JORC (2013) 1:425–451
DOI 10.1007/s40305-013-0029-4

# A Subspace Version of the Powell–Yuan Trust-Region Algorithm for Equality Constrained Optimization

**Geovani Nunes Grapiglia · Jinyun Yuan · Ya-xiang Yuan**

Received: 13 August 2013 / Revised: 30 October 2013 / Accepted: 8 November 2013 /
Published online: 11 December 2013


**Abstract** This paper studied subspace properties of the Celis–Dennis–Tapia (CDT) subproblem that arises in some trust-region algorithms for equality constrained optimization. The analysis is an extension of that presented by Wang and Yuan (Numer. Math. 104:241–269, 2006) for the standard trust-region subproblem. Under suitable conditions, it is shown that the trial step obtained from the CDT subproblem is in the subspace spanned by all the gradient vectors of the objective function and of the constraints computed until the current iteration. Based on this observation, a subspace version of the Powell–Yuan trust-region algorithm is proposed for equality constrained optimization problems where the number of constraints is much lower than

G.N. Grapiglia was supported by Coordination for the Improvement of Higher Education Personnel (CAPES), Brazil (Grant PGCI No. 12347/12-4).
J. Yuan was partially supported by Coordination for the Improvement of Higher Education Personnel (CAPES) and by the National Council for Scientific and Technological Development (CNPq), Brazil.
Y.-x. Yuan was partially supported by Natural Science Foundation of China, China (Grant No. 11331012).

G.N. Grapiglia (✉) · J. Yuan
Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico, Cx. postal 19.081, 81531-980, Curitiba, Paraná, Brazil
e-mail: geovani_mat@outlook.com

J. Yuan
e-mail: jin@ufpr.br

G.N. Grapiglia
The Capes Foundation, Ministry of Education of Brazil, Cx. postal 250, 70.040-020, Brasília, Distrito Federal, Brazil

Y.-x. Yuan
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Zhongguancundonglu 55, Beijing 100190, P.R. China
e-mail: yyx@lsec.cc.ac.cn

the number of variables. The convergence analysis is given and numerical results are also reported.

**Keywords** Constrained optimization · Trust-region methods · Subspace methods

**Mathematics Subject Classification (2010)** 90C30 · 90C53 · 49M37

## 1 Introduction

We consider the equality constrained optimization problem

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

$$\text{subject to} \qquad h_i(x) = 0, \quad i = 1, \cdots, m, \tag{1.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \cdots, m)$ are continuously differentiable, and the constraints gradients are linearly independent. For convenience, throughout this paper the following notation is used:

$$c(x) = \big(h_1(x), \cdots, h_m(x)\big)^T, \tag{1.3}$$

$$A(x) = J_c(x)^T = \big(\nabla h_1(x), \cdots, \nabla h_m(x)\big), \tag{1.4}$$

$$g(x) = \nabla f(x). \tag{1.5}$$

We also use $c_k$ for $c(x_k)$, $A_k$ for $A(x_k)$, $g_k$ for $g(x_k)$, etc.

The Powell–Yuan trust-region algorithm [11] is an iterative procedure to solve (1.1)–(1.2), which generates a sequence of points $\{x_k\}$ in the following way. At the beginning of the $k$th iteration, $x_k \in \mathbb{R}^n$, $\Delta_k > 0$ and $B_k \in \mathbb{R}^{n\times n}$ symmetric are available. If $x_k$ does not satisfy the Kuhn–Tucker conditions, a trial step $s_k$ is computed by solving the CDT subproblem (see Celis, Dennis and Tapia [2]):

$$\min_{d\in\mathbb{R}^n} \phi_k(d) \equiv g_k^T d + \frac{1}{2} d^T B_k d, \tag{1.6}$$

$$\text{s.t.} \quad \big\|c_k + A_k^T d\big\|_2 \leqslant \xi_k, \tag{1.7}$$

$$\|d\|_2 \leqslant \Delta_k, \tag{1.8}$$

where $\xi_k$ is any number satisfying the inequalities

$$\min_{\|d\|_2\leqslant b_1\Delta_k} \big\|c_k + A_k^T d\big\|_2 \leqslant \xi_k \leqslant \min_{\|d\|_2\leqslant b_2\Delta_k} \big\|c_k + A_k^T d\big\|_2, \tag{1.9}$$

and $b_1$ and $b_2$ are two given constants with $0 < b_2 \leqslant b_1 < 1$. The merit function is Fletcher's differentiable function:

$$\psi_k(x) = f(x) - \lambda(x)^T c(x) + \mu_k \big\|c(x)\big\|_2^2, \tag{1.10}$$

where $\mu_k > 0$ is a penalty parameter and $\lambda(x)$ is the minimum norm solution of

$$\min_{\lambda\in\mathbb{R}^m} \left\| g(x) - A(x)\lambda \right\|_2. \tag{1.11}$$

The predicted change $D_k$ in $\psi_k(x)$ is defined by

$$D_k = (g_k - A_k\lambda_k)^T s_k + \frac{1}{2} s_k^T B_k \hat{s}_k - \left[\lambda(x_k + s_k) - \lambda_k\right]^T \left(c_k + \frac{1}{2} A_k^T s_k\right) + \mu_k\left(\left\|c_k + A_k^T s_k\right\|_2^2 - \|c_k\|_2^2\right), \tag{1.12}$$

where $\mu_k$ is chosen so that $D_k < 0$ and where $\hat{s}_k$ is the orthogonal projection of $s_k$ to the null space of $A_k^T$, namely

$$\hat{s}_k = P_k s_k, \quad \text{with } P_k = I_n - A_k A_k^+. \tag{1.13}$$

From the ratio

$$\rho_k = \frac{\psi_k(x_k + s_k) - \psi_k(x_k)}{D_k}, \tag{1.14}$$

the next iterate $x_{k+1}$ is obtained by the formula

$$x_{k+1} = \begin{cases} x_k + s_k, & \text{if } \rho_k > 0, \\ x_k, & \text{otherwise.} \end{cases} \tag{1.15}$$

Further, the trust-region radius $\Delta_{k+1}$ for the next iteration is given by the rule

$$\Delta_{k+1} = \begin{cases} \max\{\Delta_k, 4\|s_k\|_2\}, & \text{if } \rho_k > 0.9, \\ \Delta_k, & \text{if } 0.1 \leqslant \rho_k \leqslant 0.9, \\ \min\{\frac{\Delta_k}{4}, \frac{\|s_k\|_2}{2}\}, & \text{if } \rho_k < 0.1. \end{cases} \tag{1.16}$$

Finally, a symmetric matrix $B_{k+1}$ is obtained and the process is repeated with $k := k + 1$.

We summarize the above trust-region algorithm as follows:

**Algorithm 1.1** (Powell–Yuan Trust-Region Algorithm)

Step 0 Given $x_1 \in \mathbb{R}^n$, $\Delta_1 > 0$, $B_1 \in \mathbb{R}^{n\times n}$ symmetric, $\varepsilon_s > 0$, $\mu_1 > 0$ and $0 < b_2 \leqslant b_1 < 1$, set $k := 1$.

Step 1 If $\|c_k\|_2 + \|g_k - A_k\lambda_k\|_2 \leqslant \varepsilon_s$, then stop. Otherwise, compute $\xi_k$ satisfying (1.9) and solve the CDT subproblem (1.6)–(1.8) to obtain a trial step $s_k$.

Step 2 Compute $D_k$ by (1.12). If the inequality

$$D_k \leqslant \frac{1}{2}\mu_k\left(\left\|c_k + A_k^T s_k\right\|_2^2 - \|c_k\|_2^2\right) \tag{1.17}$$

fails, then increase $\mu_k$ to the value

$$\mu_k^{\text{new}} = 2\mu_k^{\text{old}} + \max\left\{0, \frac{2D_k^{\text{old}}}{\|c_k\|_2^2 - \|c_k + A_k^T s_k\|_2^2}\right\} \tag{1.18}$$

which ensures that the new value of expression (1.12) satisfies condition (1.17).

Step 3 Compute $\rho_k$ by (1.14);
Set $x_{k+1}$ by (1.15);
Set $\Delta_{k+1}$ by (1.16).

Step 4 Generate $B_{k+1}$ symmetric, set $\mu_{k+1} := \mu_k$, $k := k+1$ and go to Step 1.

To solve the CDT subproblem (1.6)–(1.8) in Step 1, some iterative algorithms have been presented. For example, under the assumption that $B_k$ is positive definite, two different algorithms have been proposed by Yuan [16] and Zhang [17], respectively; while for a general symmetric matrix $B_k$, an algorithm has been proposed by Li and Yuan [9]. However, since these algorithms require repeated matrix factorizations in each iteration, it could be very costly to solve the CDT subproblem (1.6)–(1.8), mainly for problems with a large number of variables and constraints.

Motivated by the subspace trust-region method for unconstrained optimization proposed by Wang and Yuan [14], in this paper we explore the subspace properties of the CDT subproblem when the matrices $B_k$ are updated by quasi-Newton formulas. With an analysis totally analog to that in Wang and Yuan [14], it is found that the trial step $s_k$ defined by the CDT subproblem (1.6)–(1.8) is always in the subspace $G_k$ spanned by

$$\bigcup_{i=1}^{k} \{\nabla h_1(x_i), \cdots, \nabla h_m(x_i), g_i\}.$$

Therefore, it is equivalent to solving the subproblem within this subspace. Based on this observation, we can solve a smaller CDT subproblem in early iterations of the algorithm, reducing the computational effort for problems where the dimension of the subspace $G_k$ remains far smaller than the number of variables $n$.

This work is organized as follows. The equivalence between the CDT subproblem and that in the subspace is proved in the next section. In Sect. 3, a subspace version of the Powell–Yuan algorithm is proposed. The global convergence analysis is given in Sect. 4. Finally, preliminary numerical results on problems in CUTEr collection are reported in Sect. 5.

## 2 Subspace Properties

In this section, we shall study subspace properties of the trial step $s_k$ at the $k$th iteration, which is assumed to be a solution of the CDT subproblem (1.6)–(1.8). All the results here are developed corresponding to those presented in Sect. 2 of Wang and Yuan [14].

