Computational Optimization and Applications (2019) 74:177–194
https://doi.org/10.1007/s10589-019-00109-6

A subspace SQP method for equality constrained optimization

Jae Hwa Lee[1] · Yoon Mo Jung[2] · Ya-xiang Yuan[3] · Sangwoon Yun[4]

Received: 18 October 2018 / Published online: 16 May 2019

Abstract
In this paper, we present a subspace method for solving large scale nonlinear equality constrained optimization problems. The proposed method is based on a SQP method combined with the limited-memory BFGS update formula. Each subproblem is solved in a theoretically suitable subspace. In the case of few constraints, we show that our search direction in the subspace is equivalent to that of the SQP subproblem in the full space. In the case of many constraints, we reduce the number of constraints in the subproblem and we show that the solution of the subspace subproblem is a descent direction of a particular exact penalty function. Global convergence properties of the proposed method are given for both cases. Numerical results are given to illustrate the soundness of the proposed model.

Keywords Equality constrained optimization · SQP method · Large scale problems · Subspace techniques · Damped limited-memory BFGS update

Mathematics Subject Classification 90C30 · 90C55 · 90C06 · 65K05

✉ Sangwoon Yun
yswmathedu@skku.edu

Jae Hwa Lee
jhlee2chn@skku.edu

Yoon Mo Jung
yoonmojung@skku.edu

Ya-xiang Yuan
yyx@lsec.cc.ac.cn

1 AORC, Sungkyunkwan University, Suwon 16419, Korea

2 Department of Mathematics, Sungkyunkwan University, Suwon 16419, Korea

3 State Key Laboratory of Scientific and Engineering Computing, ICMSEC, AMSS, CAS, Beijing 100190, China

4 Department of Mathematics Education, Sungkyunkwan University, Seoul 03063, Korea

1 Introduction

In this paper, we consider equality constrained optimization problems of the form

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1a}$$

$$\text{subject to } c_j(x) = 0, \quad j = 1, \ldots, m, \tag{1b}$$

where $f, c_j : \mathbb{R}^n \to \mathbb{R}$ $(j = 1, \ldots, m)$ are continuously differentiable.

One of the most popular algorithms for (1) is the Sequential Quadratic Programming (SQP). At the k-th iteration of the SQP method, a search direction d_k is computed by solving the subproblem:

$$\min_{d \in \mathbb{R}^n} \varphi_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{2a}$$

$$\text{subject to } A_k^T d + c_k = 0, \tag{2b}$$

where

$$g_k = \nabla f(x_k), \;\; c_k = (c_1(x_k), \ldots, c_m(x_k))^T, \;\; A_k = \left[\nabla c_1(x_k), \ldots, \nabla c_m(x_k)\right],$$

and B_k is an approximation to the Hessian of the Lagrangian of (1). Formally,

$$B_k \approx \nabla_x^2 \mathcal{L}(x_k, \lambda_k).$$

We refer to [2,6] for comprehensive understanding and further references on SQP methods. SQP is a very successful algorithm for small or medium-size problems (1). However, it may not be suitable for large scale problems, due to computational cost and memory issue, for example.

Currently, solving large scale problems is one of the most important issues in nonlinear optimization. For such problems, a large number of variables, limited memory storage, and massive computations at each iteration consist of major difficulties. A recent review by Gould et al. [7] discusses the main approaches for handling large scale problems, such as step computation, active set, gradient projection and interior point methods.

To solve large scale problems, such as large scale nonlinear optimization, nonlinear equations, and nonlinear least squares, Yuan [19,20] proposed a new approach using subspace techniques. Also, a model algorithm using subspace was given for unconstrained and equality constrained optimization and possible choices for subspaces were introduced. In the subspace methods, the main issue is how to construct subproblems in a low dimension so that the computational cost in each iteration can be reduced, compared to standard approaches.

Yuan [19,20] also showed that in many standard algorithms, such as conjugate gradient method, limited-memory quasi-Newton method, projected gradient method, and null space method, there are ideas or techniques which can be viewed as subspace approaches. Based on these observations, several algorithms have been proposed. In particular, for unconstrained optimization, Wang and Yuan [18] and Wang et al. [17] proposed subspace trust region methods using standard quasi-Newton and limited-memory quasi-Newton approach, respectively. More detailed discussions on subspace methods for unconstrained optimization can be found in [5,15–18,21]. For nonlinear equality constrained optimization, Grapiglia et al. [9] investigated subspace properties for the Celis-Dennis-Tapia (CDT) problem [3] and proposed a subspace method. In this method, as iteration increases, the dimension of subspace increases, and so memory and computational cost increase. This is the main drawback of the method proposed by Grapiglia et al. [9]. Recently, subspace choices for the CDT problem and subspace properties of the quadratically constrained quadratic program (QCQP) were investigated by Zhao and Fan [22,23], respectively.

By extending the method in [19], we propose a new subspace SQP method for large scale nonlinear equality constrained optimization problems. Since our method controls the dimension of subspace at each iteration, we avoid the rapid increase of dimension. Furthermore, we adopt a line search. As far as we know, line search has not been applied yet in the context of subspace methods for equality constrained optimization.

In the case of few constraints, our method is equivalent to the standard SQP method in the full space. In the case of many constraints, we reduce the number of constraints in the subproblem and we show that the solution of the subspace subproblem is a descent direction of a particular exact penalty function. Global convergence properties of the proposed method are given for both cases.

This paper is organized as follows. In Sect. 2, we introduce the subspace method for unconstrained optimization in [19]. In Sect. 3, we describe a new subspace technique. Based on it, we propose a subspace method for equality constrained optimization by extending the method in [19]. In Sect. 4, the convergence of the proposed method is established. Numerical results are reported in Sect. 5. Finally, concluding remarks are given in Sect. 6.

2 Subspace methods for unconstrained optimization

In this section, we review the subspace method proposed in [19]. Let us consider unconstrained optimization problems of the form

$$\min_{x\in\mathbb{R}^n} f(x). \tag{3}$$

The choice of a subspace can be motivated by limited-memory quasi-Newton methods [11]. In general, to approximate the Hessian $\nabla^2 f(x_k)$, quasi-Newton methods for (3) use the secant equation

$$B_k s_{k-1} = y_{k-1},$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. A typical example is the famous BFGS method [12]:

$$B_k = B_{k-1} - \frac{B_{k-1}s_{k-1}s_{k-1}^T B_{k-1}}{s_{k-1}^T B_{k-1}s_{k-1}} + \frac{y_{k-1}y_{k-1}^T}{s_{k-1}^T y_{k-1}}.$$

Thus, B_k is updated by using B_{k-1}, s_{k-1}, and y_{k-1}. However, in large scale problems, retaining B_{k-1} may introduce a memory shortage as k increases.

