

## Accepted Manuscript

A modified Polak–Ribière–Polyak conjugate gradient algorithm for nonsmooth convex programs

Gonglin Yuan, Zengxin Wei, Guoyin Li

PII: S0377-0427(13)00218-5

DOI: <http://dx.doi.org/10.1016/j.cam.2013.04.032>

Reference: CAM 9136

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 1 August 2011

Revised date: 3 April 2013

Please cite this article as: G. Yuan, Z. Wei, G. Li, A modified Polak–Ribière–Polyak conjugate gradient algorithm for nonsmooth convex programs, *Journal of Computational and Applied Mathematics* (2013), <http://dx.doi.org/10.1016/j.cam.2013.04.032>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



# A Modified Polak-Ribière-Polyak Conjugate Gradient Algorithm for Nonsmooth Convex Programs \*

GONGLIN YUAN<sup>†</sup> ZENGXIN WEI<sup>‡</sup> GUOYIN LI<sup>§</sup>

Revised Version: April 3, 2013

## Abstract

The conjugate gradient (CG) method is one of the most popular methods for solving smooth unconstrained optimization problems due to its simplicity and low memory requirement. However, the usage of CG methods are mainly restricted in solving smooth optimization problems so far. The purpose of this paper is to present efficient conjugate gradient-type methods to solve nonsmooth optimization problems. By using the Moreau-Yosida regulation (smoothing) approach and a nonmonotone line search technique, we propose a modified Polak-Ribière-Polyak (PRP) CG algorithm for solving a nonsmooth unconstrained convex minimization problem. Our algorithm possesses the following three desired properties. (i) The search direction satisfies the sufficiently descent property and belongs to a trust region automatically; (ii) The search direction makes use of not only gradient information but also function value information; (iii) The algorithm inherits an important property of the well-known PRP method: the tendency to turn towards the steepest descent direction if a small step is generated away from the solution, preventing a sequence of tiny steps from happening. Under standard conditions, we show that the algorithm converges globally to an optimal solution. Numerical experiment shows that our algorithm is effective and suitable for solving large-scale nonsmooth unconstrained convex optimization problems.

**Key Words.** Nonsmooth convex optimization; Conjugate gradient method; Nonmonotone line search; Global convergence.

**MSC 2010 subject classifications.** 90C52, 65K05, 90C30.

---

\*This work is supported by Guangxi NSF (Grant No. 2012GXNSFAA053002 and 2012GXNSFAA053013) and China NSF (Grant No. 11261006, 11161003, 71101033 and 71001015)

<sup>†</sup>Department of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, P.R. China, E-mail address: glyuan@gxu.edu.cn.

<sup>‡</sup>Department of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, 530004, P.R. China, E-mail:zxwei@gxu.edu.cn, Tel: +86-0771-3272035, Fax:+86-0771-3232084.

<sup>§</sup>Department of Applied Mathematics, The University of New South Wales, Kensington, 2052, Sydney, Australia, E-mail: g.li@unsw.edu.au, Tel: +61-02-93857095.

## 1. Introduction

Consider the following unconstrained convex optimization problem

$$(P) \quad \min_{x \in \mathbb{R}^n} f(x), \quad (1.1)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a possibly nonsmooth convex function. In the special case when  $f$  is continuously differentiable, this optimization problem has been well studied for several decades. In particular, the conjugate gradient (CG) method and the quasi-Newton method are two major popular methods for solving smooth unconstrained convex optimization problems. The quasi-Newton method involves the computation/approximation of the Hessian matrix of the objective function, and often has fast convergence. On the other hand, the conjugate gradient method only uses the first order information and so, is suitable for solving large scale optimization problems. At present, there are many well-known conjugate gradient formulas (for example, see [11, 12, 13, 19, 31, 32, 38, 27, 53]). In particular, the so-called Polak-Ribière-Polyak (PRP) conjugate gradient methods has been well studied, and is generally believed to be one of the most efficient conjugate gradient methods. However, the usage of conjugate gradient methods are mainly restricted in solving smooth optimization problems so far.

Recently, many modern applications of optimization call for the need of studying large scale nonsmooth convex optimization problem. As an illustrating example, let us consider the image restoration problem arises in image processing. The image restoration problem is to reconstruct an image of an unknown scene from an observed image. This problem plays an important role in medical sciences, biological engineering and other areas of science and engineering [3, 8, 37]. The most common image degradation model can be represented by the following system:

$$b = Ax + \eta$$

where  $\eta \in \mathbb{R}^m$  represents the noise,  $A$  is an  $m \times n$  blurring matrix,  $x \in \mathbb{R}^n$  and  $b \in \mathbb{R}^m$  are the underlying and observed images respectively. As  $\eta$  is typically unknown, one way to solve it is to solving the least square optimization problem  $\min_{x \in \mathbb{R}^n} \|Ax + b\|^2$ . Solving this problem alone will not get a satisfactory solution since the system is very sensitive to the noise and lack of information. To overcome this, the following regularized least square problem is often used:

$$\min_{x \in \mathbb{R}^n} \|Ax + b\|^2 + \lambda \|Dx\|_1$$

where  $D$  is a linear operator,  $\lambda$  is the regularization parameter that controls the trade-off between the data-fitting term and the regularization term and  $\|\cdot\|_1$  is the  $l^1$  norm. As  $l^1$  norm is nonsmooth, the above problem is a nonsmooth convex optimization problem and is typically of large scale.

The purpose of this paper is to present efficient conjugate gradient-type methods to solve the nonsmooth optimization problem (P). By using the Moreau-Yosida regulation (smoothing) approach and a nonmonotone line search technique, we propose a modified PRP conjugate gradient algorithm for solving a nonsmooth unconstrained convex minimization problem. Our algorithm possesses the following three desired properties. (i)

The search direction satisfies the sufficiently descent property and belongs to a trust region automatically; (ii) The search direction makes use of not only gradient information but also function information; (iii) The algorithm inherits an important property of the well-known PRP method: the tendency to turn towards the steepest descent direction if a small step is generated away from the solution, preventing a sequence of tiny steps from happening.

This paper is organized as follows. In Section 2, we briefly review some basic results in convex analysis and nonsmooth analysis. In Section 3, we present our new modified PRP conjugate gradient algorithm. In Section 4, we prove the global convergence of the proposed method. In Section 5, we discuss some similar extensions and provide another three modified conjugate gradient formulas. In Section 6, we report numerical results for our algorithm and present some comparison for existing methods for both small-scale and large-scale nonsmooth convex optimization problems. Finally, we conclude our paper and mention some of the possible future research topics in Section 7. Throughout this paper, without specification,  $\|\cdot\|$  denotes the Euclidean norm of vectors or matrices.