**Lemma 2.1** *Let $s_k \in \mathbb{R}^n$ be a solution of* (1.6)–(1.8), *and assume that*

$$\xi_k > \min_{\|d\|_2 \leqslant \Delta_k} \left\| c_k + A_k^T d \right\|_2.$$

*Then, there exist non-negative constants $\alpha_k$ and $\beta_k$ such that*

$$\left(B_k + \alpha_k I_n + \beta_k A_k A_k^T\right) s_k = -(g_k + \beta_k A_k c_k), \tag{2.1}$$

*where $\alpha_k$ and $\beta_k$ satisfy the complementarity conditions*

$$\alpha_k\left[\Delta_k - \|s_k\|_2\right] = 0, \tag{2.2}$$

$$\beta_k\left[\xi_k - \left\|A_k^T s_k + c_k\right\|_2\right] = 0. \tag{2.3}$$

*Proof* See Theorem 2.1 in Yuan [15]. □

**Lemma 2.2** *Let $S_k$ be an $r$ $(1 \leqslant r \leqslant n)$ dimensional subspace in $\mathbb{R}^n$, and $Z_k \in \mathbb{R}^{n\times r}$ is an orthonormal basis matrix of $S_k$, namely*

$$S_k = \operatorname{span}\{Z_k\}, \qquad Z_k^T Z_k = I_r. \tag{2.4}$$

*Suppose that*

$$\left\{\nabla h_1(x_k), \cdots, \nabla h_m(x_k), g_k\right\} \subset S_k, \tag{2.5}$$

*and $B_k \in \mathbb{R}^{n\times n}$ is a symmetric matrix satisfying*

$$B_k u = \sigma u, \quad \forall u \in S_k^{\perp}, \tag{2.6}$$

*where $\sigma > 0$. Then, the subproblem (1.6)–(1.8) is equivalent to the following problem:*

$$\min_{\bar d \in \mathbb{R}^r} \bar\phi_k(\bar d) \equiv \bar g_k^T \bar d + \frac{1}{2}\bar d^T \bar B_k \bar d, \tag{2.7}$$

$$\text{s.t.} \quad \left\|c_k + \bar A_k^T \bar d\right\|_2 \leqslant \xi_k, \tag{2.8}$$

$$\|\bar d\|_2 \leqslant \Delta_k, \tag{2.9}$$

*where $\bar g_k = Z_k^T g_k$, $\bar B_k = Z_k^T B_k Z_k$ and $\bar A_k = Z_k^T A_k$. That is to say, if $s_k$ is a solution of (1.6)–(1.8), then $s_k = Z_k \bar s_k \in S_k$, where $\bar s_k$ is a solution of (2.7)–(2.9). On the other hand, if $\bar s_k$ is a solution of (2.7)–(2.9), then $s_k = Z_k \bar s_k$ is a solution of (1.6)–(1.8).*

*Proof* Let $U_k \in \mathbb{R}^{n\times(n-r)}$ be a matrix such that $[U_k, Z_k]$ is an $n \times n$ orthogonal matrix. Then, for each $d \in \mathbb{R}^n$, there exists one and only one pair $\bar d \in \mathbb{R}^r$, $u \in \mathbb{R}^{n-r}$ such that $d = Z_k \bar d + U_k u$. As $B_k$ is symmetric, it follows that

$$\begin{aligned}
\phi_k(d) &= g_k^T d + \frac{1}{2} d^T B_k d \\
&= g_k^T [Z_k \bar d + U_k u] + \frac{1}{2}[Z_k \bar d + U_k u]^T B_k [Z_k \bar d + U_k u] \\
&= g_k^T Z_k \bar d + g_k^T U_k u + \frac{1}{2}\bar d^T Z_k^T B_k Z_k \bar d + \frac{1}{2}\bar d^T Z_k^T B_k U_k u
\end{aligned}$$

$$+\frac{1}{2}u^TU_k^TB_kZ_k\bar{d}+\frac{1}{2}u^TU_k^TB_kU_ku$$
$$=g_k^TZ_k\bar{d}+g_k^TU_ku+\frac{1}{2}\bar{d}^TZ_k^TB_kZ_k\bar{d}+\bar{d}^TZ_k^TB_kU_ku$$
$$+\frac{1}{2}u^TU_k^TB_kU_ku$$
$$=\bar{g}_k^T\bar{d}+g_k^TU_ku+\frac{1}{2}\bar{d}^T\bar{B}_k\bar{d}+\bar{d}^TZ_k^TB_kU_ku$$
$$+\frac{1}{2}u^TU_k^TB_kU_ku, \tag{2.10}$$

where $\bar{g}_k=Z_k^Tg_k$ and $\bar{B}_k=Z_k^TB_kZ_k$. Since $g_k\in S_k$ and the columns of $U_k$ are vectors in $S_k^{\perp}$, we obtain

$$g_k^TU_k=0, \tag{2.11}$$
$$Z_k^TB_kU_k=\sigma Z_k^TU_k=0 \quad \text{and} \quad U_k^TB_kU_k=\sigma I_{n-r}, \tag{2.12}$$

where the last line is due to the assumption (2.6). Hence, (2.10)–(2.12) imply that

$$\phi_k(d)=\left(\bar{g}_k^T\bar{d}+\frac{1}{2}\bar{d}^T\bar{B}_k\bar{d}\right)+\frac{1}{2}\sigma u^Tu. \tag{2.13}$$

From the fact that the rows of $A_k^T$ are the vectors $\nabla h_i(x_k)\in S_k$ and the columns of $U_k$ belong to $S_k^{\perp}$, it follows that $A_k^TU_k=0$. Consequently,

$$\left\|c_k+A_k^Td\right\|_2=\left\|c_k+A_k^TZ_k\bar{d}\right\|_2=\left\|c_k+\bar{A}_k^T\bar{d}\right\|_2, \tag{2.14}$$

where $\bar{A}_k=Z_k^TA_k$. In addition, by the orthonormality of $Z_k$ and $U_k$, we have

$$\|d\|_2^2=\|\bar{d}\|_2^2+\|u\|_2^2. \tag{2.15}$$

Now, (2.13)–(2.15) imply that the subproblem (1.6)–(1.8) is equivalent to

$$\min_{\bar{d}\in\mathbb{R}^r,u\in\mathbb{R}^{n-r}}\left(\bar{g}_k^T\bar{d}+\frac{1}{2}\bar{d}^T\bar{B}_k\bar{d}\right)+\frac{1}{2}\sigma u^Tu, \tag{2.16}$$
$$\text{s.t.}\quad \left\|c_k+\bar{A}_k^T\bar{d}\right\|_2\leqslant\xi_k, \tag{2.17}$$
$$\|\bar{d}\|_2^2+\|u\|_2^2\leqslant\Delta_k^2, \tag{2.18}$$

with the relation $d=Z_k\bar{d}+U_ku$.

Because of $\sigma>0$, if $\bar{s}_k$ is a solution of (2.7)–(2.9) then $(\bar{s}_k,0)\in\mathbb{R}^r\times\mathbb{R}^{n-r}$ is a solution of (2.16)–(2.18) and, therefore, $s_k=Z_k\bar{s}_k$ is a solution of (1.6)–(1.8). To prove the reciprocal, we assume by contradiction that there exists a solution $s_k=Z_k\bar{s}_k+U_ku_k$ of (1.6)–(1.8) such that $u_k\neq 0$. In this case,

$$\phi_k(s_k)\leqslant\phi_k(s), \tag{2.19}$$

for all $s \in \mathbb{R}^n$ satisfying (1.7)–(1.8). In particular,

$$\phi_k(s_k) \leqslant \phi_k\big(s_k^*\big), \tag{2.20}$$

where $s_k^* = Z_k \bar{s}_k$. However, since $u_k \neq 0$ and $\sigma > 0$, from (2.13) it follows that

$$\phi_k(s_k) > \bar{g}_k^T \bar{s}_k + \frac{1}{2} \bar{s}_k^T \bar{B}_k \bar{s}_k = \phi_k\big(s_k^*\big), \tag{2.21}$$

which contradicts (2.20). This shows that if $s_k$ is a solution of (1.6)–(1.8) then $s_k = Z_k \bar{s}_k$. The fact that $\bar{s}_k$ is a solution of (2.7)–(2.9) follows from the equivalence between (1.6)–(1.8) and (2.16)–(2.18) with $u = 0$. □

*Remark 2.1* From the above lemma, if the assumptions (2.4)–(2.6) are satisfied, then we can solve the subproblem (2.7)–(2.9) in $\mathbb{R}^r$ instead of solving the subproblem (1.6)–(1.8) in $\mathbb{R}^n$, which can reduce the computational efforts significantly when $r \ll n$.

*Remark 2.2* For the further analysis, it is useful to see that

$$B_k u = \sigma u, \quad \forall u \in G_k^{\perp} \quad \Longrightarrow \quad B_k z \in G_k, \quad \forall z \in G_k. \tag{2.22}$$

Indeed, given $z \in G_k$ and $u \in G_k^{\perp}$, as $B_k$ is a symmetric matrix, we have

$$\begin{aligned} \langle B_k z, u\rangle_2 &= \big\langle z, B_k^T u\big\rangle_2 = \langle z, B_k u\rangle_2 \\ &= \langle z, \sigma u\rangle_2 = \sigma \langle z, u\rangle_2 = 0. \end{aligned}$$

Thus, $B_k z \in (G_k^{\perp})^{\perp} = G_k$ for all $z \in G_k$.

**Lemma 2.3** *Suppose that* $\xi_1 > \min_{\|d\|_2 \leqslant \Delta_1} \|c_1 + A_1^T d\|_2$, $B_1 = \sigma I_n$ $(\sigma > 0)$ *and* $B_k$ *is the kth update matrix given by one formula chosen from PSB and Broyden family. Let* $g_k = \nabla f(x_k)$, $s_k$ *be a solution of* (1.6)–(1.8) *and*

$$G_k = \operatorname{span}\left[\bigcup_{i=1}^{k} \big\{\nabla h_1(x_i), \cdots, \nabla h_m(x_i), g_i\big\}\right]. \tag{2.23}$$

*Then, for all* $k$, $s_k \in G_k$ *and* $B_k u = \sigma u$ *for all* $u \in G_k^{\perp}$.

*Proof* The PSB formula and Broyden family formulas (see, e.g., Sun and Yuan [13]) can be represented, respectively, as

$$B_{k+1}^{(\mathrm{PSB})} = B_k^{(\mathrm{PSB})} + \frac{\delta_k s_k^T + s_k \delta_k^T}{s_k^T s_k} - \frac{(\delta_k^T s_k) s_k s_k^T}{(s_k^T s_k)^2}, \tag{2.24}$$

$$B_{k+1}^{(B)} = B_k^{(B)} - \frac{B_k^{(B)} s_k s_k^T B_k^{(B)}}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} + \theta_k\big(s_k^T B_k^{(B)} s_k\big) w_k w_k^T, \tag{2.25}$$

where $s_k = x_{k+1} - x_k$, $y_k = (g_{k+1} - g_k) - (A_{k+1}\lambda_{k+1} - A_k\lambda_k)$ or $y_k = (g_{k+1} - g_k) - (A_{k+1} - A_k)\lambda_k$, $\delta_k = y_k - B_k^{(\text{PSB})} s_k$ and

$$w_k = \frac{y_k}{s_k^T y_k} - \frac{B_k^{(B)} s_k}{s_k^T B_k^{(B)} s_k}. \tag{2.26}$$

We prove the result by induction over $k$. By Lemma 2.1 and $\sigma > 0$,

$$\begin{aligned}
&\big(B_1 + \alpha_1 I_n + \beta_1 A_1 A_1^T\big)s_1 = -(g_1 + \beta_1 A_1 c_1) \\
&\quad \Longrightarrow \quad \big(\sigma I_n + \alpha_1 I_n + \beta_1 A_1 A_1^T\big)s_1 = -(g_1 + \beta_1 A_1 c_1) \\
&\quad \Longrightarrow \quad (\sigma + \alpha_1)s_1 = -\big(g_1 + \beta_1 A_1 c_1 + \beta_1 A_1 A_1^T s_1\big) \\
&\quad \Longrightarrow \quad s_1 = -(\sigma + \alpha_1)^{-1}\big(g_1 + \beta_1 A_1 c_1 + \beta_1 A_1 A_1^T s_1\big) \\
&\quad \Longrightarrow \quad s_1 \in G_1,
\end{aligned}$$

where the last line is true because $g_1$, $A_1 c_1$ and $A_1 A_1^T s_1 \in G_1$. Moreover,

$$B_1^{(\text{PSB})} u = B_1^{(B)} u = (\sigma I_n) u = \sigma u, \quad \forall u \in G_1^{\perp}. \tag{2.27}$$

