RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2014.40.295

ScienceAsia 40 (2014): **295–300**

New three-term conjugate gradient method for solving unconstrained optimization problems

Jinkui Liu[a,*], **Xuesha Wu**[b]

[a] Key Laboratory for Nonlinear Science and System Structure, Chongqing Three Gorges University, Wanzhou 404100 China
[b] College of General Education, Chongqing College of Electronic Engineering, Chongqing 401331 China

*Corresponding author, e-mail: liujinkui2006@126.com

Received 14 Jan 2013
Accepted 23 Mar 2014

ABSTRACT: Based on the Davidson-Fletcher-Powell (DFP) method which is a quasi-Newton method, an effective three-term conjugate gradient method is constructed to solve large-scale unconstrained optimization problems. The method possesses two attractive properties: (i) the famous Dai-Liao conjugate condition is satisfied and is independent of any line search; (ii) the sufficient descent property always holds without any line search. The convergence analysis is established under the general Wolfe line search. Numerical results show that the new method is effective and robust by comparing with the SPRP, PRP, and CG-DESCENT methods for the given test problems.

KEYWORDS: general Wolfe line search, Dai-Laio conjugate condition, sufficient descent property, global convergence

INTRODUCTION

In this paper, we consider the unconstrained optimization problem

$$\min f(x), \tag{1}$$

$x \in \mathbb{R}^n$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a real-valued, continuously differentiable function. Due to low memory requirements and strong local or global convergence properties, the conjugate gradient method is usually used to solve the problem (1). From an initial guess $x_1 \in \mathbb{R}^n$, it generates a sequence $\{x_k\}$ using the recurrence

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

where the step-size α_k is obtained by a line search, and the direction d_k is generated by

$$d_1 = -g_1, \quad d_{k+1} = -g_{k+1} + \beta_k d_k, \tag{3}$$

for $k \geqslant 1$. Here $g_k = \nabla f(x_k)$, and β_k is known as the conjugate gradient parameter. Well-known instances of β_k are from Polak-Ribière-Polyak (PRP), Hestenes-Stiefel (HS), and Hager-Zhang (HZ). They are defined as follows:

$$\beta_k^{\mathrm{PRP}} = \frac{g_{k+1}^{\mathrm{T}} y_k}{\|g_k\|^2}, \quad \beta_k^{\mathrm{HS}} = \frac{g_{k+1}^{\mathrm{T}} y_k}{d_k^{\mathrm{T}} y_k},$$

$$\beta_k^{\mathrm{HZ}} = \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^{\mathrm{T}} y_k}\right)^{\mathrm{T}} \frac{g_{k+1}}{d_k^{\mathrm{T}} y_k}.$$

The symbol $\|\cdot\|$ denotes the Euclidean norm and $y_k = g_{k+1} - g_k$. The corresponding methods are called the PRP[1,2], HS[3], and HZ[4] methods. If f is a strictly convex quadratic function, these methods are equivalent in the case that an exact line search is used. If f is non-convex, their types of behaviour may be distinctly different. In the past two decades, the convergence properties of the PRP and HS methods have been intensively studied by many researchers. Recently, Hager and Zhang[4] proved the global convergence of the HZ method for strong convex functions with the Wolfe line search and Goldstein line search. In order to prove global convergence of the HZ method for a general function, Hager and Zhang proposed the following modified parameter:

$$\beta_k^{\mathrm{MHZ}} = \max\{\beta_k^{\mathrm{HZ}}, \eta_k\},$$

where $\eta_k = -1/(\|d_{k-1}\| \min\{\eta, \|g_{k-1}\|\})$ and $\eta > 0$. The corresponding method is usually called the CG-DESCENT method, which is one of the most effective methods.