As a variant of quasi-Newton methods, limited-memory quasi-Newton methods [11] use only a few previous terms; given $\bar{m}$, they periodically approximate the Hessian using $B_k^{(i-1)}$, $s_{k-\bar{m}-1+i}$, and $y_{k-\bar{m}-1+i}$ for $i = 1, \ldots, \bar{m}$ from the initial guess $B_k^{(0)} = \delta_k I$ with $\delta_k > 0$. Byrd et al. [1] proposed a compact representation of limited-memory BFGS update matrix

$$B_k = B_k^{(\bar{m})} = \delta_k I + W_k D_k W_k^T,$$

where D_k is a $2\bar{m} \times 2\bar{m}$ matrix and

$$W_k = [S_k \quad Y_k] = \left[s_{k-1}, s_{k-2}, \ldots, s_{k-\bar{m}}, y_{k-1}, y_{k-2}, \ldots, y_{k-\bar{m}}\right] \in \mathbb{R}^{n\times 2\bar{m}}.$$

If a line search type method is applied, we have $s_k = \alpha_k d_k = -\alpha_k B_k^{-1} g_k$, while for trust region algorithms one has $s_k = -(B_k + \nu_k I)^{-1} g_k$ with $\nu_k \geq 0$. In either case, it can be shown that s_k belongs to the subspace

$$\operatorname{span}\{g_k, s_{k-1}, \ldots, s_{k-\bar{m}}, y_{k-1}, \ldots, y_{k-\bar{m}}\}. \tag{4}$$

Based on the subspace (4), Wang et al. [17] proposed a subspace trust region method for large scale unconstrained optimization problems (3). Preliminary numerical results show that their algorithm is encouraging. In their experiments, the scales of problem range from 500 to 10,000, but the dimensions of subspace are low, between 6 and 16. In the next section, we propose a new subspace method for the equality constrained optimization problem, using a line search framework.

3 A new subspace method for equality constrained optimization

In this section, we describe a new subspace technique for solving the equality constrained optimization problem (1). Suppose that at the k-th iteration, we have the current iterate point x_k and the subspace $\mathcal{T}_k$. We denote the dimension of $\mathcal{T}_k$ by i_k and let $Q_k = \left[q_1^{(k)}, q_2^{(k)}, \ldots, q_{i_k}^{(k)}\right]$ be a matrix of linearly independent vectors in $\mathcal{T}_k$. Since any vector $d \in \mathcal{T}_k$ can be written as $d = Q_k z$ for some $z \in \mathbb{R}^{i_k}$, the SQP subproblem with respect to the subspace $\mathcal{T}_k$ is given as

$$\min_{z\in\mathbb{R}^{i_k}} \bar{\varphi}_k(z) = (Q_k^T g_k)^T z + \frac{1}{2} z^T Q_k^T B_k Q_k z \tag{5a}$$

$$\text{subject to } \; A_k^T Q_k z + c_k = 0. \tag{5b}$$

However, when $m > i_k$, the linearized constraints (5b) may not be feasible because the system is over-determined. In this case, to guarantee the non-emptiness of feasible set, we consider the reduced constrained version of (5b):

$$\min_{z \in \mathbb{R}^{i_k}} \bar{\varphi}_k(z) = (Q_k^T g_k)^T z + \frac{1}{2} z^T Q_k^T B_k Q_k z \tag{6a}$$

$$\text{subject to } \; P_k^T (A_k^T Q_k z + c_k) = 0, \tag{6b}$$

where P_k is a projection matrix. The Karush–Kuhn–Tucker (KKT) system of (6) is as follows:

$$Q_k^T (B_k Q_k z + A_k P_k \mu + g_k) = 0, \tag{7}$$

$$P_k^T (A_k^T Q_k z + c_k) = 0, \tag{8}$$

where μ is a vector of Lagrangian multipliers.

Now we present a subspace SQP method for equality constrained optimization. It can be considered as a modification of the subspace method for equality constrained optimization proposed by Yuan [19].

Algorithm 1 *(Subspace SQP algorithm)*

Step 1 Given x_0, set an initial subspace $\mathcal{T}_0$, an initial penalty parameter $\sigma_0 > 0$, $tolerance \epsilon > 0, iteration number k$:=0.

Step 2 Solve the subspace subproblem (6):

$$\min_{z \in \mathbb{R}^{i_k}} \bar{\varphi}_k(z) = (Q_k^T g_k)^T z + \frac{1}{2} z^T Q_k^T B_k Q_k z$$

$$\text{subject to } \; P_k^T (A_k^T Q_k z + c_k) = 0,$$

to obtain z_k and Lagrangian multipliers μ_k, and set $d_k = Q_k z_k$ and $\lambda_k^+ = P_k \mu_k$. If $\|d_k\| \le \epsilon$ then stop.

Step 3 Perform a line search to obtain the stepsize α_k which satisfies

$$\phi(x_k + \alpha d_k, \sigma_k) < \phi(x_k, \sigma_k)$$

and set

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

$$\lambda_{k+1} = \lambda_k + \alpha_k (\lambda_k^+ - \lambda_k)$$

where $\phi(x, \sigma)$ is a penalty function and σ is a penalty parameter.

Step 4 Generate σ_{k+1}, subspace $\mathcal{T}_{k+1}$, and $\bar{\varphi}_{k+1}(z)$.

