ORIGINAL PAPER

Open Access

Two lower-bounding algorithms for the p -center problem in an area

Yanchao Liu

Abstract

The p -center location problem in an area is an important yet very difficult problem in location science. The objective is to determine the location of p hubs within a service area so that the distance from any point in the area to its nearest hub is as small as possible. While effective heuristic methods exist for finding good feasible solutions, research work that probes the lower bound of the problem's objective value is still limited. This paper presents an iterative solution framework along with two optimization-based heuristics for computing and improving the lower bound, which is at the core of the problem's difficulty. One method obtains the lower bound via solving the discrete version of the Euclidean p -center problem, and the other via solving a relatively easier clustering problem. Both methods have been validated in various test cases, and their performances can serve as a benchmark for future methodological improvements.

Keywords: p -center location problem, Computational geometry, Nonconvex optimization, Area coverage

1 Introduction

Let us look at this problem: given a compact polygonal region $R \subset \mathbb{R}^2$ on the Euclidean plane, the goal is to determine the coordinates of p landmark points so that for any point in the region, its distance to the nearest landmark is minimized. Let S_p denote the collection of all possible sets of p landmarks selected from the region R , i.e., $S_p = \{(s_1, \dots, s_p) \mid s_i \in R, i = 1, \dots, p\}$, then the desired distance $D(R, p)$ is expressed as

$$D(R, p) = \min_{S \in S_p} \max_{u \in R} \min_{s \in S} \|u - s\|_2 \quad (1)$$

where $\|u - s\|_2$ denotes the Euclidean distance between point u and point s . The problem is equivalent to determining the center of p circles such that all points in region R are covered by at least one circle and that the radius of the largest circle is minimized. It is an extension of the Euclidean p -center problem in which the region R is represented by a finite set of demand points, see, e.g., (Çalik et al. 2019). Suzuki and Drezner (1996) are the

first to study this problem and they termed it as the “ p -center location problem in an area”. In that paper, the authors proposed a Voronoi heuristic together with a “finishing up” algorithm (we call this algorithmic bundle *Voronoi heuristics* hereafter) to efficiently find good feasible solutions. After going through all papers citing that paper we can conclude that Suzuki and Drezner's original Voronoi heuristics method remains to be the best known heuristic framework for finding good feasible solutions (i.e., upper bounds) for $D(R, p)$ when R is a convex polygon. Wei et al. (2006) extended the Voronoi heuristic to account for cases where R is nonconvex. Key to this extension was an exact algorithm for finding constrained minimum covering circle (CMCC) when the one-center subproblem in the Voronoi heuristic becomes infeasible, i.e., when the found circle center lands outside the outer polygon or inside a hole. However, the CMCC algorithm was brute-force and inefficient. Liu (2021a) developed a more efficient CMCC algorithm which further reduced the computing time by about two orders of magnitude.

Despite all these advancements, there has been a lack of investigation for computing the lower bound of $D(R, K)$.

Correspondence: yanchaoliu@wayne.edu

Industrial & Systems Engineering, Wayne State University, 4815 4th Street, Rm 2169, Detroit 48201, MI, USA

Fig. 1 Left: The Belle Isle island located between Detroit, USA to the north and Windsor, Canada to the south. Right: A 7-center solution covering the island's convex envelope. The circle radius is 665 m, meaning that a drone operating over this 1000-acre island is within 665 m of a safe landing spot at all times

Without a good lower bound, the quality of the heuristic solutions, which only provide upper bounds for the optimal value, cannot be quantified¹. In this paper, we propose and compare two methods for computing and iteratively improving the lower bound of $D(R, p)$, under the condition that R is a convex polygonal region. The methods and their minor variants attempted in this paper all have some intuitive appeal, yet their performances differ substantially. The modeling and computational techniques documented in this paper may serve as a stepping stone, or lessons learned, for future endeavors toward discovering more efficient methods.

1.1 Motivating applications

The p -center problem in an area has many applications. Traditional applications include covering an area with p hospitals, cellular transmitters (Gonzalez-Brevis et al. 2011; McGregor and Shen 1977), warning sirens (Murray and Wei 2013), sprinklers, Doppler radar stations (Murray et al. 2008), and air defense missiles (Brown et al. 2005). A new application, which inspired our interest, is the battery depot location problem for a drone delivery network (Liu 2019; 2021b). While performing point-to-point delivery services, a delivery drone can be anywhere within the service region when an in-air contingency arises. In a contingency, the drone must fly (presumably along a straight line) to the nearest depot and land there. We call the longest distance to the nearest depot the *failsafe distance*, see also Liu (2021a). It is an important input for determining tactical parameters of drones in the fleet, such as battery failsafe thresholds, permissible trajectories, and acceptable task assignments. Here, $D(R, p)$ corresponds to

the minimum possible failsafe distance in a service region R in which p depots are to be installed. Finding $D(R, p)$ and locating the depots accordingly will enable the most efficient use of batteries under safety constraints. Another motivating application is the base station location problem for 5G cellular networks, see related work in Wang et al. (2020); Ganame et al. (2021). It is important to determine the number, specification and locations of the base stations (e.g., transceiver device) to ensure robust coverage of the service area while minimizing the fixed and variable costs of operation. Low-power transmitters have a small coverage radius so a great number of devices are needed to ensure area coverage, leading to a high setup cost, while transmitters having a larger coverage radius consume substantially more power, leading to high operation costs. The optimal decision can be formulated as an instance or a close variant of problem (1). Figure 1 illustrates covering Detroit's Belle Isle island at the US-Canada border with seven depot locations. The convex envelope of the island is substituted in the computational algorithms, since we focus on convex-shaped regions in this paper.

The remainder of the paper is organized as follows. In Section 2, we provide a brief review of the related literature. Our main contribution is developed in Section 3 and validated numerically in Section 4. In particular, we proposed an iterative framework to find $D(R, p)$, formulate several subproblems for use in the framework, and develop a cluster-based algorithm with several enhancements to improve the computational efficiency. To the author's knowledge, the proposed methods are novel and the reported optimality gaps for the stylized cases are the first, thus the best, gaps known achievable by two hours' computation on a single CPU core. Section 5 concludes the paper with pointers for future work.

