AN EFFICIENT CUTTING PLANE ALGORITHM FOR THE
SMALLEST ENCLOSED CIRCLE PROBLEM

YI JIANG *

VC/VR Lab and Department of Mathematics, Sichuan Normal University
Chengdu, Sichuan 610066, China

CHUAN LUO AND SHENGGUI LING

Department of Mathematics, Sichuan Normal University
Chengdu, Sichuan 610066, China
and
Neijiang Vocational Technical College
Neijiang, Sichuan 641100, China

(Communicated by Jie Sun)

Abstract. In this paper, we consider the problem of computing the smallest enclosing circle. An efficient cutting plane algorithm is derived. It is based on finding the valid cut and reducing the problem into solving a series of linear programs. The numerical performance of this algorithm outperforms other existing algorithms in our computational experiments.

1. Introduction. The smallest enclosing circle problem is to find the circle of minimum radius that encloses a set of given circles \( \mathcal{O} = \{o_1, ..., o_m\} \) on \( \mathbb{R}^2 \) plane with centers \( \{(a_1, b_1), ..., (a_m, b_m)\} \) and radii \( \{r_1, ..., r_m\} \). It can be formulated as follows:

\[
\min \ R \\
\text{s.t.} \ \sqrt{(x - a_i)^2 + (y - b_i)^2} + r_i \leq R, \quad i = 1, ..., m.
\]

(1)

This problem is arisen from various application areas such as location analysis, military operation, mechanical engineering (stress optimization), and computer graphics (ray tracing). The detailed descriptions of these applications can be found in [1, 3, 4, 5, 11].

For the special case of enclosing points, namely the case of \( r_i = 0 \), there are a lot of efficient algorithms; see, e.g., Chrystal [2], Eliosoff and Unger [3], and Hearn and Vijan [6].

For the general cases of \( r_i > 0 \), Xu, Freund, and Sun [12] survey four known approaches for this problem, including a quadratic programming scheme, a subgradient approach, a randomized incremental algorithm, and a second order cone reformulation. It shows that the best algorithm is the quadratic programming scheme, whose numerical performance in their computational experiment outperforms the

2010 Mathematics Subject Classification. Primary: 90C30.

Key words and phrases. Cutting plane, second order cone programming, linear programming.

The first author is supported by National Natural Science Foundation of China No.11126336 and No.11201324, New Teachers’ Fund for Doctor Stations, Ministry of Education No.201115134120001, Fok Ying Tung Education Foundation No.141114, Youth fund of Sichuan province No.2013JQ0027.
The main idea is to formulate problem (1) into a linearly constrained quadratic programming problem as follows:

\[
\begin{align*}
\min & \quad x^2 + y^2 - z \\
\text{s.t.} & \quad -2a_i x - 2b_i y + z \leq (R - r_i)^2 - a_i^2 - b_i^2, \quad i = 1, \ldots, m.
\end{align*}
\] (2)

In [12], it also shows that by introducing auxiliary variables \(x_i + x = a_i, y_i + y = b_i,\) and \(z_i + r_i = R,\) problem (1) can be reformulated as a second order cone optimization problem as follows:

\[
\begin{align*}
\min & \quad R \\
\text{s.t.} & \quad \sqrt{x_i^2 + y_i^2} \leq z_i, \quad i = 1, \ldots, m \\
x_i + x = a_i, & \quad i = 1, \ldots, m \\
y_i + y = b_i, & \quad i = 1, \ldots, m \\
z_i + r_i = R, & \quad i = 1, \ldots, m,
\end{align*}
\] (3)

which was solved by SeDuMi [8]. In fact, from [7, 10, 9, 13], we can see that there are many methods such as interior point algorithms can solve a second order cone optimization problem, then the related software CPLEX can solve it.