The three-term conjugate gradient method is another important computational innovation to solve the problem (1) which was first introduced by Beale. The iterative scheme[5] satisfies

$$d_{k+1} = -g_{k+1} + \beta_k d_k + \gamma_k d_t,$$

$$\gamma_k = \begin{cases} 0, & t = k+1, \\ \frac{g_{k+1}^{\mathrm{T}} y_t}{d_t^{\mathrm{T}} y_t}, & k > t+1, \end{cases}$$

where d_t is a restart direction. Subsequently, McGuire and Wolfe[6] and Powell[7] did further research into the Beale three-term conjugate gradient method and built an efficient restart strategy and obtained good numerical results. Nazareth[8] proposed another new three-term conjugate gradient method using the recurrence

$$d_{k+1} = -y_k + \frac{y_k^{\mathrm{T}} y_k}{y_k^{\mathrm{T}} d_k} d_k + \frac{y_{k-1}^{\mathrm{T}} y_k}{y_{k-1}^{\mathrm{T}} d_{k-1}} d_{k-1},$$

where $d_{-1} = 0$, $d_0 = 0$. When f is a convex quadratic function, the restart directions are conjugate subject to the Hessian of f without any line search. In the same content, Zhang, Zhou, and Li[9] proposed a modified PRP method with three-term structure:

$$d_k = -g_k + \frac{g_k^{\mathrm{T}} y_{k-1}}{\|g_{k-1}\|^2} d_{k-1} - \frac{g_k^{\mathrm{T}} d_{k-1}}{\|g_{k-1}\|^2} y_{k-1}.$$

For simplicity, we call it the SPRP method. A remarkable property is that its search direction d_k satisfies $g_k^{\mathrm{T}} d_k = -\|g_k\|^2$ for any k. Most recently, Andrei[10,11] analysed three-term conjugate gradient methods and proposed some new three-term conjugate gradient methods which have some good properties.

In this paper, we still focus on the three-term conjugate gradient method. Based on the DFP method[12], we further propose a new three-term conjugate gradient method to solve unconstrained optimization problems. Under some suitable conditions, the global convergence of the proposed method is established.

This paper is organized as follows. In the next section, we propose our algorithm and prove its sufficient descent property. Then global convergence analysis is provided with the general Wolfe line search. Finally, we perform numerical experiments by using a set of large problems, and do some numerical comparisons with the SPRP, PRP, and CG-DESCENT methods.

THE NEW METHOD AND ITS SUFFICIENT DESCENT PROPERTY

Firstly, we recall the DFP method[12]. The direction d_{k+1} of the DFP method is defined as $d_{k+1} = -H_{k+1} g_{k+1}$ where H_{k+1} is a positive definite matrix and is obtained by the DFP formula

$$H_{k+1} = H_k + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} - \frac{H_k y_k y_k^{\mathrm{T}} H_k}{y_k^{\mathrm{T}} H_k y_k},$$

where $s_k = x_{k+1} - x_k$. If $H_k \equiv I$, the matrix H_{k+1} is symmetric and satisfies the quasi-Newton equation, i.e., $H_{k+1} y_k = s_k$. Then d_{k+1} can also be written as

$$d_{k+1} = -g_{k+1} - \beta_k s_k + \delta_k y_k, \tag{4}$$

where $\beta_k = s_k^{\mathrm{T}} g_{k+1} / s_k^{\mathrm{T}} y_k$, $\delta_k = y_k^{\mathrm{T}} g_{k+1} / y_k^{\mathrm{T}} y_k$. From (4), we have $d_{k+1}^{\mathrm{T}} y_k = -g_{k+1}^{\mathrm{T}} s_k$ without any line search which satisfies the D-L conjugate condition[13], i.e., $d_{k+1}^{\mathrm{T}} y_k \leqslant -c g_{k+1}^{\mathrm{T}} s_k (c > 0)$. If the line search is exact, i.e., $g_{k+1}^{\mathrm{T}} s_k = 0$, then we have the well-known conjugate condition $d_{k+1}^{\mathrm{T}} y_k = 0$. This superior property motivates us to construct the following new three-term conjugate gradient method.

Algorithm 1

Step 1: Data: $x_1 \in \mathbb{R}^n$, $\varepsilon \geqslant 0$. Set $d_1 = -g_1$. If $\|g_1\| \leqslant \varepsilon$, stop.

