A Subspace Limited Memory BFGS Algorithm For Box Constrained Optimization*

ZENGXIN WEI† YANLIN WU† GONGLIN YUAN†

Abstract. In this paper, a subspace limited BFGS algorithm is proposed for bound constrained optimization. The global convergence will be established under some suitable conditions. Numerical results show that this method is more competitive than the normal method does.

Key Words. box constrained optimization; limited memory BFGS method; global convergence.

AMS subject classifications (2000). 65H10, 90C26.

1. Introduction

We consider the following nonlinear programming problem

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

where $f(x) : \Re^n \to \Re$ is a nonlinear function whose gradient $\nabla f(x)$ is available, and denote $\nabla f(x^k)$, or simply g^k, the vectors l and u represent lower and upper bounds on the variables, respectively, n is the number of variables, which is assumed to be large. For this problem, early methods tended to be of the active set variety [12, 18], which are quite efficient for problems of relatively small dimension but are unattractive for large-scale problems. The main reason is that typically at most one constraint can be added to or dropped from the active set at each iteration, and the potential worst-case complexity where each of the possible 3^n active set is visited before discovering the optimal one. Many authors have alluded to design active set methods that are capable of making rapid changes to incorrect predictions [1, 10, 11, 19, 20]. Now more common method is the gradient projection methods.

The gradient project method [1] is a constructive method, which bending the search direction along the constraint boundary to add to or drop from the current estimated active set many constraints at each iteration, and yet find the active set in a finite number of steps. This work has motivated further studies on projection techniques both for the general linearly constrained case and for the bound constrained case [2, 6, 7, 14, 17, 19], where Conn et al. [7] considered extending the trust region concepts and algorithms to the constrained minimization case and Lescrenier [14] established the superlinear convergence rate without requiring strict complementary condition under suitable conditions.

*This work is supported by China NSF grands 10761001.

†Department of Mathematics and Information Science, Guangxi University, Nanning, Guangxi, P.R. China, E-mail addresses: zxwei@gxu.edu.cn, wyl-kk@163.com, glyuan@gxu.edu.cn.

Xiao and Wei [25] presented 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 provides an efficient means for attacking large-scale optimization problems. Motivated by their ideas, we will make a further study.

For unconstrained optimization problems, there are many methods [26, 27, 28, 29] for them, where the BFGS method is one of the most effective quasi-Newton method. The normal BFGS update exploits only the gradient information, while the information of function values available is neglected. These years, lots of modified BFGS methods (see [9, 15, 16, 21, 24, 31]) have been proposed. Especially, many efficient attempts have been made to modify the usual quasi-Newton methods using both the gradient and function values information (e.g. [23, 31]). Lately, in order to get a higher order accuracy in approximating the second curvature of the objective function, Wei, Yu, Yuan, and Lian [24] proposed a new BFGS-type method, and the reported numerical results show that the average performance is better than that of the standard BFGS method. The superlinear convergence of this modified has been established for uniformly convex function. Its global convergence is established by Wei, Li, and Qi [23]. Motivated by their ideas, Yuan and Wei [30] presented a modified BFGS method which can ensure that the update matrix are positive definite for the general convex functions. Moreover, the global convergence is proved for the general convex functions. The limited memory BFGS (L-BFGS) method (see [4]) is an adaptation of the BFGS method for large-scale problems. The implementation is almost identical to that of the standard BFGS method, the only difference is that the inverse Hessian approximation is not formed explicitly, but defined by a small number of BFGS updates. It is often provided a fast rate of linear convergence, and requires minimal storage.

Inspired by the modified method of [30], we combine this technique and the limited memory technique, and give a limited memory BFGS method for bound constrained optimization. The main contributions of this paper are as follows.

- A limited memory BFGS method is introduced, which possesses not only the gradient information of (1.1) but also the information of function values, moreover the update matrix can be positive definite for generally convex functions.
- All iterates are feasible and the sequence of objective functions is decreasing.
- Rapid changes in the active set are allowed.
- The global convergence of the new method is established.
- Numerical results show that this method is effective.

This paper is organized as follows. In the next section, we briefly review some modified method and the L-BFGS method for unconstrained optimization. In Section 3, we describe the modified limited memory BFGS algorithm for (1.1). The global convergence will be established in Section 4. Numerical results are reported in Section 5.

Notation. Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm of vectors or matrix.

The number of variables in the optimization problem is n, and the number of correction pairs used in the limited memory methods is m. The hessian inverse approximation is denoted by H^k. The set of the free variables at iteration k is F^k and $\bar{H}^k$ denotes the reduced matrix of H^k with the index of rows and columns are all in F^k, the $\mid F^k \mid$ denotes the length of the set F^k. $\nabla f_i(x)$ denotes the ith component of vector $\nabla f(x)$, i.e., $\nabla f_i(x) = \frac{\partial f(x)}{\partial x_i}$.

2. Modified BFGS Formula and L-BFGS Update

This section will state some modified BFGS formula and L-BFGS formula for unconstrained optimization problems, respectively.