Hence, the lemma is true for $k = 1$. Assume that the lemma is true for $k = i$, that is,

$$s_i \in G_i, \tag{2.28}$$

and

$$B_i^{(\text{PSB})} u = B_i^{(B)} u = \sigma u, \quad \forall u \in G_i^{\perp}. \tag{2.29}$$

Consider $\tilde{u} \in G_{i+1}^{\perp}$. In particular, we have $\tilde{u} \in G_i^{\perp}$ (since $G_i \subset G_{i+1} \Longrightarrow G_{i+1}^{\perp} \subset G_i^{\perp}$). Then, as $y_i \in G_{i+1}$ and $B_i^{(\text{PSB})}$ and $B_i^{(B)}$ are symmetric matrices, it follows from (2.28) and (2.29) that

$$\begin{aligned}
B_{i+1}^{(\text{PSB})}\tilde{u} &= B_i^{(\text{PSB})}\tilde{u} + \frac{(\delta_i s_i^T + s_i \delta_i^T)\tilde{u}}{s_i^T s_i} - \frac{(\delta_i^T s_i) s_i s_i^T \tilde{u}}{(s_i^T s_i)^2} \\
&= \sigma \tilde{u} + \frac{\delta_i s_i^T \tilde{u} + s_i (y_i^T \tilde{u} - s_i^T B_i^{(\text{PSB})} \tilde{u})}{s_i^T s_i} \\
&= \sigma \tilde{u} - \sigma \frac{s_i s_i^T \tilde{u}}{s_i^T s_i} \\
&= \sigma \tilde{u},
\end{aligned}$$

and

$$B_{i+1}^{(B)}\tilde{u} = B_i^{(B)}\tilde{u} - \frac{B_i^{(B)} s_i s_i^T B_i^{(B)} \tilde{u}}{s_i^T B_i s_i} + \frac{y_i y_i^T \tilde{u}}{s_i^T y_i} + \theta_i \big(s_i^T B_i^{(B)} s_i\big) w_i w_i^T \tilde{u}$$

$$
\begin{aligned}
&= \sigma\tilde{u} - \frac{\sigma B_i^{(B)} s_i s_i^T \tilde{u}}{s_i^T B_i^{(B)} s_i} + \theta_i \left(s_i^T B_i^{(B)} s_i\right) w_i \left( \frac{y_i^T}{s_i^T y_i} - \frac{s_i^T B_i^{(B)}}{s_i^T B_i^{(B)} s_i} \right) \tilde{u} \\
&= \sigma\tilde{u} + \theta_i \left(s_i^T B_i^{(B)} s_i\right) w_i \left( \frac{y_i^T \tilde{u}}{s_i^T y_i} - \frac{s_i^T B_i^{(B)} \tilde{u}}{s_i^T B_i^{(B)} s_i} \right) \\
&= \sigma\tilde{u} - \sigma\theta_i \left(s_i^T B_i^{(B)} s_i\right) w_i \frac{s_i^T \tilde{u}}{s_i^T B_i^{(B)} s_i} \\
&= \sigma\tilde{u}.
\end{aligned}
$$

Since $\tilde{u} \in G_{i+1}^{\perp}$ is arbitrary, this proves that

$$
B_{i+1}^{(\mathrm{PSB})} u = B_{i+1}^{(B)} u = \sigma u, \quad \forall u \in G_{i+1}^{\perp}. \tag{2.30}
$$

Now, let $s_{i+1}$ be a solution of the subproblem (1.6)–(1.8) for $k = i + 1$. Then, by

$$
\left\{\nabla h_1(x_{i+1}), \cdots, \nabla h_m(x_{i+1}), g_{i+1}\right\} \subset G_{i+1},
$$

equation (2.30) and Lemma 2.2 (where $k = i + 1$), we conclude that $s_{i+1} = Z_{i+1}\bar{s}_{i+1} \in G_{i+1}$ (where $\bar{s}_{i+1}$ is a solution of the subproblem (2.7)–(2.9) for $k = i + 1$, and $Z_{i+1}$ is an orthonormal basis matrix of $G_{i+1}$). The proof is complete. □

*Remark 2.3* The result of Lemma 2.3 also is true if the matrices $B_k$ are updated by the family of formulas

$$
B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\eta_k \eta_k^T}{s_k^T \eta_k}, \tag{2.31}
$$

where $\eta_k = \theta_k y_k + (1 - \theta_k) B_k s_k$ with $\theta_k \in [0, 1]$, which includes the damped BFGS formula of Powell [10]. Indeed, if $B_1 = \sigma I_n$ ($\sigma > 0$) and $\xi_1 > \min_{\|d\|_2 \leqslant \Delta_1} \|c_1 + A_1^T d\|_2$, then by the same argument used in the proof of Lemma 2.3 we conclude that $s_1 \in G_1$ and $B_1 u = \sigma u$ for all $u \in G_1^{\perp}$. Thus, the result is true for $k = 1$. Assume that it is true for $k = i$, that is,

$$
s_i \in G_i, \tag{2.32}
$$

and

$$
B_i u = \sigma u, \quad \forall u \in G_i^{\perp}. \tag{2.33}
$$

Then, from Remark 2.2 it follows that $B_i s_i \in G_i \subset G_{i+1}$. As $y_i \in G_{i+1}$, we also have $\eta_i = \theta_i y_i + (1 - \theta_i) B_i s_i \in G_{i+1}$. Now, given $\tilde{u} \in G_{i+1}^{\perp} \subset G_i^{\perp}$, it follows from (2.32) and (2.33) that

$$
\begin{aligned}
B_{i+1}\tilde{u} &= B_i \tilde{u} - \frac{B_i s_i s_i^T B_i \tilde{u}}{s_i^T B_i s_i} + \frac{\eta_i \eta_i^T \tilde{u}}{s_i^T \eta_i} \\
&= \sigma\tilde{u} - \frac{\sigma B_i s_i s_i^T \tilde{u}}{s_i^T B_i s_i} \\
&= \sigma\tilde{u}.
\end{aligned}
$$

Since $\tilde{u} \in G_{i+1}^{\perp}$ is arbitrary, this proves that

$$B_{i+1} u = \sigma u, \quad \forall u \in G_{i+1}^{\perp}. \tag{2.34}$$

Therefore, the conclusion follows by induction in the same way as in the proof of Lemma 2.3.

By Lemmas 2.2, 2.3 and Remark 2.3, we obtain the following theorem.

**Theorem 2.1** *Let $Z_k$ be an orthonormal basis matrix of the subspace*

$$G_k = \operatorname{span}\left[\bigcup_{i=1}^{k} \{\nabla h_1(x_i), \cdots, \nabla h_m(x_i), g_i\}\right]. \tag{2.35}$$

*Suppose that $\xi_1 > \min_{\|d\|_2 \leqslant \Delta_1} \|c_1 + A_1^T d\|_2$, $B_1 = \sigma I_n$ $(\sigma > 0)$ and $B_k$ is the kth update matrix given by one formula chosen from damped BFGS, PSB and Broyden family. Let $s_k$ be a solution of the subproblem* (1.6)–(1.8). *Then, there exists a solution $\bar{s}_k$ of* (2.7)–(2.9) *such that $s_k = Z_k \bar{s}_k$, which implies $s_k \in G_k$. Reciprocally, if $\bar{s}_k$ is a solution of* (2.7)–(2.9), *then $s_k = Z_k \bar{s}_k$ is a solution of* (1.6)–(1.8).

From the above theorem, the trial step $s_k$ is in the subspace $G_k$. Hence, we can update the approximate Hessian matrix $B_k$ in the subspace $G_k$ by the damped BFGS formula, the PSB formula or any one from the Broyden family. The following result has been given by Siegel [12] and Gill and Leonard [5] for Broyden family, and by Wang and Yuan [14] including the PSB formula. We give it here for completeness.

**Lemma 2.4** *Let $Z \in \mathbb{R}^{n \times r}$ be a column orthogonal matrix. Suppose that $s_k \in \operatorname{span}\{Z\}$, and the matrix $B_{k+1} = Update(B_k, s_k, y_k)$ is obtained by the damped BFGS formula, the PSB formula or any one from the Broyden family. Then, denoting $\bar{B}_{k+1} = Z^T B_{k+1} Z$, $\tilde{B}_k = Z^T B_k Z$, $\tilde{s}_k = Z^T s_k$ and $\tilde{y}_k = Z^T y_k$, we have $\bar{B}_{k+1} = Update(\tilde{B}_k, \tilde{s}_k, \tilde{y}_k)$.*

*Proof* First, note that

$$s_k \in \operatorname{span}\{Z\} \quad \Longrightarrow \quad s_k = Z Z^T s_k. \tag{2.36}$$

Then,

$$\begin{aligned}
s_k^T y_k &= \left(Z Z^T s_k\right)^T y_k = \left(Z^T s_k\right)^T Z^T y_k = \tilde{s}_k^T \tilde{y}_k, \\
s_k^T B_k s_k &= \left(Z Z^T s_k\right)^T B_k \left(Z Z^T s_k\right) = \left(Z^T s_k\right)^T Z^T B_k Z \left(Z^T s_k\right) = \tilde{s}_k^T \tilde{B}_k \tilde{s}_k, \\
Z^T B_k s_k &= Z^T B_k Z \left(Z^T s_k\right) = \tilde{B}_k \tilde{s}_k.
\end{aligned}$$

Therefore, multiplying (2.24), (2.25), and (2.31) by $Z^T$ from the left and $Z$ from the right, we can obtain the result of the lemma. □

*Remark 2.4* By Theorem 2.1, we can solve the CDT subproblem (1.6)–(1.8) by solving (2.7)–(2.9) in the subspace $G_k$, provided that $\xi_1$ and $B_1$ are appropriately chosen and a suitable quasi-Newton formula is used to update $B_k$. Further, it follows from Lemma 2.4 that the reduced matrix $\bar{B}_k = Z_k^T B_k Z_k$ of $B_k$ in the subspace $G_k$ can be obtained by updating the reduced matrix $\tilde{B}_{k-1} = Z_k^T B_{k-1} Z_k$, where $Z_k$ is the orthonormal basis matrix of the subspace $G_k$. These subspace properties can be explored to reduce the amount of computation required to compute the trial step $s_k$ when $n \gg m$ and the dimension of the subspace $G_k$ remains far smaller than $n$.

## 3 The Algorithm

Using the subspace properties of the CDT subproblem studied in the previous section, we shall construct a subspace version of Algorithm 1.1. Suppose at the $k$th iteration, $Z_k \in \mathbb{R}^{n\times r_k}$ has been obtained, which is an orthonormal basis matrix of $G_k$. Further, suppose that $\bar{s}_k$ is obtained by solving (2.7)–(2.9) and $s_k = Z_k \bar{s}_k$, $x_{k+1} = x_k + s_k$ and $g_{k+1} = \nabla f(x_{k+1})$. Then, we have to compute $Z_{k+1}$, $\bar{g}_{k+1} = Z_{k+1}^T g_{k+1}$, $\bar{A}_{k+1} = Z_{k+1}^T A_{k+1}$ and $\bar{B}_{k+1} = Z_{k+1}^T B_{k+1} Z_{k+1}$ for the next iteration.

