

Research Article

A BSSS Algorithm for the Location Problem with Minimum Square Error

Jafar Fathali,¹ Mehdi Zaferanieh,² and Ahmad Nezakati¹

¹ Department of Mathematics, Shahrood University of Technology, University Blvd.,
Shahrood 36199 95161, Iran

² Department of Mathematics, Tarbiat Moallem University of Sabzevar, Toohid Town,
Sabzevar 9617976487, Iran

Correspondence should be addressed to Jafar Fathali, fathali@shahroodut.ac.ir

Received 7 May 2009; Revised 17 September 2009; Accepted 15 October 2009

Recommended by Khosrow Moshirvaziri

Let n weighted points be given in the plane R^2 . For each point a radius is given which is the expected ideal distance from this point to a new facility. We want to find the location of a new facility such that the sum of the weighted errors between the existing points and this new facility is minimized. This is in fact a nonconvex optimization problem. We show that the optimal solution lies in an *extended rectangular hull* of the existing points. Based on this finding then an efficient big square small square (BSSS) procedure is proposed.

Copyright © 2009 Jafar Fathali et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

In this paper we introduce a new version of single facility location problem. Let n points, called facilities, be given in the plane R^2 . The classical single facility location problem asks to find the location of a new facility such that the sum of the weighted distances from all facilities to the new one is minimized.

We consider a special single facility location problem such that each point p_i has a relevant radius r_i which is the ideal distance between the new facility and the point p_i . For example, if $r_i = r$ for $i = 1, \dots, n$ and all points lie on the circumference of a circle with radius r then center of this circle is optimal solution. Unfortunately in countless instances does not exist the location of a new facility such that its distance to each point p_i is exactly r_i . So we try to minimize the sum of the weighted square errors.

It is necessary to note that this problem differs from the covering problem which asks to find the minimum number of facilities such that the distance from any point p_i to the closest facility is less than or equal to r_i (see Mirchandani and Francis [1]). Another problem which may be thought to be near to our consideration is the problem of finding a circle closest to

the demand points. This problem is considered by Drezner et al. [2] and Brimberg et al. [3] which differs from our considered problem.

In what follows at first the problem formulation is presented in Section 2. Next in Section 3 it is shown that the problem is not convex and optimal solution lies in an *extended rectangular hull* of the existing facilities. Finally in Section 4 an efficient big square small square method is employed to solve the problem.

2. Problem Formulation

Let n points $p_i = (a_i, b_i)$, $i = 1, \dots, n$ be given in the plane R^2 . The coordinates of point p_i are corresponded to the position of a client where its weight and radius are $w_i \geq 0$ and r_i , respectively. The radius r_i is a given ideal distance between p_i and the new facility. The distance between any two points x and y is denoted by $d(x, y)$. Let $e(x, p_i) = (d(x, p_i) - r_i)^2$ be the *square error* between points x and p_i . The problem asks to find the location of a point $x \in R^2$ such that the sum of the weighted square errors over all points is minimized, that is:

$$\min F(x) = \sum_{i=1}^n w_i e(x, p_i). \quad (P_1)$$

As an application of this problem consider finding the location of a company in the vicinities of some cities with respect to the establishing and transportation costs. Suppose that cost of establishing a facility in the regions that are farther than a given distance r_i is very low. On the other hand a move away from a city causes that the transportation cost increases. Therefore a tradeoff between establishing and transportation costs seems to be reasonable. This problem has many other applications such as locating of powerhouses, stadiums, warehouses, dump sites, and other facilities that have both desirable and undesirable attributes. For instance, in the problem of locating powerhouses we want to find the location of a facility which is not to be closer than a specified distance to the population centers, because of increasing the risk of miscarriages. On the other hand, if the facility is so far from the population centers, cost of providing security, human forces, transportation installation, and other costs will increase.

3. Model Properties

In this section we pose some properties of the problem. Suppose that the distances in the plane R^2 are measured by l_2 norm. The following example shows that the problem (P_1) is nonconvex.