2.1. Modified BFGS Formula

Quasi-Newton methods for solving UNP often need update the iterate matrix B^k. In tradition, $\{B^k\}$ satisfies the following quasi-Newton equation:

$$B^{k+1}s^k = y^k, \tag{2.1}$$

where $s^k = x^{k+1} - x^k, y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$. The very famous update B^k is the BFGS formula

$$B^{k+1} = B^k - \frac{B^k s^k (s^k)^T B^k}{(s^k)^T B^k s^k} + \frac{y^k (y^k)^T}{(s^k)^T y^k}. \tag{2.2}$$

Let H^k be the inverse of B^k, then the inverse update formula of (2.2) method is represented as

$$\begin{aligned} H^{k+1} &= H^k - \frac{(y^k)^T(s^k - H^k y^k)s^k(s^k)^T}{((y^k)^T s^k)^2} + \frac{(s^k - H^k y^k)(s^k)^T + s^k(s^k - H^k y^k)^T}{((y^k)^T s^k)^2} \\ &= (I - \frac{s^k (y^k)^T}{(y^k)^T s^k}) H^k (I - \frac{y^k (s^k)^T}{(y^k)^T s^k}) + \frac{s^k (s^k)^T}{(y^k)^T s^k}, \end{aligned} \tag{2.3}$$

which is the dual form of the DFP update formula in the sense that $H^k \leftrightarrow B^k$, $H^{k+1} \leftrightarrow B^{k+1}$, and $s^k \leftrightarrow y^k$. In order to obtain a global convergence of BFGS method without convexity assumption on the objective function, Li and Fukushima [15, 16] made a slight modification to the standard BFGS method. Now we state their work [15] simply. Li and Fukushima (see [15]) advised a new quasi-Newton equation with the following form

$$B^{k+1}s^k = (y^k)^{1*},$$

where $(y^k)^{1*} = y^k + t^k \|g^k\| s^k$, $t^k > 0$ is determined by $t^k = 1 + \max\{-\frac{(s^k)^T y^k}{\|s^k\|^2}, 0\}$. Under appropriate conditions, these two methods [15, 16] are globally and superlinearly convergent for nonconvex minimization problems.

In order to get a better approximation of the objective function Hessian matrix, Wei, Yu, Yuan, and Lian (see [24]) also proposed a new quasi-Newton equation:

$$B^{k+1}(2)s^k = (y^k)^{2*} = y^k + A^k(3)s^k,$$

where $A^k(3) = \frac{2[f(x^k)-f(x^k+\alpha^k d^k)]+(\nabla f(x^k+\alpha^k d^k)+\nabla f(x^k))^T s^k}{\|s^k\|^2}$. Then the new BFGS update formula is

$$B^{k+1}(2) = B^k(2) - \frac{B^k(2)s^k(s^k)^T B^k(2)}{(s^k)^T B^k(2)s^k} + \frac{(y^k)^{2*}(y^k)^{2*T}}{(s^k)^T(y^k)^{2*}}. \tag{2.4}$$

Note that this quasi-Newton formula (3.5) contains both gradient and function value information at the current and the previous step. This modified BFGS update formula differs from the standard BFGS update, and a higher order approximation of $\nabla^2 f(x)$ can be obtained (see [23, 24]).

In order to get the positive definiteness of $B^k(2)$ based on the definition of $(y^k)^{2*}$ and y^k for the general convex functions, Yuan and Wei [30] give a modified BFGS update, i.e., the modified update formula is defined by

$$B^{k+1}(3) = B^k(3) - \frac{B^k(3)s^k(s^k)^T B^k(3)}{(s^k)^T B^k(3)s^k} + \frac{(y^k)^{3*}(y^k)^{3*T}}{(y^k)^{3*T} s^k}, \tag{2.5}$$

where $(y^k)^{3*} = y^k + A^k s^k$, $A^k = \max\{A^k(3), 0\}$. Then the corresponding quasi-Newton equation is

$$B^{k+1}(3)s^k = (y^k)^{3*}, \tag{2.6}$$

which can ensure that the condition $(s^k)^T(y^k)^{3*} > 0$ holds for the general convex function f (we will state this idea in the following section). Therefore, the update matrix B^{k+1} from (2.6) inherits the positive definiteness of B^k for the general convex function.

2.2. L-BFGS Update

The limited memory BFGS method (see [4, 5]) is an adaptation of the BFGS method to large-scale problems, the only difference is in the matrix update, for getting Hessian inverse approximate H^{k+1}, instead of storing the matrices H^k, at every iteration x^k the method stores a small number, say m, of correction pairs $\{s^i, y^i\}, i = k-1, ..., k-m$, where

$$s^k = x^{k+1} - x^k, \quad y^k = g^{k+1} - g^k.$$

The standard BFGS correction with H^k have the following form:

$$H^{k+1} = (V^k)^T H^k V^k + \rho^k s^k (s^k)^T,$$

where $\rho^k = \frac{1}{(y^k)^T s^k}$ and $V^k = I - \rho^k y^k (s^k)^T$. If we use the stored correction pairs, we get

$$H^{k+1} \quad = \quad (V^k)^T[(V^{k-1})^T H^{k-1} V^{k-1} + \rho^{k-1} s^{k-1} (s^{k-1})^T] V^k + \rho^k s^k (s^k)^T$$

$$
\begin{aligned}
&= (V^k)^T(V^{k-1})^T H^{k-1}V^{k-1} + (V^k)^T\rho^{k-1}s^{k-1}(s^{k-1})^T V^k + \rho^k s^k (s^k)^T \\
&= \cdots \\
&= [(V^k)^T \cdots (V^{k-m+1})^T]H^{k-m+1}[V^{k-m+1}\cdots V^k] + \rho^{k+m-1}[(V^{k-1})^T \\
\cdots\ & (V^{k-m+2})^T]s^{k-m+1}(s^{k-m+1})^T[V^{k-m+2}...V^{k-1}] + \cdots + \rho^k s^k (s^k)^T. \qquad (2.7)
\end{aligned}
$$

These correction pairs contain information about the curvature of the function and, in conjunction with the BFGS formula, define the limited memory iteration matrix.