¹The roles of upper and lower bounds in solving optimization problems are discussed in various textbooks, for instance, in Chapter 2 of Wolsey (2020).

2 Related problems and literature

Determining the location of service depots falls in the class of facility location problems (Aikens 1985; Daskin 1995; ReVelle and Laporte 1996; Klose and Drexl 2005; Farahani et al. 2012; ReVelle et al. 1976), more specifically the *continuous location problems* (Rosing 1992; Brimberg and Salhi 2005; Berman et al. 2009; Blanco et al. 2014; Blanco et al. 2016), the *multisource Weber problem* (Wesolowsky 1993; Hansen et al. 1998a; Brimberg et al. 2000; Megiddo and Supowit 1984; Raeisi Dehkordi 2019), and the *Euclidean p -center problem* (Megiddo and Supowit 1984).

The Weber's Problem. It is a classic location problem that requires finding a point in the plane to minimize the weighted sum of the Euclidean distances from this point to n destination points (Tellier 1972; Cooper and Katz 1981). Mathematically, it is written as

$$\text{minimize}_{z,t} \sum_{1 \leq i \leq n} w_i [(x_i - z)^2 + (y_i - t)^2]$$

where (x_i, y_i) and w_i are the coordinate and weight of the destination point i , for $i = 1, \dots, n$. In general, the weight can be either positive or negative. Since a positive weight forms an attractive effect and negative weight forms a repulsive effect, the problem is also called the *attraction-repulsion problem*, a nonconvex optimization problem. Chen et al. (1992) developed an algorithm based on outer-approximation and vertex enumeration and presented computational experience with n up to 1,000. Maranas and Floudas (1994) developed global optimization algorithms for the Weber's problem based on rectangle-subdivision lower-bounding operations and presented computational results for instances with $n = 10,000$. Today, off-the-shelf global optimization solvers (e.g., SCIP and GLoMIQO) can solve instances with $n = 100,000$ consistently within one second on a desktop computer².

The Multisource Weber Problem (MWP). As an extension to the Weber's problem, the MWP concerns finding the coordinates of K depots from a continuous feasible space to minimize the sum of weighted distances from a finite set of destinations to their closest depot (Kuenne and Soland 1972). It is written as

$$\text{minimize}_{z,t \in \mathbb{R}^p} \sum_{1 \leq i \leq n} \min_{1 \leq j \leq p} \{w_{ij} [(x_i - z_j)^2 + (y_i - t_j)^2]\}$$

The MWP is NP-hard (Megiddo and Supowit 1984), and can be formulated as a mixed integer nonlinear program. One of the most widely used heuristics for finding good feasible solutions is the alternate location-allocation

method developed by Cooper (1964). The idea is to randomly divide the destination set into p subsets and for each subset find a depot location by solving the (single-source) Weber's problem, then, if any destination point is closer to another depot than the one to which it was originally allocated, adjust the allocation and solve the Weber's problem again. The process is repeated until no reallocation is needed. Though the process may not necessarily converge to the optimal solution, it is a highly effective heuristic. The same design idea can also be found in the Voronoi heuristics (see Suzuki and Drezner (1996)) and the more famous K-means clustering algorithms. Other well-documented methods for solving MWP include Tabu search (Ohlemüller 1997), p -Median heuristic (Hansen et al. 1998b), variable neighborhood search (Hansen et al. 2008), genetic algorithms (Salhi and Gamal 2003; Brimberg et al. 2000), convex hull enumeration (Rosing 1992), and the cellular heuristic (Gamal and Salhi 2003).

The Euclidean p -center (EPC) Problem. Given a finite set of demand points, the p -center problem is to find p depots to minimize the maximum distance from any demand point to its respective nearest depot. Specifically, given a set $X = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ of demand points on the plane, the problem is to find p points with coordinates $\{(z_1, t_1), (z_2, t_2), \dots, (z_p, t_p)\}$ to minimize

$$\max_{1 \leq i \leq n} \min_{1 \leq j \leq p} \{(x_i - z_j)^2 + (y_i - t_j)^2\}. \quad (2)$$

The EPC problem is NP-hard. Furthermore, it is shown that an approximate solution less than $2/\sqrt{3}$ times the optimal is necessarily an optimal solution (Megiddo and Supowit 1984). This indicates that it is even NP-hard to approximate the p -center problem with a relative error of less than about 15%. Drezner (1984) proposed two heuristics and an optimal algorithm for solving the EPC problem. The optimal algorithm ingeniously exploited the geometric properties of the optimal solution and boasted a time complexity of $O(n^{2p+1})$, that is, for a given p , the time is polynomial in n . Callaghan et al. (2017) recently improved the efficiency of Drezner's original algorithm via performing opportunistic neighborhood reductions, which was able to reduce up to 96% of the computation originally needed for certain instances.

The p -center problem in an area is an extension of the Euclidean p -center problem: instead of covering a finite set of demand points in an area, the minimum-radius circles centering at the p depot points must cover the whole area. This is an even harder problem than the discrete version. It is also known that there does not exist any approximation algorithm for the p -center problem in a general metric space with a constant less than 2 unless $P=NP$ (Hochbaum 1997; Carlsson et al. 2016). In many location analysis contexts, it is indeed more suitable to treat demand as continuously distributed in a

²This claim is supported by our own experimentation: Run 20 randomly generated cases with $n = 100,000$ using GAMS (option opcr=0, optca=0, thread=1) with SCIP and GLoMIQO solvers. Solutions from both solvers match and all cases were solved within one second.

