# An arc-search interior-point algorithm for nonlinear constrained optimization

Yaguang Yang*

January 3, 2025

**Abstract**

This paper proposes a new arc-search interior-point algorithm for the nonlinear constrained optimization problem. The proposed algorithm uses the second-order derivatives to construct a search arc that approaches the optimizer. Because the arc stays in the interior set longer than any straight line, it is expected that the scheme will generate a better new iterate than a line search method. The computation of the second-order derivatives requires to solve the second linear system of equations, but the coefficient matrix of the second linear system of equations is the same as the first linear system of equations. Therefore, the matrix decomposition obtained while solving the first linear system of equations can be reused. In addition, most elements of the right-hand side vector of the second linear system of equations are already computed when the coefficient matrix is assembled. Therefore, the computation cost for solving the second linear system of equations is insignificant and the benefit of having a better search scheme is well justified. The convergence of the proposed algorithm is established. Some preliminary test results are reported to demonstrate the merit of the proposed algorithm.

**Keywords:** Arc-search, interior-point algorithm, nonlinear optimization.
**MSC classification:** 90C30 90C51.

## 1 Introduction

The interior-point method has been very successful in solving the large scale linear optimization problem [22, 39]. It is then extended to solve the convex quadratic optimization problem [24], the general convex optimization problem [23], the linear complementarity problem [16], and the semi-definite optimization problem [1]. Not surprisingly, there are many interior-point algorithms developed for the nonlinear optimization problem [2, 3, 4, 5, 6, 7, 8, 9, 18, 19, 26, 29, 32, 34, 35, 37, 38, 41]. Most of the proposed algorithms use the line search technique but the trust region technique is also considered by researchers, for example [3, 4, 29]. For the linear optimization problem, the path-following interior-point method is most popular because of its efficiency, but for

*Goddard Space Flight Center, NASA, 8800 Greenbelt Rd, Greenbelt, MD 20771. Email: yaguang.yang@nasa.gov.

the nonlinear optimization problem, using logarithmic barrier functions is widely considered because of its clear connection to the penalty method, exceptions are [7, 34, 41]. Although using higher-order derivatives in interior-point method is regarded as a useful strategy for linear optimization problems [22, 39], it is not be adopted by almost all interior-point algorithms designed for nonlinear optimization problems as it was pointed out in [41] that the right-hand side vector of the second-order derivatives equations involves the computation of the third-order derivatives of Lagrangian, which is very expensive.

In this paper, we propose a higher-order arc-search interior-point algorithm which is different from most existing algorithms in that (a) it is a path-following algorithm, its merit for linear optimization problems is demonstrated in [39]; (b) it uses an arc instead of a straight line to search for an optimizer, a useful strategy that has been proved by different researchers for various optimization problems [13, 14, 15, 42, 43, 44, 46, 47, 52, 53], in particular, this author showed in [46] that an arc-search infeasible-interior point algorithm using higher-order derivatives, searching in a widest neighborhood, and starting from an infeasible point, is computationally competitive to the famous Mehrotra's algorithm and has the best-known convergence rate, which solved a long-standing problem, puzzling the researchers for years [33]; and (c) it solves two linear systems of equations with the same square coefficient matrix using just one matrix decomposition, a brilliant idea introduced by Mehrotra [22] and used by interior-point algorithms designed for various optimization problems [15, 39, 46]. The proposed algorithm improves the efficiency of the algorithm of [41] by avoiding the computation of the third-order derivatives of Lagrangian on the right-hand side of the second linear system of equations. Since most elements of the reduced right-hand side of the second linear system of equations are obtained already while we construct the square coefficient matrix, the cost of solving the second linear system of equations is insignificant but we get a significantly better search arc. A warm-restart update scheme for the relaxation variables and multipliers is introduced aiming at further improving the computational efficiency. The convergence of the proposed algorithm is proved under some mild conditions similar to the ones assumed in [7, 41]. Limited by the length of this paper, we provide only some preliminary test results, and leave all implementation details and extensive test results in a follow-on paper. Some applications of the proposed algorithm to the spacecraft optimal trajectory design problems are reported in a recent conference paper [50], and another application to a spacecraft formation flying orbit optimization problem is discussed in [49].

The remainder of the paper is organized as follows. Section 2 introduces the nonlinear optimization problem. Section 3 presents the proposed arc-search interior-point algorithms. Section 4 discusses their convergence properties. Section 5 provides preliminary numerical test results on some benchmark test problems. Section 6 summarizes the conclusion of the paper.

# 2 Problem description

The following notations will be used throughout this paper. We use bold small letters for vectors, bold capital letters for matrices, and normal font for scalars. For a vector $\mathbf{x}$, we use $x_i$ for the $i$-th element of $\mathbf{x}$, and $\mathbf{X}$ for a diagonal matrix whose diagonal elements are the vector $\mathbf{x}$. For vectors or matrices, we will use the superscript $k$ to represent their values at the $k$-th iteration. For scalars, we use the subscript $k$ to represent their values at the $k$-th iteration, while using the superscript

to represent their powers. Finally, we use $\mathbb{R}^n_+$ ($\mathbb{R}^n_{++}$) to denote the space of nonnegative vectors (positive vectors, respectively), and use $\mathbf{e}$ to denote a vector of all ones with appropriate dimension.

In this paper, we consider the following nonlinear optimization problem.

$$\begin{array}{rcl} \min & : & f(\mathbf{x}) \\ \text{s.t.} & : & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^m$, and $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$. To ensure that the system has a solution, we assume $n > m \geq 0$. Without loss of generality, we assume $p \geq 1$, that is needed for interior-point method and simplifies the discussion. The decision vector is $\mathbf{x} \in \mathbb{R}^n$. We introduce a nonnegative relaxation vector $\mathbf{s} \geq \mathbf{0}$ to rewrite (1) as

$$\begin{array}{rcl} \min & : & f(\mathbf{x}) \\ \text{s.t.} & : & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & & \mathbf{g}(\mathbf{x}) - \mathbf{s} = \mathbf{0}, \\ & & \mathbf{s} \geq \mathbf{0}. \end{array} \tag{2}$$

For problem (2), the Lagrangian function is given by

$$L(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{s}, \mathbf{z}) = f(\mathbf{x}) - \mathbf{y}^{\mathrm{T}}\mathbf{h}(\mathbf{x}) - \mathbf{w}^{\mathrm{T}}(\mathbf{g}(\mathbf{x}) - \mathbf{s}) - \mathbf{z}^{\mathrm{T}}\mathbf{s}, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^p_+$, and $\mathbf{z} \in \mathbb{R}^p_+$ are multipliers. To simplify the notation, we use $\mathbf{v} = (\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{s}, \mathbf{z}) \in \mathbb{R}^{n+m+3p}$ to denote a stacked column vector composed of decision variables, relaxation variables, and multipliers. The gradients of the Lagrangian function with respect to $\mathbf{x}$ and $\mathbf{s}$ are given by

$$\nabla_{\mathbf{x}} L(\mathbf{v}) = \nabla f(\mathbf{x}) - \nabla \mathbf{h}(\mathbf{x})\mathbf{y} - \nabla \mathbf{g}(\mathbf{x})\mathbf{w}, \tag{4a}$$

$$\nabla_{\mathbf{s}} L(\mathbf{v}) = \mathbf{w} - \mathbf{z}, \tag{4b}$$

respectively, where $\nabla \mathbf{h}(\mathbf{x}) \in \mathbb{R}^{n \times m}$ and $\nabla \mathbf{g}(\mathbf{x}) \in \mathbb{R}^{n \times p}$ are Jacobians given by

$$\nabla \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_2}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_1}{\partial x_n} & \frac{\partial h_2}{\partial x_n} & \cdots & \frac{\partial h_m}{\partial x_n} \end{bmatrix} = [\nabla h_1(\mathbf{x}), \cdots, \nabla h_m(\mathbf{x})],$$

and

$$\nabla \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_2}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \frac{\partial g_2}{\partial x_n} & \cdots & \frac{\partial g_p}{\partial x_n} \end{bmatrix} = [\nabla g_1(\mathbf{x}), \cdots, \nabla g_p(\mathbf{x})].$$

Then, the KKT conditions of (2) are given by (see [27]):

$$\mathbf{k}(\mathbf{v}) = \begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{v}) \\ \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) - \mathbf{s} \\ \mathbf{w} - \mathbf{z} \\ \mathbf{Z}\mathbf{s} \end{bmatrix} = \mathbf{0}, \ (\mathbf{w}, \mathbf{s}, \mathbf{z}) \in \mathbb{R}^{3p}_+, \tag{5}$$

where $\mathbf{k}: \mathbb{R}^{n+m+3p} \to \mathbb{R}^{n+m+3p}$. It is worthwhile to note that $\nabla_{\mathbf{x}} L(\mathbf{v})$ depends on the choice of the multiplier vector $\mathbf{y}$ which has no restriction as long as its elements are real numbers. In addition, the multiplier $\mathbf{y}$ is unrelated to any other components of $\mathbf{k}(\mathbf{v})$. Moreover, $\mathbf{s} > \mathbf{0}$ is unrelated to any components of $\mathbf{k}(\mathbf{v})$ except $\mathbf{g}(\mathbf{x}) - \mathbf{s} = \mathbf{0}$. These features will be used while we introduce a warm-restart scheme to minimize $\|\mathbf{k}(\mathbf{v})\|$. For the vector $\mathbf{k}(\mathbf{v})$, its value at the $k$-th iteration is expressed by

$$\mathbf{k}(\mathbf{v}^k) = \begin{bmatrix} \nabla f(\mathbf{x}^k) - \nabla \mathbf{h}(\mathbf{x}^k)\mathbf{y}^k - \nabla \mathbf{g}(\mathbf{x}^k)\mathbf{w}^k \\ \mathbf{h}(\mathbf{x}^k) \\ \mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k \\ \mathbf{w}^k - \mathbf{z}^k \\ \mathbf{Z}^k \mathbf{s}^k \end{bmatrix}. \tag{6}$$

Therefore, the Jacobian of $\mathbf{k}(\mathbf{v})$ is given by

$$\mathbf{k}'(\mathbf{v}) = \begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}) & -\nabla \mathbf{h}(\mathbf{x}) & -\nabla \mathbf{g}(\mathbf{x}) & \mathbf{0} & \mathbf{0} \\ (\nabla \mathbf{h}(\mathbf{x}))^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\nabla \mathbf{g}(\mathbf{x}))^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z} & \mathbf{S} \end{bmatrix}, \tag{7}$$

where

$$\nabla^2_{\mathbf{x}} L(\mathbf{v}) = \nabla^2_{\mathbf{x}} f(\mathbf{x}) - \sum_{i=1}^{m} \nabla^2_{\mathbf{x}} h_i(\mathbf{x}) y_i - \sum_{i=1}^{p} \nabla^2_{\mathbf{x}} g_i(\mathbf{x}) w_i. \tag{8}$$

Let the index set of the active inequality constraints $\mathbf{g}(\mathbf{x})$ at $\mathbf{x} \in \mathbb{R}^n$ be denoted by

$$I(\mathbf{x}) = \{i \in \{1, \ldots, p\} : g_i(\mathbf{x}) = 0\}. \tag{9}$$

For convergence analysis, we need the definition of Lipschitz continuity. Let $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ be real functions of $\mathbf{x}$. Let an open set $\mathcal{M}$ contain the level set $\mathcal{L} = \{\mathbf{x} \mid \alpha(\mathbf{x}) \leq \alpha(\mathbf{x}_0)\}$, we say $\beta(\mathbf{x})$ is Lipschitz continuous if there is a constant $L > 0$ such that

$$\|\beta(\mathbf{x}) - \beta(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \tag{10}$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{M}$. Under some mild assumptions used in [7, 27, 41], similar to the proof of [48], it is straightforward to show the following result.

**Proposition 2.1** *Assume existence, smoothness, regularity, sufficiency, and strict complementarity for problem (2) described as follows:*

*(A1) Existence. There exists an optimal solution and its associate multipliers* $\mathbf{v}^* = (\mathbf{x}^*, \mathbf{y}^*, \mathbf{w}^*, \mathbf{s}^*, \mathbf{z}^*)$ *of (2). Clearly, the KKT conditions (5) hold for any optimal solution.*

*(A2) Smoothness.* $f(\mathbf{x})$, $\mathbf{h}(\mathbf{x})$ *and* $\mathbf{g}(\mathbf{x})$ *are differentiable up to the third order. In addition,* $\nabla_{\mathbf{x}} L(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, *and* $\mathbf{h}(\mathbf{x})$ *are Lipschitz continuous.*

*(A3) Regularity. The set* $\{\nabla h_j(\mathbf{x}^*) : j = 1, \ldots, m\} \cup \{\nabla g_i(\mathbf{x}^*) : i \in I(\mathbf{x}^*)\}$ *is linearly independent.*

*(A4) Sufficiency. For all* $\eta \in \mathbb{R}^n \backslash \{0\}$, *we have* $\eta^{\mathrm{T}} \nabla^2_{\mathbf{x}} L(\mathbf{v}^*)\eta > 0$.

*(A5) Strict complementarity. For each* $i \in \{1, \ldots, p\}$, *we have* $z_i^* + s_i^* > 0$ *and* $z_i^* s_i^* = 0$.

