SIAM J. OPTIM.
Vol. 26, No. 3, pp. 1429–1464

A FILTER ACTIVE-SET ALGORITHM FOR BALL/SPHERE CONSTRAINED OPTIMIZATION PROBLEM*

CHUNGEN SHEN†, LEI-HONG ZHANG‡, AND WEI HONG YANG§

Abstract. In this paper, we propose a filter active-set algorithm for the minimization problem over a product of multiple ball/sphere constraints. By making effective use of the special structure of the ball/sphere constraints, a new limited memory BFGS (L-BFGS) scheme is presented. The new L-BFGS implementation takes advantage of the sparse structure of the Jacobian of the constraints and generates curvature information of the minimization problem. At each iteration, only two or three reduced linear systems are required to solve for the search direction. The filter technique combined with the backtracking line search strategy ensures the global convergence, and the local superlinear convergence can also be established under mild conditions. The algorithm is applied to two specific applications, the nearest correlation matrix with factor structure and the maximal correlation problem. Our numerical experiments indicate that the proposed algorithm is competitive with some recently custom-designed methods for each individual application.

Key words. SQP, active set, filter, L-BFGS, ball/sphere constraints, nearest correlation matrix with factor structure, maximal correlation problem

AMS subject classifications. 65K05, 90C30

DOI. 10.1137/140989078

1. Introduction.

In this paper, we consider a class of optimization problems of minimizing a (at least) twice continuously differentiable function (probably nonconvex) $f(x) : \mathbb{R}^n \to \mathbb{R}$ over a product of multiple balls/spheres constraints. Upon rescaling the balls/spheres, we cast without loss of generality such class of minimization problems in the following form:

$$\text{(BCOP)} \begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & c_i(x) := \|x_{[i]}\|^2 - 1 = 0, \ i \in \mathcal{E}, \\ & c_i(x) := \|x_{[i]}\|^2 - 1 \le 0, \ i \in \mathcal{I}, \end{cases}$$

where $\mathcal{E} = \{1, 2, \ldots, m_1\}$, $\mathcal{I} = \{m_1 + 1, m_1 + 2, \ldots, m\}$, $x_{[i]} \in \mathbb{R}^{p_i}$, $x = (x_{[1]}^T, x_{[2]}^T, \ldots, x_{[m]}^T)^T$, $n = \sum_{i=1}^m p_i$, and $\|\cdot\|$ stands for the ℓ_2 vector norm. Here, we introduce the notation $x_{[i]} \in \mathbb{R}^{p_i}$ to represent the ith subvector of $x \in \mathbb{R}^n$ and formulate the product of multiple ball/sphere constraints as a set of equality and inequality constraints. To simplify subsequent presentation, we call the above programming the ball/sphere constrained optimization problem (BCOP). We emphasize that this problem does not allow overlap among the variables $x_{[i]}$ and therefore the constraints are separable. However, these variables $x_{[i]}$ may be linked together through the objective function $f(x)$.

*Received by the editors September 26, 2014; accepted for publication (in revised form) April 25, 2016; published electronically July 19, 2016. This research is supported by the National Natural Science Foundation of China (11101281, 11101257, 11271259, 11371102, and 91330201).
http://www.siam.org/journals/siopt/26-3/98907.html

†College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China (shenchungen@usst.edu.cn).
‡School of Mathematics and Shanghai Key Laboratory of Financial Information Technology, Shanghai University of Finance and Economics, Shanghai 200433, China (longzlh@163.com).
§Department of Mathematics, Fudan University, Shanghai 200433, China (whyang@fudan.edu.cn).

The reason that we are interested in BCOP is twofold: on the one hand, many practical applications that arise recently from, for example, correlation matrix approximation with factor structure [4, 25], factor models of asset returns [11], collateralized debt obligations [2, 12], multivariate time series [28], and the maximal correlation problem [9, 45, 46] can be recast in such form; on the other hand, general algorithms for nonlinearly constrained optimization may not be efficient as they generally do not take much advantage of the special structure of BCOP. Therefore, a custom-made algorithm for BCOP can provide a uniform and much more efficient tool for these applications. Very recently, Bai, Qi, and Xiu [3] proposed and studied an interesting problem of data representation on a sphere of unknown radius; the problem is relevant to BCOP but has its own structures and will not be considered in the present paper.

Relying upon the framework of the sequential quadratic programming (SQP) method, e.g., [5, 18, 19, 20, 27, 29, 37, 38], and making heavy use of the special structure of BCOP, we will refine the SQP method to propose a custom-made implementation. It is known that SQP is one of the most widely used methods for the general nonlinearly constrained optimization. In particular, it generates steps by solving quadratic subproblems (QPs). Traditional SQP method (see, e.g., [18]) takes a certain penalty function as the merit function to determine if a trial step is accepted or not. One known problem in this procedure is that a suitable penalty parameter is difficult to set. To get around that trouble, Fletcher and Leyffer [15] introduced the filter technique to globalize the SQP method, which turns out to be very efficient and effective, and is proved to be globally convergent [14, 16]. The filter technique was later applied to various problems and combined into other methods; examples include Ulbrich, Ulbrich, and Vincente [39], Karas et al. [24], Ribeiro, Karas, and Gonzaga [34], Wächter and Biegler [40, 41, 42], and Chiang and Zavala [8].

Unfortunately, when it is directly applied to solve BCOP, the classical SQP method based on QP subproblems encounters numerical difficulties if m and p_i get large. For instance, in the problem of the nearest correlation matrix (NCM) with $p = p_i$ $(i = 1, 2, \ldots, m)$ factors structure [4, 25] to be discussed in section 5 (see (5.1)), solving the corresponding QP subproblem is both time-consuming and memory demanding as m and p increase. It is nearly intractable with dimensions, say, $m = 500,\ p = 250$. As indicated in [4], both the Newton method and the classical SQP method fail to solve BCOP when m and p are large. The spectral projected gradient method (SPGM) is thus proposed in [4] to alleviate such heavy computational burden and uses less memory and computational costs at each iteration. The numerical results [4] show that SPGM is efficient for many medium-scale tested instances, but the number of iterations probably varies drastically from instance to instance and can perform worse in the case when p is close to m than in other situations.

Fortunately, the standard SQP method can be improved largely for BCOP by exploiting the special structure contained in the constraints. One of remarkable features is that the Jacobian matrix $\nabla c(x)$ is sparse and structured, which can be utilized to reduce computational amounts and memory requirements at each iteration. To do so, we employ the active-set technique [44, 43] to estimate the active set of inequalities associated with the minimizer and then, similar to QP-free methods [7, 17, 31, 32, 36, 43, 44], transform the QP subproblem into relevant linear systems. As m and p get large, the size of the resulting linear system can naturally be large too, but the limited memory BFGS (L-BFGS) [26] plus duality technique [38] can be effectively employed, which dramatically reduces the computational costs and memory requirements for the associated linear systems. By counting the detailed computational complexity for this procedure, we will see that there is a large amount of flops saved at each iteration. On the other hand, the local fast convergence can be preserved due to the SQP framework and the L-BFGS technique, and the global convergence is also guaranteed with the aid of filter technique. We apply this implementation to two specific practical applications: the correlation approximation problem [4, 25] and maximal correlation problem [9] in section 5; our numerical experiments demonstrate that the proposed method is robust and efficient and is competitive with some recently custom-designed methods for each individual application, including SPGM, the block relaxation method [4] and the majorization method [4] for the correlation approximation problem, and the Riemannian trust-region method [46] for the maximal correlation problem.

The rest of this paper is organized as follows. In the first part of section 2, we first reformulate the QP subproblem into a relevant linear system by duality and then introduce the L-BFGS technique to alleviate the computational burden in solving these linear systems; the detailed implementation by exploiting the sparsity of the Jacobian matrix $\nabla c(x)$ is stated; then we discuss the filter technique to globalize the SQP method; the overall algorithm is presented in the last part of section 2. In sections 3 and 4, we establish the global convergence and the local convergence rate of the proposed algorithm, respectively. The numerical experiments on the two specific applications are carried out in section 5, where we report our numerical experiences by comparing the performance of our algorithm with others. Concluding remarks are finally drawn in section 6.

There are a few words for notation. We denote the feasible region of BCOP by

$$\Omega := \{x | c_i(x) = 0, i \in \mathcal{E};\ c_i(x) \leq 0, i \in \mathcal{I}\}.$$

For the constraints $c_i(x)$ for $i = 1, 2, \ldots, m$, we let $c(x) = (c_1(x), \ldots, c_m(x))^T : \mathbb{R}^n \to \mathbb{R}^m$ and

$$\nabla c(x) = (\nabla c_1(x), \ldots, \nabla c_m(x)) \in \mathbb{R}^{n\times m};$$

for a particular index subset $\mathcal{J} = \{i_1, i_2, \ldots, i_j\}$ of $\{1, 2, \ldots, m\}$, we denote by $|\mathcal{J}|$ the cardinality of $\mathcal{J}$ and denote $c_{\mathcal{J}}(x) = (c_{i_1}(x), \ldots, c_{i_j}(x))^T : \mathbb{R}^n \to \mathbb{R}^j$ and

$$\nabla c_{\mathcal{J}}(x) = (\nabla c_{i_1}(x), \ldots, \nabla c_{i_j}(x)) \in \mathbb{R}^{n\times j};$$

thus the definitions of $c_{\mathcal{E}}(x)$, $c_{\mathcal{I}}(x)$ and $\lambda_{\mathcal{I}}$ follow naturally. Finally, suppose $\{\eta_k\}$ and $\{\nu_k\}$ are two vanishing sequences, where $\eta_k, \nu_k \in \mathbb{R}, k \in \mathbb{N}$; we denote

- $\eta_k = \mathcal{O}(\nu_k)$ if there exists a scalar $c > 0$ such that $|\eta_k| \leq c|\nu_k|$ for all k sufficiently large,
- $\eta_k = o(\nu_k)$ if $\lim_{k\to+\infty} \frac{\eta_k}{\nu_k} = 0$, and
- $\eta_k = \Theta(\nu_k)$ if both $\nu_k = \mathcal{O}(\eta_k)$ and $\eta_k = \mathcal{O}(\nu_k)$ hold.

2. Algorithm.

2.1. The working set. We begin with the first-order optimality conditions (or the KKT conditions), which can be written as

$$\nabla_x L(x, \lambda) = \nabla f(x) + \nabla c(x)\lambda = 0, \tag{2.1}$$

$$\lambda_i c_i(x) = 0,\ i \in \mathcal{I}, \tag{2.2}$$

$$c_i(x) \leq 0,\ \lambda_i \geq 0,\ i \in \mathcal{I}, \tag{2.3}$$

$$c_i(x) = 0,\ i \in \mathcal{E}, \tag{2.4}$$

where

$$L(x, \lambda) := f(x) + c(x)^T\lambda$$

is the Lagrange function and $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier.

As our method is based on the active-set approach, we next state the strategy to identify the active set. To this end, similar to [13, 21, 30], we first introduce the following function $\phi : \mathbb{R}^{n+m} \to \mathbb{R}$,

$$\phi(x, \lambda) = \sqrt{\|\Psi(x, \lambda)\|},$$

where $\Psi : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ is defined by

$$\Psi(x, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ c_{\mathcal{E}}(x) \\ \min\{-c_{\mathcal{I}}(x), \lambda_{\mathcal{I}}\} \end{pmatrix}.$$

Thus the set

$$\mathcal{A}_I(x, \lambda) = \left\{ i \in \mathcal{I} \mid c_i(x) \geq -\min\{\phi(x, \lambda), 10^{-6}\} \right\} \tag{2.5}$$

provides an estimation of the active set $I(x^*) = \{i \mid c_i(x^*) = 0, i \in \mathcal{I}\}$ of inequality constraints, where (x^*, λ^*) is the KKT point at the minimizer of BCOP. It is true that when (x, λ) is sufficiently close to (x^*, λ^*), the estimate $\mathcal{A}_I(x, \lambda)$ is accurate, provided both the Mangasarian–Fromovitz constraint qualification and the second-order sufficient condition (SOSC) hold at (x^*, λ^*) (see [30, Theorem 2.2]).

Now, suppose the current iteration (x^k, λ^k) is an approximation to (x^*, λ^*); then we define

$$\mathcal{A}_k := \mathcal{A}_I(x^k, \lambda^k) \cup \mathcal{E} \tag{2.6}$$

as our working set, which includes all equality constraints, nearly active indices of inequality constraints, and the indices of the violated inequality constraints. This choice of the working set is similar to [17, 43, 44] and is based on the following observations: it is reasonable to include $i \in \mathcal{I}$ whenever $c_i(x^k)$ is close to zero (say, $|c_i(x^k)| \leq 10^{-6}$); as for equality constraints and those violated inequality constraints (say $c_i(x^k) > 10^{-6}$), we include them in the working set in the hope of reducing the violation. After identifying the working set $\mathcal{A}_k$, a QP subproblem can be formulated which, by the QP-free technique [7, 17, 31, 32, 36, 43, 44], can alternatively be solved by solving a relevant linear system (details on the linear systems are discussed in the next subsection). The solution of the resulting linear system yields the search direction and generates curvature information of BCOP at (x^k, λ^k). One issue related to the linear system is the consistency, which is equivalent to the linear independence of the gradients of constraints corresponding to the working set $\mathcal{A}_k$. Due to the structure of BCOP, we prove in Lemma 2.1 that $\nabla c_{\mathcal{A}_k}(x^k)$ is of full column rank as long as x^k is confined to the set

$$\Omega_p := \{x \| \|x^k_{[i]}\|^2 \geq 0.5 \text{ for all } i \in \mathcal{E}\}.$$

Based on this fact, we can say that our choice of working set $\mathcal{A}_k$ does not invoke any complicated procedure as those in [36, 43, 44], where the working sets $\mathcal{A}_k$ should be determined via calculating the rank of $\nabla c_{\mathcal{A}_k}(x^k)$ or the determinant of $\nabla c_{\mathcal{A}_k}(x^k)^T \nabla c_{\mathcal{A}_k}(x^k)$ for each trial estimate $\mathcal{A}_k$ until $\nabla c_{\mathcal{A}_k}(x^k)$ is of full column rank.

LEMMA 2.1. *If $x^k \in \Omega_p$, then the vectors $\nabla c_i(x^k)$, $i \in \mathcal{A}_k$, are linearly independent, where $\mathcal{A}_k$ is defined in* (2.5)–(2.6).

Proof. Since $x^k \in \Omega_p$, it follows that $\|x^k_{[i]}\|^2 \geq 0.5$ for all $i \in \mathcal{E}$ and therefore $x^k_{[i]} \neq 0$ for all $i \in \mathcal{E}$. For $i \in \mathcal{A}_k \cap \mathcal{I}$, $c_i(x^k) = \|x^k_{[i]}\|^2 - 1 \geq -10^{-6}$ and therefore

$x^k_{[i]} \neq 0$. Suppose that there exist scalars $l_i \in \mathbb{R}$, $i \in \mathcal{A}_k$ such that $\sum_{i\in\mathcal{A}_k} l_i \nabla c_i(x^k) = 0$. Note that

$$\sum_{i\in\mathcal{A}_k} l_i \nabla c_i(x^k) = \sum_{i\in\mathcal{A}_k} \begin{pmatrix} 0 \\ \vdots \\ 2l_i x^k_{[i]} \\ \vdots \\ 0 \end{pmatrix}.$$

Because $x^k_{[i]} \neq 0$ for all $i \in \mathcal{A}_k$, we have that $l_i = 0$ for all $i \in \mathcal{A}_k$, which implies that $\nabla c_i(x^k)$, $i \in \mathcal{A}_k$ are linearly independent. □

Analogously, we have the following lemma.

LEMMA 2.2. *Suppose that $\{x^{k_l}, \lambda^{k_l}\}$ is a subsequence of $\{x^k, \lambda^k\}$ with $\{x^k\} \subset \Omega_p$ such that $\{x^{k_l}\}$ converges to x^* and $\mathcal{A}_{k_l} \equiv \mathcal{A}^*$ is a constant set for all sufficiently large l. Then $\nabla c_{\mathcal{A}^*}(x^*)$ is of full column rank.*

Proof. Since $x^{k_l} \in \Omega_p$ and $x^{k_l} \to x^*$, we have that $\|x^*_{[i]}\|^2 \geq 0.5$ for all $i \in \mathcal{E}$ and therefore $x^*_{[i]} \neq 0$ for all $i \in \mathcal{E}$. For $i \in \mathcal{A}^* \cap \mathcal{I}$, $c_i(x^{k_l}) \geq -10^{-6}$, and then $c_i(x^*) \geq -10^{-6}$ as $k_l \to \infty$. By the definition of $c(x)$, we also have that $x^*_{[i]} \neq 0$ for all $i \in \mathcal{A}^* \cap \mathcal{I}$. Analogous to the proof of Lemma 2.1, $\nabla c_i(x^*)$, $i \in \mathcal{A}^*$, are linearly independent as was to be shown. □

2.2. The QP subproblem and its reformulation. In this and the next subsections, we discuss how to compute the search direction at x^k. After the working set $\mathcal{A}_k$ is determined, the search direction d^k and its associated Lagrange multiplier λ^k can be determined via solving equality constrained QP subproblem(s) (probably two or three with different perturbed vectors $w_k \in \mathbb{R}^{\bar{m}}$ where $\bar{m} = |\mathcal{A}_k|$) in the form of

$$\begin{cases} \min\limits_{d\in\mathbb{R}^n} & \frac{1}{2} d^T B_k d + \nabla f(x^k)^T d \\ \text{s.t.} & \nabla c_{\mathcal{A}_k}(x^k)^T d = w_k, \end{cases} \tag{2.7}$$

where $B_k \in \mathbb{R}^{n\times n}$ is symmetric and positive definite that is an approximation of the Hessian of the Lagrangian function $L(x^k, \lambda^k)$. We point out that B_k can be updated by the BFGS formula [29]. The strategy of choosing different perturbed w_k is similar to [44, 43] and they correspond to two types of search directions d^k, which are designed for the purpose of the global convergence and locally superlinear convergence. In order to simplify the subsequent presentation, we identify these two cases by a boolean variable FAST, i.e., FAST=FALSE (for the global convergence) or FAST=TRUE (for the locally superlinear convergence), respectively. Details of the choice of w_k for the search direction are delayed until Algorithm 3 and Remark 2.2, and we next will discuss an efficient procedure for solving the solution d^k of (2.7).

