Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 839659, 7 pages
http://dx.doi.org/10.1155/2015/839659

Research Article

A Spectral Dai-Yuan-Type Conjugate Gradient Method for Unconstrained Optimization

Guanghui Zhou[1,2] and Qin Ni[1]

[1] *Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[2] *Huaibei Normal University, Huaibei 235000, China*

Correspondence should be addressed to Guanghui Zhou; 163zgh@163.com

Received 1 October 2015; Accepted 13 December 2015

Academic Editor: Paolo Maria Mariano

A new spectral conjugate gradient method (SDYCG) is presented for solving unconstrained optimization problems in this paper. Our method provides a new expression of spectral parameter. This formula ensures that the sufficient descent condition holds. The search direction in the SDYCG can be viewed as a combination of the spectral gradient and the Dai-Yuan conjugate gradient. The global convergence of the SDYCG is also obtained. Numerical results show that the SDYCG may be capable of solving large-scale nonlinear unconstrained optimization problems.

1. Introduction

As well known, a great deal of issues, which are studied in scientific research fields, can be translated to unconstrained optimization problems. The spectral conjugate gradient (SCG) method does nice jobs among various algorithms for solving nonlinear optimization problems. The spectral conjugate gradient combines the spectral gradient and the conjugate gradient. To the SCG method, the choice of spectral parameter is crucially important. In this paper, we propose a new spectral conjugate gradient method based on the Dai-Yuan conjugate gradient method by providing a new spectral parameter. Our purpose is to obtain an efficient algorithm for the unconstrained optimization.

An unconstrained optimization problem is customarily expressed as

$$\min_{x\in\mathscr{R}^n} \quad f(x). \tag{1}$$

The nonlinear function $f : \mathscr{R}^n \to \mathscr{R}$ considered in this paper is continuously differentiable; the gradient of f is denoted by $g(x) := \nabla f(x)$. We usually impose the following properties on function f.

(**P1**) The function f is bounded below and is continuously differentiable in a neighbourhood $\mathscr{N}$ of the level set $\mathscr{L} := \{x \in \mathscr{R}^n \mid f(x) \le f(x_0)\}$, where x_0 is the starting point.

(**P2**) The gradient $g(x)$ of f is Lipschitz continuous in $\mathscr{N}$; that is, there exists a constant $L > 0$, such that $\|g(x) - g(y)\| \le L\|x - y\|$ for all $x, y \in \mathscr{N}$.

Generally, a sequence $\{x_k\}$ is obtained in an algorithm for solving (1) and has the following format:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where d_k is a search direction and α_k is the step size. At each iterative point x_k, we usually determine d_k firstly and then compute α_k by some principles.

There are different ways to determine the direction d_k. In the classical steepest-descent method, $d_k = -g_k$. In the conjugate gradient (CG) method, d_k is of the form

$$\begin{aligned} d_0 &= -g_0, \\ d_{k+1} &= -g_{k+1} + \beta_k d_k, \quad \text{if } k \ge 0, \end{aligned} \tag{3}$$

where β_k is a scalar parameter characterizing the conjugate gradient method. The best-known expressions of β_k are

Hestenes-Stiefel (HS) [1], Fletcher-Reeves (FR) [2], Polak-Ribiere-Polyak (PRP) [3, 4], and Dai-Yuan (DY) [5] formulas. They are defined by

$$\begin{aligned}\beta_k^{\mathrm{HS}} &= \frac{g_{k+1}^T y_k}{d_k^T y_k},\\ \beta_k^{\mathrm{FR}} &= \frac{\|g_{k+1}\|^2}{\|g_k\|^2},\\ \beta_k^{\mathrm{PR}} &= \frac{g_{k+1}^T y_k}{\|g_k\|^2},\\ \beta_k^{\mathrm{DY}} &= \frac{\|g_{k+1}\|^2}{d_k^T y_k},\end{aligned} \tag{4}$$

respectively, where $\|\cdot\|$ denotes the Euclidean norm and $y_k := g_{k+1} - g_k$.