region than as discrete points, see Yin and Mu (2015) for a tradeoff approach. For instance, Drezner et al. (2019) proposed a competitive facility location model and algorithms to obtain a more accurate estimate of the buying power attracted to a facility by assuming demand is continuous over an area rather than concentrated at demand points. Baron et al. (2007) investigated the problem of locating M facilities on the unit square so as to minimize the maximal demand faced by each facility subject to closest assignments and coverage constraints, where demand was assumed to be uniformly distributed in the unit square. The authors developed upper and lower bounds on the feasibility of the problem for a given number of facilities and coverage radius, and proposed an effective heuristic method capable of achieving an average optimality gap of around 4%. Aronov et al. (2009) studied the problem of how to divide a region into m equal-area subregions, each to be served by a facility of a given location, so that the average distance between a point in the region and the facility that serves it is minimal. The authors proposed constant-factor approximation algorithms for convex polygonal regions, and suggested that the optimal partition is always induced by an additive-weighted Voronoi diagram for some appropriate choice of weights. Along a similar line, Carlsson et al. (2016) developed fast approximation algorithms for the problem of how to divide a convex region into n subregions to minimize the maximum distance between points of adjacent subregions. The distance being minimized was called the connectivity radius, as in the context where each subregion is to be served by a vehicle and the n vehicles, wherever they are within their respective subregions, must form the node set of a connected graph whose maximum link length is equal to the connectivity radius. Dinler et al. (2015) studied the problem of serving m convex polygonal demand regions using q facilities to minimize the weighted sum of squares of the maximum Euclidean distances between points in the demand regions and the facilities they are assigned to. The problem was formulated as a mixed-integer second-order cone problem (SOCP) and three heuristics were developed to solve the problem.

As suggested by Wei and Murray (2015), an ingenious approach for solving continuous space coverage problems is abstracting continuously distributed demand as discrete spatial objects (points, lines or polygons), generating a finite dominating set from continuous space as candidate facility sites, and applying a discrete location coverage model. The idea, i.e., approximating the coverage of an area by covering selected points within the area, will be used in this paper to form lower-bounding subproblems in an iterative framework.

Related to the lower bound of the p -center solution is the p -dispersion problem, which attempts to place p

points in an area such that they are maximally dispersed. Adopting the Euclidean distance, the problem is equivalent to packing p non-overlapping equal circles in the area while maximizing the circle radius. A formulation will be provided in Section 3. A moment of contemplation would reveal that the maximal radius for the $(p + 1)$ -dispersion problem for area R is a lower bound to $D(R, p)$. This observation was also made by Shier (1977) and Drezner and Erkut (1995). Drezner and Erkut (1995) formulated the p -dispersion problem as a nonlinear program and reported solutions for $p = 15, 17, 21$ and 22 in a unit square, and Nurmela and Ostergård (1999) reported the optimal solutions in a square for p up to 27 . The lower bound that the p -dispersion problem provides for the p -center problem is a static one and is rather weak, especially when p is small. Furthermore, the p -dispersion problem is itself computationally challenging to solve, so other lower-bounding methods are needed for the latter. There have been practically relevant works to formulate and solve different variants of the p -center problem, e.g., the tree network version proposed by Chandrasekaran and Daughety (1981), the spatially restricted version proposed by Wei et al. (2006) and the boundary-adhering version proposed by Du and Xu (2014). However, to the knowledge of the author, no effort is known to tackle the lower-bounding problem for the original p -center problem, a gap that this paper aims to close.

3 An iterative approach for computing the lower bound

Given a set of candidate coordinates of the p centers, $S = \{(s_1, t_1), \dots, (s_p, t_p)\}$, the minimum circle radius to cover the polygon R can be retrieved by solving the following nonlinear program (NLP).

MD(S):

$$\begin{aligned} \text{Maximize}_{x,y,d} \quad & d, \text{ subject to } d \leq \sqrt{(x - s_i)^2 + (y - t_i)^2}, \\ & i = 1, \dots, p \text{ and } (x, y) \in R \end{aligned} \quad (3)$$

In words, the problem MD(S) attempts to find the maximum distance from any point in R to its nearest center in the set S . Let us denote the optimal solution by (x^*, y^*, d^*) . The optimal value d^* will be the minimum common radius of the p circles located at the given center coordinates that are large enough to cover the region R - the whole region is covered because the farthest point (x^*, y^*) is covered, and the radius d^* is minimal because anything smaller will leave the point (x^*, y^*) uncovered. MD(S) is a nonconvex NLP. Due to the small size (i.e., three variables, p quadratic inequalities and q linear inequalities, where q is the number of sides of the polygon R), the problem can be solved very quickly by a global optimization solver such

as BARON (Sahinidis 1996), SCIP (Achterberg 2009) and GLoMIQO³ (Misener and Floudas 2013). Alternatively, the problem MD(S) can also be solved without invoking a numeric optimization solver. Specifically, we can determine the Voronoi diagram generated by the p given center points $\{(s_i, t_i) : i = 1, \dots, p\}$ and let $V_i \subset R$ be the Voronoi cell belonging to point (s_i, t_i) . Then by definition of the Voronoi diagram, every point from V_i is at least as close to (s_i, t_i) than to any other point (s_j, t_j) , $j \neq i$. The most distant point $(x_i, y_i) \in V_i$ is then necessarily an extreme point of the convex polygon V_i , and hence the solution, i.e., (x^*, y^*) , to MD(S) must be one of the p such points (x_i, y_i) . The solution to MD(S) corresponding to any set S will provide an upper bound for $D(R, p)$.

On the other hand, the optimal solution to the Euclidean p -center (EPC) problem for a finite set of points, e.g. X , in R will provide a lower bound for $D(R, p)$. The formulation is given in equation (2). Let us denote this problem as EPC(X), which is NP-hard (Megiddo and Supowit 1984). Although heuristic methods exist for finding good feasible solutions, only the (square root of the) global optimal value of EPC(X), denoted by $r^*(X)$, can serve as a valid lower bound for $D(R, p)$. As will be shown in the experiments, the difficulty of solving the EPC problem will increase quickly as the number of pivot points increases. The number of pivot points n is the dominating factor that determines the solution time of EPC instances, whereas the problem formulation and solution algorithms play a secondary role. Therefore, selecting pivot points parsimoniously to “maximally represent” the region R is important. Given a computing budget for solving an EPC(X) instance with n pivot points, for example, the set X (among infinitely many sets of n points selected from R) that would yield the largest optimal value for the EPC problem should be used, as it would provide the tightest lower bound for $D(R, p)$. However, finding such an X amounts to an inconceivable task. The pivot points have to be selected heuristically. A useful property of EPC(X) is that adding more points to X can only increase its optimal objective value. Formally, define

$$r^*(X) := \arg\min_{S \in \mathcal{S}_p} \max_{u \in X} \{\min_{s \in S} \|u - s\|_2\}$$

where $\mathcal{S}_p$ is defined in the context of (1), then we have the following proposition.

