RAIRO-Oper. Res. 54 (2020) 981–991
https://doi.org/10.1051/ro/2019045

RAIRO Operations Research
www.rairo-ro.org

A RESTART SCHEME FOR THE DAI–LIAO CONJUGATE GRADIENT METHOD BY IGNORING A DIRECTION OF MAXIMUM MAGNIFICATION BY THE SEARCH DIRECTION MATRIX

Zohre Aminifard and Saman Babaie-Kafaki*

Abstract. As known, finding an effective restart procedure for the conjugate gradient methods has been considered as an open problem. Here, we aim to study the problem for the Dai–Liao conjugate gradient method. In this context, based on a singular value analysis conducted on the Dai–Liao search direction matrix, it is shown that when the gradient approximately lies in the direction of the maximum magnification by the matrix, the method may get into some computational errors as well as it may converge hardly. In such situation, ignoring the Dai–Liao search direction in the sense of performing a restart may enhance the numerical stability as well as may accelerate the convergence. Numerical results are reported; they demonstrate effectiveness of the suggested restart procedure in the sense of the Dolan–Moré performance profile.

Mathematics Subject Classification. 90C53, 65K05, 65F35.

Received August 28, 2018. Accepted April 20, 2019.

1. Introduction

Consider the unconstrained optimization problem

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

in which $f:\mathbb{R}^n\to\mathbb{R}$ is continuously differentiable and analytic expression of its gradient is available. As a class of efficient techniques for solving large-scale cases of (1.1), conjugate gradient (CG) methods are of particular performance due to low memory requirement and strong global convergence properties [13,26]. Iterative formula of the methods is in the following form:

$$x_0\in\mathbb{R}^n,\ x_{k+1}=x_k+s_k,\ s_k=\alpha_k d_k,\ k=0,1,\ldots, \tag{1.2}$$

where α_k is a step length to be computed by a line search along the direction d_k defined by

$$d_0=-g_0,\ d_{k+1}=-g_{k+1}+\beta_k d_k,\ k=0,1,\ldots, \tag{1.3}$$

in which $g_k=\nabla f(x_k)$ and β_k is a scalar called the CG (update) parameter [19].

Keywords. Nonlinear programming, unconstrained optimization, conjugate gradient method, restart strategy, maximum magnification.

Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, Semnan University, PO Box 35195-363, Semnan, Iran.
*Corresponding author: sbk@semnan.ac.ir

Employing features of the quasi-Newton methods [28], Dai and Liao [12] (DL) proposed one of the well-known CG parameters as follows:

$$\beta_k^{\mathrm{DL}} = \frac{g_{k+1}^T y_k}{d_k^T y_k} - t\frac{g_{k+1}^T s_k}{d_k^T y_k}, \tag{1.4}$$

where $y_k = g_{k+1} - g_k$ and t is a nonnegative parameter, being an extension of the Hestenes–Stiefel (HS) parameter [20]. It is worth noting that the DL method with

$$t_{k_1}^* = \vartheta \frac{||y_k||^2}{s_k^T y_k}, \tag{1.5}$$

where $\vartheta > \dfrac{1}{4}$ is a real constant and $||.||$ stands for the Euclidean norm, possesses the sufficient descent property, *i.e.*

$$g_k^T d_k \leq -\rho ||g_k||^2, \ \forall k \geq 0, \tag{1.6}$$

for some positive constant ρ [3, 11, 19]. Also, Babaie-Kafaki and Ghanbari [5] showed that the following family of choices for t in the DL method yields descent search directions:

$$t_k^{p,q} = p\frac{||y_k||^2}{s_k^T y_k} - q\frac{s_k^T y_k}{||s_k||^2}, \tag{1.7}$$

where $p > \dfrac{1}{4}$ and $q < \dfrac{1}{4}$ are real parameters. Moreover, in [7] they suggested another DL parameter satisfying (1.6) as follows:

$$t_{k_2}^* = \max\left\{2\frac{s_k^T y_k}{||s_k||^2}, \omega\frac{||y_k||^2}{s_k^T y_k}\right\}, \tag{1.8}$$

where $\omega > 1$ is a real constant.

A popular modification in the nonlinear CG techniques is to restart the iterations by the steepest descent direction. Restarting serves to periodically refresh the algorithm by erasing the old information that may not be beneficial. Crowder and Wolfe [10] gave an example to illustrate that without restarting the convergence rate of the CG methods can be only linear. Powell [25] showed that, under some circumstances, the Fletcher–Reeves [16] method with exact line search may produce very small displacements and normally not recover unless a restart is performed (see also [17]). The restart is important because the finite-termination property as well as the other appealing features of the CG methods hold only when the initial search direction d_0 to be equal to the negative gradient [23]. Also, since CG methods could generate uphill search directions, a regular restart along a descent direction has been considered as a widespread technique to enforce the global convergence and to achieve reasonable computational outputs (see *e.g.* [1, 4, 12, 14, 25]). It is notable that the most convenient restart procedure for the CG methods has been considered as an open problem [2].

