Linear and Nonlinear Analysis
Yokohama Publishers
ISSN 2188-8167

Volume 5, Number 2, 2019, 305–316

A PROJECTION AND CONTRACTION METHOD FOR CIRCULAR CONE COMPLEMENTARITY PROBLEM

JUHE SUN AND XIAOHUI XU

Abstract. In this paper, we consider complementarity problem associated with the circular cone, which is a type of nonsymmetric cone complementarity problem. A projection and contraction method is presented which requires some projection calculations and functional computations. It is proved that the iteration sequence produced by the proposed method converges to a solution of the circular cone complementarity problem. Numerical experiments also show the effectiveness of this method.

1. Introduction

The circular cone complementarity problem(abbrevd.CCCP), which is to find $x \in R^n$ such that

$$x \in l_\theta, F(x) \in l_\theta^*, \langle x, F(x)\rangle = 0. \tag{1.1}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, R^n is an n-dimensional Euclidean space, $F : R^n \to R^n$ is a continuously differentiable operator, l_θ is a circular cone and l_θ^* is the dual cone of l_θ , which can be expressed as

$$\begin{aligned} l_\theta &:= \{x = (x_1, x_2) \in R \times R^{n-1} | \, \|x\| \, cos\theta \le x_1\} \\ &= \{x = (x_1, x_2) \in R \times R^{n-1} | \, \|x_2\| \le x_1 \tan\theta\}. \end{aligned} \tag{1.2}$$

$$\begin{aligned} l_\theta^* &:= \{y = (y_1, y_2) \in R \times R^{n-1} | \, \|y\| \sin\theta \le y_1\} \\ &= \{y = (y_1, y_2) \in R \times R^{n-1} | \, \|y_2\| \le y_1 cot\theta\}. \end{aligned} \tag{1.3}$$

The CCCP has a wide range of applications in finance, image processing and engineering problems, practical problems such as the multi-fingered robot grasping manipulation, quadruped robot dynamic optimization and so on, see [1, 3, 8]. In recent years, researchers have developed the properties of the spectral decomposition of the circular cone, see [2], and presented smoothing Newton method for solving the linear circular cone complementarity problems, see [5, 6]. However, the research on the algorithm of the circular cone constraint nonlinear complementarity problem is relatively rare. In this paper, we construct a projection and contraction algorithm to solve the circular cone complementarity problem based on the the circular cone complementarity function which is proposed and proved in [2]. This paper is organized as follows.

2010 *Mathematics Subject Classification.* 65K05, 65M12, 90C05, 90C30, 90C33.

Key words and phrases. Circular cone, complementarity problem, projection and contraction method, convergence analysis.

In Section 2, we introduce some basic concepts and useful results about the circular cone. A projection and contraction algorithm for the CCCP is presented in Section 3. The convergence property of the proposed method is analyzed in Section 4. Preliminary numerical results are reported in Section 5. Some conclusions are made in the last section.

2. Preliminaries

This section covers some basic concepts and useful results of circular cone and second-order cone, which will be extensively used in next section.

K^n represents a second-order cone, which can be expressed as

$$K^n := \{x = (x_1, x_2) \in R \times R^{n-1} | \, \|x_2\| \leq x_1\}. \tag{2.1}$$

In fact, when $\theta = \frac{\pi}{4}$, the circular cone (1.2) is exactly the second-order cone, thus the CCCP can be viewed as the generalization of the second-order cone complementarity problem. The relation between circular cone and the second-order cone as follows:

$$\begin{aligned} l_\theta &= A^{-1} K^n, \\ K^n &= A\, l_\theta, \end{aligned}$$

where $A := \begin{bmatrix} \tan\theta & 0 \\ 0 & I \end{bmatrix}$.