Example 3.1. Consider the point $p = (0, 0)$ with relevant radius $r > 0$. Then its square error function is

$$e(x, p) = \left(\sqrt{(x_1 - 0)^2 + (x_2 - 0)^2} - r \right)^2. \quad (3.1)$$

The point $\bar{x} = (0, 0)$ is a convex combination of two points $x' = (-r, 0)$ and $x'' = (r, 0)$ whereas

$$e(\bar{x}, p) = e\left(\frac{1}{2}x' + \frac{1}{2}x'', p\right) \neq \frac{1}{2}e(x', p) + \frac{1}{2}e(x'', p); \quad (3.2)$$

therefore $e(x, p)$ is nonconvex.

Note in the case that $r_i = 0$ for $i = 1, \dots, n$ the model (P_1) becomes

$$\min F(x) = \sum_{i=1}^n w_i \left((x_1 - a_i)^2 + (x_2 - b_i)^2 \right), \quad (3.3)$$

which is a convex function. The optimal solution of this problem is given with the following equations:

$$x_1 = \frac{\sum_{i=1}^n w_i a_i}{\sum_{i=1}^n w_i}, \quad x_2 = \frac{\sum_{i=1}^n w_i b_i}{\sum_{i=1}^n w_i}, \quad (3.4)$$

which is called the *center of gravity* (see, Love et al. [4]).

Since (P_1) is a minimun problem, one may guess that the optimal solution lies in the convex hull of demand points but small example with two points and different radius shows that it is not true. Example 4.2 also satisfies this claim. In the following we would show that the optimal solution of the problem (P_1) must lie in an extended rectangular hull of the existing points. The *rectangular hull* of a set of points is defined as the smallest rectangle (with sides parallel to the x and y axes) containing the set.

Definition 3.2. Let p_1, p_2, \dots, p_n be n points in the plane. We define the new points $RH_1 = (a_{\min}, b_{\min})$, $RH_2 = (a_{\min}, b_{\max})$, $RH_3 = (a_{\max}, b_{\max})$, and $RH_4 = (a_{\max}, b_{\min})$ whose coordinates are

$$\begin{aligned} a_{\min} &= \min\{a_i - r_i \mid i = 1, \dots, n\}, \\ a_{\max} &= \max\{a_i + r_i \mid i = 1, \dots, n\}, \\ b_{\min} &= \min\{b_i - r_i \mid i = 1, \dots, n\}, \\ b_{\max} &= \max\{b_i + r_i \mid i = 1, \dots, n\}. \end{aligned} \quad (3.5)$$

Lemma 3.3. Let p_1, p_2, \dots, p_n be n given points in the plane. The rectangular hull of the points RH_1, RH_2, RH_3, RH_4 which are defined as Definition 3.2, contains the optimal solution of the problem (P_1) .

Proof. Let R be the rectangular hull of the points RH_1, RH_2, RH_3 , and RH_4 , and let $x = (x_1, x_2)$ be a point out of R . We would show that x is not optimal. The proof is provided for the

case $x_1 > a_{\max}$; the other cases can be proved as the same. Let $x' = (a_{\max}, x_2)$; we show $F(x') < F(x)$. For any demand point $p_i = (a_i, b_i)$ the following inequality is held:

$$\begin{aligned} d(x, p_i) &= \sqrt{(x_1 - a_i)^2 + (x_2 - b_i)^2} \\ &> \sqrt{(a_{\max} - a_i)^2 + (x_2 - b_i)^2} = d(x', p_i) \geq r_i. \end{aligned} \quad (3.6)$$

Therefore since $w_i \geq 0$, then

$$\begin{aligned} F(x') &= \sum_{i=1}^n w_i \left(\sqrt{(a_{\max} - a_i)^2 + (x_2 - b_i)^2} - r_i \right)^2 \\ &< \sum_{i=1}^n w_i \left(\sqrt{(x_1 - a_i)^2 + (x_2 - b_i)^2} - r_i \right)^2 = F(x). \end{aligned} \quad (3.7)$$

So x cannot be an optimal solution and the proof is completed. \square

As we showed that the objective function of problem (P_1) is nonconvex, therefore using the methods of convex optimization to solve this problem is useless; hence we employ a heuristic method which is appropriate for our problem and give a global optimal solution with arbitrary accuracy. In the next section we introduce this method which is called big square small square (BSSS).