## 2. Elements of convex analysis and nonsmooth analysis

In this section, we review some basic facts and results in convex analysis and nonsmooth analysis. Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a convex (possibly nonsmooth) function. Let  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  be the so-called Moreau-Yosida regularization of  $f$  defined by

$$F(x) = \min_{z \in \mathbb{R}^n} \left\{ f(z) + \frac{1}{2\lambda} \|z - x\|^2 \right\}, \quad (2.1)$$

where  $\lambda$  is a positive parameter and  $\|\cdot\|$  denotes the Euclidean norm. Let

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

and denote  $p(x) = \operatorname{argmin}_z \theta(z, x)$ . Then,  $p(x)$  is well-defined and unique, since  $\theta(\cdot, x)$  is strongly convex for each fixed  $x$ . By (2.1),  $F$  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 important and useful properties of the Moreau-Yosida regularization function  $F$  are given as follows.

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

$$g(x) = \nabla F(x) = \frac{x - p(x)}{\lambda}. \quad (2.2)$$

Moreover, the gradient mapping  $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is globally Lipschitz continuous with modulus  $\lambda$ , i.e.,

$$\|g(x) - g(y)\| \leq \frac{1}{\lambda} \|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \quad (2.3)$$

(ii)  $x$  solves  $\min_{x \in \mathbb{R}^n} f(x)$  if and only if  $\nabla F(x) = 0$ , namely,  $p(x) = x$ .

It is obviously that  $F(x)$  and  $g(x)$  can be obtained through the optimal solution of  $\operatorname{argmin}_{z \in \mathbb{R}^n} \theta(z)$ . However,  $p(x)$ , the minimizer of  $\theta(z)$ , is difficult or even impossible to solve exactly. Thus, in the real computation, instead of calculating the exact value of  $p(x)$ ,  $F(x)$  and  $g(x)$ , we often use some appropriate approximation. Indeed, for each  $x \in \mathbb{R}^n$  and any  $\varepsilon > 0$ , there exists a vector  $p^\alpha(x, \varepsilon) \in \mathbb{R}^n$  such that

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

Thus, when  $\varepsilon$  is small, we can use  $p^\alpha(x, \varepsilon)$  to define approximations of  $F(x)$  and  $g(x)$  as follows:

$$F^\alpha(x, \varepsilon) = f(p^\alpha(x, \varepsilon)) + \frac{1}{2\lambda} \|p^\alpha(x, \varepsilon) - x\|^2 \quad (2.5)$$

and

$$g^\alpha(x, \varepsilon) = \frac{x - p^\alpha(x, \varepsilon)}{\lambda}, \quad (2.6)$$

respectively. Some implementable algorithms for computing  $p^\alpha(x, \varepsilon)$  for a nondifferentiable convex function can be found in [9]. A remarkable feature of  $F^\alpha(x, \varepsilon)$  and  $g^\alpha(x, \varepsilon)$  is given as follows [14].

**Proposition 2.1.** *Let  $p^\alpha(x, \varepsilon)$  be a vector satisfying (2.4),  $F^\alpha(x, \varepsilon)$  and  $g^\alpha(x, \varepsilon)$  are defined by (2.5) and (2.6), respectively. Then we get*

$$F(x) \leq F^\alpha(x, \varepsilon) \leq F(x) + \varepsilon, \quad (2.7)$$

$$\|p^\alpha(x, \varepsilon) - p(x)\| \leq \sqrt{2\lambda\varepsilon}, \quad (2.8)$$

and

$$\|g^\alpha(x, \varepsilon) - g(x)\| \leq \sqrt{2\varepsilon/\lambda}. \quad (2.9)$$

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

Based on these features, many algorithms have been proposed for solving problem (P) (for example, see [4]). The classical proximal point algorithm [34] can be regarded as a gradient-type method for solving problem (P), and has 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, 6, 10, 41, 42]). Lemaréchal [25] and Wolfe [44] initiated a giant stride forward in nonsmooth optimization by the bundle concept, which can handle convex and nonconvex  $f$ . Moreover, Kiwiel [24] proposed a bundle variant, which is close to bundle trust iteration method (see [36]). Recently, Haarala et al. proposed a new limited memory bundle method for large-scale nonsmooth optimization problems [16]. For other references of bundle methods, one may refer to [22, 23, 35].

### 3. Modified PRP Method for Nonsmooth Problem

In this section, we introduce our modified PRP conjugate gradient methods for solving the nonsmooth convex unconstrained optimization problem  $\min_{x \in \mathbb{R}^n} f(x)$  where  $f$  is a convex

function. Our basic idea is to make use of the Moreau-Yosida regularization to smoothing the function and also the two recently introduced acceleration techniques in modifying CG methods. To do this, we first introduce the two acceleration techniques for modifying the CG methods in solving smooth optimization problem.

### Modified CG method by enforcing sufficient descent condition

Recall that, when  $f$  is smooth, the search direction of the famous PRP conjugate gradient methods is given by

$$d_{k+1} = \begin{cases} -\nabla f_{k+1} + \beta_k^{PRP} d_k, & \text{if } k \geq 1, \\ -\nabla f_{k+1}, & \text{if } k = 0, \end{cases} \quad (3.1)$$

where  $\nabla f_k = \nabla f(x_k)$  and  $\beta_k^{PRP} = \frac{\nabla f_{k+1}^T y_k}{\|\nabla f_k\|^2}$  and  $y_k = \nabla f_{k+1} - \nabla f_k$ . In general, the PRP method may not globally converge as it does not satisfy the following so-called sufficient condition: there exists  $r > 0$  such that

$$\nabla f_k^T d_k \leq -r \|\nabla f_k\|^2 \text{ for all } k \in \mathbb{N}.$$

Note that the sufficiently descent condition usually plays an important role in the global convergent analysis of the conjugate gradient methods and so, many authors hinted that the sufficiently descent condition may be crucial for CG methods [1, 2, 15, 21]. In order to ensure that search direction has this property, many modified CG methods are presented ([17, 18, 40, 46, 47, 48, 52] etc.), where Zhang et al. [52] presented a three-terms PRP method where the search direction is defined by

$$d_{k+1} = \begin{cases} -\nabla f_{k+1} + \beta_k^{PRP} d_k - \vartheta_k y_k, & \text{if } k \geq 1 \\ -\nabla f_{k+1}, & \text{if } k = 0, \end{cases} \quad (3.2)$$

where  $\vartheta_k = \frac{\nabla f_{k+1}^T d_k}{\|\nabla f_k\|^2}$ ,  $v_{k+1} = v_k + \mu_k d_k$ ,  $\mu_k$  is the stepsize. It can be verified that  $d_k^T \nabla f_k = -\|\nabla f_k\|^2$  for all  $k$ . This method reduces to the standard PRP method if exact line search is used, its global convergence with Armijo-type line search is obtained in [52].