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

The key issue for Algorithm 1 is how to choose a subspace and solve the corresponding subspace subproblem quickly. We adapt the damped limited-memory BFGS update formula for equality constrained problems, which guarantees B_k to be positive definite [13]. With the relation $H_k = B_k^{-1}$, we apply the following formulae:

$$B_0 = I, \quad H_0 = I.$$

For $k \geq 1$,

$$B_k = \delta_k I - \begin{bmatrix} \delta_k S_k & \bar{Y}_k \end{bmatrix} \begin{bmatrix} \delta_k S_k^T S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} \delta_k S_k^T \\ \bar{Y}_k^T \end{bmatrix},$$
$$H_k = \gamma_k I + \begin{bmatrix} S_k & \gamma_k \bar{Y}_k \end{bmatrix} \begin{bmatrix} R_k^{-T}(D_k + \gamma_k \bar{Y}_k^T \bar{Y}_k) R_k^{-1} & -R_k^{-T} \\ -R_k^{-1} & 0 \end{bmatrix} \begin{bmatrix} S_k^T \\ \gamma_k \bar{Y}_k^T \end{bmatrix},$$

where

$$S_k = \begin{cases} \begin{bmatrix} s_0, \ldots, s_{k-1} \end{bmatrix} & \text{if} \quad k \leq \bar{m} \\ \begin{bmatrix} s_{k-\bar{m}}, \ldots, s_{k-1} \end{bmatrix} & \text{otherwise} \end{cases},$$
$$\bar{Y}_k = \begin{cases} \begin{bmatrix} \bar{y}_0, \ldots, \bar{y}_{k-1} \end{bmatrix} & \text{if} \quad k \leq \bar{m} \\ \begin{bmatrix} \bar{y}_{k-\bar{m}}, \ldots, \bar{y}_{k-1} \end{bmatrix} & \text{otherwise} \end{cases},$$
$$D_k = \begin{cases} \operatorname{diag}\left(s_0^T \bar{y}_0, \ldots, s_{k-1}^T \bar{y}_{k-1}\right) & \text{if} \quad k \leq \bar{m} \\ \operatorname{diag}\left(s_{k-\bar{m}}^T \bar{y}_{k-\bar{m}}, \ldots, s_{k-1}^T \bar{y}_{k-1}\right) & \text{otherwise} \end{cases},$$
$$\delta_k = \frac{s_{k-1}^T \bar{y}_{k-1}}{\|s_{k-1}\|_2^2}, \quad \gamma_k = \frac{1}{\delta_k},$$
$$(L_k)_{ij} = \begin{cases} \tilde{s}_{i-1}^T \tilde{y}_{j-1} & \text{if} \quad i > j \\ 0 & \text{otherwise} \end{cases}, \quad i, j = \begin{cases} 1, 2, \ldots, k & \text{if} \quad k \leq \bar{m} \\ 1, 2, \ldots, \bar{m} & \text{otherwise} \end{cases},$$
$$(R_k)_{ij} = \begin{cases} \tilde{s}_{i-1}^T \tilde{y}_{j-1} & \text{if} \quad i > j \\ 0 & \text{otherwise} \end{cases}, \quad i, j = \begin{cases} 1, 2, \ldots, k & \text{if} \quad k \leq \bar{m} \\ 1, 2, \ldots, \bar{m} & \text{otherwise} \end{cases},$$
$$\tilde{s}_i = \begin{cases} s_i & \text{if} \quad k \leq \bar{m} \\ s_{k-\bar{m}+i} & \text{otherwise} \end{cases}, \quad \tilde{y}_i = \begin{cases} \bar{y}_i & \text{if} \quad k \leq \bar{m} \\ \bar{y}_{k-\bar{m}+i} & \text{otherwise} \end{cases},$$
$$s_i = x_{i+1} - x_i, \quad y_i = \nabla_x \mathcal{L}(x_{i+1}, \lambda_{i+1}) - \nabla_x \mathcal{L}(x_i, \lambda_i),$$
$$\theta_i = \begin{cases} 1 & \text{if} \quad s_i^T y_i \geq 0.2 s_i^T B_i s_i \\ (0.8 s_i^T B_i s_i)/(s_i^T B_i s_i - s_i^T y_i) & \text{otherwise} \end{cases},$$
$$\bar{y}_i = \theta_i y_i + (1 - \theta_i) B_i s_i,$$

and λ_i are Lagrangian multipliers. The modified differences of the gradients of Lagrangian $\bar{y}_i$ are used to guarantee B_k to be positive definite. See [13] for details.

There are a variety of methods for solving the subproblem (2) and the corresponding subspace subproblem (6). In our numerical test, we use the range space method for these subproblems since B_k^{-1} is explicitly given as H_k through the quasi-Newton update formula [12]. More specifically, the range space method solves the following two systems

$$(A_k^T B_k^{-1} A_k)\lambda = -A_k^T B_k^{-1} g_k + c_k, \tag{9}$$

$$d = -B_k^{-1}(g_k + A_k\lambda), \tag{10}$$

for the solution of the subproblem (2).

In Step 3, we use backtracking line search, which starts with $\alpha = 1$. The stepsize $\alpha_k = \alpha$ is accepted if

$$\phi(x_k + \alpha_k d_k, \sigma_k) \leq \phi(x_k, \sigma_k) + \eta\alpha_k D_{d_k}\phi(x_k, \sigma_k),$$

where $\eta \in (0, \frac{1}{2})$, and $D_d\phi(x, \sigma)$ is the directional derivative of ϕ at x in the direction d. Otherwise, we set $\alpha \leftarrow \tau\alpha$, $\tau \in (0, 1)$ and repeat the line search. More details for our numerical experiments are given in Sect. 5.

4 Convergence analysis

In this section, we analyze convergence properties of the proposed method under the following assumptions:

Assumption 1 The starting point and all succeeding iterates lie in some closed, bounded and convex region $\mathcal{C}$ in $\mathbb{R}^n$.

Assumption 2 The columns of $A(x) = \left[\nabla c_1(x), \ldots, \nabla c_m(x)\right]$ are linearly independent for all $x \in \mathcal{C}$.

Assumption 3 For all $d \in \mathbb{R}^n$, there are positive constants β_1 and β_2 such that

$$\beta_1 \|d\|_2^2 \leq d^T B_k d \leq \beta_2 \|d\|_2^2$$

for all k.

These conditions are commonly chosen to prove global convergence of algorithms for constrained optimization problems [2].

We consider two cases based on the number of constraints m, comparing with the number of the variables n. For simplicity, we assume $k \geq \bar{m}$.

4.1 Case I: Few constraints

First, we consider the case of relatively few constraints in the problem (1). Since $m \ll n$, the subproblem (5) is used at each iteration. The following theorem shows that subproblem (2) is equivalent to subproblem (5). Its proof is an adaptation of Lemma 2.2 in [18].

