Applied and Computational Mathematics
2021; 10(6): 146-155
http://www.sciencepublishinggroup.com/j/acm
doi: 10.11648/j.acm.20211006.13
ISSN: 2328-5605 (Print); ISSN: 2328-5613 (Online)

A Weighted Analytic Center for Second-Order Cone Constraints

Bamanga Dawuda, Shafiu Jibrin*, Ibrahim Abdullahi

Department of Mathematics, Federal University, Dutse, Nigeria

Email address:
shafiuj@fud.edu.ng (S. Jibrin)
*Corresponding author

To cite this article:
Bamanga Dawuda, Shafiu Jibrin, Ibrahim Abdullahi. A Weighted Analytic Center for Second-Order Cone Constraints. *Applied and Computational Mathematics*. Vol. 10, No. 6, 2021, pp. 146-155. doi: 10.11648/j.acm.20211006.13

Received: October 18, 2021; **Accepted**: November 15, 2021; **Published**: November 27, 2021

Abstract: This paper introduces a weighted analytic center for a system of second order cone constraints. The associated barrier function is shown to be convex and conjugate gradient (CG) methods are used to compute the weighted analytic center. In contrast with Newton's-like methods, CG methods use only the gradient and not the Hessian to minimize a function. The methods considered are the HPRP and ZA with exact and inexact line searches. The exact line search uses Newton's method and quadratic interpolation is used for the inexact line search. The performance of each method on random test problems was evaluated by observing the number of iterations and time required to find the weighted analytic center. Our numerical methods indicate that ZA is better than HPRP with any of the two line searches, in terms of the number of iterations and time to find the weighted analytic center. Quadratic interpolation inexact line search gives the best success rate and fewest number of iterations for the CG methods considered. On the other hand, the fastest time for the CG methods is found with the Newton's exact line search. In addition, these results indicate that for each of the methods, our Quadratic interpolation inexact line search has a higher cost per iteration than that of the Newton's exact line search.

Keywords: Second-Order Cone Constraints, Weighted Analytic Center, Conjugate Gradient Methods, Interior-point Methods

1. Introduction

Consider a system of second-order cone constraints of the form:

$$c_j^T x + d_j - \| A_j x + b_j \| \geq 0 \quad (j=1,2,...,q), \tag{1}$$

$x \in \mathbb{R}^n$, A_j is an $m_j \times n$ matrix, $b_j \in \mathbb{R}^{m_j}$, $c_j \in \mathbb{R}^n$, and $d_j \in \mathbb{R}$. The norm is the standard Euclidean norm. Let

$$\mathcal{R} = \left\{ x \in \mathbb{R}^n \mid c_j^T x + d_j - \| A_j x + b_j \| \geq 0, j = 1, 2, \ldots, q \right\} \tag{2}$$

denotes the feasible region defined by the SOC constraints in (1). Assume $\mathcal{R}$ is bounded and that it has a non-empty interior.

Second order cone (SOC) constraints are important to study because there exist many optimization problems where the constraints can be written in this form. For instance, SOC constraints can be used to easily represent problems dealing with norms and hyperbolic constraints. An optimization problem where the constraints are SOCs is called a second-order cone programming problem. There are also a huge number of real world applications of SOCs in areas such as antenna array weight design, grasping force optimization, and portfolio optimization [29, 26]. Furthermore, there exist efficient interior point methods for solving second-order cone programming problems [29, 9, 19]. The study of second-order cone programming has continued to be of interest [12, 17, 33, 31].

This work extends the notion of weighted analytic center from linear programming (LP) to SOC constraints. This extension differs from that given for linear matrix inequality constraints [22]. In the special case of linear constraints, the weighted analytic center has been studied extensively in the past [3, 8]. The study of weighted analytic center for second-order cone constraints is of interest in its own right. Some algorithms in linear programming, Quadratic Programming and semidefinite programming are based on weighted analytic centers [25, 7, 18].

Weighted analytic center for SOCs can be found using the Standard Newton's method by minimizing the related barrier function. This approach has the drawback that both the gradient and the Hessian of the barrier function are required at each iterate. Also, Newton's method may not work well when some of the weights are relatively very large relative to the other weights as the hessian matrix becomes ill-conditioned. This study considers computing the weighted analytic center using conjugate gradient (CG) methods.

CG methods are popular for solving large scale non-linear unconstrained optimization problems. They are characterized by strong global convergence properties and low memory requirement [1]. CG methods use only the gradient of the barrier function. They do not need the hessian of the barrier function. Our work uses the ZA CG method given by Salleh et al [24], which is a modification of the classical HS method [11]. The ZA method is compared with the HPRP method presented by Alhawarat et al [2]. HPRP is a modification of the classical PRP method [20]. Both ZA and HPRP are implemented using the Newton's exact line search and Quadratic interpolation inexact line search [14, 5]. The performance of each method on random test problems will be evaluated by observing the number of iterations and time required to find the weighted analytic center.

Our results indicate that ZA is better than HPRP in terms of the number of iterations and time to find the weighted analytic center. Quadratic interpolation line search gives the best success rate and fewest number of iterations for the CG methods considered. On the other hand, the fastest time for the CG methods is found with the Newton's exact line search.

2. Weighted Analytic Center for Second Order Cone Constraints

This section introduces weighted analytic center for the system of second order cone constraints (1).

Consider a weight vector $\omega=(\omega_1,\omega_2,\ldots,\omega_q)$, $\omega\in\mathbb{R}^q$ and the feasible region $\mathcal{R}$ (2). The interior of R is given by

$$int(\mathcal{R}) = \left\{x \in \mathbb{R}^n \mid c_j^T x + d_j - \| A_j x + b_j \| > 0, j = 1,2,\ldots,q\right\} \tag{3}$$

The weighted analytic center for the system (1) is given by

$$x_{ac}(\omega) = argmin\left\{\phi_\omega(x) \mid x \in int(\mathcal{R})\right\}, \tag{4}$$

where $\phi_\omega : int(\mathcal{R}) \to \mathbb{R}$ is a barrier function over the interior of R defined by:

$$\phi_\omega(x) = -\sum_{j=1}^{q} \omega_j log(c_j^T x + d_j - \| A_j x + b_j \|). \tag{5}$$

The weighted analytic center is unique when $\phi_\omega(x)$ is strictly convex over the interior of R. Our definition of weighted analytic center extends the one for linear constraints [3, 8]. Note that the barrier function $\phi_\omega(x)$ becomes infinitely large as x approaches the boundary of the feasible region from the interior. Hence, the weighted analytic center is an interior point.