*Then the Jacobian matrix at the optimal solution* $\mathbf{v}^*$ *is nonsingular, i.e.,* $[k'(\mathbf{v}^*)]^{-1}$ *exists and is bounded.*

**Remark 2.1** *Assuming the continuity of* $[k'(\mathbf{v}^*)]^{-1}$, *Proposition 2.1 implies that all local optimizers are isolated.*

# 3 The arc-search interior-point algorithm

The main idea of the proposed arc-search interior-point algorithm is to extend a similar algorithm for linear programming presented in [46] to the nonlinear programming problem. Given the current iterate $\mathbf{v}^k = (\mathbf{x}^k, \mathbf{y}^k, \mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k)$ with $(\mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k) > \mathbf{0}$ and $t \geq 0$, we consider the solution $\mathbf{v}(t) = (\mathbf{x}(t), \mathbf{y}(t), \mathbf{w}(t), \mathbf{s}(t), \mathbf{z}(t))$ of the following nonlinear system of equations defined for $t \in [0, 1]$

$$\begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{v}(t)) \\ \mathbf{h}(\mathbf{x}(t)) \\ \mathbf{g}(\mathbf{x}(t)) - \mathbf{s}(t) \\ \nabla_{\mathbf{s}} L(\mathbf{v}(t)) \\ \mathbf{Z}(t)\mathbf{s}(t) \end{bmatrix} = t \cdot \begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{v}^k) \\ \mathbf{h}(\mathbf{x}^k) \\ (\mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k) \\ \nabla_s L(\mathbf{v}^k) \\ \mathbf{Z}^k\mathbf{s}^k \end{bmatrix}, \quad (\mathbf{w}(t), \mathbf{s}(t), \mathbf{z}(t)) \geq \mathbf{0}. \tag{11}$$

Clearly, when $t = 1$, the system has a solution of $\mathbf{v}(1) = \mathbf{v}^k$; and when $t \to 0$, the system approaches the solution of the KKT system. Under some mild conditions to be defined in (B1)-(B4), and using the inverse function theorem [21], Yamashita et al. [41] showed the following result.

**Proposition 3.1 ([41])** *Let* $\Omega(\epsilon)$ *be defined in (38), assume that conditions (B1)-(B4) given in Section 4 hold, and* $\mathbf{v}^k \subset \Omega(\epsilon)$, *then the solution* $\mathbf{v}(t)$ *of the system of the nonlinear equations (11) is unique for every* $t \in (0, 1]$.

Therefore, $\mathbf{v}(t) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^p$ is an arc that starts at $\mathbf{v}^k$ and guides the current iterate to a candidate of a local optimal solution of (1) because (11) reduces to the KKT conditions when $t \to 0$. Therefore, the proposed algorithm is a path-following algorithm.

However, solving the system of nonlinear equations of (11) is impractical. Similar to the idea used in [41, 46], our strategy is to approximate the arc $\mathbf{v}(t)$ by part of an ellipse which is defined as:

$$\mathcal{E} = \{\mathbf{v}(\alpha) : \mathbf{v}(\alpha) = \vec{\mathbf{a}}\cos(\alpha) + \vec{\mathbf{b}}\sin(\alpha) + \vec{\mathbf{c}}, \alpha \in [0, 2\pi]\}, \tag{12}$$

where $\vec{\mathbf{a}} \in \mathbb{R}^{n+m+3p}$ and $\vec{\mathbf{b}} \in \mathbb{R}^{n+m+3p}$ are the axes of the ellipse, and $\vec{\mathbf{c}} \in \mathbb{R}^{n+m+3p}$ is its center. Then, we search the optimizer along the ellipse $\mathcal{E}$ to get a new iterate $\mathbf{v}^{k+1}$ and repeat the process until an $\epsilon$-optimizer is found. To construct the ellipse of (12), we assume that the first and second derivatives of (11) and (12) are the same at current iterate $\mathbf{v}^k$ because (12) is an estimate of (11).

Taking the derivative on both sides of (11) yields

$$\begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) & -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{h}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{g}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \\ \dot{\mathbf{w}} \\ \dot{\mathbf{s}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{v}^k) \\ \mathbf{h}(\mathbf{x}^k) \\ \mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k \\ \mathbf{w}^k - \mathbf{z}^k \\ \mathbf{Z}^k \mathbf{s}^k \end{bmatrix}. \tag{13}$$

It has been shown in [39] that the search performance may be improved by adding a centering term in (13). Let the dual measure be defined as:

$$\mu = \frac{\mathbf{z}^{\mathrm{T}} \mathbf{s}}{p}. \tag{14}$$

Following the common strategy used in most interior point algorithms (see [39]), we introduce a centering parameter $\sigma$ and modify (13) as:

$$\begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) & -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{h}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{g}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{y}} \\ \dot{\mathbf{w}} \\ \dot{\mathbf{s}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{x}} L(\mathbf{v}^k) \\ \mathbf{h}(\mathbf{x}^k) \\ \mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k \\ \mathbf{w}^k - \mathbf{z}^k \\ \mathbf{Z}^k \mathbf{s}^k - \sigma \mu_k \mathbf{e} \end{bmatrix}. \tag{15}$$

where $\mathbf{e}$ is an all-one vector of dimension $p$. This modification enforces the arc $\mathbf{v}(\alpha)$ toward the interior of the constraints, thereby makes sure that a substantial segment of the ellipse satisfies $(\mathbf{w}(\alpha), \mathbf{s}(\alpha), \mathbf{z}(\alpha)) > \mathbf{0}$, which may increase the step size and improve the efficiency of the algorithm. Since (13) is a special case of (15) with $\sigma = 0$, we will use (15) throughout the rest paper.

Since the right-hand side of (15) at $\mathbf{v}^k$ is a constant vector, taking the derivative on both sides of (15) yields

$$\begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) & -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{h}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{g}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\mathbf{y}} \\ \ddot{\mathbf{w}} \\ \ddot{\mathbf{s}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{p}(\mathbf{v}^k) \\ -(\nabla^2_{\mathbf{x}} \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} \dot{\mathbf{x}} \dot{\mathbf{x}} \\ -(\nabla^2_{\mathbf{x}} \mathbf{g}(\mathbf{x}^k))^{\mathrm{T}} \dot{\mathbf{x}} \dot{\mathbf{x}} \\ \mathbf{0} \\ -2 \dot{\mathbf{Z}} \dot{\mathbf{s}} \end{bmatrix}, \tag{16}$$

where

$$\mathbf{p}(\mathbf{v}^k) = -(\nabla^3_{\mathbf{x}} L(\mathbf{v}^k)) \dot{\mathbf{x}} \dot{\mathbf{x}} + 2(\nabla^2_{\mathbf{x}} \mathbf{h}(\mathbf{x}^k)) \dot{\mathbf{y}} \dot{\mathbf{x}} + 2(\nabla^2_{\mathbf{x}} \mathbf{g}(\mathbf{x}^k)) \dot{\mathbf{w}} \dot{\mathbf{x}}.$$

The formulas for computing the elements on the right-hand side of (16) can be found in [41, Appendix 1], and we provide them below for completeness.

$$\nabla^3_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}, \mathbf{z}) \dot{\mathbf{x}} \dot{\mathbf{x}} = \frac{\partial \left( \frac{\partial^2 L(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial \mathbf{x}^2} \dot{\mathbf{x}} \right)}{\partial \mathbf{x}} \dot{\mathbf{x}} = \sum_{i=1}^{n} \dot{x}_i \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \frac{\partial^2 L(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial x_1 \partial x_i} \\ \vdots \\ \frac{\partial^2 L(\mathbf{x}, \mathbf{y}, \mathbf{z})}{\partial x_n \partial x_i} \end{bmatrix} \dot{\mathbf{x}} \tag{17}$$

$$\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x})\dot{\mathbf{y}}\dot{\mathbf{x}} = \frac{\partial\left(\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{y}}\right)}{\partial \mathbf{x}}\dot{\mathbf{x}} = \sum_{i=1}^{m}\dot{y}_i\frac{\partial}{\partial \mathbf{x}}\begin{bmatrix}\frac{\partial h_i(\mathbf{x})}{\partial x_1}\\ \vdots \\ \frac{\partial h_i(\mathbf{x})}{\partial x_n}\end{bmatrix}\dot{\mathbf{x}} = \sum_{i=1}^{m}\dot{y}_i\left(\nabla^2_{\mathbf{x}}h_i(\mathbf{x})\right)\dot{\mathbf{x}} \tag{18}$$

$$\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x})\dot{\mathbf{w}}\dot{\mathbf{x}} = \frac{\partial\left(\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\dot{\mathbf{w}}\right)}{\partial \mathbf{x}}\dot{\mathbf{x}} = \sum_{i=1}^{n}\dot{w}_i\frac{\partial}{\partial \mathbf{x}}\begin{bmatrix}\frac{\partial g_i(\mathbf{x})}{\partial x_1}\\ \vdots \\ \frac{\partial g_i(\mathbf{x})}{\partial x_n}\end{bmatrix}\dot{\mathbf{x}} = \sum_{i=1}^{n}\dot{w}_i\left(\nabla^2_{\mathbf{x}}g_i(\mathbf{x})\right)\dot{\mathbf{x}} \tag{19}$$

$$\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x})^{\mathrm{T}}\dot{\mathbf{x}}\dot{\mathbf{x}} = \left(\frac{\partial\left(\left(\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mathbf{x}}\right)}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{x}}^{\mathrm{T}}\left(\nabla^2_{\mathbf{x}}h_1(\mathbf{x})\right)\dot{\mathbf{x}}\\ \vdots \\ \dot{\mathbf{x}}^{\mathrm{T}}\left(\nabla^2_{\mathbf{x}}h_m(\mathbf{x})\right)\dot{\mathbf{x}}\end{bmatrix} \tag{20}$$

$$\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x})^{\mathrm{T}}\dot{\mathbf{x}}\dot{\mathbf{x}} = \left(\frac{\partial\left(\left(\frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mathbf{x}}\right)}{\partial \mathbf{x}}\right)^{\mathrm{T}}\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{x}}^{\mathrm{T}}\left(\nabla^2_{\mathbf{x}}g_1(\mathbf{x})\right)\dot{\mathbf{x}}\\ \vdots \\ \dot{\mathbf{x}}^{\mathrm{T}}\left(\nabla^2_{\mathbf{x}}g_p(\mathbf{x})\right)\dot{\mathbf{x}}\end{bmatrix}. \tag{21}$$

**Remark 3.1** *It is noteworthy that the Hessian matrices of* $\nabla^2_{\mathbf{x}}h_i(\mathbf{x})$ *and* $\nabla^2_{\mathbf{x}}g_i(\mathbf{x})$ *in (18), (19), (20), and (21) are obtained when* $\nabla^2_{\mathbf{x}}L(\mathbf{v})$ *is computed in (8). The major burden of the computation of the right-hand side of (16) is*

$$\frac{\partial}{\partial \mathbf{x}}\begin{bmatrix}\frac{\partial^2 L(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial x_1 \partial x_i}\\ \vdots \\ \frac{\partial^2 L(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial x_n \partial x_i}\end{bmatrix} = \sum_{i=1,\ldots,n}\frac{\partial^2}{\partial \mathbf{x}^2}\left[\frac{\partial L(\mathbf{x},\mathbf{y},\mathbf{z})}{\partial x_i}\right] \tag{22}$$

$$= \sum_{i=1,\ldots,n}\frac{\partial^2}{\partial \mathbf{x}^2}\left(\frac{df(\mathbf{x})}{dx_i}\right) - \sum_{\substack{i\,=\,1,\ldots,m\\ j\,=\,1,\ldots,n}} y_i\frac{\partial^2}{\partial \mathbf{x}^2}\left(\frac{dh_i(\mathbf{x})}{dx_j}\right) - \sum_{\substack{i\,=\,1,\ldots,p\\ j\,=\,1,\ldots,n}} w_i\frac{\partial^2}{\partial \mathbf{x}^2}\left(\frac{dg_i(\mathbf{x})}{dx_j}\right).$$

*For quadratically constrained quadratic programming (QCQP) problem, the increased computational burden of the right-hand side of (16) is minimum because all terms in (22) become zeros.*

The computation of (22) is expensive and adversely affects the performance of the algorithm of [41]. Assuming the third-order derivatives are relatively small (at least when iterates are close to the optimal solution), we consider an alternative approximation by modifying (16) to:

$$\begin{bmatrix}\nabla^2_{\mathbf{x}}L(\mathbf{v}^k) & -\nabla\mathbf{h}(\mathbf{x}^k) & -\nabla\mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0}\\ \left(\nabla\mathbf{h}(\mathbf{x}^k)\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \left(\nabla\mathbf{g}(\mathbf{x}^k)\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{x}}\\ \ddot{\mathbf{y}}\\ \ddot{\mathbf{w}}\\ \ddot{\mathbf{s}}\\ \ddot{\mathbf{z}}\end{bmatrix} = \begin{bmatrix}2[(\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k))\dot{\mathbf{w}}\dot{\mathbf{x}} + (\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k))\dot{\mathbf{y}}\dot{\mathbf{x}}]\\ -(\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k))^{\mathrm{T}}\dot{\mathbf{x}}\dot{\mathbf{x}}\\ -(\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k))^{\mathrm{T}}\dot{\mathbf{x}}\dot{\mathbf{x}}\\ \mathbf{0}\\ -2\dot{\mathbf{Z}}\dot{\mathbf{s}}\end{bmatrix}, \tag{23}$$

which ignore the third-order derivatives on the right-hand side, or even a simpler formula which ignore all high-order derivatives on the right-hand side, i.e.,

$$\begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) & -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{h}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \big(\nabla \mathbf{g}(\mathbf{x}^k)\big)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\mathbf{y}} \\ \ddot{\mathbf{w}} \\ \ddot{\mathbf{s}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -2\dot{\mathbf{Z}}\dot{\mathbf{s}} \end{bmatrix}. \tag{24}$$

Our preliminary numerical experience quickly denies the idea of using (24) in the construction of the ellipse, because it oversimplifies the right-hand side. Although, using (24) does reduce the computational cost in every iteration, but it significantly increases iteration numbers, and overall is less efficient than using (16) or (23).