Theorem 1 *Suppose B_k is the k-th updated matrix by the damped limited-memory BFGS update formula. Let d_k be a solution of problem (2), and $s_k = \alpha_k d_k$ with stepsize $\alpha_k > 0$. Let*

$$\mathcal{T}_k^F = span\{g_k, s_{k-\bar{m}}, \ldots, s_{k-1}, \bar{y}_{k-\bar{m}}, \ldots, \bar{y}_{k-1}, \nabla c_1(x_k), \ldots, \nabla c_m(x_k)\}.$$

If Q_k is an orthonormal basis matrix of $\mathcal{T}_k^F$, then subproblem (2) is equivalent to (5).

Proof We denote the dimension of the subspace $\mathcal{T}_k^F$ by r. Let $U_k \in \mathbb{R}^{n\times(n-r)}$ be a matrix such that $[Q_k,\ U_k]$ is an $n \times n$ orthogonal matrix. Then, for each $d \in \mathbb{R}^n$, there exists one and only one pair $z \in \mathbb{R}^r$, $u \in \mathbb{R}^{n-r}$ such that $d = Q_k z + U_k u$. Thus, it follows that

$$\begin{aligned}
\varphi_k(d) &= g_k^T d + \frac{1}{2} d^T B_k d \\
&= g_k^T (Q_k z + U_k u) + \frac{1}{2}(Q_k z + U_k u)^T B_k (Q_k z + U_k u) \\
&= (Q_k^T g_k)^T z + (U_k^T g_k)^T u + \frac{1}{2} z^T Q_k^T B_k Q_k z + u^T U_k^T B_k Q_k z \\
&\quad + \frac{1}{2} u^T U_k^T B_k U_k u.
\end{aligned} \tag{11}$$

By the choice of $\mathcal{T}_k^F$, we have $S_k^T U_k = 0$, $\bar{Y}_k^T U_k = 0$, and $g_k^T U_k = 0$. Hence, it follows that

$$B_k U_k = \delta_k U_k - \begin{bmatrix} \delta_k S_k & \bar{Y}_k \end{bmatrix} \begin{bmatrix} \delta_k S_k^T S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} \delta_k S_k^T U_k \\ \bar{Y}_k^T U_k \end{bmatrix} = \delta_k U_k,$$

and $U_k^T B_k Q_k = \delta_k U_k^T Q_k = 0$. By (11), we have

$$\varphi_k(d) = (Q_k^T g_k)^T z + \frac{1}{2} z^T Q_k^T B_k Q_k z + \frac{1}{2}\delta_k \|u\|_2^2.$$

Since $A_k^T U_k = 0$, we have

$$0 = A_k^T d + c_k = A_k^T (Q_k z + U_k u) + c_k = A_k^T Q_k z + c_k.$$

Now the above relation indicates (2) is equivalent to

$$\begin{aligned}
\min_{z,\,u} \quad & (Q_k^T g_k)^T z + \frac{1}{2} z^T Q_k^T B_k Q_k z + \frac{1}{2}\delta_k \|u\|_2^2 \\
\text{subject to} \quad & A_k^T Q_k z + c_k = 0.
\end{aligned}$$

Since $\delta_k > 0$, it is easy to see that the above problem is equivalent to (5) with $u = 0$. Therefore (2) is equivalent to (5) with $d_k = Q_k z_k$. □

By Theorem 1, we ensure that the proposed method generates the same iterate as the standard SQP method. Thus, we can obtain the global convergence of the proposed method using Theorem 4.3 of Boggs and Tolle [2]. For completeness, we restate this theorem for the L_1 exact penalty function [10]

$$\phi_1(x, \sigma) = f(x) + \sigma \|c(x)\|_1 .$$

Theorem 2 (Boggs and Tolle [2]) *Assume that σ is chosen by*

$$\sigma = \left\|\lambda_k^+\right\|_\infty + \bar{\rho}$$

for some constant $\bar{\rho} > 0$. Then the proposed algorithm started at any point x_0 with stepsize $\alpha \geq \bar{\alpha} > 0$ chosen to satisfy

$$\phi_1(x_k + \alpha d_k, \sigma_k) \leq \phi_1(x_k, \sigma_k) + \eta\alpha D_{d_k}\phi_1(x_k, \sigma_k)$$

converges to a KKT point of (1).

4.2 Case II: Many constraints

Now, we consider the case of relatively many constraints, i.e. $m \approx n$ in (1). In this case, problem (6) is solved at each iteration. Since we reduce the number of constraints in the subproblem, we cannot expect that the proposed method generates the same iterate as the standard SQP method. Nevertheless, we can show that the search direction is a descent direction for the L_∞ exact penalty function [10]

$$\phi_\infty(x, \sigma) = f(x) + \sigma \|c(x)\|_\infty . \tag{12}$$

Using this property, the global convergence of the proposed method is obtained.

First of all, we state the following result of Yuan [20]. It is used to show that the solution of the subproblem (6) is a descent direction of $\phi_\infty(x, \sigma)$.

Lemma 1 (Yuan [20]) *Suppose we sort the violations in decreasing order:*

$$\left|c_{l_1}(x_k)\right| \geq \left|c_{l_2}(x_k)\right| \geq \ldots \geq \left|c_{l_m}(x_k)\right|, \tag{13}$$

and $|c_{l_{j_k+1}}| < \|c_k\|_\infty$. If the search direction d_k satisfies

$$P_k^T(c_k + A_k^T d_k) = 0,$$

where

$$P_k = \left[e_{l_1}, e_{l_2}, \ldots, e_{l_{j_k}}\right] \in \mathbb{R}^{m \times j_k}, \tag{14}$$

then d_k is a descent direction of $\|c(x)\|_\infty$ at x_k.

Using Lemma 1, we derive the following theorem for a descent property of the search direction.