It is evident that the equality constrained quadratic programming (2.7) is equivalent to the linear system

$$\begin{cases} B_k d + \nabla c_{\mathcal{A}_k}(x^k)\lambda = -\nabla f(x^k), \\ \nabla c_{\mathcal{A}_k}(x^k)^T d = w_k. \end{cases} \tag{2.8}$$

However, as n gets large, solving the linear system (2.8) can be expensive. In addition, without effectively exploiting the underlying sparse structure, the associated coefficient matrix could occupy too much memory. To resolve these numerical difficulties, we make use of the duality technique and solve the dual problem of (2.7)

$$\max_{\lambda \in \mathbb{R}^{\bar{m}}} \quad \tfrac{1}{2}\lambda^T W_k \lambda + b_k^T \lambda. \tag{2.9}$$

Note that (2.9) is an unconstrained optimization problem with relatively smaller size $\bar{m}$, where

$$W_k = \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla c_{\mathcal{A}_k}(x^k), \tag{2.10}$$

$$b_k = w_k + \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k). \tag{2.11}$$

Note that B_k is positive definite and therefore strong duality follows, which implies that the guess λ^k of the associated Lagrange multiplier can be obtained from solving (2.9), instead of (2.7). In particular, observing that $W_k \in \mathbb{R}^{\bar{m}\times\bar{m}}$ and $\bar{m} \le m$ is much smaller than n, solving the KKT condition of (2.9) or, equivalently, solving a much smaller linear system,

$$W_k \lambda = -b_k, \tag{2.12}$$

is inexpensive. Once λ^k is obtained from (2.12), putting it into the first equation in (2.8) yields

$$d^k = -B_k^{-1}\left(\nabla f(x^k) + \nabla c_{\mathcal{A}_k}(x^k)\lambda^k\right). \tag{2.13}$$

The above procedure resolves most numerical difficulties. The last issue is how to calculate W_k efficiently. The idea is to adopt the L-BFGS techinique, which is the topic of the next subsection.

2.3. Compute the search direction based on the L-BFGS formula. The limited memory BFGS method [29, Chapter 9] is one of the most effective and widely used methods in the field of large-scale unconstrained optimization. The main advantage is that the L-BFGS approach does not require one to calculate or store a full Hessian matrix, which might be too expensive for large-scale problems. For BCOP, we have pointed out that the matrix $W_k = \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla c_{\mathcal{A}_k}(x^k)$ in (2.10) needs to be computed. Note that $\nabla c_{\mathcal{A}_k}(x^k)$ is large but sparse and structured, and if we adopt the L-BFGS formula to update the inverse of the Hessian approximation B_k, much storage space and computational costs can be saved.

To describe the detailed procedure, let

$$S_k = [s_{k-l}, \ldots, s_{k-1}], \quad Y_k = [y_{k-l}, \ldots, y_{k-1}],$$

where $s_i = x^{i+1} - x^i$ and $y_i = \nabla L(x^{i+1}, \lambda^i) - \nabla L(x^i, \lambda^i)$, $i = k-l, \ldots, k-1$. One may notice that the solution λ^i to (2.12) is in $\mathbb{R}^{\bar{m}}$ rather than in $\mathbb{R}^m$, and plugging λ^i into $\nabla L(x^i, \lambda^i)$ is inappropriate. Nevertheless, we can augment λ^i by setting $\lambda^i_j = 0$ for $j \in \mathcal{I}\backslash\mathcal{A}_i = \mathcal{I}\backslash\mathcal{A}_I(x^i, \lambda^i)$. With this augment scheme, in what follows, we will use λ^i to denote the estimate multiplier in $\mathbb{R}^m$ as long as no confusion is caused. By the L-BFGS formula, the matrix B_k resulting from l updates to the basic matrix $B_0 = \nu_k I$ is given by

$$B_k = \nu_k I - \begin{pmatrix} \nu_k S_k & Y_k \end{pmatrix} \begin{pmatrix} \nu_k S_k^T S_k & L_k \\ L_k^T & -D_k \end{pmatrix}^{-1} \begin{pmatrix} \nu_k S_k^T \\ Y_k^T \end{pmatrix},$$

where $L_k,\ D_k \in \mathbb{R}^{l\times l}$ are defined by

$$(L_k)_{i,j} = \begin{cases} (s_{k-l-1+i})^T(y_{k-l-1+i}) & \text{if } i > j, \\ 0 & \text{otherwise,} \end{cases}$$
$$D_k = \operatorname{diag}(s_{k-l}^T y_{k-l}, \ldots, s_{k-1}^T y_{k-1}),$$

and $\nu_k = \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}}$. To ensure the positive definiteness of B_{k+1}, we adopt the so-called damped BFGS technique to modify y_k so that $s_k^T y_k$ is "sufficiently" positive. Let $y_k \leftarrow \theta_k y_k + (1-\theta_k)B_k s_k$, where the scalar θ_k is defined as

$$\theta_k = \begin{cases} 1 & \text{if } s_k^T y_k \geq C_2 s_k^T B_k s_k, \\ (C_1 s_k^T B_k s_k)/(s_k^T B_k s_k - s_k^T y_k) & \text{if } s_k^T y_k < C_2 s_k^T B_k s_k, \end{cases}$$

where $C_1 \in (\frac{1}{2}, 1)$ and $C_2 \in (0, \frac{1}{2})$ are two scalars. In our numerical experiment, C_1 and C_2 are set to 0.98 and 0.02, respectively. We then use s_k and the modified y_k to update S_{k+1} and Y_{k+1}, respectively.

Let H_k denote the inverse of B_k; then the update formula for H_k is given by

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \tag{2.14}$$

where $\rho_k = \frac{1}{y_k^T s_k}$ and $V_k = I - \rho_k y_k s_k^T$. Using the information (S_k and Y_k) of the last l iterations and choosing $\delta_k I$ with $\delta_k = \frac{1}{\nu_k}$ as the initial approximation H_k^0, we obtain by repeatedly applying (2.14) that

$$H_k = H_k^f + H_k^s,$$

where

$$H_k^f = \delta_k (V_{k-1}^T \cdots V_{k-l}^T)(V_{k-l} \cdots V_{k-1})$$

and

$$\begin{aligned} H_k^s = {} & \rho_{k-l}(V_{k-1}^T \cdots V_{k-l+1}^T) s_{k-l} s_{k-l}^T (V_{k-l+1} \cdots V_{k-1}) \\ & + \rho_{k-l+1}(V_{k-1}^T \cdots V_{k-l+2}^T) s_{k-l+1} s_{k-l+1}^T (V_{k-l+2} \cdots V_{k-1}) + \cdots + \rho_{k-1} s_{k-1} s_{k-1}^T. \end{aligned}$$

For simplicity, we denote $\nabla c_{\mathcal{A}_k}(x^k)$ by A_k. It then follows from (2.10) that

$$W_k = A_k^T H_k A_k = A_k^T H_k^f A_k + A_k^T H_k^s A_k. \tag{2.15}$$

Since the matrix A_k is sparse (no more than n nonzero elements) and V_k is structured, we are able to accomplish matrix-chain multiplication for $A_k^T H_k^f A_k$ and $A_k^T H_k^s A_k$ rather efficiently, through transformation of the rightmost side of (2.15). In particular, it is straightforward that

$$\begin{aligned} (V_{k-l} \cdots V_{k-1}) A_k = A_k & \\ & - \rho_{k-1} y_{k-1} s_{k-1}^T A_k \\ & - \cdots \\ & - \rho_{k-l+1} y_{k-l+1} s_{k-l+1}^T (V_{k-l+2} \cdots V_{k-1}) A_k \\ & - \rho_{k-l} y_{k-l} s_{k-l}^T (V_{k-l+1} \cdots V_{k-1}) A_k. \end{aligned}$$

Let $q_i = \rho_i s_i^T (V_{i+1} \cdots V_{k-1}) A_k$ for $i = k-l, \dots, k-2$ and $q_{k-1} = \rho_{k-1} s_{k-1}^T A_k$. It then follows that

$$A_k^T H_k^f A_k = \delta_k \left(A_k^T - \sum_{i=k-l}^{k-1} q_i^T y_i^T \right) \left(A_k - \sum_{i=k-l}^{k-1} y_i q_i \right)$$

$$= \delta_k A_k^T A_k + \sum_{i=k-l}^{k-1} \sum_{j=k-l}^{k-1} \delta_k (y_i^T y_j) q_i^T q_j - \sum_{i=k-l}^{k-1} \delta_k (q_i^T y_i^T A_k + A_k^T y_i q_i). \tag{2.16}$$

Using q_i, the last item in (2.15) can be rewritten as

$$A_k^T H_k^s A_k = \sum_{i=k-l}^{k-1} \frac{q_i^T q_i}{\rho_i}. \tag{2.17}$$

Consequently, based on (2.16) and (2.17), the whole procedure for computing $W_k = A_k^T H_k A_k$ can be summarized by the pseudocode in Algorithm 1. We remark that the procedure between lines 2 and 13 computes $W_k^s = A_k^T H_k^s A_k$ and between lines 15 and 25 computes $W_k^f = A_k^T H_k^f A_k$, and line 26 finally forms W_k.

Remark 2.1. We finally count the computational complexity of computing W_k in Algorithm 1. For this purpose, we assume $p_i = p$ for $i = 1, 2, \dots, m$, only for simplicity. First, it requires at most (because $\bar{m} \leq m$)

$$(2l^2 + l + 2)mp + 2lm^2 + \mathcal{O}(m) \text{ flops}$$

for computing $W_k^s = A_k^T H_k^s A_k$ (lines 2–13) and costs at most

$$\left(\frac{3}{2}l^2 + \frac{7}{2}l + 3\right) mp + \left(\frac{3}{2}l^2 + \frac{7}{2}l\right) m^2 + \mathcal{O}(m) \text{ flops}$$

for $W_k^f = A_k^T H_k^f A_k$ (lines 15–25). Note that $mp = n$, and this implies that for $l \ll n$, computation of W_k requires at most $\mathcal{O}(m^2 + mp) = \mathcal{O}(m^2 + n)$ flops. As for b_k in (2.11) and d^k in (2.13), the main computational effort is to compute the matrix-vector product $H_k z$. Applying [29, Algorithm 9.1], it is easy to know that $6lmp = 6ln$ flops are required for computing $H_k z$, and therefore, computation of b_k and d^k needs at most $12lmp + 6mp = (12l + 6)n$ flops.

2.4. The NLP filter. Suppose we have the search direction d^k; then the step size α^k is the next important ingredient that determines the iterate

$$x^{k+1} := x^k + \alpha^k d^k.$$

In choosing α^k, we will use the filter method and the backtracking line search procedure. In particular, we will generate a decreasing sequence of trials for $\alpha^k \in (\alpha_{\min}^k, 1]$ until our preset acceptance criterion is fulfilled or the feasibility restoration phase (subsection 2.5) is called. Here, $\alpha_{\min}^k \geq 0$ is a lower bound of α^k and we will give an explicit formula of $\alpha_{\min}^k$ in the next subsection.

Let

$$\hat{x} := x^k + \hat{\alpha} d^k, \quad \hat{\alpha} \in (\alpha_{\min}^k, 1]$$

denote a trial point. Using

$$h(x) = \left\| \begin{pmatrix} c_{\mathcal{E}}(x) \\ \max\{c_{\mathcal{I}}(x), 0\} \end{pmatrix} \right\|_\infty$$

Algorithm 1: Procedure for computing W_k based on the L-BFGS formula.

Data: S_k, Y_k, A_k, δ_k
Result: W_k

% Compute $W_k^s = A_k^T H_k^s A_k$
for $i = k-l, \ldots, k-1$ **do**
  $\rho_i = 1/y_i^T s_i$;
end
$W_k^s = 0$;
for $i = k-1, \ldots, k-l$ **do**
  $u = s_i^T$;
  for $j = i, \ldots, k-2$ **do**
    $u = u - \rho_{j+1}(u y_{j+1}) s_{j+1}^T$;
  end
  $q_i = \rho_i u A_k$
  % $q_i = \begin{cases} \rho_i s_i^T (V_{i+1} \cdots V_{k-1}) A_k, & i = k-l, \ldots, k-2 \\ \rho_{k-1} s_{k-1}^T A_k, & i = k-1 \end{cases}$
  $W_k^s = W_k^s + q_i^T (q_i/\rho_i)$;
end
% Compute $W_k^f = A_k^T H_k^f A_k$
$W_k^f = \delta_k A_k^T A_k$
for $i = k-l+1, \ldots, k-1$ **do**
  for $j = k-l+1, \ldots, i$ **do**
    $\beta = \delta_k (y_i^T y_j)$;
    $W_k^f = W_k^f + (\beta q_i)^T q_j$;
    if $j < i$ **then**
      $W_k^f = W_k^f + q_j^T (\beta q_i)$;
    end
  end
  $W_k^f = W_k^f - (\delta_k q_i^T)(y_i^T A_k) - (A_k^T y_i)(\delta_k q_i)$;
end
$W_k = W_k^f + W_k^s$;

as a measure of infeasibility at the point x, we now give relevant definitions about the filter. The first one, Definition 2.1, is a variant of [16, (2.6)].

DEFINITION 2.1. *For given* $\beta \in (0,1)$ *and* $\gamma \in (0,1)$, *a trial point* $\hat{x}$ *(or equivalently the pair* $(h(\hat{x}), f(\hat{x}))$*) is* acceptable *to* x^l *(or equivalently the pair* $(h(x^l), f(x^l))$*), if*

$$h(\hat{x}) \le \beta h(x^l) \quad \text{or} \tag{2.18}$$

$$f(\hat{x}) \le f(x^l) - \gamma \min\{h(\hat{x}), h(\hat{x})^2\}. \tag{2.19}$$

In the original paper of Fletcher and Leyffer [15], a pair $(h(\hat{x}), f(\hat{x}))$ is said to *dominate* $(h(x^l), f(x^l))$ if both (2.18) and (2.19) hold with $\beta = 1$ and $\gamma = 0$, and a filter is defined as a list of pairs $(h(x^l), f(x^l))$ such that no pair dominates any other in this filter [15, Definition 2]. The condition (2.19) is a variant of [16, (2.6)] where $f(\hat{x}) \le f(x^l) - \gamma h(\hat{x})$. Note that (2.19) is equivalent to $f(\hat{x}) \le f(x^l) - \gamma h(\hat{x})$ if

$h(\hat{x}) \geq 1$ and $f(\hat{x}) \leq f(x^l) - \gamma h(\hat{x})^2$ otherwise. The reason to introduce this modified condition on $h(\hat{x})$ is that we prefer to accept the trial point $\hat{x}$ for the purpose of convergence whenever the violation of the feasibility is not severe, i.e., $h(\hat{x}) < 1$.

Similar to the original definition of the filter in [15], based on Definition 2.1, we define our filter, denoted by $\mathcal{F}_k$ at the iteration k, as a set of pairs $(h(x^l), f(x^l))$ such that any pair in the filter is acceptable to all previous pairs in $\mathcal{F}_k$ in the sense of Definition 2.1. Initially with $k = 0$, the filter $\mathcal{F}_k$ can begin with the pair $(\chi, -\infty)$, where $\chi > 0$ is imposed on $h(\hat{x})$ as an upper bound to control the constraint violation [15]. At the start of iteration k, the current pair $(h(x^k), f(x^k)) \notin \mathcal{F}_k$ but must be acceptable to it, while at the end of iteration k, the pair $(h(x^k), f(x^k))$ may or may not be added to $\mathcal{F}_k$, depending on our acceptance rule to be discussed in Remark 2.3. But once $(h(x^k), f(x^k))$ is added to $\mathcal{F}_k$, we remove all pairs in the current filter $\mathcal{F}_k$ which are worse than $(h(x^k), f(x^k))$ with respect to both the objective function value and the constraint violation; the detailed procedure for updating the filter $\mathcal{F}_k$ will be described in Algorithm 3 and Remark 2.3.

Definition 2.2. *A trial point $\hat{x}$ (or a pair $(h(\hat{x}), f(\hat{x}))$) is* acceptable to the filter *$\mathcal{F}_k$, if $\hat{x}$ (or a pair $(h(\hat{x}), f(\hat{x}))$) is acceptable to x^l in the sense of Definition* 2.1*, for all $l \in \bar{\mathcal{F}}_k := \{l \mid (h(x^l), f(x^l)) \in \mathcal{F}_k\}$.*

The trial point $\hat{x}$ is to be accepted as the next iteration if it is acceptable both to x^k (by Definition 2.1) and to the filter $\mathcal{F}_k$ (by Definition 2.2). Nevertheless, such acceptance rule for the trial $\hat{x}$ may cause the situation that we always accept the points that satisfy (2.18) alone, but not (2.19). This would result in an iterative sequence converging to a feasible, but nonoptimal, point. To avoid this situation, we impose an additional condition on $\hat{x}$:

Case 1. When FAST=FALSE[1] or $\hat{\alpha} < 1$,

$$\text{if} \quad -\hat{\alpha}\nabla f(x^k)^T d^k > \delta h^2(x^k), \tag{2.20}$$

then accepting $\hat{x}$ as the next iterate x^{k+1} while the following condition holds

$$f(\hat{x}) \leq f(x^k) + \hat{\alpha}\eta\nabla f(x^k)^T d^k. \tag{2.21}$$

Case 2. When FAST=TRUE[2] and $\hat{\alpha} = 1$,

$$\text{if} \quad -\nabla f(x^k)^T d^k > \delta h^2(x^k) \;\text{ and }\; h(x^k) \leq \zeta_1 \|d^k\|^{\zeta_2}, \tag{2.22}$$

then accepting $\hat{x}$ as the next iterate x^{k+1} while the following condition holds:

$$f(\hat{x}) \leq f(x^k) - \eta \min\{-\nabla f(x^k)^T d^k, \xi\|d^k\|^{\zeta_2}\}, \tag{2.23}$$

where $\zeta_1 > 0$, $\zeta_2 \in (2, 3)$, $\xi > 0$, $\eta \in (0, \frac{1}{2})$, and $\delta > 0$ is chosen to satisfy $\delta \geq \gamma/\eta$.

