

RESEARCH

Open Access



A limited memory BFGS subspace algorithm for bound constrained nonsmooth problems

Xiangrong Li^{1*} 

*Correspondence: xrli68@163.com

¹Business School, College of Mathematics and Information Science, Guangxi University, Nanning, P.R. China

Abstract

The subspace technique has been widely used to solve unconstrained/constrained optimization problems and there exist many results that have been obtained. In this paper, a subspace algorithm combining with limited memory BFGS update is proposed for large-scale nonsmooth optimization problems with box-constrained conditions. This algorithm can ensure that all iteration points are feasible and the sequence of objective functions is decreasing. Moreover, rapid changes in the active set are allowed. The global convergence is established under some suitable conditions. Numerical results show that this method is very effective for large-scale nonsmooth box-constrained optimization, where the largest dimension of the test problems is 11,000 variables.

MSC: 65H10; 90C26

Keywords: Nonsmooth; Large scale; Limited BFGS method; Global convergence

1 Introduction

Consider the following large-scale nonsmooth optimization problems:

$$\min f(x), \quad \text{s.t.} \quad l \leq x \leq u, \quad (1.1)$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is supposed to be locally Lipschitz continuous and the number of variables n is supposed to be large. The vectors l and u represent the lower and the upper bounds on the variables, respectively.

Many practical optimization problems involve nonsmooth functions with large amounts of variables (see, e.g., [1, 24]). The active-set method can be easily generalized when the objective function is nonsmooth. For example, Sreedharan [34] extends the method developed in [33] to solve nonsmooth problem with a special objective function and inequality constraint. Also, it is quite easy to generalize the ε -active-set method to the nondifferentiable case [25]. Yuan et al. [42, 45] use the two-point gradient method and the trust region method to solve nonsmooth problems. Wolfe [35] and Lemaréchal [22] initiated a giant stride forward in nonsmooth optimization by the bundle concept. Kiwiel [20] proposed a bundle variant, which is close to bundle trust iteration method [32]. Some good results about the bundle technique can be found in [19, 21, 31] etc. The basic assumption of the

© The Author(s) 2020. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

bundle methods is that at every point $x \in \mathfrak{N}^n$, the value of the objective function $f(x)$ and an arbitrary subgradient $\xi \in \mathfrak{N}^n$ from the subdifferential can be evaluated [7] with

$$\partial f(x) = \text{conv} \left\{ \lim_{t \rightarrow \infty} \nabla f(x_t) \mid x_t \rightarrow x \text{ and } \nabla f(x_t) \text{ exists} \right\}$$

where “conv” denotes the convex hull of a set. The idea behind bundle methods is to approximate $\partial f(x)$ by gathering subgradient information from previous iterations into a bundle. At the moment, various versions of bundle methods are regarded as the most effective and reliable methods for nonsmooth optimization. Bundle methods are efficient for small- and medium-scale problems. This is explained by the fact that bundle methods need relatively large bundles to be capable of solving the problems efficiently [19]. At present, Haarala et al. (see [14, 15] etc.) introduce the limited memory bundle methods for large-scale nonsmooth unconstrained and constrained minimization, which are a hybrid of the variable metric bundle methods and the limited memory variable metric methods and some good results are obtained. The dimension of the test problems can be arrived one thousand variables. For unconstrained nonsmooth problems, Karmita et al. [18] tested and compared different methods from both groups and some methods which may be considered as hybrids of these two and/or some others, where the largest dimension of test nonsmooth problems is 4000. Yuan et al. [38–40, 43, 46] present the conjugate gradient algorithms for solving unconstrained nonsmooth problems and the nonsmooth problems with 60,000 variables are successfully solved.

Normally the nonsmooth optimization problems are difficult to solve, even when they are unconstrained. Derivative free methods, like Powell’s method [11] or genetic algorithms [13] may be unreliable and become inefficient whenever the dimension of the problem increases. The direct application of smooth gradient-based methods to nonsmooth problems may lead to a failure in optimality conditions, in convergence, or in gradient approximation [23]. Therefore, special tools for solving nonsmooth optimization problems are needed, especially for constrained nonsmooth problems. In this paper, we present a new algorithm that combines an active-set strategy with the gradient projection method. The active sets are based on guessing technique to be identified at each iteration, the search direction in free subspace is determined by limited memory BFGS algorithm, which will provide an efficient means for attacking large-scale nonsmooth bound constrained optimization problems. This paper has the following main attributes:

- ♣ the nonsmooth objective function is descent;
- ♣ a limited memory BFGS method is given for nonsmooth problem; the iteration sequence $\{x_k\}$ is feasible;
- ♣ the global convergence of the new method is established;
- ♣ large-scale nonsmooth problems (11,000 variables) are successfully solved.

This paper is organized as follows. In the next section, we briefly review some nonsmooth analysis, the L-BFGS method for unconstrained optimization, and the motivation based on these techniques. In Sect. 3, we describe the active-set algorithm for (1.1). The global convergence will be established in Sect. 4. Numerical results are reported in Sect. 5. Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm of vectors or matrix.

2 Motivation based on nonsmooth analysis and the L-BFGS update

This section will state some results on nonsmooth analysis and the L-BFGS formula, respectively.

2.1 Results of convex analysis and nonsmooth analysis

Let $f^{MY} : \mathfrak{R}^n \rightarrow \mathfrak{R}$ be the so-called Moreau–Yosida regularization of f and be defined by

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

where λ is a positive parameter. Then it is not difficult to see that the problem (1.1) is equivalent to the following problem:

$$\min_{x \in \mathbf{B}} f^{MY}(x), \mathbf{B} = \{x \mid l \leq x \leq u\}. \tag{2.2}$$

The function f^{MY} has some good properties: it is a differentiable convex function and has a Lipschitz continuous gradient even when the function f is nondifferentiable. The gradient function of f^{MY} can be proved to be semismooth under some reasonable conditions [12, 30]. Based on these features, many algorithms have been proposed for (2.2) (see [2] etc.) when $\mathbf{B} = \mathfrak{R}^n$. Set

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

and denote $p(x) = \operatorname{argmin} \theta(z)$. Then $p(x)$ is well-defined and unique since $\theta(z)$ is strongly convex. By (2.1), $f^{MY}(x)$ can be expressed by

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

In what follows, we denote the gradient of f^{MY} by g . Some features about $f^{MY}(x)$ can be seen in [3, 8, 16]. The generalized Jacobian of $f^{MY}(x)$ and the property of BD-regular can be found in [6, 29], respectively. Some properties are given as follows.

