Journal of Computational Mathematics, Vol.21, No.3, 2003, 311–320.

TESTING DIFFERENT CONJUGATE GRADIENT METHODS FOR LARGE-SCALE UNCONSTRAINED OPTIMIZATION*[1)]

Yu-hong Dai
(*LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*)

Qin Ni
(*College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*)

Abstract

In this paper we test different conjugate gradient (CG) methods for solving large-scale unconstrained optimization problems. The methods are divided in two groups: the first group includes five basic CG methods and the second five hybrid CG methods. A collection of medium-scale and large-scale test problems are drawn from a standard code of test problems, CUTE. The conjugate gradient methods are ranked according to the numerical results. Some remarks are given.

Key words: Conjugate gradient methods, Large-scale, Unconstrained optimization, Numerical tests.

1. Introduction

We consider the unconstrained optimization problem

$$\min f(x), \quad x \in \mathcal{R}^n, \tag{1}$$

where f is smooth and its gradient g is available. The line search method for solving (1) is of the form

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

where x_1 is a given initial point, d_k is a search direction, and α_k is a stepsize obtained by a 1-dimensional line search. In the steepest descent method [4], the search direction is defined as the negative gradient direction,

$$d_k = -g_k, \tag{3}$$

and the stepsize is chosen to be the 1-dimensional minimizer

$$\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha_k d_k). \tag{4}$$

In practical computations, however, the steepest descent method performs poorly, and is badly affected by ill-conditioning [2]. Another class of methods are quasi-Newton methods (see [23] for example), where

$$d_k = -B_k g_k, \tag{5}$$

and where $B_k \in R^{n\times n}$ is updated at each iteration to capture the already-obtained second derivative information. They are very efficient for medium-scale problems, but can not be used to solve large-scale problems because of its storage of matrices. The conjugate gradient (CG)

* Received May 8, 2000.
[1)] Research partially supported by Chinese NSF grants 19801033, 19771047 and 10171104.

method [13] uses the negative gradient direction and the previous search direction to form the current search direction, namely,

$$d_k = -g_k + \beta_k d_{k-1}, \tag{6}$$

where $d_1 = -g_1$ and β_k is a scalar. In the case when f is a strictly convex quadratic

$$f(x) = \frac{1}{2} x^T A x + b^T x, \tag{7}$$

and α_k is obtained via an exact line search (4), the search directions generated by the CG method are conjugate to one another. As a result, the method gives the least value of (7) in at most n iterations. The CG method was extended by Fletcher and Reeves [11] to solve general nonconvex unconstrained optimization problem (1). Since it only requires storage of several vectors and is more rapid than the steepest descent method, the introduction of nonlinear CG method by Fletcher and Reeves marks the beginning of the field of large scale unconstrained optimization. Although the recent development of limited memory and discrete Newton methods have narrowed the class of problems for which CG methods are recommended, CG methods are still the best choice for solving very large problems with relatively inexpensive objective functions [16].

The purpose of this paper is to test and rank different nonlinear CG methods over a collection of standard test problems. As is known, for general nonconvex functions, there are many different choices for the scalar β_k in (6) and the properties of their corresponding CG methods may be very different. Another important reason is as follows. Usually, in the analyses and implementations of CG methods, the stepsize α_k is chosen by the strong Wolfe line search:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \tag{8}$$

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{9}$$

where $0 < \delta < \sigma < 1$. Recently, however, [8] proposed a new nonlinear CG method in which β_k is given by (15). The descent property and global convergence of the method can be shown provided that the stepsize is obtained by the weak Wolfe line search, namely, (8) and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \tag{10}$$

The hybrid methods related to this method are studied in [9], and the initial numerical results in [9] suggested an efficient hybrid CG algorithm that uses the weak Wolfe line search. Consequently, an overall assessment for the basic CG methods and hybrid CG methods is imperative to be done.