The Jordan product associated with second-order cone are defined as

$$x \circ y := \begin{bmatrix} \langle x, y \rangle \\ y_1 x_2 + x_1 y_2 \end{bmatrix}.$$

The Jordan product associated with circular cone are defined as

$$x \bullet y := \begin{bmatrix} \langle x, y \rangle \\ \max\{\tan^2\theta, 1\} x_1 y_2 + \max\{\cot^2\theta, 1\} y_1 x_2 \end{bmatrix}.$$

Theorem 2.1 ([4]). *Let l_θ and K^n be defined as in (1.2) and (2.1), respectively. Then, we have*

$$\begin{aligned} AK^n = l_{\frac{\pi}{2}-\theta} \quad &and \quad l_{\frac{\pi}{2}-\theta} = A^2\, l_\theta, \\ l_\theta^* = l_{\frac{\pi}{2}-\theta} \quad &and \quad (l_\theta^*)^* = l_\theta. \end{aligned}$$

Theorem 2.2 ([4]). *For any $x \in R^n$, the projection of x onto l_θ is defined as*

$$\Pi_{l_\theta}(x) = (\lambda_1(x))_+ \cdot u_x^{(1)} + (\lambda_2(x))_+ \cdot u_x^{(2)}. \tag{2.2}$$

where $(a)_+ := \max\{0, a\}$, $\forall a \in R$,

$$\begin{aligned} \lambda_1(x) &= x_1 - \|x_2\| \, cot\theta, \\ \lambda_2(x) &= x_1 + \|x_2\| \, tan\theta, \end{aligned}$$

$$u_x^{(1)} = \frac{1}{1+\cot^2\theta}\begin{bmatrix} 1 & 0 \\ 0 & \cot\theta \end{bmatrix}\begin{bmatrix} 1 \\ -w \end{bmatrix},$$
$$u_x^{(2)} = \frac{1}{1+\tan^2\theta}\begin{bmatrix} 1 & 0 \\ 0 & \tan\theta \end{bmatrix}\begin{bmatrix} 1 \\ w \end{bmatrix}.$$

with $w = \frac{x_2}{\|x_2\|}$ *if* $x_2 \neq 0$, *and any vector in* R^{n-1} *satisfying* $\|w\| = 1$ *if* $x_2 = 0$.

For convenience, we let x_+ denote the projection of x onto the l_θ , and x_- be the projection of $-x$ onto the l_θ^*. Hence, it is easy to verify that $\langle x_+, x_- \rangle = 0$ for any $x \in R^n$.

3. Projection and contraction algorithm

This section proposes a projection and contraction method for solving the circular cone complementarity problem (CCCP) (1.1). Note that, for solving the second-order cone complementarity problem (SOCCP), a popular approach is to reformulate it as an unconstrained smooth minimization by using the second-order cone complementarity functions. Two well-known complementarity functions are Natural Residual(NR) and Fischer Burmeister(FB). This inspires us to find a new complementarity function Φ for CCCP, provide that

$$\Phi(x) = 0 \Leftrightarrow x \ \textit{solves the CCCP}(1.1).$$

Hence, solving the problem (1.1) is equivalent to handing the unconstrained minimization problem

$$\min \frac{1}{2}\|\Phi(x)\|^2.$$

Reference [2] introduces a class of complementarity function, which is called the penalized natural residual function and defined as

$$\Phi_p(x, y) = x - (x - y)_+ + p(x_+ \bullet (-y)_-), p > 0. \tag{3.1}$$

It is easy to verify that when $p = 0$, $\Phi_p(x, y)$ reduces to the Natural Residual function $\Phi_{NR}(x, y)$.

Theorem 3.1 ([2]). *Let* $\Phi_p : R^n \times R^n \to R^n$ *be defined as in (3.1). Then* Φ_p *is an complementarity function for CCCP, i.e, for any* $x, y \in R^n$,

$$\Phi_p(x, y) = 0 \Leftrightarrow x \in l_\theta, y \in l_\theta^* \ \textit{and} \ \langle x, y \rangle = 0.$$

The proof of Theorem 3.1 have been completed in reference [2] and so is omitted. Let $e(x, \alpha)$ denote the residual of the equation. Then, solving the CCCP (1.1) is equivalent to finding a zero point of $e(x, \alpha)$.

$$e(x, \alpha) = x - (x - \alpha F(x))_+ + p(x_+ \bullet (-\alpha F(x))_-). \tag{3.2}$$

Now we are in position to state our algorithm.

Algorithm 3.1.

Step 0 Given $\varepsilon > 0$, let $0 < v < u < 1, \gamma \in (0,2), s \in (0,1), \alpha = \alpha_0 = 1$ and x_0 be an arbitrary starting point. Set $k := 0$.