Proposition 1 $r^*(X) \leq r^*(X')$ if $X \subset X'$.

Proof Suppose S' is the solution (i.e., set of p points) to EPC(X'), that is, $r^*(X') = \max_{u \in X'} \{\min_{s \in S'} \|u - s\|\}$.

Therefore, we have

$$\begin{aligned} r^*(X) &= \text{Minimize}_{S \in \mathcal{S}_p} \max_{u \in X} \{\min_{s \in S} \|u - s\|\} \\ &\leq \max_{u \in X} \{\min_{s \in S'} \|u - s\|\} \\ &\leq \max_{u \in X'} \{\min_{s \in S'} \|u - s\|\} \\ &= r^*(X') \end{aligned}$$

where the first inequality holds because the left side is the minimum over all possible p -point sets while the right side is for a given p -point set S' , the second inequality holds because $X \subset X'$ and that the maximum elements in a set is at least as big as the maximum elements in a subset of that set. $\square$

We propose the following algorithmic framework for finding $D(R, p)$.

Algorithm 1 Iterative Algorithm for Approximating $D(R, p)$

- 1: Choose an initial set of pivot points X from region R
 - 2: **repeat**
 - 3: Solve EPC(X) to obtain centers S and the minimal radius r^*
 - 4: Solve MD(S) to obtain solution (x^*, y^*, d^*)
 - 5: $X \leftarrow X \cup \{(x^*, y^*)\}$
 - 6: **until** $d^* - r^* < \epsilon$
-

The lower bound r^* can only increase (or remain the same) from one iteration to the next, because of the growing set of pivots to be covered. The upper bound d^* , though will vary without any monotonicity, is globally upper bounded. The discrete set X will increasingly resemble R as more points are added to it, so r^* will approach $D(R, p)$ and therefore the process outlined in Algorithm 1 will terminate for a given $\epsilon > 0$.

The framework leaves much room for how it is implemented, including how to create the starting pivots and how to formulate and solve the optimization problems in the loop. We will analyze different alternatives below.

3.1 Formulations of the EPC problem

Because the EPC problem itself is NP-hard, most work in the literature focused on developing heuristic algorithms to solve large instances efficiently. However, in the context of Algorithm 1 the EPC(X) should be solved globally in order for its objective value r^* to serve as a valid lower bound for $D(R, p)$. If EPC(X) is not solved globally, then the lower bound of its objective value must be known, which can also serve as a (weaker) lower bound for $D(R, p)$. In either case, a global optimization framework

³GLOMIQO does not handle general nonlinear expressions such as the square root function, but the square root in (3) is not essential and can be removed.

such as the branch-and-bound search must be used for solving the EPC problem, because heuristic methods do not provide lower bounds. Formulation (2) is not implementable due to the max-min structure. Here, we postulate two implementable formulations for EPC. Define the index sets $I := \{1, \dots, n\}$ and $K := \{1, \dots, p\}$, and binary variables s_{ik} to indicate if pivot i is assigned to circle k , for $i \in I$ and $k \in K$, and suppose that the polygon R has a set F of facets, each having the normal direction (a_f, b_f) and the directional distance to the origin c_f .

EPC formulation given $(x_i, y_i), i \in I$:

$$\text{Minimize } r^2 \quad (4)$$

$$\text{s.t.} \quad \begin{cases} r^2 \geq \sum_{k \in K} s_{ik} ((x_i - z_k)^2 + (y_i - t_k)^2), \forall i \in I & \text{for MINLP} \\ r^2 \geq -M_i(1 - s_{ik}) + (x_i - z_k)^2 + (y_i - t_k)^2, \forall i \in I, k \in K & \text{for MIQCP} \end{cases} \quad (5)$$

$$\sum_{k \in K} s_{ik} = 1, \forall i \in I \quad (6)$$

$$s_{ik} \in \{0, 1\}, \forall i \in I, k \in K \quad (7)$$

Both formulations minimize the maximum (squared) distance between a pivot point and its assigned circle center. The MINLP model involves integer variables and general nonlinear expressions, whereas the MIQCP formulation eliminates the products of binary variables and quadratic expressions by introducing big M constants. The constant M_i can be set to the squared distance between pivot i and the corner point of the polygon R that is farthest from pivot i . This will guarantee that $\sqrt{M_i}$ is no less than the distance from pivot i to any point in R , serving its purpose in the formulation. As we will see in numerical experiments, the two formulations differ substantially in solvability when attempted by a stable of available solvers.

3.2 Clustering method for the lower bound computation

Unless in special cases such as covering a square area with four circles, the optimal locations (not considering permutation) of the circles are not unique. At an optimal solution, it is usually the case that all but one of the p centers can move freely in a neighborhood and remain optimal. This makes the problem highly nonsmooth thus extremely difficult to solve by a nonlinear solver. To ameliorate this difficulty, we propose to substitute a simpler problem for EPC in the lower-bounding step (line 3 of Algorithm 1), described as follows: (1) Given the n pivot points (i.e., $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$), calculate the point-to-point distances and store them in an upper triangular matrix. Specifically, define and calculate $d_{ij} := [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2}$ for $i, j \in I$ and $i < j$; (2) assign these n points to p clusters to minimize the maximum within-cluster point-to-point distance. Let binary variables δ_{ij} indicate whether point i and point j belong to the same cluster, and binary variables α_{ik} indicate whether

point i is assigned to cluster k , then the clustering problem is formulated as follows.