Among the essential CG restart strategies proposed in the literature, initially Fletcher and Reeves [16] recommended to restart the CG method using the steepest descent direction every n (or $n+1$) iterations, periodically. However, it is worth noting that when frequently periodic restarts with the steepest descent direction are used, the reduction at the restart iteration is often poor compared with the reduction that would have occurred without restarting [26]. On the other hand, if the restart direction is taken as an arbitrary vector, the conjugacy condition may not hold. Motivated by these, Beale [9] dealt with a restart procedure for the HS method, developing the concept of three-term CG techniques. More exactly, if we consider d_j as an arbitrary descent restart direction at x_j, for each $k > j$ Beale [9] defined the following search direction:

$$d_{k+1} = -g_{k+1} + \beta_k^{\mathrm{HS}} d_k + \gamma_k d_j, \tag{1.9}$$

where

$$\beta_k^{\mathrm{HS}} = \frac{g_{k+1}^T y_k}{d_k^T y_k},$$

and γ_k is calculated in a way to guarantee the conjugacy condition; that is

$$\gamma_k = \frac{g_{k+1}^T y_j}{d_j^T y_j}\cdot$$

Afterwards, McGuire and Wolfe [22] showed that the Beale's restart approach is not so effective in the computational point of view. Especially, they showed that for general (nonquadratic) objective functions, when the Beale's restart technique is employed, the sequence $\{g_k\}_{k\geq 0}$ may converge to a nonzero vector. Also, the restart period that seems to be dependent on the objective function behavior (as discussed in [10, 25]) should not be only depended on the dimension of the problem [25]. Based on these facts, Powell [25] modified the Beale's approach in the sense of restarting the CG method with the search direction (1.9) when the following inequality is violated:

$$|g_{k+1}^T g_k| \leq c||g_{k+1}||^2, \tag{1.10}$$

where c is a small positive constant. This prevents the generated gradient vectors from tending to a nonzero limit and yields satisfactory numerical results [25]. Furthermore, for the projection methods in which search direction is updated by

$$d_{k+1} = -P_j g_{k+1} + \beta_k^{\text{HS}} d_k,$$

where j is the restart index and P_j is a positive definite matrix for which

$$d_j = -P_j g_j.$$

Powell [25] applied the following criterion as an extension of (1.10):

$$|(P_j g_k)^T P_j g_{k+1}| \leq c||P_j g_{k+1}||^2. \tag{1.11}$$

To avoid storing any matrix, being vital for large-scale problems, he considered a simple memoryless choice for P_j as

$$P_j = I - \frac{g_j g_j^T}{||g_j||^2}\cdot$$

Then, Dai *et al.* [14] employed the (memoryless) BFGS updating formula [26] for P_j and suggested the following revised version of (1.11):

$$|g_k^T P_j g_{k+1}| \leq c g_{k+1}^T P_j g_{k+1}.$$

Following the mentioned attempts, here we suggest another restart procedure for the DL method, based on a singular value analysis and using a scaled version of the steepest descent direction. This work is organized as follows. In Section 2, at first we carry out a singular value analysis on the matrix which generates the DL search direction and then, we deal with our restart scheme. In Section 3, we make some numerical experiments on a set of CUTEr test problems [18] to evaluate the effect of our restart strategy, using the Dolan–Moré performance profile [15]. Finally, we present some conclusions in Section 4.

2. A restart procedure for the Dai–Liao method

Here, based on the Perry's matrix point of view [24], firstly we note that search directions of the DL method can be written as

$$d_{k+1} = -Q_{k+1} g_{k+1}, \ k = 0, 1, \ldots, \tag{2.1}$$

in which

$$Q_{k+1} = I - \frac{s_k y_k^T}{s_k^T y_k} + t\frac{s_k s_k^T}{s_k^T y_k}, \tag{2.2}$$

is called the search direction matrix (see also [4–6]). Next, we compute all the singular values and a singular vector of Q_{k+1} which are necessary to explain our restart approach. Hereafter, we assume that $s_k^T y_k > 0$, as guaranteed by the Wolfe line search conditions [26], *i.e.*

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k, \tag{2.3}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{2.4}$$

with $0 < \delta < \sigma < 1$. The following preliminaries of linear algebra are also needed [27].