This paper is organized as follows. In the next section, we will give a description to the collection of test problems that are drawn from a standard code of test problems, CUTE. Other details of our numerical experiments are also provided in Section 2. In Section 3, we briefly review the five basic CG methods and report their numerical results. In Section 4, we briefly review the hybrid CG methods and report the numerical results of five hybrid CG methods. The numerical results made in Sections 3 and 4 show that the PRP, HS and DYHS2 are most efficient algorithms among all the tested CG algorithms. For the purpose of further comparisons, we draw in Section 5 some numerical results of the three efficient CG algorithms for difficult problems and listed them into a table. The table shows that one hybrid method, namely, DYHS2, outperforms the PRP and HS methods for difficult problems. Concluding remarks are given in the last section.

2. Preliminaries

Twenty-five sets of test problems are drawn from a standard code of test problems, CUTE [3]. A description of these test problems is given in Table 1, where "Name" denotes the name

Table 1. List of test problems

Problem	Name	Description	n
1	BRYBND	Broyden banded system	500, 5000
2	CRAGGLVY	Extended Cragg-Levy problem	100
3	DIXMAANA	Dixon-Maany problem (A)	300, 3000
4	DIXMAANB	Dixon-Maany problem (B)	6000
5	DIXMAANC	Dixon-Maany problem (C)	300, 3000
6	DIXMAAND	Dixon-Maany problem (D)	300, 1500
7	DIXMAANF	Dixon-Maany problem (F)	600, 3000
8	DIXMAANH	Dixon-Maany problem (H)	120, 1500
9	DIXMAANI	Dixon-Maany problem (I)	120
10	DIXMAANK	Dixon-Maany problem (K)	600
11	DQDRTIC	diagonal quadratic	500, 5000
12	DQRTIC	diagonal quartic	500, 5000
13	ENGVAL1	ENGVAL1 problem	100
14	FLETCBV2	Boundary Value problem	100
15	LIARWHD	simplified NONDIA problem	500, 2000
16	MANCINO	Mancino's function	100
17	NONDIA	nondiagonal extension of Rosenbrock	100, 1000
18	POWELLSG	Powell singular problem	100, 1000
19	POWER	Power problem by Oren	100, 1000
20	SROSENBR	separable extension of Rosenbrock	100, 1000
21	TOINTGSS	Toint's Gaussian problem	1000, 10000
22	TQUARTIC	special quartic function	100, 1000
23	TRIDIA	quadratic tridiagonal problem	100
24	VAREIGVL	variational eigenvalue	500
25	WOODS	extended Woods problem	100, 1000

of the test problem in CUTE, "Description" gives a simple description of the problem, and n is the number of variables. Because the purpose of this paper is to test different CG methods for large-scale problems, the value of n is at least set to 100. The largest value of n is set to 10000.

Our computations are carried out on an SGI Indigo R4000 XS workstation. All codes are written in FORTRAN with double precisions. For each CG method, either the strong Wolfe line search (8)–(9) or the weak Wolfe line search (8) and (10) is used. In either case, the values of δ and σ are set to 0.01 and 0.1, respectively. The initial trial value for the line search is determined according to the rule in [22]. More exactly, we set it to $1/\|g_1\|$ for the first iteration and $\alpha_{k-1}d_{k-1}^T g_{k-1}/d_k^T g_k$ for $k \geq 2$. For each test problem, the used termination condition is

$$\|g_k\|_2 \leq 10^{-6}. \tag{11}$$

In order to rank the CG methods, we compute the total number of function and gradient evaluations by the formula

$$N_{total} = N_f + m * N_g, \tag{12}$$

where N_f, N_g denote the number of function evaluations and gradient evaluations, respectively, and m is some integer. According to the results on automatic differentiation (see [12]), the value of m can be set to $m = 5$. That is to say, one gradient evaluation is equivalent to m number of function evaluations if by automatic differentiation. Here we should point out that, in the case when the technique of automatic differentiation is not used, such a way to compute N_{total} favors the algorithms that use the strong Wolfe line search, because in this case one gradient evaluation is equivalent to n number of functions evaluations, and the ratio N_g/N_f for the strong Wolfe line search is normally greater than the one for the weak Wolfe line search.

