A new heuristic algorithm for the planar minimum covering circle problem

Hasan Hosseini Nasab*, Mahdi Tavana and Mohsen Yousefi

Industrial Engineering Department, Faculty of Engineering, Yazd University, Yazd, Islamic Republic of Iran

(Received 15 November 2013; accepted 15 February 2014)

The minimum covering circle problem is widely utilized in the studies of single or multiple facility location problems. It may be employed to locate the essential locations of factories, schools, fire departments, hospital establishments, and other facilities, which could be considered as a point in a plane theoretically. In this paper, a new geometrical algorithm is presented which determines the minimum covering circle of all points on a plane in four steps. The model validity was considered by studying the coordinates of points with random numbers and different distributions. In order to show the accuracy of the proposed algorithm, numerical experiments were carried out and compared with other studies in the field. The results show that the proposed algorithm extremely outperforms other examined algorithms.

Keywords: minimum covering circle; facility location; geometrical algorithm

1. Introduction

A wide range of location problems effect today’s firm and organizational objectives and decision-makings. The interest and service level, economic profits, quality of service, security, environmental protection, availability, optimum performance are some of the many criteria and objectives of the facility location problem; and hence, different methods and approaches have been employed in the literature to maximize the achievement of these objectives.

The facility location problem may be generally categorized in the two groups of multiple or single facility location, however the multi-facility location may be assumed as a combination of several single facility problems. The importance of the single facility location problem is clear as it provides a good insight into multi-facility problems and may even present suitable initial solutions. Hence, various researchers have focused on developing the single facility location problem, as further described in the literature review. Another approach to categorize the facility location problem includes grouping the solving process, for example discrete or continuous. The continuous category itself may be further grouped in different approaches i.e. the minimum covering circle solution, which is the focus of this paper.

The history of the classical minimum covering circle tracks back to Sylvester, who introduced this problem in 1857. Also known as the Euclidean 1-center and the smallest enclosing circle, the minimum covering circle problem seeks the smallest circle which
covers or encloses all existing points on a plane. In other words, the goal is to find the exact location of a facility with the minimum distance from the farthest point of interest. Sylvester (1857) and Chrystal (1885) proposed primal approaches to solve this problem, which start from a large circle and reduce the radius until it is enclosed by 2 or 3 points, as further described below:

Let point \( p_i = (a_i, b_i) \) be a member of the set \( P \) which contains \( n \) number of points \( p_1, p_2, \ldots, p_n \) in a \( R^2 \) space. Let \( f = (x, y) \) be the center of an assumed circle. It is clear that the distance between any of the existing points and the center of the circle may be calculated from Equation (1):

\[
d(f, p_i) = \sqrt{(x - a_i)^2 + (y - b_i)^2}
\]  

In order to achieve the minimum covering circle, Equation (1) has to be minimized for all existing points. Therefore, the objective formulate is as below:

\[
\min_{f \in R^2} \max_{p_i \in P} d(f, p_i).
\]  

Similar to the model of Dearing and Zeck (2009) the objective function may be redefined as below:

\[
\min z 
\]

\[
st: z \geq \sqrt{(x - a_i)^2 + (y - b_i)^2}, \quad p_i \in P
\]

where \( \max_{p_i \in P} d(f, p_i) \) is considered as variable \( Z \) to simplify the sentence.

The minimum covering circle problem is applicable for the following fields and problems:

- To find an appropriate place to facilitate an area by a radio station.
- To find an appropriate place to construct a fire station.
- To find an appropriate place for an emergency helicopter station to service all nodes in minimal time.

The various existing approaches in the minimum covering circle problem contain two major setbacks: first, the difficult perception of each method which requires a high degree of visualization; and second, the complexity of the algorithms which are time-consuming even with the use of modern computer wizards.