As known, the induced (Euclidean) matrix norm of an arbitrary matrix A is defined by

$$||A|| = \sup_{x \neq 0} \frac{||Ax||}{||x||} = \max_{x \in \Omega} ||Ax||,$$

where $\Omega = \{x \in \mathbb{R}^n : ||x|| = 1\}$. Since the norm function is continuous and Ω is a compact set, for some nonzero vector x we have

$$||Ax|| = ||A||\ ||x||. \tag{2.5}$$

Definition 2.1 ([27])**.** For an arbitrary matrix $A \in \mathbb{R}^{n\times m}$, the scalar

$$\mathrm{maxmag}(A) = \max_{x \neq 0} \frac{||Ax||}{||x||},$$

is called the maximum magnification by A.

As seen, $\mathrm{maxmag}(A) = ||A||$. Hence, since the (spectral) condition number of A is defined by $\kappa(A) = ||A||\ ||A^{-1}||$, it can be concluded that the smaller $\mathrm{maxmag}(A)$ and $\mathrm{maxmag}(A^{-1})$ are, the more well-conditioned matrix A gets and consequently, the more stable computations achieve. So, in the numerical point of view, it is reasonable to avoid the vectors that (approximately) lie in the direction of the maximum magnification by A; that is, the vectors satisfying (2.5) (see Example 2.2.15 of [27]).

Generally, we may encounter with large quantities in a numerical problem, challenging their storage as well as their computational manipulations. As an important error source, large numbers are often of great concern in practical computations. Among the essential error sources in scientific computing with large numbers there are the swamping of data and the overflowing; for more details see [21, 27].

Now, we deal with properties of the search direction matrix Q_{k+1} given by (2.2). Based on the singular value analysis carried out in [4], since $s_k^T y_k \neq 0$, there exists a set of mutually orthonormal vectors $\{u_k^i\}_{i=2}^{n-1}$ such that

$$s_k^T u_k^i = y_k^T u_k^i = 0,\ \ i = 2, 3, \ldots, n-1, \tag{2.6}$$

which leads to

$$Q_{k+1} u_k^i = Q_{k+1}^T u_k^i = u_k^i,\ \ i = 2, 3, \ldots, n-1.$$

That is, Q_{k+1} has $n-2$ singular values equal to 1. As shown in [4], the two remaining singular values of Q_{k+1}, namely σ_k^- and σ_k^+, are given by

$$\sigma_k^{\pm} = \frac{1}{2} \frac{\sqrt{(t||s_k||^2 + s_k^T y_k)^2 + ||s_k||^2||y_k||^2 - (s_k^T y_k)^2}}{s_k^T y_k} \pm \frac{1}{2} \frac{\sqrt{(t||s_k||^2 - s_k^T y_k)^2 + ||s_k||^2||y_k||^2 - (s_k^T y_k)^2}}{s_k^T y_k},$$

for which we have $\sigma_k^- \leq 1 \leq \sigma_k^+$. Now, we find the singular vector of Q_{k+1} corresponding to σ_k^+ which is necessary in our discussions since is the vector of the maximum magnification by Q_{k+1}.

Considering (2.6), it can be concluded that orthonormal vectors respect to σ_k^+ namely u_k^1 and v_k^1 (based on Thm. 4.1.3 of [27]) are linear combinations of s_k and y_k, *i.e.*

$$v_k^1 = \zeta_1 s_k + \zeta_2 y_k, \text{ and } u_k^1 = \xi_1 s_k + \xi_2 y_k, \tag{2.7}$$

where ξ_1, ξ_2, ζ_1 and ζ_2 are real scalars. So, since

$$Q_{k+1} v_k^1 = \sigma_k^+ u_k^1, \text{ and } Q_{k+1}^T u_k^1 = \sigma_k^+ v_k^1,$$

and also, from (2.7), after some algebraic manipulations we get

$$\begin{aligned}
\zeta_1 &= \left(-\frac{s_k^T y_k}{||s_k||^2} \sigma_k^+ - \sigma_k^+ t + \frac{t}{\sigma_k^+} \right) \xi_2, \\
\zeta_2 &= \sigma_k^+ \xi_2, \\
\xi_1 &= -\frac{s_k^T y_k}{||s_k||^2} (\sigma_k^+)^2 \xi_2.
\end{aligned}$$

Now, it just remains to determine ξ_2. In this context, because u_k^1 and v_k^1 are unit vectors, ξ_2 can be computed as follows:

$$\xi_2 = \pm \frac{1}{\sqrt{||y_k||^2 + \frac{(s_k^T y_k)^2}{||s_k||^2} (\sigma_k^+)^2 ((\sigma_k^+)^2 - 2)}}\cdot$$

So, the vector v_k^1 as the direction of maximum magnification by Q_{k+1} is determined.