- (i) The function f^{MY} is finite-valued, convex, and everywhere differentiable with

$$g(x) = \nabla f^{MY}(x) = \frac{x - p(x)}{\lambda}. \tag{2.3}$$

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

$$\|g(x) - g(y)\| \leq \frac{1}{\lambda} \|x - y\|, \quad \forall x, y \in \mathfrak{R}^n. \tag{2.4}$$

- (ii) If g is BD-regular at x , which means all matrices $V \in \partial_B g(x)$ are nonsingular, then there exist constants $\mu_1 > 0$, $\mu_2 > 0$ and a neighborhood Ω of x such that

$$d^T V d \geq \mu_1 \|d\|^2, \quad \|V^{-1}\| \leq \mu_2, \quad \forall d \in \mathfrak{R}^n, V \in \partial_B g(x).$$

It is obviously that $f^{MY}(x)$ and $g(x)$ can be obtained through the optimal solution of $\operatorname{argmin}_{z \in \mathfrak{R}^n} \theta(z)$. However, $p(x)$, the minimizer of $\theta(z)$, is difficult or even impossible to solve exactly. Such makes that we cannot apply the exact value of $p(x)$ to define $f^{MY}(x)$ and $g(x)$, thus the numerical methods are often used to solve it. In the following, we always suppose that the results (i)–(ii) holds without special notes.

2.2 L-BFGS update

At every iteration x_k , the L-BFGS method stores a small number (say m) of correction pairs $\{s_i, y_i\}$ ($i = k - 1, \dots, k - m$) to get H_{k+1} , instead of storing the matrices H_k with

$$s_k = x_{k+1} - x_k, y_k = \nabla h_{k+1} - \nabla h_k,$$

where $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function. In fact, this method is an adaptation of the BFGS method to large-scale problems (see [4, 5, 37, 41, 44] for details). The L-BFGS update formula is defined by

$$\begin{aligned} H_{k+1} &= V_k^T [V_{k-1}^T H_{k-1} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^T] V_k + \rho_k s_k s_k^T \\ &= V_k^T V_{k-1}^T H_{k-1} V_{k-1} + V_k^T \rho_{k-1} s_{k-1} s_{k-1}^T V_k + \rho_k s_k s_k^T \\ &= \dots \\ &= [V_k^T \dots V_{k-m+1}^T] H_{k-m+1} [V_{k-m+1} \dots V_k] \\ &\quad + \rho_{k+m-1} [V_{k-1}^T \dots V_{k-m+2}^T] s_{k-m+1} s_{k-m+1}^T [V_{k-m+2} \dots V_{k-1}] + \dots + \rho_k s_k s_k^T, \end{aligned} \tag{2.5}$$

where $\rho_k = \frac{1}{y_k^T s_k}$ and $V_k = I - \rho_k y_k s_k^T$. These correction pairs contain information about the curvature of the function and, in conjunction with the BFGS formula, define the limited memory iteration matrix. This method often provides a fast rate of linear convergence and requires minimal storage.

It is well known that the positive definiteness of update matrix H_k is very important to analyze the convergence of the algorithm. Byrd et al. [4] show that the limited memory BFGS matrix has this property if the curvature $s_k^T y_k > 0$ is satisfied. Similarly Powell [28] proposes that y_k should be designed by

$$y_k = \begin{cases} y_k, & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \theta_k y_k + (1 - \theta_k) B_k s_k, & \text{otherwise,} \end{cases}$$

where $\theta_k = \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}$, B_k is an approximation of $\nabla^2 h(x_k)$ and $B_k = H_k^{-1}$.

Inspired by the Moreau–Yosida regularization and the limited memory technique, we will give a limited memory BFGS method for box-constrained optimization with nonsmooth objective function. In the given algorithm, we also combine an active-set strategy with the gradient projection method. The techniques of the following algorithm are similar to those in Facchinei and Lucidi [10], Ni and Yuan [26], and Xiao and Wei [36], where the main difference lies in solving of the nonsmooth optimization problem.

3 Algorithm

Setting the feasible region $\mathbf{B} = \{x \in \mathbb{R}^n : l_i \leq x^i \leq u_i, i = 1, \dots, n\}$, where x^i denotes the i th element of vector x . A vector $\bar{x} \in \mathbf{B}$ is said to be a stationary point for problem (2.2) if the following relations:

$$\begin{cases} l_i = \bar{x}^i & \Rightarrow g^i(\bar{x}) \geq 0, \\ l_i < \bar{x}^i < u_i & \Rightarrow g^i(\bar{x}) = 0, \\ \bar{x}^i = u_i & \Rightarrow g^i(\bar{x}) \leq 0, \end{cases} \tag{3.1}$$

hold, where g^i is the i th element of vector $g(x)$. Considering the observation of the above section, we will first solve (2.2). And we will develop its solution to problem (1.1). The iterative method is used to solve (2.2) and defined by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \tag{3.2}$$

where α_k is a steplength and d_k is a search direction of f^{MY} at x_k . Let $\bar{x} \in \mathbf{B}$ be a stationary point of problem (2.2) and define the active constraint set

$$\bar{\Gamma} = \{i : l_i = \bar{x}^i\}, \quad \bar{\Psi} = \{i : \bar{x}^i = u_i\}, \tag{3.3}$$

thus we can define the set of the free variables by

$$\bar{\Upsilon} = \{1, \dots, n\} \setminus (\bar{\Gamma} \cup \bar{\Psi}).$$

Therefore (3.1) can be rewritten as

$$\begin{cases} g^i(\bar{x}) \geq 0, & \forall i \in \bar{\Gamma}, \\ g^i(\bar{x}) = 0, & \forall i \in \bar{\Upsilon}, \\ g^i(\bar{x}) \leq 0, & \forall i \in \bar{\Psi}. \end{cases} \tag{3.4}$$