Thinking about numerical stability, as in Wang and Yuan [14], we could use the procedure of Gram–Schmidt with reorthogonalization (see Sect. 2 in Daniel et al. [3]) to obtain $Z_{k+1}$. For this purpose, consider the notation:

$$p_j^{(k+1)} = \begin{cases} \nabla h_j(x_{k+1}), & j = 1, \cdots, m, \\ g_{k+1}, & j = m+1. \end{cases} \tag{3.1}$$

Let $W_1 = Z_k$ and $q_1 = r_k$, where $r_k$ denotes the number of columns of $Z_k$. For $j = 1, \cdots, m+1$, by the reorthogonalization procedure, compute the decomposition

$$p_j^{(k+1)} = W_j u_j^{(k)} + \tau_j^{(k+1)} z_j^{(k+1)}, \tag{3.2}$$

where

$$u_j^{(k)} = W_j^T p_j^{(k+1)}, \qquad z_j^{(k+1)} \perp \operatorname{span}\{W_j\}, \qquad \|z_j^{(k+1)}\|_2 = 1, \tag{3.3}$$

and

$$\tau_j^{(k+1)} = \|(I - W_j W_j^T) p_j^{(k+1)}\|_2 \geqslant 0. \tag{3.4}$$

If $\tau_j^{(k+1)} > 0$, it follows that $p_j^{(k+1)} \notin \operatorname{span}\{W_j\}$, and we set

$$W_{j+1} = \begin{bmatrix} W_j & z_j^{(k+1)} \end{bmatrix} \quad \text{and} \quad q_{j+1} = q_j + 1. \tag{3.5}$$

Otherwise, it follows that $p_j^{(k+1)} \in \operatorname{span}\{W_j\}$, and we set

$$W_{j+1} = W_j \quad \text{and} \quad q_{j+1} = q_j. \tag{3.6}$$

At the end of the loop, we obtain $Z_{k+1} = W_{m+2}$ and $r_{k+1} = q_{m+2}$.

Now, using the data obtained in the calculation of $Z_{k+1}$, we can compute $\bar{g}_{k+1}$, $\bar{A}_{k+1}$ and $\bar{B}_{k+1}$ in a cheaper way. Indeed, from (3.2), (3.3), and the fact that $s_k$, $g_k \in \text{span}\{W_j\}$, it follows that

$$\left(z_j^{(k+1)}\right)^T p_j^{(k+1)} = \tau_j^{(k+1)}, \qquad \left(z_j^{(k+1)}\right)^T s_k = 0, \qquad \left(z_j^{(k+1)}\right)^T g_k = 0. \tag{3.7}$$

If $Z_{k+1} \neq Z_k$, that is, $Z_{k+1} = [Z_k \ \bar{Z}_{k+1}]$, then Lemma 2.3 and Remark 2.2 imply that $B_k \bar{Z}_{k+1} = \sigma \bar{Z}_{k+1}$ and the columns of $B_k Z_k$ belong to $G_k$. Thus, denoting $q = r_{k+1} - r_k$, we get

$$\tilde{s}_k = Z_{k+1}^T s_k = \begin{bmatrix} Z_k^T s_k \\ \bar{Z}_{k+1}^T s_k \end{bmatrix} = \begin{bmatrix} \bar{s}_k \\ 0 \end{bmatrix}, \tag{3.8}$$

$$\begin{aligned} \tilde{B}_k &= Z_{k+1}^T B_k Z_{k+1} = \begin{bmatrix} Z_k^T \\ \bar{Z}_{k+1}^T \end{bmatrix} B_k [Z_k \quad \bar{Z}_{k+1}] \\ &= \begin{bmatrix} Z_k^T \\ \bar{Z}_{k+1}^T \end{bmatrix} [B_k Z_k \quad B_k \bar{Z}_{k+1}] = \begin{bmatrix} Z_k^T \\ \bar{Z}_{k+1}^T \end{bmatrix} [B_k Z_k \quad \sigma \bar{Z}_{k+1}] \\ &= \begin{bmatrix} Z_k^T B_k Z_k & \sigma Z_k^T \bar{Z}_{k+1} \\ \bar{Z}_{k+1}^T B_k Z_k & \sigma \bar{Z}_{k+1}^T \bar{Z}_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{B}_k & 0 \\ 0 & \sigma I_q \end{bmatrix}. \end{aligned} \tag{3.9}$$

To compute $\bar{g}_{k+1}$, from (3.3) and (3.1), note that

$$\begin{aligned} W_{m+1}^T p_{m+1}^{(k+1)} = u_{m+1}^{(k)} \quad &\Longrightarrow \quad W_{m+1}^T g_{k+1} = u_{m+1}^{(k)} \\ &\Longrightarrow \quad [Z_k \quad \tilde{Z}_{k+1}]^T g_{k+1} = u_{m+1}^{(k)} \\ &\Longrightarrow \quad Z_k^T g_{k+1} = \left[\left(u_{m+1}^{(k)}\right)_1 \quad \cdots \quad \left(u_{m+1}^{(k)}\right)_{r_k}\right]^T, \end{aligned} \tag{3.10}$$

where the columns of $\tilde{Z}_{k+1}$ are distinct vectors of the set $\{z_1^{(k+1)}, \cdots, z_{m+1}^{(k+1)}\}$. Further,

$$\begin{aligned} \bar{Z}_{k+1}^T W_{m+1} &= \bar{Z}_{k+1}^T [Z_k \quad \tilde{Z}_{k+1}] \\ &= \left[0 \quad \bar{Z}_{k+1}^T \tilde{Z}_{k+1}\right] \\ &= \begin{cases} \left[\begin{array}{c|c} 0 & I_{q-1} \\ \hline 0 \cdots 0 & 0 \cdots 0 \end{array}\right], & \text{if } \tau_{m+1}^{(k+1)} > 0, \\ \left[0 \quad I_q\right], & \text{otherwise.} \end{cases} \end{aligned} \tag{3.11}$$

Then, multiplying (3.2) from the left by $\bar{Z}_{k+1}$ (with $j = m+1$), we obtain

$$\begin{aligned} \bar{Z}_{k+1}^T g_{k+1} &= \bar{Z}_{k+1}^T W_{m+1} u_{m+1}^{(k)} + \tau_{m+1}^{(k+1)} \bar{Z}_{k+1}^T z_{m+1}^{(k+1)} \\ &= \begin{cases} [(u_{m+1}^{(k)})_{r_k+1} \cdots (u_{m+1}^{(k)})_{r_{k+1}-1} \ \tau_{m+1}^{(k+1)}]^T, & \text{if } \tau_{m+1}^{(k+1)} > 0, \\ [(u_{m+1}^{(k)})_{r_k+1} \cdots (u_{m+1}^{(k)})_{r_{k+1}}]^T, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.12}$$

Hence, combining (3.10) and (3.12), we have

$$\begin{aligned}\bar{g}_{k+1} = Z_{k+1}^T g_{k+1} &= \begin{bmatrix} Z_k^T g_{k+1} \\ \bar{Z}_{k+1}^T g_{k+1} \end{bmatrix} \\ &= \begin{cases} [(u_{m+1}^{(k)})_1 \cdots (u_{m+1}^{(k)})_{r_{k+1}-1}\ \tau_{m+1}^{(k+1)}]^T, & \text{if } \tau_{m+1}^{(k+1)} > 0, \\ [(u_{m+1}^{(k)})_1 \cdots (u_{m+1}^{(k)})_{r_{k+1}}]^T, & \text{otherwise.} \end{cases}\end{aligned} \tag{3.13}$$

By (3.1),

$$\begin{aligned}\bar{A}_{k+1} = Z_{k+1}^T A_{k+1} &= \begin{bmatrix} Z_k^T A_{k+1} \\ \bar{Z}_{k+1}^T A_{k+1} \end{bmatrix} \\ &= \begin{bmatrix} [Z_k^T p_1^{(k+1)} & \cdots & Z_k^T p_m^{(k+1)}] \\ [\bar{Z}_{k+1}^T p_1^{(k+1)} & \cdots & \bar{Z}_{k+1}^T p_m^{(k+1)}] \end{bmatrix}.\end{aligned} \tag{3.14}$$

Thus, denoting

$$\bar{U}_{k+1} = \left[ Z_k^T p_1^{(k+1)} \quad \cdots \quad Z_k^T p_m^{(k+1)} \right] \tag{3.15}$$

and

$$\tilde{U}_{k+1} = \left[ \bar{Z}_{k+1}^T p_1^{(k+1)} \quad \cdots \quad \bar{Z}_{k+1}^T p_m^{(k+1)} \right], \tag{3.16}$$

it follows that

$$\bar{A}_{k+1} = \begin{bmatrix} \bar{U}_{k+1} \\ \tilde{U}_{k+1} \end{bmatrix}. \tag{3.17}$$

Again, by (3.3), for each $j = 1, \cdots, m$,

$$\begin{aligned} W_j^T p_j^{(k+1)} = u_j^{(k)} \quad &\Longrightarrow \quad \left[ Z_k \quad \tilde{Z}_{k+1}^j \right]^T p_j^{(k+1)} = u_j^{(k)} \\ &\Longrightarrow \quad Z_k^T p_j^{(k+1)} = \left[ \left(u_j^{(k)}\right)_1 \quad \cdots \quad \left(u_j^{(k)}\right)_{r_k} \right]^T, \end{aligned} \tag{3.18}$$

where the columns of $\tilde{Z}_{k+1}^j$ are distinct vectors of the set $\{z_1^{(k+1)}, \cdots, z_j^{(k+1)}\}$. Further, multiplying (3.2) from the left by $\bar{Z}_{k+1}$, we obtain

$$\bar{Z}_{k+1}^T p_j^{(k+1)} = \begin{cases} [(u_j^{(k)})_{r_k+1} \cdots (u_j^{(k)})_{q_j} \tau_j^{(k+1)} 0 \cdots 0]^T, & \text{if } \tau_j^{(k+1)} > 0, \\ [(u_j^{(k)})_{r_k+1} \cdots (u_j^{(k)})_{q_j} 0 \cdots 0]^T, & \text{otherwise,} \end{cases} \tag{3.19}$$

for each $j = 1, \cdots, m$, which completes the computation of $\bar{A}_{k+1}$.