Step 1 Compute $e(x^k, \alpha_k)$ be defined as in (3.2), if $\left\|e(x^k, \alpha_k)\right\| < \varepsilon$, stop; Otherwise, go to step 2.

Step 2 Let m_k be the smallest nonnegative integer statisfying $\alpha_k = \alpha s^{m_k}$ such that

$$\frac{\alpha_k \left\|F(x^k) - F(x^k - e(x^k, \alpha_k))\right\|}{\|e(x^k, \alpha_k)\|} \leq u. \tag{3.3}$$

If

$$\frac{\alpha_k \left\|F(x^k) - F(x^k - e(x^k, \alpha_k))\right\|}{\|e(x^k, \alpha_k)\|} \leq v, \tag{3.4}$$

then $\alpha = 1.5\alpha_k$.

Step 3 Calculate

$$\rho(x^k, \alpha_k) = e(x^k, \alpha_k)^{\mathrm{T}} \frac{d(x^k, \alpha_k)}{\|d(x^k, \alpha_k)\|^2}, \tag{3.5}$$

$$d(x^k, \alpha_k) = e(x^k, \alpha_k) - \alpha_k[F(x^k) - F(x^k - e(x^k, \alpha_k))]. \tag{3.6}$$

Step 4 Compute

$$x^{k+1} = (x^k - \gamma\rho(x^k, \alpha_k)d(x^k, \alpha_k))_+. \tag{3.7}$$

Set $k := k + 1$; go to step 1.

Remark 3.2. It follows from the reference [9] that step 2 is feasible. The idea of step 3 and step 4 refers to the references [7, 10, 12, 13]. Hence, the algorithm is available. Detailed analysis can be found in the next part.

4. Convergence analysis

Lemma 4.1. *Let $\rho(x^k, \alpha_k)$ be defined by (3.5), then $\rho(x^k, \alpha_k) \geq \frac{1}{2}$.*

Proof. It is easy to verify that the problem is equivalent to proof

$$2\left\langle e(x^k, \alpha_k), d(x^k, \alpha_k)\right\rangle - \left\|d(x^k, \alpha_k)\right\|^2 \geq 0.$$

it follows from(3.3)-(3.6), we have that

$$\begin{aligned}
&2\left\langle e(x^k,\alpha_k), d(x^k,\alpha_k)\right\rangle - \left\|d(x^k,\alpha_k)\right\|^2 \\
&= \left\langle 2e(x^k,\alpha_k) - d(x^k,\alpha_k), d(x^k,\alpha_k)\right\rangle \\
&= \left\langle e(x^k,\alpha_k) - \alpha_k[F(x^k) - F(x^k - e(x^k,\alpha_k))], d(x^k,\alpha_k)\right\rangle \\
&= \left\|e(x^k,\alpha_k)\right\|^2 - \alpha_k^2\left\|F(x^k) - F(x^k - e(x^k,\alpha_k))\right\|^2 \\
&\geq (1-L^2)\left\|e(x^k,\alpha_k)\right\|^2 \\
&\geq 0.
\end{aligned}$$

where $0 < L < 1$, and the first inequality follows from [9, Lemma 3.2] , so the proof is completed. □

Lemma 4.2 ([9]). *Suppose that $F(x)$ is monotonous and x^* is a solution of the CCCP. $e(x,\alpha)$ and $d(x,\alpha)$ are defined by (3.2) and (3.6), respectively. Then, the following inequalities hold.*

$$\begin{aligned}
&\langle x - x^*, d(x,\alpha)\rangle \geq \langle e(x,\alpha), d(x,\alpha)\rangle . \\
&\langle e(x,\alpha), d(x,\alpha)\rangle \geq (1-L)\|e(x,\alpha)\|^2.
\end{aligned}$$

Lemma 4.3 ([11]). *For all $x \in R^n$ and $\tilde{\alpha} \geq \alpha > 0$, it holds that*

$$\|e(x,\tilde{\alpha})\| \geq \|e(x,\alpha)\| .$$

Theorem 4.4. *Suppose that $F(x)$ is continuous and monotonus and the solution set of CCCP (1.1) is nonempty. Then the sequence $\{x_k\}$ generated by Algorithm 3.1 converges to a solution of the CCCP (1.1).*