In this paper, an efficient cutting plane algorithm is proposed for the smallest enclosing circle problem. The main idea is to find the valid cut and solve the second order cone problem (3) as a series of linear programs. We implement this cutting plane method and the other two approaches for this problem, i.e., a second order cone reformulation solved by the CPLEX Solver and the quadratic programming scheme reported in [12]. Under C SHARP and CPLEX, the numerical performance of the cutting plane method outperforms the other two in our computational experiment.

### 2. The cutting plane method

Now, we give the algorithm for the smallest circle problem (3), which is equivalent to problem (1) (see [12]). The main idea is using linear programming to solve the second order cone optimization problem iteratively.

**Algorithm 1** First, the constrain condition \(\sqrt{x_i^2 + y_i^2} \leq z_i (i = 1, \ldots, m)\) in problem (3) are replaced by \(|x_i| \leq z_i\) and \(|y_i| \leq z_i (i = 1, \ldots, m),\) and consider the following linear program:

\[
\begin{align*}
\min & \quad R \\
\text{s.t.} & \quad x_i + x = a_i, \quad i = 1, \ldots, m \\
y_i + y = b_i, & \quad i = 1, \ldots, m \\
z_i + r_i = R, & \quad i = 1, \ldots, m \\
-z_i \leq x_i \leq z_i, & \quad i = 1, \ldots, m \\
-z_i \leq y_i \leq z_i, & \quad i = 1, \ldots, m \\
z_i \geq 0, & \quad i = 1, \ldots, m.
\end{align*}
\] (4)

Then, problem (4) is a relaxation problem of problem (3).

Step 1. Initialize: \(k = 0.\) Use the CPLEX LP Solver to solve (4) and obtain \(u_{ki} = x_i\) and \(v_{ki} = y_i (i = 1, \ldots, m).\)

Step 2. Do while \((u_{ki} - u_{k-1i})^2 + (v_{ki} - v_{k-1i})^2 \geq \varepsilon\) (where \(\varepsilon\) is a predefined tolerance). Add the following valid cut

\[
-z_i \sqrt{u_{ki}^2 + v_{ki}^2} \leq x_i u_{ki} + y_i v_{ki} \leq z_i \sqrt{u_{ki}^2 + v_{ki}^2}, \quad i = 1, \ldots, m,
\] (5)
into problem (4), that is

\[
\begin{align*}
\min & \quad R \\
\text{s.t.} & \quad x_i u_{ki} + y_i v_{ki} \leq z_i \sqrt{u_{ki}^2 + v_{ki}^2}, \quad i = 1, \ldots, m \\
& \quad x_i u_{ki} + y_i v_{ki} \geq -z_i \sqrt{u_{ki}^2 + v_{ki}^2}, \quad i = 1, \ldots, m \\
& \quad x_i + x = a_i, \quad i = 1, \ldots, m \\
& \quad y_i + y = b_i, \quad i = 1, \ldots, m \\
& \quad z_i + r_i = R, \quad i = 1, \ldots, m \\
& \quad -z_i \leq x_i \leq z_i, \quad i = 1, \ldots, m \\
& \quad -z_i \leq y_i \leq z_i, \quad i = 1, \ldots, m \\
& \quad z_i \geq 0, \quad i = 1, \ldots, m.
\end{align*}
\]

(6)

Solve (6) by the CPLEX LP Solver, update \( k := k + 1 \), and obtain \( u_{ki} = x_i \) and \( v_{ki} = y_i \).

In this algorithm, we first consider the relaxation problem (4). The constrain condition \( \sqrt{x_i^2 + y_i^2} \leq z_i \) in problem (3) are replaced by \( |x_i| \leq z_i \) and \( |y_i| \leq z_i \) in problem (3). Then, we use the cutting plane \( -z_i \sqrt{u_{ki}^2 + v_{ki}^2} \leq x_i u_{ki} + y_i v_{ki} \leq z_i \sqrt{u_{ki}^2 + v_{ki}^2} \) to approximate the constrain condition \( \sqrt{x_i^2 + y_i^2} \leq z_i \) in problem (3).