It is reasonable to define the approximation $\Gamma(x)$, $\Upsilon(x)$ and $\Psi(x)$ to $\bar{\Gamma}$, $\bar{\Upsilon}$ and $\bar{\Psi}$, respectively:

$$\begin{aligned} \Gamma(x) &= \{i : x^i \leq l_i + a_i(x)g^i(x)\}, \\ \Psi(x) &= \{i : x^i \geq u_i + b_i(x)g^i(x)\}, \\ \Upsilon(x) &= \{1, \dots, n\} \setminus (\Gamma \cup \Psi), \end{aligned} \tag{3.5}$$

where a_i and b_i are nonnegative continuous bounded from above on \mathbf{B} , which have the properties, namely if $x^i = l_i$ or $x^i = u_i$ then $a_i(x) > 0$ or $b_i(x) > 0$, respectively. Similar to Theorem 3 in [10] for smooth optimization, we can get some results about $\Gamma(x)$, $\Upsilon(x)$, $\Psi(x)$, $\bar{\Gamma}$, $\bar{\Upsilon}$, and $\bar{\Psi}$ in nonsmooth problems.

Theorem 3.1 *For any feasible point x , $\Gamma(x) \cap \Psi(x) = \emptyset$. Suppose that strict complementarity holds and \bar{x} is a stationary point of problem (2.2), then there exists a neighborhood of \bar{x} and every feasible point x in the neighborhood, the following relations:*

$$\Gamma(x) = \bar{\Gamma}, \quad \Upsilon(x) = \bar{\Upsilon}, \quad \Psi(x) = \bar{\Psi},$$

hold.

Proof For any feasible x , if $k \in \Gamma(x)$, it is obviously that $g^k(x) \geq 0$ holds. Suppose that $k \in \Psi(x)$, then $u_k \geq x^k \geq u_k + b_k(x)g^k(x) \geq u_k$ is true. This implies that $l_k = x^k = u_k$ and $g^k(x) = 0$, which is a contradiction. Then $\Gamma(x) \cap \Psi(x) = \emptyset$ holds.

Now we prove that the second conclusion of this theorem holds. If $i \in \bar{\Gamma}$, by the definition of $\bar{\Gamma}$ and the strict complementarity, then $g^i(\bar{x}) > 0$ holds. Since a_i is nonnegative,

$\bar{x}^i \leq l_i + a_i(\bar{x})g^i(\bar{x})$. Since both a_i and g^i are continuous in \bar{x} , we deduce that $i \in \Gamma(x)$ holds. So $\bar{\Gamma} \subseteq \Gamma(x)$ is true.

Otherwise if $i \in \Gamma(x)$, then by the definition of $\Gamma(x)$, $a_i(x)g^i(x) \geq x^i - l_i \geq 0$ holds. Since a_i is nonnegative, $g^i(x) > 0$ holds. Since g^i is continuous in \bar{x} and the strict complementarity holds, we deduce that $i \in \bar{\Gamma}$ holds. Thus $\Gamma(x) \subseteq \bar{\Gamma}$ holds.

Therefore, $\Gamma(x) = \bar{\Gamma}$ holds. By the similar way, we can obtain $\Upsilon(x) = \bar{\Upsilon}$ and $\Psi(x) = \bar{\Psi}$. The proof is complete. \square

Theorem 3.1 proves that $\Gamma(x)$, $\Upsilon(x)$ and $\Psi(x)$ are “good” estimate of $\bar{\Gamma}$, $\bar{\Upsilon}$, and $\bar{\Psi}$, respectively. In the next, we give the choices of the direction d_k and the stepsize α_k along with the current point $x_k \in \mathbf{B}$, respectively. Consider the sets $\Gamma_k = \Gamma(x_k)$, $\Upsilon_k = \Upsilon(x_k)$ and $\Psi_k = \Psi(x_k)$, the search direction $d_k = (d_k^{\Gamma_k}, d_k^{\Upsilon_k}, d_k^{\Psi_k})$ is chosen as

$$d_k^i = x_k^i - l_i, \quad i \in \Gamma_k; \tag{3.6}$$

$$d_k^i = u_i - x_k^i, \quad i \in \Psi_k; \tag{3.7}$$

$$d_k^i = -(Z\bar{H}_k Z^T g_k), \quad i \in \Upsilon_k. \tag{3.8}$$

where $d_k^{\Upsilon_k}$ denotes the subspace direction for the inactive variables, $\bar{H}_k = Z^T H_k Z \in \mathfrak{R}^{|\Upsilon_k| \times |\Upsilon_k|}$ is an approximation of the reduced inverse Hessian matrix, H_k is an approximation of the full space inverse Hessian matrix, and Z is the matrix with columns $\{e^i \mid i \in \Upsilon_k\}$ and e^i is the i th column of the identity matrix in $\mathfrak{R}^{n \times n}$.

For smooth optimization problems, several authors use the projected search for quadratic and nonlinear programming problems with bounds (see [27]). The projected search finds a steplength $\alpha_k = \beta^k > 0$ with sufficient decrease in the function $\phi_k : \mathfrak{R} \rightarrow \mathfrak{R}$ such that

$$\phi_k(\alpha) \leq \phi_k(0) + \sigma \nabla \phi_k(0)\alpha, \tag{3.9}$$

where $\beta \in (0, 1)$ and $\sigma \in (0, \frac{1}{2})$ are constants,

$$\phi_k(\alpha) = f^{\text{MY}}([x_k + \alpha d_k]^+),$$

$[\cdot]^+$ is the projection into \mathbf{B} defined by

$$[x]^+ = \begin{cases} x^i, & \text{if } l_i \leq x^i \leq u_i, \\ l_i, & \text{if } x^i < l_i, \\ u_i, & \text{if } x^i > u_i. \end{cases} \tag{3.10}$$

In this paper, we also use this technique to determine the steplength α_k . Based on the selections about d_k and α_k and let $ms \leq m$ be the number of the correction pairs, we give the steps of this algorithm.