Proof. Let x^* is a solution of CCCP (1.1), then it satisfies

$$\begin{aligned}
\left\|x^{k+1} - x^*\right\|^2 &= \left\|(x^k - \gamma\rho(x^k,\alpha_k)d(x^k,\alpha_k))_+ - x^*\right\|^2 \\
&\leq \left\|x^k - x^* - \gamma\rho(x^k,\alpha_k)d(x^k,\alpha_k)\right\|^2 \\
&= \left\|x^k - x^*\right\|^2 + \gamma^2\rho^2(x^k,\alpha_k)\left\|d(x^k,\alpha_k)\right\|^2 \\
&\quad - 2\gamma\rho(x^k,\alpha_k)\left\langle x^k - x^*, d(x^k,\alpha_k)\right\rangle \\
&\leq \left\|x^k - x^*\right\|^2 - \gamma(2-\gamma)\rho(x^k,\alpha_k)\left\langle e(x^k,\alpha_k), d(x^k,\alpha_k)\right\rangle \\
&\leq \left\|x^k - x^*\right\|^2 - \frac{\gamma(2-\gamma)(1-L)}{2}\left\|e(x^k,\alpha_k)\right\|^2.
\end{aligned}$$

where the first inequality follows since the projection operator is nonexpansive, the second and the last inequality follow from Lemma 4.1 and Lemma 4.2. Let $c_0 = \frac{\gamma(2-\gamma)(1-L)}{2}$, and $\inf\{\alpha_k\} = \alpha_{\min} > 0$, we get that

$$\sum_{k=0}^{\infty} c_0\left\|e(x^k,\alpha_k)\right\|^2 \leq \left\|x^0 - x^*\right\|^2.$$

It follows from Lemma 4.3 that

$$\lim_{k\to\infty} e(x^k, \alpha_{\min}) = 0,$$

which implies that the sequence $\{x^k\}$ is bounded. Let $\tilde{x}^*$ be a cluster point of $\{x^k\}$ and the subsequence $\{x^{k_j}\}$ converge to $\tilde{x}^*$. Since $e(x, \alpha_{\min})$ is continuous, we have

$$e(\tilde{x}^*, \alpha_{\min}) = \lim_{j\to\infty} e(x^{k_j}, \alpha_{\min}) = 0.$$

Hence, $\tilde{x}^*$ is a solution of CCCP.

In the following, we prove that the sequence $\{x^k\}$ has exactly one cluster point. Assume that $\tilde{x}$ is another cluster point, and denote

$$\delta := \|\tilde{x} - \tilde{x}^*\| > 0.$$

Because $\tilde{x}^*$ is a cluster point of the sequence $\{x^k\}$, there is a $k_0 > 0$ such that

$$\left\|x^{k_0} - \tilde{x}^*\right\| \leq \delta/2.$$

On the other hand,

$$\left\|x^k - \tilde{x}^*\right\| \leq \left\|x^{k_0} - \tilde{x}^*\right\|, \forall k \geq k_0.$$

It follows that

$$\left\|x^k - \tilde{x}\right\| \geq \|\tilde{x} - \tilde{x}^*\| - \left\|x^k - \tilde{x}^*\right\| > \delta/2, \forall k \geq k_0.$$

This contradicts with the assumption. Thus, $\tilde{x}^*$ is the unique cluster point of $\{x^k\}$. □

5. Numerical experiment

In this section, we use Algorithm 3.1 to solve three examples. All the program codes are written in MATLAB and run in MATLAB R2018b. We choose $u = 0.75, \gamma = 1.95$ and employed $\varepsilon = 10^{-8}$ as the termination criterion. We use ITER to represent the average number of iterations and ACPU to represent the average CPU time.

Example 5.1. We choose the test function $F(x) = Mx + q$, where $M = N^T N$, the element of N and q is randomly generated number by MATLAB and n is the dimension of the vector x, $x_0 = e$ as the starting point. In order to investigate the numerical performance of the Algorithm 3.1, we compare its numerical results with Smoothing Newton Algorithm in [6]. The experimental results are shown in the following Tables [1-4].