To maintain the positive definiteness of the limited memory BFGS matrix, some researches discard a correction pair $[s^k, y^k]$ if the curvature $(s^k)^T y^k > 0$ is not satisfied (see [5]). Another approach was proposed by Powell [19, 20], they replace s^k with a new $(s^k)'$ by means of some relations.

3. Modified Algorithm

In order to ensure that the condition $(s^k)^T y^k > 0$ of (2.7) holds for generally convex functions, we discuss $A^k(3)$ in the following two cases to show this motivation [30].

case i: If $A^k(3) > 0$, we have

$$
(s^k)^T(y^k + \frac{A^k(3)}{\|s^k\|^2}s^k) = (s^k)^T y^k + A^k(3) > (s^k)^T y^k \geq 0. \qquad (3.1)
$$

case ii: If $\rho^k < 0$, we get

$$
\begin{aligned}
0 &> A^k(3) = 2[f(x^k) - f(x^k + \alpha^k d^k)] + (g(x^k + \alpha^k d^k) + g(x^k))^T s^k \\
&\geq -2(g^{k+1})^T s^k + (g(x^k + \alpha^k d^k) + g(x^k))^T s^k \\
&= -(s^k)^T y^k, \qquad (3.2)
\end{aligned}
$$

which means that $(s^k)^T y^k > 0$ holds. Therefore, we present our modified L-BFGS formula as follows

$$
\begin{aligned}
H^{k+1} &= (V_*^k)^T[(V_*^{k-1})^T H^{k-1}V_*^{k-1} + \rho_*^{k-1}s^{k-1}(s^{k-1})^T]V_*^k + \rho_*^k s^k (s^k)^T \\
&= (V_*^k)^T(V_*^{k-1})^T H^{k-1}V_*^{k-1} + (V_*^k)^T\rho_*^{k-1}s^{k-1}(s^{k-1})^T V_*^k + \rho_*^k s^k (s^k)^T \\
&= \cdots \\
&= [(V_*^k)^T \cdots (V_*^{k-m+1})^T]H^{k-m+1}[V_*^{k-m+1}\cdots V_*^k] + \rho_*^{k+m-1}[(V_*^{k-1})^T \\
\cdots\ & (V_*^{k-m+2})^T]s^{k-m+1}(s^{k-m+1})^T[V_*^{k-m+2}...V_*^{k-1}] + \cdots + \rho_*^k s^k (s^k)^T, \qquad (3.3)
\end{aligned}
$$

where $\rho_*^k = \frac{1}{(y_*^k)^T s^k}$, $V_*^k = I - \rho_*^k y_*^k (s^k)^T$ and $y_*^k = y^k + \max\{A^k(3), 0\}$. If $A^k(3) > 0$ holds, it is not difficult to see that the modified L-BFGS formula (3.3) contains both the gradient and function value information at the current and the previous step. In the following, the matrix H^k is generated by (3.3).

Now we show how to define at each feasible point x^k a search direction d^k which can be used in connection with the projected search. We first discuss the determination of search directions based on guessing technique in [10]. We set the feasible region $K = \{x \in \Re^n : l_i \leq x_i \leq u_i, i = 1, ..., n\}$, a vector $\overline{x} \in K$ is said to be a stationary point for problem (1.1) if it satisfies

$$\begin{cases} l_i = \overline{x}_i & \Rightarrow \quad \nabla f_i(\overline{x}) \geq 0, \\ l_i < \overline{x}_i < u_i & \Rightarrow \quad \nabla f_i(\overline{x}) = 0, \\ \overline{x}_i = u_i & \Rightarrow \quad \nabla f_i(\overline{x}) \leq 0. \end{cases} \tag{3.4}$$

Strict complementarity is said to hold at $\overline{x}$ if the strict inequality hold in the first and the third implication of (3.4). In order to introduce the procedure that estimate the active bounds,let $\overline{x} \in \Re^n$ be a stationary point of problem (1.1), and consider the associated active constraint set

$$\overline{L} = \{i : l_i = \overline{x}_i\}, \ \overline{U} = \{i : \overline{x}_i = u_i\} \tag{3.5}$$

Moreover the set of the free variables defined by

$$\overline{F} = \{1, ..., n\} \backslash (\overline{L} \cup \overline{U}).$$

Namely, instead of the condition (3.4), we get

$$\begin{cases} \nabla f_i(\overline{x}) \geq 0 & \forall i \in \overline{L}, \\ \nabla f_i(\overline{x}) = 0 & \forall i \in \overline{F}, \\ \nabla f_i(\overline{x}) \leq 0 & \forall i \in \overline{U}. \end{cases} \tag{3.6}$$

Then it seems fairly natural to define the following approximation $L(x), F(x)$ and$U(x)$ to $\overline{L}, \overline{F}$ and$\overline{U}$ respectively:

$$\begin{aligned} L(x) &= \{i : x_i \leq l_i + a_i(x) \nabla f_i(x)\}, \\ U(x) &= \{i : x_i \geq u_i + b_i(x) \nabla f_i(x)\}, \\ F(x) &= \{1, ..., n\} \backslash (L \cup U), \end{aligned} \tag{3.7}$$

where a_i and b_i are nonnegative continuous bounded from above on K, such that if $x_i = l_i$ or $x_i = u_i$ then $a_i(x) > 0$ or $b_i(x) > 0$, respectively. Other identified technique can be consult to [2, 20]. The following results shows that $L(x), F(x)$ and $U(x)$ are indeed "good" estimate of $\overline{L}, \overline{F}$ and $\overline{U}$ respectively.