There also are some approaches to determine the step size α_k in (2). Unfortunately, the steepest-descent method performs poorly. Barzilai and Borwein improved the steepest-descent method greatly by providing a spectral choice of step size in [6]. Their algorithm has the form

$$x_{k+1} = x_k - \alpha_k g_k, \tag{5}$$

where $\alpha_k = s_k^T s_k / s_k^T y_k$ or $\alpha_k = s_k^T y_k / y_k^T y_k$ with $s_k = x_{k+1} - x_k$. Many algorithms are proved convergent under the Wolfe condition; that is, the step size α_k satisfies

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta \alpha_k g_k^T d_k, \tag{6}$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \tag{7}$$

with $0 < \delta < \sigma < 1$.

In recent years, some scholars developed a new method—spectral conjugate gradient (SCG) method—for solving (1). For example, Raydan introduced the spectral gradient method for large-scale unconstrained optimization in [7]. He combined a nonmonotone line search strategy that guarantees global convergence with the Barzilai and Borwein method. Utilizing spectral gradient and conjugate gradient ideas, Birgin and Martinez proposed a spectral conjugate gradient method in [8]. In their algorithm, the search direction has the form

$$\begin{aligned} d_0 &= -g_0,\\ d_{k+1} &= -\theta_{k+1} g_{k+1} + \beta_k d_k, \quad \text{if } k \geq 0.\end{aligned} \tag{8}$$

In [8], the best combination of this formula, the scaling, and the initial choice of step-length is also studied. Following [8], some papers discussed the various choices of the spectral parameter θ_k based on different β_k. For example, Du and Chen [9] gave a modified spectral FR conjugate gradient method with Wolfe-type line search based on FR formula. Their spectral parameters θ_k and β_k are expressed as

$$\begin{aligned}\theta_{k+1} &= \frac{d_k^T y_k}{\|g_k\|^2},\\ \beta_k &= \beta_k^{\mathrm{FR}}.\end{aligned} \tag{9}$$

Yu et al. [10] presented a modification of spectral Perry's conjugate gradient formula, which possessed the sufficient descent property independent of line search condition. Their search direction d_{k+1} is defined by (8) and β_k has the form

$$\beta_k^{\mathrm{DSP}} = \beta_k^{\mathrm{SP}} - \frac{C\|y_k - \delta_{k+1} s_k\|^2}{\delta_{k+1}(y_k^T d_k)^2} g_{k+1}^T d_k, \tag{10}$$

where

$$\begin{aligned} C &\geq \frac{1}{4},\\ \beta_k^{\mathrm{SP}} &= \frac{g_{k+1}^T (y_k - \delta_{k+1} s_k)}{\delta_{k+1} y_k^T d_k},\\ \delta_{k+1} &= \frac{1}{\theta_{k+1}} = \frac{s_k^T y_k}{s_k^T s_k},\\ s_k &= x_{k+1} - x_k.\end{aligned} \tag{11}$$

Liu and Li [11] proposed a spectral DY-type projection method for nonlinear monotone systems of equations. The direction d_k is also determined by (8) and the parameters are defined by

$$\begin{aligned}\theta_{k+1} &= \frac{s_k^T s_k}{s_k^T y_k},\\ \beta_k &= \beta_k^{\mathrm{DY}}.\end{aligned} \tag{12}$$

We will propose a new SCG method based on the Dai-Yuan-type conjugate gradient method in this paper. A new selection of θ_{k+1} is introduced in our algorithm such that the sufficient descent condition holds. In addition, the global convergence of the new method is obtained.

The present paper is organized as follows. In Section 2, we outline our new method for unconstrained nonlinear optimization, and we show that the sufficient descent condition holds under mild assumptions. The global convergence is proved in Section 3, while the numerical results compared with CG-DESCENT are given in Section 4. At last section, we draw some conclusions about our new spectral conjugate gradient method.

