Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 684705, 9 pages
doi:10.1155/2010/684705

*Research Article*

# A Simple Sufficient Descent Method for Unconstrained Optimization

**Ming-Liang Zhang, Yun-Hai Xiao, and Dangzhen Zhou**

*Institute of Applied Mathematics, College of Mathematics and Information Science, Henan University, Kaifeng 475004, China*

Correspondence should be addressed to Yun-Hai Xiao, yunhai816@163.com

Received 18 April 2010; Revised 26 August 2010; Accepted 26 October 2010

Academic Editor: Joaquim J. Júdice

We develop a sufficient descent method for solving large-scale unconstrained optimization problems. At each iteration, the search direction is a linear combination of the gradient at the current and the previous steps. An attractive property of this method is that the generated directions are always descent. Under some appropriate conditions, we show that the proposed method converges globally. Numerical experiments on some unconstrained minimization problems from CUTEr library are reported, which illustrate that the proposed method is promising.

## 1. Introduction

In this paper, we consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, and its gradient at point $x_k$ is denoted by $g(x_k)$, or $g_k$ for the sake of simplicity. $n$ is the number of variables, which is automatically assumed to be large. Large-scale optimization is one of the important research areas both in optimization theory and algorithm design. There exist some kinds of effective methods available for solving (1.1), as for instance, inexact Newton, limited memory quasi-Newton, conjugate gradient, spectral gradient, and subspace methods [1–4].

The iterative formula the method is given by

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

where $\alpha_k$ is a steplength, and $d_k$ is a search direction. Generally, we say that $d_k$ is a descent direction of $f$ at $x_k$ if $d_k^\top g_k < 0$. Furthermore, if there exists a positive constant $C > 0$ such that $d_k^\top g_k \le -C\|g_k\|^2$, then $d_k$ is often called a sufficient descent direction. The descent property is very important for the iterative method to be global convergent, especially for the conjugate gradient method [5]. Unfortunately, for the earliest spectral gradient method [6] and the PRP conjugated gradient method, their generated directions are not always descent. Therefore, in the last years, much effort has been done to smooth away the drawback and develop new methods, for example, [7].

As is wellknown, if Armijo line search is used, the standard PRP method can cycle infinitely without approaching any optimal point when solving nonconvex minimization problems. To overcome this drawback, Hager and Zhang [8] made a slight modification on the PRP formula. Consequently, the resulting method own the remarkable property: for any line search, the generated direction always satisfies the sufficient descent condition $d_k^\top g_k \le -(7/8)\|g_k\|^2$. Moreover, Zhang et al. [9] further studied the PRP formula and proposed a three-term modified PRP method, in which the generated directions satisfy

$$d_k^\top g_k = -\|g_k\|^2, \tag{1.3}$$

and this property is independent of any line search. Additionally, for improving the numerical performance of the standard FR method, Zhang et al. [10] developed a modified FR method where the direction also satisfy (1.3).

Although much progress has been made in designing a sufficient descent direction in a conjugate gradient method, it seemingly receive little attention in other methods. Very recently, An et al. [5] proposed a robust technique to construct a sufficient descent direction for unconstrained optimization. The descent direction is a linear combination of the steepest descent direction and the projection of the original direction, that is,

$$d_k = -g_k + \lambda_k \left( I - \frac{g_k g_k^\top}{\|g_k\|^2} \right) \overline{d}_k, \tag{1.4}$$

where $\lambda_k$ is a scalar and $\overline{d}_k$ is an original direction. Based on the definition of $d_k$, it is easy to deduce that the sufficient descent condition (1.3) holds. This technique has been applied to the PSB updated formula and showed that the resulting method converges globally and superlinearly for an uniformly convex function. Moreover, the direction in [11] is truly a special case of (1.4).

We note that (1.4) can be considered as a general form of a nonlinear conjugate gradient method, in which

$$d_k = -g_k + \beta_k d_{k-1}, \tag{1.5}$$

where $\beta_k$ is a scalar. Comparing with (1.4), we see that the scalar $\beta_k$ takes the place of the coefficient matrix of $\overline{d}_k$. In this paper, we further study the sufficient descent technique (1.4), and propose a simple sufficient descent method for solving unconstrained optimization problems. Our motivation is simple, we choose $\overline{d}_k$ as the gradient of the previous step in (1.4). The search direction in this proposed method always satisfies a sufficient descent condition.