Theorem 3 *Suppose* $\{x_k\}$, $\{d_k\}$, $\{\sigma_k\}$, $\{c_k\}$, $\{Q_k\}$, $\{P_k\}$, *and* $\{B_k\}$ *are sequences generated by the proposed subspace SQP algorithm under Assumptions 2 and 3. If* $|c_{l_{j_k+1}}(x_k)| < \|c_k\|_\infty$, P_k *is given by* (14), *and sufficiently large* σ_k *is chosen, then the search direction* d_k *is a descent direction for* $\phi_\infty(x, \sigma)$. *That is, its directional derivative at* x_k *in the direction* d_k *satisfies*

$$D_{d_k}\phi_\infty(x_k, \sigma_k) < 0.$$

Proof Using Lemma 1, we obtain that

$$\begin{aligned} D_{d_k}\phi_\infty(x_k, \sigma_k) &= g_k^T d_k - \sigma_k \|c_k\|_\infty \\ &= g_k^T Q_k z_k - \sigma_k \|c_k\|_\infty \\ &= -z_k^T Q_k^T B_k Q_k z_k - \mu_k^T P_k^T A_k^T Q_k z_k - \sigma_k \|c_k\|_\infty \\ &= -z_k^T Q_k^T B_k Q_k z_k + \mu_k^T P_k^T c_k - \sigma_k \|c_k\|_\infty \\ &\le -z_k^T Q_k^T B_k Q_k z_k - (\sigma_k - \|\mu_k\|_1)\|c_k\|_\infty, \end{aligned}$$

where the third equation is obtained by using (7), the fourth equation is from (8), and we get the last inequality by using Hölder's inequality. Assumption 3 implies that $-z_k^T Q_k^T B_k Q_k z_k$ is always negative. Thus, if σ_k is sufficiently large, that is, $\sigma_k > \|\mu_k\|_1$, then $D_{d_k}\phi_\infty(x_k, \sigma_k) < 0$. Hence the search direction d_k is a descent direction for $\phi_\infty(x, \sigma)$. □

Now we establish the global convergence of our algorithm by applying the idea of Powell [14].

Theorem 4 *Suppose* $\{x_k\}$, $\{d_k\}$, $\{\sigma_k\}$, $\{c_k\}$, $\{Q_k\}$, $\{P_k\}$, *and* $\{B_k\}$ *are sequences generated by the proposed subspace SQP algorithm under Assumptions 1–3. If* $|c_{l_{j_k+1}}(x_k)| < \|c_k\|_\infty$, P_k *is given by* (14), σ_k *is chosen as*

$$\sigma_k = \|\mu_k\|_1 + \bar{\rho} \tag{15}$$

for some constant $\bar{\rho} > 0$, Q_k *is an orthonormal basis matrix of* $\mathcal{T}_k^M$ *where*

$$\mathcal{T}_k^M = span\left\{g_k, s_{k-\bar{m}}, \ldots, s_{k-1}, \bar{y}_{k-\bar{m}}, \ldots, \bar{y}_{k-1}, \nabla c_{l_1}(x_k), \ldots, \nabla c_{l_{j_k}}(x_k)\right\},$$

and the stepsize $\alpha_k > 0$ *satisfies*

$$\phi_\infty(x_k + \alpha_k d_k, \sigma_k) \le \phi_\infty(x_k, \sigma_k) + \eta\alpha_k D_{d_k}\phi_\infty(x_k, \sigma_k) \tag{16}$$

for $\eta \in (0, \frac{1}{2})$, *then any limit point of* $\{x_k\}$ *is a KKT point of (1).*

Proof First, we show that if $d_k = 0$, then x_k is a KKT point of original problem (1). By using $d_k = Q_k z_k,\ \lambda_k^+ = P_k \mu_k$, we can rewrite the Eqs. (7) and (8) as follows:

$$Q_k^T (B_k d_k + A_k \lambda_k^+ + g_k) = 0, \tag{17}$$
$$P_k^T (A_k^T d_k + c_k) = 0. \tag{18}$$

With a matrix U_k which makes $[Q_k,\ U_k]$ an $n \times n$ orthogonal matrix, we have the following relations:

$$\begin{aligned} &B_k U_k = \delta_k U_k,\ U_k^T A_k P_k = 0,\ U_k^T B_k Q_k = \delta_k U_k^T Q_k = 0,\ U_k^T g_k = 0, \\ &U_k^T (B_k d_k + A_k \lambda_k^+ + g_k) = \delta_k U_k^T Q_k z_k + U_k^T A_k P_k \mu_k + U_k^T g_k = 0. \end{aligned} \tag{19}$$

Using (17) and (19), we have

$$\begin{aligned} B_k d_k + A_k \lambda_k^+ + g_k = 0, \\ P_k^T (A_k^T d_k + c_k) = 0. \end{aligned} \tag{20}$$

If $d_k = 0$, then

$$\begin{aligned} A_k \lambda_k^+ + g_k = 0, \\ P_k^T c_k = 0. \end{aligned}$$

From (13), $P_k^T c_k = 0$ implies $c_k = 0$. Thus, x_k is a KKT point of the original problem (1) with a new Lagrangian multiplier λ_k^+.

Now suppose $d_k \neq 0$ for all k. We define the following two functions $\bar{L}_k$ and $\bar{\phi}_k$ of d:

$$\begin{aligned} \bar{L}_k(x_k + d) &= f_k + g_k^T (Q_k z) + \frac{1}{2}(Q_k z)^T B_k (Q_k z) + \mu_k^T P_k^T (A_k^T Q_k z + c_k) \\ &= f_k + g_k^T d + \frac{1}{2} d^T B_k d + \lambda_k^{+T} (A_k^T d + c_k), \\ \bar{\phi}_k(x_k + d) &= f_k + g_k^T (Q_k z) + \frac{1}{2}(Q_k z)^T B_k (Q_k z) + \sigma_k \left\| P_k^T (A_k^T Q_k z + c_k) \right\|_\infty \\ &= f_k + g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k \left\| P_k^T (A_k^T d + c_k) \right\|_\infty, \end{aligned} \tag{21}$$

with $d = Q_k z$, $\lambda_k^+ = P_k \mu_k$. These functions are convex and achieve their minimum values at $d = d_k$. Because of (20), we have

$$\bar{L}_k(x_k) = \bar{L}_k(x_k + d_k) + \frac{1}{2} d_k^T B_k d_k \geq \bar{L}_k(x_k + d_k). \tag{22}$$

The difference between these functions is bounded below by the estimate below.

$$\begin{aligned}\bar{\phi}_k(x_k+d)-\bar{L}_k(x_k+d) &= \sigma_k \left\| P_k^T(A_k^T d + c_k)\right\|_\infty - \mu_k^T P_k^T(A_k^T d + c_k)\\ &\geq (\sigma_k - \|\mu_k\|_1)\left\| P_k^T(A_k^T d + c_k)\right\|_\infty \\ &\geq \bar{\rho}\left\| P_k^T(A_k^T d + c_k)\right\|_\infty . \end{aligned} \tag{23}$$