It can be seen from Tables [1-4] that the ACPU and ITER generated by the Algorithm 3.1 are relatively small and stable, so the Algorithm is effective. And when the angle is close to $\frac{\pi}{4}$, the Algorithm 3.1 is appropriately better than the Smooth Newton Algorithm, as the size of the problem raises, the ITER and ACPU of the both Algorithm increase.

Table 1. Solving numerical results with different Algorithms when $\theta=\frac{\pi}{3}$

	Algorithm3.1		Algorithm[6]	
n	ITER	ACPU	ITER	ACPU
100	149.0	0.5453	5.0	0.0700
200	213.0	0.6934	5.0	0.2783
300	261.0	2.4929	5.9	0.7516
400	308.0	6.8359	6.0	1.6756
500	341.0	13.6148	6.0	2.9128
600	343.0	18.7823	6.0	4.7698
700	374.0	20.4612	6.0	6.6911
800	398.0	28.3181	6.0	9.5330
900	410.0	30.9016	6.2	13.1500
1000	412.0	34.7429	6.6	18.4252
1100	418.0	63.7764	7.0	28.1791
1200	420.0	87.0782	7.0	37.5283
1300	420.0	98.8902	7.0	45.0912
1400	421.0	137.5849	7.0	54.3626
1500	424.0	199.8247	7.0	66.0708

Table 2. Solving numerical results with different Algorithms when $\theta=\frac{\pi}{4}$

	Algorithm3.1		Algorithm[6]	
n	ITER	ACPU	ITER	ACPU
100	16.0	0.3153	6.0	0.0863
200	16.0	0.3350	6.0	0.3343
300	16.0	0.4764	6.9	0.9180
400	17.0	0.8873	7.0	1.9657
500	16.0	1.2281	7.6	3.4351
600	17.0	1.7340	7.7	5.7232
700	17.0	2.8779	8.0	8.9254
800	17.0	3.4025	8.0	12.6333
900	17.0	4.2055	8.0	16.7760
1000	17.0	5.3609	8.0	22.2793
1100	17.0	9.4493	8.8	36.0361
1200	17.0	10.4033	9.0	48.2982
1300	18.0	12.5671	9.0	57.1845
1400	18.0	15.1305	9.0	71.0518
1500	18.0	17.7670	9.0	86.9723

Table 3. Solving numerical results with different Algorithms when $\theta=\frac{\pi}{5}$

	Algorithm3.1		Algorithm[6]	
n	ITER	ACPU	ITER	ACPU
100	21.0	0.1944	6.2	0.0900
200	22.0	0.6952	7.0	0.3804
300	23.0	0.6697	8.0	1.0528
400	23.0	1.7568	8.1	2.2717
500	24.0	1.1067	8.2	3.8557
600	25.0	2.9996	9.0	6.6764
700	26.0	4.7780	9.0	10.0183
800	26.0	5.4031	9.3	14.3294
900	26.0	6.6910	9.3	19.4508
1000	26.0	9.2296	9.5	26.5724
1100	25.0	14.2795	10.0	40.4612
1200	26.0	18.1010	10.2	63.4451
1300	27.0	22.8509	10.2	80.3178
1400	27.0	27.9905	10.0	89.2193
1500	30.0	31.3124	10.6	100.3036

Table 4. Solving numerical results with different Algorithms when $\theta=\frac{\pi}{6}$

	Algorithm3.1		Algorithm[6]	
n	ITER	ACPU	ITER	ACPU
100	21.0	0.1965	6.9	0.0951
200	28.0	0.3537	7.5	0.4107
300	28.0	0.5642	7.9	1.0323
400	29.0	1.0670	9.0	2.5356
500	30.0	1.9998	8.9	4.3882
600	30.0	2.7780	9.2	7.2381
700	30.0	2.0319	9.5	10.5241
800	30.0	3.7528	10.5	16.4956
900	30.0	5.9110	10.8	22.5813
1000	42.0	7.4395	11.0	30.5843
1100	50.0	16.5665	11.4	53.8481
1200	71.0	23.6775	11.2	67.4972
1300	80.0	50.9373	11.3	82.1468
1400	93.0	90.9530	11.0	103.4186
1500	113.0	91.2199	11.4	132.5169

Example 5.2. We choose the nonlinear function $F(x) = \begin{bmatrix} x_1 + 1 \\ e^{x_2} \\ 2x_3 + 3 \\ x_4^3 \\ x_5^2 \end{bmatrix}$.