Note that Case 1 and Case 2 are mutually exclusive. The motivation for these conditions is from [35, section 2]. The switching condition for Case 1 and Case 2 and the sufficient reduction conditions (2.21) and (2.23) are useful for the global convergence and the fast local convergence as well: If (2.20) for Case 1 is satisfied, then the direction d^k is descent for $f(x)$, and therefore imposing the reduction condition (2.21) on $f(x)$ is helpful for the global convergence; if (2.22) for Case 2 is satisfied,

[1] FAST=FALSE indicates that d^k is a step contributing to global convergence.
[2] FAST=TRUE indicates that d^k is a step contributing to fast local convergence.

implying d^k a search direction for fast local convergence, the full step (i.e., $\hat{\alpha} = 1$) is expected so that the fast local convergence can be achieved. Note that the condition (2.23) is more relaxed than (2.21) as we prefer to accept the full step.

Finally, we are able to state our rule for accepting the trial point $\hat{x}$ as the next iterate.

> **Acceptance rule.**
> A trial point $\hat{x}$ is accepted as the next iterate x^{k+1} if it is acceptable to $\mathcal{F}_k \cup \{(h(x^k), f(x^k))\}$, and one of the following two conditions holds:
> (i) either (2.20) and (2.21) for Case 1 or (2.22) and (2.23) for Case 2 are satisfied;
> (ii) (2.20) for Case 1 or (2.22) for Case 2 is not satisfied.

If the trial point $\hat{x}$ does not satisfy $\hat{x} \in \Omega_p$ or the acceptance rule, we shrink $\hat{\alpha}$ until the trial point is accepted or $\hat{\alpha} \leq \alpha^k_{\min}$. Once the latter occurs, the feasibility restoration phase is called, which is discussed in the next subsection.

2.5. Feasibility restoration phase. Motivated by [40], we define the lower bound $\alpha^k_{\min}$ of $\hat{\alpha}$ by

$$\alpha^k_{\min} = \begin{cases} \min\left\{1-\beta, -\frac{\gamma h(x^k)}{\nabla f(x^k)^T d^k}, -\frac{\delta h^2(x^k)}{\nabla f(x^k)^T d^k}\right\}, & \nabla f(x^k)^T d^k < 0, \\ \alpha_\phi & \text{otherwise}, \end{cases} \tag{2.24}$$

where α_ϕ is a positive scalar. Through shrinking $\hat{\alpha}$, if we cannot find a step size $\hat{\alpha} \in (\alpha^k_{\min}, 1]$ such that the trial point $\hat{x}$ is accepted by the acceptance rule, we then turn to the feasibility restoration phase. Note that when the iteration gets into the restoration phase, x^k is infeasible. If x^k is feasible, $h(x^k) = 0$ and there must be some $\hat{\alpha} \in (\alpha^k_{\min}, 1]$ so that $\hat{x}$ is accepted (see Lemma 3.9). Based on these facts, in the restoration phase, we project x^k onto Ω to get the next iterate $x^{k+1} = \mathrm{P}_\Omega(x^k)$. Since the feasible set Ω is of special structure, projecting x^k onto Ω (Algorithm 2) is easy and costs only at most $3n$ flops.

Algorithm 2: $\mathrm{P}_\Omega(x^k)$: projection x^k onto Ω.

```
Given x^k;
for i = 1,...,m do
    if (i ≤ m_1 & ||x^k_[i]|| ≠ 1) or (i > m_1 & ||x^k_[i]|| > 1) then
        x^k_[i] ← x^k_[i] / ||x^k_[i]||;
    end
end
return x^k;
```

2.6. The statement of algorithm. We now state the overall algorithm.

Remark 2.2. In Algorithm 3, lines 3–18 state the procedure for computing the search direction d^k, the guess of Lagrange multiplier λ^k, together with some other quantities (say, $d^{k,0}$, $\lambda^{k,0}$, $d^{k,1}$, $\lambda^{k,1}$, $d^{k,2}$, and $\lambda^{k,2}$, etc.) related to d^k and λ^k, while lines 19–23 describe the procedure for the step size α^k. In computing the search direction between lines 3 and 18, there are two different cases:

(i) FAST=TRUE. The pair $(d^k, \lambda^k) = (d^{k,0}, \lambda^{k,0})$ solves

$$\begin{cases} B_k d^{k,0} + \nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,0} = -\nabla f(x^k), \\ \nabla c_{\mathcal{A}_k}(x^k)^T d^{k,0} = -c_{\mathcal{A}_k}(x^k), \end{cases} \tag{2.26}$$

Algorithm 3: Filter active-set method (FilterASM).

Given $x^0 \in \Omega_p$, $\chi > h(x^0)$, $\nu \in (2,3)$, $\beta \in (0,1)$, $\gamma \in (0,1)$, $\eta \in (0,\frac{1}{2})$, $\delta \geq \frac{\gamma}{\eta}$, $\xi > 0$, $\alpha_\phi \in (0,\frac{1}{2})$, $\zeta_1 > 0$, $\zeta_2 \in (2,3)$, $r \in (0,1)$. Initialize $\mathcal{F}_0$ with the pair $(\chi, -\infty)$;

for $k = 0,1,2,\ldots,maxit$ **do**

Determine the working set $\mathcal{A}_k$;

Compute $\lambda^{k,0}$ by (2.12) with $w_k = -c_{\mathcal{A}_k}(x^k)$ and $d^{k,0}$ by (2.13) with $\lambda^k = \lambda^{k,0}$

if $d^{k,0} = 0$ and $\lambda_i^{k,0} \geq 0$ $(\forall i \in \mathcal{A}_k \cap \mathcal{I})$, **stop**

% `Termination condition`

if

$$\lambda_i^{k,0} \geq 0 \quad \forall i \in \mathcal{A}_k \cap \mathcal{I}, \tag{2.25}$$

then

Set FAST=TRUE, $d^k = d^{k,0}$, $\lambda^k = \lambda^{k,0}$, and $w_k = -c_{\mathcal{A}_k}(x^k) - c_{\mathcal{A}_k}(x^k + d^k) - \|d^{k,0}\|^\nu e$;

else

Set FAST=FALSE, and $w_k = 0$;

end

Compute $\lambda^{k,1}$ by (2.12) with w_k and compute $d^{k,1}$ by (2.13) with $\lambda^k = \lambda^{k,1}$;

if *FAST=TRUE* **then**

Set $\hat{d}^k = \begin{cases} 0 & \text{if } \|d^{k,1} - d^{k,0}\| > \|d^{k,0}\|, \\ d^{k,1} - d^{k,0} & \text{otherwise;} \end{cases}$

else

Set

$[u_k]_i = \begin{cases} \min\{-c_{j_i}(x^k), 0\} + \lambda_{j_i}^{k,1}, & \lambda_{j_i}^{k,1} < 0 \ (j_i \in \mathcal{A}_k \cap \mathcal{I}), \\ -c_{j_i}(x^k), & \text{others,} \end{cases}$ where $\mathcal{A}_k = \{j_1, \ldots, j_{|\mathcal{A}_k|}\}$;

Compute $\lambda^{k,2}$ by (2.12) with $w_k = u_k$ and compute $d^{k,2}$ by (2.13) with $\lambda^k = \lambda^{k,2}$;

Set $d^k = d^{k,2}$, $\lambda^k = \lambda^{k,1}$;

end

if *FAST=FALSE or* $x^k + d^k + \hat{d}^k$ *does not satisfy the acceptance rule, or* $x^k + d^k + \hat{d}^k \notin \Omega_p$ **then**

Find $\alpha^k > \alpha^k_{\min}$, the first number α^k of the sequence $\{1, r, r^2, \ldots\}$ such that $\hat{x} = x^k + \alpha^k d^k$ satisfies the acceptance rule and $\hat{x} \in \Omega_p$;

else

Set $\hat{x} = x^k + d^k + \hat{d}^k$ and $\alpha^k = 1$;

end

if *the above* α^k *(i.e.,* $\alpha^k > \alpha^k_{\min}$*) does not exist* **then**

Go to the feasibility restoration phase to get $x^{k+1} = \mathrm{P}_\Omega(x^k)$ and add $(h(x^k), f(x^k))$ to $\mathcal{F}_k$;

else

if (2.20) *for Case* 1 *or* (2.22) *for Case* 2 *does not hold* **then** add $(h(x^k), f(x^k))$ to $\mathcal{F}_k$;

Set $x^{k+1} = \hat{x}$, $s_k = x^{k+1} - x^k$, $y_k = \nabla L(x^{k+1}, \lambda^k) - \nabla L(x^k, \lambda^k)$, and update $S_k,\ Y_k$ to $S_{k+1},\ Y_{k+1}$.

end

end

which is a quasi-Newton equation of KKT system (2.1)–(2.4) at the working set $\mathcal{A}_k$. To achieve fast local convergence and to overcome the Maratos effect, we adopt the second-order correction technique. In particular, we compute the second-order correction step by setting $\hat{d}^k = d^{k,1} - d^{k,0}$, where $d^{k,1}$ is from

$$\left\{ \begin{array}{l} B_k d^{k,1} + \nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,1} = -\nabla f(x^k), \\ \nabla c_{\mathcal{A}_k}(x^k)^T d^{k,1} = -(c_{\mathcal{A}_k}(x^k + d^k) + c_{\mathcal{A}_k}(x^k) + \|d^{k,0}\|^\nu e). \end{array} \right. \tag{2.27}$$

Here, $e = (1, 1, \ldots, 1)^T$ with appropriate dimension. Then we check if $\hat{x} = x^k + d^k + \hat{d}^k$ satisfies the acceptance rule. If it fails, this second-order correction step $\hat{d}^k$ is discarded, and the backtracking technique is invoked to find a step size α^k such that $x^k + \alpha^k d^k$ is accepted.

(ii) FAST=FALSE. The search direction $d^k = d^{k,2}$ is computed by solving

$$\left\{ \begin{array}{l} B_k d^{k,2} + \nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,2} = -\nabla f(x^k), \\ \nabla c_{\mathcal{A}_k}(x^k)^T d^{k,2} = u_k, \end{array} \right. \tag{2.28}$$

where u_k (line 15) uses the information of $\lambda^{k,1}$ from the system

$$\left\{ \begin{array}{l} B_k d^{k,1} + \nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,1} = -\nabla f(x^k), \\ \nabla c_{\mathcal{A}_k}(x^k)^T d^{k,1} = 0. \end{array} \right. \tag{2.29}$$

We explain the above two linear systems as follows: the solution $d^{k,1}$ of (2.29) is in the null space of $\nabla c_{\mathcal{A}_k}(x^k)^T$ and targets improving $f(x)$ rather than $h(x)$; because $d^{k,1}$ may be close to zero with a negative multiplier $\lambda^{k,1}$, a slight perturbation system (2.28) of (2.29) is to be solved and yields a new direction $d^{k,2}$, which aims at improving $h(x)$ instead, and prevents the unwelcome effect caused by a negative multiplier. In all, d^k in this case contributes to the global convergence.

Remark 2.3. The filter $\mathcal{F}_k$ is updated in either line 25 or line 27. In other words, the pair $(h(x^k), f(x^k))$ is added to $\mathcal{F}_k$ and we remove all other pairs in $\mathcal{F}_k$ dominated by $(h(x^k), f(x^k))$ if (2.20) for Case 1 or (2.22) for Case 2 is not fulfilled or the restoration phase is invoked.

Remark 2.4. For the sake of convenience for analyzing the convergence, we borrow the terminology from Fletcher, Leyffer, and Toint [16]: we call an iterate an *f-type iterate* if $x^{k+1} = x^k + \alpha^k d^k$ (or $x^{k+1} = x^k + d^k + \hat{d}^k$) is accepted according to (i) of the acceptance rule; otherwise, we call the iterate an *h-type iterate*, which means that x^{k+1} is accepted according to (ii) of the acceptance rule or is recovered from the feasibility restoration phase.

3. Global convergence. In this section we show the global convergence of Algorithm 3 under the following two assumptions:

(A1) The objective function $f(x)$ is twice continuously differentiable.

(A2) The matrix B_k is bounded and uniformly positive definite for all k; that is, there exists a scalar $\tau > 0$ such that $\frac{1}{\tau}\|d\|^2 \le d^T B_k d \le \tau\|d\|^2$ holds for any $d \in \mathbb{R}^n$ and any k.

We begin with the boundedness of the iterates.

LEMMA 3.1. *The sequence $\{x^k\}$ generated by Algorithm 3 is bounded.*

Proof. Since all iterates from Algorithm 3 satisfy the upper bound condition $h(x^k) \le \chi$ because $\mathcal{F}_0 = \{(\chi, -\infty)\}$, combining with the definitions of $h(x)$ directly leads to the boundedness of $\{x^k\}$. ∎

THEOREM 3.2. *Suppose that assumption* (A1) *holds. Let* $\{x^{k_l}\}$ *be an infinite subsequence of* $\{x^k\}$ *on which* $(h(x^{k_l}), f(x^{k_l}))$ *is added into the filter. Then*

$$\lim_{k_l \to \infty} h(x^{k_l}) = 0.$$

Proof. From assumption (A1) and Lemma 3.1, we know that $\{f(x^{k_l})\}$ is bounded from below. Applying [35, Lemma 3.1] yields the assertion. ◻

Theorem 3.2 implies that all accumulation points of $\{x^{k_l}\}$ on which $(h(x^{k_l}), f(x^{k_l}))$ is added into the filter are feasible points for BCOP.

LEMMA 3.3. *Suppose that assumptions* (A1)–(A2) *hold. If FAST=TRUE, then the sequence* $\{(d^{k,0}, \lambda^{k,0})\}$ *is bounded; if FAST=FALSE, then both sequences* $\{(d^{k,1}, \lambda^{k,1})\}$ *and* $\{(d^{k,2}, \lambda^{k,2})\}$ *are bounded.*

Proof. From Algorithm 3, $\lambda^{k,0} = -W_k^{-1} b_k$ with

$$b_k = -c_{\mathcal{A}_k}(x^k) + \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k)$$

in the case of FAST=TRUE, where $W_k = \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla c_{\mathcal{A}_k}(x^k)$ is uniformly positive definite for all k due to Lemmas 2.2, and 3.1 and assumption (A2). Again using Lemma 3.1 and assumption (A2), b_k is bounded and therefore $\lambda^{k,0}$ is bounded too, which together with the boundedness of B_k^{-1}, x^k, and $\lambda^{k,0}$ implies that d^k in (2.13) is bounded for all k.

Analogously, in the case of FAST=FALSE, W_k and its inverse are bounded for all k. Lemma 3.1 and assumption (A2) ensure the boundedness of $\nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k)$. Since $\lambda^{k,1} = -W_k^{-1} \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k)$ and $d^{k,1} = -B_k^{-1}\left(\nabla f(x^k) + \nabla c_{\mathcal{A}_k}(x^k)\lambda^k\right)$, it follows that both $\lambda^{k,1}$ and $d^{k,1}$ are bounded for all k. In view of the definition of u_k (see line 15 of Algorithm 3) and the boundedness of $\{x^k\}$, u_k is bounded too, which implies the boundedness of $\lambda^{k,2} = -W_k^{-1}(u_k + \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k))$. Consequently, $d^{k,2}$ in (2.13) with $\lambda^{k,2}$ is bounded for all k. ◻

Remark 3.1. Based on the previous lemmas, for the convenience of further reference, we assume $\|d^{k,j}\| \le M_d,\ j = 0, 1, 2$ and $\|\lambda^{k,j}\| \le M_\lambda,\ j = 0, 1, 2$ for all k, where $M_d > 0$ and $M_\lambda > 0$ are two constants.

LEMMA 3.4. *Under assumptions* (A1)–(A2)*, the following two statements are true:*

(i) *If FAST=TRUE and* $d^k = 0$*, then* x^k *is a KKT point of BCOP.*

(ii) *If FAST=FALSE,* $h(x^k) = 0$ *and* $\nabla f(x^k)^T d^k = 0$*, then* x^k *is a KKT point of BCOP.*

Proof. (i) Since $\lambda^{k,0}$ is from (2.12) with $b_k = -c_{\mathcal{A}_k}(x^k) + \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k)$, rearranging (2.12) leads to $c_{\mathcal{A}_k}(x^k) = W_k \lambda^{k,0} + \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k)$ which, using (2.13) and the definition of W_k, gives

$$c_{\mathcal{A}_k}(x^k) = \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1}(\nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,0} + \nabla f(x^k)) = -\nabla c_{\mathcal{A}_k}(x^k)^T d^{k,0}.$$

Putting $d^{k,0} = d^k = 0$ into the above equation yields $c_{\mathcal{A}_k}(x^k) = 0$; now combining with the definition of $\mathcal{A}_k$ implies that x^k is feasible, that is, $c_{\mathcal{E}}(x^k) = 0$ and $c_{\mathcal{I}}(x^k) \le 0$. From assumption (A2) and (2.13), $d^{k,0} = 0$ leads to $\nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,0} + \nabla f(x^k) = 0$, which shows the dual feasibility at x^k. In addition, the nonnegativeness of $\lambda^{k,0}$ is guaranteed by the mechanism of Algorithm 3 (in the case of FAST=TRUE). Thus, x^k satisfies a variant of the KKT conditions (2.1)–(2.4) and therefore is a KKT point.