Sub-algorithm Up-to-date $(ms, \{\bar{s}_k\}, \{\bar{y}_k\}, H_0, d, Z)$

Step 1: $d = Z^T d$;

Step 2: if $ms = 0$, $d = H_0 d$, return;

Step 3: $\alpha = (\bar{s}_k^{ms-1})^T d / (\bar{y}_k^{ms-1})^T \bar{s}_k^{ms-1}$; $d = d - \alpha \bar{y}_k^{ms-1}$;

Step 4: Call Up-to-date $(ms - 1, \{\bar{s}_k\}, \{\bar{y}_k\}, H^0, d, Z)$;

Step 5: $d = d + (\alpha - (d^T \bar{y}_k^{ms-1} / (\bar{y}_k^{ms-1})^T \bar{z}_k^{ms-1})) \bar{s}_k^{n-1}$;

Step 6: $d = Z'd$.

Now we state the algorithm for nonsmooth optimization problems (2.2) with bound constrained conditions.

Algorithm 1 (main-Algorithm)

Step 1: Given $x_0 \in \mathbf{B}$, constants $\sigma \in (0, \frac{1}{2})$ and $m \in (3, 20)$, initial matrix θI , $a_i(x)$ and $b_i(x)$; set $k = 0$.

Step 2: Using (3.5) to decide $\Gamma_k = \Gamma(x_k)$, $\Psi_k = \Psi(x_k)$, and $\Upsilon_k = \Upsilon(x_k)$.

Step 3: According to (3.6)–(3.8), obtain d_k .

Step 4: $d_k = 0$, stops.

Step 5: Find α_k by (3.9).

Step 6: Let $x_{k+1} = [x_k + \alpha_k d_k]^+$ and get $f^{\text{MY}}(x_{k+1})$ and $g(x_{k+1})$.

Step 7: Update H_k by (2.5).

Step 8: Set $k = k + 1$ and go to Step 2.

Remark The given algorithm can be regarded as an extension of [36] from smooth optimization to nonsmooth optimization.

4 Global convergence

Assumption A The matrix \bar{H}_k ($k = 1, 2, \dots$) is positive definite, namely, there exist constants $0 < \lambda_1 \leq \lambda_2$ satisfying

$$\lambda_1 \|y\|^2 \leq y^T \bar{H}_k y \leq \lambda_2 \|y\|^2, \quad \text{for all nonzero } y \in \mathfrak{N}^{|\Upsilon_k|}.$$

Assumption B The level set $\phi = \{x \in \mathfrak{N}^n \mid f^{\text{MY}}(x) \leq f^{\text{MY}}(x_0)\} \cap K$ is compact and f^{MY} is bounded from below.

Similar to the proof techniques of paper [36] on smooth box optimization, we can get the following lemma. So we only state it as follows but omit the proof.

Lemma 4.1 Assume that $d_k \neq 0$ be defined by (3.6)–(3.8) and $x_k \in \mathbf{B}$, we have

$$\min \left\{ 1, \frac{\|u - l\|_\infty}{\|d_k\|_\infty} \right\} \geq \beta^k \geq \min \left\{ 1, \frac{\chi_k}{\|d_k\|_\infty} \right\} \tag{4.1}$$

and

$$\|d_k\|^2 \leq -\varrho g_k^T d_k. \tag{4.2}$$

where $\varrho > 0$ is a constant, $\chi_k = \min\{|a_i(x_k)g_k^i|, |b_i(x_k)g_k^i|, i \in \Upsilon_k, g_k^i \neq 0\}$, $\beta^k = \sup_{0 \leq \omega \leq 1} \{\omega \mid l \leq x_k + \omega d_k \leq u\}$, and g_k^i is the i th element of $g(x_k)$. Moreover, x_k is a KKT point of (2.2) if and only if $d_k = 0$.

If x_k is not a KKT point, by Lemmas 4.1, it is easy to deduce that d_k is descent.

Theorem 4.1 *The sequence $\{x_k, d_k, \overline{H}_k\}$ is generated by Algorithm 1. Let Assumption A and B hold, and $\|Z^T \overline{H}_k Z\| \leq \zeta_1$ hold for all k with constant $\zeta_1 > 0$. Then every accumulation point of $\{x_k\}$ is a KKT point of (2.2).*

Proof We will prove this theorem by contradiction. Let x_* be any accumulation point of $\{x_i\}$, then for $i = 1, 2, \dots$, there exists a subsequence $\{x_{k_i}\}$ satisfying $\lim_{k \rightarrow \infty} x_{k_i} = x_*$. Suppose that x_* is not a KKT point, by (3.3) and (3.4), we can conclude that there exists $j \in \overline{T}$ or $j \in \overline{\Psi}$ such that

$$g(x_*)^j < 0 \quad \text{or} \quad g(x_*)^j > 0 \tag{4.3}$$

or $j \in \overline{\mathcal{T}}$ satisfying

$$g(x_*)^j \neq 0. \tag{4.4}$$

By the line search (3.9) and (4.2), we can deduce that the sequence $\{f^{\text{MY}}(x_k)\}$ is descent. Using f^{MY} is bounded from below in Assumption B, we have

$$\infty > \sum_{k=1}^{\infty} (f^{\text{MY}}(x_k) - f^{\text{MY}}(x_{k+1})) \geq \sum_{k=1}^{\infty} -\sigma \alpha_k g_k^T d_k, \tag{4.5}$$

this implies that

$$\sum_{k=1}^{\infty} -\alpha_k g_k^T d_k < \infty.$$

In particular, we get

$$\lim_{k \rightarrow \infty} -\alpha_k g_k^T d_k = 0. \tag{4.6}$$

Using the definition of d_k , setting $\vartheta_1 = \max_{x \in \mathbf{B}} \|g_k\|^2$, and

$$\nu_k = \sum_{i \in \Gamma_k} (a_i(x_k))^2 + \sum_{i \in \Psi_k} (b_i(x_k))^2,$$

we get

$$\begin{aligned} \|d_k\|^2 &= \|Z \overline{H}_k Z^T g_k\|^2 + \sum_{i \in \Gamma_k} (l_i - x_k^i)^2 + \sum_{i \in \Psi_k} (u_i - x_k^i)^2 \\ &\leq \zeta_1^2 \|g_k\|^2 + \sum_{i \in \Gamma_k} (a_i(x_k) g_k^i)^2 + \sum_{i \in \Psi_k} (b_i(x_k) g_k^i)^2 \\ &= (\zeta_1^2 + \nu_k) \|g_k\|^2 \\ &\leq (\zeta_1^2 + \nu_k) \theta_1. \end{aligned}$$