Step 2: Compute α_k by the general Wolfe line search $(0 < \delta < \sigma_1 < 1, \sigma_2 \geqslant 0)$:

$$f(x_k + \alpha_k d_k) \leqslant f(x_k) + \delta \alpha_k g_k^{\mathrm{T}} d_k, \tag{5}$$

$$\sigma_1 g_k^{\mathrm{T}} d_k \leqslant g(x_k + \alpha_k d_k)^{\mathrm{T}} d_k \leqslant -\sigma_2 g_k^{\mathrm{T}} d_k. \tag{6}$$

Step 3: Update x_{k+1} by (2). If $\|g_{k+1}\| \leqslant \varepsilon$, stop.

Step 4: d_{k+1} is computed by (4).

Step 5: Set $k = k + 1$. Go to Step 2.

Lemma 1 *Let the sequences* $\{d_k\}$ *and* $\{g_k\}$ *be generated by Algorithm 1. Then there exists a constant* $u \in (0, 1)$ *such that*

$$g_k^{\mathrm{T}} d_k \leqslant -u \|g_k\|^2 \ \forall k \geqslant 1. \tag{7}$$

Proof: This conclusion can be proved by induction. Obviously, (7) holds for $k = 1$. Now we assume that (7) is true for $k > 1$. Then $g_k^{\mathrm{T}} d_k < 0$. By (6), we have

$$d_k^{\mathrm{T}} y_k \geqslant d_k^{\mathrm{T}} (g_{k+1} - g_k) \geqslant (\sigma_1 - 1) g_k^{\mathrm{T}} d_k \rangle 0.$$

Multiplying (4) by g_{k+1}^{T}, we obtain

$$g_{k+1}^{\mathrm{T}} d_{k+1} = -\|g_{k+1}\|^2 - \alpha_k \frac{(d_k^{\mathrm{T}} g_{k+1})^2}{d_k^{\mathrm{T}} y_k} + \frac{(y_k^{\mathrm{T}} g_{k+1})^2}{y_k^{\mathrm{T}} y_k}$$

$$\leqslant -\|g_{k+1}\|^2 + \frac{(y_k^{\mathrm{T}} g_{k+1})^2}{y_k^{\mathrm{T}} y_k}. \tag{8}$$

Using the inner product,

$$y_k^{\mathrm{T}} g_{k+1} = \|y_k\| \|g_{k+1}\| \cos(y_k, g_{k+1})$$

where $\cos(a, b)$ is the cosine of the angle between a and b. Obviously, if $\cos(y_k, g_{k+1}) = 0$, we have

$$g_{k+1}^{\mathrm{T}} d_{k+1} \leqslant -\|g_{k+1}\|^2.$$

If $\cos(y_k, g_{k+1}) \neq 0$, then there always exists a constant $u \in (0, 1)$ such that

$$\cos^2(y_k, g_{k+1}) \leqslant 1 - u. \tag{9}$$

Hence from (8) and (9), we show that

$$g_{k+1}^{\mathrm{T}} d_{k+1} \leqslant -\|g_{k+1}\|^2 + \|g_{k+1}\|^2 \cos^2(y_k, g_{k+1}).$$

Clearly, (7) holds. □

GLOBAL CONVERGENCE

In this section, we establish the global convergence of Algorithm 1. In what follows, we assume that f satisfies the following assumptions:

(i) the level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leqslant f(x_1)\}$ is bounded, i.e., there exists a positive constant $B > 0$ such that $\|x\| \leqslant B$, for $\forall x \in \Omega$.

(ii) in a neighbourhood V of Ω, f is continuously differentiable, and its gradient g is Lipchitz continuous, i.e., there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leqslant L \|x - y\| \quad \forall x, y \in V. \tag{10}$$

These assumptions imply that there is a positive constant $\gamma > 0$ such that

$$\|g(x)\| \leqslant \gamma, \quad \forall x \in \Omega. \tag{11}$$

The conclusion of the following lemma, often called the Zoutendijk condition, is used to prove the global convergence of conjugate gradient method. It was originally given by Zoutendijk[14] with the Wolfe line search. When $\sigma_2 = +\infty$, the general Wolfe line search reduces to the Wolfe line search. Since the general Wolfe line search is a special case of the Wolfe line search, we also obtain the Zoutendijk condition with the general Wolfe line search.