(ii) By Algorithm 3, if FAST=FALSE, then

$$\lambda^{k,1} = -W_k^{-1} \nabla c_{\mathcal{A}_k}(x^k)^T B_k^{-1} \nabla f(x^k), \tag{3.1}$$

$$d^{k,1} = -B_k^{-1}(\nabla f(x^k) + \nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,1}), \tag{3.2}$$

$$\lambda^{k,2} = -W_k^{-1}u_k + \lambda^{k,1}, \tag{3.3}$$

$$d^{k,2} = d^{k,1} + B_k^{-1}\nabla c_{\mathcal{A}_k}(x^k)W_k^{-1}u_k. \tag{3.4}$$

From (3.4) and (3.1), we have that

$$\begin{aligned}\nabla f(x^k)^T d^{k,2} &= \nabla f(x^k)^T d^{k,1} + \nabla f(x^k)^T B_k^{-1}\nabla c_{\mathcal{A}_k}(x^k)W_k^{-1}u_k \\ &= \nabla f(x^k)^T d^{k,1} - (\lambda^{k,1})^T u_k. \end{aligned}\tag{3.5}$$

By premultiplying the first equation of (2.29) by $(d^{k,1})^T$ and using the second equation of (2.29), we get $\nabla f(x^k)^T d^{k,1} = -(d^{k,1})^T B_k d^{k,1}$. Substituting it into (3.5) yields

$$\nabla f(x^k)^T d^{k,2} = -(d^{k,1})^T B_k d^{k,1} - (\lambda^{k,1})^T u_k. \tag{3.6}$$

According to hypothesis (ii) of this lemma, $c_{\mathcal{E}}(x^k) = 0$, $c_{\mathcal{I}}(x^k) \leq 0$, and $\nabla f(x^k)^T d^{k,2} = 0$. Combining with the definition of u_k at line 15 in Algorithm 3, the second term in the right-hand side of (3.6) can be changed to

$$\sum_{\lambda_i^{k,1}<0, i\in\mathcal{A}_k\cap\mathcal{I}} [(\lambda_i^{k,1})^2 + \max\{-\lambda_i^{k,1}c_i(x^k), 0\}] - \sum_{\lambda_i^{k,1}\geq 0, i\in\mathcal{A}_k\cap\mathcal{I}} \lambda_i^{k,1}c_i(x^k),$$

and then

$$\begin{aligned}0 = -(d^{k,1})^T B_k d^{k,1} &- \sum_{\lambda_i^{k,1}<0, i\in\mathcal{A}_k\cap\mathcal{I}} [(\lambda_i^{k,1})^2 + \max\{-\lambda_i^{k,1}c_i(x^k), 0\}] \\ &+ \sum_{\lambda_i^{k,1}\geq 0, i\in\mathcal{A}_k\cap\mathcal{I}} \lambda_i^{k,1}c_i(x^k).\end{aligned}$$

It is easy to see that the first two terms (the sign excluded) in the right-hand side are nonnegative and the last term is nonpositive, which implies that all terms in the right-hand side must be zero. In particular, the first term $(d^{k,1})^T B_k d^{k,1} = 0$ implies the primal optimality condition $\nabla c_{\mathcal{A}_k}(x^k)\lambda^{k,1} + \nabla f(x^k) = 0$ due to assumption (A2) and (3.2); the second term is indeed absent, because otherwise from

$$\sum_{\lambda_i^{k,1}<0, i\in\mathcal{A}_k\cap\mathcal{I}} [(\lambda_i^{k,1})^2 + \max\{-\lambda_i^{k,1}c_i(x^k), 0\}] = 0,$$

it implies $\lambda_i^{k,1} = 0$, which is a contradiction; and the third term equal to zero implies $\lambda_i^{k,1}c_i(x^k) = 0,\ i \in \mathcal{A}_k \cap \mathcal{I}$, which gives the complementarity condition. Thus, x^k is a KKT point of BCOP. ∎

Remark 3.2. Since B_k is uniformly positive definite and uniformly bounded, by Lemma 2.2, the conclusion of Lemma 3.4 can be extended to its limit form:

(i) if FAST=TRUE and $d^{k_l} \to 0$, then any limit point x^* of $\{x^{k_l}\}$ is a KKT point of BCOP, where $\{k_l\}$ is an infinite subsequence of $\{k\}$;

(ii) if FAST=FALSE, $h(x^{k_l}) \to 0$ and $\nabla f(x^{k_l})^T d^{k_l} \to 0$, then any limit point x^* of $\{x^{k_l}\}$ is a KKT point of BCOP, where $\{k_l\}$ is an infinite subsequence of $\{k\}$.

We next establish a series of lemmas concerning the *f-type* iterates.

LEMMA 3.5. *Suppose that assumptions* (A1)–(A2) *hold. Then there exist scalars* $M_h, M_f > 0$ *and* $\alpha_u^k \in (0,1]$ *such that*

$$h(x^k + \alpha d^k) - (1-\alpha)h(x^k) \le \frac{M_h \alpha^2}{2}\|d^k\|^2 \tag{3.7}$$

holds for all $\alpha \in (0, \alpha_u^k]$, *and*

$$\left|f(x^k + \alpha d^k) - f(x^k) - \alpha \nabla f(x^k)^T d^k\right| \le \frac{M_f \alpha^2}{2}\|d^k\|^2 \tag{3.8}$$

holds for all $\alpha \in (0,1]$, *where* d^k *is generated by Algorithm* 3.

Proof. If FAST=TRUE, $(d^k, \lambda^k) = (d^{k,0}, \lambda^{k,0})$ solves (2.26), implying that

$$c_{\mathcal{A}_k}(x^k) + \nabla c_{\mathcal{A}_k}(x^k)^T d^k = 0, \tag{3.9}$$

and if FAST=FALSE, $(d^k, \lambda^k) = (d^{k,2}, \lambda^{k,2})$ solves (2.28), which together with the definition of u_k yields

$$c_i(x^k) + \nabla c_i(x^k)^T d^k = c_i(x^k) + [u_k]_i \begin{cases} = 0, & i \in \mathcal{E}, \\ \le 0, & i \in \mathcal{A}_k \cap \mathcal{I}. \end{cases} \tag{3.10}$$

Since $c_i(x^k)$, $i \in \mathcal{A}_k$ are quadratic functions, it follows that for $i \in \mathcal{A}_k$

$$c_i(x^k + \alpha d^k) = c_i(x^k) + \alpha \nabla c_i(x^k)^T d^k + \frac{\alpha^2}{2}(d^k)^T Q_i d^k,$$

where Q_i is the Hessian of $c_i(x)$. As a result, for either FAST=TRUE or FAST=FALSE, using (3.9) and (3.10) we have

$$c_i(x^k + \alpha d^k) = (1-\alpha)c_i(x^k) + \frac{\alpha^2}{2}(d^k)^T Q_i d^k \quad \forall i \in \mathcal{E},$$
$$c_i(x^k + \alpha d^k) \le (1-\alpha)c_i(x^k) + \frac{\alpha^2}{2}(d^k)^T Q_i d^k \quad \forall i \in \mathcal{A}_k \cap \mathcal{I}.$$

Therefore, it is straightforward to get that for all $i \in \mathcal{E}$

$$|c_i(x^k + \alpha d^k)| \le (1-\alpha)|c_i(x^k)| + \frac{M_h \alpha^2}{2}\|d^k\|^2 \tag{3.11}$$

and for all $i \in \mathcal{A}_k \cap \mathcal{I}$

$$\max\{0, c_i(x^k + \alpha d^k)\} \le (1-\alpha)\max\{0, c_i(x^k)\} + \frac{M_h \alpha^2}{2}\|d^k\|^2, \tag{3.12}$$

where $M_h > 0$ is a scalar satisfying $\|Q_i\| \le M_h$ for all $i \in \mathcal{A}_k$. On the other hand, for $i \in \mathcal{I}\backslash\mathcal{A}_k$, $c_i(x^k) < 0$ due to the definition of $\mathcal{A}_k$; by the continuity of $c_i(x)$, there exists a scalar $\alpha_u^k \in (0,1]$ such that $c_i(x^k + \alpha d^k) < 0$ for all $i \in \mathcal{I}\backslash\mathcal{A}_k$ and all $\alpha \in (0, \alpha_u^k]$. Consequently, in view of the definition of $h(x)$, for all $\alpha \in (0, \alpha_u^k]$, we have

$$h(x^k) = \left\|\begin{pmatrix} c_{\mathcal{E}}(x^k) \\ \max\{c_{\mathcal{I}}(x^k), 0\}\end{pmatrix}\right\|_\infty = \left\|\begin{pmatrix} c_{\mathcal{E}}(x^k) \\ \max\{c_{\mathcal{A}_k \cap \mathcal{I}}(x^k), 0\}\end{pmatrix}\right\|_\infty$$

and

$$h(x^k + \alpha d^k) = \left\|\begin{pmatrix} c_{\mathcal{E}}(x^k + \alpha d^k) \\ \max\{c_{\mathcal{I}}(x^k + \alpha d^k), 0\}\end{pmatrix}\right\|_\infty = \left\|\begin{pmatrix} c_{\mathcal{E}}(x^k + \alpha d^k) \\ \max\{c_{\mathcal{A}_k \cap \mathcal{I}}(x^k + \alpha d^k), 0\}\end{pmatrix}\right\|_\infty,$$

which together with (3.11) and (3.12) gives (3.7).

As for (3.8), it readily follows from Taylor's theorem that

$$f(x^k + \alpha d^k) - f(x^k) - \alpha \nabla f(x^k)^T d^k = \frac{\alpha^2}{2}(d^k)^T \nabla^2 f(\xi^k) d^k, \tag{3.13}$$

where $\xi^k \in \mathbb{R}^n$ lies in the line segment from x^k to $x^k + d^k$. Since x^k and d^k are bounded for all k, and the objective function $f(x)$ is twice continuously differentiable, there exists a scalar $M_f > 0$ such that $\|\nabla^2 f(\xi^k)\| \le M_f$ for all ξ^k, and thus using (3.13) gives (3.8). ∎

We remark that α_u^k in Lemma 3.5 is related to x^k; however, with some additional conditions, α_u^k can be reduced to a constant, which is shown in the following corollary.

COROLLARY 3.6. *Suppose that assumptions* (A1)–(A2) *hold. Let* $\{x^{k_l}\}$ *converge to a nonoptimal point* x^* *and* $\mathcal{A}_{k_l}$ *keeps unchanged for all* k_l. *Then there exist scalars* $M_h > 0$ *and* $\alpha_u \in (0,1]$ *such that*

$$h(x^{k_l} + \alpha d^{k_l}) - (1-\alpha) h(x^{k_l}) \le \frac{M_h \alpha^2}{2} \|d^{k_l}\|^2 \tag{3.14}$$

holds for all $\alpha \in (0, \alpha_u]$, *where* d^{k_l} *is generated by Algorithm* 3.

Proof. According to the hypothesis of this corollary, $\mathcal{A}_{k_l} \equiv \mathcal{A}^*$ for all k_l, where $\mathcal{A}^*$ is a finite index set independent of k_l. Recalling the definition of $\mathcal{A}^*$ (i.e., $\mathcal{A}_{k_l}$) and $x^{k_l} \to x^*$, we obtain that $c_i(x^*) < 0$ for all $i \in \mathcal{I} \backslash \mathcal{A}^*$ and by continuity of $c_i(x)$, there exists an open ball $B(x^*; \bar{r})$ of radius $\bar{r} > 0$ centered at x^* such that for any $y \in B(x^*; \bar{r})$, $c_i(y) < 0$, $i \in \mathcal{I} \backslash \mathcal{A}^*$. Again using $x^{k_l} \to x^*$, and $\|d^{k_l}\| \le M_d$ due to Remark 3.1, there exists a scalar $\bar{\alpha} > 0$ and an integer $k_{\bar{l}} > 0$ such that $c_i(x^{k_l} + \alpha d^{k_l}) < 0$, $i \in \mathcal{I} \backslash \mathcal{A}^*$ for all $\alpha \in (0, \bar{\alpha}]$ and all $k_l \ge k_{\bar{l}}$. Thus for all $\alpha \in (0, \bar{\alpha}]$ and $k_l \ge k_{\bar{l}}$,

$$h(x^{k_l}) = \left\| \begin{pmatrix} c_{\mathcal{E}}(x^{k_l}) \\ \max\{c_{\mathcal{A}^* \cap \mathcal{I}}(x^{k_l}), 0\} \end{pmatrix} \right\|_\infty \quad \text{and}$$

$$h(x^{k_l} + \alpha d^{k_l}) = \left\| \begin{pmatrix} c_{\mathcal{E}}(x^{k_l} + \alpha d^{k_l}) \\ \max\{c_{\mathcal{A}^* \cap \mathcal{I}}(x^{k_l} + \alpha d^{k_l}), 0\} \end{pmatrix} \right\|_\infty .$$

Following the proof of Lemma 3.5, for all $i \in \mathcal{E}$, (3.11) holds with k_l in place of k, and for all $i \in \mathcal{A}^* \cap \mathcal{I}$, (3.12) holds with k_l in place of k, and therefore (3.14) holds for all $\alpha \in (0, \bar{\alpha}]$ and $k_l \ge k_{\bar{l}}$. On the other hand, for those iterations with $k_l < k_{\bar{l}}$, it follows from Lemma 3.5 that (3.14) holds for all $\alpha \in (0, \alpha_u^{k_l}]$. Define $\alpha_u = \min\{\alpha_u^{k_1}, \alpha_u^{k_2}, \ldots, \alpha_u^{k_{\bar{l}}-1}, \bar{\alpha}\}$. We therefore conclude that (3.14) holds for all $\alpha \in (0, \alpha_u]$, which completes the proof. ∎

Define the quantity

$$\Upsilon_k := \begin{cases} \|d^{k,0}\|, & FAST = TRUE, \\ h(x^k) + |\nabla f(x^k)^T d^{k,2}|, & FAST = FALSE, \end{cases}$$

which is actually another first-order optimality measure due to Lemma 3.4. The proofs of the following lemmas and theorem are related to the optimality measure Υ_k. In particular, the next lemma reveals that the search direction d^k generated by Algorithm 3 is descent for the objective function if a point is "nearly" feasible but nonoptimal.

LEMMA 3.7. *Suppose that assumptions* (A1)–(A2) *hold. Let* $\{x^{k_l}\}$ *be a subsequence of* $\{x^k\}$ *for which* $\Upsilon_{k_l} \geq \epsilon$ *with a constant* $\epsilon > 0$*. Then there exist two scalars* $\epsilon_1 > 0$ *and* $\epsilon_2 > 0$ *such that the following statement is true: if*

$$h(x^{k_l}) \leq \epsilon_1, \tag{3.15}$$

then

$$\nabla f(x^{k_l})^T d^{k_l} \leq -\epsilon_2. \tag{3.16}$$

Proof. We first consider the case FAST=TRUE. In this situation, $\Upsilon_{k_l} = \|d^{k_l,0}\| \geq \epsilon$, and $(d^{k_l}, \lambda^{k_l}) = (d^{k_l,0}, \lambda^{k_l,0})$ solves (2.26). Premultiplying the first equation of (2.26) by $(d^{k_l,0})^T$, we have that

$$\nabla f(x^{k_l})^T d^{k_l} = -(d^{k_l,0})^T B_k d^{k_l,0} - (d^{k_l,0})^T \nabla c_{\mathcal{A}_{k_l}}(x^{k_l})\lambda^{k_l,0},$$

while premultiplying the second equation of (2.26) by $(\lambda^{k_l,0})^T$ and substituting it into above equation yield

$$\nabla f(x^{k_l})^T d^{k_l} = -(d^{k_l,0})^T B_k d^{k_l,0} + \sum_{i\in\mathcal{A}_{k_l}} \lambda_i^{k_l,0} c_i(x^{k_l}). \tag{3.17}$$

Due to FAST=TRUE, we have $\lambda^{k_l,0} \geq 0$, and using Remark 3.1 gives $\|\lambda^{k_l,0}\| \leq M_\lambda$. It is straightforward that

$$\sum_{i\in\mathcal{A}_{k_l}} \lambda_i^{k_l,0} c_i(x^{k_l}) \leq \sqrt{m} h(x^{k_l}) \|\lambda^{k_l,0}\| \leq \sqrt{m} M_\lambda h(x^{k_l}),$$

which together with (3.17), assumption (A2), and $\|d^{k_l,0}\| \geq \epsilon$ gives

$$\nabla f(x^{k_l})^T d^{k_l} \leq -\frac{\epsilon^2}{\tau} + \sqrt{m} M_\lambda h(x^{k_l}).$$

Let $\epsilon_1 := \frac{\epsilon^2}{2\sqrt{m}\tau M_\lambda}$. If $h(x^{k_l}) \leq \epsilon_1$, we then get (3.16) with $\epsilon_2 := \frac{\epsilon^2}{2\tau}$.