Theorem 3.1 *[Theorem 3 in [10]] For any feasible x, $L(x) \cap U(x) = \emptyset$. Furthermore, if $\overline{x}$ is a stationary point of problem (1.1) where strict complementarity holds, then there exists a neighborhood of $\overline{x}$ such that for every feasible point x in this neighborhood we have*

$$L(x) = \overline{L}, \quad F(x) = \overline{F}, \quad U(x) = \overline{U}.$$

Let $x^k \in k$ be the current point at iteration k. Consider the sets $L^k = L(x^k), F^k = F(x^k)$ and $U^k = U(x^k)$, the subspace direction $d^k_{F^k}$ is chosen as the search direction for the inactive variables. Let Z be the matrix whose columns are $\{e_i | i \in F^k\}$, where e_i is the ith column of the identity matrix in $\Re^{n\times n}$, and H^k be an approximation of the full space inverse Hessian matrix. Let $\overline{H}^k \in \Re^{|F^k|\times|F^k|}$ be an approximation of the reduced inverse Hessian matrix, then $\overline{H}^k = Z^T H^k Z$. The search direction $d^k = (d^k_{L^k}, d^k_{F^k}, d^k_{U^k})$ chosen as

$$d_i^k = l_i - x_i^k, \qquad i \in L^k; \tag{3.8}$$

$$d_i^k = u_i - x_i^k, \qquad i \in U^k; \tag{3.9}$$

$$d_i^k = -(Z\overline{H}^k Z^T g^k), \qquad i \in F^k. \tag{3.10}$$

The projected search has been used by several authors for solving quadratic and nonlinear programming problems with bound bounds on the variables (see [1, 3, 20]). The projected search requires that a steplength, $\alpha_k > 0$, which produces a sufficient decrease in the function $\phi^k : \Re \to \Re$ defined by

$$\phi^k(\alpha) = f([x^k + \alpha d^k]^+)$$

where $[\cdot]^+$ is the projection into K defined by

$$[x]^+ = \begin{cases} x_i & if \;\; l_i \le x_i \le u_i, \\ l_i & if \;\; x_i < l_i, \\ u_i & if \;\; x_i > u_i. \end{cases} \tag{3.11}$$

The sufficient decrease condition requires that $\alpha_k > 0$ satisfy

$$\phi^k(\alpha) \le \phi^k(0) + \sigma \nabla \phi^k(0)\alpha, \tag{3.12}$$

where $\sigma \in (0, \frac{1}{2})$.

Lemma 3.1 *If H^k is positive definite, then d^k defined by (3.8)-(3.10) satisfies*

$$(d^k)^T g^k \le 0 \tag{3.13}$$

Proof. By the definition of search direction d^k, (3.8)-(3.10), we get

$$(d^k)^T g^k = \sum_{i\in L^k} (l_i - x_i^k) g_i^k + \sum_{i \in U^k} (u_i - x_i^k) g_i^k + \sum_{i\in F^k} -g_i^k (Z\overline{H}^k Z^T g^k)_i \le 0$$

The above relation follows the positive definite of H^k (so $\overline{H}^k$) and the definition of the active set (3.7), which also indicate $(d^k)^T g^k = 0$ if and only if $d^k = 0$. □

Lemma 3.1 shows that whenever $d^k \neq 0$, it is at least a descent direction for objective function $f(x)$ at current point x^k, the property is very important to establish our global convergence.

If the limited memory update is used in the bounded constrained optimization problems, the set of active constraints should be changes at first finite steps. One approach is to store $\{s^k, y_*^k\}_{k-m+1}^k$, and update it as a full matrix, and reduced in the free subspace. This is very costly for even moderately large problems. Another approach of updating H^k is to use only reduced gradient and projected steps, but too much information may be cost.

In our numerical experiments, we used the approach based on the recursive BFGS update that does that discard information corresponding to that part of the inactive set that is not changed. At each iteration, we stores the sequence $\{\overline{s}^k\}$ and $\{\overline{y}_*^k\}$ according to the reduced gradient and projected steps.

Algorithm 1 Update (ns, $\{\overline{s}^k\}$,$\{\overline{y}_*^k\}$,H^0,d,Z)

Step 1: $d = Z'd$;

Step 2: if $ns = 0, d = H^0 d$;return;

Step 3: $\alpha = \overline{s}_{ns-1}^T d / \overline{y}_{ns-1}^T \overline{s}_{ns-1}^k; d = d - \alpha \overline{y}_{ns-1}^k$;

Step 4: call Update ($ns - 1, \{\overline{s}^k\}$,$\{\overline{y}_*^k\}$,$H^0$,$d$,$Z$);

Step 5: $d = d + (\alpha - (d^T \overline{y}_{ns-1}^k / \overline{y}_{ns-1}^{kT} \overline{s}_{ns-1}^k)) \overline{s}_{n-1}^k$;

Step 6: $d = Z'd$.

where $ns \leq m$ is the number of the correction pairs. Note that we reinitialize to zero, when $(\overline{y}_*^k)^T \overline{s}^k \leq 0$. Now we state the algorithm for solving the bound constrained optimization problems (1.1) and call it projected active set limited memory BFGS (PAS-L-BFGS) algorithm.