2. Spectral Dai-Yuan-Type Conjugate Gradient Method

In this paper, we consider the spectral conjugate gradient method in which the search direction is of the form

$$\begin{aligned} d_0 &= -g_0,\\ d_{k+1} &= -\theta_{k+1} g_{k+1} + \beta_k^{\mathrm{DY}} d_k, \quad \text{if } k \geq 0,\end{aligned} \tag{13}$$

where

$$\theta_{k+1} = \max\left\{\frac{2\left|g_k^T d_k\right|}{\left|d_k^T y_k\right|}, \frac{2 g_{k+1}^T d_k - g_k^T d_k}{d_k^T y_k}\right\} \tag{14}$$

and the scalar parameter β_k^{DY} is defined by (4). This is a new spectral conjugate gradient method for solving problem (1) because the expression (14) of spectral parameter θ_{k+1} is completely different from those in other papers. The search direction (13) is a combination of the spectral gradient and the Dai-Yuan conjugate gradient. We hope that (14) may be an efficient choice.

In order to obtain the global convergence of our method, we assume that the step size satisfies the strong Wolfe condition; that is, the step size α_k satisfies (6) and

$$\left| g\left(x_k + \alpha_k d_k\right)^T d_k \right| \le -\sigma g_k^T d_k \tag{15}$$

with $0 < \delta < \sigma < 1$. It is easy to see that (7) holds if (15) holds.

The following is a detailed description of the spectral Dai-Yuan-type conjugate gradient (SDYCG) algorithm.

SDYCG Algorithm

Step 1 (initialization). Choose $x_0 \in \mathscr{R}^n$, set $k = 0$, and take $\epsilon > 0$.

Step 2 (check the convergence condition). Compute g_k; if $\|g_k\| \le \epsilon$, then stop.

Step 3 (form the search direction). If $k = 0$, then $d_0 = -g_0$. Else, compute θ_{k+1} and β_k^{DY} by formulas (14) and (4), respectively; then compute d_k by (13) and (14).

Step 4 (line search). Find α_k which satisfies the strong Wolfe conditions (6) and (15).

Step 5 (compute the new point). Set $x_{k+1} = x_k + \alpha_k d_k$ and $k = k + 1$ and go to Step 2.

The framework of the SDYCG algorithm is similar to other spectral conjugate gradient algorithms. However we choose a different spectral parameter θ_{k+1} (see (14)) which is the main difference between SDYCG and the others.

The global convergence of SDYCG algorithm will be given in the next section. Before that, we will show that the search direction (13) can ensure the sufficient descent condition.

Lemma 1. *Suppose that the sequence $\{d_k\}$ is generated by the SDYCG algorithm; then*

$$g_k^T d_k \le -\|g_k\|^2 \tag{16}$$

for all $k \ge 0$.

Proof. Since d_k is calculated by formula (13), we can get, if $k = 0$, that $g_0^T d_0 = -\|g_0\|^2$ whereas when $k \ge 0$, from (14), we find

$$\begin{aligned} \theta_{k+1} &\ge \frac{2g_{k+1}^T d_k - g_k^T d_k}{d_k^T y_k} = \frac{g_{k+1}^T d_k}{d_k^T y_k} + 1 \\ &= \frac{\|g_{k+1}\|^2}{d_k^T y_k} \cdot \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2} + 1 = \beta_k^{\mathrm{DY}} \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2} + 1. \end{aligned} \tag{17}$$

Furthermore,

$$-\theta_{k+1}\|g_{k+1}\|^2 + \beta_k^{\mathrm{DY}} g_{k+1}^T d_k \le -\|g_{k+1}\|^2. \tag{18}$$

From the second formula of (13), we obtain

$$g_{k+1}^T d_{k+1} \le -\|g_{k+1}\|^2. \tag{19}$$

The proof is completed. □

This lemma gives the fact that the direction d_k is a descent direction. Besides, d_k possesses the following property.