Let $\dot{\mathbf{v}} = (\dot{\mathbf{x}}, \dot{\mathbf{y}}, \dot{\mathbf{w}}, \dot{\mathbf{s}}, \dot{\mathbf{z}})$ be obtained by using (15), and $\ddot{\mathbf{v}} = (\ddot{\mathbf{x}}, \ddot{\mathbf{y}}, \ddot{\mathbf{w}}, \ddot{\mathbf{s}}, \ddot{\mathbf{z}})$ be obtained by using either (16) or (23), which are the first-order and second-order derivatives of the ellipse $\mathcal{E}$ at the current iterate. Then, the following theorem provides an equivalent formula of the ellipse (12).

**Theorem 3.1** [44] Suppose that the ellipse $\mathcal{E}$ of (12) passes through the current iterate $\mathbf{v}^k$ at $\alpha = 0$, and its first and second order derivatives at $\alpha = 0$ are $\dot{\mathbf{v}}$ and $\ddot{\mathbf{v}}$, respectively. Then $\mathcal{E}$ is given by $\mathbf{v}(\alpha) = (\mathbf{x}(\alpha), \mathbf{y}(\alpha), \mathbf{w}(\alpha), \mathbf{s}(\alpha), \mathbf{z}(\alpha))$ which can be calculated by

$$\mathbf{v}(\alpha) = \mathbf{v}^k - \dot{\mathbf{v}} \sin(\alpha) + \ddot{\mathbf{v}}(1 - \cos(\alpha)). \tag{25}$$

The proof of the theorem is given in [44]. It is worthwhile to indicate that $\mathbf{v}(t)$ starts from $t = 1$ and ends at $t = 0$ for the arc represented by (11), but $\mathbf{v}(\alpha)$ starts from $\alpha = 0$ and ends at $\alpha = \frac{\pi}{2}$ for the arc represented by (12), i.e., $\mathbf{v}(t)|_{t=1} = \mathbf{v}(\alpha)|_{\alpha=0} = \mathbf{v}^k$ is the current iterate. Since the ellipse approximates the arc defined by (11), we assume that $\dot{\mathbf{v}}(t)|_{t=1} = \dot{\mathbf{v}}(\alpha)|_{\alpha=0}$ and $\ddot{\mathbf{v}}(t)|_{t=1} = \ddot{\mathbf{v}}(\alpha)|_{\alpha=0}$. For the sake of simplicity, we will use $\dot{\mathbf{v}}$ for $\dot{\mathbf{v}}(\alpha)|_{\alpha=0}$ and $\ddot{\mathbf{v}}$ for $\ddot{\mathbf{v}}(\alpha)|_{\alpha=0}$.

It is noteworthy that the direction of $\dot{\mathbf{v}} = (\dot{\mathbf{x}}, \dot{\mathbf{y}}, \dot{\mathbf{w}}, \dot{\mathbf{s}}, \dot{\mathbf{z}})$ calculated by using (13) is the same one (with a different sign) that is used by the steepest descent method for solving (5). To see this, for a sequence of $\mathbf{v}^k$, we use $o(t_k)$ for the claim that the condition $\|\mathbf{v}_k\| \leq t_k$ holds. At iteration $k$, steepest descent method calculates the next iterate $\mathbf{v}^{k+1} = \mathbf{v}^k + \alpha_k \mathbf{d}^k$ by using the search direction $\mathbf{d}^k$ and searching an appropriate step size $\alpha_k$. Using Taylor's expansion for (5), we have

$$\begin{aligned} \mathbf{k}(\mathbf{v}^{k+1}) &= \mathbf{k}(\mathbf{v}^k) + \mathbf{k}'(\mathbf{v}^k)(\mathbf{v}^{k+1} - \mathbf{v}^k) + o(\|\mathbf{v}^{k+1} - \mathbf{v}^k\|^2) \\ &= \mathbf{k}(\mathbf{v}^k) + \mathbf{k}'(\mathbf{v}^k)(\alpha_k \mathbf{d}^k) + o(\|\alpha_k \mathbf{d}^k\|^2) \end{aligned} \tag{26}$$

For steepest descent method, $\mathbf{d}^k$ is calculated by setting $\mathbf{k}(\mathbf{v}^k) + \mathbf{k}'(\mathbf{v}^k)\mathbf{d}^k = \mathbf{0}$ (the same system of equations of (13) with a different sign), i.e.,

$$\mathbf{d}^k = -[\mathbf{k}'(\mathbf{v}^k)]^{-1}\mathbf{k}(\mathbf{v}^k) = -\dot{\mathbf{v}}, \tag{27}$$

which is essentially the same search direction of (25) when the second-order term $\ddot{\mathbf{v}}$ is not used. That is the reason why the first-order interior-point algorithm is not efficient, and the second-order correction interior-point algorithm is used by some of the latest interior-point algorithms, such as IPOPT [37] and [41]. It is emphasized that the second-order correction by using the arc-search is fundamentally different from the one in [37].

**Remark 3.2** *It is worthwhile to note that the direction* $\mathbf{d}^k$ *of (27) involves the inverse of the Jacobian matrix and for this reason many researchers working on linear optimization refer to the method as Newton's method. But* $\mathbf{d}^k$ *is not a conventional Newton direction because of the definition of the vector* $\mathbf{k}(\mathbf{v}^k)$ *which is not the gradient of an objective function (see [27, Page 259]).*

**Remark 3.3** *The analytic arc-search formula provided by (25) is simple and its computational cost is negligible compared to other ones such as solving linear systems of equations. In addition, for boundary constraints, no arc-search is required because one may use analytic formulas provided in Appendix 2 of [41] to calculate the step size. It is emphasized that all the analyses in the rest of the paper are applicable to both of the following two cases: (a)* $\dot{\mathbf{v}}$ *is calculated by (15) and* $\ddot{\mathbf{v}}$ *is calculated by (16), and (b)* $\dot{\mathbf{v}}$ *is calculated by (15) and* $\ddot{\mathbf{v}}$ *is calculated by (23).*

Note that the fourth row of (15) indicates that if $\mathbf{w}^k = \mathbf{z}^k$, then $\dot{\mathbf{w}} - \dot{\mathbf{z}} = \mathbf{w}^k - \mathbf{z}^k = \mathbf{0}$. Similarly, in view of the fourth row of (16) or (23), we have $\ddot{\mathbf{w}} - \ddot{\mathbf{z}} = \mathbf{w}^k - \mathbf{z}^k = \mathbf{0}$. In view of Theorem 3.1, we have the following result.

**Lemma 3.1 ([41])** *If* $\mathbf{v}^k$ *satisfies* $\mathbf{w}^k = \mathbf{z}^k$*, then* $\mathbf{w}(\alpha) = \mathbf{z}(\alpha)$ *holds for any* $\alpha \in [0, \frac{\pi}{2}]$.

To search for the optimizer along the ellipse given by (25), we expect that every iterate will reduce the magnitude of $\mathbf{k}(\mathbf{v}^k)$ and eventually we will have $\mathbf{k}(\mathbf{v}^k) \to \mathbf{0}$ as $k \to 0$, thereby the KKT conditions will be satisfied. Therefore, we define the merit function by

$$\phi(\mathbf{v}) = \|\mathbf{k}(\mathbf{v})\|^2 \tag{28}$$

which should sufficiently decrease at $\mathbf{v}(\alpha)$ for some constants, $\rho \in (0, \frac{1}{2})$, $\sigma \in [0, 1)$, and step size $\alpha \in (0, \pi/2]$. The following lemma shows that this is achievable under the condition (29).

**Lemma 3.2** *Let* $\alpha \in (0, \frac{\pi}{2}]$*,* $\rho \in (0, \frac{1}{2})$ *and* $\sigma \in [0, 1)$*. Let* $\mu = \frac{\mathbf{z}^{\mathrm{T}}\mathbf{s}}{p}$*. If*

$$\phi(\mathbf{v}(\alpha)) \leq \phi(\mathbf{v}) + \rho \sin(\alpha) \phi'(\mathbf{v}(\alpha))|_{\alpha=0}, \tag{29}$$

*then*

$$\phi(\mathbf{v}(\alpha)) \leq \phi(\mathbf{v})(1 - 2\rho(1-\sigma)\sin(\alpha)) < \phi(\mathbf{v}). \tag{30}$$

**Proof:** Given the ranges of $\alpha, \rho$, and $\sigma$, the last inequality in (30) clearly holds. The first inequality in (30) follows from a similar argument used in [7]. Since $\dot{\mathbf{v}}$ is defined as the solution of (15) $\mathbf{k}'(\mathbf{v})\dot{\mathbf{v}} = \mathbf{k}(\mathbf{v}) - \sigma\mu\bar{\mathbf{e}}$, where $\bar{\mathbf{e}} = (\mathbf{0}, \mathbf{e})$ with $\mathbf{0} \in \mathbb{R}^{n+m+2p}$ and $\mathbf{e} \in \mathbb{R}^p$, using the definition (28) and then (25), we have

$$\begin{aligned}
\left.\frac{d\phi(\mathbf{v}(\alpha))}{d\alpha}\right|_{\alpha=0} &= 2\mathbf{k}(\mathbf{v}(\alpha))^{\mathrm{T}}\mathbf{k}'(\mathbf{v}(\alpha))\left.\frac{d\mathbf{v}(\alpha)}{d\alpha}\right|_{\alpha=0} \\
&= 2\mathbf{k}(\mathbf{v}(\alpha))^{\mathrm{T}}\mathbf{k}'(\mathbf{v}(\alpha))\left[-\dot{\mathbf{v}}\cos(\alpha) + \ddot{\mathbf{v}}\sin(\alpha)\right]\big|_{\alpha=0} \\
&= -2\mathbf{k}(\mathbf{v})^{\mathrm{T}}\mathbf{k}'(\mathbf{v})\dot{\mathbf{v}} \qquad (31) \\
&= -2\mathbf{k}(\mathbf{v})^{\mathrm{T}}[\mathbf{k}(\mathbf{v}) - \sigma\mu\bar{\mathbf{e}}] \\
&= -2\phi(\mathbf{v}) + 2\sigma\mu^2/p, \qquad (32)
\end{aligned}$$

where the last equality is derived from $\mathbf{k}(\mathbf{v})^{\mathrm{T}}(\sigma\mu\bar{\mathbf{e}}) = \sigma\mu\sum_{i=1}^{p} z_i s_i = \sigma\mu\mathbf{z}^{\mathrm{T}}\mathbf{s} = \sigma\mu^2/p$. Since $|\mathbf{z}^{\mathrm{T}}\mathbf{s}| \leq \sqrt{p}\|\mathbf{Z}\mathbf{s}\|_2$ and $p \geq 1$, we have

$$\mu^2/p = (\mathbf{z}^{\mathrm{T}}\mathbf{s})^2/p \cdot (1/p^2) \leq \|\mathbf{Z}\mathbf{s}\|_2^2 \cdot 1 \leq \|\mathbf{k}(\mathbf{v})\|_2^2 = \phi(\mathbf{v}).$$

Substituting this inequality into (32), we have

$$\phi'(\mathbf{v}(\alpha))|_{\alpha=0} \leq -2\phi(\mathbf{v}) + 2\sigma\phi(\mathbf{v}) \leq -2\phi(\mathbf{v})(1-\sigma). \tag{33}$$

From (29), it holds

$$\phi(\mathbf{v}(\alpha)) \leq \phi(\mathbf{v}) - 2\rho\sin(\alpha)\phi(\mathbf{v})(1-\sigma) = \phi(\mathbf{v})(1 - 2\rho(1-\sigma)\sin(\alpha)).$$

This completes the proof. ∎

**Remark 3.4** *If Condition (29) and $\sigma < \phi(\mathbf{v})p/\mu^2$ hold, then $\phi(\mathbf{v}(\alpha))$ decreases because $\alpha, \rho > 0$ and according to (32), $\phi'(\mathbf{v}(\alpha))|_{\alpha=0} < 0$.*

As a matter of fact, Condition (29) holds for $\sigma = 0$. In view of (31) and (13), since $\frac{d\phi(\mathbf{v}(\alpha))}{d\alpha}\Big|_{\alpha=0} = -2\mathbf{k}(\mathbf{v})^{\mathrm{T}}\mathbf{k}'(\mathbf{v})\dot{\mathbf{v}}$, it follows

$$\begin{aligned}
\phi(\mathbf{v}(\alpha)) &= \phi(\mathbf{v}(\alpha)) + \phi'(\mathbf{v})\alpha + o(\alpha^2) \\
&= \phi(\mathbf{v}(\alpha)) - 2\mathbf{k}(\mathbf{v})^{\mathrm{T}}\mathbf{k}'(\mathbf{v})\dot{\mathbf{v}}\alpha + o(\alpha^2) \\
&= \phi(\mathbf{v}) - 2\phi(\mathbf{v})\alpha + o(\alpha^2),
\end{aligned} \tag{34}$$

the last equation holds because of $\sigma = 0$. Therefore, for $\sigma = 0$ and $\alpha > 0$ small enough, we have $\phi(\mathbf{v}(\alpha)) < \phi(\mathbf{v})$. We summarize this result as the following lemma.

**Lemma 3.3** *Assume that $\sigma = 0$, then searching along the ellipse using (25) will always reduce the merit function.*

**Remark 3.5** *As $\phi(\mathbf{v}(\alpha)) \to 0$, we find a KKT point which may be a maximizer. In a follow-on paper, we will provide some implementation tricks to avoid this case.*

For any interior-point method, every iterate must be an interior-point, i.e., we must have $(\mathbf{w}, \mathbf{s}, \mathbf{z}) > 0$ and $\mathbf{g}(\mathbf{x}) > \mathbf{0}$ in all the iterations before the algorithm stops. In addition, all iterates should improve the merit function. To make sure that the algorithm is well-posed, we need that all iterates are bounded. In summary, we should choose the step size $\alpha \in (0, \frac{\pi}{2}]$ such that the following conditions hold.