Under some conditions, we show that the algorithm converges globally by using a special line search. The performance of the method on some CUTEr test problems indicates that it is encouraging.

We organize this paper as follows. In the next section, we construct the sufficient descent direction. In Section 3, we state the steps of our new algorithm with a special line search. We also report some experiments on some large-scale unconstrained optimization problems. Finally, we conclude this paper with some remarks in the last section. Throughout this paper, the symbol $\|\cdot\|$ denotes the Euclidean norm of a vector.

## 2. New Search Direction

This section aims to state the new direction formula and investigate its properties. Now, if we take $g_{k-1}$ as $\overline{d}_k$ and set $\lambda_k \equiv 1$ in (1.4), we get

$$d_k = -g_k + \left( I - \frac{g_k g_k^\top}{\|g_k\|^2} \right) g_{k-1}. \tag{2.1}$$

Obviously, the direction is a linear combination of the gradient at the current and the previous steps. Additionally, to obtain the global convergence of PRP method, Cheng [11] introduced a descent direction defined as

$$d_k = -g_k + \beta_k^{\mathrm{PRP}} \left( I - \frac{g_k g_k^\top}{\|g_k\|^2} \right) d_{k-1}. \tag{2.2}$$

We note that (2.2) is only a special choice of (1.4), in which $\overline{d}_k$ is replaced by $d_{k-1}$ and $\lambda_k$ is chosen as the scalar $\beta_k^{\mathrm{PRP}}$. If we denote

$$H_k = I - \frac{g_k g_k^\top}{\|g_k\|^2}, \tag{2.3}$$

then it is easy to see that for any $y \neq 0$, we have

$$y^\top H_k y = y^\top H_k H_k^\top y = \left\| H_k^\top y \right\|^2 \geq 0, \tag{2.4}$$

which indicates that $H_k$ is a symmetric and positive semidefinite matrix.

## 3. Sufficient Descent Method

As we have stated in the previous section, the directions in (2.1) satisfy a sufficient descent condition. In this section, we list our algorithm and establish its global convergence.

Firstly, we state the steps of the simple sufficient descent method. In order to achieve the global convergence of our method, we consider the backtracking line search of

Grippo and Lucidi (GL) [12]. That is, for given $\delta \in (0,1)$, $\beta > 0$, $\rho \in (0,1)$, find the first $j_k$ ($j = 1, 2, \ldots$) such that $\alpha_k = \beta\rho^{j_k}$ satisfies

$$f(x_k + \alpha_k d_k) \le f(x_k) - \delta\alpha_k^2 \|d_k\|^2. \tag{3.1}$$

Now, we are ready to state the steps of the Simple Sufficient Descent (SSD) method.

*Algorithm 3.1* (SSD). We have the following steps.

*Step 1.* Given $x_0 \in \mathbb{R}^n$. Let $0 < \delta < 1$, and $d_0 = -g_0$. Set $k := 0$.

*Step 2.* If $\|g_k\| = 0$, then stop.

*Step 3.* Compute $d_k$ by (2.1).

*Step 4.* Find the steplength $\alpha_k$ satisfying (3.1). Let $x_{k+1} = x_k + \alpha_k d_k$.

*Step 5.* Set $k := k + 1$, go to Step 2.

The remainder of this section is devoted to investigate the global convergence of Algorithm 3.1. We first state some assumptions.

*Assumption 3.2.* Function $f$ is continuously differentiable and the level set $\mathcal{F} = \{x \in \mathbb{R}^n \mid f(x) \le f(x_0)\}$ is bounded.

*Assumption 3.3.* The gradient of $f$ is Lipschitz continuous, that is, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{3.2}$$

We first prove the following lemma.

**Lemma 3.4.** *Let the sequence $\{x_k\}$ be generated by Algorithm 3.1. Suppose that Assumptions 3.2 and 3.3 hold. Then,*

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\|d_k\|^2} < \infty. \tag{3.3}$$

*Proof.* We have the following cases.