### Modified CG method by incorporating function value information

Before we proceed to our modified PRP algorithm, let us introduce the second accelerating technique in modifying conjugate gradient method which was used by the authors in [26]. Recall that another effective method for solving smooth optimization problem is the quasi-Newton secant method where the iterate  $x_k$  satisfy  $x_{k+1} = x_k - B_k^{-1} \nabla f_k$ , where  $B_k$  is an approximation Hessian of  $f$  at  $x_k$ . The sequence of matrix  $\{B_k\}$  satisfies the following secant equation

$$B_{k+1} s_k = y_k, \quad (3.3)$$

where  $s_k = x_{k+1} - x_k$ . Obviously, only gradient information is exploited in (3.3), while function values available are neglected. Hence, techniques using gradient values as well as function values have been studied by several authors. A significant attempt that modified

usual secant equation by using both function values and gradient values is due to Wei et al. (see [39]), where the secant equation is defined by

$$B_{k+1}s_k = y_k^*, \quad (3.4)$$

where  $y_k^* = y_k + \gamma_k^* s_k$  and  $\gamma_k^* = \frac{(\nabla f(x_{k+1}) + \nabla f(x_k))^T s_k + 2(f(x_k) - f(x_{k+1}))}{\|s_k\|^2}$ . A remarkable property of this secant equation (3.4) is that, if  $f$  is twice continuously differentiable and  $B_{k+1}$  is updated by BFGS method formula, then the following equality

$$f(x_k) = f(x_{k+1}) + \nabla f(x_{k+1})^T s_k + \frac{1}{2} s_k^T B_{k+1} s_k \quad (3.5)$$

holds for all  $k$ . Moreover, this property is independent of any convexity assumption on the objective function. The theoretical advantage of the new quasi-Newton equation (3.4) can be seen from the following theorem.

**Theorem 3.1.** ([39]) *Assume that the function  $f$  is sufficiently smooth and  $\|s_k\|$  is sufficiently small, then we have*

$$s_k^T \nabla^2 f_{k+1} s_k - s_k^T y_k^* - \frac{1}{3} s_k^T (T_{k+1} s_k) s_k = O(\|s_k\|^4) \quad (3.6)$$

and

$$s_k^T \nabla^2 f_{k+1} s_k - s_k^T y_k - \frac{1}{2} s_k^T (T_{k+1} s_k) s_k = O(\|s_k\|^4) \quad (3.7)$$

where  $\nabla^2 f_{k+1}$  denotes the Hessian matrix of  $f$  at  $x_{k+1}$ ,  $T_{k+1}$  is the tensor of  $f$  at  $x_{k+1}$ , and

$$s_k^T (T_{k+1} s_k) s_k = \sum_{i,j,l=1}^n \frac{\partial^3 f(x_{k+1})}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l.$$

The above result shows that the new quasi-Newton equation (3.4) has better approximate relation than that of (3.3). Based on this new quasi-Newton equation, various efficient new conjugate gradient method were proposed and had lead to numerical improvement by replacing  $y_k$  with  $y_k^*$  (see [43, 45, 49, 50] etc.).

Motivated by the three-terms PRP formula (3.2) and the new quasi-Newton equation (3.4), we now present a modified three-terms PRP method for solving nonsmooth optimization problem.

## New PRP-Method for nonsmooth convex program

Recall that  $g^\alpha(x_k, \varepsilon_k)$  is an approximation of the gradient of  $F$  (the Moreau-Yosida regularization of the objective function  $f$ ) at  $x_k$ . By replacing  $\nabla f_k$  and  $y_k$  with  $g^\alpha(x_k, \varepsilon_k)$  and  $y_k^*$  respectively in the three-terms PRP formula (3.2), we now propose a modified PRP conjugate gradient formula for solving (P) as follows:

$$d_{k+1} = \begin{cases} -g^\alpha(x_{k+1}, \varepsilon_{k+1}) + \frac{g^\alpha(x_{k+1}, \varepsilon_{k+1})^T y_k^* d_k - d_k^T g^\alpha(x_{k+1}, \varepsilon_{k+1}) y_k^*}{\max\{2c\|d_k\| \|y_k^*\|, \|g^\alpha(x_k, \varepsilon_k)\|^2\}}, & \text{if } k \geq 1 \\ -g^\alpha(x_{k+1}, \varepsilon_{k+1}), & \text{if } k = 0, \end{cases} \quad (3.8)$$

where  $y_k^* = y_k + \gamma_k^* s_k$ ,  $y_k = g^\alpha(x_{k+1}, \varepsilon_{k+1}) - g^\alpha(x_k, \varepsilon_k)$ ,  $s_k = x_{k+1} - x_k$ ,  $d_k$  is the search direction at the  $k$ th,  $c > 0$  is a constant, and

$$\gamma_k^* = \frac{(g^\alpha(x_{k+1}, \varepsilon_{k+1}) + g^\alpha(x_k, \varepsilon_k))^T s_k + 2(F^\alpha(x_k, \varepsilon_k) - F^\alpha(x_{k+1}, \varepsilon_{k+1}))}{\|s_k\|^2}.$$

It is worth noting that the scaling term  $\|\nabla f(x_k)\|^2$  in the denominator of the three-terms PRP formula (3.2) is adjusted to  $\max\{2c\|d_k\|\|y_k^*\|, \|g^\alpha(x_k, \varepsilon_k)\|^2\}$  in our new PRP conjugate gradient formula. This modification will help us to show that all the search direction will stay in a trust region automatically (see Lemma 3.1 below).

The algorithm of the modified PRP conjugate gradient method is stated as follows:

**Algorithm 1.** Nonmonotone Conjugate Gradient Algorithm.

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

Step 1. If  $x_k$  satisfies the termination condition  $\|g^\alpha(x_k, \varepsilon_k)\| < \epsilon$ , then stop. Otherwise, go to Step 2.

Step 2: Choose a scalar  $\varepsilon_{k+1}$  satisfying  $0 < \varepsilon_{k+1} < \varepsilon_k$ , and compute the step size  $\alpha_k$  by the following nonmonotone Armijo-type line search:

$$F^\alpha(x_k + \alpha_k d_k, \varepsilon_{k+1}) - J_k \leq \sigma \alpha_k g^\alpha(x_k, \varepsilon_k)^T d_k, \quad (3.9)$$

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

Step 3: Let  $x_{k+1} = x_k + \alpha_k d_k$ . If  $\|g^\alpha(x_{k+1}, \varepsilon_{k+1})\| < \epsilon$ , then stop. Otherwise, go to Step 4.