Lemma 2. *Suppose that the SDYCG algorithm is implemented with the step size α_k that satisfies the strong Wolfe conditions (6) and (15). If $g_k \neq 0$ for all $k \ge 0$, then*

$$\frac{\left(\beta_k^{\mathrm{DY}}\right)^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \le \frac{1}{\left(g_k^T d_k\right)^2}. \tag{20}$$

Proof. By (14) and $0 < \sigma < 1$, we have

$$\begin{aligned} \theta_{k+1} &\ge (\sigma+1)\frac{\left|g_k^T d_k\right|}{\left|d_k^T y_k\right|} = (\sigma+1)\frac{\|g_{k+1}\|^2}{\left|d_k^T y_k\right|} \cdot \frac{\left|g_k^T d_k\right|}{\|g_{k+1}\|^2} \\ &= (\sigma+1)\frac{\left|\beta_k^{\mathrm{DY}} g_k^T d_k\right|}{\|g_{k+1}\|^2}. \end{aligned} \tag{21}$$

With (15) and (21), we get

$$\begin{aligned} \left|\beta_k^{\mathrm{DY}} g_k^T d_k\right| &\le \theta_k \|g_{k+1}\|^2 - \sigma\left|\beta_k^{\mathrm{DY}} g_k^T d_k\right| \\ &\le \theta_k \|g_{k+1}\|^2 - \left|\beta_k^{\mathrm{DY}} g_{k+1}^T d_k\right| \\ &\le \left|-\theta_k \|g_{k+1}\|^2 + \beta_k^{\mathrm{DY}} g_{k+1}^T d_k\right| \\ &= \left|g_{k+1}^T d_{k+1}\right|. \end{aligned} \tag{22}$$

Therefore, inequality (20) follows. This completes the proof. □

Inequality (20) gives the close relationship of the inner product of gradient and direction between the adjacent two iterations. It will play an important role in our global convergence analysis.

3. Convergence Analysis

Dai and Yuan stated in [5] that the following result had been essentially proved by Zoutendijk and Wolfe.

Lemma 3. *Suppose that the function $f(x)$ has the properties* (**P1**) *and* (**P2**). *Assume that d_k is a descent direction and α_k is obtained by the Wolfe conditions (6) and (7). Then*

$$\sum_{k\ge 0} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < +\infty. \tag{23}$$

One customarily calls (23) the Zoutendijk condition.

Theorem 4. *Suppose that the function $f(x)$ has the properties* (**P1**) *and* (**P2**). *Sequences $\{x_k\}$, $\{g_k\}$, and $\{d_k\}$ are generated by SDYCG algorithm. Then either $g_k = 0$ for some k or*

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{24}$$

Proof. Suppose that $g_k \neq 0$ for all k and (24) is not true. Then there exists a constant $c > 0$, such that

$$\|g_k\| \geq c \tag{25}$$

for all k of the iterations.

The second equality of (13) implies

$$d_{k+1} + \theta_{k+1} g_{k+1} = \beta_k^{\mathrm{DY}} d_k; \tag{26}$$

we get

$$\begin{aligned}\|d_{k+1}\|^2 = {} & \left(\beta_k^{\mathrm{DY}}\right)^2 \|d_k\|^2 - 2\theta_{k+1} g_{k+1}^T d_{k+1} \\ & - \theta_{k+1}^2 \|g_{k+1}\|^2 .\end{aligned} \tag{27}$$

Dividing both sides by $g_{k+1}^T d_{k+1}$, we have

$$\begin{aligned}\frac{\|d_{k+1}\|^2}{\left(g_{k+1}^T d_{k+1}\right)^2} = {} & \frac{\left(\beta_k^{\mathrm{DY}}\right)^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \|d_k\|^2 - \frac{2\theta_{k+1}}{g_{k+1}^T d_{k+1}} \\ & - \frac{\theta_{k+1}^2 \|g_{k+1}\|^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \\ = {} & \frac{\left(\beta_k^{\mathrm{DY}}\right)^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \|d_k\|^2 \\ & - \left(\frac{1}{\|g_{k+1}\|} + \frac{\theta_{k+1} \|g_{k+1}\|}{g_{k+1}^T d_{k+1}}\right)^2 \\ & + \frac{1}{\|g_{k+1}\|^2}.\end{aligned} \tag{28}$$