Next, we show the assertion for the case FAST=FALSE. In this situation, $d^{k_l} = d^{k_l,2}$ and $\Upsilon_{k_l} = h(x^{k_l}) + |\nabla f(x^{k_l})^T d^{k_l,2}| \geq \epsilon$. If $h(x^{k_l}) \leq \frac{\epsilon}{2}$, then

$$|\nabla f(x^{k_l})^T d^{k_l}| = |\nabla f(x^{k_l})^T d^{k_l,2}| \geq \frac{\epsilon}{2}. \tag{3.18}$$

From (3.6) and the definition of u_k,

$$\begin{aligned}\nabla f(x^{k_l})^T d^{k_l,2} = -(d^{k_l,1})^T B_{k_l} d^{k_l,1} - \sum_{\lambda_i^{k_l,1}<0, i\in\mathcal{A}_{k_l}\cap\mathcal{I}} [(\lambda_i^{k_l,1})^2 + \max\{-\lambda_i^{k_l,1} c_i(x^{k_l}), 0\}] \\ + \sum_{\lambda_i^{k_l,1}\geq 0, i\in\mathcal{A}_{k_l}\cap\mathcal{I}} \lambda_i^{k_l,1} c_i(x^{k_l}) + \sum_{i\in\mathcal{E}} \lambda_i^{k_l,1} c_i(x^{k_l}).\end{aligned}$$

By assumption (A2), one has

$$\begin{aligned}\nabla f(x^{k_l})^T d^{k_l,2} &\leq \sum_{\lambda_i^{k_l,1}\geq 0, i\in\mathcal{A}_{k_l}\cap\mathcal{I}} \lambda_i^{k,1} c_i(x^{k_l}) + \sum_{i\in\mathcal{E}} \lambda_i^{k_l,1} c_i(x^{k_l}) \\ &\leq \sqrt{m} h(x^{k_l}) \|\lambda^{k_l,1}\| \\ &\leq \sqrt{m} M_\lambda h(x^{k_l}), \end{aligned} \tag{3.19}$$

where the third inequality follows from Remark 3.1. Let $\epsilon_1 := \min\{\frac{\epsilon}{2}, \frac{\epsilon}{3\sqrt{m}M_\lambda}\}$ and $\epsilon_2 := \frac{\epsilon}{2}$. If (3.15) is true, then $\sqrt{m}M_\lambda h(x^{k_l}) \le \frac{\epsilon}{3}$, which combining with (3.19) and (3.18) yields (3.16). ∎

Lemma 3.8. *Suppose that assumptions* (A1)–(A2) *hold. If* $h(x^{k_l}) > 0$ *and* (3.16) *holds, then* $x^{k_l} + \alpha d^{k_l}$ *is acceptable to the* k_l*th filter for all* $\alpha \le \bar{\alpha}^{k_l}$*, where* $\bar{\alpha}^{k_l} = \min\{q_1 h(x^{k_l}), q_2, \alpha_u^{k_l}\}$, $q_1 = \frac{2}{M_h M_d^2}$*, and* $q_2 = \frac{2\epsilon_2}{M_f M_d^2}$.

Proof. The mechanism of Algorithm 3 (lines 19–23) ensures that $(h(x^{k_l}), f(x^{k_l}))$ is acceptable to the k_lth filter. We now show that $x^{k_l} + \alpha d^{k_l}$ is no worse than x^{k_l} for all sufficiently small $\alpha > 0$ in both feasibility and the objective function, implying that $x^{k_l} + \alpha d^{k_l}$ is acceptable to the k_lth filter. Since $\|d^{k_l}\| \le M_d$ due to Remark 3.1, it follows from (3.7) in Lemma 3.5 that

$$h(x^{k_l} + \alpha d^{k_l}) - h(x^{k_l}) \le -\alpha h(x^{k_l}) + \frac{\alpha^2 M_h M_d^2}{2} \tag{3.20}$$

for $\alpha \in (0, \alpha_u^{k_l}]$, which turns out to be

$$h(x^{k_l} + \alpha d^{k_l}) \le h(x^{k_l})$$

if $0 \le \alpha \le \min\{q_1 h(x^{k_l}), \alpha_u^{k_l}\}$ with $q_1 := \frac{2}{M_h M_d^2}$. Similarly, using (3.8) in Lemma 3.5 and the boundedness of d^{k_l}, we have that

$$f(x^{k_l} + \alpha d^{k_l}) - f(x^{k_l}) \le \alpha \nabla f(x^{k_l})^T d^{k_l} + \frac{\alpha^2 M_f M_d^2}{2},$$

which together with the assumption $\nabla f(x^{k_l})^T d^{k_l} \le -\epsilon_2$ yields

$$f(x^{k_l} + \alpha d^{k_l}) - f(x^{k_l}) \le -\alpha\epsilon_2 + \frac{\alpha^2 M_f M_d^2}{2}.$$

Define $q_2 := \frac{2\epsilon_2}{M_f M_d^2}$. If $0 \le \alpha \le q_2$, then $f(x^{k_l} + \alpha d^{k_l}) \le f(x^{k_l})$. Therefore, $x^{k_l} + \alpha d^{k_l}$ is acceptable to the k_lth filter for all $\alpha \le \bar{\alpha}^{k_l} := \min\{q_1 h(x^{k_l}), \alpha_u^{k_l}, q_2\}$. ∎

With the help of Lemma 3.8, the following two lemmas show that there always exists some acceptable step size α such that $x^k + \alpha d^k$ is accepted as an *f-type* iteration point under certain conditions.

Lemma 3.9. *Suppose that assumptions* (A1)–(A2) *hold. If* x^{k_l} *is feasible but not optimal, then either* $x^{k_l} + d^{k_l} + \hat{d}^{k_l}$ *is an* f-type *iteration point or there exists* $\alpha_0^{k_l} > \alpha_{\min}^{k_l}$ *such that* $x^{k_l} + \alpha_0^{k_l} d^{k_l}$ *is an* f-type *iteration point.*

Proof. The conclusion follows immediately if $x^{k_l} + d^{k_l} + \hat{d}^{k_l}$ is an *f-type* iteration point. Otherwise, we need to prove that $x^{k_l} + \alpha_0^{k_l} d^{k_l}$ is an *f-type* iteration point for some $\alpha_0^{k_l} > \alpha_{\min}^{k_l}$. Since x^{k_l} is feasible but not optimal, we must have that $h(x^{k_l}) = 0$ and $\Upsilon_{k_l} \ge \epsilon$ with some scalar $\epsilon > 0$. Let

$$\pi^{k_l} := \min\{h(x^p) | (h(x^p), f(x^p)) \in \mathcal{F}_{k_l}\}.$$

By the mechanism of Algorithm 3 (line 27) and Lemma 3.7, the condition (2.20) is always true if $h(x^p) = 0$ for the iterate x^p, and therefore only pairs with $h(x^p) > 0$ can possibly be added into the filter. Thus $\pi^{k_l} > 0$. According to Lemma 3.5 and $\|d^{k_l}\| \le M_d$,

$$h(x^{k_l} + \alpha d^{k_l}) \le \frac{\alpha^2 M_h M_d^2}{2} \tag{3.21}$$

holds for all $\alpha \in (0, \alpha_u^{k_l}]$. If $0 \le \alpha \le \min\left\{\alpha_u^{k_l}, \sqrt{\frac{2\beta\pi^{k_l}}{M_h M_d^2}}\right\}$, then $h(x^{k_l} + \alpha d^{k_l}) \le \beta\pi^{k_l}$, which implies that $x^{k_l} + \alpha d^{k_l}$ is acceptable to the k_lth filter. Since x^{k_l} is feasible, it follows from the definition of Ω_p that x^{k_l} is in the interior of Ω_p, which together with the boundedness of d^{k_l} shows $x^{k_l} + \alpha d^{k_l} \in \Omega_p$ for all α in some subinterval of $(0, 1]$, and therefore, we can assume without loss of generality that $x^{k_l} + \alpha d^{k_l} \in \Omega_p$ for all $\alpha \in (0, \alpha_u^{k_l}]$. By Lemma 3.7, $h(x^{k_l}) = 0$ implies (3.16), which means that the switching condition for Case 1 and Case 2 holds trivially no matter $\alpha < 1$ or $\alpha = 1$. It follows from (3.8) in Lemma 3.5 and the boundedness of d^{k_l} that

$$f(x^{k_l} + \alpha d^{k_l}) - f(x^{k_l}) - \alpha\eta\nabla f(x^{k_l})^T d^{k_l} \le -\alpha(1-\eta)\epsilon_2 + \frac{\alpha^2 M_f M_d^2}{2}.$$

Thus, the sufficient reduction condition (2.21) holds if $0 \le \alpha \le \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}$. When $0 \le \alpha \le \min\{\alpha_u^{k_l}, \frac{2\eta\epsilon_2}{\gamma M_h M_d^2}\}$, it is true from (3.21) that

$$h(x^{k_l} + \alpha d^{k_l}) \le \frac{\alpha\eta\epsilon_2}{\gamma}.$$

Combining with (2.21) and (3.16) yields

$$f(x^{k_l} + \alpha d^{k_l}) - f(x^{k_l}) \le -\gamma h(x^{k_l} + \alpha d^{k_l}),$$

i.e., $x^{k_l} + \alpha d^{k_l}$ is acceptable to x^{k_l}. From (2.24) and the above proof, we have $\alpha_{\min}^k = 0$, and we can choose any α in $(0, \bar{\alpha}^{k_l}]$ as $\alpha_0^{k_l}$ such that $x^{k_l} + \alpha_0^{k_l} d^{k_l}$ is an *f-type* iteration point, where

$$\bar{\alpha}^{k_l} := \min\left\{\alpha_u^{k_l}, \sqrt{\frac{2\beta\pi^{k_l}}{M_h M_d^2}}, \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}, \frac{2\eta\epsilon_2}{\gamma M_h M_d^2}\right\}. \qquad \square$$

Lemma 3.10. *Suppose that assumptions* (A1)–(A2) *hold. Let* $\{x^{k_l}\}$ *be an infinite subsequence of* $\{x^k\}$ *on which* $(h(x^{k_l}), f(x^{k_l}))$ *is added into the filter, and assume that* $\{x^{k_l}\}$ *converges to* x^* *and* $\mathcal{A}_{k_l}$ *keeps unchanged for all* k_l. *If* x^* *is not a KKT point, then for all sufficiently large* k_l, *either* $x^{k_l} + d^{k_l} + \hat{d}^{k_l}$ *is an* f-type *iteration point or there exists* $\alpha_0^{k_l} > \alpha_{\min}^{k_l}$ *such that* $x^{k_l} + \alpha_0^{k_l} d^{k_l}$ *is an* f-type *iteration point.*

Proof. If $x^{k_l} + d^{k_l} + \hat{d}^{k_l}$ is an *f-type* iteration point, the conclusion follows immediately. It suffices to prove the assertion for $x^{k_l} + \alpha d^{k_l}$. Since x^* is not a KKT point, it follows from Remark 3.2 that there exists a scalar $\epsilon > 0$ such that $\Upsilon_{k_l} \ge \epsilon$ for all sufficiently large k_l. In the case of $h(x^{k_l}) = 0$, the conclusion follows from Lemma 3.9.

We now consider the remaining iteration k_l with $h(x^{k_l}) > 0$. As $\Upsilon_{k_l} \ge \epsilon$, if $h(x^{k_l}) \le \epsilon_1$, then by Lemma 3.7, (3.16) holds. If $h(x^{k_l}) < \epsilon_1$ and $\alpha \le \min\{q_1 h(x^{k_l}), \alpha_u^{k_l}, q_2\}$, it follows from Lemma 3.8 that $x^{k_l} + \alpha d^{k_l}$ is acceptable to the k_lth filter. Since $\{x^{k_l}\}$ converges to x^* and $\mathcal{A}_{k_l}$ keeps unchanged for all k_l, Corollary 3.6 implies $\alpha_u^{k_l}$ is independent of k_l and we therefore drop the superscript k_l in $\alpha_u^{k_l}$ for simplicity in the following proof. Analogous to the proof of Lemma 3.9, if $0 \le \alpha \le \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}$, the sufficient reduction condition (2.21) is fulfilled. Again using Corollary 3.6 and the boundedness of d^{k_l}, for $0 \le \alpha \le \alpha_u$, (3.20) holds, and therefore $h(x^{k_l} + \alpha d^{k_l}) \le h(x^{k_l})$ if $0 \le \alpha \le \min\{q_1 h(x^{k_l}), \alpha_u\}$, where q_1 is defined as Lemma 3.8. On the other hand, if (2.20) is true, it follows from (2.21) that

$$\begin{aligned} f(x^{k_l}+\alpha d^{k_l})-f(x^{k_l}) &\le \alpha\eta\nabla f(x^{k_l})^T d^{k_l} \\ &\le -\eta\delta h^2(x^{k_l}) \\ &\le -\eta\delta h^2(x^{k_l}+\alpha d^{k_l}) \\ &\le -\gamma h^2(x^{k_l}+\alpha d^{k_l}), \end{aligned}$$

where the last inequality follows from $\delta \ge \gamma/\eta$. Hence, $x^{k_l}+\alpha d^{k_l}$ is acceptable to x^{k_l}. Since $h(x^{k_l}) \to 0$ due to Theorem 3.2, according to the definition of Ω_p, x^{k_l} is in the interior of Ω_p for all sufficiently large k_l, which together with the boundedness of d^{k_l} implies $x^{k_l}+\alpha d^{k_l} \in \Omega_p$ for all α in some subinterval of $(0,1]$ and all sufficiently large k_l, and we assume without loss of generality that $x^{k_l}+\alpha d^{k_l} \in \Omega_p$ for all $\alpha \in (0,\alpha_u]$ for all sufficiently large k_l. Therefore, we have now shown that for all sufficiently large k_l, $x^{k_l}+\alpha d^{k_l}$ is acceptable to x^{k_l} and the k_lth filter, $x^{k_l}+\alpha d^{k_l} \in \Omega_p$, and the sufficient reduction condition (2.21) holds if (2.20) is satisfied, $0 \le \alpha \le \tilde{\alpha}^{k_l}$, and

$$h(x^{k_l}) \le \min\left\{1, \epsilon_1, \frac{rq_1\epsilon_2}{\delta}\right\}, \tag{3.22}$$

where

$$\tilde{\alpha}^{k_l} := \min\left\{q_1 h(x^{k_l}), q_2, \alpha_u, \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}\right\},$$

and r is from line 20 in Algorithm 3. Let $\alpha_0^{k_l}$ denote the first trial step size in the sequence $\{1, r, r^2, \ldots\}$ that satisfies

$$\alpha \le \tilde{\alpha}^{k_l}.$$

In view of Theorem 3.2, $h(x^{k_l})$ tends to zero as $k_l \to \infty$, and therefore $\tilde{\alpha}^{k_l} = q_1 h(x^{k_l})$ and (3.22) is satisfied for all sufficiently large k_l. It is evident that

$$\alpha_0^{k_l} \ge r\tilde{\alpha}^{k_l} = rq_1 h(x^{k_l}) \tag{3.23}$$

for all sufficiently large k_l. Using (3.16) and (3.23) we have

$$-\alpha_0^{k_l}\nabla f(x^{k_l})^T d^{k_l} \ge rq_1\epsilon_2 h(x^{k_l}),$$

which together with (3.22) implies that the switching condition (2.20) for Case 1 is satisfied.

Last, we show $\alpha_0^{k_l} > \alpha_{\min}^{k_l}$. Noting the definition (2.24) of $\alpha_{\min}^{k_l}$, and using (3.16) and Theorem 3.2, we know

$$\alpha_{\min}^{k_l} = \frac{\delta h(x^{k_l})^2}{-\nabla f(x^{k_l})^T d^{k_l}}$$

for all sufficiently large k_l. By (2.20), we have $\alpha_0^{k_l} > \alpha_{\min}^{k_l}$ and overall, $x^{k_l}+\alpha_0^{k_l} d^{k_l}$ is an *f-type* iteration point for all sufficiently large k_l. ∎

Now we are in a position to present the main result of this section, the global convergence of Algorithm 3.

THEOREM 3.11. *Suppose that assumptions* (A1)–(A2) *hold. Let the sequence* $\{x^k\}$ *be generated by Algorithm* 3. *Then one of the following outcomes occurs:*

(i) *a KKT point of BCOP is found at some iteration,*

(ii) *there exists an accumulation point of the sequence* $\{x^k\}$ *that is a KKT point of BCOP.*

Proof. It is sufficient to prove (ii) only. We divide the following proof into two cases.

Case (a) There is a finite number of iterations entering in the filter. Without loss of generality, all iterations are assumed to be *f-type* and therefore the sufficient reduction condition (2.21) for Case 1 or (2.23) for Case 2 holds, implying that $\{f(x^k)\}$ is monotonically decreasing. Since $\{x^k\}$ is bounded, it follows from assumption (A1) that $f(x)$ is bounded and thus the sequence $\{f(x^k)\}$ is convergent. Because all iterations are *f-type*, either (2.20) and (2.21) for Case 1 or (2.22) and (2.23) for Case 2 hold for all k, each to be considered separately.

Subcase (a-1). If there exists an infinite index set $\mathcal{K}$ such that (2.22) and (2.23) for Case 2 hold for all $k \in \mathcal{K}$, this implies that

$$f(x^k) - f(x^{k+1}) \geq \eta \min\{-\nabla f(x^k)^T d^k, \xi \|d^k\|^{\zeta_2}\} \geq 0 \quad \forall k \in \mathcal{K}.$$

Using the fact that $\{f(x^k)\}$ is convergent, and letting $k \in \mathcal{K}$ tend to infinity in both sides, we obtain that

$$\lim_{k\to\infty, k\in\mathcal{K}} \min\{-\nabla f(x^k)^T d^k, \xi \|d^k\|^{\zeta_2}\} = 0.$$

If there exists an infinite index set $\mathcal{G} \subset \mathcal{K}$ such that $\min\{-\nabla f(x^k)^T d^k, \xi\|d^k\|^{\zeta_2}\} = \xi\|d^k\|^{\zeta_2}$ for all $k \in \mathcal{G}$, then we have, by noting FAST=TRUE and $d^k = d^{k,0}$ for $k \in \mathcal{K}$, that $d^{k,0} \to 0$ as $k \to \infty$ and $k \in \mathcal{G}$; this together with Remark 3.2 gives the conclusion (ii). Otherwise, we have that

$$\lim_{k\to\infty, k\in\mathcal{K}} \nabla f(x^k)^T d^k = 0, \tag{3.24}$$

and combining with (2.22), we know that all accumulation points of $\{x^k\}_{\mathcal{K}}$ are feasible. Since FAST=TRUE for all $k \in \mathcal{K}$, $d^k = d^{k,0}$ and $\lambda^{k,0} \geq 0$. Premultiplying the first equation of (2.26) by $(d^{k,0})^T$ gives that

$$\nabla f(x^k)^T d^k = \nabla f(x^k)^T d^{k,0} = -(d^{k,0})^T B_k^{k,0} d^{k,0} - (\lambda^{k,0})^T \nabla c_{\mathcal{A}_k}(x^k)^T d^{k,0}, \tag{3.25}$$

and premultiplying the second equation of (2.26) by $(\lambda^{k,0})^T$ and putting it into (3.25) yield

$$\nabla f(x^k)^T d^k = \nabla f(x^k)^T d^{k,0} = -(d^{k,0})^T B_k^{k,0} d^{k,0} + (\lambda^{k,0})^T c_{\mathcal{A}_k}(x^k). \tag{3.26}$$

By Lemma 3.3 and assumption (A2), $\{x^k\}_{\mathcal{K}}$, $\{d^{k,0}\}_{\mathcal{K}}$, $\{\lambda^{k,0}\}_{\mathcal{K}}$, and $\{B_k\}_{\mathcal{K}}$ each have convergent subsequences, and we assume without loss of generality that x^*, d^*, λ^*, and B^* are the limits of $\{x^k\}_{\mathcal{K}}$, $\{d^{k,0}\}_{\mathcal{K}}$, $\{\lambda^{k,0}\}_{\mathcal{K}}$, and $\{B_k\}_{\mathcal{K}}$, respectively. The fact $\lambda^{k,0} \geq 0$ and assumption (A2) ensure that λ^* is nonnegative and B^* is positive definite. Since there are only finitely many choices for the subsets $\mathcal{A}_k \subseteq \mathcal{E} \cup \mathcal{I}$, we can assume also that $\mathcal{A}_k \equiv \mathcal{A}^*$, $k \in \mathcal{K}$, where $\mathcal{A}^*$ is a constant set. Letting k tend to infinity and using (3.24) and (3.26), we obtain

$$(d^*)^T B^* d^* = (\lambda^*)^T c_{\mathcal{A}^*}(x^*),$$

whose right-hand side is nonpositive because of $\lambda^* \geq 0$ and the feasibility of x^*. Thus, $(d^*)^T B^* d^* \leq 0$, which together with the positive definiteness of B^* leads to $d^* = 0$, and from Remark 3.2, we know x^* is a KKT point.