Based on the above discussion, when g_{k+1} approximately lies in the direction of the maximum magnification by Q_{k+1}, from (2.1) we may have

$$||g_{k+1}|| \ll ||d_{k+1}||.$$

In this situation, because $||d_{k+1}||$ may be extremely large in contrast to $||g_{k+1}||$, computational errors may appear with a great probability. To speak with more details, when $||g_{k+2}|| \approx ||g_{k+1}||$ and the line search is approximately exact (as probable near the solution), from (1.3) we can write

$$d_{k+2} \approx -g_{k+2} + \beta_{k+1}^{\mathrm{HS}} d_{k+1},$$

and consequently, for not so small values of $\beta_{k+1}^{\mathrm{HS}}$, the vector g_{k+2} may be swamped by $\beta_{k+1}^{\mathrm{HS}} d_{k+1}$. Also, since $d_{k+2} \approx \beta_{k+1}^{\mathrm{HS}}\, d_{k+1}$, the method fails to generate a new search line. This phenomenon can be continued for the next iterations unless the method is restarted. In addition, we should take into account the probable overflowing error for d_{k+1} that can be considered as another troublesome of the algorithm. In what follows, we describe another motivation persuading us to restart the DL method when g_{k+1} lies in the direction of the maximum magnification by Q_{k+1}. In this context, the following preliminaries are needed [17].

Assumptions 2.2. *(i) The level set $\mathcal{L} = \{x\colon f(x) \le f(x_0)\}$ is bounded. (ii) In some neighborhood $\mathcal{N}$ of $\mathcal{L}$, the objective function f is continuously differentiable and its gradient is Lipschitz continuous; that is, there exists a positive constant L such that*

$$||g(x) - g(\tilde{x})|| \le L||x - \tilde{x}||, \ \forall x, \tilde{x} \in \mathcal{N}.$$

Note that these assumptions imply that there exists a constant $\bar{\gamma}$ such that

$$||g(x)|| \le \bar{\gamma}, \ \forall x \in \mathcal{L}.$$

As known, the angle θ_{k+1} between $-g_{k+1}$ and d_{k+1} is defined by

$$\cos\theta_{k+1} = -\frac{g_{k+1}^T d_{k+1}}{||g_{k+1}||\ ||d_{k+1}||}\cdot \tag{2.8}$$

If the exact line search is employed ($g_{k+1}^T d_k = 0$), then from (1.3) and (2.8) we have

$$\cos\theta_{k+1} = \frac{||g_{k+1}||}{||d_{k+1}||}. \tag{2.9}$$

The following important theorem is now immediate.

Theorem 2.3 ([29]). *Suppose that Assumptions 2.2 hold and consider any iteration of the form (1.2) where d_k is a descent direction and α_k satisfies the Wolfe conditions (2.3) and (2.4). Then*

$$\sum_{k\geq 0} \cos^2\theta_k ||g_k||^2 < \infty. \tag{2.10}$$

Now, substituting (2.9) in (2.10), we obtain

$$\sum_{k\geq 0} \frac{||g_k||^4}{||d_k||^2} < \infty. \tag{2.11}$$

So, if one can show that $\{||d_k||/||g_k||\}_{k\geq 0}$ is bounded, then (2.11) gives

$$\lim_{k\to\infty} g_k = 0.$$

Based on this fact, note that in the DL method when g_{k+1} is in the direction of the maximum magnification by Q_{k+1}, from (2.1) it may be hard for the sequence $\{||d_k||/||g_k||\}_{k\geq 0}$ to be bounded and consequently, convergence of $\{g_k\}_{k\geq 0}$ may be happen hardly. Moreover, if the DL method satisfies the sufficient descent condition (1.6), from (2.8) we get

$$\cos\theta_{k+1} \geq \rho \frac{||g_{k+1}||}{||d_{k+1}||}. \tag{2.12}$$

So, for inexact line searches relation (2.12) can be used instead of (2.9) to achieve (2.11), being necessary to conduct a similar analysis demonstrating the same probable undesirable effect in the convergence of the DL method. To overcome the mentioned defect, performing a restart can be helpful; it can accelerate the convergence of the DL method.