Algorithm 2 (PAS-L-BFGS Algorithm)

Step 1: Given starting point $x^0 \in K$,constant $\sigma \in (0, \frac{1}{2})$ and $m \in (3, 20)$, the "basic matrix" θI, nonnegative continuous $a_i(x)$ and $b_i(x)$; compute $f(x^0)$, $\nabla f(x^0)$ and set $k = 0$.

Step 2: Initialize. Determine $L^k = L(x^k), U^k = U(x^k)$, and $F^k = F(x^k)$ according to (3.7).

Step 3: Determine the search direction. Compute d^k from (3.8)-(3.10).

Step 4: Stopping test. If $d^k = 0$,stop; otherwise ,continue.

Step 5: Backtracking line search . Using the projected line search rule which find α_k satisfy (3.12).

Step 6: Accept the new point. Set $x^{k+1} = [x^k + \alpha_k d^k]^+$. Compute $f(x^{k+1})$ and $\nabla f(x^{k+1})$.

Step 7: Update. Update H^k by meas of (2.7).

Step 8: Continue with the next iteration. Increase the iteration counter $k = k + 1$ and go to back to Step 2.

4. Global Convergence

In order to get the global convergence of Algorithm 2, we need the following assumption.

Assumption A. There exists positive scalars ρ_1, ρ_2 such that any matrix $\overline{H}^k, k = 1, 2, ...,$ satisfies

$$\rho_1\|z\|^2 \leq Z^T\overline{H}^k Z \leq \rho_2\|z\|^2, for\ all\ nonzero\ z \in R^{|F^k|}.$$

Lemma 4.1 *Assume that $d^k \neq 0$ and let d^k be the search direction from (3.8)-(3.10), then*

$$\min\{1, \frac{\|u-l\|_\infty}{\|d^k\|_\infty}\} \geq \beta^k \geq \min\{1, \frac{\epsilon_k}{\|d^k\|_\infty}\} \tag{4.1}$$

where $\epsilon_k = \min\{|a_i(x^k)g_i(x^k)|, |b_i(x^k)g_i(x^k)|, i \in F^k, g_i(x^k) \neq 0\}$ and $\beta^k = \sup_{0\leq\gamma\leq 1}\{\gamma | l \leq x^k + \gamma d^k \leq u\}$.

Proof. From the definition of β^k, x^k and $x^k + \beta^k d^k$ are feasible points of (1.1), we have

$$\|\beta^k d^k\|_\infty \leq \|u - l\|_\infty.$$

So the first part of (4.1) is true.
Now we prove that the second part of (4.1) holds. It is sufficient to show that

$$x_i^k + \overline{\beta} d_i^k \in [l_i, u_i] \tag{4.2}$$

for all $i = 1, ..., n$, where $\overline{\beta} = \min\{1, \frac{\epsilon_k}{\|d^k\|_\infty}\}$. If $i \in L(x^k)$, it follows from definition (3.8) that $x_i^k + d_i^k = l_i$, similarly for $i \in U(x^k)$. If $i \in F(x^k)$, we get

$$x_i^k > l_i + a_i(x^k)\nabla f_i(x^k),$$

$$x_i^k < u_i + b_i(x^k)\nabla f_i(x^k).$$

Suppose that there exists an $i \in F^k$ such that $\nabla f_i(x^k) < 0$, by the definition (3.10), we get $d_i^k > 0$, thus

$$u_i > x_i^k + (-b_i(x^k)\nabla f_i(x^k)) \geq x_i^k + \epsilon_k \frac{d_i^k}{\|d_i^k\|_\infty} \geq x_i^k + \overline{\beta} d_i^k.$$

Similarly, for $\nabla f_i(x^k) > 0$, we have $x_i^k + \overline{\beta} d_i^k \geq l_i$. When $i \in F(x^k)$ and $g_i(x^k) = 0$, the conclusion is obvious. Therefore we have shown that (4.2) holds for all $i = 1, ..., n$. The proof is complete. □

Lemma 4.2 *Let $x^k \in K$ and d^k is the direction defined by (3.8)-(3.10), then we have*

$$\|d^k\|^2 \leq -\gamma \nabla f(x^k)^T d^k, \tag{4.3}$$

for some positive scalar γ.

Proof. Since $\overline{H}^k$ is a symmetric positive definite matrix, from (3.10), we get

$$d^k_{F^k} = -(Z\overline{H}^k Z^T g^k)_{F^k}.$$

Assumption A yields

$$\rho_1 \|d^k_{F^k}\|^2 \leq -\nabla f_{F^k}(x^k)^T d^k_{F^k} \leq \rho_2 \|d^k_{F^k}\|^2,$$

thus

$$\nabla f_{F^k}(x^k)^T d^k_{F^k} \leq -\rho_1 \|d^k_{F^k}\|^2. \tag{4.4}$$

Now we show that there exists a positive scalar γ_i satisfying

$$\nabla f_i(x^k)^T d^k_i \leq -\gamma_i (d_i)^2, \tag{4.5}$$

for each $i \in L^k \cup U^k$. If $d^k_i = 0$ the inequality holds trivially. So suppose that $d^k_i \neq 0$. We only show the inequality for $i \in L^k$, since the case of $i \in U^k$ is analogous.
Considering $x^k \in K$ and $d^k_i = l_i - x^k_i$ for each $i \in L^k$, then all nonzero d^k_i must be negative. Then it follows from the definition of the set L^k that

$$a_i(x^k)\nabla f_i(x^k) \geq -d^k_i. \tag{4.6}$$

If $a_i(x^k) = 0$ then $x^k_i = l_i$ and this implies $d^k_i = 0$, so $a_i(x^k) > 0$. Then $a_i(x^k) > 0$, $d^k_i < 0$, from (4.6), we have

$$\nabla f_i(x^k)^T d^k_i \leq -\frac{1}{a_i(x^k)}(d^k_i)^2.$$

but $a_i(x^k)$ is bounded from above on K, whence there exists

$$\xi_i \geq \sup_{l \leq x \leq u} a_i(x) > 0$$

and (4.5) holds with $\gamma_i = 1/\xi_i$. So some positive constant exist shows our claims. This completes the proof. □