4. Big Square Small Square Algorithm

The big square small square (BSSS) method is a geometrical branch and bound algorithm originally suggested by Hansen et al. [5] to solve obnoxious facility location problem. The idea is to divide the plane into regions (squares), over each of which a lower (upper) bound for the problem is found. If the objective over a given square is worse than an existing upper (lower) bound, then that square is fathomed. The procedure continues until a prescribed tolerance is achieved. Another version of BSSS is later discussed by Hansen et al. [6]. Plastria [7] also presents a modified version of this algorithm. This approach has been used by McGarvey and Cavalier [8] to solve location problems with barrier and forbidden regions. More recently they have applied this method to find the location of competitive facilities [9]. Zaferanieh et al. [10] used this method to solve a single facility location problem in a special case that the plane R^2 has been divided into two regions with different norms by a straight line; here we also apply the BSSS algorithm to solve our problem.

By Lemma 3.3 we know that the optimal solution lies on the rectangular hull of RH_1, RH_2, RH_3 , and RH_4 . Based on this finding we employ the BSSS algorithm to obtain the optimal solution. At first the algorithm divides the rectangular hull, R , into four subrectangles, R_1, R_2, R_3 , and R_4 , by drawing the lines parallel to x and y axes through the middle of its sides. Then the algorithm selects a subrectangle for which its lower bound is the smallest. And a subrectangle for which the lower bound exceeds the value of the best known solution is fathomed. The process continues until the larger side of the subrectangle is less than the given tolerance, ε .

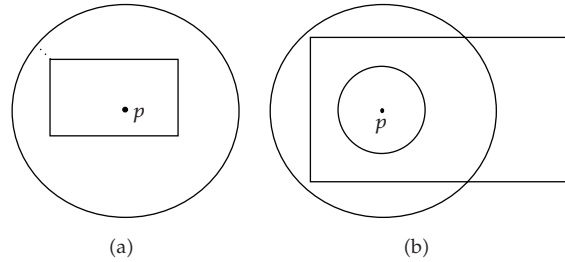


Figure 1: (a) The error is positive. (b) The error is zero.

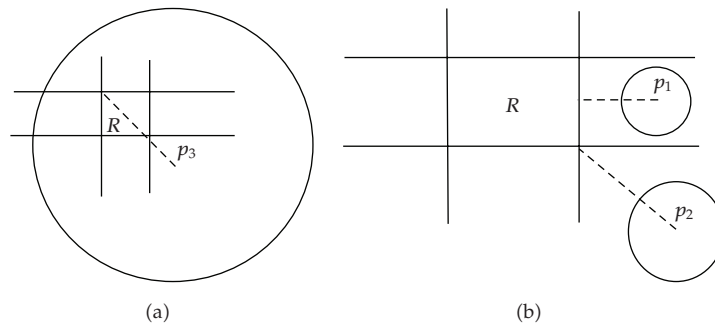


Figure 2: (a) The rectangle R is inside the circle. (b) The rectangle R is outside the circles.

The rectangle R is in fact an indicator of all the points inside it. The error of point p_i and rectangle R is considered instead of the errors of p_i and the points inside R . The error of a point p_i and a rectangle R is taken to be the square of distance between boundary of circle with center p_i and radius r_i and rectangle R . Note that since R is closed and convex, such point indeed exists.