Theorem 2.1 $\phi_\omega(x)$ is convex over $int(\mathcal{R})$.

Proof: For convenience, let $f(x)=c^T x+d-\| Ax+b \|$. Then, $f(x)>0$ over $int(\mathcal{R})$ and $f(x)$ is concave over $\mathbb{R}^n$ [30]. Now, since *log* is an increasing function and concave over $\mathbb{R}_+$, then $log(c^T x+d-\| Ax+b\|)$ is concave over $int(\mathcal{R})$ [27]. The sum of concave functions is concave and so, $\sum_{j=1}^{q} \omega_j log(c_j^T x + d_j - \| A_j x + b_j \|)$ is concave over $int(\mathcal{R})$. Hence, $\phi_\omega(x) = -\sum_{j=1}^{q} \omega_j log(c_j^T x + d_j - \| A_j x + b_j \|)$ is convex over $int(\mathcal{R})$.

Lemma 2.1 *Let* $f(x)=c^T x+d-\| Ax+b\|$. *The gradient and the Hessian of* $f(x)$ *are given by*

$$\nabla f(x) = c - \frac{A^T(Ax+b)}{\| Ax+b \|} \tag{6}$$

$$H_f(x) = \frac{A^T(Ax+b)(A^T(Ax+b))^T}{\| Ax+b \|^3} - \frac{A^T A}{\| Ax+b \|}. \tag{7}$$

Proof: For the gradient, use the fact that $f(x)=c^T x+d-\sqrt{(Ax+b)^T(Ax+b)}$. To get the Hessian, apply the quotient differentiation rule to the gradient function (6).

Theorem 2.2 *The gradient denoted by* $g(x)=\nabla\phi_\omega(x)$ *and the Hessian* $H_{\phi_\omega(x)}$ *of the barrier function* $\phi_\omega(x)$ *are given by*

$$g(x) = -\sum_{j=1}^{q} \omega_j \frac{1}{c_j^T x + d_j - \| A_j x + b_j \|}\left(c_j - \frac{A_j^T(A_j x + b_j)}{\| A_j x + b_j \|}\right) \tag{8}$$

$$H_{\phi_\omega(x)} = \sum_{j=1}^{q} \omega_j (B_j + u_j v_j^T), \tag{9}$$

where

$$B_j = -\frac{1}{c_j^T x + d_j - \| A_j x + b_j \|} \left(\frac{\| A_j x + b_j \|^2 (A_j^T A_j) - A_j^T (A_j x + b_j)(A_j^T (A_j x + b_j))^T}{\| A_j x + b_j \|^3} \right)$$

$$u_j = -\frac{1}{(c_j^T x + d_j - \| A_j x + b_j \|)^2} \left(c_j - \frac{A_j^T (A_j x + b_j)}{\| A_j x + b_j \|} \right)$$

$$v_j = c_j - \frac{A_j^T (A_j x + b_j)}{\| A_j x + b_j \|}$$

Proof: Let $f_j(x) = log(c_j^T x + d_j - \| A_j x + b_j \|)$. By Lemma 2.1 and using the composition of functions differentiation rule, the gradient of $f_j(x)$ is given by

$$\nabla f_j(x) = \frac{1}{c_j^T x + d_j - \| A_j x + b_j \|} \left(c_j - \frac{A_j^T (A_j x + b_j)}{\| A_j x + b_j \|} \right)$$

Now, let $h_1(x) = \log(x)$ and $h_2(x) = c_j^T x + d_j - \| A_j x + b_j \|$. Then $f_j(x) = (h_1 \circ h_2)(x)$ and $\frac{\partial h}{\partial x_i}(x) = h_1'(h_2(x)) \frac{\partial h_2}{\partial x_i}(x)$. So,

$$\frac{\partial^2 h}{\partial x_i \partial x_j}(x) = h_1'(h_2(x)) \frac{\partial^2 h_2}{\partial x_i \partial x_j}(x) + h_1''(h_2(x)) \frac{\partial h_2}{\partial x_i}(x) \frac{\partial h_2}{\partial x_j}(x).$$

Hence, using a result in the proof of Theorem 1 in Barta et al study [4], the Hessian of $f_j(x)$ is obtained as:

$$H_{f_j}(x) = B_j + u_j v_j^T,$$

where B_j, u_j, v_j are given by:

$$B_j = h_1'(h_2(x)) H_{h_2}(x)$$

$$u_j = h_1''(h_2(x)) \nabla h_2(x)$$

$$v_j = \nabla h_2(x)$$

By Lemma 2.1, the above gives B_j, u_j, v_j as defined in the theorem. The rest of the proof follows by extending the gradient and Hessian obtained above to sum of functions and their scalar product.

It can be seen from Theorem 2.2 that the gradient of the barrier function does not exist when $\| A_j x + b_j \| = 0$ for some *j*. But, in practice, one would only expect trouble if an iterate happens to be at a place of nondifferentiability. This is easy to address by simply randomizing the starting interior point. For example, one can use an iteration of any of the Monte Carlo methods applied to SOC constraints [15]. The result also shows that the computation of the Hessian of the barrier function is quite expensive. Hence, the result of Theorem 2.2 indicates that conjugate gradient (CG) methods would be a good choice in comparison with the standard Newton's method in finding the weighted analytic center.

3. HPRP and ZA Conjugate Gradient Methods for Computing Weighted Analytic Center

In this section, a brief description of the conjugate gradient methods is given. The discussion focuses on the HPRP and ZA methods used in our work.