*Case 1.* Consider firstly the GL line search (3.1). If $\alpha_k \ne \beta$, then $\rho^{-1}\alpha_k$ may not satisfy (3.1). That is,

$$f\left(x_k + \rho^{-1}\alpha_k d_k\right) > f(x_k) - \delta\rho^{-2}\alpha_k^2 \|d_k\|^2. \tag{3.4}$$

By the mean value theorem and Assumption 3.3, there exists $\theta_k \in (0,1)$ such that

$$\begin{aligned} f\left(x_k + \rho^{-1}\alpha_k d_k\right) - f(x_k) &= \rho^{-1}\alpha_k g\left(x_k + \theta_k \rho^{-1}\alpha_k d_k\right)^{\top} d_k \\ &= \rho^{-1}\alpha_k g_k^{\top} d_k + \rho^{-1}\alpha_k \left( g\left(x_k + \theta_k \rho^{-1}\alpha_k d_k\right)^{\top} - g_k \right) d_k \\ &\leq \rho^{-1}\alpha_k g_k^{\top} d_k + L\rho^{-2}\alpha_k^2 \|d_k\|^2, \end{aligned} \tag{3.5}$$

where $L > 0$ is the Lipschitz constant of $g$. Substituting the last inequality into (3.4), we get

$$\alpha_k \geq -\frac{1}{\rho^{-1}(L+\delta)} \frac{\|g_k\|^2}{\|d_k\|^2} = c_1 \frac{\|g_k\|^2}{\|d_k\|^2}. \tag{3.6}$$

This implies that there is a constant $M > 0$ such that

$$\frac{\|g_k\|^4}{\|d_k\|^2} \leq M\left(f(x_k) - f(x_{k+1})\right). \tag{3.7}$$

*Case 2.* If $\alpha = \beta$, then we have

$$\frac{\|g_k\|^4}{\|d_k\|^2} \leq \|g_k\|^2 \leq \|d_k\|^2 \leq \delta^{-1}\beta^{-2}\left(f(x_k) - f(x_{k-1})\right), \tag{3.8}$$

which shows that (3.7) holds with $M = \delta^{-1}\beta^{-2}$. Moreover, (3.7) indicates the lemma claims. □

**Theorem 3.5.** *Let* $\{x_k\}$ *be generated by Algorithm 3.1 with GL backtracking line search* (3.1)*. Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.9}$$

*Proof.* From Assumption 3.3, we know that there exists a positive constant $\overline{\gamma}$ such that

$$\|g_k\| \leq \overline{\gamma}. \tag{3.10}$$

By the definition of $d_k$ in (2.1), we have

$$\|d_k\| \leq \|g_k\| + \|H_k g_{k-1}\| \leq \|g_k\| + \|g_{k-1}\| \leq 2\overline{\gamma}, \tag{3.11}$$

which shows that $\|d_k\|$ is bounded. We get from (3.1) that

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_k\|^2 < \infty. \tag{3.12}$$

Consequently, we have

$$\lim_{k\to\infty} \alpha_k \|d_k\| = 0. \tag{3.13}$$

For the sake of contradiction, we suppose that (3.9) does not hold. Then there is a constant $\epsilon > 0$ such that $\|g_k\| \geq \epsilon$ for all $k \geq 0$.

If $\lim\inf_{k\to\infty}\alpha_k > 0$, we get from (3.13) that $\lim\inf_{k\to\infty}\|d_k\| = 0$, which shows $\lim\inf_{k\to\infty}\|g_k\| = 0$. This contradicts the assumption (3.9).

If $\lim\inf_{k\to\infty}\alpha_k = 0$, that is, there is an infinite index set $K$ such that

$$\lim_{k\in K, k\to\infty} \alpha_k = 0. \tag{3.14}$$

It follows from the line search step in Algorithm 3.1 that when $k \in K$ is sufficiently large, $\rho^{-1}\alpha_k$ does not satisfy (3.1). Then from the first part of the proof of Lemma 3.4 we see that (3.6) holds. Since $\|d_k\|$ is bounded and $\lim_{k\in K,k\to\infty}\alpha_k = 0$, (3.6) implies $\lim\inf_{k\to\infty}\|g_k\| = 0$. This also yields a contradiction. The proof is complete. □

## 4. Numerical Results

In this section, we test the feasibility and effectiveness of the Algorithm SSD. The algorithm is implemented in Fortran77 code using double precision arithmetic. All runs are performed on a PC ( Intel Pentium Dual E2140 1.6 GHz, 256 MB SDRAM) with Linux operations system. The algorithm stops if the infinity-norm of the final gradient is below $10^{-5}$, that is

$$\left\|\nabla f(x)\right\| \leq 10^{-5}. \tag{4.1}$$

The iteration is also stopped if the number of iterations exceeds 10000 or the number of function evaluations reaches 20000. Our experiments are performed on the subset of the nonlinear unconstrained problems from CUTEr [13] collection. The second-order derivatives of all the selected problems are available. Since we are interested in large problems, we refined this selection by considering only problems where the number of variables is at least 50. The parameter in line search (3.1) is taken as: $\beta = 1$, $\rho = 0.1$, and $\delta = 10^{-4}$. The numerical results of the algorithms SSD are listed in Table 1. The columns have the following meanings:

Problem: name of the test problem,

Dim: dimension of the test problem,

Iter: number of iterations,

Nf: number of function evaluations,

Time: CPU time in seconds,

Fv: final function value,

Norm-2: $l_2$-norm of the final gradient, and

Norm-0: $\infty$-norm of the final gradient.