3. Testing Five Basic CG Methods

3.1. Five Basic CG Methods

As once mentioned in Section 1, different choices for the scalar β_k result in different nonlinear conjugate gradient methods. Some formulae for β_k are called as the Fletcher-Reeves [11], conjugate descent [10], Dai-Yuan [8], Polak-Ribière-Polyak [18, 19] and Hestenes-Stiefel [13] ones, and are given by

$$\beta_k^{FR} = \|g_k\|^2/\|g_{k-1}\|^2, \tag{13}$$

$$\beta_k^{CD} = -\|g_k\|^2/d_{k-1}^T g_{k-1}, \tag{14}$$

$$\beta_k^{DY} = \|g_k\|^2/d_{k-1}^T y_{k-1}, \tag{15}$$

$$\beta_k^{PRP} = g_k^T y_{k-1}/\|g_{k-1}\|^2, \tag{16}$$

$$\beta_k^{HS} = g_k^T y_{k-1}/d_{k-1}^T y_{k-1}, \tag{17}$$

when $\|\cdot\|$ means the 2-norm and $y_{k-1} = g_k - g_{k-1}$, respectively. Their corresponding CG methods are abbreviated as FR, CD, DY, PRP and HS methods. Although all these methods reduce to the linear CG method in the case when f is given by (7) and α_k is chosen by (4), their behaviors for general objective functions may be far different.

Assume that the strong Wolfe line search (8)–(9) is used. Then the FR method converges globally if the scalar in (9) is chosen such that $\sigma \le 0.5$ (see [1, 15, 6]). If $\sigma > 0.5$, the FR method may fail due to producing an uphill search direction [6]. Although for any $\sigma < 1$, the CD method using the strong Wolfe line search ensures a descent direction at each iteration, its global convergence can only be proved [7] when the stepsize α_k satisfies the line search conditions (8) and

$$\sigma g_k^T d_k \le g(x_k + \alpha_k^T d_k)^T d_k \le 0. \tag{18}$$

Unlike the FR and CD methods, the DY method can be proved to generate a descent direction at each iteration and converge globally provided that the weak Wolfe line search (8) and (10) is used [8].

Powell [20] analyzed a major numerical drawback of the FR method using exact line searches, namely, if a small step is generated away from the solution point, the subsequent steps may be also very short. These analyses are also efficient for the CD and DY methods since all the three methods are the same in case of exact line searches. In spite of this fact, we will test and rank the three methods so that one can have a glimpse into the behaviors of themselves and the methods related to them.

In the case when a small step occurs, the search direction generated by the PRP and HS methods will automatically be close to the negative gradient direction, thus avoiding the numerical drawback mentioned in the above paragraph. However, the PRP and HS methods using exact line searches may cycle near several non-stationary points, see the counter-example in Powell [21]. Gilbert and Nocedal [17] was able to establish the global convergence result of the PRP and HS methods by restricting the scalar β_k to be nonnegative and using a complicated line search. The resulting PRP and HS algorithms perform almost all the same as the original PRP and HS algorithms, respectively.

3.2. Numerical Results

In Tables 2 and 3 list the numerical results of five basic CG methods for medium-scale problems and large-scale problems, respectively. For each method, our line search subroutine computes a stepsize α_k for which (8)–(9) hold with $\delta = 0.01$ and $\sigma = 0.1$. In Tables 2 and 3, the numerical results are written in the form of $N_{iter}/N_f/N_g$, where N_{iter}, N_f and N_g denote the numbers of iterations, function evaluations and gradient evaluations, respectively.