(C1) $(\mathbf{w}^k(\alpha_k)), \mathbf{s}^k(\alpha_k), \mathbf{z}^k(\alpha_k)) > \mathbf{0}$.

(C2) $\mathbf{g}(\mathbf{x}^k(\alpha_k)) > \mathbf{0}$.

(C3) $\phi(\mathbf{v}^{k+1}) = \phi(\mathbf{v}^k(\alpha_k)) < \phi(\mathbf{v}^k)$.

(C4) The generated sequence $\{\mathbf{v}^k\}$ should be bounded.

Condition (C1) can be achieved by a process discussed in [45, 46]. Because of Lemma 3.1, we always have $\mathbf{z}^k(\alpha) = \mathbf{w}^k(\alpha)$. Therefore, we do not need to calculate both $\mathbf{w}^k(\alpha)$ and $\mathbf{z}^k(\alpha)$. For a fixed a small $\delta_1 \in (0,1)$, we will select the largest $\tilde{\alpha}_k$ such that all $\alpha \in [0, \tilde{\alpha}_k]$ satisfy

$$\mathbf{w}^{k+1} = \mathbf{w}^k(\alpha) = \mathbf{w}^k - \dot{\mathbf{w}}^k \sin(\alpha) + \ddot{\mathbf{w}}^k(1 - \cos(\alpha)) \geq \delta_1 \mathbf{w}^k, \tag{35}$$

To this end, for each $i \in \{1, \ldots, p\}$, we select the largest $\alpha^k_{w_i}$ such that the $i$th inequality of (35) holds for all $\alpha \in [0, \alpha^k_{w_i}]$. Then, we define

$$\tilde{\alpha}_k = \min_{i \in \{1,\ldots,p\}} \left\{ \min \left\{ \alpha^k_{w_i}, \frac{\pi}{2} \right\} \right\}. \tag{36}$$

The largest $\alpha^k_{w_i}$ can be computed in analytical forms (see the proof in [45]), which are provided in Appendix 2 of [41]. A lower bound estimation of $\tilde{\alpha}_k$ will be provided in Lemma 4.9. We may select $\mathbf{s}^{k+1}$ using the same process as we did for $\mathbf{w}^{k+1}$, but a warm start [51] is better because it further reduces the merit function $\phi(\mathbf{v}^{k+1}) = \phi(\mathbf{v}(\alpha))$. We will discuss this shortly.

To meet Condition (C2), we search for $\bar{\alpha}_k \leq \tilde{\alpha}_k$ for all $\alpha \in (0, \bar{\alpha}_k]$ and some $\delta_1 \in (0,1)$ such that $\mathbf{g}(\mathbf{x}^k(\alpha)) \geq \delta_1 \mathbf{s}^k > \mathbf{0}$. This can be achieved if $\mathbf{g}(\mathbf{x}^k) > \mathbf{0}$. Therefore, for all $\alpha \in (0, \bar{\alpha}_k]$, Condition (C2) holds.

To make sure that Condition (C3) holds, we search for $\check{\alpha}_k \in (0, \bar{\alpha}_k]$ such that condition (29) holds, i.e.,

$$\check{\alpha}_k = \max \left\{ \alpha \in (0, \bar{\alpha}_k] : \phi(\mathbf{v}^k(\alpha)) < \phi(\mathbf{v}^k) - \delta_2 \right\} \tag{37}$$

where $\delta_2 \geq 0$ is a constant. The existence of $\check{\alpha}_k$ is guaranteed due to Lemma 3.3. According to Remark 3.4, $\sigma_k < \phi(\mathbf{v}^k)p/\mu^2$ should be selected. A lower bound estimation for $\check{\alpha}_k$ will be provided in Lemma 4.11.

Finally, we will show that Condition (C4) holds under some mild conditions. We will prove in Lemma 4.5 that $\{\mathbf{v}^k\}$ is bounded, and $\{(\mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k)\}$ is bounded below and away from zero[1]. To this end, we would like to have $\mathbf{v}^k$ stay in a desired set before our algorithms converge. Let $\epsilon > 0$ be the convergence criterion selected in the algorithms to be discussed, we define the desired set $\Omega(\epsilon)$ as follows:

$$\Omega(\epsilon) = \left\{ \mathbf{v} \in \mathbb{R}^{n+m+3p} : \epsilon \leq \phi(\mathbf{v}) \leq \phi(\mathbf{v}^0),\ \min(\mathbf{Zs}) \geq \frac{1}{2} \frac{\mathbf{Z}^0 \mathbf{s}^0}{\phi(\mathbf{v}^0)} \min(\phi(\mathbf{v})) \right\}. \tag{38}$$

To make sure that $\mathbf{v}^k$ stays in $\Omega(\epsilon)$, let

$$\hat{m}_k(\alpha) = \min(\mathbf{Z}^k(\alpha)\mathbf{s}^k(\alpha)) - \frac{1}{2} \frac{\mathbf{Z}^0 \mathbf{s}^0}{\phi(\mathbf{v}^0)} \min(\phi(\mathbf{v}(\alpha))). \tag{39}$$

An $\hat{\alpha}_k$ is selected to satisfy $\hat{m}_k(\alpha_k) \geq 0$ for $\forall \alpha_k \in (0, \hat{\alpha}_k]$, i.e.,

$$\hat{\alpha}_k = \max \left\{ \alpha \in (0, \check{\alpha}_k] : \hat{m}_k(\alpha) \geq 0 \right\}. \tag{40}$$

[1]A sequence $\{\mathbf{c}^k\} \subset \mathbb{R}^n$ is said to be *bounded below and away from zero if there exists* $\bar{c} > 0$ *such that every element of* $\mathbf{c}^k$ *satisfies* $c^k_i \geq \bar{c}$ *for all* $k \geq 1$.

This requirement has a similar effect as the wide neighborhood of interior-point methods [39]. We will prove in Lemma 4.13 that $\hat{\alpha}_k > 0$ is bounded below and away from zero before our algorithms converge.

Summarizing the above process, we may select the step size in the $k$th iteration as:

$$\alpha_k = \min\{\tilde{\alpha}, \check{\alpha}_k, \bar{\alpha}_k, \hat{\alpha}_k\} = \hat{\alpha} > 0. \tag{41}$$

A reasonable update is $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{w}^{k+1}, \mathbf{z}^{k+1}) = (\mathbf{x}(\alpha_k), \mathbf{y}(\alpha_k), \mathbf{w}(\alpha_k), \mathbf{z}(\alpha_k))$. As we mentioned above, our intuition and numerical experience made us believe that a warm start strategy will be more efficient because it further reduces the merit function in every iteration. Given $\mathbf{x}^{k+1}$, if we select $\mathbf{s}^{k+1} = \mathbf{g}(\mathbf{x}^{k+1}) > 0$, then, $\mathbf{g}(\mathbf{x}^{k+1}) - \mathbf{s}^{k+1} = \mathbf{0}$, this reduces the third component of $\mathbf{k}(\mathbf{v}^{k+1})$ to zero and therefore reduces the merit function $\phi(\mathbf{v}^{k+1})$. We may also select $\mathbf{y}^{k+1}$ to reduce the first component of $\mathbf{k}(\mathbf{v}^{k+1})$ by minimizing $\|\nabla f(\mathbf{x}^{k+1}) - \nabla \mathbf{h}(\mathbf{x}^{k+1})\mathbf{y}^{k+1} - \nabla \mathbf{g}(\mathbf{x}^{k+1})\mathbf{w}^{k+1}\|$ because there is no restriction on $\mathbf{y}^{k+1}$ except that $\mathbf{y}^{k+1}$ is a real vector. To achieve this, we can simply solve the following linear system of equations

$$\nabla \mathbf{h}(\mathbf{x}^{k+1})\mathbf{y}^{k+1} = \nabla f(\mathbf{x}^{k+1}) - \nabla \mathbf{g}(\mathbf{x}^{k+1})\mathbf{w}^{k+1} \tag{42}$$

for $\mathbf{y}^{k+1}$.

We will compare algorithms proposed in this paper to the algorithm of Yamashita et. al [41] which is given as follows:

**Algorithm 3.1 (The first infeasible arc-search interior-point algorithm [41])**

*Parameters:* $\epsilon > 0$, $\delta_1 > 0$, $\delta_2 > 0$, $\rho \in (0, \frac{1}{2})$, $\bar{\sigma} \in (0, \frac{1}{2})$, *and* $\gamma_{-1} \in [0.5, 1)$.
*Initial point:* $\mathbf{v}^0 = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0)$ *such that* $(\mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0) > \mathbf{0}$, $\mathbf{g}(\mathbf{x}^0) > 0$, *and* $\mathbf{w}^0 = \mathbf{z}^0$.
***for*** *k=0,1,2,...* ***do***
  *Calculate* $\nabla_{\mathbf{x}} L(\mathbf{v}^k)$, $\mathbf{h}(\mathbf{x}^k)$, $\mathbf{g}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}} \mathbf{g}(\mathbf{x}^k)$. *If* $\phi(\mathbf{v}^k) \leq \epsilon$, *stop.*
  *Calculate* $\nabla^2_{\mathbf{x}} \mathbf{h}(\mathbf{x}^k)$, $\nabla^2_{\mathbf{x}} \mathbf{g}(\mathbf{x}^k)$, *and* $\nabla^2_{\mathbf{x}} L(\mathbf{v}^k)$.
  *Select* $\sigma_k$ *such that* $\bar{\sigma} \leq \sigma_k < \min\{\frac{1}{2}, \phi(\mathbf{v}^k)p/\mu^2\}$.
  *Calculate* $\dot{\mathbf{v}}^k$ *by solving (15) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate the righthand side of (16).*
  *Calculate* $\ddot{\mathbf{v}}^k$ *by solving (16) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate* $\tilde{\alpha}_k$ *using (36) and search* $\bar{\alpha}_k \in (0, \tilde{\alpha}_k]$ *such that* $\mathbf{g}(\mathbf{x}(\bar{\alpha}_k)) > \mathbf{0}$.
  *Search* $\check{\alpha}_k \in (0, \bar{\alpha}_k]$ *such that (37) holds.*
  *Determine* $\hat{\alpha}_k > 0$ *using (39) and (40).*
  *Set* $\alpha_k = \min\{\hat{\alpha}_k, \check{\alpha}_k\}$.
  *Update* $\mathbf{v}^{k+1} = \mathbf{v}^k(\alpha_k) = \mathbf{v}^k - \dot{\mathbf{v}}^k \sin(\alpha_k) + \ddot{\mathbf{v}}^k(1 - \cos(\alpha_k))$ .
  $k + 1 \to k$.
***end for***

Our first algorithm uses the accurate righthand side of (16).

**Algorithm 3.2 (The second infeasible arc-search interior-point algorithm)**

*Parameters:* $\epsilon > 0$, $\delta_1 > 0$, $\delta_2 > 0$, $\rho \in (0, \frac{1}{2})$, *and* $\bar{\sigma} \in [0, \frac{1}{2})$.
*Initial point:* $\mathbf{v}^0 = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0)$ *such that* $(\mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0) > \mathbf{0}$, $\mathbf{g}(\mathbf{x}^0) > 0$, *and* $\mathbf{w}^0 = \mathbf{z}^0$.
***for*** *k=0,1,2,...* ***do***
  *Calculate* $\nabla_{\mathbf{x}} L(\mathbf{v}^k)$, $\mathbf{h}(\mathbf{x}^k)$, $\mathbf{g}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k)$. *If* $\phi(\mathbf{v}^k) \leq \epsilon$, *stop.*
  *Calculate* $\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k)$, $\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k)$, *and* $\nabla^2_{\mathbf{x}} L(\mathbf{v}^k)$.
  *Select* $\sigma_k$ *such that* $\bar{\sigma} \leq \sigma_k < \min\{\frac{1}{8}, \phi(\mathbf{v}^k)p/\mu^2\}$.
  *Calculate* $\dot{\mathbf{v}}^k$ *by solving (15) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate the righthand side of (16).*
  *Calculate* $\ddot{\mathbf{v}}^k$ *by solving (16) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate* $\tilde{\alpha}_k$ *using (36) and search* $\bar{\alpha}_k \in (0, \tilde{\alpha}_k]$ *such that* $\mathbf{g}(\mathbf{x}(\bar{\alpha}_k)) > \mathbf{0}$.
  *Search* $\check{\alpha}_k \in (0, \bar{\alpha}_k]$ *such that (37) holds.*
  *Determine* $\hat{\alpha}_k > 0$ *using (39) and (40).*
  *Set* $\alpha_k = \min\{\hat{\alpha}_k, \check{\alpha}_k\}$.
  *Update* $(\mathbf{x}^{k+1}, \mathbf{w}^{k+1}) = (\mathbf{x}^k, \mathbf{w}^k) - (\dot{\mathbf{x}}^k, \dot{\mathbf{w}}^k)\sin(\alpha_k) + (\ddot{\mathbf{x}}^k, \ddot{\mathbf{w}}^k)(1 - \cos(\alpha_k))$.
  *Select* $\mathbf{y}^{k+1}$ *by solving (42), and set* $\mathbf{s}^{k+1} = \mathbf{g}(\mathbf{x}^{k+1}) > \mathbf{0}$ *and* $\mathbf{z}^{k+1} = \mathbf{w}^{k+1}$.
  $k + 1 \to k$.
***end for***

**Remark 3.6** *A major difference between Algorithms 3.1 and 3.2 is to use a warm start in Step 15 of Algorithm 3.2, where we select* $\mathbf{y}^{k+1}$ *to minimize* $\|\nabla f(\mathbf{x}^{k+1}) - \nabla\mathbf{h}(\mathbf{x}^{k+1})\mathbf{y}^{k+1} - \nabla\mathbf{g}(\mathbf{x}^{k+1})\mathbf{w}^{k+1}\|$ *and enforce equation* $\mathbf{g}(\mathbf{x}^{k+1}) - \mathbf{s}^{k+1} = \mathbf{0}$ *in every iteration* $k$, *which makes Algorithm 3.2 converge faster and more robust. Finding an appropriate initial point satisfying* $\mathbf{g}(\mathbf{x}^0) > 0$ *is an important step and was not discussed in [41], but will be discussed in a follow-on paper.*

Our extensive experience shows (i) the most expensive cost in Algorithms 3.1 and 3.2 is the computation of the third order derivatives on the right hand side of (16), and (ii) when the iterates are close to the optimal solution, the magnitude of the third-order derivative is small. Therefore, we may ignore the third-order derivative terms in Algorithm 3.2. This leads to an alternative infeasible arc-search interior-point algorithm that calculates $\dot{\mathbf{v}}$ and $\ddot{\mathbf{v}}$ by (15) and (23).