Cluster(X):

$$\text{Minimize } l \quad (8)$$

$$\text{s.t.} \quad l \geq d_{ij}\delta_{ij}, \quad i, j \in I, i < j \quad (9)$$

$$\sum_{k=1}^p \alpha_{ik} = 1, \quad \forall i \in I \quad (10)$$

$$\delta_{ij} \geq \alpha_{ik} + \alpha_{jk} - 1, \quad \forall i, j \in I, i < j \quad (11)$$

If an upper bound for $D(R, p)$, denoted by $\bar{d}$, is known, we can fix the value of the variable δ_{ij} to 0 for those $(i, j), i < j$ pairs having $d_{ij} > \bar{d}$ to reduce the effective size of the problem. The objective value l represents the “diameter” of the largest cluster. This is a simpler problem than EPC, since the locational aspect (i.e., where to place the centers) is out of the scope. Furthermore, the optimal value of Cluster(X), denoted by l^* , will provide a valid lower bound for $D(R, p)$, as shown below.

Proposition 2 *Given a set of pivot points X , let l^* denote the optimal objective value of Cluster(X) and r^* the positive square root of the optimal objective value of EPC(X), then $r^* \geq l^*/2$.*

Proof Assume for contradiction that $r^* < l^*/2$ and the corresponding assignment solutions are s_{ik}^* and α_{ik}^* , respectively for the EPC and Cluster problem. Now consider another candidate solution $\{\hat{\alpha}_{ik}\}$ of Cluster(X) with values $\hat{\alpha}_{ik} = s_{ik}^*, i \in I$ and $k \in K$. It is first of all feasible because constraint (10) is satisfied since constraint (6) is satisfied. The associated δ_{ij} values are $\hat{\delta}_{ij} = \sum_{k=1}^p s_{ik}^* s_{jk}^*$. Then we must have $\hat{\delta}_{ij} = 0$ for any pair of points (i, j) that are more than $2r^*$ apart, because such a pair of points could not be covered by the same circle of radius r^* . Given this condition, if one were to minimize l subject to (9), one would get an optimal value no more than $2r^*$. This contradicts the original assumption that $l^* > 2r^*$. $\square$

Once the p clusters are formed, we can determine the cluster centers by solving a smallest enclosing circle (SEC) problem (Shamos and Hoey 1975; de Berg et al. 2008) separately for each cluster. The SEC problem is also referred to as the Euclidean delivery boy problem (Elzinga and Hearn 1972a) and the minimum covering sphere problem (Elzinga and Hearn 1972b). Specifically in our case, for cluster k that has n_k points with coordinates $\{(x_1, y_1), \dots, (x_{n_k}, y_{n_k})\}$, the problem is to find the circle of smallest radius that encloses these points, formulated as follows.

SEC(k):

$$\text{Minimize}_{(r_k, z_k, t_k)} r_k \text{ subject to } r_k \geq \left[(z_k - x_i)^2 + (t_k - y_i)^2 \right]^{1/2}, \quad (12)$$

$$i = 1, \dots, n_k$$

This is a convex optimization problem because the feasible region is the intersection of n 3-dimensional quadratic cones (MOSEK 2019), which is convex. A nonlinear optimization solver such as CONOPT (Drud 1985; CONOPT Solver Manual in GAMS Documentation 2020) can efficiently solve this problem. Note that algorithms that do not rely on a numerical optimization routine are also available, see, e.g., Elzinga and Hearn (1972a) and de Berg et al. (2008). In particular, Megiddo (1983) provided a linear time algorithm (in the context of formulation (12), an $O(n_k)$ algorithm) for this problem, and Dyer (1986) developed a linear time algorithm for the weighted case. Therefore, SEC(k) is an easy problem to solve. Let the optimal solution of SEC(k), $k \in K$, be denoted by (r_k^*, z_k^*, t_k^*) . It is evident that the radius of the largest circle, $\bar{r} = \max \{r_1^*, \dots, r_p^*\}$, together with the SEC centers $\{(z_1^*, t_1^*), \dots, (z_p^*, t_p^*)\}$ forms a feasible solution to EPC(X). Since r^* is the optimal solution to EPC(X), we have $\bar{r} \geq r^*$. It is worth noting that there is usually a lot of leeway in deciding the circle center locations S while keeping all pivot points covered according to the clustering solution - after all, the circle radius of cluster k can be set to that of the largest circle, not necessarily to r_k^* .

We can set $S = \{(z_1^*, t_1^*), \dots, (z_p^*, t_p^*)\}$ and solve MD(S) to propose a new pivot point. However, if the optimal objective of MD(S), d^* , turned out to be equal to $\bar{r}$, then the addition of the new pivot would not change the cluster solution. In this case, an alternative pivot point generation method should be used.

Analogous to solving MD(S) for identifying a new pivot point based on the EPC(X) solution, we can solve another optimization problem to identify a new pivot point based on the Cluster(X) solution, with additional properties. Specifically, it is desirable that the addition of the new point to X would maximally alter the cluster solution. First choose a positive scalar e , and for each existing pivot point $i \in I$, define a parameter g_i and set it equal to the maximum separation distance l_k of the cluster that i belongs to, that is, $g_i := l_k$ for each (i, k) having $\alpha_{ik}^* = 1$. We can solve the following MINLP problem to try to identify a new point (x', y') such that there is at least one existing pivot point from each cluster whose distance to (x', y') is greater than the cluster's maximum separation distance by a margin of e . Such a point could not be included in any existing cluster without changing the cluster's radius or changing the cluster membership. So the addition of this new point in X would force a change in the Cluster(X) solution in the next iteration.