Lemma 2 *Suppose assumptions* (i) *and* (ii) *hold. Consider any iteration* (2) *and* (4)*, where d_k satisfies $g_k^{\mathrm{T}} d_k < 0$ for $k \in N^+$ and α_k satisfies the general Wolfe line search. Then we obtain*

$$\sum_{k \geqslant 1} (g_k^{\mathrm{T}} d_k)^2 / \|d_k\|^2 < \infty. \tag{12}$$

Lemma 3 *Suppose assumptions* (i) *and* (ii) *hold. Let the sequences $\{d_k\}$ and $\{g_k\}$ be generated by Algorithm 1. Then there exists a positive constant $M > 0$ such that*

$$\|d_k\| \leqslant M \quad \forall k \geqslant 1. \tag{13}$$

Proof: From (6), we get

$$d_k^{\mathrm{T}} y_k \geqslant -c d_k^{\mathrm{T}} g_k, \quad \left|d_k^{\mathrm{T}} g_{k+1}\right| \leqslant \sigma \left|d_k^{\mathrm{T}} g_k\right|,$$

where $c = 1 - \sigma_1$, and $\sigma = \min\{\sigma_1, \sigma_2\}$.

Using (4), the Cauchy inequality, assumptions (i) and (ii), and the above inequalities, we have

$$\begin{aligned} \|d_{k+1}\| &\leqslant \|g_{k+1}\| + \left|\frac{s_k^{\mathrm{T}} g_{k+1}}{d_k^{\mathrm{T}} y_k}\right| \|d_k\| + \left|\frac{y_k^{\mathrm{T}} g_{k+1}}{y_k^{\mathrm{T}} y_k}\right| \|y_k\| \\ &\leqslant \|g_{k+1}\| + \frac{\alpha_k \sigma \left|d_k^{\mathrm{T}} g_k\right|}{c \left|d_k^{\mathrm{T}} g_k\right|} \cdot \|d_k\| + \frac{\|y_k\|^2 \cdot \|g_{k+1}\|}{\|y_k\|^2} \\ &= 2\|g_{k+1}\| + \frac{\sigma \|x_{k+1} - x_k\|}{c} \leqslant 2\gamma + \frac{2\sigma B}{c}. \end{aligned}$$

Since $\|d_1\| \leqslant \|g_1\| \leqslant \gamma$, and letting $M = 2\gamma + 2\sigma B/(1 - \sigma_1)$, it is not difficult to show that (13) holds. □

Theorem 1 *Suppose that the assumptions* (i) *and* (ii) *hold. Let the sequences $\{d_k\}$ and $\{g_k\}$ be generated by Algorithm 1. Then we obtain*

$$\liminf_{k \to +\infty} \|g_k\| = 0. \tag{14}$$

Proof: Suppose that (14) does not hold, i.e., there exists a constant $r > 0$ such that

$$\|g_k\| \geqslant r, \quad \forall k \geqslant 1. \tag{15}$$

From (12), (7), and (15), we show that

$$u^2 r^4 \sum_{k \geqslant 1} 1/\|d_k\|^2 \leqslant u^2 \sum_{k \geqslant 1} \|g_k\|^4 / \|d_k\|^2 < +\infty,$$

which contradicts (13). Hence (14) holds. □

NUMERICAL EXPERIMENTS

In this section, we compare the performance of the new method with those of the PRP, SPRP, and CG-DESCENT methods with the general Wolfe line search. The test problems are from the unconstrained optimization problems in the CUTE library[15] along with other large-scale unconstrained optimization problems[16]. We selected 31 large-scale problems in extended or generalized forms. The parameters values are: $\eta = 0.01$, $\sigma_1 = 0.1$, and $\sigma_2 = 0.01$. The iterations were terminated when $\|g_k\|_2 \leqslant 10^{-6}$. If this condition was not satisfied after 5000 iterations, we declare failure.

All numerical results are listed in Tables 1 and 2. All codes were written in FORTRAN 90 and run on a PC with 2.0 GHz CPU and 512 MB memory and Windows XP operating system.