The above relation and (4.1) implies that there exists a constant $\beta^* \in (0, 1)$ such that

$$\beta^k \geq \beta^*, \quad \forall k. \tag{4.7}$$

Suppose that α_k satisfies the line search (3.9), if $\alpha_k < 0.1\beta^*$, then there exists an unacceptable steplength $\alpha_{k,i} \leq 10\alpha_k$ with $i \geq 0$ satisfying

$$f^{\text{MY}}(x_k) + \alpha_{k,i} g_k^T d_k + \zeta_2 \alpha_{k,i}^2 \|d_k\|^2 \geq f^{\text{MY}}(x_k + \alpha_k d_k) \geq f^{\text{MY}}(x_k) + \sigma \alpha_{k,i} g_k^T d_k,$$

where $\zeta_2 = \frac{1}{2} \max_{x \in B} \|\nabla^2 f^{\text{MY}}(x)\|$. Then we obtain

$$\alpha_{k,i} \geq \frac{-(1-\sigma)g_k^T d_k}{\zeta_2 \|d_k\|^2} \geq \frac{(1-\sigma)}{\zeta_2 \varrho},$$

where the second inequality follows from (4.2). Therefore, we get

$$\alpha_k \geq \min\{0.1\alpha_{k,i}, 0.1\beta^*\} = \min\left\{\frac{(1-\sigma)}{10\zeta_2\zeta}, 0.1\beta^*\right\} > 0, \quad \forall k.$$

This together with (4.6) implies that

$$\lim_{k \rightarrow \infty} g_k^T d_k = 0. \tag{4.8}$$

By the definition of Γ_k and Ψ_k , we can conclude that every term of the right side in the following relation:

$$-d_k^T g_k = g_k^T Z \bar{H}_k Z^T g_k - \sum_{i \in \Gamma_k} (l_i - x_k^i) g_k^i - \sum_{i \in \Psi_k} (u_i - x_k^i) g_k^i$$

is larger than zero. Thus, by (4.8), we get

$$\lim_{k \rightarrow \infty} \|Z^T g^k\| = 0, \quad \lim_{k \rightarrow \infty} \sum_{i \in \Gamma_k} (l_i - x_k^i) g_k^i = 0, \quad \lim_{k \rightarrow \infty} \sum_{i \in \Psi_k} (u_i - x_k^i) g_k^i = 0.$$

Therefore, for some $j \in \bar{\Upsilon}$ such that (4.4). By the above three relations, for all sufficiently large i , we get $j \notin \Gamma(x_{k_i}) \cup \Psi(x_{k_i}) \cup \Upsilon(x_{k_i})$, this is a contradiction. This completes the proof. □

Remark If the condition $g(\bar{x}) = 0$ holds, by (2.3) and the convexity of $f^{\text{MY}}(x)$, it is not difficult to get $\bar{x} = p(\bar{x})$. Accordingly the point \bar{x} is the unique optimal solution of (1.1).

5 Numerical results

In the experiments, all codes were written in *MATLAB r2010a* and run on a PC with CPU Intel(R) Core(TM) i3-3217U CPU 1.80 GHz, 4.00 G bytes of RAM memory, and Windows 7 operating system. Our experiments are performed on a set of the nonlinear box-constrained nonsmooth problems from Karmitsa [17] which have the given initial points; these problems are listed in Table 1. In the experiment, we choose $\sigma = \beta = 0.1$, $a_i(x) = b_i(x) = 10^{-5}$ in (3.5), $\theta = 1$ and the “basic matrix” to be the identity matrix I in the limited memory BFGS method, and $m = 5$. For the subproblem (2.1), we use the PRP conjugate gradient algorithm to solve it where the iteration number and the function number are added to the main-Algorithm program. The PRP conjugate gradient algorithm for (2.1) is listed as follows:

Table 1 Tested problems

No.	Name of problem	No.	Name of problem
1	Generalization of MAXQ	6	Number of active faces
2	Generalization of MXHILB	7	Generalization of Brown function 2
3	Chained LQ	8	Chained Mifflin 2
4	Chained CB3 I	9	Chained Crescent I
5	Chained CB3 II	10	Chained Crescent II

PRP algorithm for subproblem (2.1)

Step 1: Given z_0 , constants $\lambda > 0$, $\tau_0 > 0$, and $\iota \in (0, 1)$, $d_0^{\text{sub}} = -g_0^{\text{sub}} = -\frac{z_0 - x_0}{\lambda}$, set $k = 0$.

Step 2: If $\|g_k^{\text{sub}}\| < \iota$, stop.

Step 3: Find α_k^{sub} satisfying the following Armijo line search:

$$f^{\text{MY}}(z_k + \alpha_k^{\text{sub}} d_k^{\text{sub}}) - f^{\text{MY}}(z_0) \leq \alpha_k^{\text{sub}} \tau_0 (d_k^{\text{sub}})^T g_k^{\text{sub}},$$

where $\alpha_k^{\text{sub}} = \max\{1, \frac{1}{2}, \frac{1}{2^2}, \frac{1}{2^3}, \dots\}$ such that the above inequality.

Step 4: Let $z_{k+1} = z_k + \alpha_k^{\text{sub}} d_k^{\text{sub}}$ and $g_{k+1}^{\text{sub}} = \frac{z_{k+1} - x_0}{\lambda}$.

Step 5: If $\|g_{k+1}^{\text{sub}}\| < \iota$, stops, otherwise compute the search direction by

$$d_{k+1}^{\text{sub}} = -g_{k+1}^{\text{sub}} + \frac{(g_{k+1}^{\text{sub}} - g_k^{\text{sub}})^T g_{k+1}^{\text{sub}}}{\|g_k^{\text{sub}}\|}.$$

Step 6: Set $k = k + 1$ and go to Step 3.