Since the average performances of the PRP method are the best among the five basic CG methods, we compare the other four CG methods with the PRP method. For each of the other

Table 2. Test results on five basic CG methods (medium-scale problems)

P	n	FR	CD	DY	PRP	HS
1	500	40/131/50	40/131/55	39/128/54	39/129/58	39/129/60
2	100	111/222/201	116/237/215	104/225/211	106/206/190	118/281/263
3	300	7/15/9	8/17/11	7/15/9	7/15/9	7/15/9
5	300	8/20/12	8/20/12	8/20/12	8/21/13	8/23/15
6	300	9/32/15	9/32/15	9/32/15	11/40/21	9/33/16
7	600	114/214/204	5993/7957/7922	113/213/203	146/270/260	138/261/251
8	120	110/203/194	144/265/256	117/214/205	66/137/127	62/133/119
9	120	488/764/759	2669/3786/3781	396/621/614	678/1065/1055	596/906/900
11	500	51/115/83	36/88/56	19/51/31	16/46/25	33/80/51
12	500	22/92/65	20/79/50	20/88/60	21/71/45	18/58/34
13	100	25/58/32	25/87/53	25/87/56	25/96/60	26/69/42
14	100	222/408/408	348/676/662	168/299/299	296/433/433	183/275/275
15	500	45/122/80	66/162/109	21/67/40	16/49/28	12/35/21
16	100	11/56/12	12/61/13	12/61/13	11/56/12	11/56/12
17	100	18/56/32	31/92/58	30/86/54	9/30/13	14/60/38
18	100	2477/4825/4075	683/1309/1242	2538/4328/4294	274/631/487	142/318/238
19	100	57/123/108	59/127/115	58/126/114	38/89/60	37/84/57
20	100	24/71/45	25/74/46	29/84/54	9/33/19	8/30/18
21	1000	6/21/10	6/21/10	6/21/10	7/26/14	7/25/15
22	100	97/237/183	74/198/145	113/255/194	11/34/20	8/26/18
23	100	198/350/345	257/451/444	185/330/324	309/526/524	243/425/422
25	100	56/123/88	51/114/84	64/132/100	58/132/96	77/173/132

Table 3. Test results on five basic CG methods (large-scale problems)

Prob	n	FR	CD	DY	PRP	HS
1	1000	51/170/66	55/186/75	77/259/103	31/117/72	27/103/65
3	3000	8/28/14	8/28/14	8/28/14	6/23/11	7/25/13
4	6000	7/27/14	7/27/14	7/27/14	7/25/13	9/34/18
5	3000	6/23/11	6/23/11	6/23/11	6/21/10	6/21/10
6	1500	7/28/13	7/28/13	7/28/13	10/37/20	13/43/24
7	3000	223/459/456	227/463/462	220/449/446	197/402/399	196/403/399
8	1500	2616/5259/2693	3366/9989/6693	2658/5374/2740	171/341/338	218/443/438
10	600	1208/1617/1606	5343/6911/6900	959/1175/1165	3423/5172/5163	2379/2925/2917
11	5000	40/96/66	40/94/63	19/52/32	16/47/26	33/79/50
12	5000	29/164/121	32/193/140	29/174/124	33/179/114	30/150/96
15	2000	57/155/103	65/179/122	39/115/76	18/56/38	11/34/21
17	1000	14/54/33	11/42/22	11/42/22	9/29/14	12/53/31
18	1000	3741/7389/5943	904/1744/1530	4206/6933/6916	95/205/156	160/387/307
19	1000	182/331/305	210/430/418	194/397/386	135/268/254	133/265/252
20	1000	34/94/58	18/55/36	14/46/26	10/32/16	8/25/13
21	10000	4/18/8	4/18/8	4/18/8	5/26/15	4/22/13
22	1000	385/766/606	357/719/557	547/1078/864	11/36/24	10/36/22
24	500	273/441/432	841/1195/1187	256/407/399	289/468/459	278/439/429
25	1000	45/106/74	40/99/66	36/89/60	37/92/59	56/126/90