According to the above analysis, here if g_{k+1} is approximately parallel to v_k^1 (the direction of maximum magnification by Q_{k+1}) in the sense that

$$\left| \frac{|g_{k+1}^T v_k^1|}{||g_{k+1}||} - 1 \right| < \varepsilon, \tag{2.13}$$

where ε is a small positive constant, then we restart the DL method by the following effective scaled steepest descent direction [8]:

$$d_{k+1} = -\tau_k g_{k+1}, \tag{2.14}$$

in which

$$\tau_k = \frac{||s_k||^2}{s_k^T y_k},$$

is called the scaling parameter, computed based on a two-point approximation of the secant equation [26] in the sense of the following least-squares problem:

$$\min_{\tau\geq 0} ||\frac{1}{\tau} s_k - y_k||.$$

Table 1. Outputs (1).

		DL			RDL1-0.05			
Function	n	N_i	N_f	N_g	N_i	N_f	N_g	N_r
ARGLINA	200	2	3	3	2	3	3	0
ARGLINB	200	3	6	4	3	6	4	0
BDEXP	5000	2	3	3	2	3	3	0
BDQRTIC	5000	168	518	461	146	365	328	3
BIGGSB1	5000	9095	9113	9102	9095	9113	9102	0
BOX	10 000	8	19	14	8	19	14	0
BQPGABIM	50	48	60	49	48	60	49	0
BQPGASIM	50	48	60	49	48	60	49	0
BRYBND	5000	38	61	48	33	59	44	7
CHAINWOO	4000	548	573	558	478	513	494	1
CHENHARK	5000	139	150	150	139	150	150	0
CHNROSNB	50	548	588	566	555	594	570	2
CLPLATEB	5041	10 000	10 062	10 044	10 000	10 041	10 029	5
COSINE	10 000	4	11	7	4	11	7	1
CRAGGLVY	5000	65	76	68	67	77	68	5
CURLY10	10 000	46	70	63	46	70	63	0
CURLY20	10 000	101	125	118	99	123	116	1
CURLY30	10 000	150	175	168	155	180	173	1
DECONVU	63	1005	1056	1029	1005	1056	1029	0
DIXMAANB	3000	5	9	6	5	9	6	0
DIXMAANC	3000	6	11	7	6	11	7	0
DIXMAAND	3000	7	13	8	7	13	8	1
DIXMAANE	3000	382	398	386	334	346	338	1
DIXMAANF	3000	292	306	294	200	215	202	1
DIXMAANG	3000	284	299	287	261	276	265	1
DIXMAANI	3000	1490	1502	1493	994	1009	996	1
DIXMAANJ	3000	164	176	165	188	200	190	1
DIXMAANL	3000	162	174	164	150	162	153	1

3. Numerical experiments

Here, we present some numerical results obtained by applying MATLAB 7.7.0.471 (R2008b) implementations of the CG methods in the form of (1.2)–(1.3) with $\beta_k = \beta_k^{\mathrm{DL}}$ computed by (1.4) in which

- DL1: $t = t^*_{k_1}$ given by (1.5) with $\vartheta = 1$;
- DL2: $t = t^{p,q}_k$ given by (1.7) with $(p,q) = \left(\frac{1}{2}, -\frac{1}{2}\right)$;
- DL3: $t = t^*_{k_2}$ given by (1.8) with $\omega = 1.3$ as suggested in [7];
- RDL1–0.05: DL1 with restart procedure described in Section 2 in which $\varepsilon = 0.05$ in (2.13);
- RDL2–0.05: DL2 with restart procedure described in Section 2 in which $\varepsilon = 0.05$ in (2.13);
- RDL3–0.05: DL3 with restart procedure described in Section 2 in which $\varepsilon = 0.05$ in (2.13);
- RDL1–0.2: DL1 with restart procedure described in Section 2 in which $\varepsilon = 0.2$ in (2.13);
- RDL–FR: DL1 with restart procedure suggested by Fletcher and Reeves [16];
- RDL–P: DL1 with restart procedure suggested by Powell [25] in which $c = 0.2$ in (1.10).

The runs were performed on a set of 84 unconstrained optimization test problems of the CUTEr collection [18] with the minimum dimension being equal to 50, as given in Tables 1–3, using a computer Intel(R) Core(TM)2 Duo CPU 2.3 GHz with 8 GB of RAM and with Centos 6.2 server Linux operating system.

Table 2. Outputs (2).