2. Literature of study

The smallest circle problem or minimum covering circle problem is a mathematical problem of computing the smallest circle that contains all of a given set of points in the Euclidean plane. The corresponding problem in n-dimensional space, the smallest bounding sphere problem, is to compute the smallest n-sphere that contains all of a given set of points. The smallest circle problem was initially proposed by the English mathematician Sylvester in 1857. Nair and Chandrasekaran (1971) developed a new method for the minimum covering circle problem using the polynomial quadratic method. A minimum covering circle problem with Euclidean distance was considered by Elzinga and Hearn (1972). The first linear algorithm was proposed by Megiddo (1983a) that was established based on Voronoi diagram. He showed that any fixed
dimension problem can be modeled as a convex quadratic programming and be solved in an $O(n)$ time. In another paper, Megiddo (1983b) investigated a weighted problem in the plane and presented an $O(n(\log n)^3(\log \log n)^2)$ algorithm for finding the minimum covering circle. An iterative algorithm based on Voronoi diagram was posed by Skyum (1991). The algorithm was appropriate for only medium size problem and cannot be used to obtain an optimum solution for large size problems. Welzl (1991) solved minimum covering circle based on Seidel’s LP algorithm in $d \geq 2$ and showed some efficiency of a heuristic algorithm. He considered 1000 and 5000 points with 2, 3, 5, and 10 dimensions to prove the advantages of the algorithm.

Minimum covering problem was solved with $d \geq 2$ by Hopp and Reeve (1996), Gartner (1999), Fisher, Gartner, and Kutz (2003), and Zhou, Tohemail, and Sun (2005). In addition, Gartner (1999) and Fisher et al. (2003) presented a robust algorithm based on the pivoting approach that solves the problem in a short time for $d \leq 20$.

Fuzzy approach is also used for minimum covering circle problem. Li, Kabadi, and Nair (2002, 2005) presented two fuzzy models that were based on possibility constraints and necessary constraints. In both algorithms, the coordination of points was considered using fuzzy variables. The algorithms were able to find the minimum covering circle in a fuzzy method.

For the large size problem, Zhou et al. (2005) presented two new algorithms. Its computational results showed that for a large size problem such as 512,000 points with $d = 100$, the result can be gained in a timescale less than an hour. Two other algorithms were proposed by Yildirim (2007) which was structured with the aim of Frank–Wolfe approach.

Shamos and Hoey (1975) proposed an $O(n \log n)$ time algorithm for the problem based on the observation that the center of the smallest enclosing circle must be a vertex of the farthest-point Voronoi diagram of the input point set.

Dearing and Zeck (2009) used the dual algorithm and solved the problem with ($d = 32$ and $d = 64$). El-Tamimi and Al-Zahrani (2012) developed a new geometrical algorithm based on the quadratic modeling, originally proposed by Nair and Chandrasekaran (1971). Their computational result was examined for the dimension of ($d = 32$).

The minimum covering circle problem is a mathematical problem of computing the smallest circle that contains all of a given set of points in the Euclidean plane. The minimum covering circle of a set $S$ can be determined by at most three points in $S$ which lie on the boundary of the circle. In such cases we need to investigate $(n-1)(n-2)/3$ solutions. However, the minimum enclosing circle can be found in linear time, but it is still time-consuming especially when the number of points increases. Several exact and heuristic algorithms are proposed for finding the minimum covering circle but most of them perform in many steps and need a lot of CPU time to reach the final circle. As discussed in literature of study, many approaches exist, however the difficult perception of each method, which needs a high degree, or the time-consuming overall processes due to the complexity of existing algorithms present much room for improvement. In this paper, a new geometrical algorithm is presented which determines the minimum covering circle of all points on a plane in four simple steps. At the result, in respect to other methods, the required mathematical calculations and the CPU time to find the minimum covering circle decrease. The proposed algorithm is described below.

3. Achieving the minimum covering circle

The proposed method in this paper consists of four consecutive steps, as presented below.