Combining (20) in Lemma 2, we see the inequality

$$\frac{\|d_{k+1}\|^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \leq \frac{\|d_k\|^2}{\left(g_k^T d_k\right)^2} + \frac{1}{\|g_{k+1}\|^2}. \tag{29}$$

Summing both sides, we obtain

$$\sum_{i=0}^{k} \frac{\|d_{i+1}\|^2}{\left(g_{i+1}^T d_{i+1}\right)^2} \leq \sum_{i=0}^{k} \frac{\|d_i\|^2}{\left(g_i^T d_i\right)^2} + \sum_{i=0}^{k} \frac{1}{\|g_{i+1}\|^2}. \tag{30}$$

So, from (25) and (30), we get

$$\frac{\|d_{k+1}\|^2}{\left(g_{k+1}^T d_{k+1}\right)^2} \leq \sum_{i=0}^{k+1} \frac{1}{\|g_i\|^2} \leq \frac{k+2}{c}. \tag{31}$$

This relation is equivalent to

$$\frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \geq \frac{c}{k+2}. \tag{32}$$

Summing over k, we obtain

$$\sum_{k\geq 0} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} = +\infty. \tag{33}$$

From the SDYCG algorithm, the step size satisfies the strong Wolfe condition, so the Wolfe condition (7) holds. And the directions obtained by the algorithm are descent from Lemma 1. But the last equality contradicts the Zoutendijk condition (23). Hence, our original assertion (25) must be false, giving that either $g_k = 0$ for some k or (24) holds. □

4. Numerical Results

In order to test the numerical performance of the SDYCG algorithm, we choose some unconstrained problems with the initial points from CUTEr library [12, 13]. They are listed in Table 1.

The experiments are run on a personal computer with a 64-bit processor, 2.5 GHz of CPU, and 4 GB of RAM memory. All the codes are written in MATLAB language and are compiled with this software.

We would like to compare the SDYCG with the CG-DESCENT. The CG-DESCENT is a conjugate gradient algorithm with guaranteed descent proposed by Hager and Zhang in [14]. It has been proven an excellent algorithm in recent years.

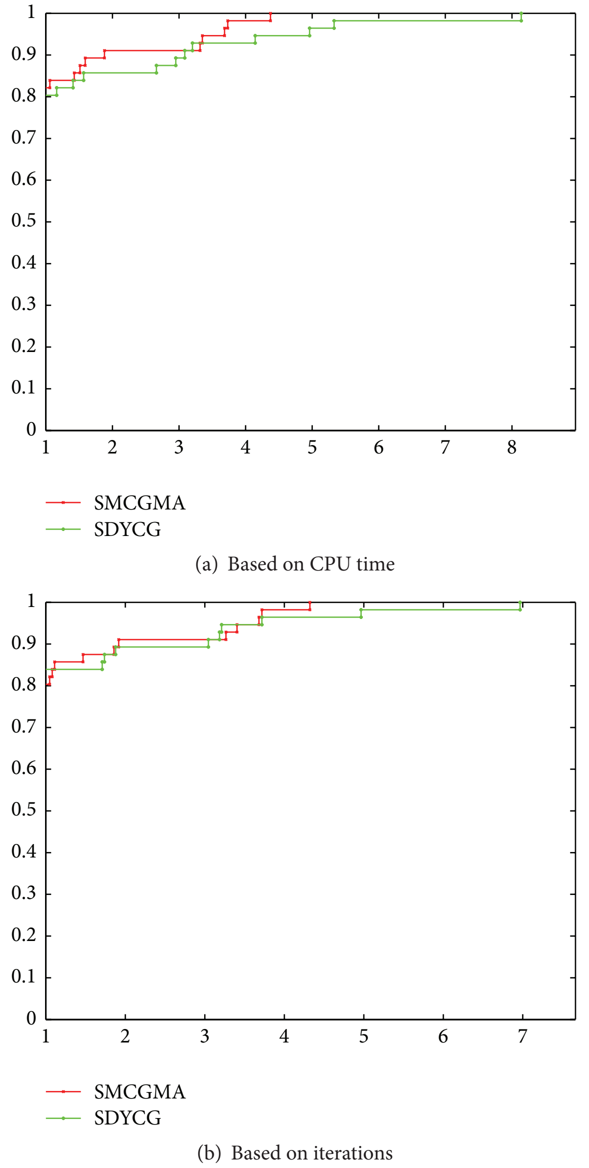
To make the comparison as fair as possible, we use the criterion $\|g\| \leq 10^{-5}$ to terminate the executions and impose restriction on the number of iterations less than 500 in both algorithms. All the step sizes satisfy the strong Wolfe conditions (6) and (15).