Consider a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and the following unconstrained optimization problem

$$min\{f(x) : x \in \mathbb{R}^n\} \tag{10}$$

Let $g(x)$ denote the gradient of $f(x)$. A conjugate gradient method to find a solution to problem 3.1 works as follows. Given an initial guess $x_o \in \mathbb{R}^n$, a sequence $\{x_k\}$ is generated by:

$$x_{k+1} = x_k + \sigma_k s_k \tag{11}$$

and the direction s_k is defined as

$$s_{k+1} = \begin{cases} -g_k & if\ k = 0, \\ -g_{k+1} + \beta_k s_k & if\ k \geq 1, \end{cases} \tag{12}$$

where x_k is the current iterate, $g_k = g(x_k)$, β_k is the CG coefficient and $\sigma_k > 0$ is the step-length obtained by a line search. A common convergence criterion for a CG method is $\| g(x_k) \| \leq TOL$, where TOL is a given tolerance and $\|.\|$ denotes the Euclidean norm.

Over the years, a variety of CG formulas have been given, where the main differences is in the parameter β_k. The work by [10] discussed details on some CG methods with special emphasis on their global convergence. In recent times, research has focused on modified CG methods [32, 23]. The

PRP and HS classical CG methods are given in the Table 1.

Table 1. *PRP, HS and NPRP Methods.*

No.	β_k	Method name	References
1	$\frac{g_k^T(g_k-g_{k-1})}{\|\|g_{k-1}\|\|^2}$	PRP method	[20]
2	$\frac{g_k^T(g_k-g_{k-1})}{s_{k-1}^T(g_k-g_{k-1})}$	HS method	[11]
3	$\frac{\|\|g_k\|\|^2-\frac{\|\|g_k\|\|}{\|\|g_{k-1}\|\|}\|g_k^Tg_{k-1}\|}{\|\|g_{k-1}\|\|^2}$	NPRP method	[24]

A CG method is called globally convergent if $g_k = 0$ for some k or $\liminf_{k\to 0} g_k = 0$ [10]. It is said to have the jamming property if the method packs tightly into a specified space due to a very small step size σ_k. This results in the method taking infinitely many steps without converging. However, it is said to have a restart property if the method can jump out of the loop whenever jamming is encountered and take a bigger step size σ_k to continue the search for a minimum of the given function [10].

PRP is globally convergent when $f(x)$ is strongly convex and the line search is exact [20]. PRP and HS method with exact line search coincide and each is globally convergent if $x_{k+1} - x_k$. converges to 0 and g is Lipschitz continuous in a neighborhood of the level set $\mathcal{L} = \{x \in \mathbb{R}^n \mid g(x) \le g(x_0)\}$ [21]. Both PRP and HS have the restart property but may not be globally convergent with the strong Wolfe-Powell (SWP) line search.

A modification of the PRP method called the HPRP method is given by Alwaharat et al [2]. For this method, the parameter β_k is defined by

$$\beta_k^{HPRP} = \begin{cases} \beta_k^{PRP} & if \; \|g_k\|^2 > \|g_k^T g_{k-1}, \\ \beta_k^{NPRP} & otherwise, \end{cases} \tag{13}$$

HPRP has the restart property and globally convergent with SWP line search. Numerical results show that HPRP is almost better than other related PRP formulas in both the number of iterations and the CPU time [2]. The ZA method is a modification of HS given by Salleh et al [24]. Here, β_k is given by

$$\beta_k^{ZA} = \begin{cases} \dfrac{\|g_k\|^2 - g_k^T g_{k-1}}{s_{k-1}^T(g_k - g_{k-1})} & if \; \|g_k\|^2 > \|g_k^T g_{k-1}, \\ 0 & otherwise, \end{cases} \tag{14}$$

ZA is globally convergent with the SWP line search and it has the restart property. The numerical results indicate that ZA is better than HS, PRP, and HPRP [24].

We will apply HPRP and ZA to compute the weighted analytic center (4) for second-order constraints. Both exact line and Quadratic interpolation inexact line search will be used. The computational strengths of the methods is compared.

4. Quadratic Interpolation and Newton's Method Line Searches

This section describes exact and inexact stepsizes for our conjugate gradient methods. Newton's method is used to find the exact stepsize and inexact stepsize is found using Quadratic interpolation. The section also discusses convergence for the methods.

Let x_k be an interior point of $\mathcal{R}$ and s_k be a search direction vector. The following theorem shows how compute the distance from x_k to the boundary of the *jth* SOC in the direction s_k. The result will be used in our Quadratic interpolation line search.

Theorem 4.1 Suppose the ray $\{x_k + \sigma s_k \mid \sigma \ge 0\}$ intersects the boundary of $c_j^T x + d_j - \|A_j x + b_j\| \ge 0$ at the point $x_k + \sigma_+^{(j)} s_k$. Then, the distance $\sigma_+^{(j)}$ is the minimum positive value from the solution

$$\sigma_+^{(j)} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{15}$$

where

$$a = (c_j^T s_k)^2 - s_k^T A_j^T A_j s_k \tag{16}$$

$$b = 2c_j^T s_k (c_j^T x_k + d_j) - 2x_k^T A_j^T A_j s_k - 2s_k^T A_j^T b_j \tag{17}$$

$$c = (c_j^T x_k + d_j)^2 - x_k^T A_j^T A_j x_k - 2x_k^T A_j^T b_j - b_j^T b_j \tag{18}$$

Proof: By assumption $\sigma_+^{(j)}$ exists, is positive and it satisfies

$$c_j^T (x_k + \sigma_+^{(j)} s_k) + d_j - \|A_j x + b_j\| = 0$$

Then

$$(c_j^T x_k + \sigma_+^{(j)} c_j^T s_k + d_j)^2 = (A_j x + b_j)^T (A_j x + b_j)$$

Solve the quadratic equation for $\sigma_+^{(j)}$. Since

$c_j^T x + d_j - \| A_j x + b_j \| > 0$ in the interior of $\mathcal{R}$, then $\sigma_+^{(j)}$ is the smallest positive solution of 4.1.