If any of the equalities (6b) are violated, then we have

$$\bar{\phi}_k(x_k+d) > \bar{L}_k(x_k+d).$$

In this case, if x_k is infeasible with respect to the given nonlinear constraints (1b), we have the bound

$$\bar{\phi}_k(x_k) > \bar{L}_k(x_k) \geq \bar{L}_k(x_k+d_k) = \bar{\phi}_k(x_k+d_k). \tag{24}$$

While, if x_k is feasible, we have the condition

$$\bar{\phi}_k(x_k) = \bar{L}_k(x_k) > \bar{L}_k(x_k+d_k) = \bar{\phi}_k(x_k+d_k), \tag{25}$$

where the strict inequality comes from that the term $d_k^T B_k d_k$ in (22) is positive. Hence the number

$$t_k = \bar{\phi}_k(x_k) - \bar{\phi}_k(x_k+d_k)$$

is positive. In addition, we have

$$\begin{aligned}\|c(x_k+\alpha d_k)\|_\infty &= \|c_k\|_\infty + \alpha D_{d_k}\|c(x_k)\|_\infty + o(\alpha)\\ &= \|c_k\|_\infty - \alpha\|c_k\|_\infty + o(\alpha)\end{aligned}$$

and

$$\bar{\phi}_k(x_k+\alpha d_k) = f_k + \alpha g_k^T d_k + \frac{1}{2}\alpha^2 d_k^T B_k d_k + \sigma_k(1-\alpha)\left\|c_k\right\|_\infty .$$

Owing to the above two relations and the convexity of the function (21), we have

$$\begin{aligned}\phi_\infty(x_k,\sigma) - \phi_\infty(x_k+\alpha d_k,\sigma) &= \bar{\phi}_k(x_k) - \bar{\phi}_k(x_k+\alpha d_k) + o(\alpha)\\ &\geq \alpha[\bar{\phi}_k(x_k) - \bar{\phi}_k(x_k+d_k)] + o(\alpha)\\ &= \alpha t_k + o(\alpha), \quad 0\leq\alpha\leq 1.\end{aligned} \tag{26}$$

Therefore the reduction of the penalty function

$$\phi_\infty(x_k+\alpha_k d_k,\sigma_k) < \phi_\infty(x_k,\sigma_k)$$

can be achieved by a sufficiently small and positive α_k. From the choice of α_k satisfying (16), we can replace the above condition on the stepsize by the inequality

$$\phi_\infty(x_{k+1}, \sigma) \le \phi_\infty(x_k, \sigma) - \eta\alpha_k t_k, \tag{27}$$

since $D_{d_k}\phi_\infty(x_k, \sigma_k) \le \bar{\phi}_k(x_k + d_k) - \bar{\phi}_k(x_k) = -t_k < 0$.

Let ξ be a small positive constant, and consider iterations with $t_k > \xi$. Because the continuity of the first order derivatives in a compact domain introduces uniform continuity, and the vector d_k in (26) is bounded, there is a positive constant $\beta(\xi)$ such that the inequality (27) holds for any $\alpha_k \in [0, \beta(\xi)]$. Thus, by the line search process, we can deduce that

$$[\phi_\infty(x_k, \sigma) - \phi_\infty(x_{k+1}, \sigma)] \ge \eta\xi\bar{\alpha},$$

where $\bar{\alpha} \in (0, \beta(\xi)]$. However, the reduction in the line search tends to zero, because $\{\phi_\infty(x_k, \sigma) : k = 1, 2, 3, \ldots\}$ is a monotonically decreasing sequence which is bounded below. Therefore t_k tends to zero, that is,

$$\lim_{k\to\infty} [\bar{\phi}_k(x_k) - \bar{\phi}_k(x_k + d_k)] = 0. \tag{28}$$

Because either (24) or (25) is satisfied for each k, the limit (28) gives

$$\lim_{k\to\infty} [\bar{\phi}_k(x_k) - \bar{L}_k(x_k)] = 0 \tag{29}$$

and

$$\lim_{k\to\infty} [\bar{L}_k(x_k) - \bar{L}_k(x_k + d_k)] = 0. \tag{30}$$

By letting $d = 0$ in the estimate (23), and using the condition (15), we deduce from (29) that all limit points of the sequence $\{x_k : k = 1, 2, 3, \ldots\}$ satisfy the nonlinear constraints (1b).

Without loss of generality, we assume that $\{x_k : k = 1, 2, 3, \ldots\}$ has only one limit point, say x_*. By the above arguments, $c_j(x_*) = 0$ for all $j = 1, \ldots, m$. From (20), we have that

$$\lim_{k\to\infty} \left\| B_k d_k + A_k\lambda_k^+ + g_k \right\| = 0. \tag{31}$$

Since (22) and (30) imply $B_k^{1/2} d_k \to 0$, and the matrices $\{B_k : k = 1, 2, 3, ...\}$ are uniformly bounded, the limit (31) still holds after eliminating the term $B_k d_k$. Because the Lagrange multipliers are bounded, by passing to a subsequence if necessary, we deduce that $A_*\lambda_*^+ + g_* = 0$, where λ_*^+ is any cluster points of $\{\lambda_k^+\}$, $\lim_{k\to\infty} A_k = A_*$, and $\lim_{k\to\infty} g_k = g_*$. Hence x_* is a KKT point of (1). □

5 Numerical results

In this section, we illustrate the implementation of the proposed algorithm SubSQP and report numerical results on some of CUTEst problems [8]. To show the effectiveness of SubSQP in large scale nonlinear equality constrained problems, we test it on 11 CUTEst problems. In most of these problems, the number of variables is greater than or equal to 5000, and the number of constraints is greater than half of the number of variables, consisting of only nonlinear equality constraints. The experiment also includes 5 problems with few constraints. To show the soundness and efficiency of SubSQP, we compare it with the standard SQP method. All tests are performed on an Intel Core i5 3.50 GHz CPU with 8 GB memory, running on Ubuntu 16.04 LTS and MATLAB R2016b.