Lemma 4.3 *Let x^k, d^k be given iterates of Algorithm 2. Then x^k is a KKT point of (1.1) if and only if $d^k = 0$.*

Proof. Let $d^k = 0$. If $i \in L^k$, by (3.7) and (3.8), we get

$$0 = d^k_i = l_i - x^k_i \geq -a_i(x^k)\nabla f_i(x^k).$$

Since $x^k_i = l_i, a_i(x^k) > 0$, whence $\nabla f_i(x^k) \geq 0$. On the other hand, for each $i \in U^k$, we have

$$0 = d^k_i = u_i - x^k_i \leq -b_i(x^k)\nabla f_i(x^k),$$

whence $\nabla f_i(x^k) \leq 0$. If $d^k_{F^k} = 0$, by

$$d^k_i = -(Z\overline{H}^k Z^T g^k)_i, i \in F^k.$$

Since $\overline{H}^k$ is a positive definite matrix, we must have $\nabla f_i(x^k) = 0$.
Suppose that x^k is a stationary point of on k. Hence it follows from (3.4) and (3.6) that

$$L^k = \{i : x^k_i = l^i\},\ F^k = \{i : l_i < x^k_i < u_i\},\ U^k = \{i : x^k_i = u_i\}.$$

Thus, by (3.8) and (3.9), $d_{L^k} = d_{U^k} = 0$. Since $\nabla f_{F^k}(x^k) = 0$, $\overline{H}^k$ is a positive definite matrix, and (3.10), Then $d_{F^k} = 0$. Therefore $d^k = 0$. The proof is complete. □

By Lemmas 3.1 and 4.3, we know that d^k is a descent direction if x^k is not a KKT point. Similar to [25], it is not difficult to get the global convergence theorem of Algorithm 2. Here we also prove it as follows.

Theorem 4.1 *Suppose that Assumption A holds. Let x^k, d^k, and $\overline{H}^k$ be computed by the Algorithm 2 for solving the problem (1.1) and assume that $f(x)$ is twice continuously differentiable in K and, there exists a positive constant γ_1 such that $\|Z^T\overline{H}^k Z\| \leq \gamma_1$ for all k. Then every accumulation point of $\{x^k\}$ is a KKT point of the problem (1.1).*

Proof. In view of Lemma 4.2, we get

$$\|d^k\|^2 \leq -\gamma \nabla f(x^k)^T d^k.$$

Further,

$$\begin{aligned}\|d^k\|^2 &= \|Z\overline{H}^k Z^T g^k\|^2 + \sum_{i\in L^k}(l_i - x^k_i)^2 + \sum_{i\in U^k}(u_i - x^k_i)^2 \\ &\leq \gamma_1\|g^k\|^2 + \sum_{i\in L^k}(a_i(x^k)\nabla f_i(x^k))^2 + \sum_{i\in U^k}(b_i(x^k)\nabla f_i(x^k))^2 \\ &= (\gamma_1 + \mu^k)\|g^k\|^2 \leq (\gamma_1 + \mu^k)\eta_1, \end{aligned} \tag{4.7}$$

where $\eta_1 = \max_{x\in K}\|g^k\|^2$ and $\mu^k = \sum_{i\in L^k}(a_i(x^k))^2 + \sum_{i\in U^k}(b_i(x^k))^2$. By (4.1) and (4.7), there exists a constant $\overline{\beta} \in (0, 1)$ such that

$$\beta^k \geq \overline{\beta}, \quad \forall\ k. \tag{4.8}$$

If $\alpha_k < 0.1\overline{\beta}$, from the definition of α_k there exists $j \geq 0$ satisfying $\alpha_{k,j} \leq 10\alpha_k$, and $\alpha_{k,j}$ is an unacceptable steplength, which means that

$$f(x^k) + \sigma\alpha_{k,j}(g^k)^T d^k \leq f(x^k + \alpha_k d^k) \leq f(x^k) + \alpha_{k,j}(g^k)^T d^k + \frac{1}{2}\eta_2\alpha^2_{k,j}\|d^k\|^2, \tag{4.9}$$

where $\eta_2 = \max_{x\in K}\|\nabla^2 f(x)\|$.
The above inequality and (4.3) imply that

$$\alpha_{k,j} \geq \frac{-2(1-\sigma)(g^k)^T d^k}{\eta_2\|d^k\|^2} \geq \frac{2(1-\sigma)}{\eta_2\gamma}. \tag{4.10}$$

Thus $\alpha^k \geq 0.1\alpha_{k,j}$ and the above inequality yields to

$$\alpha_k \geq \min\{\frac{-(1-\sigma)}{5\eta_2\gamma}, 0.1\overline{\beta}\} > 0, \ \forall\ k. \tag{4.11}$$