Note that the solution (15) will have no positive values if $\{x_k + \sigma s_k \mid \sigma \geq 0\}$ does not intersect the boundary of $c_j^T x + d_j - \| A_j x + b_j \| \geq 0$. It follows from Theorem 4.1 that that the distance σ_+ from x_k to the boundary of the bounded feasible region $\mathcal{R}$ in the direction s_k is given by

$$\sigma_+ = \min\{\sigma_+^{(j)} \mid 1 \leq j \leq q\}, \tag{19}$$

where $\sigma_+^{(j)}$ is given by (15). Note that σ_+ exists since $\mathcal{R}$ is bounded and x_k is an interior point of $\mathcal{R}$.

Consider the objective barrier function $\phi_\omega(x)$ (5). Let s_k be a conjugate direction generated by the CG algorithm at the current iterate x_k. Let $h(\sigma) = \phi_\omega(x_k + \sigma s_k)$. The exact stepsize σ_k is given by

$$\sigma_k = argmin\{h(\sigma) \mid \sigma \geq 0\}. \tag{20}$$

The following result will be used in finding the stepsizes using Newton's method. First, note that

$$h(\sigma) = -\sum_{j=1}^{q} \omega_j log(c_j^T x + \sigma c_j^T s_k + d_j - \| A_j x + \sigma A_j s_k + b_j \|) \tag{21}$$

Lemma 4.1 Let x_k be a conjugate direction generated by the CG algorithm at the current *iterate* x_k. *Then,*

$$h'(\sigma) = -\sum_{j=1}^{q} \omega_j \frac{f_j}{e_j} \tag{22}$$

$$h''(\sigma) = \sum_{j=1}^{q} \omega_j \left(\left(\frac{f_j}{e_j} \right)^2 - \frac{g_j}{e_j} \right), \tag{23}$$

where,

$$e_j = c_j^T x_k + \sigma c_j^T s_k + d_j - \| A_j x_k + \sigma A_j s_k + b_j \|$$

$$f_j = c_j^T s_k - \frac{x_k^T A_j^T A_j s_k + \sigma s_k^T A_j^T A_j s_k + s_k^T A_j^T b_j}{\| A_j x_k + \sigma A_j s_k + b_j \|}$$

$$g_j = \frac{(x_k^T A_j^T A_j s_k + \sigma s_k^T A_j^T A_j s_k + s_k^T A_j^T b_j)^2}{\| A_j x_k + \sigma A_j s_k + b_j \|^3} - \frac{s_k^T A_j^T A_j s_k}{\| A_j x_k + \sigma A_j s_k + b_j \|}$$

Proof: Let

$$f(x) = log(c_j^T x + \sigma c_j^T s_k + d_j - \| A_j x + \sigma A_j s_k + b_j \|)$$

and

$$h_1(\sigma) = c_j^T x + \sigma c_j^T s_k + d_j - \| A_j x + \sigma A_j s_k + b_j \|$$

Let $e_j = h_1(\sigma)$, $f_j = h_1'(\sigma)$ and $g_j = h_1''(\sigma)$. Then

$$f'(\sigma) = \frac{h_1'(\sigma)}{h_1(\sigma)} = \frac{f_j}{e_j}$$

$$f''(\sigma) = \frac{h_1''(\sigma)}{h_1(\sigma)} - \frac{h_1'(\sigma)^2}{h_1(\sigma)^2} = \frac{g_j}{e_j} - \left(\frac{f_j}{e_j} \right)^2$$

The rest of the proof follows by extending the result obtained to sum of functions and their scalar product.

The following describes Quadratic interpolation line search for approximating the stepsize σ_k (20) in our CG algorithms [5].

Quadratic Interpolation

Step 1: Use (19) to find the distance σ_+ from x_k to the boundary of the bounded feasible region R in the direction s_k

Step 2: Set $\alpha_1 = 0$ and $\alpha_3 = \sigma_+$

Step 3: Consider $h(\alpha) = \phi_\omega(x_k + \alpha s_k)$

```
Repeat
    $$\alpha_3 = \alpha_3 / 2$$
Until $h(\alpha_3) < h(\alpha_1)$
Let $\alpha_2 = \alpha_3 / 2$
```

Step 4: Compute the zero of the quadratic polynomial $P(\alpha)$ passing through the points $(\alpha_1, P(\alpha_1))$, $(\alpha_2, P(\alpha_2))$ and $(\alpha_3, P(\alpha_3))$. The zero is given by

$$\alpha^* = \frac{1}{2}\left(\alpha_2 - \frac{h_1}{h_3}\right),$$

where

$$\mathrm{h}_1 = \frac{\mathrm{h}(\alpha_2) - \mathrm{h}(\alpha_1)}{\alpha_2 - \backslash \mathrm{al}_1}$$

$$h_2 = \frac{h(\alpha_3) - h(\alpha_2)}{\alpha_3 - \alpha_2}$$

$$h_3 = \frac{h_2 - h_1}{\alpha_3 - \alpha_1}.$$

```
Step 5:
if $\alpha_3 < \alpha^*$
    Set $\alpha_k = \alpha_3$
else
    Set $\alpha_k = \alpha^*$
end
```

5. Numerical Experiments

This section gives numerical experiments to compare HPRP and ZA methods. The methods use Newton's exact line search and the quadratic interpolation inexact line search.

Table 2 describes the 31 random test problems used for our numerical experiments. For each SOC test problem, integer values q and n are generated uniformly in the intervals [1, 6] and [2, 10] respectively. Then $m \in \mathbb{Z}^q$ is generated such that each entry m_i is uniform in the interval [1, 10]. After this, q SOCs of the form $c^T x + d - \| Ax + b \| \geq 0$ are generated using m and n, where each entry of A, b and c are randomly chosen from the standard normal distribution $N(0,1)$ while d is uniform in the interval $(0,1)$. The process ensures that only SOCs test problems where all the q SOCs are satisfied at the origin are chosen. Table 2 lists the SOC test problems. The second column of Table 2 gives the dimension n of the ambient space and the third column is the number q of constraints. The fourth column gives the vector m. The last column gives the weight vector ω used for each test problem. One random entry of ω is set at 100 and each other entry is an integer uniform in the interval [1, 10].