**Algorithm 3.3 (The third infeasible arc-search interior-point algorithm)**

*Parameters:* $\epsilon > 0$, $\delta_1 > 0$, $\delta_2 > 0$, $\rho \in (0, \frac{1}{2})$, *and* $\bar{\sigma} \in [0, \frac{1}{2})$.
*Initial point:* $\mathbf{v}^0 = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0)$ *such that* $(\mathbf{w}^0, \mathbf{s}^0, \mathbf{z}^0) > \mathbf{0}$, $\mathbf{g}(\mathbf{x}^0) > 0$, *and* $\mathbf{w}^0 = \mathbf{z}^0$.
***for*** *k=0,1,2,...* ***do***
  *Calculate* $\nabla_{\mathbf{x}} L(\mathbf{v}^k)$, $\mathbf{h}(\mathbf{x}^k)$, $\mathbf{g}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k)$, $\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k)$. *If* $\phi(\mathbf{v}^k) \leq \epsilon$, *stop.*
  *Calculate* $\nabla^2_{\mathbf{x}}\mathbf{h}(\mathbf{x}^k)$, $\nabla^2_{\mathbf{x}}\mathbf{g}(\mathbf{x}^k)$, *and* $\nabla^2_{\mathbf{x}} L(\mathbf{v}^k)$.
  *Select* $\sigma_k$ *such that* $\bar{\sigma} \leq \sigma_k < \min\{\frac{1}{8}, \phi(\mathbf{v}^k)p/\mu^2\}$.
  *Calculate* $\dot{\mathbf{v}}^k$ *by solving (15) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate the righthand side of (23).*
  *Calculate* $\ddot{\mathbf{v}}^k$ *by solving (23) at* $\mathbf{v} = \mathbf{v}^k$.
  *Calculate* $\tilde{\alpha}_k$ *using (36) and search* $\bar{\alpha}_k \in (0, \tilde{\alpha}_k]$ *such that* $\mathbf{g}(\mathbf{x}(\bar{\alpha}_k)) > \mathbf{0}$.
  *Search* $\check{\alpha}_k \in (0, \bar{\alpha}_k]$ *such that (37) holds.*
  *Determine* $\hat{\alpha}_k > 0$ *using (39) and (40).*

*13:* *Set* $\alpha_k = \min\{\hat{\alpha}_k, \check{\alpha}_k\}$.
*14:* *Update* $(\mathbf{x}^{k+1}, \mathbf{w}^{k+1}) = (\mathbf{x}^k, \mathbf{w}^k) - (\dot{\mathbf{x}}^k, \dot{\mathbf{w}}^k)\sin(\alpha_k) + (\ddot{\mathbf{x}}^k, \ddot{\mathbf{w}}^k)(1 - \cos(\alpha_k))$.
*15:* *Select* $\mathbf{y}^{k+1}$ *by solving (42), and set* $\mathbf{s}^{k+1} = \mathbf{g}(\mathbf{x}^{k+1}) > \mathbf{0}$ *and* $\mathbf{z}^{k+1} = \mathbf{w}^{k+1}$.
*16:* $k + 1 \to k$.
*17:* ***end for***

**Remark 3.7** *By avoiding the calculation of the third-order derivatives on the right-hand side of (16), the major cost in the computation of the right-hand side of (23) are Hessians and they have been calculated when we construct the matrix of (15). Since the matrix decomposition of (23) is obtained when we solve (15), this decomposition can be reused when we solve (23). Therefore, the computational cost of the higher-order Algorithm 3.3 in every iteration is comparable to many first-order interior-point Algorithms such as [3, 4, 7, 26, 29, 32, 34, 35]. With a careful implementation, our experience shows that the convergence of the Algorithm 3.3 is faster than Algorithm 3.2. All implementation details for this algorithm will be discussed in a follow-on paper.*

# 4 Convergence analysis

Our convergence analysis is very general in the sense that it is applicable to all three algorithms presented in the previous section. We will use the following assumptions like the ones used in [7, 41].

**Assumptions**

(B1) The sequence $\{\mathbf{x}^k\} \subset \mathbb{R}^n$ is bounded.

(B2) For $\mathbf{v} \in \Omega(\epsilon)$, let $I_s^k$ be the index set $\{i : 1 \leq i \leq p,\ s_i^k = 0\}$. Then, the determinant of $(\mathbf{J}^k)^{\mathrm{T}}\mathbf{J}^k$ is bounded below and away from zero, where $\mathbf{J}^k$ is a matrix whose column vectors are composed of

$$\{\nabla h_j(\mathbf{x}^k) : j = 1, \ldots, m\} \cup \{\nabla g_i(\mathbf{x}^k) : i \in I_s^k\}.$$

**Remark 4.1** *Assumption (B2) is the standard LICQ condition [27, page 328], which makes sure that the linear systems of equations (15), (16), and (23) have solutions and helps us to establish a strong convergence result, i.e., the algorithm converges in finite steps. It is worthwhile to note that (B2) implies that in the set* $\Omega(\epsilon)$*, the columns of* $\nabla\mathbf{h}(\mathbf{x})$ *are linearly independent.*

The idea of the convergence analysis is similar to but not exactly the same as the one of [41]. We provide all proofs because it will be easy for readers to check that the claims hold for all three algorithms. First, we establish the boundness results for series of functions and vectors.

**Lemma 4.1** *Assume that* $\phi(\mathbf{v}^0)$ *is bounded. Then the series* $\{\phi(\mathbf{v}^k)\}$ *created by any of the three Algorithms is bounded. Therefore, the series* $\{\mathbf{k}(\mathbf{v}^k)\}$ *is bounded. This implies that* $\{\nabla_{\mathbf{x}} L(\mathbf{v}^k)\}$, $\{\mathbf{h}(\mathbf{x}^k)\}$, $\{\mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k\}$, *and* $\{\mathbf{Z}^k\mathbf{s}^k\}$ *are all bounded.*

**Proof:** In view of Lemmas 3.2 and 3.3, it follows that for all three Algorithms, $\{\phi(\mathbf{v}^k)\}$ is monotonically decreasing. This shows the first claim. By the definition of $\{\phi(\mathbf{v}^k)\}$, the second claim is true. The last claim follows from the definition of $\mathbf{k}(\mathbf{v}^k)$ in (5). ∎

In the following few lemmas, we will show that $\{\mathbf{v}^k\}$ is bounded.

**Lemma 4.2** *Assume that (B1) holds, then $\{\mathbf{s}^k\} > 0$ is bounded.*

**Proof:** From (B1) and the continuity of $\mathbf{g}$, the boundedness of $\{\mathbf{x}^k\}$ implies that $\{\mathbf{g}(\mathbf{x}^k)\}$ is bounded because $\mathbf{g}$ is differentiable and therefore continuous. In view of Lemma 4.1, since $\|\mathbf{s}^k\| \leq \|\mathbf{g}(\mathbf{x}^k) - \mathbf{s}^k\| + \|\mathbf{g}(\mathbf{x}^k)\|$, it follows that $\{\mathbf{s}^k\}$ is bounded. The selection process of $\mathbf{s}^k$ guarantees that $\{\mathbf{s}^k\} > 0$ holds. ∎

**Lemma 4.3** *Assume that (B1) and (B2) hold and $\{\mathbf{z}^0\} = \{\mathbf{w}^0\} > 0$, then $\{\mathbf{z}^k\} = \{\mathbf{w}^k\} > 0$ is bounded.*

**Proof:** In view of Lemma 3.1, we have $\mathbf{w}^k = \mathbf{z}^k$. Since $\nabla f$ is continuous, (B1) implies that $\{\nabla f(\mathbf{x}^k)\}$ is bounded. From Lemma 4.1, $\{\nabla_{\mathbf{x}} L(\mathbf{v}^k)\}$ is bounded. In view of (4), it follows

$$\|\nabla \mathbf{h}(\mathbf{x}^k)\mathbf{y}^k + \nabla \mathbf{g}(\mathbf{x}^k)\mathbf{w}^k\| \leq \|\nabla_{\mathbf{x}} L(\mathbf{v}^k)\| + \|\nabla f(\mathbf{x}^k)\|,$$

therefore, $\{\nabla \mathbf{h}(\mathbf{x}^k)\mathbf{y}^k + \nabla \mathbf{g}(\mathbf{x}^k)\mathbf{w}^k\}$ is bounded. Let $(\hat{\mathbf{y}}, \hat{\mathbf{w}}) = (\mathbf{y}, \mathbf{w})/\|(\mathbf{y}, \mathbf{w})\|$. Suppose by contradiction that $z_i^k = w_i^k \to \infty$ as $k \to \infty$ for some $i$ and we denote this set as $I_s$. For $z_j^k \notin I_s$, $z_j^k = w_j^k < \infty$ as $k \to \infty$. Since $\|(\hat{\mathbf{y}}, \hat{\mathbf{w}})\| = 1$ and $\|(\mathbf{y}, \mathbf{w})\| \to \infty$, $\hat{w}_i \neq 0$ if $i \in I_s$ and $\hat{w}_j = 0$ if $j \notin I_s$. Note that $\{\mathbf{Z}^k\mathbf{s}^k\} = \{\mathbf{W}^k\mathbf{s}^k\}$ is bounded, if $w_i^k \to \infty$, then $s_i^k \to 0$, this shows that $I_s$ is the same set as defined in (B2). Since $\{\nabla \mathbf{h}(\mathbf{x}^k)\mathbf{y}^k + \nabla \mathbf{g}(\mathbf{x}^k)\mathbf{w}^k\}$ is bounded and $\hat{w}_j = 0$ for $j \notin I_s$, it follows

$$\nabla \mathbf{h}(\mathbf{x}^k)\hat{\mathbf{y}}^k + \nabla \mathbf{g}(\mathbf{x}^k)\hat{\mathbf{w}}^k = \nabla \mathbf{h}(\mathbf{x}^k)\hat{\mathbf{y}} + \sum_{i \in I_s} \nabla g_i(\mathbf{x}^k)\hat{w}_i \to 0. \tag{43}$$

This contradicts to the assumption of (B2). Therefore, $\{\mathbf{z}^k\} = \{\mathbf{w}^k\}$ is bounded. The selection process of $(\mathbf{w}^k, \mathbf{z}^k)$ gurantees that $(\mathbf{w}^k, \mathbf{z}^k) > 0$ holds. ∎

**Lemma 4.4** *Assume that (B1) and (B2) hold, and $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, then $\{s_i^k\}$ and $\{z_i^k\}$ is bounded below from zero, and $\{\mathbf{y}^k\}$ is bounded.*

**Proof:** Since $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, the sequence $\{z_i^k s_i^k\}$ is all bounded below and away from zero for each $i = 1, \ldots, p$; more precisely, $z_i^k s_i^k \geq \frac{1}{2}\frac{z_i^0 s_i^0}{\phi(\mathbf{v}^0)}\phi(\mathbf{v}^k) \geq \frac{1}{2}\frac{z_i^0 s_i^0}{\phi(\mathbf{v}^0)}\epsilon$ for each $i$. Since $\{s_i^k\}$ is bounded, this shows $\{z_i^k\}$ is bounded below and away from zero. Similarly, $\{s_i^k\}$ is also bounded below and away from zero. In view of Remark 4.1, (B2) implies $\nabla \mathbf{h}(\mathbf{x}^k)$ is linearly independent. From (4), it follows

$$\mathbf{y}^k = -((\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} \nabla \mathbf{h}(\mathbf{x}^k))^{-1} (\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} \left[\nabla_{\mathbf{x}} L(\mathbf{v}^k) - \nabla f(\mathbf{x}^k) + \nabla \mathbf{g}(\mathbf{x}^k)\mathbf{w}^k\right],$$

hence, $\{\mathbf{y}^k\}$ is bounded because of the following claims: (a) $\{\mathbf{w}^k\}$ is bounded, (b) (B1) implies $\{\nabla f(\mathbf{x}^k)\}$ and $\{\nabla \mathbf{g}(\mathbf{x}^k)\}$ are bounded, (c) Lemma 4.1 indicates $\{\nabla \mathbf{h}(\mathbf{x}^k)\}$ and $\{\nabla_{\mathbf{x}} L(\mathbf{v}^k)\}$ are bounded, and (d) (B2) implies $((\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} \nabla \mathbf{h}(\mathbf{x}^k))^{-1}$ exists and is bounded. ∎

Summarizing Lemmas 4.1, 4.2, 4.3, and 4.4, we have the following Lemma.