Because k is a bounded set,

$$\infty > \sum_{k=1}^{\infty}(f(x^k) - f(x^{k+1})) \geq \sum_{k=1}^{\infty} -\sigma\alpha_k(g^k)^T d^k. \tag{4.12}$$

Combining (4.11) and (4.12), we obtain

$$\sum_{k=1}^{\infty} -(g^k)^T d^k < \infty \tag{4.13}$$

this means

$$\lim_{k\to\infty}(g^k)^T d^k = 0. \tag{4.14}$$

Using (4.14) and

$$(d^k)^T g^k = -(g^k)^T Z\overline{H}^k Z^T g^k + \sum_{i\in L^k}(l_i - x_i^k)g_i^k + \sum_{i\in U^k}(u_i - x_i^k)g_i^k,$$

therefore, we get

$$\lim_{k\to\infty}\|Z^T g^k\| = 0. \tag{4.15}$$

$$\lim_{k\to\infty}\sum_{i\in L^k}(l_i - x_i^k)g_i^k = 0. \tag{4.16}$$

$$\lim_{k\to\infty}\sum_{i\in U^k}(u_i - x_i^k)g_i^k = 0. \tag{4.17}$$

Let $\overline{x}$ be any accumulation point of $\{x^j\}$, then there exists a subspace $\{x^{k_i}\}$ $(i = 1, 2, ...)$ satisfying

$$\lim_{k\to\infty} x^{k_i} = \overline{x}. \tag{4.18}$$

By (3.5) and (3.6), if $\overline{x}$ is not a KKT point, there exists $j \in \overline{J}$ $(or\ j \in \overline{U})$ such that

$$g_j(\overline{x}) < 0 \ \ (or\ \ g_j(\overline{x}) > 0) \tag{4.19}$$

or there exists $j \in \overline{F}$ satisfying

$$g_i(\overline{x}) \neq 0 \tag{4.20}$$

For some $j \in \overline{F}$, if (4.20) holds. By ((4.15)-(4.17)), for all sufficiently large i, we have $j \in L(x^{k_i}) \cup U(x^{k_i}) \cup F(x^{k_i})$, it is impossible. The proof is complete. □

5. Numerical Results

This section reports detailed results on a set of test problems from CUTE (Conn et al. [8]) by Algorithm 1. Problems $EDENSCH$ and $PLENTY1$ were from [5]. All codes were written in MATLAB 7.5 and run on PC with 2.60GHz CPU processor and 480MB memory and Windows XP operation system. In paper [25], the authors propose a new subspace L-BFGS method (called XW-method) for (1.1), numerical results shows that their method is competitive to PAL-BFGS method that is the projected BFGS method PROJBFGS [13]. Then we will test these problems by our presented method against the method of Xiao and Wei [25]. The following stop rule and parameters are the same as Xiao and Wei [25]. We will stop this program if the condition

$$\|P_\Omega(x^k - \nabla f(x^k)) - x^k\| \leq 10^{-5}.$$

Choosing $\sigma = 10^{-1}$ in Armijo line search, $a_i(x) = b_i(x) = 10^{-5}$ in (3.7), $\theta = 1$ and the "basic matrix" to be the identity matrix I in the limited memory BFGS method, and $m = 5$. In the L-BFGS update (2.7) of [25], the update rule in [22] will be used. The columns of the table have the following meaning:

Problem: name of the test problem.

Dim: the dimension of the problem.

NI: the total number of iterations.

NF: the iteration number of the function evaluations.

Time: cpu time in seconds.

TABLE 1
Test results for Algorithm 1 and XW-method

		XW-method	Algorithm 1
Problem	Dim	NI/NF/Time	NI/NF/Time
NONSCOMP	1000	19/37/5.637487e+000	6/15/1.106377e+000
EXPLIN	500	44/61/2.997471e+000	19/37/2.376656e+000
EXPLIN2	1000	110/127/1.535500e+001	104/173/1.552215e+001
EXPQUAD	500	8/9/6.827298e-001	8/9/6.750947e-001
MCCORMCK	1000	2/3/5.549596e-001	3/15/7.082210e-001
PROBPENL	500	9/25/7.762503e-001	2/4/3.413140e-001
QRTQUAD	1000	331/332/4.420028e+001	352/353/4.653462e+001
HATFLDC	1000	53/106/7.697684e+000	42/84/5.952208e+000
HS110	500	0/1/1.752058e-001	0/1/1.776128e-001
BIGGSB1	1000	1/2/3.501677e-001	1/2/3.488427e-001
HATFLDA	500	10/22/8.141783e-001	12/22/8.881171e-001
EDENSCH2	500	6/8/5.633389e-001	1/2/2.818517e-001
EDENSCH3	1000	3/5/6.952280e-001	2/3/5.085141e-001
EDENSCH5	1000	12/89/2.639427e+000	40/263/6.938283e+000
PENALTY1	500	60/107/3.618748e+000	34/98/2.142757e+000
PENALTY1	1000	9/97/2.141580e+000	4/46/1.332802e+000
PENALTY1	1000	9/73/1.726358e+000	1/29/8.641336e-001
PENALTY1	1000	9/97/2.115441e+000	4/46/1.334251e+000

The numerical results indicate that the Algorithm 1 performs better than the XW-method does for the test problems from the table 1. Moreover, for most problems, the number of the iterations and the function iteration on Algorithm 1 is less than those of the normal L-BFGS method.

Acknowledgments. The authors would like to thank Professor Y. H. Xiao of Henan University for providing us the CUTE test problems.