**Table 1:** Test results for SSD method.

| Problem | Dim | Iter | Nf | Time | Fv | Norm-2 | Norm-0 |
|---|---|---|---|---|---|---|---|
| ARGLINA | 200 | 2 | 3 | 0.01 | $0.2000000E+03$ | $0.15350E-12$ | $0.28232E-23$ |
| ARWHEAD | 5000 | 12 | 34 | 0.03 | $0.5550005E-12$ | $0.80345E-05$ | $0.89063E-10$ |
| BROWNAL | 200 | 3175 | 5126 | 1.40 | $0.1475222E-08$ | $0.29801E-05$ | $0.93967E-10$ |
| BROYDN7D | 5000 | 5986 | 9664 | 20.09 | $0.1815779E+04$ | $0.17731E-05$ | $0.20076E-10$ |
| BRYBND | 5000 | 105 | 160 | 0.28 | $0.3534654E-11$ | $0.29153E-05$ | $0.29287E-10$ |
| DIXMAANA | 3000 | 140 | 190 | 0.11 | $0.1000000E+01$ | $0.24750E-06$ | $0.93362E-10$ |
| DIXMAANB | 3000 | 130 | 179 | 0.10 | $0.1000000E+01$ | $0.30449E-06$ | $0.61730E-10$ |
| DIXMAANC | 3000 | 120 | 166 | 0.10 | $0.1000000E+01$ | $0.10046E-05$ | $0.86335E-10$ |
| DIXMAAND | 3000 | 139 | 188 | 0.11 | $0.1000000E+01$ | $0.15385E-05$ | $0.77064E-10$ |
| QDRTIC | 5000 | 1095 | 1757 | 1.72 | $0.2290216E-10$ | $0.95704E-05$ | $0.94807E-10$ |
| DQRTIC | 5000 | 196 | 282 | 0.11 | $0.3480734E-06$ | $0.68218E-06$ | $0.46755E-10$ |
| EDENSCH | 2000 | 87 | 136 | 0.06 | $0.1200328E+05$ | $0.38002E-05$ | $0.97655E-10$ |
| ENGVAL1 | 5000 | 70 | 120 | 0.12 | $0.5548668E+04$ | $0.37840E-05$ | $0.82778E-10$ |
| FLETCBV2 | 5000 | 0 | 1 | 0.00 | $-0.5002682E+00$ | $0.79960E-07$ | $0.19455E-10$ |
| MANCINO | 100 | 26 | 56 | 0.71 | $0.3102390E-17$ | $0.91840E-06$ | $0.23394E-10$ |
| QUARTC | 5000 | 196 | 282 | 0.11 | $0.3480734E-06$ | $0.68218E-06$ | $0.46755E-10$ |
| SCHMVETT | 5000 | 255 | 386 | 1.12 | $-0.1499400E+05$ | $0.68017E-05$ | $0.67249E-10$ |
| SENSORS | 100 | 56 | 184 | 0.85 | $-0.2101781E+04$ | $0.46106E-05$ | $0.35706E-10$ |
| SPMSRTLS | 4999 | 3489 | 5629 | 8.74 | $0.1833147E-08$ | $0.90139E-05$ | $0.96204E-10$ |
| TOINTGSS | 5000 | 91 | 132 | 0.11 | $0.1000400E+02$ | $0.76757E-05$ | $0.60378E-10$ |
| VARDIM | 200 | 1 | 2 | 0.00 | $0.1479114E-30$ | $0.44409E-15$ | $0.59165E-30$ |
| VAREIGVL | 50 | 67 | 106 | 0.00 | $0.1042896E-10$ | $0.38234E-05$ | $0.87845E-10$ |
| EG2 | 1000 | 4 | 5 | 0.00 | $-0.9989474E+03$ | $0.13514E-05$ | $0.18263E-11$ |
| MOREBV | 5000 | 172 | 279 | 0.22 | $0.1462269E-08$ | $0.42417E-05$ | $0.88342E-10$ |
| TOINTGOR | 50 | 1432 | 2303 | 0.02 | $0.1373905E+04$ | $0.49818E-05$ | $0.88106E-10$ |
| TOINTQOR | 50 | 247 | 376 | 0.00 | $0.1175472E+04$ | $0.55492E-05$ | $0.88045E-10$ |