Table 4. Relative efficiency of five basic CG methods

PRP	HS	DY	FR	CD
1	1.01	1.13	1.18	1.36

four CG methods, we evaluate its efficiency with respect to the PRP method as follows: for each problem i, compute the total numbers of function evaluations and gradient evaluations required by the evaluated method and the PRP method by formula (12), and denote them by $N_{total,i}(EM)$ and $N_{total,i}(PRP)$; then calculate the ratio

$$r_i(EM) = \frac{N_{total,i}(EM)}{N_{total,i}(PRP)}, \tag{19}$$

and the geometric mean of these ratios over all the test problems:

$$r(EM) = (\prod_{i \in S} r_i(EM))^{1/|S|}, \tag{20}$$

where S denotes the set of the test problems and $|S|$ the number of elements in S. One advantage of the above rule is that, the comparison is relative and hence does not be dominated by a few problems for which the method requires a great deal of function evaluations and gradient functions.

According to the above rule, it is clear that $r(PRP) = 1$. The values of $r(FR)$, $r(CD)$, (DY) and $r(HS)$ are listed in Table 4. From Table 4, one can see that the HS method performs similarly to the PRP method, whereas the performances of the FR, CD and DY methods are relatively bad. Among the latter three methods, the DY method seems the best and the CD method the worst, as accords with the rank list of their convergence properties mentioned in §3.1. This can partly explain why the DYHS1 method outperforms the FRPRP1 method, as shown in Table 7, because in the case when

$$g_k^T g_{k-1} < 0, \tag{21}$$

the DYHS1 and FRPRP1 methods reduce to DY and FR, respectively.

4. Testing Five Hybrid CG Methods

4.1. Five Hybrid CG Methods

To combine the nice global convergence properties of the FR method and the good numerical performances of the PRP method, Hu and Storey [14] considered the hybrid method (2) and (6) with

$$\beta_k = \max\{0, \min\{\beta_k^{PRP}, \beta_k^{FR}\}\}. \tag{22}$$

Gilbert and Nocedal [17] further considered the method

$$\beta_k \in [-\beta_k^{FR}, \beta_k^{FR}]. \tag{23}$$

Both the methods (22) and (23) are globally convergent under the same condition as that required for the FR method [14, 17], whereas they have the advantage of avoiding the propensity of short steps. Along this line, Dai and Yuan [9] studied methods related the DY method. They proved that under the weak Wolfe line search, any method (2) and (6) with

$$\beta_k \in [-\frac{\sigma - 1}{1+\sigma}\beta_k^{DY}, \beta_k^{DY}] \tag{24}$$

produces a descent direction at every iteration and converges globally. Dai and Yuan [9] tested the following two hybrid methods of the DY method and the HS method:

$$\beta_k = \max\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\} \tag{25}$$

Table 5. Test results on five hybrid cg methods (medium-scale problems)

P	n	FRPRP1	FRPRP2	DYHS1	DYHS2	DYHS3
1	500	27/65/38	27/65/38	30/72/42	67/117/80	75/136/94
2	100	112/223/212	105/213/194	105/220/207	103/167/123	108/170/126
3	300	6/21/10	6/25/11	6/23/11	8/25/12	8/25/11
5	300	7/27/14	7/27/13	7/27/14	8/29/13	9/31/14
6	300	8/32/16	8/29/14	8/32/16	8/29/14	11/37/20
7	600	114/215/205	114/217/207	113/213/203	152/202/163	144/203/162
8	120	72/139/127	73/145/134	74/150/140	73/119/85	78/125/90
9	120	602/898/893	565/878/873	494/775/770	713/867/800	687/834/769
11	500	28/74/47	28/74/47	20/52/31	21/53/31	21/53/31
12	500	17/61/35	14/50/27	16/51/26	18/57/32	16/51/26
13	100	24/55/30	24/55/30	24/56/31	26/56/31	26/107/70
14	100	236/387/384	236/387/384	213/361/357	333/405/368	410/527/479
15	500	122/258/200	23/66/41	181/357/282	61/130/88	61/130/88
16	100	11/38/15	11/36/15	11/38/15	11/36/15	11/38/15
17	100	16/52/30	10/37/21	15/49/30	12/36/20	61/123/87
18	100	208/450/352	167/365/283	3975/6382/6373	239/437/340	206/370/290
19	100	58/126/116	58/124/109	58/126/114	47/93/61	42/80/54
20	100	25/73/46	25/73/46	24/71/44	23/62/39	31/78/49
21	1000	6/21/10	6/21/10	6/21/10	8/24/11	8/24/11
22	100	179/362/283	209/421/334	380/741/629	12/34/22	17/54/33
23	100	274/475/473	274/475/473	208/381/378	273/360/302	347/455/391
25	100	38/86/57	63/136/108	22/59/35	54/101/70	34/76/48