Finally, if $y_k = (g_{k+1} - g_k) - (A_{k+1}\lambda_{k+1} - A_k\lambda_k)$ then[1]

$$\tilde{y}_k = Z_{k+1}^T y_k = \begin{bmatrix} Z_k^T y_k \\ \bar{Z}_{k+1}^T y_k \end{bmatrix}$$
$$= \begin{bmatrix} Z_k^T [g_{k+1} - g_k - A_{k+1}\lambda_{k+1} + A_k\lambda_k] \\ \bar{Z}_{k+1}^T [g_{k+1} - g_k - A_{k+1}\lambda_{k+1} + A_k\lambda_k] \end{bmatrix}$$
$$= \begin{bmatrix} Z_k^T g_{k+1} - \bar{g}_k - \bar{U}_{k+1}\lambda_{k+1} + \bar{A}_k\lambda_k \\ \bar{Z}_{k+1}^T g_{k+1} - \tilde{U}_{k+1}\lambda_{k+1} \end{bmatrix}. \tag{3.20}$$

For the case in which $Z_{k+1} = Z_k$, it follows that

$$\tilde{s}_k = Z_k^T s_k = \bar{s}_k, \tag{3.21}$$
$$\tilde{B}_k = Z_k^T B_k Z_k = \bar{B}_k, \tag{3.22}$$
$$\bar{g}_{k+1} = Z_k^T g_{k+1} = \left[ \left(u_{m+1}^{(k)}\right)_1 \quad \cdots \quad \left(u_{m+1}^{(k)}\right)_{r_k} \right]^T, \tag{3.23}$$
$$\bar{A}_{k+1} = Z_k^T A_{k+1} = \bar{U}_{k+1}, \tag{3.24}$$
$$\tilde{y}_k = Z_k^T y_k = \bar{g}_{k+1} - \bar{g}_k - \bar{U}_{k+1}\lambda_{k+1} + \bar{A}_k\lambda_k. \tag{3.25}$$

According to Lemma 2.4, the reduced matrix

$$\bar{B}_{k+1} = Z_{k+1}^T B_{k+1} Z_{k+1}$$

in the subspace span$\{Z_{k+1}\}$ can be obtained by any formula among the damped BFGS, PSB and Broyden family, by use of $\tilde{s}_k$, $\tilde{B}_k$ and $\tilde{y}_k$ computed by (3.8), (3.9), and (3.20), or by (3.21), (3.22), and (3.25). Then, by Theorem 2.1 we can solve the subproblem (2.7)–(2.9) with the reduced matrix $\bar{B}_{k+1}$, the reduced matrix $\bar{A}_{k+1}$ and the reduced gradient $\bar{g}_{k+1}$ to obtain $\bar{s}_{k+1}$ and the trial step $s_{k+1} = Z_{k+1}\bar{s}_{k+1}$.

We summarize the above observations in the following algorithm.

**Algorithm 3.1** (Subspace Version of the Powell–Yuan Algorithm)

Step 0 Given $x_1 \in \mathbb{R}^n$, $\Delta_1 > 0$, $\varepsilon_s > 0$, $\gamma \in [0, 1)$, $\mu_1 > 0$, and $0 < b_2 \leqslant b_1 < 1$, choose one matrix updating formula among the damped BFGS, PSB and Broyden family, and compute $\nabla h_1(x_1), \cdots, \nabla h_m(x_1)$ and $g_1 = \nabla f(x_1)$. Apply the procedure of Gram–Schmidt with reorthogonalization to

$$\{\nabla h_1(x_1), \cdots, \nabla h_m(x_1), g_1\}$$

[1] Similarly, if $y_k = (g_{k+1} - g_k) - (A_{k+1} - A_k)\lambda_k$ then

$$\tilde{y}_k = \begin{bmatrix} Z_k^T g_{k+1} - \bar{g}_k - \bar{U}_{k+1}\lambda_k + \bar{A}_k\lambda_k \\ \bar{Z}_{k+1}^T g_{k+1} - \tilde{U}_{k+1}\lambda_k \end{bmatrix}.$$

in order to obtain a column orthogonal matrix $Z_1 \in \mathbb{R}^{n\times r_1}$ such that

$$\operatorname{span}\{Z_1\} = \operatorname{span}\{\nabla h_1(x_1), \cdots, \nabla h_m(x_1), g_1\}. \tag{3.26}$$

Set $\bar{B}_1 = \sigma I_{r_1}$, $\bar{g}_1 = Z_1^T g_1$, $\bar{A}_1 = Z_1^T A_1$ and $k := 1$.

Step 1 If $\|c_k\|_2 + \|\bar{g}_k - \bar{A}_k\bar{\lambda}_k\|_2 \leqslant \varepsilon_s$ (where $\bar{\lambda}_k = \bar{A}_k^+\bar{g}_k$), then stop. Otherwise, compute $\xi_k$ satisfying (1.9), with $\bar{A}_k$ in place of $A_k$, and solve the CDT subproblem (2.7)–(2.9) to obtain $\bar{s}_k$.

Step 2 Compute $s_k = Z_k\bar{s}_k$ and $D_k$ by (1.12). If the inequality

$$D_k \leqslant \frac{1}{2}\mu_k\left(\left\|c_k + A_k^T s_k\right\|_2^2 - \|c_k\|_2^2\right) \tag{3.27}$$

fails, then increase $\mu_k$ to the value

$$\mu_k^{\text{new}} = 2\mu_k^{\text{old}} + \max\left\{0, \frac{2D_k^{\text{old}}}{\|c_k\|_2^2 - \|c_k + A_k^T s_k\|_2^2}\right\} \tag{3.28}$$

which ensures that the new value of expression (1.12) satisfies condition (3.27).

Step 3 Compute $\rho_k$ by (1.14);
Set $x_{k+1}$ by (1.15);
Set $\Delta_{k+1}$ by (1.16).

Step 4 If $r_k = n$, set $\bar{A}_{k+1} = A_{k+1}$, $\bar{g}_{k+1} = g_{k+1}$, $\tilde{s}_k = s_k$, $\tilde{B}_k = \bar{B}_k$, $\tilde{y}_k = (g_{k+1} - g_k) - (A_{k+1}\lambda_{k+1} - A_k\lambda_k)$, $Z_{k+1} = I_n$, $r_{k+1} = n$ and go to Step 6.

Step 5 Set $W_1 = Z_k$, $q_1 = r_k$, and consider the notation (3.1);
**For** $j = 1 : m+1$
(a) Obtain (3.2) by the reorthogonalization procedure;
(b) If $\tau_j^{(k+1)} > \gamma\|p_j^{(k+1)}\|_2$, set $W_{j+1} = [W_j \quad z_j^{(k+1)}]$ and $q_{j+1} = q_j + 1$. Otherwise, set $W_{j+1} = W_j$ and $q_{j+1} = q_j$.
**End(For).**
Set $Z_{k+1} = W_{m+2}$ and $r_{k+1} = q_{m+2}$;
If $Z_{k+1} \neq Z_k$ compute $\tilde{s}_k$, $\tilde{B}_k$, $\bar{g}_{k+1}$, $\bar{A}_{k+1}$, $\tilde{y}_k$ according to (3.8), (3.9), (3.13), (3.17) and (3.20), respectively. Otherwise, compute $\tilde{s}_k$, $\tilde{B}_k$, $\bar{g}_{k+1}$, $\bar{A}_{k+1}$, $\tilde{y}_k$ by (3.21)–(3.25), respectively.

Step 6 Obtain $\bar{B}_{k+1} = \mathit{Update}(\tilde{B}_k, \tilde{s}_k, \tilde{y}_k)$ by the chosen matrix updating formula. Set $\mu_{k+1} := \mu_k$, $k := k+1$ and go to Step 1.

*Remark 3.1* By Step 4, when the dimension $r_k$ of the subspace $\operatorname{span}\{Z_k\}$ reaches $n$, Algorithm 3.1 reduces to Algorithm 1.1. The reason for this step is to avoid the computational effort required by Step 5, when it is not necessary anymore.

*Remark 3.2* The subspace properties of the CDT subproblem described in Sect. 2 can be used in the same way to construct a subspace version of the CDT trust-region algorithm for equality constrained optimization proposed by Celis, Dennis and Tapia [2], as well of any algorithm based on the CDT subproblem.

In order to compare Algorithms 1.1 and 3.1 with respect to the number of floating point operations per iteration, recall that $n$ denotes the number of variables, $m$ denotes the number of constraints and $r_k$ denotes the number of columns of the matrix $Z_k$. First, let us consider Algorithm 3.1. The computation of $\bar{\lambda}_k$ in Step 1 by Algorithm 5.3.2 in Golub and Van Loan [6] requires $O(m^2 r_k)$ flops. As will be described in Sect. 5, the number $\xi_k$ can be obtained as a solution of an LSQI problem. In this case, the computation of $\xi_k$ in Step 1 by Algorithm 12.1.1 in Golub and Van Loan [6] requires approximately $O(mr_k^2) + O(r_k)$ flops (see p. 208 in Bjorck [1]). Still in the Step 1, the computation of a solution of the CDT subproblem (2.7)–(2.9) by the dual algorithm of Yuan [16] requires about $O(r_k^3) + O(r_k^2) + O(r_k)$ flops.[2] The computation of $s_k = Z_k \bar{s}_k$ in Step 2 requires $O(nr_k)$ flops. The reorthogonalization procedure in Step 5 requires about $O((m+1)nr_k) + O(mn) + O(n)$ flops. Finally, the update $\bar{B}_{k+1}$ of $\bar{B}_k$ in Step 6 requires about $O(r_k^2) + O(r_k)$ flops. Therefore, Algorithm 3.1 requires approximately

$$\begin{aligned} &O\big(r_k^3\big) + O\big(mr_k^2\big) + O\big(r_k^2\big) + O\big(m^2 r_k\big) + O(r_k) + O(nr_k) + O\big((m+1)nr_k\big) \\ &\quad + O(mn) + O(n) \end{aligned}$$

flops for each iteration (after the first one). The Algorithm 1.1, by its turn, requires approximately

$$O\big(n^3\big) + O\big(mn^2\big) + O\big(n^2\big) + O\big(m^2 n\big) + O(n)$$

flops for each iteration, with the same update formula for $B_k$. Thus, when $n$ is large, $m$ is small and $r_k \ll n$, the Algorithm 3.1 can reduce the amount of computation in comparison with the Algorithm 1.1.

## 4 Global Convergence

If we suppose that $G_k = \text{span}\{Z_k\}$ and $\xi_1 > \min_{\|d\|_2 \leqslant \Delta_1} \|c_1 + A_1^T d\|_2$ then, by Theorem 2.1 and Lemma 2.4, Algorithm 3.1 is equivalent to Algorithm 1.1. As pointed in Remark 3.1, the same is true from the moment in which $r_k$ reaches $n$. In both cases the global convergence of the Algorithm 3.1 follows from the fact that the Algorithm 1.1 is globally convergent (see Theorem 3.9 in Powell and Yuan [11]). In this section, we shall study the convergence of Algorithm 3.1 in a more general setting, allowing more freedom for the choice of the matrix $Z_k$ in Step 5. Specifically, we consider the assumptions:

- A1 The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $h_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \cdots, m)$ are continuously differentiable;
- A2 There exists a compact and convex set $\Omega \in \mathbb{R}^n$ such that $x_k$ and $x_k + s_k$ are in $\Omega$ for all $k$;
- A3 $A(x)$ has full column rank for all $x \in \Omega$;

[2]This estimates is obtained if we assume a maximum number of iterations for the algorithm and that the numbers $I(k)$ in its Step 7 are bounded from above (see Algorithm 3.1 in Yuan [16]).

A4 For each $k$, $Z_k^T Z_k = I_{r_k}$, $\{\nabla h_1(x_k), \cdots, \nabla h_m(x_k), g_k\} \subset \text{span}\{Z_k\}$ and $B_k z \in \text{span}\{Z_k\}$ for all $z \in \text{span}\{Z_k\}$.
A5 The sequence $(\|\bar{B}_k\|_2)_{k\in\mathbb{N}}$ is bounded.

We also consider the following remark, which will be extensively called in the proofs.