		DL			RDL1-0.05			
Function	n	N_i	N_f	N_g	N_i	N_f	N_g	N_r
DIXON3DQ	10 000	10 000	10 010	10 003	10 000	10 010	10 003	0
DMN15102	66	10 000	10 527	10 338	10 000	10 527	10 338	0
DMN15103	99	10 000	10 085	10 056	10 000	10 085	10 056	0
DMN37142	66	10 000	11 803	11 183	10 000	10 588	10 358	2
DMN37143	99	10 000	11 817	11 328	10 000	11 877	11 366	8
DQDRTIC	5000	98	122	107	98	122	107	0
DQRTIC	5000	0	1	1	0	1	1	0
DRCAV1LQ	4489	0	1	1	0	1	1	0
DRCAV2LQ	4489	0	1	1	0	1	1	0
DRCAV3LQ	4489	0	1	1	0	1	1	0
EDENSCH	2000	17	24	18	17	24	18	0
EG2	1000	4	8	5	4	8	5	0
EIGENALS	2550	10 000	10 269	10 170	10 000	10 058	10 037	5
EIGENBLS	2550	10 000	10 036	10 019	10 000	10 043	10 025	3
EIGENCLS	2652	10 000	10 035	10 017	10 000	10 035	10 017	0
ENGVAL1	5000	11	18	12	12	19	13	2
ERRINROS	50	10 000	13 013	12 078	10 000	12 795	11 896	54
EXTROSNB	1000	4920	5103	5031	4920	5103	5031	0
FLETCBV2	5000	0	1	1	0	1	1	0
FLETCBV3	5000	3	41	41	3	41	41	0
FLETCHBV	5000	3	39	39	3	39	39	0
FLETCHCR	1000	374	420	397	362	414	387	1
FMINSRF2	5625	534	550	537	534	550	537	0
FMINSURF	5625	896	915	904	896	915	904	0
FREUROTH	5000	18	30	24	30	43	36	1
GENHUMPS	5000	0	1	1	0	1	1	0
GENROSE	500	2335	2433	2388	2030	2114	2077	2
MANCINO	100	15	30	21	15	30	21	0

In RDL-FR and RDL-P, the restart has been performed by the descent direction (2.14). In the line search procedure, the Wolfe conditions (2.3) and (2.4) have been employed using Algorithm 3.5 of [23] with $\delta = 0.0001$ and $\sigma = 0.9$, ensuring the descent condition for both of the methods [3]. Moreover, all attempts for finding an approximation of the solution were terminated by reaching maximum of 10 000 iterations or achieving a solution with $||g_k|| < 10^{-5}(1 + |f(x_k)|)$.

Efficiency comparisons were drawn using the Dolan–Moré performance profile [15] on the norm of gradient and the total number of function and gradient evaluations being equal to $N_f + 3N_g$, where N_f and N_g respectively denote the number of function and gradient evaluations. Performance profile gives, for every $\omega \geq 1$, the proportion $p(\omega)$ of the test problems that each considered algorithmic variant has a performance within a factor of ω of the best. Figure 1 illustrates the results of comparisons.

Generally, comparisons show that the proposed restart strategy turns out to be practically effective. The restart occurred for RDLi-0.05, $i = 1, 2, 3$, and RDL1-0.2 averagely in 1.76%, 2.46%, 11.30% and 7.83% of the iterations respectively. To support our theoretical assertions, we report the value of $\eta_k = \dfrac{||d_{k+1}||}{||g_{k+1}||}$ for RDL1-0.2 on some test problems with considerable percentage of the restart as follows:

– BRYBND: 20 restarts out of 44 iterations for which in 58.33% of the restarts η_k variation range is $[2.53, 9.34]$;

TABLE 3. Outputs (3).

Function	n	DL			RDL1-0.05			
		N_i	N_f	N_g	N_i	N_f	N_g	N_r
MSQRTBLS	1024	4058	4084	4065	3969	3991	3976	4
NCB20	5010	132	153	141	149	176	160	6
NCB20B	5000	66	78	70	66	79	71	2
NONCVXU2	5000	0	1	1	0	1	1	0
NONDIA	5000	67	329	298	67	329	298	0
NONDQUAR	5000	10 000	10 092	10 060	4587	4686	4651	1
PENALTY1	1000	141	223	199	154	250	225	4
PENALTY2	200	0	1	1	0	1	1	0
POWELLSG	5000	225	329	294	219	321	288	1
POWER	10 000	762	776	763	682	697	684	1
QUARTC	5000	0	1	1	0	1	1	0
SCHMVETT	5000	11	19	12	11	19	12	0
SENSORS	100	15	24	17	15	24	17	0
SINQUAD	5000	17	36	31	13	31	26	3
SPARSINE	5000	10 000	10 048	10 021	10 000	10 048	10 021	0
SPARSQUR	10 000	44	49	47	44	49	47	0
SPMSRTLS	4999	298	317	303	320	337	325	5
SROSENBR	5000	79	231	201	70	186	162	2
TESTQUAD	5000	10 000	10 134	10 075	10 000	10 134	10 076	1
TOINTGOR	50	81	90	83	103	111	104	2
TOINTGSS	5000	18	25	19	16	22	17	1
TOINTPSP	50	120	139	129	117	132	122	1
TOINTQOR	50	26	36	28	24	33	26	1
TQUARTIC	5000	89	189	170	19	47	40	1
TRIDIA	5000	4221	4259	4234	3389	3424	3397	1
VARDIM	200	1	2	2	1	2	2	0
WOODS	4000	30	38	32	33	41	34	1