and

$$\beta_k = \max\{-\frac{\sigma - 1}{1+\sigma}\beta_k^{DY}, \min\{\beta_k^{HS}, \beta_k^{DY}\}\}. \tag{26}$$

In this paper, we will test the hybrid method (25) using the strong Wolfe line search or the weak Wolfe line search, and (26) using the weak Wolfe line search. The five hybrid CG methods to be tested are simply described as follows:

FRPRP1: (22) with the strong Wolfe line search;
FRPRP2: (23) with the strong Wolfe line search;
DYHS1: (25) with the strong Wolfe line search;
DYHS2: (25) with the weak Wolfe line search;
DYHS3: (26) with the weak Wolfe line search.

4.2. Numerical Results

In Tables 5 and 6 list the numerical results of five hybrid CG methods for medium-scale problems and large-scale problems, respectively. Note that the weak Wolfe line search is used for the DYHS2 and DYHS3 methods, whereas the strong Wolfe line search for the other three hybrid CG methods. Nevertheless, the parameters δ and σ are always set to $\delta = 0.01$ and $\sigma = 0.1$. As is the same as before, the numerical results are written in the form of $N_{iter}/N_f/N_g$, where N_{iter}, N_f and N_g denote the numbers of iterations, function evaluations and gradient evaluations, respectively.

Table 6. Test results on five hybrid CG methods (large-scale problems)

Prob	n	FRPRP1	FRPRP2	DYHS1	DYHS2	DYHS3
1	5000	31/74/45	31/74/45	47/105/75	47/103/64	34/81/50
3	3000	7/27/14	6/24/13	7/26/14	7/26/12	7/26/12
4	6000	7/27/13	7/30/16	7/27/13	9/30/14	9/30/14
5	3000	6/23/11	6/23/11	6/23/11	8/25/11	8/25/11
6	1500	7/28/13	6/25/12	7/28/13	12/36/18	11/34/17
7	3000	270/430/421	270/430/421	225/391/382	356/441/391	356/441/391
8	1500	173/309/298	169/303/292	155/288/277	208/282/237	218/290/247
10	600	2653/3445/3434	2477/3184/3172	1619/1950/1940	2941/3371/3246	2181/2537/2423
11	5000	29/76/48	29/76/48	20/53/32	21/54/32	21/54/32
12	5000	21/60/35	17/60/32	20/66/37	18/60/32	38/110/73
15	2000	39/112/77	22/62/43	117/268/200	25/69/45	43/110/73
17	1000	14/54/33	14/54/33	11/42/22	16/51/30	21/63/38
18	1000	714/1357/1249	714/1357/1249	4805/7726/7693	289/541/427	252/485/376
19	1000	177/323/290	183/333/304	194/398/386	169/254/201	163/246/197
20	1000	25/71/44	25/72/44	18/57/35	11/39/22	115/225/169
21	10000	4/18/8	4/18/8	4/18/8	4/18/8	4/18/8
22	1000	263/532/409	263/532/409	439/867/683	189/374/284	189/374/284
24	500	266/428/419	266/428/419	256/407/399	290/388/336	265/337/295
25	1000	76/156/120	84/162/125	74/158/126	37/72/47	40/85/54