In addition, we also present extensive numerical results of the state-of-the-art algorithm CG_DESCENT [8, 14]. CG_DESCENT is a conjugate gradient descent method for solving large-scale unconstrained optimization problems. A new nonmonotone line search were used in CG_DESCENT which makes this algorithm very efficient. The Fortran code can be obtained from http://www.math.ufl.edu/~hager/. When running of CG_DESCENT, default values are used for all parameters. The results are summarized in Table 2.

Observing the tables, we see that SSD works well, as it reached a stationary point based on the stopping criterion (4.1) for almost all these test problems. Although SSD requires large number of iterations or function evaluations, it needs less time consuming comparatively. However, the numerical comparisons tell us that the state-of-the-art algorithm CG_DESCENT performs a little better than SSD, but for some specific problems, the enhancement of SSD is still noticeable. The numerical experiments also show that SSD can potentially used to solve unconstrained optimization problems with higher dimensions.

## 5. Concluding Remarks

In this paper, motivated by a descent technique, we have proposed a simple sufficient descent method for solving large-scale unconstrained optimization problems. At each iteration, the generated directions are only related to the gradient information of two successive points.

**Table 2:** Test results for CG_DESCENT.

| Problem | Dim | Iter | Nf | Ng | Time | Fv | Norm-2 |
|---|---|---|---|---|---|---|---|
| ARGLINA | 200 | 1 | 3 | 2 | 0.01 | $0.20000E + 03$ | $0.21133E - 06$ |
| ARWHEAD | 5000 | 9 | 19 | 12 | 0.02 | $0.00000E + 00$ | $0.83931E - 06$ |
| BROWNAL | 200 | 4 | 15 | 13 | 0.01 | $0.14731E - 08$ | $0.11790E - 05$ |
| BROYDN7D | 5000 | 1444 | 2885 | 1453 | 5.13 | $0.19785E + 04$ | $0.68854E - 05$ |
| BRYBND | 5000 | 31 | 64 | 34 | 0.10 | $0.14292E - 10$ | $0.65055E - 05$ |
| DIXMAANA | 3000 | 8 | 17 | 9 | 0.01 | $0.10000E + 01$ | $0.15586E - 05$ |
| DIXMAANB | 3000 | 9 | 19 | 10 | 0.00 | $0.10000E + 01$ | $0.35546E - 06$ |
| DIXMAANC | 3000 | 10 | 21 | 11 | 0.01 | $0.10000E + 01$ | $0.70236E - 06$ |
| DIXMAAND | 3000 | 11 | 23 | 12 | 0.01 | $0.10000E + 01$ | $0.32593E - 05$ |
| DQDRTIC | 5000 | 7 | 15 | 8 | 0.02 | $0.41826E - 14$ | $0.18957E - 06$ |
| DQRTIC | 5000 | 32 | 65 | 33 | 0.02 | $0.28560E - 04$ | $0.30250E - 05$ |
| EDENSCH | 2000 | 29 | 56 | 33 | 0.02 | $0.12003E + 05$ | $0.74432E - 05$ |
| ENGVAL1 | 5000 | 23 | 45 | 32 | 0.05 | $0.55487E + 04$ | $0.53164E - 05$ |
| FLETCBV2 | 5000 | 0 | 1 | 1 | 0.01 | $-0.50027E + 00$ | $0.79960E - 07$ |
| MANCINO | 100 | 11 | 23 | 12 | 0.30 | $0.20215E - 17$ | $0.97074E - 06$ |
| QUARTC | 5000 | 32 | 65 | 33 | 0.02 | $0.28560E - 04$ | $0.30250E - 05$ |
| SCHMVETT | 5000 | 35 | 63 | 46 | 0.19 | $-0.14994E + 05$ | $0.90275E - 05$ |
| SENSORS | 100 | 23 | 55 | 40 | 0.19 | $-0.21085E + 04$ | $0.29929E - 05$ |
| SPMSRTLS | 4999 | 186 | 379 | 195 | 0.51 | $0.19502E - 08$ | $0.82452E - 05$ |
| TOINTGSS | 5000 | 3 | 7 | 4 | 0.01 | $0.10002E + 02$ | $0.60471E - 05$ |
| VARDIM | 200 | 28 | 57 | 29 | 0.00 | $0.27052E - 22$ | $0.20805E - 08$ |
| VAREIGVL | 50 | 52 | 142 | 90 | 0.00 | $0.40508E - 10$ | $0.71079E - 05$ |
| EG2 | 1000 | 3 | 7 | 4 | 0.00 | $-0.99895E + 03$ | $0.81318E - 05$ |
| MOREBV | 5000 | 32 | 65 | 34 | 0.04 | $0.10103E - 08$ | $0.89736E - 05$ |
| TOINTGOR | 50 | 107 | 206 | 121 | 0.00 | $0.13739E + 04$ | $0.88894E - 05$ |
| TOINTQOR | 50 | 28 | 55 | 31 | 0.00 | $0.11755E + 04$ | $0.49970E - 05$ |