Step 4: Update  $J_k$  by the following formula

$$E_{k+1} = \rho E_k + 1, \quad J_{k+1} = \frac{\rho E_k J_k + F^\alpha(x_k + \alpha_k d_k, \varepsilon_{k+1})}{E_{k+1}}. \quad (3.10)$$

Step 5: Compute the search direction  $d_{k+1}$  by (3.8).

Step 6: Let  $k := k + 1$ , and go back to Step 1.

**Remark i.** The line search technique (3.9) is motivated by Zhang and Hager [51]. It is not difficult to see that  $J_{k+1}$  is a convex combination of  $J_k$  and  $F^\alpha(x_{k+1}, \varepsilon_{k+1})$ . Noticing  $J_0 = F^\alpha(x_0, \varepsilon_0)$ , it follows that  $J_k$  is a convex combination of the function values  $F^\alpha(x_0, \varepsilon_0), F^\alpha(x_1, \varepsilon_1), \dots, F^\alpha(x_k, \varepsilon_k)$ . The choice of  $\rho$  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.

The following lemma shows that our CG method satisfy the sufficiently descent property and the corresponding CG searching direction  $d_k$  belongs to a trust region automatically.



**Lemma 3.1. (Sufficient Descent Property)** For all  $k \in \mathbb{N} \cup \{0\}$ , we have

$$g^\alpha(x_k, \varepsilon_k)^T d_k = -\|g^\alpha(x_k, \varepsilon_k)\|^2 \quad (3.11)$$

and

$$\|d_k\| \leq (1 + \frac{1}{c})\|g^\alpha(x_k, \varepsilon_k)\|. \quad (3.12)$$

*Proof.* For  $k = 0$ , we have  $d_0 = -g^\alpha(x_0, \varepsilon_0)$ . So, (3.11) and (3.12) obviously hold. For  $k \geq 1$ , by the definition of  $d_k$ , we get

$$\begin{aligned} & d_{k+1}^T g^\alpha(x_{k+1}, \varepsilon_{k+1}) \\ = & -\|g^\alpha(x_{k+1}, \varepsilon_{k+1})\|^2 + \left[ \frac{g^\alpha(x_{k+1}, \varepsilon_{k+1})^T y_k^* d_k - d_k^T g^\alpha(x_{k+1}, \varepsilon_{k+1}) y_k^*}{\max\{2c\|d_k\|\|y_k^*\|, \|g^\alpha(x_k, \varepsilon_k)\|^2\}} \right]^T g^\alpha(x_{k+1}, \varepsilon_{k+1}) \\ = & -\|g^\alpha(x_{k+1}, \varepsilon_{k+1})\|^2. \end{aligned} \quad (3.13)$$

So (3.11) holds. Now we show (3.12). Using the definition of  $d_k$  again, we have

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

where the last inequality follows as

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

The proof is complete.  $\square$

## 4. Convergence Analysis

In this section, we provide a global convergence analysis for our modified PRP algorithm. In order to get the global convergence of Algorithm 1, the following assumption is assumed throughout this section.

**Assumption A.** (i) There exists a positive constant  $M$  such that

$$\|\nabla F(u_k)\| \leq M, \quad \forall u_k \in [x_k, x_{k+1}] \text{ and } \forall k \in \mathbb{N}. \quad (4.1)$$

where  $F$  is the Moreau-Yosida regularization of  $f$ .

(ii)  $F$  is bounded from below.

(iii) The sequence  $\varepsilon_k$  converges to zero.

**Remark ii.** If  $s_k \rightarrow 0$ , it is not difficult to get  $\|g^\alpha(x_{k+1}, \varepsilon_{k+1})\| - \|g^\alpha(x_k, \varepsilon_k)\| \rightarrow 0$  and  $F^\alpha(x_{k+1}, \varepsilon_{k+1}) - F^\alpha(x_k, \varepsilon_k) \rightarrow 0$ . Then, we see that

$$\|d_{k+1} - (-g^\alpha(x_{k+1}, \varepsilon_{k+1}))\| \rightarrow 0.$$

This shows that the proposed method inherits the property of the well-known PRP conjugate gradient method: the tendency to turn towards the steepest descent direction if a

small step is generated away from the solution, preventing a sequence of tiny steps from happening.

Using Lemma 3.1 and Assumption A, similar to Lemma 1.1 in [51], we can show that Algorithm 1 is well-defined as in the following lemma. As the proof is essentially the same as Lemma 1.1 in [51], we omit its proof here.

**Lemma 4.1.** *Suppose that Assumption A holds. Then, for the iterates generated by Algorithm 1, we have  $F^\alpha(x_k, \varepsilon_k) \leq J_k \leq C_k$  for each  $k$ , where  $C_k = \frac{1}{k+1} \sum_{i=0}^k F^\alpha(x_i, \varepsilon_i)$ . Moreover, there exists  $\alpha_k$  satisfying Armijo conditions of the line search update.*

**Lemma 4.2.** *Suppose that Assumption A holds. Let  $\{(x_k, \alpha_k)\}$  be the sequence generated by Algorithm 1. Suppose that  $\varepsilon_k = o(\alpha_k^2 \|d_k\|^2)$  holds. Then, there exists a constant  $m_0 > 0$  such that*

$$\alpha_k \geq m_0. \quad (4.2)$$

*Proof.* Let  $\alpha_k$  satisfy the line search (3.9). We proceed by the method of contradiction and suppose that  $\liminf_{k \rightarrow \infty} \alpha_k = 0$ . By passing to a subsequence if necessary, we may assume that  $\alpha_k \rightarrow 0$ . Then, by the line search,  $\alpha'_k = \frac{\alpha_k}{2}$  satisfies

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

Using  $F^\alpha(x_k, \varepsilon_k) \leq J_k \leq C_k$  in Lemma 4.1, we get

$$F^\alpha(x_k + \alpha'_k d_k, \varepsilon_{k+1}) - F^\alpha(x_k, \varepsilon_k) \geq F^\alpha(x_k + \alpha'_k d_k, \varepsilon_{k+1}) - J_k > \sigma \alpha'_k g^\alpha(x_k, \varepsilon_k)^T d_k. \quad (4.3)$$