For numerical tests, parameters are chosen as $\eta = 10^{-3}$, $\tau = 0.5$, $\sigma_k = \|\mu_k\|_1 + \bar{\rho}$, $\bar{\rho} = 10^{-3}$, $\epsilon = 10^{-3}$ and the maximum number of iterations is 100. Since we are mostly interested in large scale problems, we use the L_∞ exact penalty function

$$\phi_\infty(x, \sigma) = f(x) + \sigma \|c(x)\|_\infty ,$$

to measure progress towards a solution.

In order to prevent a rapid increase of the dimension of subspace in the case of many constraints, we limit the number of constraint gradients $\nabla c_{l_j}(x_k)$ in the subspace to $\bar{m}$, for all k unless specified otherwise. To effectively decrease constraint violations, we select the $\bar{m}$ largest gradients in magnitude after sorting:

$$\left|c_{l_1}(x_k)\right| \geq \left|c_{l_2}(x_k)\right| \geq ... \geq \left|c_{l_m}(x_k)\right| .$$

By including those constraint gradients in the subspace $\mathcal{T}_k^M$, the step $s_k = \alpha_k d_k$ generated by the proposed method moves in the direction of reducing severe constraint violations. Consequently, we choose P_k in (14) as follows:

$$P_k = \left[e_{l_1}, e_{l_2}, \ldots, e_{l_{\bar{m}}}\right] \in \mathbb{R}^{m \times \bar{m}},$$

and set the subspace $\mathcal{T}_k^M$:

$$\mathcal{T}_k^M = \text{span}\left\{g_k, s_{k-\bar{m}}, \ldots, s_{k-1}, \bar{y}_{k-\bar{m}}, \ldots, \bar{y}_{k-1}, \nabla c_{l_1}(x_k), \ldots, \nabla c_{l_{\bar{m}}}(x_k)\right\} .$$

In the case of few constraints, all m constraint gradients are used to construct the subspace $\mathcal{T}_k^F$.

To compute the orthonormal basis matrix Q_k, the reorthogonalization procedure in [4] is applied. If A_k is nearly rank deficient, SubSQP may eliminate some constraint gradients in the orthogonalization process. We use the conjugate gradient method to solve (9), and the relative residual norm is chosen for its stopping condition with tolerance 10^{-4}.

To improve the performance of the proposed method, we skip the update $(s_k, \bar{y}_k)$ if

$$10^{-8}\bar{y}_k^T\bar{y}_k \geq \bar{y}_k^T s_k,$$

and reset the subspace $\mathcal{T}_k$ if

$$|f_k - f_{k+1}| \leq 10^{-4}|f_k|$$

and

$$|\|c_k\|_\infty - \|c_{k+1}\|_\infty| \leq 10^{-4}\|c_k\|_\infty.$$

The stopping criterion consists of

$$||g_k + A_k\lambda_k||_\infty \leq \epsilon,\ ||c_k||_\infty \leq \epsilon \quad \text{or} \quad ||d_k||_2 \leq \epsilon.$$

We also terminate each algorithm when it generates a tiny stepsize $\alpha_k < 10^{-6}$.

The numerical results on a few CUTEst test problems are reported in Tables 1–2. In Table 1, n represents the number of variables, m represents the number of constraints, $\bar{m}$ is a parameter for constructing the subspace. Since SQP and SubSQP give

Table 1 Numerical results on CUTEst problems

Problem	n	m	$\bar{m}$	SQP		SubSQP	
				itr	time (s)	itr	time (s)
PORTFL1[a]	12	1	1	4	0.02	4	0.06
DECONVC[a]	63	1	2	19	0.03	19	0.10
HIMMELBI[a]	100	12	5	4	0.02	4	0.04
PRIMAL4[a]	1489	75	10	3	0.03	9	0.26
LUKVLE9	10,000	6	15	28	0.25	8	0.41
MSS2	756	703	5	8	0.25	13	0.50
MSS3	2070	1981	5	6	2.05	14	0.98
ORTHRGDS	5003	2500	5	5	6.43	16	2.29
DTOC2	5998	3996	10	9	25.74	43	10.07
DTOC1ND	5998	3996	10	5	14.60	81	18.48
LUKVLE14	9998	6664	10	4	42.19	32	18.59
LUKVLE11	9998	6664	10	5	59.98	100	62.40
LUKVLE16	9997	7497	15	5	72.11	45	31.05
LUKVLE17	9997	7497	15	14	156.65	100	67.01
LUKVLE2	10,000	4999	15	1	6.58	9	4.87
LUKVLE1	10,000	9998	10	6	116.58	14	13.28

[a]For numerical test, inequality constraints are considered as equalities, and the bounds on the variables are ignored

Table 2 Function values and constraints violation on CUTEst problems

Problem	SQP		SubSQP	
	f_*	$\|\|c_*\|\|$	f_*	$\|\|c_*\|\|$
PORTFL1	2.1548×10^{-2}	5.5511×10^{-17}	2.1548×10^{-2}	0.0000×10^{0}
DECONVC	1.3600×10^{-1}	4.2301×10^{-9}	1.3600×10^{-1}	4.2301×10^{-9}
HIMMELBI	-1.6815×10^{3}	3.4617×10^{-2}	-1.6815×10^{3}	3.4253×10^{-2}
PRIMAL4	-7.4255×10^{-1}	5.1518×10^{-4}	-7.4255×10^{-1}	1.6748×10^{-4}
LUKVLE9	2.1311×10^{3}	2.1381×10^{-2}	5.0533×10^{3}	9.7771×10^{-1}
MSS2	-3.7622×10^{2}	4.6892×10^{-3}	-3.7909×10^{2}	2.9689×10^{-3}
MSS3	-2.6856×10^{3}	1.5948×10^{0}	-1.0379×10^{3}	2.7987×10^{-3}
ORTHRGDS	8.0722×10^{2}	3.1541×10^{0}	4.7501×10^{-1}	1.1691×10^{2}
DTOC2	5.0866×10^{-5}	3.7468×10^{-4}	9.9092×10^{-8}	1.5166×10^{-4}
DTOC1ND	4.7728×10^{1}	3.0867×10^{-3}	4.0664×10^{-2}	6.7570×10^{-1}
LUKVLE14	7.2702×10^{6}	1.2247×10^{1}	4.2148×10^{4}	1.5268×10^{1}
LUKVLE11	8.1023×10^{4}	2.1530×10^{0}	3.8694×10^{-2}	2.3749×10^{-1}
LUKVLE16	2.0930×10^{4}	4.5928×10^{0}	2.4740×10^{1}	1.1493×10^{0}
LUKVLE17	1.6189×10^{4}	1.0026×10^{0}	8.7603×10^{2}	5.1676×10^{0}
LUKVLE2	1.4279×10^{18}	3.6759×10^{4}	-1.4065×10^{6}	1.9098×10^{1}
LUKVLE1	1.1701×10^{4}	7.1104×10^{0}	1.0375×10^{4}	8.0343×10^{0}

different solutions due to the adoption of subspace in SubSQP, we report the number of iterations, CPU time, the final objective value f_* and the infinity norm of the final constraint violation $||c_*||$ in Tables 1 and 2.