In the above algorithm, x_0 follows Algorithm 1, $\iota = 1e-3$, and $\tau_0 = 0.25$. Since we aim to design the given algorithm to solve large-scale nonsmooth problems, the dimensions of the test problems are 5000, 7000, 10,000, and 11,000. The following *Himmeblau* stop rule is used: If $|f^{\text{MY}}(x_k)| > 1e-4$, let $stop1 = \frac{|f^{\text{MY}}(x_k) - f^{\text{MY}}(x_{k+1})|}{|f^{\text{MY}}(x_k)|}$; otherwise, let $stop1 = |f^{\text{MY}}(x_k) - f^{\text{MY}}(x_{k+1})|$. The program stops if $stop1 < 1e-4$ is satisfied. In the experiment, we find that the different stop rules will influence the iteration number and the function number obviously but for the final function value. Since the results of iteration number are stable, we choose the *Himmeblau* stop rule. In order to show the efficiency of the given method, we also test the normal Active-set algorithm with L-BFGS update and compare their performance. The columns of Table 2 has the following meaning:

dim: the dimension of the problem;

NI: the total number of iterations;

NF: the total number of the function value;

cpu-time: the CPU time in second;

$f(\bar{x})$: denotes the function value at the point \bar{x} when the program is stopped.

From the numerical results of Table 2, it is not difficult to see that both of these two methods are effective for solving these ten box-constrained nonsmooth problems. The iteration number and the function number do not change with the dimension increasing except for problems Generalization of MAXQ and Chained Crescent I, which shows that the given algorithm is feasible and stable. For problems Chained Crescent I and Chained Crescent II, since they have many similar properties, they have the same optimization value, respectively, with the determined dimensions. The cpu-time is interesting although it is becoming large with the dimension increasing.

Table 2 Numerical results

No.	dim	Algorithm 1			Active-set algorithm		
		NI/NF	$f(\bar{x})$	cpu-time	NI/NF	$f(\bar{x})$	cpu-time
1	5000	15/32	3.486539e-003	5.163633e+000	397/804	1.061325e-001	1.560166e+002
	7000	13/28	3.489925e-003	8.143252e+000	211/432	5.293420e-002	1.418673e+002
	10,000	13/28	3.491556e-003	1.987453e+001	581/1172	2.028926e-002	8.082568e+002
	11,000	13/28	3.491902e-003	4.258827e+001	161/332	5.865048e-002	5.825857e+002
2	5000	7/16	4.547255e-001	1.734731e+001	7/16	4.547255e-001	1.712891e+001
	7000	7/16	4.715477e-001	3.138740e+001	7/16	4.715477e-001	3.159020e+001
	10,000	7/16	4.893804e-001	6.361721e+001	7/16	4.893804e-001	6.094959e+001
	11,000	7/16	4.941456e-001	8.386614e+001	7/16	4.941456e-001	8.375694e+001
3	5000	6/28	-4.034727e+003	1.731611e+000	16/58	-5.766614e+003	2.558416e+000
	7000	6/28	-5.648940e+003	2.137214e+000	16/58	-8.073721e+003	4.258827e+000
	10,000	6/28	-8.070261e+003	4.196427e+000	16/58	-1.153438e+004	7.488048e+000
	11,000	6/28	-8.877367e+003	8.190052e+000	16/58	-1.268794e+004	1.755011e+001
4	5000	23/63	9.998007e+003	5.257234e+000	23/63	9.998007e+003	4.929632e+000
	7000	23/63	1.399801e+004	6.630042e+000	23/63	1.399801e+004	6.520842e+000
	10,000	23/63	1.999801e+004	9.921664e+000	23/63	1.999801e+004	9.406860e+000
	11,000	23/63	2.199802e+004	1.096687e+001	23/63	2.199802e+004	1.063927e+001
5	5000	23/63	9.998007e+003	1.388409e+000	23/63	9.998007e+003	9.828063e-001
	7000	23/63	1.399801e+004	2.199614e+000	23/63	1.399801e+004	1.887612e+000
	10,000	23/63	1.999801e+004	3.759624e+000	23/63	1.999801e+004	3.198020e+000
	11,000	23/63	2.199802e+004	4.539629e+000	23/63	2.199802e+004	3.978025e+000
6	5000	7/16	5.525453e+000	2.979619e+000	7/16	5.525453e+000	2.792418e+000
	7000	7/16	5.860786e+000	4.960832e+000	7/16	5.860786e+000	4.898431e+000
	10,000	7/16	6.216606e+000	1.113847e+001	7/16	6.216606e+000	8.767256e+000
	11,000	7/16	6.311735e+000	1.753451e+001	7/16	6.311735e+000	2.138774e+001
7	5000	9/20	4.961701e+002	3.603623e+000	11/24	4.961701e+002	4.446029e+000
	7000	9/20	6.946779e+002	5.350834e+000	11/24	6.946779e+002	6.583242e+000
	10,000	9/20	9.924395e+002	9.578461e+000	11/24	9.924395e+002	1.235528e+001
	11,000	9/20	1.091693e+003	2.368095e+001	11/24	1.091693e+003	3.383662e+001
8	5000	5/12	-2.859465e+003	2.028013e+000	29/80	-2.983598e+003	1.918812e+000
	7000	5/12	-4.003716e+003	2.496016e+000	29/80	-4.177659e+003	2.386815e+000
	10,000	5/12	-5.720094e+003	4.586429e+000	29/80	-5.968752e+003	4.758031e+000
	11,000	5/12	-6.292219e+003	8.580055e+000	29/80	-6.565783e+003	1.244888e+001
9	5000	9/20	7.248551e+002	3.478822e+000	11/24	7.248551e+002	5.007632e+000
	7000	9/20	1.014855e+003	5.272834e+000	11/24	1.014855e+003	6.536442e+000
	10,000	9/20	1.449855e+003	9.204059e+000	11/24	1.449855e+003	1.343169e+001
	11,000	23/48	1.594855e+003	2.447656e+001	25/52	1.594855e+003	2.750298e+001
10	5000	9/20	7.248551e+002	3.400822e+000	11/24	7.248551e+002	4.383628e+000
	7000	9/20	1.014855e+003	4.992032e+000	11/24	1.014855e+003	6.520842e+000
	10,000	9/20	1.449855e+003	9.656462e+000	11/24	1.449855e+003	1.171568e+001
	11,000	9/20	1.594855e+003	2.772138e+001	11/24	1.594855e+003	2.436736e+001