*Remark 4.1* From $Z_k^T Z_k = I_{r_k}$, it follows that

$$v \in \text{span}\{Z_k\} \quad \Longrightarrow \quad v = Z_k Z_k^T v. \tag{4.1}$$

**Lemma 4.1** *Suppose that* A1–A4 *hold. Then, the sequence* $(\|\bar{A}_k^+\|_2)_{k\in\mathbb{N}}$ *is bounded.*

*Proof* By A1 and A2, there exists $\kappa_1 > 0$ such that

$$\|A_k\|_2 \leqslant \kappa_1, \quad \text{for all } k. \tag{4.2}$$

On the other hand, given $x \in \mathbb{R}^m$, by A4 we have $A_k x \in \text{span}\{Z_k\}$, and from Remark 4.1 it follows that

$$\begin{aligned}\|\bar{A}_k x\|_2^2 &= \left\|Z_k^T A_k x\right\|_2^2 \\ &= \left(Z_k^T A_k x\right)^T \left(Z_k^T A_k x\right) \\ &= (A_k x)^T Z_k Z_k^T A_k x \\ &= (A_k x)^T A_k x \\ &= \|A_k x\|_2^2.\end{aligned} \tag{4.3}$$

Hence,

$$\|\bar{A}_k\|_2 = \max_{\|x\|_2=1} \|\bar{A}_k x\|_2 = \max_{\|x\|_2=1} \|A_k x\|_2 = \|A_k\|_2 \leqslant \kappa_1, \quad \text{for all } k, \tag{4.4}$$

and, consequently, there exists $\kappa_2 > 0$ such that

$$\left\|\bar{A}_k^T \bar{A}_k\right\|_2 \leqslant \kappa_2, \quad \text{for all } k. \tag{4.5}$$

Now, since $\{\nabla h_1(x_k), \cdots, \nabla h_m(x_k)\} \subset \text{span}\{Z_k\}$, from Remark 4.1 it follows that

$$A_k = Z_k Z_k^T A_k. \tag{4.6}$$

Thus,

$$\bar{A}_k^T \bar{A}_k = \left(Z_k^T A_k\right)^T \left(Z_k^T A_k\right) = A_k^T Z_k Z_k^T A_k = A_k^T A_k, \tag{4.7}$$

and, by A3, the matrix $\bar{A}_k^T \bar{A}_k$ is invertible. This implies that $\bar{A}_k$ has full column rank and, therefore,

$$\bar{A}_k^+ = \left(\bar{A}_k^T \bar{A}_k\right)^{-1} \bar{A}_k^T. \tag{4.8}$$

Let $GL(n, \mathbb{R})$ be the set of $n \times n$ invertible matrices of real numbers. It is well known that the matrix inversion $\varphi : GL(n, \mathbb{R}) \to GL(n, \mathbb{R})$ defined by $\varphi(M) = M^{-1}$ is a continuous function (see, e.g., Theorem 2.3.4 in Golub and Van Loan [6]). Hence, by (4.5), there exists $\kappa_3 > 0$ such that

$$\|(\bar{A}_k^T \bar{A}_k)^{-1}\| \leqslant \kappa_3, \quad \text{for all } k. \tag{4.9}$$

Finally, by (4.8), (4.9), and (4.4), there exists $\kappa_4 > 0$ such that

$$\|\bar{A}_k^+\| \leqslant \kappa_4, \quad \text{for all } k, \tag{4.10}$$

and the proof is complete. □

**Lemma 4.2** *The inequality*

$$\|c_k\|_2 - \|c_k + A_k^T s_k\|_2 \geqslant \min\left\{\|c_k\|_2, \frac{b_2 \Delta_k}{\|\bar{A}_k^+\|_2}\right\} \tag{4.11}$$

*holds for all $k$, where $b_2$ is introduced in (1.9).*

*Proof* By following the same argument as in the proof of Lemma 3.3 in Powell and Yuan [11], we conclude that the inequality

$$\|c_k\|_2 - \|c_k + \bar{A}_k^T \bar{s}_k\|_2 \geqslant \min\left\{\|c_k\|_2, \frac{b_2 \Delta_k}{\|\bar{A}_k^+\|_2}\right\} \tag{4.12}$$

holds for all $k$. Since $s_k = Z_k \bar{s}_k \in \text{span}\{Z_k\}$, it follows from Remark 4.1 that $s_k = Z_k Z_k^T s_k$, and then

$$\bar{A}_k^T \bar{s}_k = (Z_k^T A_k)^T Z_k^T s_k = A_k^T Z_k Z_k^T s_k = A_k^T s_k. \tag{4.13}$$

Now, by replacing (4.13) in (4.12) we obtain (4.11). □

**Lemma 4.3** *There exists a positive constant $m_1$ such that the inequality*

$$\begin{aligned} D_k &+ \frac{1}{2}\mu_k\left(\|c_k\|_2^2 - \|c_k + A_k^T s_k\|_2^2\right) \\ &\leqslant -\frac{1}{4}\|P_k g_k^*\|_2^2 \min\left\{\frac{1}{2\|\bar{B}_k\|_2}, \frac{\Delta_k^*}{\|P_k g_k^*\|_2}\right\} + m_1 \|s_k\|_2 \|c_k\|_2 \\ &\quad - \frac{1}{2}\mu_k \|c_k\|_2 \min\left\{\|c_k\|_2, \frac{b_2 \Delta_k}{\|\bar{A}_k^+\|_2}\right\} \end{aligned} \tag{4.14}$$

*holds for all $k$, where $D_k$ is given by (1.12) and we use the notation*

$$g_k^* = g_k + B_k s_k^*, \tag{4.15}$$

$$\Delta_k^* = \left(\Delta_k^2 - \|s_k^*\|_2^2\right)^{\frac{1}{2}}, \tag{4.16}$$

$$s_k^* = (I_n - P_k) s_k, \tag{4.17}$$

$$P_k = I_n - A_k A_k^+. \tag{4.18}$$

*Proof* By following the same argument as in the proof of Lemma 3.4 in Powell and Yuan [11], we conclude that there exists a positive constant $m_1$ for which the inequality

$$\begin{aligned}
&\tilde{D}_k + \frac{1}{2}\mu_k\left(\|c_k\|_2^2 - \left\|c_k + \bar{A}_k^T\bar{s}_k\right\|_2^2\right) \\
&\quad\leqslant -\frac{1}{4}\|\tilde{P}_k\tilde{g}_k\|_2^2\min\left\{\frac{1}{2\|\bar{B}_k\|_2}, \frac{\tilde{\Delta}_k}{\|\tilde{P}_k\tilde{g}_k\|_2}\right\} + m_1\|\bar{s}_k\|_2\|c_k\|_2 \\
&\qquad - \frac{1}{2}\mu_k\|c_k\|_2\min\left\{\|c_k\|_2, \frac{b_2\Delta_k}{\|\bar{A}_k^+\|_2}\right\}
\end{aligned} \tag{4.19}$$

holds for all $k$, where

$$\begin{aligned}
\tilde{D}_k &= (\bar{g}_k - \bar{A}_k\bar{\lambda}_k)^T\bar{s}_k + \frac{1}{2}\check{s}_k^T\bar{B}_k\check{s}_k - [\bar{\lambda}_{k+1} - \bar{\lambda}_k]^T\left(c_k + \frac{1}{2}\bar{A}_k^T\bar{s}_k\right) \\
&\quad + \mu_k\left(\left\|c_k + \bar{A}_k^T\bar{s}_k\right\|_2^2 - \|c_k\|_2^2\right),
\end{aligned} \tag{4.20}$$

$$\bar{\lambda}_k = \bar{A}_k^+\bar{g}_k, \tag{4.21}$$

$$\check{s}_k = \tilde{P}_k\bar{s}_k, \tag{4.22}$$

$$\tilde{P}_k = I_{r_k} - \bar{A}_k\bar{A}_k^+, \tag{4.23}$$

$$\tilde{g}_k = \bar{g}_k + \bar{B}_k\tilde{s}_k, \tag{4.24}$$

$$\tilde{\Delta}_k = \left(\Delta_k^2 - \|\tilde{s}_k\|_2^2\right)^{\frac{1}{2}}, \tag{4.25}$$

$$\tilde{s}_k = (I_{r_k} - \tilde{P}_k)\bar{s}_k. \tag{4.26}$$

From (4.13) we have

$$\left\|c_k + \bar{A}_k^T\bar{s}_k\right\|_2 = \left\|c_k + A_k^Ts_k\right\|_2. \tag{4.27}$$

We shall prove that

$$\tilde{D}_k = D_k, \qquad \tilde{\Delta}_k = \Delta_k^*, \qquad \left\|\tilde{P}_k\tilde{g}_k\right\|_2 = \left\|P_kg_k^*\right\|_2, \quad \text{and} \quad \|\bar{s}_k\|_2 = \|s_k\|_2. \tag{4.28}$$

Then, (4.14) will follow directly from (4.19). Since $s_k = Z_k\bar{s}_k$ and $g_k$ belong to $\text{span}\{Z_k\}$, from Remark 4.1 it follows that

$$s_k = Z_kZ_k^Ts_k, \tag{4.29}$$

$$g_k = Z_kZ_k^Tg_k. \tag{4.30}$$

Moreover, recalling the definitions of $g_k^*$, $s_k^*$, $\hat{s}_k$ and $P_k$ (in (4.15), (4.17), (1.13) and (4.18), respectively) and assumption A4, we see that $\{g_k^*, s_k^*, \hat{s}_k, P_kg_k^*\} \subset \text{span}\{Z_k\}$. Consequently, by Remark 4.1,

$$g_k^* = Z_kZ_k^Tg_k^*, \tag{4.31}$$

$$s_k^* = Z_k Z_k^T s_k^*, \tag{4.32}$$

$$\hat{s}_k = Z_k Z_k^T \hat{s}_k, \tag{4.33}$$

$$P_k g_k^* = Z_k Z_k^T P_k g_k^*. \tag{4.34}$$

From (4.21), (4.8), (4.7), and (4.30), it follows that

$$\begin{aligned}\bar{\lambda}_k &= \bar{A}_k^+ \bar{g}_k \\ &= \left(\bar{A}_k^T \bar{A}_k\right)^{-1} \bar{A}_k^T \bar{g}_k \\ &= \left(A_k^T A_k\right)^{-1} A_k^T Z_k Z_k^T g_k \\ &= \left(A_k^T A_k\right)^{-1} A_k^T g_k \\ &= A_k^+ g_k \\ &= \lambda_k. \end{aligned} \tag{4.35}$$

By (4.35) and (4.29) we obtain

$$\begin{aligned}(\bar{g}_k - \bar{A}_k \bar{\lambda}_k)^T \bar{s}_k &= \left(Z_k^T g_k - Z_k^T A_k \lambda_k\right)^T Z_k^T s_k \\ &= (g_k - A_k \lambda_k)^T Z_k Z_k^T s_k \\ &= (g_k - A_k \lambda_k)^T s_k. \end{aligned} \tag{4.36}$$