We adopt the performance profiles of Dolan and Moré[17] to compare the new method with the PRP, SPRP, and CG-DESCENT methods in the performances of CPU time, respectively. However, some CPU times are zero. Hence we take the average value of the CPU time for methods 1 and 2 and denote these by $\bar{C}_1$ and $\bar{C}_2$, respectively. The final CPU time for problem i using method j is given by

$$\mathrm{FCPU}_{i,j} = C_{i,j} + \tfrac{1}{2}(\bar{C}_1 + \bar{C}_2)$$

where $C_{i,j}$ denotes the CPU time of the ith test problem using method j. Then we apply the Dolan and Moré[17] technique to compare these methods. The performance profiles with respect to CPU time

Table 1 Numerical results of the new, SPRP, PRP, and CG-DESCENT methods.

Problem	Dim	New	SPRP	PRP	CG-DESCENT
Extended Freudenstein and Roth	5000	7/0.02	1424/3.75	14/0.00	1245/8.11
	10 000	7/0.01	238/1.22	29/0.21	13/0.01
Extended Rosenbrock	5000	27/0.03	29/0.02	31/0.01	27/0.02
	10 000	27/0.03	28/0.04	26/0.05	23/0.05
Extended White and Holst	5000	32/0.03	27/0.02	36/0.03	25/0.01
	10 000	31/0.05	26/0.03	27/0.05	27/0.05
Extended Beale	5000	10/0.02	13/0.03	15/0.01	15/0.01
	10 000	10/0.01	11/0.02	16/0.03	15/0.03
Extended Penalty	5000	259/1.56	10/0.01	189/1.19	75/0.41
	10 000	10/0.02	36/0.13	21/0.08	16/0.03
Raydan 2	5000	4/0.02	4/0.02	4/0.02	4/0.02
	10 000	4/0.03	4/0.03	4/0.03	4/0.01
Extended Tridiagonal 1	5000	16/0.02	11/0.02	16/0.03	21/0.02
	10 000	12/0.01	7/0.01	13/0.05	24/0.03
Extended Three Expo Terms	5000	10/0.07	7/0.02	8/0.11	8/0.05
	10 000	10/0.11	7/0.10	8/0.23	9/0.15
Diagonal 4	5000	4/0.00	4/0.00	4/0.00	4/0.00
	10 000	4/0.00	7/0.02	4/0.02	4/0.02
Diagonal 5	5000	4/0.05	4/0.06	4/0.08	4/0.03
	10 000	4/0.06	4/0.08	4/0.15	4/0.06

Dim: dimension of the test problem. The detailed numerical results are listed in the form NI/CPU, whee NI and CPU denote the number of iterations and CPU time in seconds, respectively. New: the proposed method in this paper; PRP: the famous PRP method [1,2]; SPRP: the three-term conjugate gradient method proposed by Zhang, Zhou, and Li [9]; CG-DESCENT: the most popular conjugate gradient method proposed by Hager and Zhang [4].

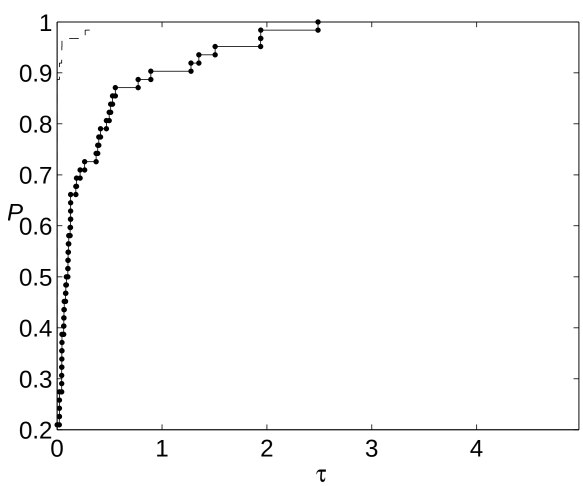

Fig. 1 CPU time of methods. Dashed line: new method; line with dots: PRP method.