Under the comparison rule in §3.2, we compare the relative performances of the five hybrid CG methods with the PRP method. See Table 7. From the Table, one can see that the performances of the DYHS2 method are the best; they are comparable to those of the PRP method. Note that the weak Wolfe line search is used in the DYHS2 method instead of the strong Wolfe line search. Thus it is now safe to say that efficient CG algorithms can all the same be designed based on the weak Wolfe line search, not necessarily the strong Wolfe line search. Another point we should point out here is that, the global convergence studies can also lead to efficient hybrid CG algorithms. In the next section, we will find that the DYHS2 method is superior to the PRP and HS methods for difficult problems.

Another hybrid method that uses the weak Wolfe line search is DYHS3. It ranks the second in Table 7. In addition, one can also see that the hybrid methods of the FR and PRP methods are worse than those of the DY and HS methods.

Table 7. Relative efficiency of five hybrid CG methods

DYHS2	DYHS3	FRPRP2	DYHS1	FRPRP1
1	1.09	1.10	1.13	1.16

5. Comparing PRP, HS and DYHS2 for Difficult Problems

From Tables 4 and 7, we can see that the PRP, HS and DYHS2 methods perform similarly for the given medium-scale and large-scale problems. For the purpose of further comparisons, we draw the numerical results of the three methods for the difficult problems from Tables 2-3 and 5-6 and list them into Table 8. Here we say that a problem is "difficult" if the number of function evaluations required by any of the PRP, HS and DYHS2 methods is greater than or equal to 100.

From Table 8, we see that the DYHS2 method outperforms the PRP and HS methods for most of the 18 difficult problems. Therefore comparing with the PRP and HS methods, the DYHS2 method are more efficient for difficult problems.

Table 8. Comparing PRP, HS and DYHS2

P	n	PRP	HS	DYHS2
1	500	39/129/58	39/129/60	67/117/80
1	1000	31/117/72	27/103/65	47/103/64
2	100	106/206/190	118/281/263	103/167/123
7	600	146/270/260	138/261/251	152/202/163
7	3000	197/402/399	196/403/399	356/441/391
8	120	66/137/127	62/133/119	73/119/85
8	1500	171/341/338	218/443/438	208/282/237
9	120	678/1065/1055	596/906/900	713/867/800
10	600	3423/5172/5163	2379/2925/2917	2181/2537/2423
12	5000	33/179/114	30/150/96	18/60/32
14	100	296/433/433	183/275/275	333/405/368
18	100	274/631/487	142/318/238	239/437/340
18	1000	95/205/156	160/387/307	289/541/427
19	1000	135/268/254	133/265/252	169/254/201
22	1000	11/36/24	10/36/22	189/374/284
23	100	309/526/524	243/425/422	273/360/302
24	500	289/468/459	278/439/429	290/388/336
25	100	58/132/96	77/173/132	54/101/70

6. Concluding Remarks

We have tested and ranked different nonlinear CG methods over a collection of standard test problems in CUTE. The methods can be divided into two groups: the first group includes five basic CG methods and the second five hybrid CG methods. From the numerical results, we can come to a conclusion that conjugate gradient methods are efficient for solving large-scale unconstrained optimization problems, where the PRP, HS and DYHS2 methods are most efficient. For difficult problems, however, the DYHS2 method outperforms the PRP and HS methods.

Since the weak Wolfe line search is used in the DYHS2 method instead of the strong Wolfe line search, it is safe to say that efficient CG algorithms can all the same be designed based on the weak Wolfe line search. In addition, although the global convergence studies in [14] failed to give a hybrid algorithm more efficient than the PRP method, we finally find an efficient CG algorithm, namely, DYHS2, along the line. Besides the superiority of the DYHS2 method over the PRP method for difficult problems, another advantage of the DYHS2 method is that, it is globally convergent for general nonconvex functions, as mentioned in §4.1, whereas the PRP method with exact line searches needs not converge [21].