Further, by (4.22), (4.23), (4.8), (4.7), (4.29), and (1.13),

$$\begin{aligned}\check{s}_k &= \tilde{P}_k \bar{s}_k \\ &= \left(I_{r_k} - \bar{A}_k \bar{A}_k^+\right)\bar{s}_k \\ &= \bar{s}_k - \bar{A}_k \left(\bar{A}_k^T \bar{A}_k\right)^{-1} \bar{A}_k^T \bar{s}_k \\ &= Z_k^T s_k - Z_k^T A_k \left(A_k^T A_k\right)^{-1} A_k^T Z_k \bar{s}_k \\ &= Z_k^T \left(s_k - A_k \left(A_k^T A_k\right)^{-1} A_k^T s_k\right) \\ &= Z_k^T \left[\left(I_n - A_k A_k^+\right) s_k\right] \\ &= Z_k^T P_k s_k \\ &= Z_k^T \hat{s}_k. \end{aligned} \tag{4.37}$$

Note that the equalities (4.37), (4.29), and (4.33) imply that

$$\bar{s}_k^T \bar{B}_k \check{s}_k = \bar{s}_k \bar{B}_k Z_k^T \hat{s}_k$$

$$
\begin{aligned}
&= (Z_k^T s_k)^T \big(Z_k^T B_k Z_k\big) Z_k^T \hat{s}_k \\
&= \big(s_k^T Z_k Z_k^T\big) B_k \big(Z_k Z_k^T \hat{s}_k\big) \\
&= \big(Z_k Z_k^T s_k\big)^T B_k \big(Z_k Z_k^T \hat{s}_k\big) \\
&= s_k^T B_k \hat{s}_k. \qquad (4.38)
\end{aligned}
$$

Now, by (4.36), (4.38), (4.35), (4.13), and (4.27), we conclude that

$$\tilde{D}_k = D_k. \qquad (4.39)$$

From (4.26), (4.23), (4.8), (4.7), (4.29), (4.18), and (4.17) it follows that

$$
\begin{aligned}
\tilde{s}_k &= \bar{A}_k \bar{A}_k^+ \bar{s}_k \\
&= \bar{A}_k \big(\bar{A}_k^T \bar{A}_k\big)^{-1} \bar{A}_k^T \bar{s}_k \\
&= Z_k^T A_k \big(A_k^T A_k\big)^{-1} A_k^T Z_k Z_k^T s_k \\
&= Z_k^T A_k \big(A_k^T A_k\big)^{-1} A_k^T s_k \\
&= Z_k^T A_k A_k^+ s_k \\
&= Z_k^T \big[(I_n - P_k) s_k\big] \\
&= Z_k^T s_k^*. \qquad (4.40)
\end{aligned}
$$

Then, by (4.32),

$$\|\tilde{s}_k\|_2^2 = \big\|Z_k^T s_k^*\big\|_2^2 = \big(s_k^*\big)^T Z_k Z_k^T s_k^* = \big(s_k^*\big)^T s_k^* = \big\|s_k^*\big\|_2^2, \qquad (4.41)$$

which implies that

$$\tilde{\Delta}_k = \big(\Delta_k^2 - \|\tilde{s}_k\|_2^2\big)^{\frac{1}{2}} = \big(\Delta_k^2 - \big\|s_k^*\big\|_2^2\big)^{\frac{1}{2}} = \Delta_k^*. \qquad (4.42)$$

On the other hand, from (4.24), (4.40), (4.32), and (4.15) it follows that

$$
\begin{aligned}
\tilde{g}_k &= \bar{g}_k + \bar{B}_k \tilde{s}_k \\
&= Z_k^T g_k + Z_k^T B_k Z_k Z_k^T s_k^* \\
&= Z_k^T \big(g_k + B_k s_k^*\big) \\
&= Z_k^T g_k^*. \qquad (4.43)
\end{aligned}
$$

Thus, by (4.23), (4.8), (4.7), (4.31), and (4.18),

$$
\begin{aligned}
\tilde{P}_k \tilde{g}_k &= \big(I_{r_k} - \bar{A}_k \bar{A}_k^+\big) \tilde{g}_k \\
&= \big(I_{r_k} - Z_k^T A_k \big(A_k^T A_k\big)^{-1} A_k^T Z_k\big) Z_k^T g_k^*
\end{aligned}
$$

$$
\begin{aligned}
&= Z_k^T \left[ g_k^* - A_k A_k^+ g_k^* \right] \\
&= Z_k^T P_k g_k^*. 
\end{aligned} \tag{4.44}
$$

Now, equalities (4.44) and (4.34) imply that

$$
\begin{aligned}
\|\tilde{P}_k \tilde{g}_k\|_2^2 &= \left\| Z_k^T P_k g_k^* \right\|_2^2 \\
&= \left(P_k g_k^*\right)^T Z_k Z_k^T P_k g_k^* \\
&= \left(P_k g_k^*\right)^T \left(P_k g_k^*\right) \\
&= \left\| P_k g_k^* \right\|_2^2 \\
\Longrightarrow \quad \|\tilde{P}_k \tilde{g}_k\|_2 &= \left\| P_k g_k^* \right\|_2.
\end{aligned} \tag{4.45}
$$

Finally, by (4.29),

$$
\begin{aligned}
&\|\bar{s}_k\|_2^2 = \left\| Z_k^T s_k \right\|_2^2 = s_k^T Z_k Z_k^T s_k = s_k^T s_k = \|s_k\|_2^2 \\
&\Longrightarrow \quad \|\bar{s}_k\|_2 = \|s_k\|_2.
\end{aligned} \tag{4.46}
$$

Hence, by (4.39), (4.27), (4.45), (4.42), and (4.46), the inequality (4.19) reduces to the inequality (4.14) and the proof is complete. □

**Theorem 4.1** *Suppose that* A1–A5 *hold. Then, Algorithm* 3.1 *will terminate after finitely many iterations. In other words, if we remove the convergence test from Step* 1, *then* $s_k = 0$ *for some k or the limit*

$$
\liminf_{k \to \infty} \left[ \|c_k\|_2 + \|P_k g_k\|_2 \right] = 0 \tag{4.47}
$$

*is obtained, which ensures that* $\{x_k\}$ *is not bounded away from stationary points of the problem* (1.1)–(1.2).

*Proof* It follows from Lemmas 4.1, 4.2 and 4.3 by the same argument as in Powell and Yuan [11]. □

*Remark 4.2* By Theorem 4.1, the Algorithm 3.1 is globally convergent for any subspace $S_k = \text{span}\{Z_k\}$ such that $Z_k$ satisfies A4.

## 5 Numerical Results

In order to investigate the proposed algorithm from a computational point of view, and to explore its potentialities and limitations, we have tested MATLAB implementations of Algorithms 1.1 and 3.1 on a set of 50 problems from CUTEr collection [8]. The dimension of the problems varies from 3 to 1498, while the number of constraints are between 1 and 96. Here, we refer to our implementations of Algorithms 1.1 and 3.1 as "PYtr" and "SPYtr", respectively. No attempt is made to compare either of the codes with other solvers.

In both implementations, the CDT subproblem is solved by the dual algorithm proposed by Yuan [16], with the parameters $s_0 = 1$, $\upsilon = 0.001$ and $\varepsilon = 10^{-12}$. In this algorithm, instead of update $M_k$ by the rule

$$M_k = \max\{M_{k-1}, d^T H^{-1} d + y^T H^{-1} y\},$$

we use

$$M_k = d^T H^{-1} d + y^T H^{-1} y,$$

since the latter rule allowed a faster convergence in the numerical tests (see Algorithm 3.1 in [16]). Moreover, the maximum number of iterations for this algorithm was fixed as 200.

To find a value of $\xi_k$ in the interval (1.9), the LSQI problem

$$\begin{aligned} &\min \|c_k + A_k^T d\|_2, \\ &\text{s.t.} \quad \|d\|_2 \leqslant b_1 \Delta_k, \end{aligned}$$

is solved by Algorithm 12.1.1 described in Golub and Van Loan [6], which provides a solution $d_k$. Then, $\xi_k$ is taken as

$$\xi_k = \|c_k + A_k^T d_k\|_2.$$

For both implementations, the parameters in Step 0 are chosen as $\Delta_1 = 1$, $\varepsilon_s = 10^{-4}$, $\mu_1 = 1$, $\gamma = 10^{-8}$ and $b_1 = b_2 = 0.9$. Therefore, each implementation was terminated when $\|c_k\|_2 + \|g_k - A_k \lambda_k\|_2 \leqslant 10^{-4}$. The initial matrix $B_1$ is chosen as the identity matrix and $B_k$ is updated by the damped BFGS formula of Powell [10], namely

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\eta_k \eta_k^T}{s_k^T \eta_k},$$

where

$$s_k = x_{k+1} - x_k, \qquad \eta_k = \theta_k y_k + (1 - \theta_k) B_k s_k,$$

and

$$\theta_k = \begin{cases} 1, & \text{if } s_k^T y_k \geqslant 0.2 s_k^T B_k s_k \\ 0.8 s_k^T B_k s_k / [s_k^T B_k s_k - s_k^T y_k], & \text{otherwise.} \end{cases}$$

The algorithms were coded in MATLAB language, and the tests were performed with MATLAB 7.8.0 (R2009a), on an PC with a 2.53 GHz Intel(R) i3 microprocessor, and using a Ubunto virtual machine with memory limited to 896 MB.

Problems and results are given in Table 1, where "Itr" represents the number of iterations, "Time" represents the CPU time (in seconds), "$n$" represents the number of variables, "$m$" represents the number of constraints, and an entry "F" indicates that the code stopped due some error during the solution of the CDT subproblem.