Now we give the numerical results.

Table 5. Solving numerical results of CCCP with $x_0 = [00000]$

	$\pi/4$		$\pi/6$		$\pi/8$	
n	ITER	ACPU	ITER	ACPU	ITER	ACPU
$p = 0.01$	12.0	0.1120	13.0	0.1281	14.0	0.1329
$p = 0.03$	12.0	0.1024	12.0	0.1213	14.0	0.1178
$p = 0.05$	14.0	0.9242	14.0	0.0995	14.0	0.1259
$p = 0.07$	14.0	0.1273	14.0	0.1567	15.0	0.1193
$p = 0.09$	14.0	0.1061	15.0	0.1110	16.0	0.1085
$p = 0.10$	16.0	0.1036	15.0	0.1377	16.0	0.1269
$p = 0.30$	125.0	0.2303	13.0	0.1623	16.0	0.1059
$p = 0.50$	15.0	0.2396	21.0	0.1276	25.0	0.1110
$p = 0.70$	16.0	0.2111	23.0	0.1135	27.0	0.1123
$p = 0.90$	17.0	0.2036	24.0	0.2648	27.0	0.1061
$p = 1.00$	23.0	0.2027	28.0	0.1354	29.0	0.1299

Example 5.3. We choose the nonlinear function

Table 6. Solving numerical results of CCCP with $x_0 = [10000]$

	$\pi/4$		$\pi/6$		$\pi/8$	
n	ITER	ACPU	ITER	ACPU	ITER	ACPU
$p = 0.01$	14.0	0.1028	15.0	0.1223	15.0	0.1231
$p = 0.03$	15.0	0.1120	15.0	0.1196	15.0	0.1223
$p = 0.05$	15.0	0.1138	14.0	0.0995	16.0	0.0855
$p = 0.07$	15.0	0.1203	15.0	0.0921	17.0	0.1120
$p = 0.09$	16.0	0.0884	17.0	0.1297	17.0	0.1259
$p = 0.10$	20.0	0.1157	17.0	0.1369	21.0	0.0875
$p = 0.30$	124.0	0.1350	17.0	0.1822	26.0	0.2262
$p = 0.50$	22.0	0.2396	26.0	0.1488	27.0	0.1803
$p = 0.70$	21.0	0.2499	28.0	0.0977	29.0	0.1878
$p = 0.90$	22.0	0.1580	30.0	0.1012	33.0	0.1004
$p = 1.00$	24.0	0.1010	30.0	0.1524	38.0	0.1084

$$F(x) = \begin{bmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{bmatrix}.$$

Now we give the numerical results.

Table 7. Solving numerical results of CCCP with $x_0 = [00000]$

	$\pi/4$		$\pi/6$		$\pi/8$	
n	ITER	ACPU	ITER	ACPU	ITER	ACPU
$p = 10.0$	15.0	0.1473	16.0	0.1068	17.0	0.0641
$p = 20.0$	16.0	0.1259	16.0	0.1001	17.0	0.0693
$p = 30.0$	16.0	0.1302	16.0	0.0966	17.0	0.0718
$p = 40.0$	16.0	0.1156	16.0	0.1073	17.0	0.1004
$p = 50.0$	16.0	0.1296	16.0	0.1142	17.0	0.0944
$p = 60.0$	16.0	0.1054	16.0	0.1069	17.0	0.0921
$p = 70.0$	18.0	0.1180	16.0	0.1056	17.0	0.0985
$p = 80.0$	19.0	0.1152	16.0	0.1012	17.0	0.1035
$p = 90.0$	22.0	0.1213	26.0	0.1044	27.0	0.0988
$p = 100.0$	27.0	0.1279	29.0	0.1076	29.0	0.0959