- TOINTPSP: 22 restarts out of 142 iterations for which in 100% of the restarts η_k variation range is $[2.13, 17.55]$;
- NCB20B: 5 restarts out of 53 iterations for which in 100% of the restarts η_k variation range is $[3.59, 22.36]$;
- NCB20: 14 restarts out of 153 iterations for which in 64.29% of the restarts η_k variation range is $[2.82, 13.44]$.

Some detailed results have been also provided in Tables 1–3 in which N_i and N_r respectively denote the number of iterations and the number of restarts occurred in each case.

4. Conclusions

In an attempt to find a reasonable solution for an open problem in the nonlinear conjugate gradient methods, a restart procedure has been proposed for the Dai–Liao method based on the singular value decomposition of the search direction matrix. The given restart scheme can be helpful to avoid the probable errors that may appear in computations with possible large quantities generated in the Dai–Liao method. It can also improve the convergence behavior of the method. Results of numerical comparisons showed efficiency of the proposed restart procedure for the Dai–Liao method, especially in the perspective of the quality of the approximate solutions.

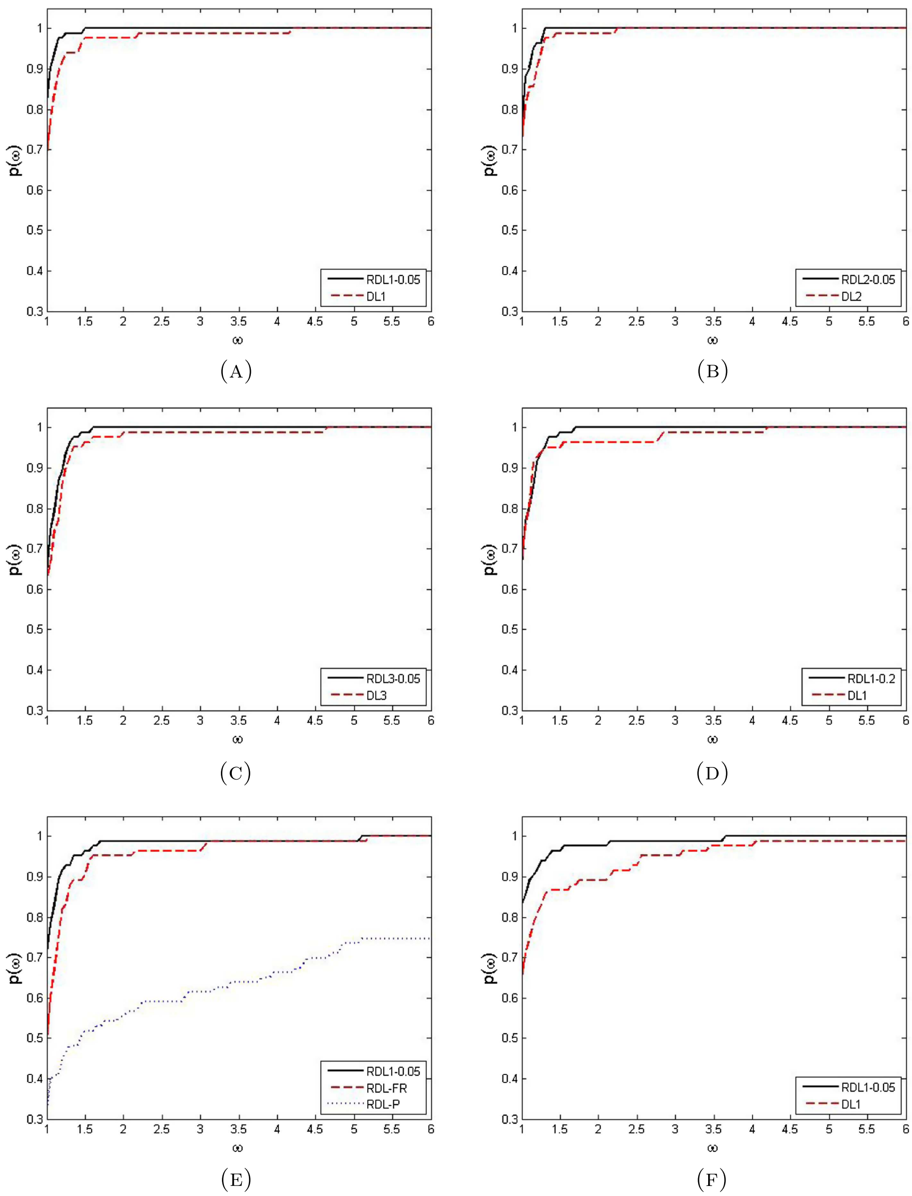

FIGURE 1. Performance profiles for the total number of function and gradient evaluations ((A)–(E)) and the norm of gradient (F).