---

144    H. Hosseini Nasab et al.
In typical assumption of the problem it is implied that there are specified points which can be customers of a department store, applicants for wireless communication service, patients or citizens who want to take advantage from services. Coordinate of each point is determined with respect to a hypothetical source. The purpose is, to determine the minimum covering circle or in other words to determine the center and radius of the circle.

Defining the center can help decision-makers determine the site of the facility and the radius indicates the consideration of the power and capacities of service. For example, in a communication antenna problem and/or radio mast, in addition to determining the construction site of the facility, decision-makers need to define the power and capacity for the facility.

The proposed method of this paper to determine the minimum covering circle is defined in the following four steps:

1. Measure the distances between each pair of points and sort them in a chronological order. Choose the two points with the maximum distance. These points will be referred to as $P_1$ and $P_2$ (Figure 1). Using these two points form a circle with the diameter of the measured maximum distance. If all available points are covered with the drawn circle, the smallest covering circle is achieved, otherwise go to step 2.

$$d(p_i, p_j) = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}$$

Step 2: Measure the distance of all points from the center of the drawn circle in step 1, and determine the farthest uncovered point. Refer to this point as $P_3$. Draw a unique circle using the three points of $P_1$, $P_2$, and $P_3$ which are calculated using the following formula:

For any three points $P_1$, $P_2$, and $P_3$ with coordinates $(a, b)$, $(c, d)$, and $(e, f)$, respectively, the center of circle, $(X, Y)$ and the radius $R$, can be calculated as:

$$X = \frac{1}{2}\left((a^2 + b^2) \times (f - d) + (c^2 + d^2) \times (b - f) + (e^2 + f^2) \times (d - b)\right)/(a \times (f - d) + c \times (b - f) + e \times (d - b))$$

$$Y = \frac{1}{2}\left((a^2 + b^2) \times (e - c) + (c^2 + d^2) \times (a - e) + (e^2 + f^2) \times (c - a)\right)/(b \times (e - c) + d \times (a - e) + f \times (c - a))$$

Figure 1. The first circle which is drawn by $P_1$ and $P_2$. 
The achieved circle drawn by these three points (Figure 2) may be optimum.

It is necessary to note that $P_3$ must only be located in the two areas of $R_1$ or $R_2$ as shown in yellow color in Figure 3 otherwise the distance of this point from one of the two previous points will exceed the maximum calculated distance in step 1.

If all points have been covered by the drawn circle, this circle is optimum otherwise go to step 3.

**Step 3**: Repeat the process of finding the farthest point from the center of the last drawn circle in step 2 and refer to it as $P_4$. Using the three points, that is $P_3$, $P_4$ and either one of the points $P_1$ or $P_2$, draw the final circle Figure 4. Based on our observations, one of these two circles will be the minimum covering circle. Meaning that no points will exist outside of the drawn circle, in addition to that the drawn circle is the minimum circle that may be formed according to the following: whenever an arc is bigger than half of the circumference of the circle which no points are located on, the circle has the potential to be improved, otherwise the existing circle is the minimum circle (Elzinga & Hearn, 1972). To choose between $P_1$ and $P_2$ go to step 4.

![Figure 2](image2.png)

**Figure 2.** Second circle which is drawn by $P_1$, $P_2$, and $P_3$.

![Figure 3](image3.png)

**Figure 3.** Possible zone for $P_3$. 

$$R = \sqrt{(X - a)^2 + (Y - b)^2}$$
Step 4: Let $d_1$ represent the distance between the points $P_1$ and $P_3$, $d_2$ represent the distance between $P_1$ and $P_4$, $d_3$ represent the distance between $P_2$ and $P_3$, and $d_4$ represent the distance between $P_2$ and $P_4$ as shown in Figure 5. If the minimum value among $P_1$, $P_2$, $P_3$, and $P_4$ belongs to $P_1$ or $P_2$ then the point $P_1$ will be omitted otherwise the point $P_2$ will be omitted. For example, in Figure 5, $d_2$ has the smallest value, so the circle will be drawn using $P_2$, $P_3$, and $P_4$ which is the smallest covering circle.