For a usual implementation process of the BSSS algorithm the distance between points inside a rectangle is taken to be zero (see McGarvey and Cavalier [8, 9] and Zaferanieh et al. [10]) but in our approach the error value in this case is positive or zero. First let $p_i = (a_i, b_i)$ with relevant radius r_i be a point inside a rectangle. Two cases maybe occurred; if the circle with center p_i and radius r_i envelops the rectangle R , then the error is calculated as

$$e(p_i, R) = \min_{j=1, \dots, 4} (d(p_i, RH_j) - r_i)^2, \tag{4.1}$$

see Figure 1(a), where dash line shows the error. Otherwise, that is, when the circle crosses the circumference or lies inside a rectangle, the error is taken to be zero; see Figure 1(b).

To provide convincing explanation note that the rectangle which is employed to calculate lower bound originally can be considered as an extensive facility. And locating a facility on the boundary of a circle is ideal because the error does not occur. So in Figure 1(b) if all of a circle lies inside a rectangle or a circle crosses rectangle, then there is a point which lies on the area of rectangle and on the boundary of circle; consequently the error is zero.

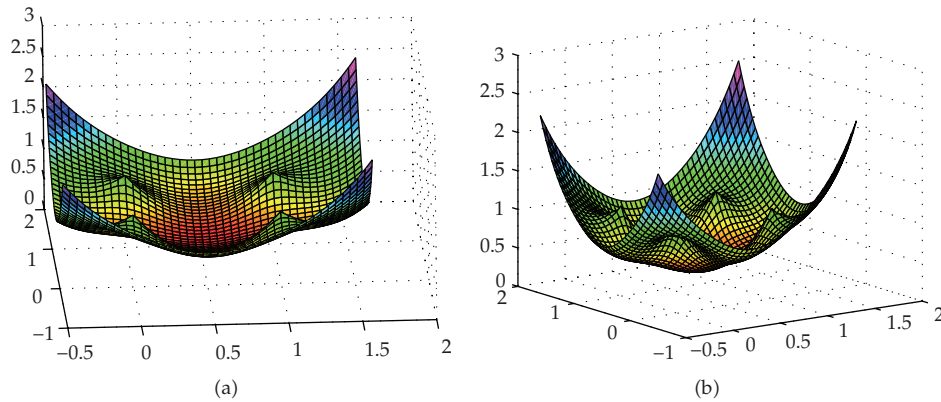


Figure 3: Two different perspectives of objective function for first case of Example 4.2.

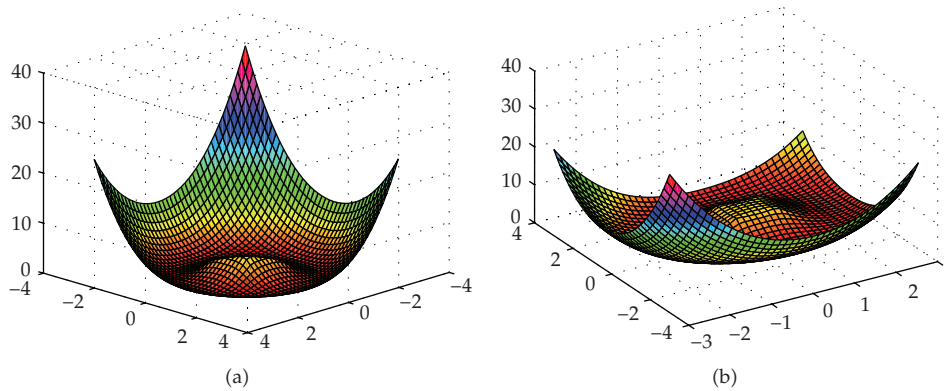


Figure 4: Two different perspectives of objective function for third case of Example 4.2.