By (4.3), (2.7) and Taylor's formula, we have

$$\begin{aligned} \sigma \alpha'_k g^\alpha(x_k, \varepsilon_k)^T d_k &< F^\alpha(x_k + \alpha'_k d_k, \varepsilon_{k+1}) - F^\alpha(x_k, \varepsilon_k) \\ &\leq F(x_k + \alpha'_k d_k) - F(x_k) + \varepsilon_{k+1} \\ &= \alpha'_k d_k^T g(x_k) + \frac{1}{2} (\alpha'_k)^2 d_k^T \nabla F(u_k) d_k + \varepsilon_{k+1} \\ &\leq \alpha'_k d_k^T g(x_k) + \frac{M}{2} (\alpha'_k)^2 \|d_k\|^2 + \varepsilon_{k+1}, \end{aligned} \quad (4.4)$$

where  $u_k = x_k + \iota \alpha'_k d_k$ ,  $\iota \in (0, 1)$ , and the last inequality follows from (4.1). It follows from (4.4) that

$$\begin{aligned} \frac{\alpha_k}{2} = \alpha'_k &> \left[ \frac{(g^\alpha(x_k, \varepsilon_k) - g(x_k))^T d_k - (1 - \sigma) g^\alpha(x_k, \varepsilon_k)^T d_k - \varepsilon_{k+1} / (\alpha'_k)^2}{\|d_k\|^2} \right] \frac{2}{M} \\ &\geq \left[ \frac{(1 - \sigma) \|g^\alpha(x_k, \varepsilon_k)\|^2 - \sqrt{2\varepsilon_k / \lambda} \|d_k\| - \varepsilon_k}{\|d_k\|^2} \right] \frac{2}{M} \\ &= \left[ \frac{(1 - \sigma) \|g^\alpha(x_k, \varepsilon_k)\|^2}{\|d_k\|^2} - o(\alpha_k) / \sqrt{\lambda} - o(\alpha_k^2) \right] \frac{2}{M} \\ &\geq \left[ \frac{(1 - \sigma)}{(1 + \frac{1}{c})^2} - o(\alpha_k) / \sqrt{\lambda} - o(\alpha_k^2) \right] \frac{2}{M}, \end{aligned} \quad (4.5)$$

where the equality follows  $\varepsilon_k = o(\alpha_k^2 \|d_k\|^2)$ , the second inequality follows from (2.9), (3.11) and  $\varepsilon_{k+1} \leq \varepsilon_k$ , and the last inequality follows (3.12). Dividing both sides by  $\alpha_k$  and passing to limit, we see that

$$\frac{1}{2} \geq \lim_{k \rightarrow \infty} \left( \frac{2(1 - \sigma)}{(1 + \frac{1}{c})^2 M} \right) \frac{1}{\alpha_k} = +\infty.$$

This is impossible, and so, the conclusion follows.  $\square$

We are now ready to show the global convergence of Algorithm 1.

**Theorem 4.1. (Global Convergence)** *Suppose that the conditions in Lemma 4.2 hold. Then,  $\lim_{k \rightarrow \infty} \|g(x_k)\| = 0$ , and any accumulation point of  $x_k$  is an optimal solution of (1.1).*

*Proof.* We first prove that

$$\lim_{k \rightarrow \infty} \|g^\alpha(x_k, \varepsilon_k)\| = 0. \quad (4.6)$$

To see this, we proceed by the method of contradiction and suppose that (4.6) fails. Without loss generality, we may assume that there exist constants  $\epsilon_0 > 0$  and  $k_0 > 0$  satisfying

$$\|g^\alpha(x_k, \varepsilon_k)\| \geq \epsilon_0, \quad \forall k > k_0. \quad (4.7)$$

Using (3.9), (3.11), (4.2), and (4.7), we get

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

It follows from the definition of  $J_{k+1}$  that

$$\begin{aligned} J_{k+1} &= \frac{\rho E_k J_k + F^\alpha(x_k + \alpha_k d_k, \varepsilon_{k+1})}{E_{k+1}} \\ &\leq \frac{\rho E_k J_k + J_k - \sigma m_0 \epsilon_0}{E_{k+1}} \\ &= J_k - \frac{\sigma m_0 \epsilon_0}{E_{k+1}}. \end{aligned} \quad (4.8)$$

Since  $F$  is bounded below, we see that  $F^\alpha(x, \varepsilon)$  is bounded from below. This together with  $F^\alpha(x_k, \varepsilon_k) \leq J_k$  for all  $k$ , implies that  $J_k$  is also bounded from below. By (4.8), we have

$$\sum_{k=k_0}^{\infty} \frac{\sigma m_0 \epsilon_0}{E_{k+1}} < \infty. \quad (4.9)$$

By the definition of  $E_{k+1}$ , we get  $E_{k+1} \leq k + 2$ . It follows that

$$\sum_{k=k_0}^{\infty} \frac{\sigma m_0 \epsilon_0}{E_{k+1}} \geq \sum_{k=k_0}^{\infty} \frac{\sigma m_0 \epsilon_0}{k+2} = +\infty.$$

This makes contradiction and so, (4.6) holds.

We now show the second assertion. Using (2.9), we first see that

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

This together with Assumption A(iii) implies that

$$\lim_{k \rightarrow \infty} \|g(x_k)\| = 0. \quad (4.10)$$

Let  $x^*$  be an accumulation point of  $\{x_k\}$ . Then, there exists a subsequence  $\{x_k\}_K$  satisfying

$$\lim_{k \in K, k \rightarrow \infty} x_k = x^*. \quad (4.11)$$

From properties of  $F(x)$ , we obtain  $g(x_k) = (x_k - p(x_k))/\lambda$ . Thus, by (4.10) and (4.11),  $x^* = p(x^*)$  holds. Therefore  $x^*$  is an optimal solution of (1.1).  $\square$

Before we end this section, we remark that, some conjugate gradient type methods were proposed in [27, 53, 54] for solving possible nonsmooth optimization problems very recently\*. One of the key features of our method which differs from the methods proposed in [27, 53, 54] is that our method makes use of not only the gradient information but also the function information.

## 5. More Modified Conjugate Gradient Methods

Similar to the new method (3.8) introduced in the previous section, we can employ the same idea and present various other modified conjugate gradient methods for solving problem (P). They are listed as follows:

(i) The Hestenes-Stiefel (HS) conjugate gradient formula for smooth unconstrained optimization problem (see [19])

$$d_{k+1} = \begin{cases} -\nabla f_{k+1} + \frac{\nabla f_{k+1}^T y_k}{d_k^T y_k} d_k, & \text{if } k \geq 1, \\ -\nabla f_{k+1}, & \text{if } k = 0. \end{cases}$$

Using the similar idea in the preceding section, one can propose the following *modified HS conjugate gradient formula* for solving problem (P)

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

(ii) The Liu-Storey (LS) conjugate gradient formula for smooth unconstrained optimization problem (see [28])

$$d_{k+1} = \begin{cases} -\nabla f_{k+1} + \frac{\nabla f_{k+1}^T y_k}{-d_k^T \nabla f_k} d_k, & \text{if } k \geq 1, \\ -\nabla f_{k+1}, & \text{if } k = 0. \end{cases}$$

Using the similar idea in the preceding section, one can propose the following *modified LS conjugate gradient formula* for solving problem (P)

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

(iii) The Dai-Yuan (DY) conjugate gradient formula for smooth unconstrained optimization problem (see [11])

$$d_{k+1} = \begin{cases} -\nabla f_{k+1} + \frac{\|\nabla f_{k+1}\|^2}{d_k^T y_k} d_k, & \text{if } k \geq 1, \\ -\nabla f_{k+1}, & \text{if } k = 0. \end{cases}$$

---

\*We are grateful to one referee kindly pointing these references to us

Using the similar idea in the preceding section, one can propose the following *modified DY conjugate gradient formula* for solving problem (P)

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

Similar to Algorithm 1, it is not difficult to construct new algorithms and to obtain the global convergence of these methods. For simplicity, we omit the details here.