We use the performance profiles proposed by Dolan and Moré [15] to show the efficiency of comparisons. Performance profiles can be used as a tool for benchmarking and comparing optimization software. The performance profile for a solver is the (cumulative) distribution function for a performance metric. For example, if computing time is chosen as a metric, then we compute the ratio of the computing time of the solver versus the best time of all of the solvers. That is, for each method, we plot the fraction $P(y$-axis) of problems for which the method is within a factor $\tau(x$-axis) of the best time. The curve of a solver being above others means that it has the highest probability of being the optimal solver. We use a $\log_2$ scale for τ to capture the performance of all the solvers.

In order to observe the numerical results of the SDYCG and the CG-DESCENT, we choose three different dimensions of each test function. The dimensions are $n = 10^2$, $n = 10^3$, and $n = 10^4$, respectively. According to the numerical results obtained in every dimension, we plot two figures based on CPU time and iterations, respectively.

Table 1: Test problems.

Number	Function name
1	ENGVAL1
2	FLETCBV2
3	TOINTGSS
4	COSINE
5	ARWHEAD
6	EDENSCH
7	EG2
8	GENROSE
9	LIARWHD
10	Generalized White & Holst
11	Extended Wood
12	Extended quadratic penalty QP1
13	BDEXP
14	HIMMELBG
15	Hager
16	Extended TET
17	Diagonal 5
18	Extended Himmelblau
19	Diagonal 6
20	Extended DENSCHNF
21	LIARWHD
22	Extended BD1
23	Extended Hiebert
24	Extended Tridiagonal 2
25	QUARTC
26	Extended DENSCHNB
27	Extended Rosenbrock
28	Raydan 2
29	Diagonal 2
30	Diagonal 4
31	Extended Maratos
32	Quadratic QF1
33	Extended quadratic exponential EP1
34	DQDRTIC
35	NONSCOMP
36	Extended Freudenstein & Roth
37	Extended White & Holst
38	Raydan 1
39	Extended Tridiagonal 1
40	Extended Cliff
41	Extended Trigonometric
42	Extended Beale
43	Generalized Tridiagonal 1
44	Generalized PSC1
45	Extended PSC1
46	Extended Powell
47	BDQRTIC
48	FLETCBV3
49	FLETCHCR
50	FREUROTH
51	GENHUMPS
52	NONDIA
53	NONDQUAR
54	SROSENBR
55	TQUARTIC
56	Extended Penalty