Subcase (a-2). Since (2.20) and (2.21) for Case 1 hold for all k, we obtain that $f(x^k) - f(x^{k+1}) \geq -\alpha^k \eta \nabla f(x^k)^T d^k \geq \eta\delta h(x^k)^2$, which using the convergence of

$\{f(x^k)\}$ gives $h(x^k) \to 0$ as $k \to \infty$. Thus, all accumulation points of $\{x^k\}$ are feasible. Let x^* be any accumulation point of $\{x^k\}$ and we assume, without loss of generality, that $x^k \to x^*$, $k \in \mathcal{K}$. The proof is complete if x^* is a KKT point of BCOP; otherwise, similar to the earlier proof, we can assume that $\mathcal{A}_k \equiv \mathcal{A}^*$, $k \in \mathcal{K}$, is a constant set. We next show that α^k generated from Algorithm 3 is larger than some positive scalar for all $k \in \mathcal{K}$. Since x^* is not a KKT point, by Remark 3.2, there exists a scalar $\epsilon > 0$ such that $\Upsilon_k \geq \epsilon$ for all sufficiently large $k \in \mathcal{K}$. Lemma 3.7 shows that if $h(x^k) \leq \epsilon_1$, then $\nabla f(x^k)^T d^k \leq -\epsilon_2$, and combining with Lemma 3.5 and $\|d^k\| \leq M_d$, we know that the sufficient reduction condition

$$f(x^k + \alpha d^k) - f(x^k) \leq \alpha\eta\nabla f(x^k)^T d^k \tag{3.27}$$

holds if $0 \leq \alpha \leq \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}$. By Corollary 3.6 and $\|d^k\| \leq M_d$, it follows that for $0 \leq \alpha \leq \alpha_u$ and $k \in \mathcal{K}$, (3.20) holds with k in place of k_l. If $h(x^k) \leq \frac{\alpha\eta\epsilon_2}{2\gamma}$ and $0 \leq \alpha \leq \min\{\alpha_u, \frac{\eta\epsilon_2}{\gamma M_h M_d^2}\}$, then

$$h(x^k + \alpha d^k) \leq \frac{\alpha\eta\epsilon_2}{\gamma}. \tag{3.28}$$

Substituting $\nabla f(x^k)^T d^k \leq -\epsilon_2$ into (3.27) and using (3.28) yield

$$f(x^k + \alpha d^k) - f(x^k) \leq -\gamma h(x^k + \alpha d^k),$$

which means that $x^k + \alpha d^k$ is acceptable to x^k. Similar to the proof of Lemma 3.10, using $h(x^k) \to 0$, $k \in \mathcal{K}$, we know $x^k + \alpha d^k \in \Omega_p$ for all α in some subinterval of $(0, 1]$ and all sufficiently large $k \in \mathcal{K}$. Without loss of generality, we assume that $x^k + \alpha d^k \in \Omega_p$ for all $\alpha \in (0, \alpha_u]$ and for all sufficiently large $k \in \mathcal{K}$. Since all iterations are assumed to be *f-type*, the filter includes the only pair $(\chi, -\infty)$. If $h(x^k) \leq \beta\chi$ and $\alpha \leq \min\{\frac{2\beta\chi}{M_h M_d^2}, \alpha_u\}$, (3.20) with k in place of k_l gives

$$\begin{aligned} h(x^k + \alpha d^k) &\leq (1-\alpha)\beta\chi + \alpha\beta\chi \\ &\leq \beta\chi \quad \forall k \in \mathcal{K}, \end{aligned}$$

which implies that for $k \in \mathcal{K}$, $x^k + \alpha d^k$ is acceptable to the kth filter. Therefore, we have shown that if

$$h(x^k) \leq \min\left\{\epsilon_1, \beta\chi, \frac{\alpha\eta\epsilon_2}{2\gamma}\right\} \tag{3.29}$$

and

$$0 \leq \alpha \leq \bar{\alpha} := \min\left\{\alpha_u, \frac{2(1-\eta)\epsilon_2}{M_f M_d^2}, \frac{\eta\epsilon_2}{\gamma M_h M_d^2}, \frac{2\beta\chi}{M_h M_d^2}\right\},$$

then, for all sufficiently large $k \in \mathcal{K}$, $x^k + \alpha d^k$ is acceptable to x^k and the kth filter, $x^k + \alpha d^k \in \Omega_p$, and the sufficient reduction condition (3.27) holds. Since $h(x^k) \to 0$ and $\bar{\alpha} > 0$, the condition (3.29) with $\alpha = r\bar{\alpha}$ is satisfied for all sufficiently large $k \in \mathcal{K}$, where r is from line 20 in Algorithm 3. By Algorithm 3, we know that $\alpha^k \geq r\bar{\alpha}$ for all sufficiently large $k \in \mathcal{K}$.

Since $\{f(x^k)\}$ is convergent, it follows from (3.27), (2.20), and $\alpha^k \geq r\bar{\alpha}$ that (3.24) is true. If FAST=TRUE for all $k \in \mathcal{K}$, our previous argument has shown that x^* is a KKT point and results in a contradiction with the previous assumption. Otherwise, there exists an infinite subset $\mathcal{K}_0$ of $\mathcal{K}$, say, $\mathcal{K}$ itself, such that FAST=FALSE for all $k \in \mathcal{K}_0$. By Algorithm 3, $d^k = d^{k,2}$ for $k \in \mathcal{K}$. As $h(x^k) \to 0$, it follows from (3.24) that

$$\lim_{k\to\infty, k\in\mathcal{K}} \Upsilon_k = \lim_{k\to\infty, k\in\mathcal{K}} \left(h(x^k) + |\nabla f(x^k)^T d^{k,2}|\right) = 0.$$

In view of Remark 3.2, x^* is a KKT point, which again is a contradiction. Hence, we have proven that the second conclusion of this theorem is true for Case (a).

Case (b) There are infinitely many iterations entering in the filter. Let $\mathcal{K}$ be an infinite index set such that all pairs $(h(x^k), f(x^k))$ with $k \in \mathcal{K}$ are added into the filter (though some of them are removed later). Without loss of generality, we assume that $\{x^k\}_{\mathcal{K}} \to x^*$ and $\{\Upsilon_k\}_{\mathcal{K}} \to \bar{\Upsilon}$. If $\bar{\Upsilon} = 0$, it follows from Remark 3.2 that x^* is also a KKT point of BCOP; otherwise, there must exist a scalar $\epsilon > 0$ such that $\Upsilon_k \geq \epsilon$ for all $k \in \mathcal{K}$. Since there are only finitely many choices for the subsets $\mathcal{A}_k \subseteq \mathcal{I}$, $k \in \mathcal{K}$, we can assume that $\{x^k\}_{\mathcal{K}} \to x^*$ and $\mathcal{A}_k \equiv \mathcal{A}^*$ for all $k \in \mathcal{K}$ is a constant set. Upon using Lemma 3.10, we obtain that for sufficiently large $k \in \mathcal{K}$, either $x^k + d^k + \hat{d}^k$ or $x^k + \alpha_0^k d^k$ is accepted as an *f-type iterate* which together with the mechanism of Algorithm 3 implies that $(h(x^k), f(x^k))$ cannot be added into the filter. This contradicts the definition of the sequence $\{x^k\}_{k\in\mathcal{K}}$, indicating that the assumption $\bar{\Upsilon} > 0$ is not true. Therefore, there exists an accumulation point of the sequence $\{x^k\}$ being a KKT point of BCOP and we complete the proof for Case (b). ∎

4. Local convergence. In this section, we prove the locally superlinear convergence of Algorithm 3. Theorem 3.11 has already shown that there exists a subsequence $\{x^k\}_{k\in\mathcal{K}}$ of $\{x^k\}$ converging to a KKT point x^*; let λ^* be a corresponding Lagrange multiplier. The local convergence is established upon additional assumptions:

(A3) The strict complementarity condition holds at (x^*, λ^*), that is, $\lambda_i^* > c_i(x^*)$ for all $i \in \mathcal{I}$.

(A4) The SOSC holds at (x^*, λ^*); that is, there exists a scalar $\bar{\nu} > 0$ such that

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*) d \geq \bar{\nu} \|d\|^2$$

for any $d \in \mathcal{C}(x^*)$, where $\mathcal{C}(x^*) = \{d | \nabla c_i(x^*)^T d = 0,\ i \in \mathcal{E} \cup I(x^*)\}$ is the critical cone at x^*;

(A5)

$$\lim_{k\to+\infty} \frac{\|P_k(B_k - \nabla_{xx}^2 L(x^*, \lambda^*))d^k\|}{\|d^k\|} = 0,$$

where $P_k = I - \nabla c_{\mathcal{A}_k}(x^k)(\nabla c_{\mathcal{A}_k}(x^k)^T \nabla c_{\mathcal{A}_k}(x^k))^{-1} \nabla c_{\mathcal{A}_k}(x^k)^T$ is the projection onto the null space of $\nabla c_{\mathcal{A}_k}(x^k)^T$.

We remark that assumptions (A1)–(A5) are standard for SQP algorithms (see, i.e., [29, Chapter 18]). Since x^* is feasible, similar to the proof of Lemma 2.1, we know that $\nabla c_i(x^*),\ i \in \mathcal{E} \cup I(x^*)$, is linearly independent, which implies that the LICQ condition holds at x^*. Hence, λ^* is the unique multiplier corresponding to x^*. By [29, Theorem 12.6], the LICQ and the SOSC (i.e., assumption **(A4)**) imply x^* is a strict local solution of BCOP, and from [35, Lemma 4.2], the whole sequence (x^k, λ^k) converges to (x^*, λ^*). Furthermore, assumption **(A3)** and the LICQ ensure that $\mathcal{A}_k = \mathcal{E} \cup I(x^*)$ and the condition (2.25) is satisfied for all sufficiently large k, which implies that FAST=FALSE never occurs after some iterations. Therefore, for all sufficiently large k, $(d^k, \lambda^k) = (d^{k,0}, \lambda^{k,0})$, and $(d^{k,1}, \lambda^{k,1})$ solves the linear system (2.27). Due to (2.26), assumption **(A2)** and $x^k \to x^*$, we have $d^k \to 0$. Moreover, from the definition of Ω_p, x^k is in the interior of Ω_p when k is sufficiently large. Based on these facts, in what follows, we assume k is sufficiently large so that all above conclusions hold.

First, we show that the full step ensures the superlinear convergence.

LEMMA 4.1. *Suppose that assumptions* (A1)–(A5) *hold. Then it follows that*

$$\|x^k + d^k - x^*\| = o(\|x^k - x^*\|)$$

and

$$\|d^k\| = \Theta(\|x^k - x^*\|).$$

Proof. See the proof of [35, Lemma 4.3]. ∎

The next lemma reveals the relationship between d^k and $\hat{d}^k$.

LEMMA 4.2. *Suppose that assumptions* (A1)–(A5) *hold. Then* $\|\hat{d}^k\| = \mathcal{O}(\|d^k\|^2)$.

Proof. From (2.26) and (2.27), we have that

$$\begin{cases} B_k\hat{d}^k + \nabla c_{\mathcal{A}_k}(x^k)(\lambda^{k,1} - \lambda^k) = 0, \\ \nabla c_{\mathcal{A}_k}(x^k)^T\hat{d}^k = -c_{\mathcal{A}_k}(x^k + d^k) - \|d^k\|^\nu e. \end{cases} \tag{4.1}$$

By Taylor's theorem and the linear system (2.26),

$$c_i(x^k + d^k) = c_i(x^k) + \nabla c_i(x^k)^T d^k + \mathcal{O}(\|d^k\|^2) = \mathcal{O}(\|d^k\|^2),\ i \in \mathcal{A}_k. \tag{4.2}$$

Since $\mathcal{A}_k$ is fixed for all k, it follows from Lemma 2.2 and assumption **(A2)** that the inverse of the coefficient matrix of (4.1) is uniformly bounded, which together with (4.1), $\nu \in (2, 3)$, and (4.2) leads to the desired result. ∎

LEMMA 4.3. *Suppose that assumptions* (A1)–(A5) *hold. Then*

$$c_i(x^k + d^k + \hat{d}^k) = o(\|d^k\|^2),\ i \in \mathcal{E} \cup I(x^*).$$

Proof. From Taylor's theorem,

$$\begin{aligned} c_i(x^k + d^k + \hat{d}^k) &= c_i(x^k + d^k) + \nabla c_i(x^k)^T\hat{d}^k + \mathcal{O}(\|d^k\|\|\hat{d}^k\|) + \mathcal{O}(\|\hat{d}^k\|^2) \\ &= \mathcal{O}(\|d^k\|\|\hat{d}^k\|) + \mathcal{O}(\|\hat{d}^k\|^2) \\ &= o(\|d^k\|^2), \end{aligned}$$

where the second equality follows from (2.26) and (2.27), and the third equality follows from Lemma 4.2. ∎

To prove the local convergence of Algorithm 3, in the next lemmas we make use of two conclusions in [10], which is concerned with the second-order correction steps on the exact penalty function

$$\Phi_\psi(x) = f(x) + \psi h(x), \tag{4.3}$$

where the penalty parameter ψ is chosen to be no less than $m\|\lambda^*\|_\infty$. The introduction of the exact penalty function (4.3) is only for a technical proof but is not involved in Algorithm 3.

LEMMA 4.4. *Suppose that assumptions* (A1)–(A5) *hold. Then there exists an integer* $K_1 > 0$ *such that if* (2.22) *for Case* 2 *is satisfied for all* $k \geq K_1$*, the sufficient reduction condition* (2.23) *holds.*

Proof. We only have to prove that

$$f(x^k + d^k + \hat{d}^k) + \eta\xi\|d^k\|^{\zeta_2} \le f(x^k) \tag{4.4}$$

holds for all sufficiently large k whenever the switching condition (2.22) for Case 2 is fulfilled. By the definition of $L(x, \lambda)$,

$$\begin{aligned} f(x^k+d^k+\hat{d}^k) &= L(x^k + d^k + \hat{d}^k, \lambda^*) - L(x^*, \lambda^*) + f(x^*) - \sum_{i\in\mathcal{E}\cup I(x^*)} \lambda_i^* c_i(x^k + d^k + \hat{d}^k) \\ &= f(x^*) - \sum_{i\in\mathcal{E}\cup I(x^*)} \lambda_i^* c_i(x^k + d^k + \hat{d}^k) + \mathcal{O}(\|x^k + d^k + \hat{d}^k - x^*\|^2) \\ &= f(x^*) + o(\|d^k\|^2) + \mathcal{O}(\|x^k + d^k + \hat{d}^k - x^*\|^2) \\ &= f(x^*) + o(\|x^k - x^*\|^2), \end{aligned} \tag{4.5}$$

where the second equality follows from the KKT conditions of BCOP and Taylor's theorem, the third equality follows from Lemma 4.3, and the fourth equality follows from Lemmas 4.1 and 4.2. Hence, combining with Lemma 4.1 and (2.22) yields

$$f(x^k + d^k + \hat{d}^k) + \eta\xi\|d^k\|^{\zeta_2} + \psi h(x^k) = f(x^*) + o(\|x^k - x^*\|^2). \tag{4.6}$$

On the other hand, from [6, Lemma 1] and assumptions (A1)–(A5), we know that there exists a scalar $\bar{c} > 0$ such that when x is sufficiently close to x^*,

$$\Phi_\psi(x) \ge f(x^*) + \bar{c}\|x - x^*\|^2. \tag{4.7}$$

Hence, it follows from (4.6) and (4.7) that there exists an integer $K_1 > 0$ such that (4.4) holds for all $k \ge K_1$. ∎

The following two lemmas give preparations for proving acceptance of the full steps.

LEMMA 4.5. *Suppose that assumptions* (A1)–(A5) *hold. Then there exists an integer $K_2 \ge K_1$ such that for all $k \ge K_2$*

$$\Phi_\psi(x^k) - \Phi_\psi\left(x^k + d^k + \hat{d}^k\right) \ge \left(\gamma + \left(\frac{1}{\beta} - 1\right)\psi\right) h\left(x^k + d^k + \hat{d}^k\right). \tag{4.8}$$

Proof. From (4.5),

$$\begin{aligned} &\Phi_\psi(x^k + d^k + \hat{d}^k) + \left(\gamma + \left(\frac{1}{\beta} - 1\right)\psi\right) h(x^k + d^k + \hat{d}^k) \\ &= f(x^k + d^k + \hat{d}^k) + \left(\gamma + \frac{\psi}{\beta}\right) h(x^k + d^k + \hat{d}^k) \\ &= f(x^*) + \left(\gamma + \frac{\psi}{\beta}\right) h(x^k + d^k + \hat{d}^k) + o(\|x^k - x^*\|^2) \\ &= f(x^*) + o(\|x^k - x^*\|^2), \end{aligned}$$

where the last equality follows from Lemmas 4.1, 4.2, and 4.3. This together with (4.7) yields (4.8). ∎

LEMMA 4.6. *For any two points x and x^l, if*

$$\Phi_\psi(x^l) - \Phi_\psi(x) \ge \left(\gamma + \left(\frac{1}{\beta} - 1\right)\psi\right) h(x),$$

then x is acceptable to x^l.