In the next two sections, the validation procedure and the comparison of the proposed method with other existing methods are presented.
4. Validation of the proposed algorithm

The minimum covering circle can be drawn by many algorithms proposed in the literature or by drawing the circle crossing any three points which are not located in a direct line. But the important issue is the necessary steps to find the minimum covering circle that has effect on the calculation time. Therefore, the required time for reaching the final solution of the proposed algorithm is compared with the existing algorithms. Point scatterings and the total number of points in the study are also considered as two important factors with probabilistic impact on the speed and accuracy of the optimum solution. Therefore, it should be ensured that the proposed algorithm is capable of achieving optimum solutions based on any numbers of points and scattering distribution. The other aspect of attention is the impacts of the number of points and their distribution on the required computational time. Generally, it is expected that increasing the number of the points will lead to an increase in CPU time, which remains to be observed.

In this regard, four tests based on various probability distributions were executed with a number of 100, 1000, and 10,000 points each. Each test defines the distribution of points on a plane as presented below:

Test 1: \(x, y \sim \text{Uniform}(0, 100)\),
Test 2: \(x, y \sim \exp(\beta = 50)\),
Test 3: \(x, y \sim \text{Normal}(\mu = 100, \sigma = 50)\),
Test 4: \(x, y \sim \text{Normal}(\mu = \exp(\beta = 50), \sigma = \exp(\beta = 100))\).

4.1. Assessing the CPU computational time

A statistical two-tailed hypothesis test was designed to examine the impacts of point scattering and the number of points on the CPU computational time.

The independent variables are classified into two categories: (1) the total number of points and (2) the points’ scattering distribution; the response variable is determined as the CPU time. The below two null hypotheses are considered:

(1) \(H_0\): The point scattering distribution does not have an impact on CPU time.
(2) \(H_0\): The total number of points does not have an impact on CPU time.

In the designed hypothesis test, two numeric results (CPU time) are obtained for each intersection of independent variables which are shown in Table 1. In relation to the numeric results of Table 1, 12 figures which describe the point scattering and drawn circle are exhibited in Appendix 1.

Based on the numeric results presented in Table 1, the hypothesis test is considered at a 5% significant level. The SPSS 18 software was used for the variance analysis. As

| Number of points | Scattering (distribution) |
|------------------|---------------------------|
|                  | Test 1 | Test 2 | Test 3 | Test 4 |
| 100              | .0029  | .0028  | .003   | .0023  |
|                  | .0023  | .0027  | .0029  | .0024  |
| 1000             | .122   | .122   | .1291  | .1312  |
|                  | .1252  | .1298  | .1261  | .1267  |
| 10,000           | 16.493 | 16.148 | 16.417 | 16.587 |
|                  | 16.273 | 16.289 | 16.793 | 16.886 |
shown in Table 2, for the first independent variable ‘distribution’ the p-value is equal to .08, which means the first null hypothesis (H: the point scattering distribution does not have an impact on CPU time) is not rejected. In other words points’ scattering distribution does not have any impact on CPU time. In another case, for the ‘number of points’ variable the p-value equals .000, which shows an extreme impact on CPU time. Therefore, the second null hypothesis (H: the total number of points does not have an impact on CPU time) is rejected. As mentioned before, it was predictable that an increase in the number of points will lead to an increase in CPU time.

4.2. Assessing the number of circles

According to the proposed algorithm, the solution can be obtained by drawing one, two, or three circles. In this, the impact of the two independent variables (refer to 4.1) on the number of drawn circles which lead to the final solution is examined. The final solution here is the minimum covering circle. Hence, two null hypotheses were defined as below:

(1) $H_0$: Point scattering does not have an impact on the number of drawn circles.

(2) $H_0$: The total number of points does not have an impact on the number of drawn circles.

The results of studying the impacts of the points’ scattering distribution and their total number are provided in Table 3.