For each test problem and method, HPRP and ZA conjugate gradient methods are applied using a maximum of 100 iterations and a tolerance $TOL = 10^{-4}$. Each method is stopped after 100 iterations or if $\| g(x_k) \| \leq TOL$ and the number of iterations and the CPU time taken are recorded. All codes were written in MATLAB. We deliberately chose test problems in Table 2 and weights where both ZA and HPRP were successful with Newton's exact line search and Quadratic interpolation inexact line search in finding the weighted analytic center.

Table 3 compares ZA and HPRP methods using the quadratic interpolation line search. The number of iterations and time taken to find the weighted analytic center are given in the table. Table 4 compares ZA and HPRP methods using the Newton's exact line search. Let S be our set of solvers and F be the set of our test problems. Then, S = {ZA, HPRP} and F = {1, 2, 3,..., 31}. Let $t_{f,s}$ be the number of iterations (or CPU time) to solve problem f using solver s. For comparison, scale $t_{f,s}$ by the ratio

$$r_{f,s} = \frac{t_{f,s}}{min\{t_{f,s} \mid s \in \mathcal{S}\}}.$$

Now, as in [24], define the probability measure

$$P_s(t) = \frac{1}{n_f} size\{f \in \mathcal{F} \mid \log r_{f,s} \leq t\},$$

where n_f is the size of set $\mathcal{F}$. In our case, $n_f = 31$. Note that for solver s, $P_s(t)$ is the probability that its performance ratio $r_{f,s}$ on $\mathcal{F}$ is within the value e^t of the best possible ratio.

The graph of $P_s(t)$ is used to compare the performance profile of the methods on the test problem set F. Figure 1 gives the performance profile of HPRP vs ZA based on the number of iterations and quadratic interpolation line search. The performance profile of HPRP vs ZA based on the CPU time and quadratic interpolation line search is given in Figure 2. Figure 3 shows the performance profile of HPRP vs ZA based on the number of iterations and Newton's exact line search. The performance profile of HPRP vs ZA based on the CPU time and Newton's exact line search is given in Figure 4. It can be seen from the graphs Figure 1 - Figure 4 that ZA outperforms HPRP in both number of iterations and CPU time on our test problems.

Figure 5 and Figure 6 are given to investigate the effect of line search on the performance of ZA. Figure 5 compares ZA-Int (ZA with quadratic interpolation line search) with ZA-Exact (ZA with Newton's exact line search) on the number of iterations. Figure 6 compares ZA-Int with ZA-Exact on the CPU time. It is clear from Figure 5 that ZA-Int is better than ZA-Exact in terms of number of iterations. However, Figure 6 shows that ZA-Exact outperforms ZA-Int in CPU time to find the weighted analytic center. This is not surprising since the quadratic interpolation line search requires finding the boundary point along each direction of search. The results in Figure 5 and Figure 6 indicates that the interpolation line search is more expensive than the Newton's exact line search.

To compare the success rates of our methods in finding the weighted analytic, 100 random problems are generated as before. But, here the interest is in whether the method finds the weighted analytic center or not after 100 iterations and using $TOL = 10^{-4}$. The results are given in Table 5. It is observed that ZA with quadratic interpolation has the highest success rate and almost equals that of HPRP with quadratic interpolation. It is also seen that the success rate depends much more on the line search used and not on the two methods.

Table 2. *SOC Test Problems.*

Problem	*n*	*q*	*M*	*W*
1	3	4	[6, 10, 3, 6]	[10, 3, 100, 3]
2	2	6	[3, 7, 6, 5, 7, 5]	[9, 9, 6, 2, 100, 10]
3	9	4	[1, 4, 10, 9]	[7, 10, 7, 100]
4	4	3	[6, 9, 1]	[10, 100, 9]
5	10	6	[2, 4, 10, 8, 5, 2]	[10, 10, 10, 100, 5, 3]
6	5	1	[7]	100
7	4	4	[5, 1, 8, 4]	[5, 6, 100, 5]
8	4	2	[5, 6]	[10, 100]
9	4	5	[2, 3, 9, 4, 6]	[5, 9, 100, 4, 4]
10	6	5	[9, 7, 10, 2, 9]	[100, 4, 7, 7, 2]
11	4	2	[4, 5]	[7, 100]
12	6	3	[7, 6, 10]	[2, 10, 100]
13	2	3	[8, 4, 3]	[100, 6, 2]]
14	9	5	[10, 5, 6, 7, 10]	[8, 100, 10, 5, 1]
15	2	2	[7, 9]	[8, 100]
16	7	5	[10, 1, 10, 4, 1]	[100, 8, 7, 3, 1]
17	6	1	[10]	[100]
18	3	1	[4]	[100]
19	8	4	[6, 10, 5, 8]	[9, 1, 8, 100]
20	10	5	[8, 10, 3, 7, 8]	[10, 100, 5, 1, 6]
21	10	3	[8, 3, 8]	[10, 9, 100]
22	3	3	[6, 1, 9]	[4, 8, 100]
23	2	3	[3, 4, 2]	[8, 100, 7]
24	9	6	[1, 8, 6, 8, 4, 9]	[3, 7, 10, 100, 4, 3]
25	4	1	[9]	[100]
26	9	2	[5, 9, 1]	[100, 9]
27	4	3	[4, 10, 8]	[100, 10, 7]
28	8	5	[7, 10, 5, 8, 5]	[3, 9, 8, 100, 2]
29	5	5	[9, 2, 5, 2, 2]	[10, 4, 7, 9, 100]
30	5	4	[5, 4, 5, 7]	[9, 100, 9, 3]
31	9	3	[9, 6, 1]	[100, 5, 1]