Therefore, we should prove the cutting plane (5) is a valid cut, Algorithm 1 is convergent, and the optimal solution produced by Algorithm 1 is also the optimal solution to problem (3).

**Theorem 2.1.** The cutting plane (5) is a valid cut. More precisely, if \((x_i, y_i)\) belongs to the feasible region of problem (3), then \((x_i, y_i)\) also belongs to the feasible region of problem (6).

**Proof.** In fact, we only need to prove that for any feasible solution of

\[
\sqrt{x_i^2 + y_i^2} \leq z_i,
\]

in problem (3), they also satisfy

\[
\begin{align*}
- z_i \sqrt{u_{ki}^2 + v_{ki}^2} & \leq x_i u_{ki} + y_i v_{ki} \leq z_i \sqrt{u_{ki}^2 + v_{ki}^2} \\
- z_i & \leq x_i \leq z_i,
\end{align*}
\]

(8)

and

\[
- z_i \leq y_i \leq z_i
\]

(9)

in problem (6). Here, \( i = 1, \ldots, m \). It is obvious that (7) implies inequalities (8) hold. By the Cauchy-Schwarz inequality, we have

\[
- \sqrt{x_i^2 + y_i^2} \sqrt{u_{ki}^2 + v_{ki}^2} \leq x_i u_{ki} + y_i v_{ki} \leq \sqrt{x_i^2 + y_i^2} \sqrt{u_{ki}^2 + v_{ki}^2}.
\]

Then, inequality (7) yields

\[
- z_i \sqrt{u_{ki}^2 + v_{ki}^2} \leq x_i u_{ki} + y_i v_{ki} \leq z_i \sqrt{u_{ki}^2 + v_{ki}^2}.
\]

Hence, inequality (8) holds.

**Theorem 2.2.** Let \( \{R_k\}_{k=1}^{\infty} \) and \( \{(u_k, v_k)\}_{k=1}^{\infty} \) be the sequence of points produced by Algorithm 1. Then they are convergent.
Proof. \( \{R_k\}_{k=1}^{\infty} \) is convergent because it is decreasing monotone and lower bounded with \( R \geq r_i, i = 1, \ldots, m \). Now, to have \( \{(u_k, v_k)\}_{k=1}^{\infty} \) be convergent in \( l^1 \) normed spaces with the norm \( \| (u, v) \|_1 = \| u \| + \| v \| \), we first prove \( \{|u_k|\}_{k=1}^{\infty} \) and \( \{|v_k|\}_{k=1}^{\infty} \) are convergent. Because \( |u_{k+1}| \leq R_{k+1}, |u_k| \leq R_k \), and \( \{R_k\}_{k=1}^{\infty} \) is decreasing monotone, we can obtain
\[
0 \leq |u_{k+1}| - |u_k| \leq \frac{R_{k+1}}{R_k} \leq 1.
\]
It means that \( \{|u_k|\}_{k=1}^{\infty} \) is decreasing monotone and lower bounded. By the same method, we can see that \( \{|v_k|\}_{k=1}^{\infty} \) is also convergent. Therefore, \( \{(u_k, v_k)\}_{k=1}^{\infty} \) is convergent in normed spaces \( l^1 \).

Remark 1. Because \( \{(u_k, v_k)\}_{k=1}^{\infty} \) is convergent, the stop condition of Algorithm 1 is well defined as
\[
(|u_{k+1}| - |u_k|)^2 + (|v_{k+1}| - |v_k|)^2 \leq (u_{k+1} - u_k)^2 + (v_{k+1} - v_k)^2 < \varepsilon, \tag{11}
\]
where \( \varepsilon \) is the predefined tolerance.

Theorem 2.3. If \((x^*, y^*, R^*)\) is an optimal solution produced by Algorithm 1, then \((x^*, y^*, R^*)\) is also an optimal solution to problem \( (3) \).