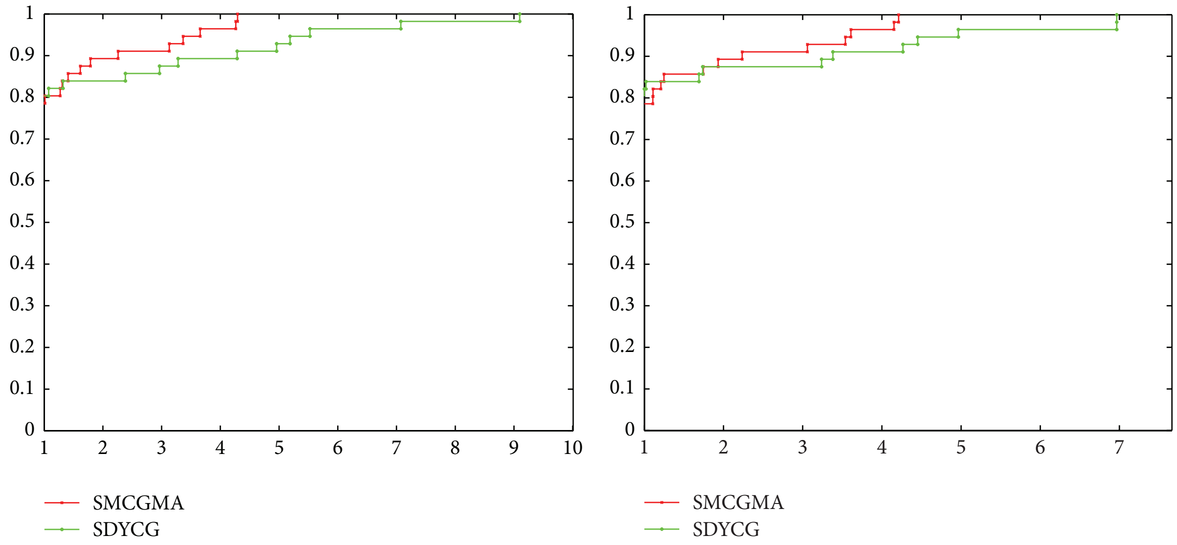


(a) Based on CPU time

(b) Based on iterations

Figure 1: Performance profiles in a $\log_2$ scale ($n = 10^2$).

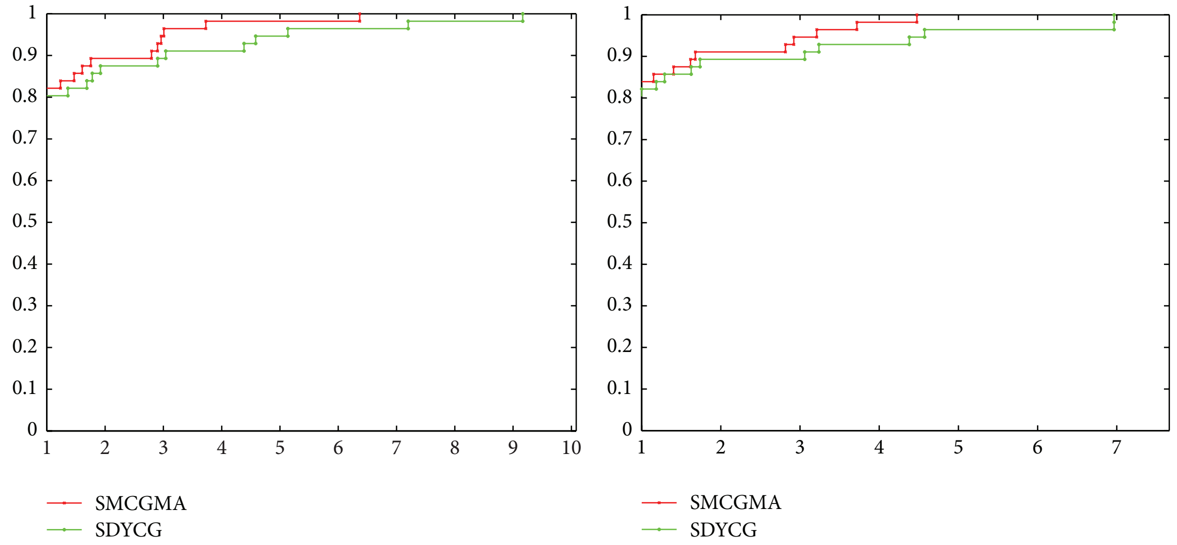
We can find from Figure 1 that the SDYCG is similar to the CG-DESCENT when $n = 10^2$ because their curves crosses each other. The predominance of the SDYCG appears in Figure 2 when $n = 10^3$. If the test dimension is chosen as $n = 10^4$, the SDYCG is better than the CG-DESCENT from the fact that its curve is almost completely above that of the CG-DESCENT in Figure 3.