Proof. See the proof of [35, Lemma 4.7]. ∎

Since $x^k \to x^*$, $d^k \to 0$, and $\hat{d}^k \to 0$, it follows from the definition of Ω_p that $x^k + d^k + \hat{d}^k$ is in the interior of Ω for all sufficiently large k, and therefore we assume from now on that $x^k + d^k + \hat{d}^k$ is contained in the interior of Ω. Next, we show that the full step is accepted eventually.

LEMMA 4.7. *Suppose that assumptions* (A1)–(A5) *hold. Then there exists an integer $K_3 \geq K_2$ such that for all $k \geq K_3$, $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted.*

Proof. Let

$$\Gamma_\psi^{K_2} = \min_{l \in \bar{\mathcal{F}}_{K_2} \cup \{K_2\}} \{f(x^l) + \psi h(x^l)\},$$

where $\bar{\mathcal{F}}_k$ is defined in Definition 2.2. Since all iterates x^l, $l \in \bar{\mathcal{F}}_{K_2} \cup \{K_2\}$ are nonoptimal, then $\Gamma_\psi^{K_2} > f(x^*)$, and therefore there exists an $K_3 > K_2$ such that

$$\Phi_\psi(x^{K_3}) < \Gamma_\psi^{K_2}. \tag{4.9}$$

If all iterates after K_2 are never included into the filter, then $\mathcal{F}_k \equiv \mathcal{F}_{K_2}$ for all $k \geq K_2$. Due to Lemma 4.5, (4.8) is satisfied for all $k \geq K_3$, and therefore $\Phi_\psi(x^k) < \Gamma_\psi^{K_2}$ for all $k \geq K_3$. Applying Lemma 4.6, $x^k + d^k + \hat{d}^k$ is acceptable to x^k and $\mathcal{F}_k$ for all $k \geq K_3$, which together with Lemma 4.4 implies the desired conclusion.

If there exist infinite many iterations entering in the filter, we assume without loss of generality that K_3 is the first iteration $K_3 > K_2$ such that $(h(x^{K_3}), f(x^{K_3}))$ is added into the filter and (4.9) is satisfied. By the mechanism of Algorithm 3, $\mathcal{F}_{K_3} \subset \mathcal{F}_{K_2} \cup \{(h(x^{K_2}), f(x^{K_2}))\}$.

First, we show that $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted for $k = K_3$. In view of Lemma 4.5, the inequality (4.8) is satisfied for $k = K_3$, which together with Lemma 4.6 implies that $x^{K_3} + d^{K_3} + \hat{d}^{K_3}$ is acceptable to x^{K_3}. According to (4.9), the definition of $\Gamma_\psi^{K_2}$, and the choice of K_3, we have that $\Phi_\psi(x^{K_3}) < \Gamma_\psi^{K_2} \leq \Phi_\psi(x^l)$ for all $l \in \bar{\mathcal{F}}_{K_3}$. Applying Lemma 4.6 one gets that $x^{K_3} + d^{K_3} + \hat{d}^{K_3}$ is acceptable to $\mathcal{F}_{K_3}$. Therefore, $x^{K_3+1} = x^{K_3} + d^{K_3} + \hat{d}^{K_3}$ is accepted as an *h-type* iteration since $(h(x^{K_3}), f(x^{K_3}))$ is added into the filter.

Next, suppose that $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted for $k = K_3, K_3+1, \ldots, K_3+j-1$ for some $j > 0$. By induction, we attempt to prove that $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted for $k = K_3 + j$. To this end, from Lemma 4.5, we have that

$$\Phi_\psi(x^k) - \Phi_\psi(x^{K_3+j} + d^{K_3+j} + \hat{d}^{K_3+j}) \geq \left(\gamma + \left(\frac{1}{\beta} - 1\right)\psi\right) h(x^{K_3+j} + d^{K_3+j} + \hat{d}^{K_3+j})$$

for all $k = K_3, K_3 + 1, \ldots, K_3 + j$ and all $k \in \bar{\mathcal{F}}_{K_3}$, and applying Lemma 4.6 yields that $x^{K_3+j} + d^{K_3+j} + \hat{d}^{K_3+j}$ is acceptable to x^k with all $k \in \bar{\mathcal{F}}_{K_3} \cup \{K_3, K_3 + 1, \ldots, K_3 + j\}$, which implies that $x^{K_3+j} + d^{K_3+j} + \hat{d}^{K_3+j}$ is acceptable to $\mathcal{F}_{K_3+j} \cup \{(h(x^{K_3+j}), f(x^{K_3+j}))\}$. Moreover, at the iteration $K_3 + j$, if the switching condition (2.22) for Case 2 is met, it follows from Lemma 4.4 that the sufficient reduction condition (2.23) holds, and therefore an *f-type* iteration $x^{k+1} = x^k + d^k + \hat{d}^k$ with $k = K_3 + j$ is generated; otherwise, an *h-type* iteration $x^{k+1} = x^k + d^k + \hat{d}^k$ with $k = K_3 + j$ is generated. Thus, we have proved that $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted for $k = K_3 + j$, and by induction, we assert that $x^{k+1} = x^k + d^k + \hat{d}^k$ is accepted for all $k \geq K_3$. ∎

Consequently, we can state the main result of this section whose proof follows directly from Lemmas 4.1 and 4.7.

THEOREM 4.8. *Suppose that assumptions* (A1)–(A5) *hold. Then* $\{x^k\}$ *converges to* x^* *superlinearly.*

5. Numerical experiment. In this section, we test FilterASM on two specific practical applications: the correlation matrix approximation problem [4, 25] and the maximal correlation problem [45], upon MATLAB 7.10 on a PC with Intel Core i5-2320 CPU (3.0 GHz) and 4 GB memory. For the stopping criteria, we terminate Algorithm 3 whenever

$$h(x^k) \le \varepsilon, \ \left\|\min\{\lambda^k, -c(x^k)\}\right\|_\infty \le \varepsilon \ \text{ and } \ \left\|\nabla f(x^k) + \nabla c(x^k)\lambda^k\right\|_\infty \le \varepsilon, \quad \text{where } \varepsilon = 10^{-6};$$

other parameters in FilterASM are set as follows:

$$\chi = \max\{10, h(x^0)\},\ \mathit{maxit}{=}2000,\ \ \zeta_2 = \nu = 2.5,\ \delta = \xi = \zeta_1 = 1,\ r = 0.5,\ \eta = 10^{-2},\ \beta = \gamma = 10^{-4},\ \alpha_\phi = 10^{-8}.$$

5.1. Approximation problem of correlation matrix with factor structure. We first apply FilterASM to solve the problem of the NCM with p factor structure:

$$\begin{cases} \min\limits_{X\in\mathbb{R}^{m\times p}} & \|G - (I + XX^T - \mathrm{Diag}(XX^T))\|_F^2 \\ \text{s.t.} & \mathrm{diag}(XX^T) \le 1, \end{cases} \tag{5.1}$$

where G is a given real symmetric m-by-m matrix. The structure in (5.1) mainly arises in factor models of asset returns [11], collateralized debt obligations [2, 12], and multivariate time series [28]. The problem (5.1) is posed in the context of credit basket securities by Anderson, Sidenius, and Basu [2]; recently, Borsdorf, Higham, and Raydan [4] analyzed the properties of the problem and its data matrices in some special case, and they also proposed some numerical algorithms. Later, Li, Qi, and Xiu [25] proposed two numerical methods (the alternating block relaxation method and the alternating majorization method) for this problem. More details and references about this problem can be found in [4]. To apply FilterASM, we reformulate (5.1) as

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & f(x) := \|G - (I + XX^T - \mathrm{Diag}(XX^T)\|_F^2 \\ \text{s.t.} & c_i(x) := \|X_i\|^2 - 1 \le 0,\ i = 1, 2, \dots, m, \end{cases}$$

where

$$x = \begin{pmatrix} X_1^T \\ \vdots \\ X_m^T \end{pmatrix},\ X = \begin{pmatrix} X_1 \\ \vdots \\ X_m \end{pmatrix}$$

and $n = mp$.

Test problems.

E1. G is a random correlation matrix generated by `gallery('randcorr',m)`.

E2. G occurring in annual forward rate correlations associated with LIBOR models [1] is generated by $G_{ij} = \exp(-|i-j|), i, j = 1, \dots, m$.

E3. G is a random correlation matrix generated by $G = \mathrm{Diag}(I - YY^T) + YY^T$, where $Y \in \mathbb{R}^{m\times p}$ is generated in a two-stage scheme: we first generate a random matrix with elements from the uniform distribution on $[-1, 1]$ and then project it onto $\{Y \in \mathbb{R}^{m\times p} | \|Y_i\|^2 \le 1, i = 1, 2, \ldots, m\}$ to get Y.

E4. G is a random correlation matrix generated by $G = \frac{1}{2}(B+B^T)+\mathrm{Diag}(I-B)$, where B is the first matrix out of a sequence of random matrices with elements from the uniform distribution on $[-1, 1]$ such that G has a negative eigenvalue.

E5. G is the 387×387 one-day correlation matrix (as of October 10, 2008) from the lagged datasets of RiskMetrics.

5.1.1. Numerical results and comparison with other solvers. To verify the efficiency of FilterASM, in this subsection we first compare its numerical performance of FilterASM with three other methods:

- SPGM: the spectral projected gradient method [4];
- BRscg: the block relaxation method [25]; and
- Major: the majorization method [25];

The stopping criteria for SPGM and BRscg are

$$\|P_\Omega(x^k - \nabla f(x^k)) - x^k\| \le 10^{-6} \quad \text{and} \quad \frac{f(x^k) - f(x^{k+1})}{f(x^k)} \le 10^{-4},$$

respectively, where $P_\Omega(x)$ denotes the projection of x onto the feasible region Ω. Major has the same stopping criterion with BRscg. In [4, 25], the NCM strategy is shown to produce effective starting points for accelerating their algorithms, where NCM employes the semismooth Newton method [33] to generate starting points. In our numerical experiment on E1–E5 in this subsection, we adopt the NCM strategy to produce starting points for all solvers. In Tables 1–6, we report results averaged over two instances of each problem since some problems (say, E1, E3, and E4) are related to random matrices. To understand these numerical results, we point out that f^*, time(s), iter, and $*$ stand for the computed optimal solution, the computational time (seconds), the number of (outer) iterations, and the failure of a solver in finding a solution within 1800 seconds, respectively.

We now make several comments on the results in Tables 1–6.

- Tables 1–3 give the numerical results of test problems E1–E4 with $m = 1000$ and p varying from 5 to 500. The test problem E5 is a correlation matrix from the real market, and its numerical results are listed in Tables 4–6.
- From these tables, we observed that SPGM needs more iterations than FilterASM, Major, and BRscg. Nevertheless, the number of outer iterations alone is not sufficient for measuring the performance of the algorithm as different solvers need different computational effort for each iteration. In particular, Major and BRscg require many inner iterations at each outer iteration, especially for large p.
- The CPU time is then another factor for the efficiency of the algorithm. We observe that FilterASM is the clear winner among these solvers in terms of CPU time. Except for SPGM and Major, all other algorithms solve all instances within 1800 seconds (see Table 3), but the accuracy of some solutions is not satisfactory. For problems E1, E2, E4, and E5, the CPU time used in each solver increases as p does, but the CPU time required by FilterASM increases much more slowly than others.
- The problem E3 is an exception as the objective function is nearly zero at the global solutions. The initial points generated by the semismooth Newton

Table 1
Results for approximation problem of correlation matrix.

Prob.	$m = 1000$, $p = 5$				$m = 1000$, $p = 10$			
	Algorithm	f^*	time(s)	iter	Algorithm	f^*	time(s)	iter
E1	FilterASM	1.80132e+1	3.3	127	FilterASM	1.77820e+1	7.5	255
	SPGM	1.80132e+1	7.2	216	SPGM	1.77820e+1	21.0	602
	Major	1.80134e+1	6.3	3	Major	1.77824e+1	8.5	3
	BRscg	1.80135e+1	6.8	3	BRscg	1.77825e+1	9.1	3
E2	FilterASM	1.74848e+1	2.7	85	FilterASM	1.72848e+1	2.7	79
	SPGM	1.74848e+1	3.7	106	SPGM	1.72848e+1	3.9	110
	Major	1.74889e+1	4.2	2	Major	1.72915e+1	8.5	3
	BRscg	1.74889e+1	4.6	2	BRscg	1.72915e+1	9.0	2
E3	FilterASM	7.97512e-5	5.3	30	FilterASM	2.95352e-5	2.9	11
	SPGM	1.15455e-8	0.7	14	SPGM	2.64062e-9	0.6	7
	Major	5.32996e-5	9.1	5	Major	3.25934e-5	7.5	2
	BRscg	4.37226e-2	9.8	4	BRscg	1.42947e-3	7.7	1
E4	FilterASM	4.04293e+2	2.9	107	FilterASM	4.00305e+2	4.1	118
	SPGM	4.04293e+2	9.9	304	SPGM	4.00305e+2	10.5	312
	Major	4.04300e+2	6.3	3	Major	4.00320e+2	8.5	3
	BRscg	4.04300e+2	6.9	3	BRscg	4.00321e+2	9.2	3

Table 2
Results for approximation problem of correlation matrix.

Prob.	$m = 1000$, $p = 50$				$m = 1000$, $p = 100$			
	Algorithm	f^*	time(s)	iter	Algorithm	f^*	time(s)	iter
E1	FilterASM	1.66560e+1	7.2	110	FilterASM	1.53840e+1	19.7	183
	SPGM	1.66560e+1	11.6	233	SPGM	1.53840e+1	51.5	567
	Major	1.66570e+1	40.8	4	Major	1.53855e+1	91.0	5
	BRscg	1.66572e+1	42.2	4	BRscg	1.53855e+1	94.1	5
E2	FilterASM	1.55955e+1	28.0	372	FilterASM	1.33235e+1	19.2	146
	SPGM	1.55955e+1	219.2	3781	SPGM	1.33235e+1	47.8	566
	Major	1.55992e+1	51.3	3	Major	1.33244e+1	113.5	3
	BRscg	1.55991e+1	52.4	3	BRscg	1.33244e+1	115.4	3
E3	FilterASM	2.82704e-13	0.1	0	FilterASM	2.45434e-13	0.1	0
	SPGM	5.68309e-13	1.0	1	SPGM	5.75110e-13	1.7	1
	Major	2.82704e-13	0.0	0	Major	2.45434e-13	0.0	0
	BRscg	2.82704e-13	0.0	0	BRscg	2.45434e-13	0.0	0
E4	FilterASM	3.74448e+2	72.1	195	FilterASM	3.65310e+2	76.8	165
	SPGM	3.74448e+2	233.3	3257	SPGM	3.65310e+2	328.7	3941
	Major	3.74477e+2	28.7	3	Major	3.65313e+2	39.1	3
	BRscg	3.76348e+2	29.6	3	BRscg	3.67540e+2	40.1	2

method [33] are almost the global solutions for $p = 50,\ 100,\ 250$, and 500, and we observed from Table 6 that nearly all solvers terminate at the initial points.

- From all these tables, we observed that SPGM and FilterASM are better than Major and BRscg in terms of the quality of solutions. Overall, FilterASM obtained satisfactory solutions using the least computational time.

5.1.2. Further evaluation of the behavior of FilterASM. To evaluate the performance of the active-set strategy in FilterASM, we next conduct more experiments on test problems E1–E5 with $m = 400$ ($m = 387$ for E5) and p varying from 100 to 350. Unlike the previous experiments, we use `rand` to generate the random vector, project it onto the feasible region, and then take the projection vector as the starting point for each test problem. The detailed numerical results are summarized

Table 3
Results for approximation problem of correlation matrix.

Prob.	$m = 1000$, $p = 250$				$m = 1000$, $p = 500$			
	Algorithm	f^*	time(s)	iter	Algorithm	f^*	time(s)	iter
E1	FilterASM	1.19373e+1	33.4	127	FilterASM	6.32102	96.9	174
	SPGM	1.19373e+1	90.1	356	SPGM	6.32102	396.7	427
	Major	1.19399e+1	248.4	7	Major	*	*	*
	BRscg	1.19387e+1	260.9	6	BRscg	6.32373	789.0	9
E2	FilterASM	7.18086	31.8	106	FilterASM	2.56446e-1	114.3	136
	SPGM	7.18086	69.0	284	SPGM	2.56446e-1	360.5	429
	Major	7.18144	582.5	5	Major	*	*	*
	BRscg	7.18223	589.1	3	BRscg	2.56544e-1	1114.5	41
E3	FilterASM	2.26022e-13	0.2	0	FilterASM	2.39635e-13	0.5	0
	SPGM	6.34100e-13	4.5	1	SPGM	6.97391e-13	27.2	1
	Major	2.26022e-13	0.0	0	Major	2.39635e-13	0.0	0
	BRscg	2.26022e-13	0.1	0	BRscg	2.39635e-13	0.1	0
E4	FilterASM	3.63335e+2	0.2	0	FilterASM	3.63398e+2	0.4	0
	SPGM	3.63335e+2	2.0	1	SPGM	*	*	*
	Major	3.63335e+2	20.5	1	Major	3.63398e+2	33.0	1
	BRscg	3.63335e+2	22.4	1	BRscg	3.63398e+2	38.6	1

Table 4
Results for approximation problem of correlation matrix.