However, we should see that for some problems (though small), the PRP and HS methods perform much better than the DYHS2. An illustrative example is Problem 22 with $n = 1000$, see Table 8. Therefore it still remains under study how to design a more efficient algorithm by combining the PRP and/or HS methods and the DYHS2 method.

References

[1] M. Al-Baali, Descent property and global convergence of the Fletcher-Reeves method with inexact line search, *IMA J. Numer. Anal.,* **5** (1985), 121-124.

[2] H. Akaike, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Statist. Math. Tokyo,* **11** (1959), 1-17.

[3] I. Bongartz, A.R. Conn, N. Gould, and Ph.L. Toint, CUTE: constrained and unconstrained testing environment, *ACM Transactions on Mathematical Software,* **21** (1995), 123-160.

[4] A. Cauchy, Méthode générale pour la résolution des systèms d'équations simultanées, Comp. Rend. Sci. Paris, **25** (1847), 536-538.

[5] A.R. Conn, N.I.M. Gould and Ph.L. Toint, Testing a class of methods for solving minimization problems with simple bounds on the variables, *Math. Comp.,* **50** (1988), 399-430.

[6] Y. H. Dai and Y. Yuan, Convergence properties of the Fletcher-Reeves method, *IMA J. Numer. Anal.,* **16**:2 (1996), 155-164.

[7] Y. H. Dai and Y. Yuan, Convergence properties of the conjugate descent method, *Advances in Mathematics,* **6** (1996), 552-562.

[8] Y. H. Dai and Y. Yuan, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM J. Optimization,* **10**:1 (1999), 177-182.

[9] Y. H. Dai and Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, *Annals of Operations Research,* **103** (2001), 33-47.

[10] R. Fletcher, Practical Methods of Optimization vol. 1: Unconstrained Optimization, John Wiley & Sons, New York, 1987.

[11] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, *Comput. J.,* **7** (1964), 149-154.

[12] A. Griewank, On automatic differentiation, In: Mathematical programming:Recent Developments and Applications, eds. Iri M, Tanabe K. Kluwer Academic Publishers, 1989, 83-108

[13] M. R. Hestenes and E. L. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards Sect.,* **5:**49 (1952), 409-436.

[14] Y. F. Hu and C. Storey, Global convergence result for conjugate gradient methods, *JOTA,* **71**:2 (1991), 399-405.

[15] G. H. Liu, J. Y. Han and H. X. Yin, Global convergence of the Fletcher-Reeves algorithm with an inexact line search, *Appl. Math. J. Chinese Univ. Ser. B,* **10** (1995), 75-82.

[16] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Mathematical Programming,* **15** (1989), 503-538.

[17] J. C. Gilbert and J. Nocedal, Global convergence properties of conjugate gradient methods for optimization, *SIAM. J. Optimization,* **2**:1 (1992), 21-42.

[18] E. Polak and G. Ribière, Note sur la convergence de directions conjugées, Rev. Francaise Informat Recherche Opertionelle, *3e Année,* **16** (1969), 35-43.

[19] B.T. Polyak, The conjugate gradient method in extremem problems, *USSR Comp. Math. and Math. Phys.,* **9** (1969), 94-112.

[20] M. J. D. Powell, Restart procedures for the conjugate gradient method, *Mathematical Programming,* **12** (1977), 241-254.

[21] M. J. D. Powell, Nonconvex minimization calculations and the conjugate gradient method, in: Lecture Notes in Mathematics vol. 1066, Springer-Verlag (Berlin) (1984), pp. 122-141.

[22] D. F. Shanno and K. H. Phua, Remark on algorithm 500 : Minimization of unconstrained multivariate functions, *ACM Trans. Math. Software,* **6** (1980), 618-622.

[23] Y. Yuan, Numerical Methods For nonlinear Programming, Shanghai Scientific and Technical Publishers, 1993 (in Chinese).