NEWP(e):

$$\text{Maximize}_{x', y', w} \sum_{i \in I} w_i \quad (13)$$

$$\text{s.t.} \quad a_f x' + b_f y' \leq c_f \quad \forall f \in F \quad (14)$$

$$(x' - x_i)^2 + (y' - y_i)^2 \geq w_i (g_i + e)^2 \quad \forall i \in I \quad (15)$$

$$\sum_{i \in I} \alpha_{ik}^* w_i \geq 1 \quad \forall k \in \{1, \dots, p\} \quad (16)$$

$$w_i \in \{0, 1\} \quad \forall i \in I \quad (17)$$

The objective function (13) is a dummy one. In fact, any feasible solution to NEWP(e) would provide a new point (x', y') that meets the “at least one from each cluster” criterion above. The solution process is usually very fast by a global solver such as SCIP. In case the problem is infeasible, one can reduce the value of e and try again. If the problem remains to be infeasible for a small enough value of e , e.g., when $e < 0.0001$, it means that no new pivot points can be identified to alter the cluster solution. The cluster radius $l^*/2$ returned at this point would be the best lower bound achievable by the Cluster-based method.

Solving Cluster(X) in lieu of solving EPC(X) amounts to a relaxation in the lower bounding procedure, and the lower bound thus obtained will not be as tight. Two questions are important: (1) How much worse can the surrogate bound $l^*/2$ be compared to that obtained by solving the EPC problem? (2) Will the surrogate bound $l^*/2$ converge to $D(R, p)$ as X increases?

The first question can be answered by geometrical analysis. As stated in Elzinga and Hearn (1972a), given a cluster of points P , “it is evident that the optimal [smallest enclosing] circle is defined by either two or three points [in X] that are vertices of the convex hull [of X]. If the circle whose diameter joins the points of maximum separation encloses all points, then that is the solution. Otherwise, there are some three of the points, forming an acute angle, such that the optimal circle is the circumscribed circle of that triangle.” For the latter case, the points of maximum separation must both lie on the circle. For a cluster of points with an SEC radius of R , the maximum separation distance is minimized when the inscribing triangle that contains the maximum separation points as its vertices is equilateral, having an equal side length of $\sqrt{3}R$. In our problem, let l_k denote the maximum separation distance of cluster k , that is, $l_k := \max \{[(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2} : i, j \in I \text{ and } \alpha_{ik}^* = \alpha_{jk}^* = 1\}$. We will have $l_k \geq \sqrt{3}r_k^*$, for $k = 1, \dots, p$. Subsequently,

$$l^* = \max\{l_1, \dots, l_p\} \geq \sqrt{3} \max\{r_1^*, \dots, r_p^*\} = \sqrt{3}\bar{r} \geq \sqrt{3}r^*$$

This answers the first question: even in the worst case, the substitute bound $l^*/2$ will be no less than $\sqrt{3}/2 \approx 0.866$ of the bound r^* obtained by solving the EPC problem.

The answer to the second question depends on the shape of the region R and the number of centers p . For a special case, when R is a regular triangle and $p = 1$, the largest value $l^*/2$ can take is $\sqrt{3}D(R,p)/2$. Clearly, in this case the substitute bound does not converge to the global optimum as X approaches R , and this is probably the worst case one can get in terms of the relative gap between the bound and the global optimum. In cases where R has more sides or takes a more irregular (e.g., asymmetric) shape, and where p takes a nontrivial value (e.g., greater 2), the gap between the best possible substitute bound and the global optimum should be much smaller (observed via numerical experiments), though there is not a theory to quantify that gap.

We now summarize the clustering-based algorithm for the p -center problem. Suppose that the polygonal region R is described by a set F of facets, and each facet $f \in F$ is described by its normal direction (a_f, b_f) and the directional distance to the origin c_f , so that a point (z, t) in R must satisfy $a_f z + b_f t \leq c_f$. Suppose we are given a set $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$ of initial pivot points with $n \geq p + 1$, and an upper bound of $D(R, p)$, denoted by $\bar{d}$, then Algorithm 2 can approximately solve the problem and return the upper and lower bounds.

Remarks on Algorithm 2: (1) Several numeric constants have been used in the algorithm description. These are the values adopted in our implementation for polygonal regions scaled to fit in a 100x100 canvas area. It should be understood that these parameters can be set to other values as needed. (2) The lower bounding algorithm proposed here is a heuristic - while it is able to improve the lower bounds as more pivot points are included, we do not have proof for whether and where the improvement process converges. The effectiveness of the algorithm has to be assessed via computational experiments. (3) The initialization step leaves open two questions: how to select the initial pivot points X and how to find a good upper bound $\bar{d}$. We will answer these questions now.

3.3 Upper bounding methods

We employ the Voronoi heuristics developed by Suzuki and Drezner (1996) to find feasible solutions, and pick the best one as the initial upper bound for $D(R, p)$. A Voronoi diagram is a subdivision of the space into the nearest neighborhoods of a given set of points (Shamos and Hoey 1975; Aurenhammer 1991). The idea of the Voronoi heuristic is as follows: (1) randomly generate p centers in the polygon R ; (2) use these centers to construct a Voronoi diagram, which will divide the polygon into p polygonal cells; (3) for each cell, solve a 1-center problem to find the center of a minimum-radius circle to cover all

Algorithm 2 Clustering-based method for lower bounding the p -center problem

```

1: Initialize  $X \leftarrow$  set of pivot points,  $\bar{d} \leftarrow$  an upper bound
2: repeat
3:   Solve Cluster( $X$ ) to obtain  $l^*$ 
4:   for  $k \in \{1, \dots, p\}$  do
5:     Solve SEC( $k$ ) to obtain the center  $(t_k^*, z_k^*)$  and radius  $r_k^*$ 
6:   end for
7:   Update centers  $S \leftarrow \{(t_1^*, z_1^*), \dots, (t_p^*, z_p^*)\}$ 
8:   Solve MD( $S$ ) to obtain  $(x^*, y^*, d^*)$ 
9:   Update upper bound  $\bar{d} \leftarrow \min\{d^*, \bar{d}\}$ 
10:  if  $(d^* - \max_{k=1}^p r_k^*) \geq 0.0001$  then
11:     $X \leftarrow X \cup \{(x^*, y^*)\}$ 
12:  else
13:     $e \leftarrow 1$ 
14:    repeat
15:      Status  $\leftarrow$  Solve NEWP( $e$ )
16:       $e \leftarrow 0.6e$  if Status is infeasible
17:    until Status is feasible or  $e < 0.0001$ 
18:    if Status is feasible then
19:       $X \leftarrow X \cup \{(x', y')\}$ 
20:    else
21:      Terminate
22:    end if
23:  end if
24: until  $\bar{d} - l^*/2 < 0.0001$  or time limit is reached

```

corner points of the cell, and relocate the cell center by this solution; (4) repeat steps (1) - (3) until the changes in center locations across successive iterations are small enough. After the process terminates, a finish-up step is performed to wiggle the centers so that the radii of the 1-center circles are equal. We call the Voronoi heuristic and the finish-up algorithm bundle the Voronoi heuristics.