As shown in Table 4, the p-value is .044 for the ‘distribution’ factor which means the first hypothesis (H: point scattering does not have an impact on the number of drawn circles) is rejected. In other words altering the points’ scattering distribution has a

| Number of points | Scattering (distribution) |
|------------------|--------------------------|
|                  | Test 1 | Test 2 | Test 3 | Test 4 |
| 100              | 3      | 3      | 1      | 1      |
| 1000             | 2      | 2      | 2      | 1      |
| 10,000           | 3      | 2      | 2      | 1      |
|                  | 1      | 3      | 2      | 1      |
significant effect on the number of drawn circles. For the second hypothesis in this section (H: the total number of points does not have an impact on the number of drawn circles), the $p$-value equals .783 which shows that the number of points does not have any impact on the number of drawn circles.

Based on Sections 4.1 and 4.2, the validity of the proposed algorithm is confirmed. For further verification and validation the power of the proposed algorithm is compared with a recent proposed algorithm for finding minimum covering circle in Section 5.

5. Comparing computational results

In this study, the proposed algorithm was coded by the MATLAB 10 software, and the experiments were performed by a PC with an Intel Pentium Dual-Core (2.2 GHz) processor under Windows 7 using 3 GB of RAM. One of the factors in the proposed algorithm’s capability is the computational speed in comparison with the other studies in this field. Therefore, the recent study was chosen for comparison. The results of the comparison are provided in Table 5. The position of the experimented points is also provided in Figure A4.

The results show that our algorithm extremely outperforms other examined algorithms. For more perception, the difference in CPU time with El-Tamimi and Al-Zahrani is visualized in Figure 6.

Based on the CPU time results, the CPU computational time was fitted using a function which may be utilized to predict the CPU time in different numbers of points. The fitting process was performed using the SPSS 18 software. Figure 7 shows the resultant polynomial function.

Table 4. Analysis of variance based on the data in Table 3.

| Response variable: drawn circle | Type III sum of squares | df | Mean square | $F$  | Sig. |
|---------------------------------|-------------------------|----|-------------|------|------|
| Corrected model                 | 6.500                   | 11 | .591        | 1.182| .388 |
| Intercept                       | 73.500                  | 1  | 73.500      | 147.000| .000 |
| Distribution                    | 5.500                   | 3  | 1.833       | 3.667| .044 |
| No. of point                    | .250                    | 2  | .125        | .250 | .783 |
| Distribution × no. of point     | .750                    | 6  | .125        | .250 | .950 |
| Error                           | 6.000                   | 12 | .500        |      |      |
| Total                           | 86.000                  | 24 |             |      |      |
| Corrected total                 | 12.500                  | 23 |             |      |      |

Table 5. CPU time (seconds), comparing the results of the proposed algorithm with some algorithms in the literature.

| Algorithms                                      | 500      | 1000     | 3000     | 5000     |
|-------------------------------------------------|----------|----------|----------|----------|
| Proposed algorithm                              | .0311    | .1246    | 1.0990   | 3.2231   |
| El-Tamimi and Al-Zahrani (2012)                  | 43.08    | 172.11   | 1085.39  | 4240.23  |
| Patel (1995)                                    | 39.224   | 83.234   | 301.45   | 1389.45  |
| Sarkar and Chaudhuri (1996)                      | 34.180   | 60.111   | 200.01   | 1004.87  |
| Das, Chakraborti, and Chaudhuri (2001)           | 24.255   | 80.569   | 349.99   | 1012.2   |
6. Conclusion and suggestions for future research

The minimum covering circle is one of the most important problems in common location problems and optimization scopes. Minimum covering circle solutions may be
employed to locate the essential locations of facilities such as hospital establishments or fire departments.

In this scope, many approaches exist; however, the difficult perception of each method, which need a high degree or the time-consuming overall processes due to the complexity of existing algorithms present much room for improvement.