Table 3. *ZA vs. HPRP Methods Using Interpolation Line Search.*

Problem	ZA		HPRP	
	Iter	Time (sec)	Iter	Time (sec)
1	12	0.0593	13	0.0634
2	9	0.0301	9	0.0315
3	26	0.0748	30	0.0880
4	14	0.0354	14	0.0634
5	37	0.1622	39	0.1745
6	22	0.0487	25	0.0508
7	17	0.0392	23	0.0620
8	16	0.0224	20	0.0329
9	14	0.0473	20	0.0727
10	30	0.1074	32	0.1333
11	20	0.0283	24	0.0487
12	29	0.0627	26	0.0554
13	7	0.0155	6	0.0109
14	33	0.1148	40	0.1395
15	10	0.0163	10	0.0162
16	34	0.1298	26	0.1043
17	34	0.0366	33	0.0336
18	11	0.0093	18	0.0192
19	22	0.0645	30	0.1020
20	22	0.0745	23	0.0843
21	43	0.1040	54	0.1518
22	15	0.0334	13	0.0441
23	9	0.0162	10	0.0231
24	29	0.1246	36	0.1712
25	14	0.0142	14	0.0153
26	77	0.1415	81	0.1399
27	17	0.0367	17	0.0380
28	29	0.0948	36	0.1252
29	37	0.1365	32	0.1139
30	18	0.0544	16	0.0461
31	42	0.0942	37	0.0878

Table 4. *ZA vs. HPRP Methods Using Exact Line Search.*

Problem	ZA		HPRP	
	Iter	Time (sec)	Iter	Time (sec)
1	13	0.0359	12	0.0524
2	11	0.0155	10	0.0142
3	32	0.0369	33	0.0387
4	20	0.0187	22	0.0221
5	52	0.0721	51	0.0705
6	30	0.0343	43	0.0377
7	23	0.0257	31	0.0337
8	32	0.0192	26	0.0165
9	26	0.0298	26	0.0290
10	33	0.0366	31	0.0353
11	20	0.0141	9	0.0169
12	39	0.0354	36	0.0310
13	9	0.0071	9	0.0064
14	34	0.0426	33	0.0428
15	12	0.0071	15	0.0084
16	32	0.0441	27	0.0371
17	28	0.0120	33	0.0138
18	21	0.0087	28	0.0108
19	31	0.0359	28	0.0329
20	25	0.0319	29	0.0404
21	48	0.0366	50	0.0408
22	13	0.0126	12	0.0118
23	7	0.0066	9	0.0088
24	31	0.0453	50	0.0779
25	13	0.0060	19	0.0078
26	71	0.0428	88	0.0527
27	18	0.0169	18	0.0173
28	40	0.0483	31	0.0396
29	35	0.0478	29	0.0398
30	14	0.0159	18	0.0203
31	42	0.0398	48	0.0438

Table 5. *Comparing Success Rates on 100 Randomly Generated Problems.*

Method	Line Search	Success Rate (%)
ZA	Interpolation Inexact	66
HPRP	Interpolation Inexact	65
ZA	Newton's Exact	51
HPRP	Newton's Exact	49

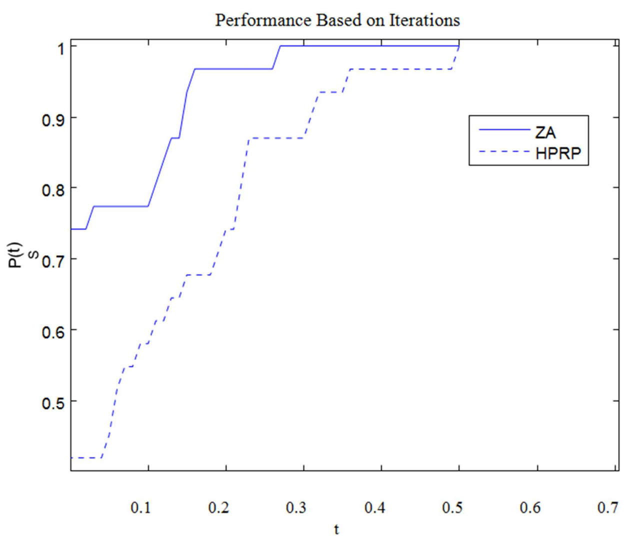

Figure 1. *Performance profile of HPRP vs ZA based on number of iterations and interpolation inexact line search.*

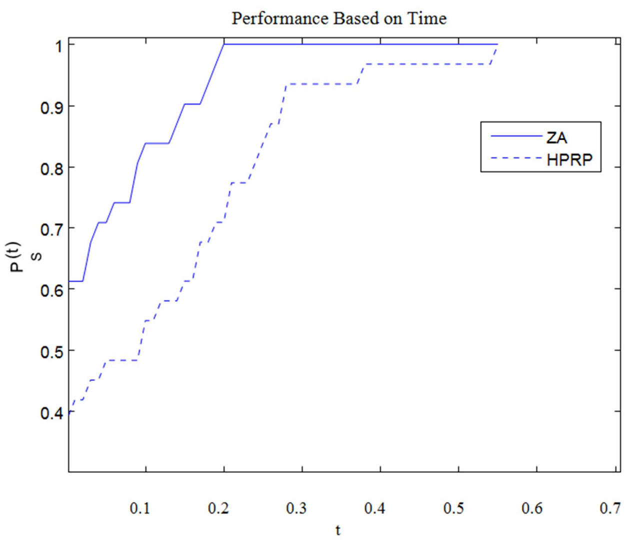

Figure 4. *Performance profile of HPRP vs ZA based on CPU time and Newton's exact line search.*