The covering circles obtained by different runs of the Voronoi heuristics may be located differently, depending on the random starting points generated in step (1). Different sets of circles may have the same or different radii (objective values). Therefore, in practice we can run the heuristics many times and pick the best solution for use as the upper bound. Figure 2 demonstrates two runs of the Voronoi heuristics for the same problem instance - covering a square area with 6 circles. The two runs converged to different solutions with different radii. In general, we observed that regardless of the starting center locations, the Voronoi heuristic, i.e., steps (1) to (4), usually converges quickly as shown in the figure. The finish-up step, which solves a convex optimization problem, is also a quick one. So the Voronoi heuristics are computationally efficient. Repeated experiments suggest

that there are not many unique solutions that the random-starting Voronoi heuristics is capable of identifying. For example, 100 runs of the Voronoi heuristics on the same problem as shown in Fig. 2 produced only six unique (up to the 7th digit after the decimal point) objective values. They are {29.873, 29.895, 30.712, 31.756, 31.870, 31.965}. We conjecture that the number of unique solutions that can be returned by the Voronoi heuristics is finite despite the infinitely many random starting locations. We also conjecture that the best solution in this finite set is actually the global optimal solution, i.e., $D(R, p)$.

3.4 Strategies for selecting the initial pivot set

The initial pivot set X must contain at least $p + 1$ points to make both the $EPC(X)$ and the $Cluster(X)$ nontrivial in the first iteration. Different strategies may be considered for generating the initial X . A “lean approach” would start with $p + 1$ pivots. In this case, we would be facing an initial EPC problem of using p circles to cover $p + 1$ points, which means that at least one circle must cover two points. Since the EPC objective is to minimize the circle radius and we

want the optimal objective value to be as large as possible (to serve as a good lower bound for $D(R, p)$), the ideal distribution of the $p + 1$ pivots should be such that the minimum of the pairwise distances among these points is maximized. Finding p points in a compact set in $\mathbb{R}^2$ that satisfy this property is called the p -dispersion problem (Shier 1977; Drezner and Erkut 1995). The n -dispersion problem in R can be formulated as the following nonlinear program

$$\begin{aligned} \text{Maximize } \delta, \text{ subject to } & \delta \leq \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \\ & 1 \leq i < j \leq n \text{ and } (x_i, y_i) \in R, \\ & 1 \leq i \leq n \end{aligned} \quad (18)$$

Solving this problem with n equal to $p + 1$ will create the best initial set of $p + 1$ pivot points in the context of both Algorithm 1 and 2. Moreover, as analyzed above the objective value δ of any feasible solution forms a lower bound to $2D(R, p)$. Note that the problem can also be solved as a quadratically constrained problem (QCP), via squaring both sides of the main inequality constraint.

Fig. 3 Initial pivot point strategies for the 4-center problem covering a polygonal area. Left: Solution to the 5-dispersion problem. Middle: Isometric grid. Right: Corner points of the polygon and intersection points extracted from the Voronoi heuristics

An “expansive approach” would attempt to represent the polygon at a higher fidelity by sampling more points in its interior. To this end, one can use an isometric grid, i.e., a grid of points such that the distances between each point and its nearest neighbor(s) are the same for all points. Specifically, overlay an isometric grid on the polygon to be covered and include all grid points that fall inside the polygon in X , see Fig. 3 for an illustration. The isometric grid is also said to form a hexagon tessellation, similar to the one shown in Drezner et al. (2019). The quality of the lower bound will be largely determined by the resolution of the grid - the smaller the inter-pivot distance, the better the lower bound will be. The drawback is that not all interior points will turn out to have the same importance for the EPC objective, so it would be a waste to include a large set of points upfront.

A third approach is to heuristically include a small set of points guessed to be highly “pivotal” in keeping the optimal objective value of $EPC(X)$ as large as possible. To this end, we propose the following strategy. First, find a good feasible solution (i.e., using the Voronoi heuristics) which will provide a set of p covering circles, then find the intersection points of these circles, as well as the intersection points between the circles and the polygon boundaries. The initial set of pivot points will include these intersection points, and all the corner points (vertices) of the polygonal area R being covered. The three approaches are illustrated in Fig. 3. In reality, if a nontrivial solution to the $(p+1)$ -dispersion problem is available, the solution points can be included in X as well, e.g., set X to be the union of the five points in the left subfigure and the 14 points in the right subfigure of Fig. 3.

4 Computational experiments

In this section, we evaluate the computational performances of different candidate approaches, and apply the EPC-based and clustering-based approaches to several stylized cases and report the best bound found for them. All experiments have been performed on a Dell Precision Tower 8520 with an Intel(R) Core(TM) i9-9900X CPU @ 3.50 GHz, 16.0 GB RAM and Windows 10 Enterprise

Operating System. We used GAMS 30.2.0 and associated solvers for optimization modeling and solution. All solvers are limited to using a single thread (by GAMS option “threads=1”) and all optimization models were solved to global optimality (by using global solvers and setting GAMS option “optcr=0, optca=0”). The Voronoi heuristics method is implemented in Python by utilizing the `scipy.spatial.Voronoi` package and the GAMS Python API⁴.