In the case of many constraints, the computational cost of SubSQP per iteration is less than that of SQP, especially on large scale problems since it performs in the proposed subspace. However, SubSQP may require more iterations, depending on the choice of subspace. SubSQP overall outperforms SQP in terms of final objective value, given in Table 2. Especially, for the problems DTOC2, LUKVLE16, and LUKVLE2, SubSQP is faster than SQP with a less constraint violation. In the case of few constraints, the results of SubSQP are similar to those of SQP. Since SubSQP requires an orthogonalization process, SubSQP is slower than SQP.

6 Conclusion

In this paper, we present a subspace SQP method to solve large scale nonlinear equality constrained optimization problems. The subspace technique is a promising approach for solving large scale optimization problems [21], since it can reduce both computational cost and memory requirements. To efficiently apply the technique to equality constrained problems, we propose a subspace generated by the gradient of the objective function, previous steps, the modified differences of the gradients of Lagrangian, and gradients of constraints with substantial violation, based on damped limited-memory BFGS update.

With this subspace, we prove that our method has a global convergence property. Numerical results on some of CUTEst problems show that our proposed algorithm works better for large scale problems when compared with the standard SQP method. Future work concerns a different choice of subspace and accompanied analyses of such space, including implementation.

Acknowledgements The authors would like to thank Professor Jinyan Fan and anonymous referees for their kind advice and modifications. This work was supported by the National Research Foundation of Korea (NRF) NRF-2016R1A5A1008055. The second author was supported by the National Research Foundation of Korea (NRF) NRF-2016R1D1A1B03931337. The fourth author was supported by the National Research Foundation of Korea (NRF) NRF-2016R1D1A1B03934371.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Byrd, R.H., Nocedal, J., Schnabel, R.B.: Representations of quasi-Newton matrices and their use in limited memory methods. Math. Program. **63**, 129–156 (1994)
2. Boggs, P.T., Tolle, J.W.: Sequential quadratic programming. Acta Numer. **4**, 1–51 (1995)
3. 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 1984, pp. 71–82. SIAM, Philadelphia (1985)
4. 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)
5. Gill, P.E., Leonard, M.W.: Reduced-Hessian quasi-Newton methods for unconstrained optimization. SIAM J. Optim. **6**, 418–445 (1996)
6. Gould, N.I.M., Toint, P.L.: SQP methods for large-scale nonlinear programming. In: Powell, M.J.D., Scholtes, S. (eds.) System Modelling and Optimization: Methods, Theory and Applications. Kluwer, Dordrecht (2000)
7. Gould, N.I.M., Orban, D., Toint, PhL: Numerical methods for large-scale nonlinear optimization. Acta Numer. **14**, 299–361 (2005)
8. Gould, N.I.M., Orban, D., Toint, PhL: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comput. Optim. Appl. **60**, 545–557 (2015)
9. Grapiglia, G.N., Yuan, J., Yuan, Y.: A subspace version of the Powell–Yuan trust region algorithm for equality constrained optimization. J. Oper. Res. Soc. China **1**(4), 425–451 (2013)
10. Han, S.-P., Mangasarian, O.L.: Exact penalty functions in nonlinear programming. Math. Program. **17**, 251–269 (1979)
11. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**(151), 773–782 (1980)
12. Nocedal, J., Wright, S.J.: Numerical Optimization (Springer Series in Operations Research), 2nd edn. Springer, New York (2006)
13. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G.A. (ed.) Numerical Analysis Dundee 1977 (Lecture Notes in Mathematics 630), pp. 144–157. Springer, Berlin (1978)
14. Powell, M.J.D.: Variable metric methods for constrained optimization. In: Bachem, A., Grötschel, M., Korte, B. (eds.) Mathematical Programming: The State of the Art, Bonn, 1982, pp. 288–311. Springer, Berlin (1983)

15. Stoer, J., Yuan, Y.: A subspace study on conjugate gradient algorithms. ZAMM Z. Angew. Math. Mech. **75**, 69–77 (1995)
16. Vlček, J., Lukšan, L.: New variable metric methods for unconstrained minimization covering the large-scale case. Technical report No. V876, Institute of Computer Science, Academy of Sciences of the Czech Republic (2002)
17. Wang, Z.-H., Wen, Z.-W., Yuan, Y.: A subspace trust region method for large scale unconstrained optimization. In: Yuan, Y. (ed.) Numerical Linear Algebra and Optimization, pp. 265–274. Science Press, Beijing (2004)
18. Wang, Z.-H., Yuan, Y.: A subspace implementation of quasi-Newton trust region methods for unconstrained optimization. Numer. Math. **104**, 241–269 (2006)
19. Yuan, Y.: Subspace techniques for nonlinear optimization. In: Jeltsch, R., Li, T., Sloan, I.H. (eds.) Some Topics in Industrial and Applied Mathematics (Series in Contemporary Applied Mathematics CAM 8), pp. 206–218. Higher Education Press, Beijing (2007)
20. Yuan, Y.: Subspace methods for large scale nonlinear equations and nonlinear least squares. Optim. Eng. **10**, 207–218 (2009)
21. Yuan, Y.: A review on subspace methods for nonlinear optimization. In: Proceedings of International Congress of Mathematicians 2014 Seoul, Korea, pp. 807–827 (2014)
22. Zhao, X., Fan, J.: Subspace choices for the Celis–Dennis–Tapia problem. Sci. China Math. **60**, 1717–1732 (2017)
23. Zhao, X., Fan, J.: On subspace properties of the quadratically constrained quadratic program. J. Ind. Manag. Optim. **13**, 1625–1640 (2017)