**Lemma 4.5** *Assume that (B1) and (B2) hold, and $\{\phi(\mathbf{v}^0)\}$ is bounded. For some fixed $\epsilon > 0$, if the sequence $\{\mathbf{v}^k\}$ satisfies $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, then $\{\mathbf{v}^k\}$ is bounded and $\{(\mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k)\} > \mathbf{0}$ is bounded below and away from zero.*

Since $\mathbf{S}^k$ and $\mathbf{Z}^k$ are bounded below and away from zero, we would like introduce two more assumptions.

**Assumptions**

(B3) The matrix $\nabla^2_{\mathbf{x}} L(\mathbf{v}) + \nabla \mathbf{g}(\mathbf{x})\mathbf{S}^{-1}\mathbf{Z}(\nabla \mathbf{g}(\mathbf{x}))^{\mathrm{T}}$ is invertible for any $\mathbf{v}$ in $\Omega(\epsilon)$, and the set $\{(\nabla^2_{\mathbf{x}} L(\mathbf{v}) + \nabla \mathbf{g}(\mathbf{x})\mathbf{S}^{-1}\mathbf{Z}(\nabla \mathbf{g}(\mathbf{x}))^{\mathrm{T}})^{-1} : \mathbf{v} \in \Omega(\epsilon)\}$ is bounded.

(B4) For each k, the function $\{\phi'(\mathbf{x}^k(\alpha))\}$ is Lipschitz continuous, i.e., $\phi'(\mathbf{x}^k(\alpha)) - \phi'(\mathbf{x}^k) \leq L\alpha$ for a constant $L$.

**Remark 4.2** *Assumption (B3) is necessary and explains why an extensively studied problem [36] is not globally convergent (see problem (53)).*

The following lemma will be used to show the boundedness of the inverse of the Jacobian $\{\mathbf{k}'(\mathbf{v}^k)\}$.

**Lemma 4.6 ([20])** *Let* $\mathbf{R}$ *be a block matrix*

$$\mathbf{R} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}.$$

*If* $\mathbf{A}$ *and* $\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ *are invertible, or* $\mathbf{D}$ *and* $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ *are invertible, then* $\mathbf{R}$ *is invertible.*

The next lemma is given in [41], we provide the proof here so that the readers can check that the result is applicable to all three algorithms listed in the previous section.

**Lemma 4.7 ([41])** *Assume that (B1)-(B3) hold. If* $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$ *for a fixed* $\epsilon > 0$*, then* $\{[\mathbf{k}'(\mathbf{v}^k)]^{-1}\}$ *is bounded.*

**Proof:** Let

$$\mathbf{A}^k = \begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) \\ (\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} & \mathbf{0} \end{bmatrix}, \mathbf{B}^k = \begin{bmatrix} -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$
$$\mathbf{C}^k = \begin{bmatrix} (\nabla \mathbf{g}(\mathbf{x}^k))^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \text{ and } \mathbf{D}^k = \begin{bmatrix} \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & 0 & -\mathbf{I} \\ \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix}.$$

Then, we have

$$\mathbf{k}'(\mathbf{v}^k) = \begin{bmatrix} \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) & -\nabla \mathbf{h}(\mathbf{x}^k) & -\nabla \mathbf{g}(\mathbf{x}^k) & \mathbf{0} & \mathbf{0} \\ \left(\nabla \mathbf{h}(\mathbf{x}^k)\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \left(\nabla \mathbf{g}(\mathbf{x}^k)\right)^{\mathrm{T}} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}^k & \mathbf{S}^k \end{bmatrix} = \begin{bmatrix} \mathbf{A}^k & \mathbf{B}^k \\ \mathbf{C}^k & \mathbf{D}^k \end{bmatrix} \tag{44}$$

From Lemma 4.5, the two sequences $\{\mathbf{s}^k\}$ and $\{\mathbf{z}^k\}$ are bounded and each component of the two sequences are bounded below and away from zeros, therefore, the sequence $\{(\mathbf{D}^k)^{-1}\}$ is also bounded, where

$$(\mathbf{D}^k)^{-1} = \begin{bmatrix} (\mathbf{S}^k)^{-1}\mathbf{Z}^k & \mathbf{I} & (\mathbf{S}^k)^{-1} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ (\mathbf{S}^k)^{-1}\mathbf{Z}^k & \mathbf{0} & (\mathbf{S}^k)^{-1} \end{bmatrix}.$$

Let $\bar{\mathbf{L}} = \nabla^2_{\mathbf{x}} L(\mathbf{v}^k) + \nabla \mathbf{g}(\mathbf{x}^k)(\mathbf{S}^k)^{-1}\mathbf{Z}^k \nabla \mathbf{g}(\mathbf{x}^k)^{\mathrm{T}}$. We know that $\bar{\mathbf{L}}$ is invertible from Lemma 4.5 and (B3), therefore, $(\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} \bar{\mathbf{L}}^{-1} \nabla \mathbf{h}(\mathbf{x}^k)$ is also invertible from (B2). Therefore,

$$\mathbf{H}^k := \mathbf{A}^k - \mathbf{B}^k(\mathbf{D}^k)^{-1}\mathbf{C}^k = \begin{bmatrix} \bar{\mathbf{L}} & -\nabla \mathbf{h}(\mathbf{x}^k) \\ (\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}} & \mathbf{0} \end{bmatrix}$$

is invertible from Lemma 4.6. Since $\mathbf{D}^k$ and $\mathbf{H}^k$ are invertible, using Lemma 4.6 again, we conclude that $\mathbf{k}'(\mathbf{v}^k)$ is invertible.

Next, we show the boundedness of $\{[\mathbf{k}'(\mathbf{v}^k)]^{-1}\}$. Since $[\mathbf{k}'(\mathbf{v}^k)]^{-1}$ is given by

$$[\mathbf{k}'(\mathbf{v}^k)]^{-1} = \begin{bmatrix} (\mathbf{H}^k)^{-1} & -(\mathbf{H}^k)^{-1}\mathbf{B}^k(\mathbf{D}^k)^{-1} \\ -(\mathbf{D}^k)^{-1}\mathbf{C}^k(\mathbf{H}^k)^{-1} & (\mathbf{D}^k)^{-1}\mathbf{C}^k(\mathbf{H}^k)^{-1}\mathbf{B}^k(\mathbf{D}^k)^{-1} + (\mathbf{D}^k)^{-1} \end{bmatrix},$$

we need to show that $\{(\mathbf{H}^k)^{-1}\}$ is bounded. For each $k$, $(\mathbf{H}^k)^{-1}$ is given as follows:

$$(\mathbf{H}^k)^{-1} = \begin{bmatrix} \bar{\mathbf{L}}^{-1} + \bar{\mathbf{L}}^{-1}\nabla \mathbf{h}(\mathbf{x}^k)\bar{\mathbf{H}}^{-1}(\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}}\bar{\mathbf{L}}^{-1} & \bar{\mathbf{L}}^{-1}\nabla \mathbf{h}(\mathbf{x}^k)\bar{\mathbf{H}}^{-1} \\ \bar{\mathbf{H}}^{-1}(\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}}\bar{\mathbf{L}}^{-1} & +\bar{\mathbf{H}}^{-1} \end{bmatrix},$$

$\bar{\mathbf{H}} = (\nabla \mathbf{h}(\mathbf{x}^k))^{\mathrm{T}}\bar{\mathbf{L}}^{-1}\nabla \mathbf{h}(\mathbf{x}^k)$. Therefore, it is enough to show the boundedness of $\bar{\mathbf{L}}, \bar{\mathbf{H}}, \bar{\mathbf{L}}^{-1}$ and $\bar{\mathbf{H}}^{-1}$, and this is done by Assumptions (B3) and (B2), and Lemma 4.5. This completes the proof. ∎

The next lemma follows directly from Lemma 4.7.

**Lemma 4.8** *Assume that (B1)-(B3) hold and $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, then (a) Lines 7 and 9 in Algorithms 3.1, 3.2, and 3.3 are well-defined, and (b) the sequences $\{\dot{\mathbf{v}}^k\}$ and $\{\ddot{\mathbf{v}}^k\}$ are bounded.*

**Proof:** The claim (a) follows directly from Lemma 4.7. In view of (15), the boundedness of $\{[\mathbf{k}'(\mathbf{v}^k)]^{-1}\}$ and $\{\mathbf{v}^k\}$ guarantees the boundedness of $\{\dot{\mathbf{v}}^k\}$. Using (16) or (23), plus the boundedness of $\{\dot{\mathbf{v}}^k\}$, the boundedness of $\{\ddot{\mathbf{v}}^k\}$ follows from a similar argument. This shows claim (b). ∎

The boundedness results proved in the above lemmas guarantee that the step size $\alpha_k$ is bounded below and away from zero for every $k$ before the algorithms converge. The following lemmas are aimed at showing this claim.

**Lemma 4.9** *Let $\zeta_0 > 0$ be the minimum value of the elements in $\{(\mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k)\}$, and $(\zeta_1, \zeta_2) > \mathbf{0}$ be the upper bound of the absolute values of the element in $\{(\dot{\mathbf{w}}^k, \dot{\mathbf{s}}^k, \dot{\mathbf{z}}^k)\}$ and $\{(\ddot{\mathbf{w}}^k, \ddot{\mathbf{s}}^k, \ddot{\mathbf{z}}^k)\}$. Assume that (B1)-(B3) hold. If $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, then the sequence $\{\tilde{\alpha}_k\}$ is bounded below and away from zero. More specifically, $\tilde{\alpha}_k \geq \frac{\zeta_1 + \sqrt{\zeta_1^2 + 4(1-\delta_1)\zeta_0\zeta_2}}{2\zeta_2}$ for all $k \geq 0$.*

**Proof:** We can rewrite (35) as

$$(1-\delta_1)\mathbf{w}^k - \dot{\mathbf{w}}^k \sin(\alpha) + \ddot{\mathbf{w}}^k(1-\cos(\alpha)) \geq 0. \tag{45}$$

In view of Lemma 4.5, $\{(\mathbf{w}^k, \mathbf{s}^k, \mathbf{z}^k)\} > \mathbf{0}$ is bounded below and away from zero, thus $\{(1-\delta_1)\mathbf{w}^k\}$ is bounded below and away from zero. Since $\{\dot{\mathbf{w}}^k\}$ and $\{\ddot{\mathbf{w}}^k\}$ are bounded from Lemma 4.8, $\{\tilde{\alpha}_k\}$ should be bounded below and away from zero such that the inequality (45) holds for all $\alpha \in [0, \tilde{\alpha}_k]$. More precisely, since we take $\alpha$ from the range $\left[0, \frac{\pi}{2}\right]$, it holds that $\sin(\alpha) \leq \alpha$ and $1-\cos(\alpha) \leq 1-\cos^2(\alpha) = \sin^2(\alpha) \leq \alpha^2$. Therefore, (45) holds when

$$(1-\delta_1)\zeta_0 - \zeta_1 \sin(\alpha) - \zeta_2(1-\cos(\alpha)) \geq (1-\delta_1)\zeta_0 - \zeta_1\alpha - \zeta_2\alpha^2 \geq 0.$$

This indicates that (45) holds for any $\alpha \in \left[0, \frac{\zeta_1+\sqrt{\zeta_1^2+4(1-\delta_1)\zeta_0\zeta_2}}{2\zeta_2}\right]$. We can apply the same arguments to $\{\mathbf{s}^k\}$ and $\{\mathbf{z}^k\}$. This proves the lemma. ∎

Next, we show that $\{\bar{\alpha}_k\}$ is bounded below and away from zero.

**Lemma 4.10** *Assume that $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$, and Assumptions (B1)-(B3) hold. Then the sequence $\{\bar{\alpha}_k\}$ is bounded below and away from zero.*

**Proof:** According to Lemma 4.5, $\{\mathbf{s}^k\}$ is bounded below and away from zero, it follows that $\mathbf{g}(\mathbf{x}^k) \geq \mathbf{s}^k$ is bounded below and away from zero for all $k$ before the algorithms converge. Since $\mathbf{g}(\mathbf{x})$ is continuous, and according to Lemma 4.8, $\{\dot{\mathbf{v}}^k\}$ and $\{\ddot{\mathbf{v}}^k\}$ are bounded, it must have an $\bar{\alpha}_k > 0$ bounded below and away from zero such that for all $\alpha \in (0, \bar{\alpha}_k]$, $\mathbf{g}(\mathbf{x}(\alpha)) = \mathbf{g}(\mathbf{x}^k - \dot{\mathbf{x}}\sin(\alpha) + \ddot{\mathbf{x}}(1-\cos(\alpha))) > 0$. ∎

Then, we show that $\{\check{\alpha}_k\}$ is bounded below and away from zero.