Prob.	$m = 387$, $p = 5$				$m = 387$, $p = 10$			
	Algorithm	f^*	time(s)	iter	Algorithm	f^*	time(s)	iter
E5	FilterASM	2.57370e+1	0.1	29	FilterASM	1.27045e+1	0.2	44
	SPGM	2.57370e+1	0.4	62	SPGM	1.27045e+1	0.6	104
	Major	2.57411e+1	1.6	2	Major	1.27108e+1	6.5	6
	BRscg	2.57371e+1	1.8	2	BRscg	1.27046e+1	6.9	3

Table 5
Results for approximation problem of correlation matrix.

Prob.	$m = 387$, $p = 50$				$m = 387$, $p = 100$			
	Algorithm	f^*	time(s)	iter	Algorithm	f^*	time(s)	iter
E5	FilterASM	2.25928e-1	11.8	547	FilterASM	4.08981e-3	0.6	22
	SPGM	2.25925e-1	93.0	7247	SPGM	3.78499e-3	1231.4	38923
	Major	2.34394e-1	164.6	90	Major	4.13419e-3	76.7	38
	BRscg	2.31586e-1	166.3	5	BRscg	4.17553e-3	77.2	1

Table 6
Results for approximation problem of correlation matrix.

Prob.	$m = 387$, $p = 250$			
	Algorithm	f^*	time(s)	iter
E5	FilterASM	4.93719e-5	0.1	0
	SPGM	4.93719e-5	1.5	1
	Major	4.93719e-5	0.0	0
	BRscg	4.93719e-5	0.0	0

in Table 7, where nls stands for the number of linear systems solved, and initial, final, min, max, mean, and std stand for the first element, the final element, the smallest element, the largest element, the mean value, and the standard deviation of $\{|\mathcal{A}_k|\}_{k=0,1,\ldots,\mathrm{iter}}$, respectively.

Table 7 partially reveals the numerical behaviors of FilterASM. There are four observations. First, on average 2.6 linear systems are needed to solve at each iteration.

TABLE 7
Further numerical results of FilterASM for approximation problems of correlation matrix.

Prob.	m	p	iter	nls	initial	final	min	max	mean	std
E1	400	100	72	146	400	0	0	400	27	101.7
	400	150	83	168	400	0	0	400	29	103.6
	400	200	64	132	400	2	0	400	38	116.4
	400	250	92	201	400	13	0	400	31	97.6
	400	300	92	221	400	54	0	400	46	95.6
	400	350	133	337	400	143	0	400	90	85.2
E2	400	100	80	162	400	0	0	400	25	96.9
	400	150	75	152	400	0	0	400	32	108.6
	400	200	94	228	400	142	0	400	63	98.9
	400	250	121	294	400	75	0	400	42	85.9
	400	300	65	132	400	0	0	400	36	115.9
	400	350	54	110	400	0	0	400	44	125.9
E3	400	100	47	117	400	379	0	400	175	157.6
	400	150	66	182	400	392	0	400	181	146.6
	400	200	78	209	400	355	0	400	161	141.3
	400	250	86	226	400	320	0	400	128	131.2
	400	300	147	367	400	284	0	400	128	110.3
	400	350	103	259	400	122	0	400	67	94.2
E4	400	100	101	205	400	400	0	400	359	108.7
	400	150	91	184	400	400	0	400	369	97.1
	400	200	75	153	400	400	0	400	374	96.6
	400	250	69	143	400	400	0	400	349	131.5
	400	300	69	142	400	400	0	400	349	134.1
	400	350	69	141	400	400	0	400	357	118.2
E5	387	100	896	2539	387	25	0	387	56	48.7
	387	150	527	1531	387	268	0	387	265	56.8
	387	200	494	1381	387	361	0	387	307	75.7
	387	250	438	1199	387	365	0	387	325	68.7
	387	300	314	856	387	377	0	387	347	70.1
	387	350	303	851	387	376	0	387	346	68.6

Second, we noted that the initial and maximum size of the working set $\mathcal{A}_k$ are the number of constraints for all test problems; this situation is due to the fact that our starting points are the projections of random vectors onto the feasible region. Third, min $= 0$ for all test problems, which indicates that at least for some k the working set $\mathcal{A}_k$ is empty during the entire iteration process in FilterASM. Last, except for several extreme cases, the mean value of the working set $\mathcal{A}_k$ is smaller than the final value, while the standard deviation is large in comparison with the mean value.

5.2. The maximal correlation problem. We next test FilterASM on the maximal correlation problem, which is of the form

$$\text{(5.2)} \qquad \text{(MCP)} \begin{cases} \max\limits_{x \in \mathbb{R}^n} & f(x) := x^T G x \\ \text{s.t.} & c_i(x) := \|x_{[i]}\|^2 - 1 = 0, \ i = 1, 2, \ldots, m, \end{cases}$$

where $G \in \mathbb{R}^{n \times n}$ is a given matrix with $n = pm$, $x = (x_{[1]}^T, x_{[2]}^T, \ldots, x_{[m]}^T)^T$, and $x_{[i]} \in \mathbb{R}^p$, $i = 1, 2, \ldots, m$. A brief introduction about the statistical background of the maximal correlation problem can be found, e.g., in Chu and Watterson [9, section 2].

In our experiment, we use the command `randn` to generate a matrix which is then symmetrized to get a test G. Since the Horst–Jacobi algorithm and the Gauss–Seidel algorithm [45] perform much worse than the other solvers for large n, we only compare FilterASM with the Riemannian trust-region algorithm (RTR) [46] and adaptive

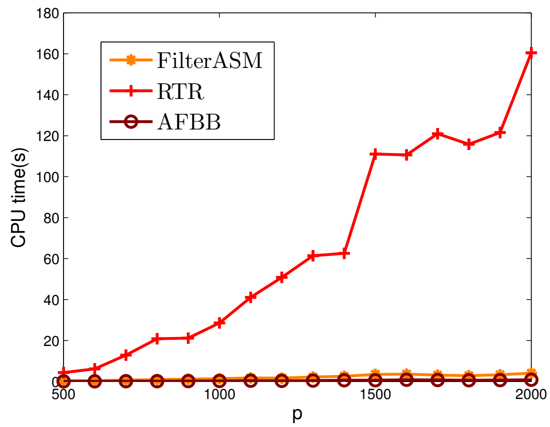

FIG. 1. *CPU time (s).*

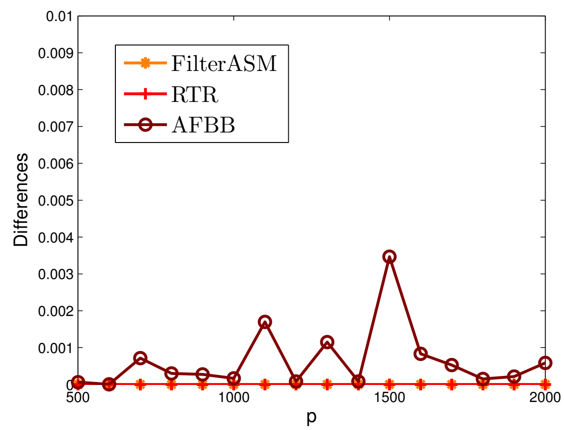

FIG. 2. *Differences of* $f^*_{\max}$ *and* f^*.

feasible BB-like (AFBB) method [23]. In our experiment, RTR terminates if

$$\|P_\Omega(\nabla f(x^k))\| \leq 5 \times 10^{-5}.$$

For AFFB, it terminates if one of the following three conditions is met:

$$\frac{\|x^{k+1} - x^k\|}{\sqrt{m}} \leq 10^{-5}, \quad \frac{|f(x^{k+1}) - f(x^k)|}{|f(x^k)| + 1} \leq 10^{-8}, \quad \|P_\Omega(\nabla f(x^k))\| \leq \times 10^{-5}.$$

We first test the instances with $m = 2$ and p varying from 500 to 2000; the numerical results averaged over two random tests are profiled in Figures 1 and 2, where $f^*_{\max}$ denotes the best (maximum) f^* computed by FilterASM, RTR, and AFBB. We observed that, compared with RTR and AFBB, FilterASM obtains fairly good solutions with reasonable CPU time.

Last, we test the situation $p = 100$ when m varies; the numerical results averaged over two random tests are summarized in Table 8. The following observations are from Table 8.

TABLE 8
Numerical results of FilterASM, RTR, and AFBB for MCP.

m	$f^*_{\max}$	FilterASM			RTR			AFBB		
		$f^*_{\max} - f^*$	time(s)	iter	$f^*_{\max} - f^*$	time(s)	iter	$f^*_{\max} - f^*$	time(s)	iter
10	4.44e+2	1.4e-8	0.4	94	0	3.0	15	2.0e-4	0.1	72
20	1.26e+3	3.1e-8	1.2	133	0	15.7	17	4.7e-4	0.3	124
30	2.31e+3	6.8e-8	2.2	121	0	37.1	16	1.2e-4	0.5	105
40	3.56e+3	7.0e-8	4.3	143	0	106.2	18	2.9e-3	1.2	162
50	4.98e+3	1.3e-7	6.5	147	0	155.6	18	1.3e-3	2.2	185
60	6.54e+3	4.5e-7	11.3	178	0	271.5	20	8.0e-3	3.2	188
70	8.25e+3	3.4e-7	16.3	194	0	382.0	20	3.4e-3	4.1	178
80	1.01e+4	3.5e-7	18.4	167	0	518.7	18	1.3e-3	4.9	163
90	1.20e+4	0	32.8	244	5.3e-1	1105.5	24	5.4e-1	14.2	370
100	1.41e+4	9.1e-7	39.3	233	0	1136.4	20	5.0e-2	10.0	201

- In terms of CPU time, AFBB is best, FilterASM is second, and RTR is third.
- In terms of the objective function values, RTR is best (except for one case), FilterASM is second but much closer to RTR, and AFBB is third.
- Overall, FilterASM obtains satisfactory solutions with moderate CPU time, though it requires more iterations than RTR and AFBB.

6. Conclusion. In this paper, we proposed the FilterASM for the ball/sphere constrained optimization problem, which uses economic computational costs at each iteration but guarantees the global convergence and locally superlinear convergence. The active-set technique is used to generate the working set, and at each iteration, only two or three reduced linear systems need to be solved for the search direction. Taking advantage of the structure of BCOP, a new L-BFGS scheme and duality technique are exploited to reduce the computational effort for solving the resulting linear systems; the L-BFGS formula also provides approximate second-order information to accelerate the speed. We used the filter technique to globalize the convergence of the iteration, where an economic feasibility restoration phase is embedded. Under some mild conditions, the global and local convergence is established. Finally, we conducted preliminary numerical experiments on two specific applications and our numerical results show that FilterASM is competitive with some custom-made methods proposed recently for each individual application.

Acknowledgments. We would like to thank the two anonymous referees for their helpful suggestions and comments, which have improved the presentation of this paper significantly. We also thank Dr. Borsdorf Rüdiger for sharing us with the codes of SPGM, Dr. Li Qingna for the data of problem E5 and the codes of Major and BRscg, and Dr. Jiang Bo for the codes of AFBB.

REFERENCES

[1] C. ALEXANDER, *Common correlation and calibrating the lognormal forward rate model*, Wilmott Mag., 2 (2003), pp. 68–78.

[2] L. ANDERSON, J. SIDENIUS, AND S. BASU, *All Your Hedges in One Basket*, Risk. net, 2003, pp. 67–72.

[3] S. BAI, H.-D QI, AND N. XIU, *Constrained best Euclidean distance embedding on a sphere: A matrix optimization approach*, SIAM J. on Optim., 25 (2015), pp. 439–467.

[4] R. BORSDORF, N. J. HIGHAM, AND M. RAYDAN, *Computing a nearest correlation matrix with factor structure*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2603–2622.

[5] J. V. BURKE, F. E. CURTIS, AND H. WANG, *A sequential quadratic optimization algorithm with rapid infeasibility detection*, SIAM J. Optim., 24 (2014), pp. 839–872.

[6] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, and H. C. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Program. Stud., 16 (1982), pp. 1–17.

[7] L. F. Chen, Y. L. Wang, and G. P. He, *A feasible active set QP-free method for nonlinear programming*, SIAM J. Optim., 17 (2006), pp. 401–429.

[8] N. Chiang and V. M. Zavala, *An Inertia-Free Filter Line-Search Algorithm for Large-Scale Nonlinear Programming*, http://www.optimization-online.org/DB_HTML/2014/09/4550.html (2014).

[9] M. T. Chu and J. L. Watterson, *On a multivariate eigenvalue problem:* I. *Algebraic theory and power method*, SIAM J. Sci. Comput., 14 (1993), pp. 1089–1106.

[10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[11] M. Crouhy, D. Galai, and R. Mark, *A comparative analysis of current credit risk models*, J. Banking Finance, 24 (2000), pp. 59–117.

[12] P. Glasserman and S. Suchintabandid, *Correlation expansions for CDO pricing*, J. Banking Finance, 31 (2007), pp. 1375–1398.

[13] F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998), pp. 14–32.

[14] R. Fletcher, N. Gould, S. Leyffer, Ph.L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter allgorithm for general nonlinear programming*, SIAM J. Optim., 13 (2002), pp. 635–659.

[15] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program, 91 (2002), pp. 239–269.

[16] R. Fletcher, S. Leyffer, and Ph. L. Toint, *On the global convergence of a filter-SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.

[17] Z. Y. Gao, G. P. He, and F. Wu, *An algorithm of sequential systems of linear equations for nonlinear optimization problems with arbitrary initial point*, Sci. China Ser. A, 401 (1997), pp. 561–57.

[18] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006.

[19] N. Gould and D. P. Robinson, *A second derivative SQP method: Global convergence*, SIAM J. Optim., 20 (2010), pp. 2023–2048.

[20] N. Gould and D. P. Robinson, *A second derivative SQP method: Local convergence and practical issues*, SIAM J. Optim., 20 (2010), pp. 2049–2079.

[21] W. W. Hager and M. S. Gowda, *Stability in the presence of degeneracy and error estimation*, Math. Program, 85 (1999), pp. 181–192.

[22] J. B. Jian and W. X. Cheng, *A superlinearly convergent strongly sub-feasible SSLE-type algorithm with working set for nonlinearly constrained optimization*, J. Comput. Appl. Math., 225 (2009), pp. 172–186.

[23] B. Jiang and Y. H. Dai, *A framework of constraint preserving update schemes for optimization on Stiefel manifold*, Math. Program, 153 (2015), pp. 535–575.

[24] E. Karas, A. Ribeiro, C. Sagastizbal, and M. Solodov, *A bundle-filter method for nonsmooth convex constrained optimization*, Math. Program., 116 (2009), pp. 297–320.

[25] Q. N. Li, H. D. Qi, and N. H. Xiu, *Block relaxation and majorization methods for the nearest correlation matrix with factor structure*, Comput. Optim. Appl., 50 (2011), pp. 327–349.

[26] D. C. Liu and J. Nocedal, *On the limited memory method for large scale optimization*, Math. Program. B, 45 (1989), pp. 503–528.

[27] X. W. Liu and Y. X. Yuan, *A sequential quadratic programming method without a penalty function or a filter for nonlinear equality constrained optimization*, SIAM J. Optim., 21 (2011), pp. 545–571.

[28] F. Lillo and R. N. Mantegna, *Spectral density of the correlation matrix of factor models: A random matrix theory approach*, Phys. Rev. E, 72 (2005), 016219.

[29] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New york, 2006.

[30] C. Oberlin and S. J. Wright, *Active set identification in nonlinear programming*, SIAM J. Optim., 17 (2006), pp. 577–605.

[31] E. R. Panier, A. L. Tits, and J. N. Herskovits, *A QP-free globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization*, SIAM J. Control Optim., 26 (1988), pp. 788–811.

[32] H. D. Qi and L. Qi, *A new QP-free globally convergent, locally superlinear convergent algorithm for inequality constrained optimization*, SIAM J. Optim., 11 (2000), pp. 113–132.

[33] H. D. Qi and D. F. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 360–385.

[34] A. A. Ribeiro, E. W. Karas, and C. C. Gonzaga, *Global convergence of filter methods for nonlinear programming*, SIAM J. Optim., 19 (2008), pp. 1231–1249.

[35] C. G. Shen, S. Leyffer, and R. Fletcher, *A nonmonotone filter method for nonlinear optimization*, Comput. Optim. Appl., 52 (2012), pp. 583–607.

[36] C. G. Shen, W. J. Xue, and D.G. Pu, *An infeasible SSLE filter algorithm for general constrained optimization without strict complementarity*, Asia-Pac. J. Oper. Res., 28 (2011), pp. 361–399.

[37] C. G. Shen, L. H. Zhang, B. Wang, and W. Q. Shao, *Global and local convergence of a nonmonotone SQP Method for constrained nonlinear optimization*, Comput. Optim. Appl., 59 (2014), pp. 435–473.

[38] W. Y. Sun and Y. X. Yuan, *Optimization Theory and Methods*, in Nonlinear Programming, Springer, New York, 2006.

[39] M. Ulbrich, S. Ulbrich, and L. N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Math. Program., 100 (2004), pp. 379–410.

[40] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: Motivation and global convergence*, SIAM J. Optim., 16 (2005), pp. 1–31.

[41] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: Local convergence*, SIAM J. Optim., 16 (2005), pp. 32–48.

[42] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.

[43] Y. L. Wang, L. F. Chen, and G. P. He, *Sequential systems of linear equations method for general constrained optimization without strict complementarity*, J. Comput. Appl. Math., 182 (2005), pp. 447–471.

[44] Y. F. Yang, D. H. Li, and L. Q. Qi, *A feasible sequential linear equation method for inequality constrained optimization*, SIAM J. Optim., 13 (2003), pp. 1222–1244.

[45] L. H. Zhang and M. T. Chu, *Computing absolute maximum correlation*, IMA J. Numer. Anal., 32 (2011), pp. 163–184.

[46] L. H. Zhang, *Riemannian trust-region method for the maximal correlation problem*, Numer. Funct. Anal. Optim., 33 (2012), pp. 338–362.