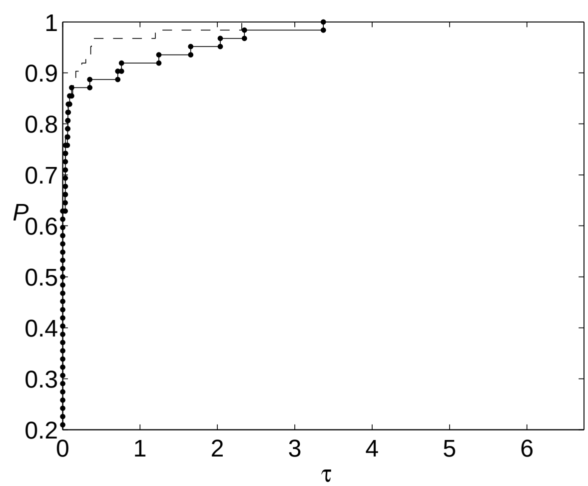

Fig. 2 CPU time of methods. Dashed line: new method; line with dots: SPRP method.

means that for each method, we plot the fraction P of problems for which the method is within a factor τ of the best time. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time.

For CPU time, Fig. 1 shows that the new method performs much better than the PRP method. Fig. 2 and Fig. 3 indicate that the new method is comparable with the SPRP and CG-DESCENT methods. Furthermore, from the tables, the new method also has some

Table 2 Numerical results of the new, SPRP, PRP, and CG-DESCENT methods.

Problem	Dim	New	SPRP	PRP	CG-DESCENT
Extended Himmelblau	5000	9/0.00	7/0.00	24/0.05	9/0.02
	10 000	9/0.02	7/0.02	25/0.08	9/0.01
Extended PSC1	5000	9/0.04	7/0.02	11/0.07	9/0.03
	10 000	9/0.06	7/0.05	27/1.99	134/12.19
Extended Block-Diagonal BD1	5000	25/0.05	36/0.07	25/0.09	23/0.05
	10 000	27/0.11	34/0.45	36/0.41	30/0.11
Extended Maratos	5000	47/0.03	49/0.05	59/0.09	49/0.03
	10 000	49/0.08	56/0.10	115/0.39	47/0.08
Extended Cliff	5000	19/0.04	54/1.76	58/3.08	56/1.65
	10 000	20/0.11	25/0.78	29/0.42	37/1.52
Extended Hiebert	5000	50/0.04	60/0.04	57/0.09	58/0.04
	10 000	54/0.08	59/0.11	61/0.22	57/0.08
Extended quadratic penalty QP2	5000	10/0.03	33/0.04	33/0.28	34/0.12
	10 000	11/0.10	35/0.14	38/0.61	34/0.29
Extended EP1	5000	3/0.00	3/0.00	3/0.02	3/0.00
	10 000	34/1.09	90/4.25	83/5.95	104/3.62
Extended Tridiagonal 2	5000	251/1.64	305/2.93	306/5.15	82/0.45
	10 000	362/5.60	215/4.27	480/17.06	334/5.41
ARWHEAD	5000	8/0.04	28/0.05	33/0.35	6/0.02
	10 000	5/0.03	6/0.04	15/0.23	5/0.01
NONDIA	5000	21/0.02	10/0.01	11/0.03	9/0.00
	10 000	20/0.05	8/0.01	10/0.03	9/0.02
DIXMAANA	5000	8/0.05	8/0.04	9/0.04	8/0.09
	10 000	7/0.03	8/0.04	9/0.08	8/0.03
DIXMAANB	5000	17/0.03	12/0.02	13/0.05	13/0.04
	10 000	17/0.05	12/0.05	13/0.11	13/0.04
DIXMAANC	5000	19/0.05	15/0.03	16/0.06	16/0.03
	10 000	19/0.06	15/0.03	15/0.11	16/0.07
EDENSCH	5000	115/0.79	54/0.13	109/2.00	95/0.76
	10 000	113/1.25	36/1.11	62/1.80	76/1.14
LIARWHD	5000	14/0.01	22/0.01	16/0.03	17/0.01
	10 000	12/0.02	14/0.02	17/0.08	18/0.05
Diagonal 6	5000	4/0.00	4/0.00	4/0.03	4/0.01
	10 000	4/0.03	4/0.03	4/0.06	4/0.02
ENGVAL1	5000	272/1.91	195/1.55	175/2.75	219/1.67
	10 000	257/3.75	149/2.81	290/9.97	674/11.15
COSINE	5000	11/0.05	11/0.04	11/0.10	12/0.04
	10 000	19/0.11	12/0.10	13/0.21	11/0.08
ENSCHNB	5000	7/0.01	6/0.01	8/0.01	6/0.00
	10 000	7/0.02	5/0.02	8/0.03	6/0.01
ENSCHNF	5000	30/0.03	24/0.02	24/0.05	25/0.03
	10 000	50/0.37	28/0.23	25/0.18	21/0.05

advantages in the number of iterations for some test problems. Thus the numerical results show that the proposed method is encouraging.