**Lemma 4.11** *If $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$ and Assumptions (B1)-(B4) hold, then the sequence $\{\check{\alpha}_k\}$ is bounded below and away from zero. More precisely, let $\epsilon > 0$, $\sigma \geq \bar{\sigma} \geq 0$, and $\delta_2$ satisfy $\frac{\phi'(0)^2}{2L} \geq \frac{2\epsilon^2(1-\sigma)^2}{L} \geq \delta_2 > 0$. Then, Condition (37) will hold for $\check{\alpha}_k \geq \frac{2\epsilon(1-\sigma)}{L}$.*

**Proof:** Using (25), for the sake of brevity, we will use $\phi(\alpha)$ for $\phi(\mathbf{v}(\alpha))$ and use $\phi(0)$ for $\phi(\mathbf{v}^k)$, respectively. Using the mean-value theorem, Assumption (B4), and (32), we have

$$\begin{aligned}
\phi(\alpha) &= \phi(0) + \int_0^1 \phi'(t\alpha)dt\alpha \\
&= \phi(0) + \int_0^1 [\phi'(t\alpha) - \phi'(0) + \phi'(0)]dt\alpha \\
&= \phi(0) + \phi'(0)\alpha t|_0^1 + \int_0^1 [\phi'(t\alpha) - \phi'(0)]dt\alpha \\
&\leq \phi(0) + \phi'(0)\alpha + \int_0^1 L\alpha^2 t dt \\
&= \phi(0) + \phi'(0)\alpha + \frac{1}{2}L\alpha^2 t^2|_{t=0}^1 \\
&= \phi(0) + \phi'(0)\alpha + \frac{1}{2}L\alpha^2.
\end{aligned} \tag{46}$$

In order for $\phi(\alpha) \leq \phi(0) - \delta_2$, it may require that $\phi'(0)\alpha + \frac{1}{2}L\alpha^2 \leq -\delta_2$, which will hold for $\alpha \in \left[\frac{-\phi'(0)-\sqrt{\phi'(0)^2-2L\delta_2}}{L}, \frac{-\phi'(0)+\sqrt{\phi'(0)^2-2L\delta_2}}{L}\right]$. Multiplying $-1$ on both sides of (33) yields $-\phi'(0) \geq 2\phi(0)(1-\sigma) = 2\phi(\mathbf{v}^k)(1-\sigma) \geq 2\epsilon(1-\sigma)$. Therefore, assuming that we have selected $\delta_2 \leq \frac{\phi'(0)^2}{2L}$, we have

$$\begin{aligned} \check{\alpha}_k &\geq \frac{-\phi'(0) + \sqrt{\phi'(0)^2 - 2L\delta_2}}{L} \\ &\geq \frac{-\phi'(0)}{L} \\ &\geq \frac{2\phi(\mathbf{v}^k)(1-\sigma)}{L} \\ &\geq \frac{2\epsilon(1-\sigma)}{L} \end{aligned} \tag{47}$$

is bounded below and away from zero because $(1-\sigma)$ is bounded below and away from zero, and $L$ is bounded. ∎

The following Lemma is used to show in Lemma 4.13 that $\{\hat{\alpha}_k\}$ is bounded below and away from zero.

**Lemma 4.12 ([45])** *Assume that $\mathbf{v}^k$ is the current point, $\dot{\mathbf{v}}$ satisfies (13), and $\ddot{\mathbf{v}}$ satisfies (16) or (23). Let $\mathbf{v}(\alpha)$ be computed by using (25). Then,*

$$\begin{aligned} z_i(\alpha)s_i(\alpha) &= z_i s_i(1-\sin(\alpha)) + \sigma\mu\sin(\alpha) - (\dot{z}_i\ddot{s}_i + \ddot{z}_i\dot{s}_i)\sin(\alpha)(1-\cos(\alpha)) \\ &\quad +(\ddot{z}_i\ddot{s}_i - \dot{z}_i\dot{s}_i)(1-\cos(\alpha))^2. \end{aligned} \tag{48}$$

**Proof:** Using the last rows of (15) and (16), we have

$$\begin{aligned} z_i(\alpha)s_i(\alpha) &= [z_i - \dot{z}_i\sin(\alpha) + \ddot{z}_i(1-\cos(\alpha))][s_i - \dot{s}_i\sin(\alpha) + \ddot{s}_i(1-\cos(\alpha))] \\ &= z_i s_i - (\dot{z}_i s_i + z_i\dot{s}_i)\sin(\alpha) + (\ddot{z}_i s_i + z_i\ddot{s}_i)(1-\cos(\alpha)) + \dot{z}_i\dot{s}_i\sin^2(\alpha) \\ &\quad -(\dot{z}_i\ddot{s}_i + \ddot{z}_i\dot{s}_i)\sin(\alpha)(1-\cos(\alpha)) + \ddot{z}_i\ddot{s}_i(1-\cos(\alpha))^2 \\ &= z_i s_i(1-\sin(\alpha)) + \sigma\mu\sin(\alpha) - 2\dot{z}_i\dot{s}_i(1-\cos(\alpha)) + \dot{z}_i\dot{s}_i\sin^2(\alpha) \\ &\quad -(\dot{z}_i\ddot{s}_i + \ddot{z}_i\dot{s}_i)\sin(\alpha)(1-\cos(\alpha)) + \ddot{z}_i\ddot{s}_i(1-\cos(\alpha))^2 \\ &= z_i s_i(1-\sin(\alpha)) + \sigma\mu\sin(\alpha) + \dot{z}_i\dot{s}_i(\sin^2(\alpha) + 2\cos(\alpha) - 2) \\ &\quad -(\dot{z}_i\ddot{s}_i + \ddot{z}_i\dot{s}_i)\sin(\alpha)(1-\cos(\alpha)) + \ddot{z}_i\ddot{s}_i(1-\cos(\alpha))^2. \end{aligned}$$

Substituting $\sin^2(\alpha)+2\cos(\alpha)-2 = -1+2\cos(\alpha)-\cos^2(\alpha) = -(1-\cos(\alpha))^2$ into the last equation gives (48). Since the last rows of (16) and (23) are the same, this conclusion applies to Algorithms 3.1, 3.2, and 3.3. ∎

**Lemma 4.13 ([41])** *Assume that (B1)-(B4) hold. If $\{\mathbf{v}^k\} \subset \Omega(\epsilon)$ for some $\epsilon > 0$, then $\{\hat{\alpha}_k\}$ is bounded below and away from zero.*

**Proof:** For each $k$, find $i$ such that $z_i^k s_i^k = \min(\mathbf{Z}^k\mathbf{s}^k)$, and let $\eta_1^k = \dot{z}_i^k\ddot{s}_i^k + \ddot{z}_i^k\dot{s}_i^k$ and $\eta_2^k = \ddot{z}_i^k\ddot{s}_i^k - \dot{z}_i^k\dot{s}_i^k$. Since $\{\dot{\mathbf{v}}^k\}$ and $\{\ddot{\mathbf{v}}^k\}$ are bounded according to Lemma 4.8, the sequences $\{|\eta_1^k|\}$ and $\{|\eta_2^k|\}$ are also bounded. The proof is based on induction. For $k = 1$, from (30) and (48), we have

$$\begin{aligned}
& \min(\mathbf{Z}^1\mathbf{s}^1) - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^1)}{\phi(\mathbf{v}^0)} \\
\geq\ & z_i^1 s_i^1 - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)[1 - 2\rho(1-\sigma_0)\sin(\alpha_0)] \\
\geq\ & z_i^0 s_i^0(1-\sin(\alpha_0)) + \sigma_0\mu_0\sin(\alpha_0) - \eta_1^0\sin(\alpha_0)(1-\cos(\alpha_0)) + \eta_2^0(1-\cos(\alpha_0))^2 \\
& -\frac{1}{2}(z_i^0 s_i^0)[1 - 2\rho(1-\sigma_0)\sin(\alpha_0)] \\
=\ & \frac{1}{2}z_i^0 s_i^0 - z_i^0 s_i^0\sin(\alpha_0) + \sigma_0\mu_0\sin(\alpha_0) - \eta_1^0\sin(\alpha_0)(1-\cos(\alpha_0)) \\
& +\eta_2^0(1-\cos(\alpha_0))^2 + z_i^0 s_i^0\rho(1-\sigma_0)\sin(\alpha_0). \qquad (49)
\end{aligned}$$

Since $\mathbf{v}^k \in \Omega(\epsilon)$, we know $z_i^k s_i^k \geq \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^k)}{\phi(\mathbf{v}^0)} \geq \frac{1}{2}\min(\mathbf{Z}^0 s^0)\frac{\epsilon}{\phi(\mathbf{v}^0)} > 0$, therefore there must be $\alpha_0 > 0$ such that the last express in (49) is greater than zero. Next, for $k > 1$, assume that there exists $\alpha_{k-1} > 0$ such that

$$\min(\mathbf{Z}^k\mathbf{s}^k) - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^k)}{\phi(\mathbf{v}^0)} > 0, \qquad (50)$$

then we show that there exists $\alpha_k > 0$ such that

$$\min(\mathbf{Z}^{k+1}\mathbf{s}^{k+1}) - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^{k+1})}{\phi(\mathbf{v}^0)} > 0.$$

From (30) and (48), it follows

$$\begin{aligned}
& \min(\mathbf{Z}^{k+1}\mathbf{s}^{k+1}) - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^{k+1})}{\phi(\mathbf{v}^0)} \\
\geq\ & z_i^{k+1} s_i^{k+1} - \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^k)}{\phi(\mathbf{v}^0)}[1 - 2\rho(1-\sigma_k)\sin(\alpha_k)] \\
\geq\ & z_i^k s_i^k(1-\sin(\alpha_k)) + \sigma_k\mu_k\sin(\alpha_k) - \eta_1^k\sin(\alpha_k)(1-\cos(\alpha_k)) + \eta_2^k(1-\cos(\alpha_k))^2 \\
& -\frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\phi(\mathbf{v}^k)}{\phi(\mathbf{v}^0)}[1 - 2\rho(1-\sigma_k)\sin(\alpha_k)] \qquad (51)
\end{aligned}$$

Since $z_i^k s_i^k \geq \min(\mathbf{Z}^k\mathbf{s}^k) \geq \frac{1}{2}\min(\mathbf{Z}^0\mathbf{s}^0)\frac{\epsilon}{\phi(\mathbf{v}^0)} > 0$ and (50), we can find $\alpha_k > 0$ such that the last express in (51) is greater than zero.

We know that $\{z_i^k s_i^k\}$ is bounded below and away from zero, $\{\sigma_k\} \geq 0$, and $\{|\eta_1^k|\}$ and $\{|\eta_2^k|\}$ are bounded due to Lemma 4.8. Therefore, $\{\hat{\alpha}_k\}$ is bounded below and away from zero. ∎

From Lemmas 4.9, 4.10, 4.11, 4.13, we immediately have the following result.

**Lemma 4.14** *Assume (B1)-(B4) hold, then for Algorithms 3.1, 3.2, and 3.3, $\{\alpha_k\}$ is bounded below and away from zero.*

**Proof:** For Algorithms 3.1, 3.2, and 3.3, since $\tilde{\alpha}_k$, $\bar{\alpha}_k$, $\check{\alpha}_k$, and $\hat{\alpha}_k$ are all bounded below and away from zero, $\alpha_k = \min\{\tilde{\alpha}_k, \bar{\alpha}_k, \check{\alpha}_k, \hat{\alpha}_k\}$ is bounded below and away from zero. This completes the proof. ∎

We are now ready to prove the convergence for all three Algorithms.

**Theorem 4.1** *Assume (B1)-(B4) hold. Then, for Algorithms 3.1, 3.2, and 3.3, we have (i) for all $k \geq 0$, the sequence $\{\phi(\mathbf{v}^k)\}$ decreases faster than a constant rate, and (ii) all three algorithms terminate in finite iterations.*

**Proof:** Since $\{\alpha_k\}$ is bounded below and away from zero, there must be an $\alpha^* > 0$ such that $\alpha_k \geq \alpha^* = \min\{\tilde{\alpha}_k, \bar{\alpha}_k, \check{\alpha}_k, \hat{\alpha}_k\} > 0$ for Algorithm 3.1, 3.2, and 3.3. Since $\alpha^*$ is bounded below and away from zeros, this shows that $\{\phi(\mathbf{v}^k)\}$ decreases faster than a constant rate due to (30). Since $\phi(\mathbf{v}^0)$ is bounded, and $\{\phi(\mathbf{v}^k)\}$ decreases faster than a constant rate, it needs only finite iterations $K$ to find $\phi(\mathbf{v}^K) \leq \epsilon$ with $(\mathbf{w}^K, \mathbf{s}^K, \mathbf{z}^K) \in \mathbb{R}^{3p}_+$. ∎

# 5 Preliminary test results

The benefit of using arc-search can easily be seen from Problem 19 (HS19) in [12] which is given as follows:

$$\begin{array}{rcl} \min & : & (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{s.t.} & : & (x_1 - 5)^2 + (x_2 - 5)^2 - 100 \geq 0, \\ & & -(x_2 - 5)^2 - (x_1 - 6)^2 + 82.81 \geq 0, \\ & & 13 \leq x_1 \leq 100, \\ & & 0 \leq x_2 \leq 100. \end{array} \tag{52}$$

Figure 1(a) shows the real constraint set of HS19 which is between two curves in red lines. The contour lines describe the level of the objective function. The optimal solution is at the intersection of the two red curves marked by 'x'. Figure 1(b) shows that the ellipse constructed by using (25) passes the current iterate marked by a red circle 'o'. Clearly, searching along the ellipse is better than searching any straight line for this nonlinear constraint problem. Figure 1(c) shows the iterates that approach to the optimal solution.

## 5.1 Two simple but hard problems

In [18], two simple and hard problems are tested. To demonstrate the efficiency of the proposed algorithm, we tested the algorithm against these problems. The first one was extensively studied in [36].

$$\begin{array}{rcl} \min_{x_1,x_2,x_3} & : & x_1 \\ \text{s.t.} & : & x_1^2 - x_2 - 1 = 0, \\ & & x_1 - x_3 - 2 = 0, \\ & & x_2 \geq 0, \quad x_3 \geq 0. \end{array} \tag{53}$$

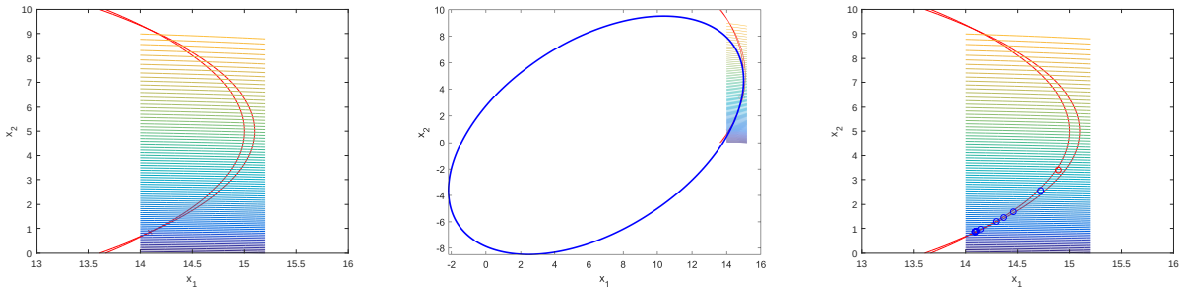

Figure 1: (a) The constraint set of problem HS19. (b) Ellipse approximation of the central-path of problem HS19. (c) Ellipse iterates of the central-path of problem HS19.