Table 8. Solving numerical results of CCCP with $x_0 = [10000]$

	$\pi/4$		$\pi/6$		$\pi/8$	
n	ITER	ACPU	ITER	ACPU	ITER	ACPU
$p = 10.0$	16.0	0.0735	16.0	0.0864	17.0	0.0926
$p = 20.0$	18.0	0.1094	19.0	0.1431	20.0	0.1017
$p = 30.0$	19.0	0.0747	19.0	0.1045	20.0	0.0930
$p = 40.0$	20.0	0.1227	19.0	0.1042	20.0	0.0988
$p = 50.0$	21.0	0.0844	20.0	0.0952	21.0	0.1021
$p = 60.0$	21.0	0.1040	21.0	0.1019	22.0	0.0909
$p = 70.0$	21.0	0.1246	22.0	0.0743	23.0	0.1005
$p = 80.0$	24.0	0.1282	25.0	0.1125	26.0	0.0959
$p = 90.0$	24.0	0.1048	27.0	0.1189	29.0	0.1037
$p = 100$	31.0	0.1286	38.0	0.1130	40.0	0.0940

It can be seen from the Tables [5-8] that the starting point and angle have a little effect on the ITER and ACPU. As the value of p raises, the ITER increases significantly, and the ITER and ACPU are a little bigger only when $p = 0.30$ and $\theta = \frac{\pi}{4}$. So the Algorithm 3.1 is effective in general.

6. Conclusions

In this paper, we propose a projection and contraction method for the CCCP, and we prove the iteration sequence produced by the method converges to a solution of the CCCP. The results of numerical experiments show the effectiveness of this method.

References

[1] I. M. Bomze, *Copositivc optimization - recent developments and applications*, European J Operational Research. **216** (2012), 509–520.

[2] J. S. Chen, S. J Guo, X. H. Miao and Q. Nuo, *Constructions of complementarity functions and merit functions for circular cone complementarity problem*, Computational Optimization and Applicationd. **63** (2016), 495–522.

[3] J. S. Chen and C. H. Ko, *Optimal grasping manipulation for multifingered robots using semismooth Newton method*, Mathematical Problems in Engineering. **3** (2013), 206–226.

[4] J. S. Chen and J. C. Zhou, *Properties of circular cone and spectral factorizaton associated with circular cone*, Journal of Nonlinear and Convex Analysis **14** (2013), 807–816.

[5] X. N. Chi, Y. Wang and S. B. Zhang, *A Nonmonotone Inexact smoothing Newton method for linear circular cone complementarity problems*, Journal of Sichuan Normal Univcrsity(Natural Science). 2018.

[6] X. N. Chi, Y. Wang and S. B. Zhang, *Smoothing Newton method for linear circular cone complementarity problems*, Journal of Jilin Univcrsity(Science edition). **57** (2019), 0258–07.

[7] W. S. Guo and H. Y. Yin , *Projection and contraction methods for solving monotone F-complementary problem*, Journal of Xian Univcrsity of Art and Science. **15(1)** (2012), 0033–03.

[8] J. Y. Han, Y. N. Wang and N. H. Xiu, *On cone of nonsymmetric positive semidefinite matrices*, Linear Algebra and lts Applications. **433(4)** (2010), 718–736.

[9] D. Han and K. L. Hong, *Two new self-adaptive projection methods for variational inequality problems*, Comput. Math .Appl. **43** (2002), 1529–1537.

[10] B. S. He, *A new method for a class linear variational inequalities*, Mathematical Programming. **66** (1994), 137–144.

[11] B. S. He and L. Liao , *Improvements of some projection methods for monotone nonlinear variational inequalities*, Appl.J.Optimiz.Theory App. **112** (2002), 111–128.

[12] S. Pu and S. F. Zhao, *Projection and contraction methods for nonlinear complementarity problem*, Journal of Wuhan Normal Univcrsity(Natural Science) **5(4)** (2000), 391–396.

[13] D. F. Sun, *Projection Shrinkage Method for Generalized Nonlinear Complementarity Problems*, Computational Mathematics. **2** (1994), 183–194.

Manuscript received 30 May 2019
revised 12 July 2019

J. H. Sun
School of Science, Shenyang Aerospace University, 110136 Shenyang, China
E-mail address: juhesun@163.com

X. H. Xu
School of Science, Shenyang Aerospace University, 110136 Shenyang, China
E-mail address: janellexu@qq.com