(a) Based on CPU time

(b) Based on iterations

Figure 2: Performance profiles in a $\log_2$ scale ($n = 10^3$).



(a) Based on CPU time

(b) Based on iterations

Figure 3: Performance profiles in a $\log_2$ scale ($n = 10^4$).

Furthermore, we are interested in the robustness of our SDYCG algorithm. Ten problems are selected from CUTEr to be tested. The numerical results listed in Table 2 are obtained by changing the initial iteration point every time. "x_0" represents the standard initial iteration point of the problem; "iter." and "time(s)" indicate the iterative number and time (in seconds).

The conclusion that can be drawn is that the SDYCG is a robust algorithm and it may be capable of solving large-scale nonlinear unconstrained optimization problem.

5. Conclusions

We propose a new spectral conjugate gradient method for nonlinear unconstrained optimization. This method, which we call the SDYCG, is built based on the Dai-Yuan conjugate gradient method. A new spectral choice is provided in the search direction. Numerical results show that the SDYCG is comparable with the CG-DESCENT. The SDYCG algorithm may be capable of solving large-scale nonlinear unconstrained optimization problems.

Table 2: Test of the robustness of SDYCG.

Number	Function name	x_0		$10 * x_0$		$100 * x_0$	
		Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)
1	DQDRTIC	245	0.3436	290	0.3951	338	0.4567
2	QUARTC	1	0.0012	28	0.1273	27	0.1482
3	Diagonal 6	5	0.0176	8	0.0267	4	0.0153
4	Extended DENSCHNB	13	0.0110	18	0.0159	16	0.0177
5	Extended DENSCHNF	22	0.0636	17	0.0519	24	0.0820
6	LIARWHD	12	0.0186	11	0.0193	17	0.0320
7	EDENSCH	30	0.1897	30	0.18753	30	0.1860
8	EG2	9	0.0404	9	0.0392	9	0.0404
9	ENGVAL1	28	0.0893	29	0.1064	31	0.1294
10	FLETCBV2	53	0.1080	53	0.1377	53	0.1355

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (11471159), the Natural Science Foundation of Jiangsu Province (BK20141409), and the Foundation of Education Department of Anhui Province (2014jyxm161).

References

[1] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436 (1953), 1952.

[2] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[3] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française d'Informatique et de Recherche Opérationnelle Série Rouge*, vol. 3, pp. 35–43, 1969.

[4] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[5] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[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] 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.

[8] E. G. Birgin and J. M. Martinez, "A spectral conjugate gradient method for unconstrained optimization," *Applied Mathematics and Optimization*, vol. 43, no. 2, pp. 117–128, 2001.

[9] S.-Q. Du and Y.-Y. Chen, "Global convergence of a modified spectral FR conjugate gradient method," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 766–770, 2008.

[10] G. Yu, L. Guan, and W. Chen, "Spectral conjugate gradient methods with sufficient descent property for large-scale unconstrained optimization," *Optimization Methods & Software*, vol. 23, no. 2, pp. 275–293, 2008.

[11] J. Liu and S. Li, "Spectral DY-type projection method for nonlinear monotone systems of equations," *Journal of Computational Mathematics*, vol. 33, no. 4, pp. 341–355, 2015.

[12] N. I. Gould, D. Orban, and P. L. Toint, "CUTEr and SifDec: a constrained and unconstrained testing environment, revisited," *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 373–394, 2003.

[13] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.

[14] 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.

[15] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization