times 10^{-2}$	$21492/22259/0.19520086832185 \times 10^{-4}$
3	1000	$202/2782/2.22714300277783$	$467/3873/0.249859802632258 \times 10^4$
4	1000	$52/125/0.245600546442004 \times 10^4$	$1254/7355/0.446064309620646 \times 10^3$
5	1000	$187/2411/0.916993560367606 \times 10^4$	$138/560/1.80388489502507$
6	1000	$187/1767/0.499346466626978 \times 10^4$	$763/7522/0.728629687398744 \times 10^3$

better than those of the given algorithm. Taking everything together, the preliminary numerical results indicate the proposed method is efficient (Table 4).

6 Conclusion

In this paper, we propose a conjugate gradient method for nonsmooth convex minimization. The global convergence is established under suitable conditions. Numerical results show that this method is interesting. The CG method has simplicity and low memory requirement, the PRP method is one of the most effective CG methods, the nonmonotone line search technique of [45] is competitive to other line search techniques, and the secant equation of [42] possesses better properties for general convex functions. Based on the above four cases, we present a modified PRP CG algorithm with a nonmonotone line search technique for nonsmooth optimization problems. This method has sufficiently descent property and the search direction belongs to a trust region. Moreover this method possesses the gradients information but also the functions information. The main work of this paper is to extend the CG method to solve nonsmooth problems.

Acknowledgments We would like to thank the anonymous referees for catching several typos of the paper, and their useful suggestions and comments which improved the paper greatly. This work is supported by the Guangxi NSF (Grant No. 2012GXNSFAA053002) and the China NSF (Grant Nos. 11261006 and 11161003).

References

1. Ahmed, T., Storey, D.: Efficient hybrid conjugate gradient techniques. Journal of Optimization Theory and Applications **64**, 379–394 (1990)
2. Al-Baali, A.: Descent property and global convergence of the Flecher–Reeves method with inexact line search. IMA Journal of Numerical Analysis **5**, 121–124 (1985)
3. Birge, J.R., Qi, L., Wei, Z.: A general approach to convergence properties of some methods for nonsmooth convex optimization. Applied Mathematics & Optimization **38**, 141–158 (1998)
4. Birgin, E.G., Martinez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10**, 1196–1121 (2000)
5. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: A family of veriable metric proximal methods. Mathematical Programming **68**, 15–47 (1995)

6. Calamai, P.H., Moré, J.J.: Projected gradient methods for linear constrained problems. Mathematical Programming **39**, 93–116 (1987)
7. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minization. Mathematical Programming **62**, 261–273 (1993)
8. Dai, Y., Yuan, Y.: Nonlinear Conjugate Gradient Methods. Shanghai Scientific and Technical Publishers, Shanghai (1998)
9. Fukushima, M., Qi, L.: A global and superlinearly convergent algorithm for nonsmooth convex minimization. SIAM Journal on Optimization **6**, 1106–1120 (1996)
10. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM Journal on Optimization **2**, 21–42 (1992)
11. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for newton's method. SIAM Journal on Numerical Analysis **23**, 707–716 (1986)
12. Grippo, L., Lucidi, S.: A globally convergent version of the Polak–Ribière gradient method. Mathematical Programming **78**, 375–391 (1997)
13. Haarala, M., Miettinen, K., Mäkelä, M.M.: New limited memory bundle method for large-scale nonsmooth optimization. Optimization Methods and Software **19**, 673–692 (2004)
14. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM Journal on Optimization **16**, 170–192 (2005)
15. Hager, W.W., Zhang, H.: Algorithm 851: *CGDESCENT*, a conjugate gradient method with guaranteed descent. ACM Transactions on Mathematical Software **32**, 113–137 (2006)
16. Han, J.Y., Liu, G.H.: Global convergence analysis of a new nonmonotone BFGS algorithm on convex objective functions. Computational Optimization and Applications **7**, 277–289 (1997)
17. Hiriart-Urruty, J.B., Lemmaréchal, C.: Convex Analysis and Minimization Algorithms II. Springer, Berlin (1983)
18. Kiwiel, K.C.: Methods of Descent for Nondifferentiable Optimization. Springer, Berlin (1985)
19. Kiwiel, K.C.: Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. Mathematical Programming **69**, 89–109 (1995)
20. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable optimization. Mathematical Programming **46**, 105–122 (1990)
21. Lemaréchal, C.: Extensions Diverses des Médthodes de Gradient et Applications. Thèse d'Etat, Paris (1980)
22. Lemaréchal, C.: Nondifferentiable optimization. In: Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (eds.) Handbooks in Operations Research and Management Science, Vol. 1, Optimization. North-Holland, Amsterdam (1989)
23. Lemaréchal, C., Sagastizábal, C.: Practical aspects of the Moreau-Yosida regularization: Theoretical preliminaries. SIAM Journal on Optimization **7**, 367–385 (1997)
24. Liu, G.H., Peng, J.M.: The convergence properties of a nonmonotonic algorithm. Journal of Computational Mathematics **1**, 65–71 (1992)
25. Lukšan, L., Vlček, J.: A bundle-Newton method for nonsmooth unconstrained minimization. Mathematical Programming **83**, 373–391 (1998)
26. Lukšan, L., Vlček, J. (2000). Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization, Technical Report No. 798, Institute of Computer Science, Academy of Sciences of the Czech Republic
27. Polak, E., Ribière, G.: Note sur la convergence de directions conjugées. Revue Francaise informat Recherche Opératinelle **3**, 35–43 (1969)
28. Polyak, B.T.: The conjugate gradient method in extreme problems. USSR Computational Mathematics and Mathematical Physics **9**, 94–112 (1969)
29. Powell, M.J.D.: Nonconvex Minimization Calculations and the Conjugate Gradient Method, Vol. 1066 of Lecture Notes in Mathematics. Spinger, Berlin (1984)
30. Powell, M.J.D.: Convergence properties of algorithm for nonlinear optimization. SIAM Review **28**, 487–500 (1986)
31. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Mathematics of Operations Research **18**, 227–245 (1993)
32. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Mathematical Programming **58**, 353–367 (1993)

33. Schramm, H.: Eine kombination yon bundle-und trust-region-verfahren zur Lösung nicht- differenzierbare optimierungsprobleme, Bayreuther Mathematische Schriften, Heft 30. University of Bayreuth, Bayreuth (1989)
34. Schramm, H., Zowe, J.: A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. SIAM Journal on Optimization **2**, 121–152 (1992)
35. Toint, P.L.: An assessment of non-monotone line search techniques for unconstrained minimization problem. SIAM Journal on Computing **17**, 725–739 (1996)
36. Wei, Z., Li, G., Qi, L.: Global convergence of the Polak–Ribiere–Polyak conjugate gradient methods with inexact line search for nonconvex unconstrained optimization problems. Mathematics of Computation **77**, 2173–2193 (2008)
37. Wei, Z., Li, G., Qi, L.: New quasi-Newton methods for unconstrained optimization problems. Applied Mathematics and Computation **175**, 1156–1188 (2006)
38. Wei, Z., Yu, G., Yuan, G., Lian, Z.: The superlinear convergence of a modified BFGS-type method for unconstrained optimization. Computational Optimization and Application **29**, 315–332 (2004)
39. Wolfe, P.: A method of conjugate subgradients for minimizing nondifferentiable convex functions. Mathematical Programming Study **3**, 145–173 (1975)
40. Xiao, Y., Wei, Z., Wang, Z.: A limited memory BFGS-type method for large-scale unconstrained optimization. Computational Mathematics Applications **56**, 1001–1009 (2008)
41. Yuan, G.: Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems. Optimization Letters **3**, 11–21 (2009)
42. Yuan, G., Wei, Z.: Convergence analysis of a modified BFGS method on convex minimizations. Computational Optimization and Applications **47**, 237–255 (2010)
43. Yuan, G., Wei, Z., Li, G.: A modified PolakCRibièreCPolyak conjugate gradient algorithm for nonsmooth convex programs. Journal of Computational and Applied Mathematics **255**, 86–96 (2014)
44. Zhang, J., Deng, N., Chen, L.: New quasi-Newton equation and related methods for unconstrained optimization. Journal of Optimization Theory and Application **102**, 147–167 (1999)
45. Zhang, H., Hager, W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM Journal on Optimization **14**, 1043–1056 (2004)
46. Zhang, L., Zhou, W., Li, D.: A descent modified Polak–Ribière–Polyak conjugate method and its global convergence. IMA Journal on Numerical Analysis **26**, 629–649 (2006)
47. Zhou, J., Tits, A.: Nonmonotone line search for minimax problem. Journal Optimization Theory and Applications **76**, 455–476 (1993)
48. Zowe, J.: Computational mathematical programming. In: Schittkowski, K. (ed.) Nondifferentiable optimization, pp. 323–356. Springer, Berlin (1985)