Acknowledgements: The authors express their heartfelt thanks to the referees and editor for their detailed and helpful suggestions for revising the manuscript. This study was supported by the Fund of Scientific Research in Southeast University (the support project of fundamental research) and Chongqing Three Gorges University (grant number: 13QN17).

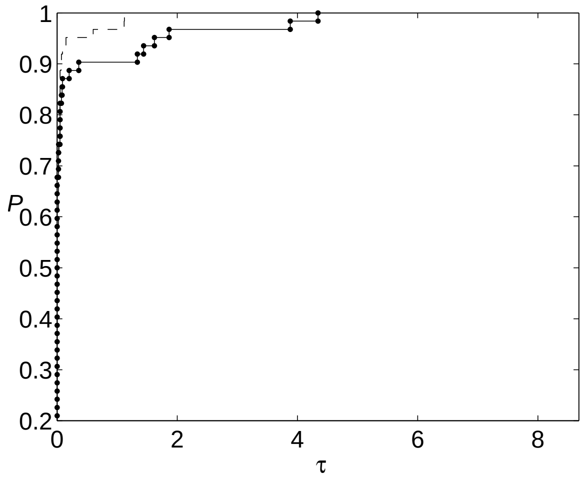

Fig. 3 CPU time of methods. Dashed line: new method; line with dots: CG-DESCENT method.

REFERENCES

1. Polak E, Ribière G (1969) Note sur la convergence de méthodes de directions conjugées. *Rev Fr Informat Rech Opér* **16**, 35–43.
2. Polyak BT (1969) The conjugate gradient method in extreme problems. *USSR Comput Math Math Phys* **9**, 94–112.
3. Hestenes MR, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. *J Res Natl Bur Stand* **49**, 409–36.
4. Hager WW, Zhang H (2005) A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J Optim* **16**, 170–92.
5. Beale EML (1972) A derivative of conjugate gradients. In: Lootsma FA (ed) *Numerical Methods for Nonlinear Optimization*, Academic Press, London, pp 39–43.
6. McGuire MF, Wolfe P (1973) Evaluating a restart procedure for conjugate gradients. Report RC-4382, IBM Research Center, Yorktown Heights.
7. Powell MJD (1984) Nonconvex minimization calculations and the conjugate gradient method. In: Griffiths DF (ed) *Lecture Notes in Mathematics*, vol 1066, 122–41.
8. Nazareth L (1977) A conjugate direction algorithm without line searches. *J Optim Theor Appl* **23**, 373–87.
9. Zhang L, Zhou W, Li DH (2006) A descent modified Polak-Ribière-Polyak conjugate gradient method and its global convergence. *IMA J Numer Anal* **26**, 629–40.
10. Andrei N (2013) On three-term conjugate gradient algorithms for unconstrained optimization. *Appl Math Comput* **219**, 6316–27.
11. Andrei N (2013) A simple three-term conjugate gradient algorithm for unconstrained optimization. *J Comput Appl Math* **241**, 19–29.
12. Davidon WC (1959) Variable metric method for minimization. AEC Research and Development Report ANL-5990(Rev.).
13. Dai YH, Liao LZ (2001) New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl Math Optim* **43**, 87–101.
14. Zoutendijk G (1970) Nonlinear programming, computational methods. In: Abadie J (ed) *Integer and Nonlinear Programming*, North-Holland, Amsterdam, pp 37–86.
15. Bongartz I, Conn AR, Gould N, Toint PL (1995) CUTE: constrained and unconstrained testing environment. *ACM Trans Math Software* **1**, 123–60.
16. Andrei N (2008) An unconstrained optimization test functions collection. *Adv Model Optim* **10**, 147–61.
17. Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. *Math Program* **91**, 201–13.