Table 1: Test results for a subset of problems in [12]

| | Agorithm 3.3 | | | |
|---|---|---|---|---|
| Prob | Obj | Iter | Seconds | Conv $\phi$ |
| 16 | 0.25 | 22 | 0.231982 | 1.2313e-15 |
| 17 | 1 | 22 | 0.229433 | 1.3279e-15 |
| 19 | -6961.8139 | 22 | 0.206065 | 7.563e-11 |
| 23 | 2 | 22 | 0.253886 | 3.0712e-13 |
| 32 | 1 | 22 | 0.273155 | 7.6672e-16 |
| 59 | -7.8028 | 24 | 0.253580 | 2.4705e-12 |
| 64 | 6299.8424 | 19 | 0.167498 | 1.139e-10 |
| 66 | 0.51816 | 22 | 0.275467 | 1.5841e-12 |
| 71 | 17.014 | 37 | 0.576570 | 3.1672e-13 |
| 80 | 0.05395 | 20 | 0.432452 | 5.7296e-15 |
| 84 | -5280335.2971 | 27 | 0.689569 | 6.1572e-05 |
| 95 | 0.015621 | 23 | 0.656584 | 4.4154e-13 |
| 96 | 0.015621 | 20 | 0.520563 | 3.6668e-13 |
| 97 | 4.6451 | 25 | 0.662390 | 2.6823e-11 |
| 98 | 4.6451 | 26 | 0.793426 | 6.9447e-12 |
| 101 | 1809.7648 | 53 | 10.802482 | 1.5096e-09 |
| 108 | -0.86603 | 22 | 1.674474 | 1.1108e-15 |

It is showed in [36] that many interior-point algorims using line search fail to solve this problem because the iterates is confined to a region of an inconsistent system of equations and inequalities. It is easy to check that Condition (B3) is violated for this problem. However, a simple implementation trick (which will be discussed in a follow-on paper) will get rid of the difficulty. As a matter of fact, the algorithm converges to the optimal solution $(2, 3, 0)$ in 39 iterations from the standard initial condition (-4, 1,1 ).

The second example is a standard test problem taken from [12, Problem 13]:

$$\begin{array}{rcl} \min_{x_1,x_2} & : & (x_1-2)^2+x_2^2 \\ \text{s.t.} & : & (1-x_1)^3-x_2\geq 0, \\ & & x_1\geq 0, \quad x_2\geq 0. \end{array} \tag{54}$$

This problem was not solved in [31, 40]. Starting from the standard initial point (-2,-2), the proposed algorithm terminates at (0.9997,0) after 25 iterations, which is closer to the optimal solution (1,0) than the result of (0.9905,-0.0000) given in [18].

## 5.2 Test on some unsolvable problems by [41]

We tested Algorithm 3.3 on a subset of widely used benchmark problems (see Table 1) in [12]. It is worthwhile to note that the algorithm proposed in [41] failed to solve this subset of problems. But

Algorithm 3.3 solves all these problems. The test result for Algorithm 3.3 is summarized in Table 1.

The preliminary test result for these benchmark problems shows that (a) the newly developed algorithms are likely more robust than the one in [41], and (b) it is very important to appropriately handle the ill-conditioned matrix (13) (this will be discussed in detail in a follow-on paper that focuses on the implementation and will report more extensive test results).

The proposed algorithm is also tested for some spacecraft trajectory optimization problems [50] and solves a spacecraft formation flying orbit design problem [49].

# 6 Conclusions

In this paper, we proposed an arc-search interior-point algorithm for the nonlinear optimization problem. The algorithm searches the optimizer along an arc that is part of an ellipse instead of a traditional straight line. Since the arc stays in the interior set longer than any straight line, the arc step is longer than a line step, thereby producing a better iterate and improving the efficiency of the algorithm. Convergence of the algorithm is established under some mild conditions. Preliminary test results show that the algorithm is very promising. A follow-on paper is under preparation to provide details of the algorithm implementation and to report extensive test result.

# 7 Acknowledgment

This research is partially supported by NASA's IRAD 2023 fund SSMX22023D. The author is grateful to the anonymous reviewers and Dr. Robert Pritchett at Goddard Space Fight Center of NASA for their valuable comments that helped the author to improve the presentation of the paper.

# 8 Declarations

## 8.1 Conflict of interests

Author declares no conflict of interests.

## 8.2 Data availability

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

# References

[1] F. Alizadeh, Combinatorial optimization with interior-point methods and semi-definite matrices, Ph.D. thesis, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1993.

[2] H. Y. Benson and D. F. Shanno, Interior-point methods for nonconvex nonlinear programming: regularization and warmstarts, Comput. Optim. Appl. 40, no. 2, 143-189, 2008.

[3] R.H. Byrd, M.E., Hribar, and J. Nocedal, An interior point algorithm for large-scale nonlinear programming, SIAM J. Optimization 9, 877–900, 1999.

[4] R.H. Byrd, J.C. Gilbert, J. Nocedal, A trust region method based on interior point techniques for nonlinear programming, Mathematical Programming. 89, 149-185, 2000.

[5] R. H. Byrd, M. Marazzi, and J. Nocedal, On the convergence of Newton iterations to nonstationary points, Math. Program. 99, no. 1, Ser. A, 127-148, 2004.

[6] F. E. Curtis, A penalty-interior-point algorithm for nonlinear constrained optimization, Mathematical Programming Computation, 4, pp. 181-209, 2012.

[7] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, On the formulation and theory of the Newton interior-point method for nonlinear programming, Journal of Optimization Theory and Applications, 89, pp. 507-541, 1996.

[8] A. Forsgren, P. E. Gill, and M. H. Wright, Interior methods for nonlinear optimization, SIAM Review, 44, pp. 525-597, 2002.

[9] P. E. Gill and D. P. Robinson, A primal-dual augmented Lagrangian, Comput. Optim. Appl. 51, no. 1, 1-25, 2012.

[10] G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins Studies in Mathematical Sciences, 3rd Edition, 1996.

[11] N. I.M. Gould, D. Orban, and P. L. Toint, CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, Computational Optimization and Applications, 60(3), 545-557, 2015.

[12] W. Hock and K. Schittkowski, Test examples for nonlinear programming codes, in Lecture Notes in Economics and Mathematical Systems, volume 187, Springer, 1981.

[13] E. Iida and M. Yamashita, An infeasible interior-point arc-search method with Nesterov's restarting strategy for linear programming problems, Computational Optimization and Applications, 2024.

[14] B. Kheirfam, An arc-search infeasible interior-point algorithm for horizontal linear complementarity problem in the $N^{-\infty}$ neighbourhood of the central path, International Journal of Computer Mathematics, 94, pp. 2271-2282, 2017.

[15] B. Kheirfam, A polynomial-iteration infeasible interior-point algorithm with arc-search for semidefinite optimization, Journal of Scientific Computing, DOI: 10.1007/s10915-021-01609-6, 2021.

[16] M. Kojima, S. Mizuno, and A. Yoshise, A polynomial-time algorithm for a class of linear complementarity problem, Mathematical Programming, 44, pp. 1039-1091, 1989.

[17] J. Laurent-Varin, J. F. Bonnans, N. Bérend, M. Haddou, and C. Talbot, Interior-Point Approach to Trajectory Optimization, Journal of Guidance, Control, and Dynamics, 30(5), 1228-1238, 2007.

[18] Xin-Wei Liu and Yu-Hong Dai, A globally convergent primal-dual interior-point relaxation method for nonlinear programs, Mathematics of Computation, 89, 1301-1329, 2020.

[19] X. Liu and J. Sun, A robust primal-dual interior-point algorithm for nonlinear programs, SIAM J. Optim. 14, no. 4, 1163–1186, 2004.

[20] T.T. Lu and S.H. Shiou, Inverses of $2 \times 2$ Block Matrices, Computers and Mathematics with Applications, 43, 119-129, 2002.

[21] D. Luenberger, Optimization by Vector Space Methods. New York, John Wiley & Sons. pp. 240–242, 1969.

[22] S. Mehrotra, On the implementation of a primal-dual interior point method. SIAM journal on Optimization 2, 575–601,1992.

[23] R. D. C. Monteiro, A globally convergent primal—dual interior point algorithm for convex programming, Mathematical Programming 64, pp. 123–147,1994.

[24] R. Monteiro and I. Adler, Interior path following primal-dual algorithms. Part II: convex quadratic programming, Mathematical Programming, 44, pp. 43-66, 1989.

[25] W. Murray and M. H. Wright, Line search procedures for the Logarithmic barrier function, SIAM Journal on Optimization, 4 (1994), pp. 229–246.

[26] J. Nocedal, A. Wachter, and R.A. Waltz, Adaptive barrier update strategies for nonlinear interior methods, SIAM J. Optimization, 19, 1674–1693, 2009.

[27] J. Nocedal and S. J. Wright, Numerical Optimization, Springer, New York, 2006.

[28] M. A. Patterson and A. V. Rao, GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming, ACM Transactions on Mathematical Software, 41(1), Article 1, October 2014.

[29] T. Plantenga, A trust region method for nonlinear programming based on primal interior-point techniques, SIAM journal on Scientific Computing, 20, 282-305, 1998.

[30] Siegfried M. Rump, INTLAB - INTerval LABoratory, https://www.tuhh.de/ti3/rump/intlab/, 2023.

[31] D. F. Shanno and R. J. Vanderbei, Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods, Math. Program. 87(2), Ser. B, 303–316, 2000.

[32] A.L. Tits, A. Wachter, S. Bakhtiarl, T.J. Urban, and C.T. Lawrence, A primal-dual method for nonlinear programming with strong global and local convergence properties, SIAM Journal on Optimization, 14(1), 173-199, 2003.

[33] M. J. Todd, The many facets of linear programming, Math. Program., Ser. B, 91, pp. 417-436, 2002.

[34] M. Ulbrich, S. Ulbrich, and L.N. Vicente, A globally convergent primal-dual interior-point filter method for nonlinear programming, Mathematical Programming, 100, 379-410, 2004.

[35] R. Vanderbei and D. Shanno, An interior-point algorithm for nonconvex nonlinear programming, Comput. Optimization Appl. 13, 231-252, 1999.

[36] A. Wächter and L. T. Biegler, Failure of global convergence for a class of interior point methods for nonlinear programming, Mathematical Program. 88(3), Ser. A, 565–574, 2000.

[37] A. Wächter and L. T. Biegler, Line search filter methods for nonlinear programming: motivation and global convergence, SIAM Journal on Optimization 16(1), 1-31, 2005.

[38] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Program., Ser. A 106, 25-57, 2006.

[39] S. Wright, Primal-Dual Interior-Point Methods, SIAM, Philadelphia, 1997.

[40] H. Yamashita, A globally convergent primal-dual interior point method for constrained optimization, Optim. Methods Softw. 10(2), 443–469, 1998.

[41] M. Yamashita, E. Iida, and Y. Yang An infeasible interior-point arc-search algorithm for nonlinear constrained optimization, Numerical Algorithms, 89, 249-275, 2018.

[42] X. Yang, H. Liu, and Y. Zhang, An arc-search infeasible-interior-point method for symmetric optimization in a wide neighborhood of the central path, Optimization Letters, 11, pp. 135-152, 2017.

[43] Y. Yang, A polynomial arc-search interior-point algorithm for convex quadratic programming, European Journal of Operational Research, 215, 25-38, 2011.

[44] Y. Yang, A polynomial arc-search interior-point algorithm for linear programming, Journal of Optimization Theory and Applications 158, 859-873, 2013.

[45] Y. Yang, CurveLP-A MATLAB implementation of an infeasible interior-point algorithm for linear programming, Numerical Algorithms, 74, 967-996, 2017.

[46] Y. Yang, Two computationally efficient polynomial-iteration infeasible interior-point algorithms for linear programming, Numerical Algorithms, 79, 957-992, 2018.

[47] Y. Yang, Arc-search techniques for interior-point methods, CRC Press, Baco Raton, 2020.

[48] Y. Yang, A polynomial time infeasible interior-point arc-search algorithm for convex optimization, Optimization and Engineering, 24, pp. 885-914, 2023.

[49] Y. Yang, On optimal LISA orbit design, arXiv:2501.00936 [gr-qc], 2025.

[50] Y. Yang, R. Pritchett, and N. Hatten, An infeasible interior-point arc-search algorithm for spacecraft trajectory optimization, 2024 INFORMS Optimization Society Conference, Houston, March 22-24, 2024. Available on https://ntrs.nasa.gov/citations/20240002037.

[51] E. A. Yildirim and S. J. Wright, Warm-start strategies in interior-point methods for linear programming, SIAM Journal on Optimization, 12(3), pp. 782–810, 2002.

[52] M. Zhang, B. Yuan, Y. Zhou, X. Luo, and Z. Huang, A primal-dual interior-point algorithm with arc-search for semidefinite programming, Optimization Letters, 13, pp. 1157-1175, 2019.

[53] M. Zhang, K. Huang, and Y. Lv, A wide neighborhood arc-search interior-point algorithm for convex quadratic programming with box constraints and linear constraints, Optimization and Engineering,23, pp. 1117-1137, 2022.