Acknowledgements. This research was supported by Research Council of Semnan University. The authors thank the anonymous reviewers for their valuable comments and suggestions helped to improve the quality of this work. They are also grateful to Professor Michael Navon for providing the line search code.

REFERENCES

[1] N. Andrei, Numerical comparison of conjugate gradient algorithms for unconstrained optimization. *Stud. Inform. Control* **16** (2007) 333–352.

[2] N. Andrei, Open problems in conjugate gradient algorithms for unconstrained optimization. *B. Malays. Math. Sci. Soc.* **34** (2011) 319–330.

[3] S. Babaie-Kafaki, On the sufficient descent condition of the Hager–Zhang conjugate gradient methods. *4OR* **12** (2014) 285–292.

[4] S. Babaie-Kafaki and R. Ghanbari, The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. *Eur. J. Oper. Res.* **234** (2014) 625–630.

[5] S. Babaie-Kafaki and R. Ghanbari, A descent family of Dai–Liao conjugate gradient methods. *Optim. Methods Softw.* **29** (2014) 583–591.

[6] S. Babaie-Kafaki and R. Ghanbari, Two optimal Dai–Liao conjugate gradient methods. *Optimization* **64** (2015) 2277–2287.

[7] S. Babaie-Kafaki and R. Ghanbari, Descent symmetrization of the Dai–Liao conjugate gradient method. *Asia–Pac. J. Oper. Res.* **33** (2016) 1650008.

[8] J. Barzilai and J.M. Borwein, Two–point stepsize gradient methods. *IMA J. Numer. Anal.* **8** (1988) 141–148.

[9] E.M.L. Beale, A derivation of conjugate gradients. In: Numerical Methods for Nonlinear Optimization, edited by F.A. Lootsma. Academic Press, New York, NY (1972) 39–43.

[10] H. Crowder and P. Wolfe, Linear convergence of the conjugate gradient method. *IBM J. Res. Dev.* **16** (1972) 431–433.

[11] Y.H. Dai and C.X. Kou, A nonlinear conjugate gradient algorithm with an optimal property and an improved Wolfe line search. *SIAM J. Optim.* **23** (2013) 296–320.

[12] Y.H. Dai and L.Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods. *Appl. Math. Optim.* **43** (2001) 87–101.

[13] Y.H. Dai, J.Y. Han, G.H. Liu, D.F. Sun, H.X. Yin and Y.X. Yuan, Convergence properties of nonlinear conjugate gradient methods. *SIAM J. Optim.* **10** (1999) 348–358.

[14] Y.H. Dai, L.Z. Liao and D. Li, On restart procedures for the conjugate gradient method. *Numer. Algorithms* **35** (2004) 249–260.

[15] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **91** (2002) 201–213.

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

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

[18] N.I.M. Gould, D. Orban and P.L. Toint, CUTEr and SifDec: a constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Softw.* **29** (2003) 373–394.

[19] W.W. Hager and H. Zhang, A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2** (2006) 35–58.

[20] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **49** (1952) 409–436.

[21] N.J. Higham, Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia, PA (2002).

[22] M.F. McGuire and P. Wolfe, *Evaluating a restart procedure for conjugate gradients*. Report RC–4382. IBM Research Center, Yorktown Heights (1973).

[23] J. Nocedal and S.J. Wright, Numerical Optimization. Springer, New York, NY (2006).

[24] A. Perry, A modified conjugate gradient algorithm. *Oper. Res.* **26** (1978) 1073–1078.

[25] M.J.D. Powell, Restart procedures for the conjugate gradient method. *Math. Program.* **12** (1977) 241–254.

[26] W. Sun and Y.X. Yuan, Optimization Theory and Methods: Nonlinear Programming. Springer, New York, NY (2006).

[27] D.S. Watkins, Fundamentals of Matrix Computations. John Wiley and Sons, New York, NY (2002).

[28] C. Xu and J.Z. Zhang, A survey of quasi-Newton equations and quasi-Newton methods for optimization. *Ann. Oper. Res.* **103** (2001) 213–234.

[29] G. Zoutendijk, Nonlinear programming computational methods, edited by J. Abadie. In: Integer and Nonlinear Programming. North–Holland, Amsterdam (1970) 37–86.