Table 1: The results of Example 4.2.

case	radius	X_b	BSSS		CPU/sec	LINGO	
			lb	f_b		X	f
1	(1, 1, 1, 1)	(0.49, 0.51)	0.33	0.34	0.34	(1.13, 1.13)	1.06
2	(1, 2, 1, 2)	(-0.90, 0.51)	0.00	0.00	0.05	(-0.90, 0.50)	0.00
3	(2, 2, 2, 2)	(-1.42, 0.51)	0.91	0.93	2.53	(1.87, 1.87)	1.02

Table 2: The results of Example 4.3.

radius	X_b	lb	f_b	CPU/sec
case 1	(5.26, 4.62)	275.06	275.76	3.62
case 2	(5.25, 4.42)	181.40	181.94	3.73
case 3	(5.18, 4.69)	63.55	63.89	3.68

Table 3: The results of Example 4.4.

radius	X_b	lb	f_b	CPU/sec
case 1	(8.35, 7.71)	1635.40	1638.20	5.74
case 2	(8.38, 7.76)	1158.60	1161.43	4.91
case 3	(8.36, 7.76)	753.11	755.39	3.97

Table 4: The results of Example 4.5.

n	X_b	lb	f_b	CPU/sec
50	(27.57, 34.19)	33657.91	33674.61	31.54
100	(30.70, 31.58)	73774.99	73813.48	62.91
200	(31.11, 31.10)	151267.25	151348.92	129.43
300	(30.84, 31.73)	228371.47	228498.30	184.41
400	(31.28, 31.04)	300070.07	300238.98	255.33
500	(30.70, 30.83)	381561.67	381777.87	287.45

Table 5: The results of Example 4.5 with radius equal to zero.

n	Center of gravity	Objective function	BSSS
50	51202.22		51202.23
100	111032.93		111032.93
200	225490.50		225490.50
300	346396.00		346396.05
400	456804.57		456804.59
500	578703.13		578703.26

Now let p_i be a point with radius r_i outside the rectangle R . If the circle with center p_i and radius r_i crosses the rectangle R , then obviously the error is zero. Otherwise the error is calculated according to the following equation:

$$e(p_i, R) = \min_{RH_b \in \partial R} (d(p_i, RH_b) - r_i)^2, \quad (4.2)$$

where ∂R is the boundary of R . Note that the point that minimizes (4.2) could be either a corner point of R or a projection of p_i onto R ; see Figure 2.

The outline of the BSSS algorithm is described below. It is in essence similar to the one presented by McGarvey and Cavalier [8]. The output of algorithm is the approximated optimal solution.

Algorithm 4.1. (1) Find RH_1, \dots, RH_4 with respect to the Definition 3.2 and set R the rectangular hull of them.

(2) Set $L = 0$, $f_b = \infty$ (the best objective value obtained yet), and let $d^* = \max\{a_{\max} - a_{\min}, b_{\max} - b_{\min}\}$.

(3) Set $L = L + 1$, and divide R into four equal subrectangles R_{L1}, R_{L2}, R_{L3} , and R_{L4} .

(4) Calculate the objective value, f_{Lr} , at the midpoint of each subrectangle. If $\min_{r=1, \dots, 4} \{f_{Lr}\} < f_b$, then update f_b with this value.

(5) Calculate the lower bound value for each subrectangle, that is, $lb_{Lr} = \sum_{i=1}^n w_i e(p_i, R_{Lr})$, $r = 1, \dots, 4$. If $lb_{Lr} > f_b$, fathom R_{Lr} and set $lb_{Lr} = \infty$.

(6) Set $lb' = \min_{r=1, \dots, 4} \{lb_{Lr}\}$ and $r' = \arg \min_{r=1, \dots, 4} \{lb_{Lr}\}$. If $lb' = \infty$, go to step (8); else if $(0.5)^L d^* < \epsilon$, go to step (7); else set $R = R_{Lr'}$; fathom $R_{Lr'}$, set $lb_{Lr'} = \infty$, and go to step (3).

(7) If $lb' < f_b$, set $f_b = lb'$, define X_b as the center of subrectangle $R_{Lr'}$, and fathom this rectangle.

Table 6: Data for 18-point problem.