To directly show the performance of these two algorithms, the tool of Dolan and Moré [9] is used to analyze them. Figure 1 shows that the performance of Algorithm 1 and Active-set algorithm is relative to cpu-time of Table 2. It is not difficult to see that Algorithm 1 has won, since it has the higher probability of being the optimal solver. Figure 1 shows that Algorithm 1 can successfully solve 100% of the test problems at $t \approx 2$ and Active-set algorithm completely solves the test problems at about $t \approx 41$. All in all, although the proposed method does not obtain significant development as we have expected, we think that the enhancement of this proposed method is still noticeable.

6 Conclusions

(i) It is well known that the nonsmooth problems are very difficult to solve even when the objective function is unconstrained, especially for large-scale nonsmooth problems.

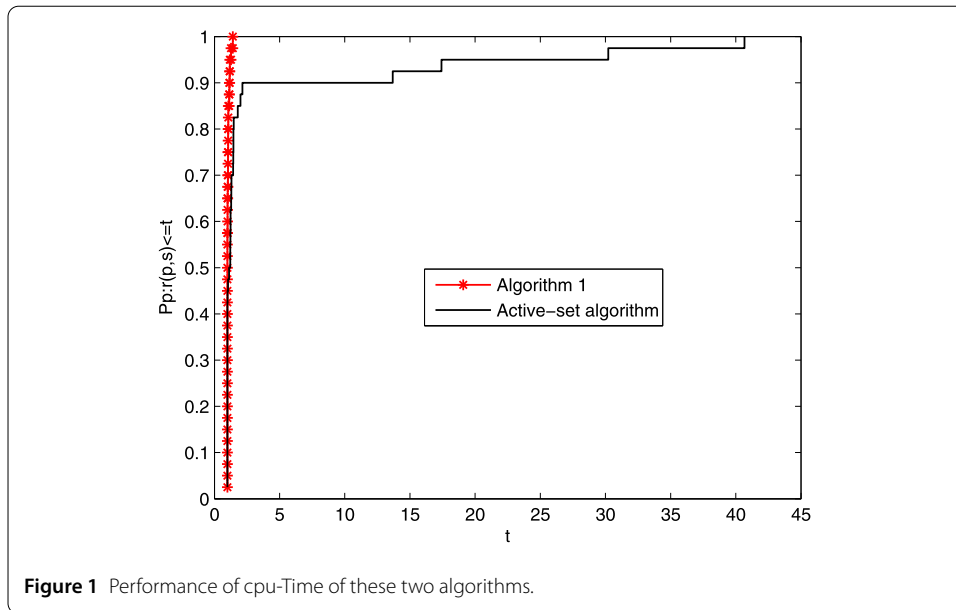


Figure 1 Performance of cpu-Time of these two algorithms.

The Moreau–Yosida regularization technique is an effective tool to deal with this problem. Then we use this technique and propose a subspace algorithm for solving box-constrained nonsmooth optimization problems.

(ii) In order to decrease the workload of the computer and get good numerical performance, the limited memory BFGS method is utilizable in the given algorithm. The numerical performance of the test problems show that the presented algorithm is very interesting for large-scale problems. The dimension is from 5000 to 11,000 variables, which are larger than those of the unconstrained nonsmooth problems of [18].

(iii) In the experiment, we find the different stop rules will obviously influence the iteration numbers and the function numbers but for the final functions. The reason lies in the stop rule, then further work in the future is to find more correct stop rules of the given algorithms.

(iv) Inspired by the idea of [26] and [10], we extend their techniques to nonsmooth problems. The proof methods of this paper are similar to [36]. However, all of these three papers concentrate on the continuous differentiable optimization problems.

(v) Considering the above discussions, we think there are at least three issues that could lead to improvements. The first point is the constant m in the L-BFGS update formula that could be adjusted. Another important point that should be considered is probably the choice of the parameters in the active-set identification technique, since the value of the used parameters is not the only choice. The last one is the most important one, which is from the numerical performance, namely whether are there better stop rules, other optimality conditions and convergence conditions in the nonsmooth problems? All of these aspects are for further work in the future.

Overall, we think that the method provide a valid approach for solving large-scale box-constrained nonsmooth problems, since the numerical performance is interesting.

Acknowledgements

The author would like to thank the editor for his/(or her) work and the referees for their valuable comments, which greatly improved our paper.

Funding

This work is supported by the High Level Innovation Teams and Excellent Scholars Program in Guangxi institutions of higher education (Grant No. [2019]52), the National Natural Science Foundation of China (Grant No. 11661009), the Guangxi Natural Science Key Fund (No. 2017GXNSFDA198046), and the National Social Science Key Foundation of China (Grant No. 17AJL012).

Availability of data and materials

All data are included in this paper and they be freely used.

Competing interests

There is no potential conflicts of interest.

Authors' contributions

The main idea of this paper was proposed by XL. XL prepared the manuscript initially and performed all the steps of the proofs in this research. The author read and approved the final manuscript.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 14 October 2019 Accepted: 24 April 2020 Published online: 11 May 2020