## 6. Numerical Results

In this section, we test our modified PRP algorithm (Algorithm 1) for both small-scale problems and also large-scale problems.

**Small-scale problems.** We first test our algorithm for some small-scale problems and compare it with the proximal bundle method in [29]. All the nonsmooth problems of Table 6.1 can be found in [30]. Table 6.1 contains problem dimensions and optimum function values.

TABLE 6.1

Nr.	Problems	Dim	$f_{ops}(x)$	Nr.	Problems	Dim	$f_{ops}(x)$
1	Rosenbrock	2	0	7	LQ	2	-1.4142136
2	Crescent	2	0	8	Mifflin 1	2	-1.0
3	CB2	2	1.9522245	9	Mifflin 2	2	-1.0
4	CB3	2	2.0	10	Rosen-Suzuki	4	-44
5	DEM	2	-3	11	Shor	5	22.600162
6	QL	2	7.20	12	Colville	5	-32.348679

Where  $f_{ops}(x)$  is the optimization function value. The algorithm was implemented by Matlab 7.6, and all experiments were 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  $s = \lambda = 1$ ,  $\rho = 0.5$ ,  $\sigma = 0.8$ , and  $\varepsilon_k = 1/(k+2)^2$ . We stopped the algorithm 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 [29] (proximal bundle method, PBL) and the paper [36] (trust region concept, BT). The numerical results of PBL and BT can be found in [29].

The columns of Table 6.2 have the following meanings:

Problem: the name of the test problem. NF: the number of the function evaluations.

NI: the total number of iterations.  $f(x)$ : the function value at the final iteration.

$f_{ops}(x)$ : the optimization function evaluation.

TABLE 6.2  
Test results

Nr.	Algorithm 1	PBL	BT	$f_{ops}(x)$
	NI/NF/ $f(x)$	NI/NF/ $f(x)$	NI/NF/ $f(x)$	
1	46/48/7.091824 $\times 10^{-7}$	42/45/0.381 $\times 10^{-6}$	79/88/0.130 $\times 10^{-11}$	0
2	11/13/6.735123 $\times 10^{-5}$	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	2/6/2.000098	14/16/2.0	13/21/2.0	2.0
5	4/6/-2.999866	17/19/-3.0	9/13/-3.0	-3
6	10/12/7.200011	13/15/7.2000015	12/17/7.200009	7.20
7	2/3/-1.414214	11/12/-1.4142136	10/11/-1.414214	-1.4142136
8	4/6/-0.9919815	66/68/-0.99999941	49/74/-1.0	-1.0
9	20/23/-0.9999925	13/15/-1.0	6/13/-1.0	-1.0
10	28/58/-43.99986	43/45/-43.999999	22/32/-43.99998	-44
11	33/91/22.60023	27/29/22.600162	29/30/-22.60016	22.600162
12	17/23/-32.34329	62/64/-32.348679	45/45/-32.3486	-32.348679

From the numerical results in Table 6.2, it can be seen that Algorithm 1 performs overall the best among the three methods for the tested problem listed and the PBL method is competitive to the BT method.

**Large-scale problems.** As the conjugate gradient-type method is particular useful for large scale problem. We also present some numerical experiment for large scale non-smooth convex problems. In particular, as the recent successful limited memory bundle method in [16] is proved to be one of the most efficient methods for solving large scaled nonsmooth convex problem, we compare our algorithm with it. The following problems of Table 6.3 can be found in [16], where Problems 1-3 are convex function and others are non-convex function. The numbers of variables used were 1000, 5000, 10000, and 50000. The values of parameters were similar to the small-scale problems. The following experiments were implemented in Fortran 90. In order to show the performance of Algorithm 1, we compared it with the method (LMBM) of paper [16]. The stopping rule and parameters were set as in [16].

TABLE 6.3

Nr.	Problems	$x_0$	Nr.	Problems	$x_0$
1	Generalization of MAXQ	$(1, 2, \dots, n/2, -(n/2 + 1), \dots, -n)$	5	Nonsmooth generalization of Brown function 2	$(1, 0, \dots)$
2	Generalization of MXHILB	$(1, 1, \dots)$	6	Chained Mifflin 2	$(-1, -1, \dots)$
3	Chained LQ	$(-0.5, -0.5, \dots)$	7	Chained Crescent I	$(-1.5, 2, \dots)$
4	Number of active faces	$(1, 1, \dots)$	8	Chained Crescent II	$(1, 0, \dots)$

LMBM [16]. 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.karmita.fi/lmbm/>.