(x, y, w)	Radius	(x, y, w)	Radius	(x, y, w)	Radius
(1, 2, 3)	(1, 2, 3)	(4, 4, 1)	(1, 2, 3)	(7, 1, 2)	(1, 1, 3)
(1, 3, 2)	(1, 2, 3)	(4, 9, 2)	(1, 1, 3)	(7, 2, 3)	(1, 2, 3)
(2, 5, 1)	(1, 3, 3)	(5, 3, 2)	(1, 2, 3)	(8, 5, 1)	(1, 2, 3)
(3, 6, 3)	(1, 2, 3)	(5, 5, 1)	(1, 2, 3)	(8, 8, 3)	(1, 1, 3)
(4, 8, 2)	(1, 2, 3)	(6, 6, 3)	(1, 3, 3)	(9, 7, 3)	(1, 2, 3)
(4, 1, 3)	(1, 1, 3)	(6, 3, 3)	(1, 1, 3)	(9, 6, 2)	(1, 3, 3)

(8) Set $L = L - 1$; if $L = 0$, go to step (9); else if unfathomed subrectangle at level L are found choose the one with the most favorable lb ; denote it as R and return to step (3). Else repeat step (8).

(9) Terminate the algorithm with the new facility location X_b and having the objective function value f_b .

4.1. Computational Results

In this section we show the efficiency of BSSS algorithm by giving four examples. The first example is small and contains just four points. We give this example to compare the results of BSSS algorithm with those obtained by LINGO software and show that while the NLP solver software may be trapped on a local optimum, the BSSS method could be an appropriate method. The second and third examples are presented with the coordinates, weights, and radius of points to make it possible to compare the obtained results of this method with other methods in the future works. And finally the last one which contains more points is presented to become the CPU time of BSSS method comparable with other methods.

The above algorithm was written in MATLAB and run on a PC with *Pentium IV* processor and 1 GB of RAM and CPU with 2 GHz. The tolerance for all the problems was taken to be $\epsilon = 0.01$ and the results are shown with two decimal points of accuracy.

Example 4.2. The first example contains four points $(0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$. The relevant weights for all the points are taken to be 1. The results those obtained by BSSS method and LINGO software are given in Table 1. The CPU time of LINGO software for three cases is zero; however in the first and second cases it could not find the global minimum.

As the results show that the solutions in the second and third cases do not lie on the convex hull of existing points, however as we expected, they are inside the extended rectangular hull of these points. Figures 3 and 4 show two perspectives of objective function in the first and third cases, respectively. As figures show objective functions are not convex.

Example 4.3. The second example contains 18 points which are randomly generated and are presented in Table 6. The results are given in Table 2. The column under heading radius in this table indicates the relevant radius of points. The case 1, case 2, and case 3 indicate the first, second, and third components of the column under heading radius in Table 6, respectively, that are taken as the radius of points.

Table 7: Data for 30-point problem.

(x, y, w)	Radius	(x, y, w)	Radius	(x, y, w)	Radius
(1, 3, 3)	(1, 2, 3)	(6, 11, 1)	(1, 2, 3)	(11, 4, 2)	(1, 1, 3)
(1, 4, 2)	(1, 2, 3)	(7, 14, 2)	(1, 1, 3)	(11, 13, 3)	(1, 2, 3)
(2, 15, 1)	(1, 3, 3)	(7, 15, 2)	(1, 2, 3)	(13, 3, 1)	(1, 2, 3)
(2, 4, 3)	(1, 2, 3)	(8, 3, 1)	(1, 2, 3)	(13, 7, 3)	(1, 1, 3)
(3, 6, 2)	(1, 2, 3)	(8, 6, 3)	(1, 3, 3)	(14, 15, 1)	(1, 2, 3)
(3, 10, 2)	(1, 2, 3)	(8, 5, 1)	(1, 3, 3)	(14, 3, 1)	(1, 2, 3)
(3, 2, 1)	(1, 2, 3)	(8, 2, 1)	(1, 2, 3)	(14, 1, 2)	(1, 2, 3)
(4, 6, 1)	(1, 2, 3)	(9, 11, 2)	(1, 3, 3)	(15, 8, 3)	(1, 2, 3)
(4, 3, 2)	(1, 2, 3)	(10, 8, 1)	(1, 3, 3)	(15, 10, 3)	(1, 2, 3)
(6, 8, 3)	(1, 1, 3)	(10, 10, 3)	(1, 1, 3)	(15, 15, 2)	(1, 3, 3)

Example 4.4. As the third example we consider a problem with 30 points. The relevant data and results are given in Tables 7 and 3, respectively. The column under heading radius in Table 3 indicates the radius as described in Example 4.3.