Proof. If \((x^*, y^*, R^*)\) is not an optimal solution of problem \( (3) \), then there exists an optimal solution \((x, y, R)\) and \( R \leq R^* \). By Theorem 2.1, we can see \((x, y, R)\) belongs to the feasible region of problem \( (6) \). Because \((x^*, y^*, R^*)\) is an optimal solution of problem \( (6) \), the convergence and the stop condition of Algorithm 1 yield \( R = R^*, x = x^*, \) and \( y = y^* \).

3. Computational results. In this section, we compare the numerical performance of the cutting plane method presented in this paper, the second order cone optimization method by CPLEX, and the quadratic programming method given in [12].

We implement the three methods to solve the smallest enclosing problem using C SHARP. We also use CPLEX to solve the second order cone optimization, the quadratic programming, and the linear programming formulation of this problem.

The test data is generated randomly. The coordinates of the center of each input circle are generated following an independent normal distribution \( N(0, 16) \), and the radius of each of them is generated according to a uniform distribution \( U(0, 1) \). Then we check these circles so that no circle are contained in any other. The tolerance \( \varepsilon \) is set to be \( 10^{-6} \) in the cutting plane method presented in this paper.

We test random instances with different sizes, ranging from 50 to 12800 circles, and 50 random instances are solved for each problem size using an Intel core 1.8 GHz personal computer with 4GB memory.

The numerical results are summarized in Tables 1-4. In these tables, \( m \) denotes the number of circles, \( k \) denotes the number of iterations on the cutting plane method, Average is the average value of iterations, Maximum is the maximum of iterations, Minimum is the minimum of iterations, Obj Value denote the value of the objective function, Time denotes the CPU time in millisecond for solving each problem, SOCP denotes the method to solve second order cone optimization [3] by CPLEX. QP is the best algorithm in [12]. Algorithm 1 is the cutting plane method presented in this paper.
In Figure 1, we show a numerical example according to Algorithm 1. Giving eight circles, we can see that the smallest enclosing circle is derived, which encloses the eight circles accurately.

Table 1. Computational results for 12800 circles by the cutting plane method

| k | R            | (x, y)                                  | Time |
|---|--------------|-----------------------------------------|------|
| 1 | 20.9058731071536 | (0.545051135217305, 1.44782921739902) | 3122 |
| 2 | 20.9105991582134 | (0.2991926622599897, 1.31150575665027) | 3666 |
| 3 | 20.9125332765331 | (0.298922131451905, 1.31135826866618) | 4291 |
| 4 | 20.9125332765331 | (0.298922131451905, 1.31135826866618) | 4934 |

The results reported in Table 1 show that taking the 12800 circles instance as an example, it takes four iterations to reach the final results. It is worth noting that the value of the objective function $R$ and the center of the circles $(x, y)$ at $k = 3$ is equal to the value of them at $k = 4$. Hence, Algorithm 1 is able to reach very good accuracy for moderately large problems.
Table 2. The number of iterations on the cutting plane method

| m   | Average | Maximum | Minimum |
|-----|---------|---------|---------|
| 50  | 3.14    | 5       | 2       |
| 200 | 3.08    | 4       | 3       |
| 800 | 3.22    | 4       | 3       |
| 3200| 3.16    | 5       | 3       |
| 12800| 3.46  | 7       | 3       |

In Table 2, it gives the number of iterations on Algorithm 1. We test the random instances of different sizes, ranging from 50 to 12800 circles, and 50 random instances are solved for each problem size. The average value of iterations is about three. The maximum of iterations is up to seven and the minimum of iterations is only two. We can see that there is no relation between the number of iterations and the number of circles.

We next show the numerical performance of the cutting plane method presented in this paper, the second order cone optimization method by CPLEX, and the quadratic programming method given in [12].