References

1. Ben-tal, A., Nemirovski, A.: Non-Euclidean restricted memory level method for large-scale convex optimization. *Math. Program.* **3**, 407–456 (2005)
2. Birge, J.R., Qi, L., Wei, Z.: A general approach to convergence properties of some methods for nonsmooth convex optimization. *Appl. Math. Optim.* **38**, 141–158 (1998)
3. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: A family of variable metric proximal methods. *Math. Program.* **68**, 15–47 (1995)
4. Byrd, R.H., Lu, P.H., Nocedal, J.: A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Stat. Comput.* **16**, 1190–1208 (1995)
5. Byrd, R.H., Nocedal, J., Schnabel, R.B.: Representations of quasi-Newton matrices and their use in limited memory methods. *Math. Program.* **63**, 129–156 (1994)
6. Calamai, P., Moré, J.J.: Projected gradient for linearly constrained programs. *Math. Program.* **39**, 93–116 (1987)
7. Clarke, F.H.: *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York (1983)
8. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. *Math. Program.* **62**, 261–273 (1993)
9. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)
10. Facchinei, F., Júdice, J., Soares, J.: An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM J. Optim.* **8**, 158–186 (1998)
11. Fletcher, R.: *Practical Methods of Optimization*, 2nd edn. Wiley, Chichester (1987)
12. Fukushima, M., Qi, L.: A global and superlinearly convergent algorithm for nonsmooth convex minimization. *SIAM J. Optim.* **6**, 1106–1120 (1996)
13. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1998)
14. Haara, M., Mäkelä, M.M.: Limited memory bundle algorithm for large bound constrained nonsmooth minimization problems. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, No. B. 1/2006, University of Jyväskylä, Finland (2006)
15. Haara, M., Miettinen, K., Mäkelä, M.M.: New limited memory bundle method for large-scale nonsmooth optimization. *Optim. Methods Softw.* **19**, 673–692 (2004)
16. Hiriart-Urruty, J.B., Lemaréchal, C.: *Convex Analysis and Minimization Algorithms II*. Springer, Berlin (1983)
17. Karmitsa, N.: Test problems for large-scale nonsmooth minimization. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, No. B. 4/2007, University of Jyväskylä, Finland (2007)
18. Karmitsa, N., Bagirov, A., Mäkelä, M.M.: Comparing different nonsmooth minimization methods and software. *Optim. Methods Softw.* **27**, 131–153 (2012)
19. Kiwiel, K.C.: *Methods of Descent for Nondifferentiable Optimization*. Lecture Notes in Mathematics, vol. 1133. Springer, Berlin (1985)
20. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable optimization. *Math. Program.* **46**, 105–122 (1990)
21. Kiwiel, K.C.: Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Math. Program.* **69**, 89–109 (1995)
22. Lemaréchal, C.: Extensions diverses des méthodes de gradient et applications, Thèse d'Etat. Paris (1980)
23. Lemaréchal, C.: Nondifferentiable optimization. In: Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (eds.) *Optimization*, pp. 529–572. Elsevier, New York (1989)
24. Majava, K., Haara, M., Kärkkäinen, T.: Solving variational image denoising problems using limited memory bundle method. In: Proceedings of the 2nd International Conference on Scientific Computing and Partial Differential Equations and the First East Asia SIAM Symposium, Hongkong (2005)
25. Nguyen, V.H., Strodick, J.J.: A linearly constrained algorithm not requiring derivative continuity. *Eng. Struct.* **6**, 7–11 (1984)
26. Ni, Q.: A subspace projected conjugate algorithm for large bound constrained quadratic programming. *Numer. Math. J. Chin. Univ.* **7**, 51–60 (1998)

27. Ni, Q., Yuan, Y.X.: A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization. *Math. Comput.* **66**, 1509–1520 (1997)
28. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: *Numerical Analysis*, pp. 155–157 (1978)
29. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.* **18**, 227–245 (1993)
30. Qi, L., Sun, J.: A nonsmooth version of Newton's method. *Math. Program.* **58**, 353–367 (1993)
31. Schramm, H.: Eine kombination von bundle- und trust-region-verfahren zur Lösung nichtdifferenzierbare optimierungsprobleme. *Bayreuther Mathematische Schriften*, vol. 30 (1989)
32. Schramm, H., Zowe, J.: A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim.* **2**, 121–152 (1992)
33. Schultz, H.K.: A Kuhn–Tucker algorithm. *SIAM J. Control Optim.* **11**, 438–445 (1973)
34. Sreedharan, V.P.: A subgradient projection algorithm. *J. Approx. Theory* **35**, 111–126 (1982)
35. Wolfe, P.: A method of conjugate subgradients for minimizing nondifferentiable convex functions. *Math. Program. Stud.* **3**, 145–173 (1975)
36. Xiao, Y., Wei, Z.: A new subspace limited memory BFGS algorithm for large-scale bound constrained optimization. *Appl. Math. Comput.* **185**, 350–359 (2007)
37. Yuan, G., Li, P., Lu, J.: The global convergence of the BFGS method with a modified WWP line search for nonconvex functions. *Numer. Algorithms* (2019, accepted)
38. Yuan, G., Li, T., Hu, W.: A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems. *Appl. Numer. Math.* **147**, 129–141 (2020)
39. Yuan, G., Meng, Z., Li, Y.: A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. *J. Optim. Theory Appl.* **168**, 129–152 (2016)
40. Yuan, G., Sheng, Z.: *Optimization Algorithm of Nonsmooth*. Science Press, Beijing (2017)
41. Yuan, G., Sheng, Z., Wang, P., Hu, W., Li, C.: The global convergence of a modified BFGS method for nonconvex functions. *J. Comput. Appl. Math.* **327**, 274–294 (2018)
42. Yuan, G., Wei, Z.: The Barzilai and Borwein gradient method with nonmonotone line search for nonsmooth convex optimization problems. *Math. Model. Anal.* **17**, 203–216 (2012)
43. Yuan, G., Wei, Z., Li, G.: A modified Polak–Ribière–Polyak conjugate gradient algorithm with nonmonotone line search for nonsmooth convex minimization. *J. Comput. Appl. Math.* **255**, 86–96 (2014)
44. Yuan, G., Wei, Z., Lu, X.: Global convergence of the BFGS method and the PRP method for general functions under a modified weak Wolfe–Powell line search. *Appl. Math. Model.* **47**, 811–825 (2017)
45. Yuan, G., Wei, Z., Wang, Z.: Gradient trust region algorithm with limited memory BFGS update for nonsmooth convex minimization. *Comput. Optim. Appl.* **54**, 45–64 (2013)
46. Yuan, G., Wei, Z., Yang, Y.: The global convergence of the Polak–Ribière–Polyak conjugate gradient algorithm under inexact line search for nonconvex functions. *J. Comput. Appl. Math.* **362**, 262–275 (2019)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