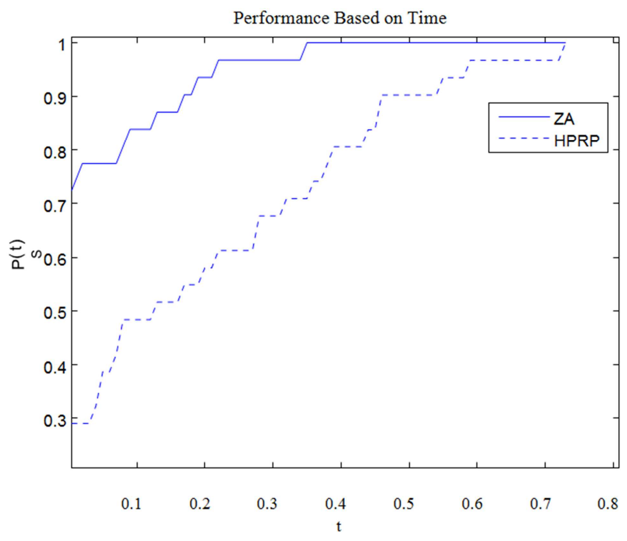

Figure 2. *Performance profile of HPRP vs ZA based on CPU time and interpolation inexact line search.*

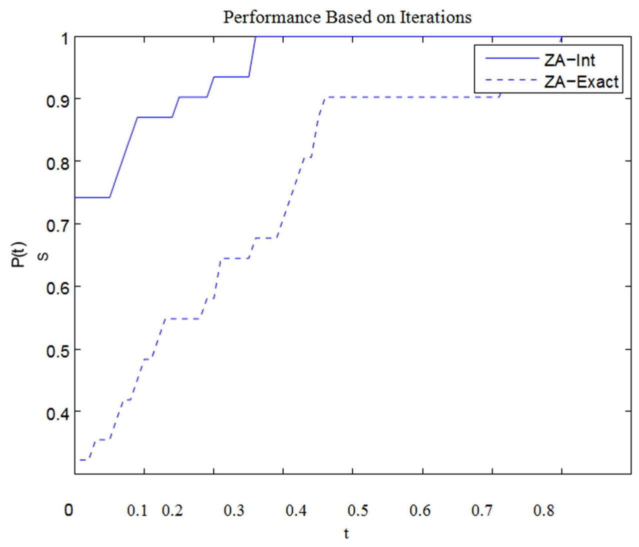

Figure 5. *Performance profile of ZA-Int vs ZA-Exact based on number of iterations.*

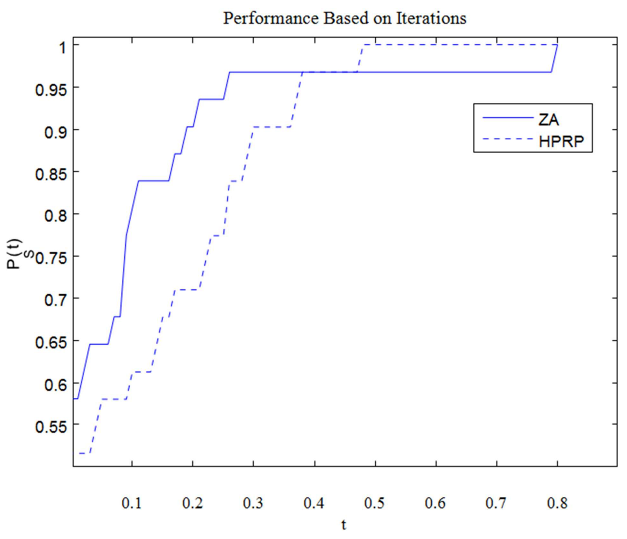

Figure 3. *Performance profile of HPRP vs ZA based on number of iterations and Newton's exact line search.*

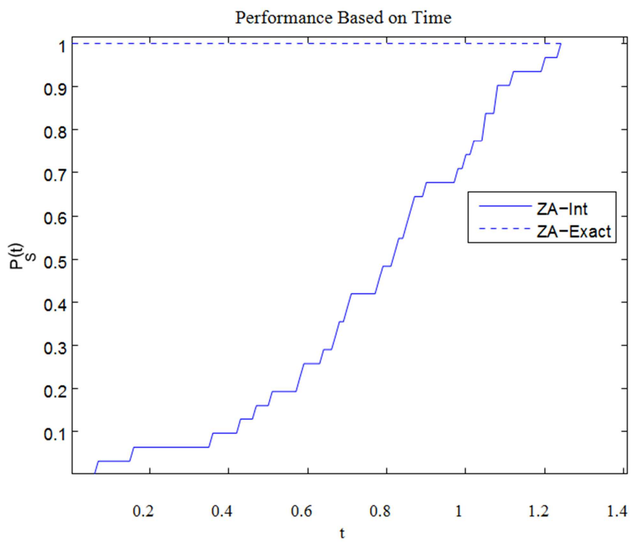

Figure 6. *Performance profile of ZA-Int vs ZA-Exact based on CPU time.*

6. Conclusion

This work presented a concept of weighted analytic center for second-order cone constraints. It also showed that ZA and HPRP are effective conjugate gradient methods for computing the weighted analytic center.

The results of our numerical experiments show that ZA is more effective than HPRP in both the number of iterations and CPU time in finding the weighted analytic center. The study also investigated the effect of line search on the performance of ZA. It is found that ZA with quadratic interpolation takes more time than ZA with Newton's exact line search. However, ZA with quadratic interpolation takes less iterations than ZA with Newton's exact line search.

This work assumed that an interior point is available for each test problem. It would be of interest to study other methods for finding the weighted analytic center that do not require any interior point. This is under investigation.

Acknowledgements

The second author would like to thank the Department of Mathematics and Statistics at Northern Arizona University for providing him with a sabbatical leave home in 2021. A part of the research work was done during this leave.

References

[1] I. Abdullahi and R. Ahmad, "Convergence analysis of a new conjugate gradient method for unconstrained optimization". *Applied Mathematical Sciences*, vol. 9, no. 140, 2015, pp 6969-6984.

[2] A. Alhawarat, M. Mamat, M. Rivaie and Z. Salleh, "An efficient hybrid conjugate gradient method with the strong Wolfe-Powell line search, *Mathematical Problems in Engineering*, vol. 2015, Article ID 103517, 2015.

[3] D. S. Atkinson and P. M. Vaidya, "A scaling technique for finding the weighted analytic center of a polytope," *Math. Prog.*, vol. 57, 1992, pp. 163–192.