References

[1] D. P. Bertsekas, Projected Newton methods for optimization problems with simple constraints, SIAM. J. Contr. Opt., 20(1982), pp. 221-246.

[2] J. V. Burke, J. J. Moré, On the identification of active constraints, SIAM J. Numer. Anal., 25(1998), pp. 1197-1211.

[3] E. G. Birgin, J. M. Martinez, Large-scale active-set box-constrained optimization method with spectral projected gradients, Comput. Opt. Appl., 23(2002), pp. 101-125.

[4] R. H. Byrd, J. Nocedal, R. B. Schnabel, Representations of quasi-Newton matrices and their use in limited memory methods, Math. Prgram., 63(1994), pp. 129-156.

[5] R. H. Byrd, P. H. Lu, J. Nocedal, A limited memory algorithm for bound constrained optimization, SIAM J. Statist. Sci. Comput., 16(1995), pp. 1190-1208.

[6] P. Calamai, J. J. Moré, Projected gradient for linearly constrained programms, Math. Program., 39(1987), pp. 93-116.

[7] A. R. Conn, N. I. M. Gould, Ph. L. Toint, Global convergence of a class of trust region algorithm for optimization with simple bounds, SIAM J. Numer. Anal., 25(1988), pp. 433-460.

[8] A. R. Conn, N. I. M. Gould, Ph. L. Toint, CUTE: constrained and unconstrained testing environment, ACM Trans. Math. Softw., 21(1995), pp. 123-160.

[9] W. C. Davidon, Variable metric methods for minimization, Argonne National Labs Report, Number: ANL-5990, 1959.

[10] F. Facchinei, J. Júdice, J. Soares, An active set Newton algorithm for large-scale nonlinear programs with box constraints, SIAM J. Opt., 8(1998), pp. 158-186.

[11] F. Facchinei, S. Lucidi, L. Palagi, A truncated Newton algorithm for large scale box constrained optimization, SIAM J. Opt., 12(2002), pp. 1100-1125.

[12] D. Goldfarb, Extension of Davidon's variable metric algorithm to maximization under linear inequality and constraineds, SIAM J. Appl. Math., 17(1969), pp. 739-764.

[13] C. T. Kelly, Iterative methods for optimization, SIAM, Philadelphia, 1999, pp. 102-104.

[14] M. Lescrenier, Convergence of trust region algorithm for optimization with bounds when strict complementarity does not hold, SIAM J. Numer. Anal., 28(1991), pp. 467-695.

[15] D. Li, M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, Journal of Computational and Applied Mathematics, 129(2001), pp. 15-35.

[16] D. Li, M. Fukushima, On the global convergence of the BFGS methods for nonconvex unconstrained optimization problems, SIAM J. Optim., 11(2001), pp. 1054-1064.

[17] C. J. Lin, J. J. Moré, Newton's method for large bound-constrained optimization problems, SIAM J. Opt., 9(1999), pp. 1100-1127.

[18] D. G. Lueberger, Introduction to linear and nonlinear programming, Addison-Wesley, Reading, MA, 1973, Ch. 11.

[19] Q. Ni, A subspace projected conjugate algorithm for large bound constrained quadratic programming, Numer. Math. (a Journal of Chese Universities), 7(1998), pp. 51-60.

[20] Q. Ni, Y. X. Yuan, A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound consrtained optimization, Math.Comp., 66(1997), pp. 1509-1520.

[21] M. J. D. Powell, A new algorithm for unconstrained optimation, In: Nonlinear Programming, J. B. Rosen , O. L. Mangasarian and K. Ritter, eds. Academic Press, New York, 1970.

[22] M. J. D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, Numer. Anal., (1978), pp. 155-157.

[23] Z. Wei, G. Li, L. Qi, New Quasi-Newton Methods for unconstrained optimization problems, Appl. Math. Comput., 175(2006), pp. 1156-1188.

[24] Z. Wei, G. Yu, G. Yuan, Z. Lian, The superlinear convergence of a modified BFGS-type method for unconstrained optimization, Computational Optimization and Applications, 29(2004), pp. 315-332.

[25] Y. Xiao, Z. Wei, A new subspace limited memory BFGS algorithm for large-scale bound constrained optimization, Applied Mathematics and Computation, (1)185(2007), pp. 350-359.

[26] G. L. Yuan, X. W. Lu, A new line search method with trust region for unconstrained optimization, Communications on Applied Nonlinear Analysis, (1)15(2008), pp. 35-49.

[27] G. L. Yuan, X. W. Lu, A Modified PRP Conjugate Gradient Method, Annals of Operations Research, 2008, DOI: 10.1007/s10479-008-0420-4.

[28] G. L. Yuan, Z. X. Wei, New Line Search Methods for Unconstrained Optimization, Journal of the Korean Statistical Society, 2008, DOI: 10.1016/j.jkss.2008.05.004.

[29] G. L. Yuan, Z. X. Wei, The Superlinear Convergence Analysis of a Nonmonotone BFGS Algorithm on Convex Objective Functions, Acta Mathematica Sinica, English Series, (1)14(2008), pp. 35-42.

[30] G. L. Yuan, Z. X. Wei, Convergence Analysis of A Modified BFGS Method on Convex Minimizations, Computational Optimization and Applications, Revised, 2008.

[31] J. Z. Zhang, N. Y. Deng, L. H. Chen, New quasi-Newton equation and related methods for unconstrained optimization, J. Optim. Theory Appl., 102(1999), pp. 147-167.