**Table 1** Numerical results for CUTEr problems

| Problem | Dim | | PYtr | | SPYtr | |
|---|---|---|---|---|---|---|
| | $n$ | $m$ | Itr | Time | Itr | Time |
| ALLINITC* | 4 | 1 | 10 | 0.7 sec | F | F |
| BT3 | 5 | 3 | 9 | 0.7 sec | 11 | 0.8 sec |
| BT6 | 5 | 2 | 14 | 0.9 sec | 10 | 0.7 sec |
| BT8 | 5 | 2 | 94 | 8.8 sec | 111 | 9.8 sec |
| BT9 | 4 | 2 | 30 | 1.9 sec | 35 | 1.9 sec |
| BT11 | 5 | 3 | 9 | 0.6 sec | 18 | 1.3 sec |
| BT12 | 5 | 3 | 19 | 1.6 sec | 63 | 3.8 sec |
| HS21MOD | 7 | 1 | 4 | 0.3 sec | 4 | 0.4 sec |
| HS26 | 3 | 1 | 16 | 0.9 sec | 17 | 0.9 sec |
| HS27 | 3 | 1 | 32 | 1.9 sec | 34 | 1.7 sec |
| HS28 | 3 | 1 | 8 | 0.6 sec | 10 | 0.8 sec |
| HS29* | 3 | 1 | 10 | 0.7 sec | 9 | 0.7 sec |
| HS30* | 3 | 1 | 3 | 0.4 sec | 3 | 0.5 sec |
| HS31* | 3 | 1 | 17 | 1.3 sec | 9 | 0.9 sec |
| HS35* | 3 | 1 | 6 | 0.5 sec | 6 | 0.6 sec |
| HS36 | 3 | 1 | F | F | 8 | 0.4 sec |
| HS39 | 4 | 2 | 30 | 1.7 sec | 35 | 1.8 sec |
| HS42 | 4 | 2 | 7 | 0.8 sec | 5 | 0.6 sec |
| HS46 | 5 | 2 | 12 | 0.8 sec | 10 | 0.7 sec |
| HS47 | 5 | 3 | 22 | 1.1 sec | 17 | 0.9 sec |
| HS48 | 5 | 2 | 9 | 0.7 sec | 10 | 0.8 sec |
| HS49 | 5 | 2 | 19 | 1.1 sec | 24 | 1.3 sec |
| HS50 | 5 | 3 | 15 | 1.1 sec | 14 | 1.0 sec |
| HS51 | 5 | 3 | 7 | 0.6 sec | 6 | 0.7 sec |
| HS52 | 5 | 3 | 9 | 0.7 sec | 13 | 1.0 sec |
| HS53* | 5 | 3 | 8 | 0.7 sec | 9 | 0.8 sec |
| HS54* | 6 | 1 | 2 | 0.3 sec | 2 | 0.3 sec |
| HS56 | 7 | 4 | 15 | 1.6 sec | 16 | 1.5 sec |
| HS60* | 3 | 1 | 11 | 0.7 sec | 16 | 1.4 sec |
| HS65* | 3 | 1 | 25 | 2.4 sec | 25 | 2.4 sec |
| HS77 | 5 | 2 | 15 | 0.9 sec | 9 | 0.7 sec |
| HS78 | 5 | 3 | 9 | 0.7 sec | 6 | 0.6 sec |
| HS79 | 5 | 3 | 10 | 0.5 sec | 9 | 0.5 sec |
| HS80* | 5 | 3 | 5 | 0.4 sec | 5 | 0.5 sec |
| HS100LNP | 7 | 4 | 18 | 1.9 sec | 20 | 2.3 sec |
| DECONVC* | 61 | 1 | 129 | 28.0 sec | 129 | 21.2 sec |
| DUAL1* | 85 | 1 | 244 | 94.0 sec | 293 | 103.4 sec |
| DUAL2* | 96 | 1 | 104 | 43.8 sec | 104 | 21.4 sec |
| DUAL3* | 111 | 1 | 120 | 64.3 sec | 113 | 23.4 sec |
| DUAL4* | 75 | 1 | 52 | 16.7 sec | 52 | 7.1 sec |

**Table 1** (*Continued*)

| Problem | Dim | | PYtr | | SPYtr | |
|---|---|---|---|---|---|---|
| | $n$ | $m$ | Itr | Time | Itr | Time |
| FCCU* | 19 | 8 | 35 | 3.3 sec | 20 | 1.8 sec |
| GENHS28 | 10 | 8 | 6 | 0.6 sec | 7 | 0.7 sec |
| HIMMELBI* | 100 | 12 | 38 | 6.4 sec | 38 | 2.1 sec |
| HS111LNP | 10 | 3 | 48 | 2.7 sec | F | F |
| ORTHREGB | 27 | 6 | 7 | 0.4 sec | 9 | 0.6 sec |
| PORTFL1* | 12 | 1 | 59 | 1.8 sec | 65 | 2.2 sec |
| PRIMAL4* | 1498 | 75 | 3 | 605.2 sec | 3 | 1.1 sec |
| STEENBRA* | 432 | 96 | F | F | 84 | 34.3 sec |
| ZAMB2-8* | 138 | 48 | 784 | 388.9 sec | 775 | 265.3 sec |
| ZAMB2-9* | 138 | 48 | 914 | 530.5 sec | 733 | 227.2 sec |

The asterisk indicates that the original CUTEr problem has been modified for our case, for example, inequalities constraints may have been considered as equalities, or the bounds on the variables may have been ignored. We report only the number of iterations Itr because the number of evaluations of $f(x)$, $c(x)$, $g(x)$ and $A(x)$ is equal to Itr+1 in both algorithms. For each problem in which both codes were successful, the optimal objective function values obtained were the same.

To facilitate comparison between the two algorithms, we use the performance profile proposed by Dolan and Moré [4]. This tool for benchmarking and comparing optimization softwares works in the following way. Let $t_{p,s}$ denote the time to solve problem $p$ by solver $s$. The performance ratio is defined as

$$r_{p,s} = \frac{t_{p,s}}{t_p^*},$$

where $t_p^*$ is the lowest time required by any solver to solve problem $p$. Therefore, $r_{p,s} \geqslant 1$ for all $p$ and $s$. If a solver does not solve a problem, the ratio $r_{p,s}$ is assigned a large number $r_M$, which satisfies $r_{p,s} < r_M$ for all $p, s$ where solver $s$ succeeds in solving problem $p$. The performance profile for each code $s$ is defined as the cumulative distribution function for the performance ratio $r_{p,s}$, which is

$$\rho_s(\tau) = \frac{\text{no. of problems s.t. } r_{p,s} \leqslant \tau}{\text{total no. of problems}}.$$

If $\tau = 1$, then $\rho_s(1)$ represents the percentage of problems for which the solver $s$'s runtime is the best. The performance profile can also be used to analyze the number of iterations required to satisfy the stopping criteria.

Based on the numerical results in Table 1, we give the performance profile for the codes PYtr and SPYtr considering two distinct subsets of problems. The first one corresponds to the first 35 problems in Table 1 (for which $n < 10$), while the second subset corresponds to the remaining 15 problems (for which $n \geqslant 10$). The performance profiles in Fig. 1 for the first subset of problems show that PYtr is slightly

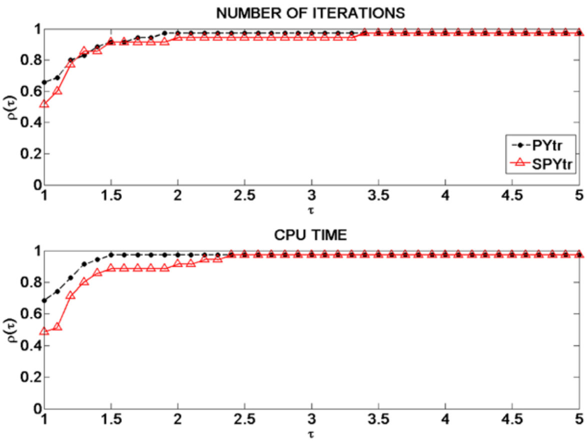


**Fig. 1** Performance profiles for problems with $n < 10$

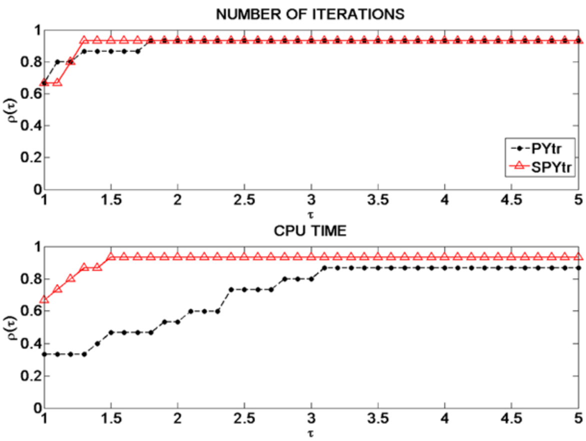


**Fig. 2** Performance profiles for problems with $n \geqslant 10$

more efficient than SPYtr with respect to the number of iterations and the computational time required to reduce the stationarity measure below $\varepsilon_s$. Regarding the computational time, this result is not surprising, since in the problems considered the gap between $n$ and $m$ is very small. In this case, the trial step is computed on the subspaces only in very few iterations, and the time saved in this computation is not enough to compensate the time consumed in the reorthogonalization procedure.

On the other hand, the performance profiles in Fig. 2 show a different picture for the second subset of problems, which includes medium size instances where $n \gg m$. For these problems, both codes require almost the same number of iterations, but SPYtr is significantly faster than PYtr.

## 6 Conclusion and Future Research

Based on subspace properties of the CDT subproblem, we have presented a subspace version of the Powell–Yuan trust-region algorithm for equality constrained optimization. Under suitable conditions, the new algorithm is proved to be globally convergent. Preliminary numerical experiments indicate that the subspace algorithm outperforms its "full space" counterpart on problems where the number of constraints is much lower than the number of variables. Future research include the conducting of extensive numerical tests using more sophisticated implementations, and the development of a strategy to control the size of the subspaces, similar that one proposed by Gong [7] for unconstrained optimization. Further, it is worth to mention that the subspace properties of the CDT subproblem derived in this work can be used to develop subspace versions of any algorithm based on the CDT subproblem, such as the algorithm of Celis, Dennis and Tapia [2].

**Acknowledgements** This work was carried out while the first author was visiting Institute of Computational Mathematics and Scientific/Engineering Computing of the Chinese Academy of Sciences. He would like to thank Professor Ya-xiang Yuan, Professor Yu-hong Dai, Dr. Xin Liu and Dr. Ya-feng Liu for their warm hospitality. The authors also are grateful to Dr. Wei Leng for his help in installing and configuring the CUTEr. Finally, the authors would like to thank the two referees for their helpful comments.

## References

[1] Bjorck, A.: Numerical Methods for Least Square Problems. SIAM, Philadelphia (1996)
[2] Celis, M.R., Dennis, J.E., Tapia, R.A.: A trust region strategy for nonlinear equality constrained optimization. In: Boggs, P.T., Byrd, R.H., Schnabel, R.B. (eds.) Numerical Optimization, pp. 71–82. SIAM, Philadelphia (1985)
[3] Daniel, J.W., Gragg, W.B., Kaufman, L., Stewart, G.W.: Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. Math. Comput. **30**, 772–795 (1976)
[4] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
[5] Gill, P.E., Leonard, M.W.: Reduced-Hessian quasi-Newton methods for unconstrained optimization. SIAM J. Optim. **12**, 209–237 (2001)
[6] Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
[7] Gong, L.: A trust region subspace method for large-scale unconstrained optimization. Asia-Pac. J. Oper. Res. **29**, 1250021 (2012)
[8] Gould, N.I.M., Orban, D., Toint, Ph.L.: CUTEr and SifDec: a constrained and unconstrained testing environment, revisited. ACM Trans. Math. Softw. **29**, 373–394 (2003)
[9] Li, G., Yuan, Y.: Compute a Celis–Dennis–Tapia step. J. Comput. Math. **23**, 463–478 (2005)
[10] Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G.A. (ed.) Numerical Analysis, pp. 144–157. Springer, Berlin (1978)
[11] Powell, M.J.D., Yuan, Y.: A trust region algorithm for equality constrained optimization. Math. Program. **49**, 189–211 (1991)
[12] Siegel, D.: Implementing and modifying Broyden class updates for large scale optimization. Report DAMPT 1992/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England (1992)
[13] Sun, W., Yuan, Y.: Optimization Theory and Methods: Nonlinear Programming. Springer, Berlin (2006)
[14] Wang, Z.H., Yuan, Y.: A subspace implementation of quasi-Newton trust region methods for unconstrained optimization. Numer. Math. **104**, 241–269 (2006)
[15] Yuan, Y.: On a subproblem of trust region algorithms for constrained optimization. Math. Program. **47**, 53–63 (1990)
[16] Yuan, Y.: A dual algorithm for minimizing a quadratic function with two quadratic constraints. J. Comput. Math. **9**, 348–359 (1991)
[17] Zhang, Y.: Computing a Celis–Dennis–Tapia trust region step for equality constrained optimization. Math. Program. **55**, 109–124 (1992)