[4] C. Barta, M. Kolar and J. Sustek, "On Hessians of Composite Functions", *Applied Mathematical Sciences*, vol. 8, no. 172, 2014, pp. 8601 – 8609.

[5] R. Burden and D. Faires, "Numerical Analysis", 9th Ed., Addison Wesley, 2011.

[6] V. L. Basescu and J. E. Mitchell, "An Analytic Center Cutting Plane Approach for Conic Programming", *Mathematics of Operations Research*, vol. 33, no. 3, 2008, pp. 529-551.

[7] S. Boyd and L. El Ghaoui, "Method of Centers for Minimizing Generalized Eigenvalues", *Linear Algebra and its Applications*, vol. 188, 1993, pp. 63-111.

[8] J. L. Goffin and J. P. Vial, "On the computation of weighted analytic centers and dual ellipsoids with the projective algorithm", *Mathematical Programming*, vol. 60, 1993, pp. 81-92.

[9] N. Goldberg and S. Leyffer, "An active-set method for second-order conic-constrained quadratic programming", *SIAM Journal on Optimization*, vol. 25, no. 3, 2015, pp. 1455-1477.

[10] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient method", *Pacific Journal of Optimization*, vol. 2, no. 1, 2006, pp. 35-58.

[11] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems", *J. Res. Natl. Bur. Stand.*, vol. 49, no. 6, 1952, pp. 409-436

[12] W. Jia, T. Ding and M. Shahidehpour, "Second-Order Cone Programming for Data-Driven Fluid and Gas Energy Flow With a Tight Reformulation, "*IEEE Transactions on Power Systems*, vol. 36, Issue 2, 2021, pp. 1652 - 1655.

[13] S. Jibrin, "Computing Weighted Analytic Center for Linear Matrix Inequalities Using Infeasible Newton's Method", *Journal of Mathematics, vol. 2015, Article ID 456392, 2015, 9 pages.*

[14] S. Jibrin, I. Abdullahi, "Conjugate Gradient Methods for Computing Weighted Analytic Center for Linear Matrix Inequalities Under Exact and Quadratic Interpolation Line Searches", *American Journal of Applied Mathematics*, vol. 8, Issue 1, 2020, pp. 1-10.

[15] S. Jibrin and I. Pressman, "Monte Carlo Algorithms for the Detection of Nonredundant Linear Matrix Inequality Constraints," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 2, 2001, pp. 139–153.

[16] S. Jibrin and J. W. Swift, "The Boundary of the Weighted Analytic Center for Linear Matrix inequalities." *Journal of Inequalities in Pure and Applied Mathematics*, vol. 5, Issue 1, Article 14, 2004.

[17] X. Liu, Z. Shen and P. Lu, "Entry Trajectory Optimization by Second-Order Cone Programming", *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, 2016, pp. 227-241.

[18] J. Machacek and "S. Jibrin, An Interior Point Method for Solving Semidefinite Programs Using Cutting Planes and Weighted Analytic Centers", *Journal of Applied Mathematics*, Vol. 2012, Article ID 946893, 2012, 21 pages.

[19] R. D. C. Monteiro and T. Tsuchiya, "Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions, *Math. Programming*, vol. 88, 2000, pp. 61-83.

[20] E. Polak and G. Ribiere, "Note sur la convergence de methodes de directions conjuguees", *ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique*, vol. 3, Issue R1, 1969, pp. 35-43.

[21] M. J. D. Powell, "Restart procedures of the conjugate gradient method", *Math. Prog.*, vol. 2, 1977, pp. 241-254.

[22] I. S. Pressman and S. Jibrin, "A Weighted Analytic Center for Linear Matrix Inequalities", *Journal of Inequalities in Pure and Applied Mathematics*, vol. 2, Issue 3, Article 29, 2002.

[23] M. Rivaie, A. Abashar, M. Mamat and M. Ismail, The convergence properties of a new type of conjugate gradient methods, *Applied Mathematical Sciences, vol.* 9, no. **54,** 2015, pp. 2671-2682.

[24] Z. Salleh and A. Alhawarat, “An efficient modification of the Hestenes-Stiefel nonlinear conjugate gradient method with restart property”, *Journal of Inequalities and Applications*, vol. 2016, no. 1, Article 110, 2016, 14 pages.

[25] G. Sonnevend, “New algorithms in convex programming based on a notion of ‘centre’ (for systems of analytic inequalities) and on rational extrapolation”, *International Series of Numerical Mathematics*, vol. 84, 1988, pp. 311-326.

[26] R. J. Vanderbei and H. Yurttan, “Using LOQO to Solve Second-Order Cone Programming Problems”, Technical Report SOR-98-09, Statistics and Operations Research, Princeton University, 1998.

[27] L. Vandenberghe and S. Boyd, “Convex Optimization”, Cambridge University Press, New York 2004.

[28] L. Vandenberghe and S. Boyd, “Semidefinite Programming”, *SIAM Review*, vol. 38, 1996, pp. 49-95.

[29] L. Vandenberghe, S. Boyd, H. Lebret and M. S. Lobo, “Applications of second-order cone programming”, *Linear Algebra and Its Applications*, vol. 284, 1998, pp. 193-228.

[30] A. Weigandt, K. Tuthill and S. Jibrin, “Constraint Consensus Methods for Finding Interior Feasible Points in Second-Order Cone Constraints”, *Journal of Applied Mathematics*, vol. 2010, Article ID 307209, 2010, 19 pages.

[31] M. Yamashita, T. J. Mullin and S. Safarina, “An efficient second-order cone programming approach for optimal selection in tree breeding”, *Optimization Letters*, https://doi.org/10.1007/s11590-018-1229-y, 2018.

[32] Y. Zhang, H. Zheng, C. Zhang, Global convergence of a modified PRP conjugate gradient method, *Procedia Engineering,* vol. 31*,* 2012, pp. 986-995.

[33] Q. Zhao, W. Fu and Z. Chen, “A Sensitivity Result for Quadratic Second-Order Cone Programming and its Application”, *Applications of Mathematics*, vol. 66, 2021, pp. 413-436.