Under some mild conditions, we have shown that this method is global convergent. The numerical results indicate that it works well on some selected problems from CUTEr library. We think there are several other issues that could be lead to improvements. A first point that should be considered is probably the choice of a nonmonotone line search technique. We have adopted the monotone scheme of Grippo and Lucidi (3.1), but this is not the only possible choice, and the framework developed in [15, 16] may allow one to achieve a better performance. Similar to [5], another important point that should be further investigated is a modification of the search direction without violating the nice sufficient descent property. To this end, it is interesting to note that this method is capable of solving nonlinear systems of equations without Jacobian information and cheaply computing.

## Acknowledgments

The authors thank Professor Zengxin Wei of Guangxi University for his careful reading. The authors are also grateful two anonymous referees for their useful comments and suggestions to improve the paper. The work is supported by the Chinese Postdoctoral Foundation Grant no. 20090461094, the Natural Science Foundation of China Grant no. 11001075, the Natural Science Foundation of Henan Province Eduction Department Grant no. 2010B110004.

## References

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, NY, USA, 1999.

[2] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.

[3] Y. Xiao, Z. Wei, and Z. Wang, "A limited memory BFGS-type method for large-scale unconstrained optimization," *Computers & Mathematics with Applications*, vol. 56, no. 4, pp. 1001–1009, 2008.

[4] Y. Yuan, "Subspace techniques for nonlinear optimization," in *Some Topics in Industrial and Applied Mathematics*, R. Jeltsch, D. Q. Li, and I. H. Sloan, Eds., vol. 8 of *Series in Contemporary Applied Mathematics*, pp. 206–218, Higher Education Press, Beijing, China, 2007.

[5] X.-M. An, D.-H. Li, and Y. Xiao, "Sufficient descent directions in unconstrained optimization," *Computational Optimization and Applications*. In press.

[6] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[7] J. Zhang, Y. Xiao, and Z. Wei, "Nonlinear conjugate gradient methods with sufficient descent condition for large-scale unconstrained optimization," *Mathematical Problems in Engineering*, vol. 2009, Article ID 243290, 16 pages, 2009.

[8] W. W. Hager and H. Zhang, "A new conjugate gradient method with guaranteed descent and an efficient line search," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 170–192, 2005.

[9] L. Zhang, W. Zhou, and D.-H. Li, "A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence," *IMA Journal of Numerical Analysis*, vol. 26, no. 4, pp. 629–640, 2006.

[10] L. Zhang, W. Zhou, and D. Li, "Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search," *Numerische Mathematik*, vol. 104, no. 4, pp. 561–572, 2006.

[11] W. Cheng, "A two-term PRP-based descent method," *Numerical Functional Analysis and Optimization*, vol. 28, no. 11-12, pp. 1217–1230, 2007.

[12] L. Grippo and S. Lucidi, "A globally convergent version of the Polak-Ribière conjugate gradient method," *Mathematical Programming*, vol. 78, no. 3, pp. 375–391, 1997.

[13] I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint, "CUTE: constrained and unconstrained testing environment," *ACM Transactions on Mathematical Software*, vol. 21, no. 1, pp. 123–160, 1995.

[14] W. W. Hager and H. Zhang, "Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent," *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 113–137, 2006.

[15] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[16] H. Zhang and W. W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.