Table 3. Objective function value

| Problem | SOCP       | QP         | Algorithm 1 |
|---------|------------|------------|-------------|
| m       | Obj Value  | Obj Value  | Obj Value   |
| 50      | 11.2096459 | 11.2096469 | 11.2096459  |
| 200     | 14.3832518 | 14.3832526 | 14.3832516  |
| 800     | 16.9222886 | 16.9222890 | 16.9222882  |
| 3200    | 19.1035735 | 19.1035733 | 19.1035728  |
| 12800   | 20.9684117 | Out of Memory | 20.9684103  |

Table 4. Average CPU time of three methods

| Problem | Time | Time | Time |
|---------|------|------|------|
| m       | SOCP | QP   | Algorithm 1 |
| 50      | 82.2 | 88.4 | 79.8  |
| 200     | 115.0| 180.2| 103.4 |
| 800     | 257.4| 1859.4| 254.6 |
| 3200    | 1041.6| 34883.9| 973.1 |
| 12800   | 8555.7| Out of Memory | 5024.6 |

The results reported in Table 3&4 show that Algorithm 1 performs very well. We can see from Table 3 that all algorithms are able to obtain good accuracy
for moderately large problems. Moreover, Table 4 presents that Algorithm 1 use smaller CPU time than SOCP and QP. It is interesting to note that for the largest problems implemented, SOCP is about 1.7 times more expensive (in terms of CPU time) than Algorithm 1 and QP fails due to the excessive memory requirement. Therefore, Algorithm 1 is more efficient (in terms of CPU time) than SOCP and QP for all test problem scales.

REFERENCES

[1] M. Berg, *Computational Geometry: Algorithms and Applications*, Springer, New York, 1997.
[2] P. Chrystal, On the problem to construct the minimum circle enclosing n given points in a plan, in *Proceedings of the Edinburgh Mathematical Society*, Tird Meeting, 1885, 30.
[3] J. Eliosoff and R. Unger, Minimal spanning circle of a set of points, *Computer Science: Computational Geometry Project*, School of Computer Science, McGill University, 1998, 308–507.
[4] J. Elzinga and D. Hearn, The minimum covering sphere problem, *Management Sci.*, 19 (1972), 96–104.
[5] B. Gärtner, Fast and robust smallest enclosing balls, in *Algorithms-ESA’ 99:7th Annual European Symposium, Proceedings* (eds. J. Nestril), Lecture Notes in Computer Science, 1643, Springer-Verlag, 1999, 325–338.
[6] D. W. Hearn and J. Vijan, Efficient algorithms for the minimum circle problem, *Oper. Res.*, 30 (1982), 777–795.
[7] M. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, Applications of second-order cone programming, *Linear Algebra and its Applications*, 284 (1998), 193–228.
[8] J. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, *Optim. Methods Softw.*, 11/12 (1999), 625-653.
[9] Y. Tian, S. C. Fang, Z. B. Deng and W. X. Xing, Computable representation of the cone of nonnegative quadratic forms over a general second-order cone and its application to completely positive programming, *Journal of Industrial and Management Optimization*, 9 (2013), 703–721.
[10] S. Y. Wang, Y. J. Liu and Y. Jiang, A majorized penalty approach to inverse linear second order cone programming problems, *Journal of Industrial and Management Optimization*, 10 (2014), 965–976.
[11] E. Welzl, Smallest enclosing disks (balls and ellipsoids) in New Results and New Trends in Computer Science, Springer-Verlag, 1991, 359–370.
[12] S. Xu, R. Freund and J. Sun, Solution methodologies for the smallest enclosing circle problem *Comput. Optim. Appl.*, 25 (2003), 283–292.
[13] Y. Zhang, Y. Jiang, L. Zhang and J. Zhang, A perturbation approach for an inverse linear second-order cone programming, *Journal of Industrial and Management Optimization*, 9 (2013), 171–189.

Received April 2015; 1st revision July 2015; 2nd revision December 2015.

E-mail address: yijiang103@163.com
E-mail address: luochuan.sicnu@gmail.com
E-mail address: ling2006@163.com