In this paper, a simple method for determining the minimum covering circle is presented, which is based on geometric approaches. The model validity was considered by studying the coordinates of points with random numbers and different distributions.

The observations indicated that an increase in the scattering of points led to the achievement of the final solution in the first or second step. The distribution of points scattering was also assessed by the means of ANOVA which resulted in the conclusion that changes the number of total points which will have a significant effect on the computational speed in reaching the solution. In contrast, the points’ scattering distribution (Test 1–Test 4) has no impact on the solution speed.

Compared with the recent proposed methods, the proposed algorithm of this paper has a much higher computational speed and less sensitivity to an increase in the total number of points.

The proposed method is designed for achieving the minimum covering circle in a two-dimensional space, regarding various applications, this algorithm can be developed for three- or more than three-dimensional spaces. Also, using the concept of reliability and the service level considering the available budget may be a new concept of study for further research.

References
Chrystal, G. (1885). On the problem to construct the minimum circle enclosing n given points in the plane. *Proceedings of the Edinburgh Mathematical Society*, 3, 30–33.
Das, P., Chakraborti, N. R., & Chaudhuri, P. K. (2001). Spherical minimax location problem. *Computational Optimization and Applications*, 18, 311–326.
Dearing, P. M., & Zeck, C. R. (2009). A dual algorithm for the minimum covering ball problem in \( \mathbb{R}^n \). *Operations Research Letters*, 37, 171–175.
El-Tamimi, A. A., & Al-Zahrani, K. (2012). An improved quadratic program for unweighted Euclidean 1-center location problem. *Journal of King Saud University – Engineering*, 25, 161–165. Retrieved from [http://dx.doi.org/10.1016/j.jksues.2012.04.003](http://dx.doi.org/10.1016/j.jksues.2012.04.003)
Elzinga, J., & Hearn, D. W. (1972). Geometrical solutions for some minimax location problems. *Transportation Science*, 6, 379–394.
Fisher, K., Gartner, B., & Kutz, M. (2003). Fast smallest-enclosing-ball computation in high dimensions. In *Algorithms – ESA*, volume 2832 of *Lecture Notes in Computer Science* (pp. 630–641).
Gartner, B. (1999). Fast and robust smallest enclosing balls. In *Proceedings of the 7th Annual European Symposium on Algorithms, ESA*. In *Lecture Notes in Computer Science* (pp. 325–338). Zurich: Springer-Verlag.
Hopp, T. H., & Reeve, C. P. (1996). *An algorithm for computing the minimum covering sphere in any dimension* (Technical Report NISTIR 5831). Gaithersburg, MD: National Institute of Standards and Technology.
Li, L., Kabadi, S. N., & Nair, K. P. K. (2002). Fuzzy versions of the covering circle problem. *European Journal of Operational Research*, 137, 93–109.
Li, L., Kabadi, S. N., & Nair, K. P. K. (2005). Fuzzy disk for covering fuzzy points. *European Journal of Operational Research*, 160, 560–573.
Megiddo, N. (1983a). The weighted Euclidean 1-center problem. *Mathematics of Operations Research*, 8, 498–504.
Megiddo, N. (1983b). Linear-time algorithms for linear programming in R3 and related problems. *SIAM Journal on Computing*, 12, 759–776.
Appendix 1

The test figures of Tables 1 and 3 are provided below. Figures A1–A3 show the algorithm’s general solution processes when the number of points are 100, 1000, and 10,000, respectively. The blue, red, and green circles represent the first, second, and third (final) circle, respectively.

Figure A1. Point scattering of the four executed tests with the total number of points of 100.
Figure A2. Point scattering of the four executed tests with the total number of points of 1000.
Visual results are shown in Figure A4 in accordance to Section 5.

Figure A3. Point scattering of the four executed tests with the total number of points of 10,000. Visual results are shown in Figure A4 in accordance to Section 5.

Figure A4. The proposed algorithm results, in accordance to the comparison tests.