TABLE 6.4

Test results

Nr.	Dim	Algorithm 1	LMBM
		NI/NF/ $f(x)$	NI/NF/ $f(x)$
1	1000	$225/4710/6.93540513648899 \times 10^{-8}$	$21492/22259/6.71025884158921 \times 10^{-6}$
	5000	$250/5235/6.87979764297566 \times 10^{-8}$	$191470/196034/3.44987308783816 \times 10^{-5}$
	10000	$261/5466/6.65278880046151 \times 10^{-8}$	$512415/523351/5.83498629004021 \times 10^{-5}$
	50000	$286/5991/6.59944730089708 \times 10^{-8}$	$4999996/5000000/5.77622918225693 \times 10^{-2}$
2	1000	$91/1482/8.27377560377777 \times 10^{-9}$	$441/861/6.16640522014495 \times 10^{-3}$
	5000	$111/1938/9.72057262375235 \times 10^{-9}$	$1258/2487/3.52143538240154 \times 10^{-2}$
	10000	$120/2127/5.85235371545923 \times 10^{-9}$	$7027/7810/5.11605598924595 \times 10^{-2}$
	50000	$141/2604/6.17794513724740 \times 10^{-9}$	$1036/1515/2.88716335417572$
3	1000	$37/114/7.26870782386942 \times 10^{-9}$	$300/1824/-1.41277614588146 \times 10^5$
	5000	$39/120/9.09316258227670 \times 10^{-9}$	$365/2198/-7.06961404239955 \times 10^5$
	10000	$40/123/9.09407298984668 \times 10^{-9}$	$376/2281/-1.41406858071499 \times 10^6$
	50000	$55/153/4.54740666059811 \times 10^{-8}$	$582/2998/-7.07092005296846 \times 10^6$
4	1000	$77/1026/6.80373977063602 \times 10^{-9}$	$523/569/1.37667655053518 \times 10^{-14}$
	5000	$90/1281/7.84048733972188 \times 10^{-9}$	$2585/2586/1.21306742421471 \times 10^{-10}$
	10000	$96/1401/9.93659283247496 \times 10^{-9}$	$5069/5073/5.38381117320365 \times 10^{-10}$
	50000	$110/1665/6.13431592528028 \times 10^{-9}$	$184/217/9.99980628711393 \times 10^6$
5	1000	$38/117/7.26870855073998 \times 10^{-9}$	$467/3873/4.05785228342877 \times 10^{-9}$
	5000	$40/123/9.09316349159170 \times 10^{-9}$	$453/4073/1.08041333809065 \times 10^{-8}$
	10000	$41/125/1.81881459796775 \times 10^{-8}$	$736/7453/2.52215161529255 \times 10^{-8}$
	50000	$55/153/9.09481332115074 \times 10^{-8}$	$1293/10995/7.26535476752977 \times 10^{-7}$
6	1000	$37/114/-2.49749999992731 \times 10^4$	$1254/7355/-7.06476909407459 \times 10^4$
	5000	$39/120/-1.24974999999091 \times 10^5$	$219/782/-3.53493693696815 \times 10^5$
	10000	$40/123/-2.49974999999091 \times 10^5$	$267/743/-7.07042033377240 \times 10^5$
	50000	$43/132/-1.24997499999986 \times 10^6$	$532/2220/-3.53546169719885 \times 10^6$
7	1000	$37/114/5.48971001990139 \times 10^{-9}$	$138/560/2.46289254738352 \times 10^{-4}$
	5000	$39/120/6.82939571561292 \times 10^{-9}$	$116/281/2.45751603804887 \times 10^2$
	10000	$40/123/6.82530298945494 \times 10^{-9}$	$188/267/2.00248216613019 \times 10^{-5}$
	50000	$56/157/8.52753601066070 \times 10^{-9}$	$391/725/4.75680539402390 \times 10^{-9}$
8	1000	$39/120/6.81848177919164 \times 10^{-9}$	$763/7522/1.39417095132655 \times 10^{-4}$
	5000	$41/126/8.52583070809487 \times 10^{-9}$	$943/8490/1.59176433435915 \times 10^{-3}$
	10000	$42/129/8.52617176860804 \times 10^{-9}$	$1364/13919/1.06009303953474 \times 10^{-2}$
	50000	$87/222/5.32902788563661 \times 10^{-9}$	$4657/61720/8.01042336673219 \times 10^{-4}$

For these large-scale problems, the iteration number and the number of function evaluations of Algorithm 1 are competitive to those of the LMBM method. Moreover, the

number does not change obviously when the dimension increases. The final function value of the given algorithm is better than those of LMBM except for Problem 3. The performance of the numerical results show that Algorithm 1 can also be used to solve not only smooth convex optimization problems but also nonconvex nonsmooth optimization problems. Taking everything together, the preliminary numerical results indicate the proposed method is efficient.

## 7. Conclusion

The CG method has the simplicity and the very low memory requirement and The PRP method is one of the most effective conjugate gradient methods. By making use of the Moreau-Yosida regularization, a nonmonotone line search technique of [51] and a new secant equation of [39] derived by the authors earlier, we present a modified PRP conjugate gradient algorithm for solving nonsmooth convex optimization problems. Our method satisfies the sufficiently descent property automatically, and the corresponding search direction belongs to a trust region. Another interesting feature of our method is that it involves not only the gradient information but also the function information. Numerical results show that this method is effective and is more competitive than existing methods for both small-scale and large-scale nonsmooth problems.

It would be interesting to see how our algorithm performs if we apply it to solve some optimization problem arises in the image processing area. It would be also interesting to compare the performance of our method with the method proposed in [27, 53, 54]. These will be our future research topics and will be examined in a forthcoming paper.

**Acknowledgement:** The authors are grateful to the referees and the editor for their constructive comments and helpful suggestions which have contributed to the final preparation of the paper.

## References

- [1] T. Ahmed, D. Storey, Efficient hybrid conjugate gradient techniques, *Journal of Optimization Theory and Applications*, 64(1990), pp. 379-394.
- [2] A. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA Journal of Numerical Analysis*, 5(1985), pp. 121-124.
- [3] M. R. Banham, A. K. Katsaggelos, Digital image restoration, *IEEE Signal Processing Magazine*, 14(1997), pp. 24-41.
- [4] J. R. Birge, L. Qi, Z. Wei, A general approach to convergence properties of some methods for nonsmooth convex optimization, *Applied Mathematics & Optimization*, 38(1998), pp. 141-158.
- [5] J. R. Birge, L. Qi, Z. Wei, Convergence analysis of some methods for minimizing a nonsmooth convex function, *Journal of Optimization and Applications*, 97(1998), pp. 357-383.

- [6] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. A. Sagastizábal, A family of variable metric proximal methods, *Mathematical Programming*, 68(1995), pp. 15-47.
- [7] P. H. Calamai, J. J. Moré, Projected gradient methods for linear constrained problems, *Mathematical Programming*, 39(1987), pp. 93-116.
- [8] C. L. Chan, A. K. Katsaggelos, A. V. Sahakian, Image sequence filtering in quantum-limited noise with applications to low-dose fluoroscopy, *IEEE Trans. Med. Imag.*, 12(1993), pp. 610-621.
- [9] A. R. Conn, N. I. M. Gould, P. L. Toint, Trust-region methods, SIAM, Philadelphia, USA, 2000.
- [10] R. Correa, C. Lemaréchal, Convergence of some algorithms for convex minimization, *Mathematical Programming*, 62(1993), pp. 261-273.
- [11] Y. Dai, Y. Yuan, A nonlinear conjugate gradient with a strong global convergence properties, *SIAM Journal on Optimization*, 10(2000), pp. 177-182.
- [12] R. Fletcher, Practical method of optimization, Vol I: Unconstrained Optimization, 2nd edition, Wiley, New York, 1997.
- [13] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, *The Computer Journal*, 7(1964), pp. 149-154.
- [14] M. Fukushima, L. Qi, A global and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM Journal on Optimization*, 6(1996), pp. 1106-1120.
- [15] J. C. Gilbert, J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM Journal on Optimization*, 2(1992), pp. 21-42.
- [16] M. Haarala, K. Miettinen, and M. M. Mäkelä, New limited memory bundle method for large-scale nonsmooth optimization, *Optimization Methods and Software*, 19(2004), pp. 673-692.
- [17] W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and an efficient line search, *SIAM Journal on Optimization*, 16(2005), pp. 170-192.
- [18] W. Hager, H. Zhang, Algorithm 851: *CG\_DESCENT*, A conjugate gradient method with guaranteed descent, *ACM Transactions on Mathematical Software*, 32(2006), pp. 113-137.
- [19] M. R. Hestenes, E. Stiefel, Method of conjugate gradient for solving linear equations, *Journal of Research of the National Bureau of Standards*, 49(1952), pp. 409-436.
- [20] J. B. Hiriart-Urruty, C. Lemaréchal, Convex analysis and minimization algorithms II, Springer-Verlag, Berlin, Heidelberg, 1983.
- [21] Y. F. Hu, C. Storey, Global convergence result for conjugate method, *Journal Optimization Theory and Applications*, 71(1991), pp. 399-405.