Example 4.5. Finally we consider the problems with 50, 100, ..., 500 points. The points and their weights and radius all are integers and randomly generated in the intervals $[1, 60]$, $[1, 3]$, and $[1, 10]$, respectively. The problems with 100–500 points contain the points of smaller problems. Table 4 shows the results.

In order to be able to make a better judgment about the efficiency of the proposed algorithm we have tried to solve the problems in Example 4.5 with radius equal to zero for all the points. As we mentioned before the optimal solution of the problem in this case is the center of gravity that is obtained by (3.4). The results are shown in Table 5.

5. Summary and Conclusion

We considered the problem of finding the location of a single facility such that the sum of the weighted square errors over all points is minimized. We showed that the problem in general is nonconvex and then proved that the optimal solution lies in an extended rectangular hull of the existing points. Based on this finding then a big square small square (BSSS) was applied to solve the problem.

Other nonconvex solver methods such as big triangle Small triangle (BTST) method of Drezner and Suzuki [11] can be used to solve this problem which may results in better solutions.

Appendix

See Tables 6 and 7.

Acknowledgment

The authors would like to express their sincere gratitude to the anonymous referees for their careful reading of the manuscript and their constructive comments which resulted in the improvement of the paper.

References

- [1] P. B. Mirchandani and R. Francis, *Discrete Location Theory*, John Wiley & Sons, New York, NY, USA, 1990.
- [2] Z. Drezner, S. Steiner, and G. O. Wesolowsky, "On the circle closest to a set of points," *Computers and Operations Research*, vol. 29, no. 6, pp. 637–650, 2002.
- [3] J. Brimberg, H. Juel, and A. Schöbel, "Locating a minisum circle in the plane," *Discrete Applied Mathematics*, vol. 157, no. 5, pp. 901–912, 2009.
- [4] R. F. Love, J. G. Morris, and G. O. Wesolowsky, *Facilities Location: Models and Methods*, North-Holland, Amsterdam, The Netherlands, 1988.
- [5] P. Hansen, D. Peeters, and J. F. Thisse, "On the location of an obnoxious facility," *Sistemi Urbani*, vol. 3, pp. 299–317, 1981.
- [6] P. Hansen, D. Peeters, D. Richard, and J.-F. Thisse, "The minisum and minimax location problems revisited," *Operations Research*, vol. 33, no. 6, pp. 1251–1265, 1985.
- [7] F. Plastria, "GBSSS: the generalized big square small square method for planar single-facility location," *European Journal of Operational Research*, vol. 62, no. 2, pp. 163–174, 1992.
- [8] R. G. McGarvey and T. M. Cavalier, "A global optimal approach to facility location in the presence of forbidden regions," *Computers and Industrial Engineering*, vol. 45, no. 1, pp. 1–15, 2003.
- [9] R. G. McGarvey and T. M. Cavalier, "Constrained location of competitive facilities in the plane," *Computers and Operations Research*, vol. 32, no. 2, pp. 359–378, 2005.
- [10] M. Zaferanieh, H. Taghizadeh Kakhki, J. Brimberg, and G. O. Wesolowsky, "A BSSS algorithm for the single facility location problem in two regions with different norms," *European Journal of Operational Research*, vol. 190, no. 1, pp. 79–89, 2008.
- [11] Z. Drezner and A. Suzuki, "The big triangle small triangle method for the solution of nonconvex facility location problems," *Operations Research*, vol. 52, no. 1, pp. 128–135, 2004.