- [22] K. C. Kiwiel, *Methods of descent for nondifferentiable optimization*, Springer-Verlag, Berlin, New York, 1985.
- [23] K. C. Kiwiel, Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities, *Mathematical Programming*, 69(1995), pp. 89-109.
- [24] K. C. Kiwiel, Proximity control in bundle methods for convex nondifferentiable optimization, *Mathematical Programming*, 46(1990), pp. 105-122.
- [25] C. Lemaréchal, *Extensions diverses des méthodes de gradient et applications*, Thèse d'Etat, Paris, 1980.
- [26] G. Li, C.M. Tang, Z.X. Wei, New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. *J. Comput. Appl. Math.* 202 (2007), no. 2, 523-539.
- [27] Q. Li, Conjugate gradient type methods for the nondifferentiable convex minimization, *Optimization Letters*, 7 (2013), no. 3, pp. 533-545.
- [28] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms part 1: Theory, *Journal of Applied Mathematics and Computing*, 69(1992), pp. 17-41.
- [29] L. Lukšan, J. Vlček, A bundle-Newton method for nonsmooth unconstrained minimization, *Mathematical Programming*, 83(1998), pp. 373-391.
- [30] L. Lukšan, 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.
- [31] E. Polak, G. Ribière, Note sur la convergence de directions conjuguées, *Rev. Française informat Recherche Opérative*, 3(1969), pp. 35-43.
- [32] B. T. Polyak, The conjugate gradient method in extreme problems, *USSR Comp. Math. Math. Phys.*, 9(1969), pp. 94-112.
- [33] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Mathematics of Operations Research*, 18(1993), pp. 227-245.
- [34] R. T. Rockafellar, Monotone operators and proximal point algorithm, *SIAM Journal on Control and Optimization*, 14(1976), pp. 877-898.
- [35] H. Schramm, *Eine kombination von bundle-und trust-region-verfahren zur Lösung nicht- differenzierbare optimierungsprobleme*, Bayreuther Mathematische Schriften, Heft 30, Bayreuth, Germany, 1989.
- [36] H. Schramm, J. Zowe, A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results, *SIAM Journal on Optimization*, 2(1992), pp. 121-152.

- [37] C. H. Slump, Real-time image restoration in siagnostic X-ray imaging, the effects on quantum noise, in Proceedings of he 11th IAPR International Conference on Pattern Recognition, Vol. II, Conference B: Pattern Recongnition methodology and Systems, 1992, pp. 693-696.
- [38] Z. Wei, G. Li, L. Qi, New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems, *Applied Mathematics and Computation*, 179(2006), pp. 407-430.
- [39] Z. Wei, G. Li, L. Qi, New quasi-Newton methods for unconstrained optimization problems, *Applied Mathematics and Computation*, 175(2006), pp. 1156-1188.
- [40] Z. Wei, G. Li, L. Qi, Global convergence of the Polak-Ribiere-Polyak conjugate gradient methods with inexact line search for nonconvex unconstrained optimization problems, *Mathematics of Computation*, 77(2008), pp. 2173-2193.
- [41] Z. Wei, L. Qi, Convergence analysis of a proximal Newton method, *Numerical Functional Analysis and Optimization*, 17(1996), pp. 463-472.
- [42] Z. Wei, L. Qi, J. R. Birge, A new methods for nonsmooth convex optimization, *Journal of Inequalities and Applications*, 2(1998), pp. 157-179.
- [43] Z. Wei, G. Yu, G. Yuan, Z. Lian, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, *Computational Optimization and Application*, 29(2004), pp. 315-332.
- [44] P. Wolfe, A method of conjugate subgradients for minimizing nondifferentiable convex functions, *Mathematical Programming Study*, 3(1975), pp. 145-173.
- [45] Y. Xiao, Z. Wei, Z. Wang, A limited memory BFGS-type method for large-scale unconstrained optimization, *Computational Mathematics Applications*, 56(2008), pp. 1001-1009.
- [46] G. Yuan, Modified nonlinear conjugate gradient methods with sufficient descent property for large-scale optimization problems, *Optimization Letters*, 3(2009), pp. 11-21.
- [47] G. Yuan, X. Lu, A modified PRP conjugate gradient method, *Annals of Operations Research*, 166(2009), pp. 73-90.
- [48] G. Yuan, X. Lu, Z. Wei, A conjugate gradient method with descent direction for unconstrained optimization, *Journal of Computational and Applied Mathematics*, 233(2009), pp. 519-530.
- [49] G. Yuan, Z. Wei, Convergence analysis of a modified BFGS method on convex minimizations, *Computational Optimization and Applications*, 47(2010), pp. 237-255.
- [50] G. Yuan, Z. Wei, Y. Wu, Modified limited memory BFGS method with nonmonotone line search for unconstrained optimization, *Journal of The Korean Mathematical Society*, 47(2010), pp. 767-788.

- [51] H. Zhang, W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM Journal on Optimization*, 14(2004), pp. 1043-1056.
- [52] L. Zhang, W. Zhou, D. Li, A descent modified Polak-Ribière-Polyak conjugate method and its global convergence, *IMA Journal on Numerical Analysis*, 26(2006), pp. 629-649.
- [53] L. Zhang , J. Li, A new globalization technique for nonlinear conjugate gradient methods for nonconvex minimization, *Applied Mathematics and Computation*, 217 (2011) pp.10295-10304.
- [54] W. Zhou, A globally and R-linearly convergent hybrid HS and PRP method and its inexact version with applications, [http://www.optimization-online.org/DB\\_HTML/2011/11/3241.html](http://www.optimization-online.org/DB_HTML/2011/11/3241.html), (2011).