4.1 Model and solver selection

Repetitively solving the Euclidean p -center (EPC) problem or the Cluster problem to global optimality constitutes the central piece of the iterative lower-bounding algorithm. As we rely on off-the-shelf solvers, it is critical to choose a model, formulation and solver combination that scales well as the number of pivot points increases. Using the 5-center problem for a square area, we test the performances of $EPC(X)$ and $Cluster(X)$ at different cardinalities of X , from 10 to 50 with an increment of 5. For the EPC problem, we run both the MINLP formulation and the MIQCP formulation. We have tested all available global solvers for these model types in GAMS, and only report here the ones that have been capable of solving most cases in 1000 seconds. Specifically, the MINLP formulation was solved by SCIP, and the MIQCP formulation was solved by GLoMIQO, GUROBI (Gurobi Optimization 2020) and SCIP. The Cluster model, was solved by GLoMIQO, GUROBI and SCIP.

The solution times are plotted in Fig. 4. It is evident that the Cluster model is overall easier to solve than the EPC formulation - GUROBI and SCIP consistently scored the lowest computing times in all cases. For the EPC problem, the MIQCP formulation was quite amenable to the SCIP solver but scaled poorly for the GLoMIQO and GUROBI. The MINLP formulation, for which SCIP is the only viable solver, does not scale well and becomes insolvable when the number of pivots exceeded 35. Given these observations, we conclude that the Cluster model and the

⁴The computer program will be published as supplemental material and in the author’s Github repository upon publication of the manuscript.

Fig. 4 Scalability comparison of different formulations and solvers for the EPC problem

MIQCP formulation of the EPC model are most scalable and should be used in the iterative framework for computing the lower bound. We will use SCIP to solve the Cluster model and use GUROBI to solve the EPC model in subsequent experiments.

4.2 Solving and utilizing the dispersion problem

The solution to the $(p + 1)$ -dispersion problem provides both a benchmark lower bound of $D(R, p)$ and a set of effective pivot points to kick off Algorithms 1 and 2. However, obtaining the global optimal solution to the problem is computationally intensive even for small p values. Computational experiments on small cases indicated that a global solver such as SCIP and GLoMIQO was able to find the true global solution very quickly, whereas improving the lower bound and ultimately proving global optimality takes the majority of the solver time. For instance, for the 6-dispersion problem in a square area, it took SCIP less than 1 second to identify the global solution but 900 seconds to prove its global optimality, that is, the incumbent solution has never changed during the entire global optimization process. This phenomenon is observed in other instances (i.e., different region R and different p values) as well. Therefore, we can harvest a good (guessed global) dispersion solution by running problem (18) for a brief period using SCIP and take the incumbent solution thereof.

There is an ad-hoc way to improve the solution time of the problem (18) that is worth mentioning here. We can make reasonable judgments about the location of the points being maximally dispersed, and add these conjectures as additional constraints. Taking the 6-dispersion problem in a square area for example, it is apparent

that some point must lie on one of the sides of the square (including the junction point of two sides). Due to symmetry, we can express this insight by dictating that “point 1 must lie on side 1”, which corresponds to the constraint $a_1x_1 + b_1y_1 = c_1$. Adding this constraint reduced the solution time from 908 s to 263 s. Furthermore, we can reasonably guess that some point should lie at one of the corner points of the square. Suppose side 2 intersects side 1, then we can add the constraint $a_2x_1 + b_2y_1 = c_2$, which, together with the previously added one, forces point 1 to be at the corner point. Introducing this constraint could further reduce the solution time to 52 s. More insights like this can be added to further reduce the computing time. The 6-dispersion solution is visualized in Fig. 5. This method generalizes well for symmetrically shaped areas since different sides and corner points are interchangeable. For irregular shapes, insights should be case by case and there may be a risk of over-constraining which causes loss of optimality.

The same constraint generation technique can be applied to improving the computational efficiency for solving $EPC(X)$ and $Cluster(X)$. Consider the 5-center problem in a square area for which the 6 points of the 6-dispersion problem (shown in Fig. 5) are used as pivot points in X . Given that the points 1, 2 and 6 are far apart from each other, they must be covered by three different circles in the solution of $EPC(X)$, and likewise they must belong to three different clusters in the solution of $Cluster(X)$.

To be precise, if we can identify a maximal subset of the set of $(p + 1)$ points from the dispersion solution such that the distance between each pair of points in this subset is greater than twice the upper bound (UB) of $D(R, p)$, then

we can constrain, without losing optimality, the $EPC(X)$ and $Cluster(X)$ models by fixing the assignment of these points to different circles and clusters, respectively. Identifying such a subset is a combinatorial problem, formulated as follows.

$$\begin{aligned}
 &\text{Maximize} && \sum_{i=1}^{p+1} w_i \\
 \text{s.t} &&& (2\bar{d} + \epsilon)w_i w_j \leq d_{ij}, \quad \forall i, j \in \{1, \dots, p+1\}, i < j \\
 &&& w_i \in \{0, 1\}, \quad \forall i \in \{1, \dots, p+1\}
 \end{aligned}$$

where $\bar{d}$ is the UB of $D(R, p)$, d_{ij} is the distance between point i and point j and ϵ is a small positive number. Solving this problem for the points shown in Fig. 5 we obtain the optimal solution $w_i^* = 1$ for $i \in \{1, 3, 4, 6\}$, which is actually better than the eyeballed solution $\{1, 2, 6\}$ in the last paragraph. The optimal solution can be translated to the following variable-fixing constraints for the EPC problem (the MIQCP formulation), $s_{11} = 1, s_{32} = 1, s_{43} = 1, s_{64} = 1$, and to the following constraints for the Cluster problem, $\alpha_{11} = 1, \alpha_{32} = 1, \alpha_{43} = 1, \alpha_{64} = 1$. Note here that the four different circles (clusters) assigned to the four points are arbitrarily indexed by $\{1, 2, 3, 4\}$.

The inclusion of these auxiliary constraints could help reduce the solution time by a substantial amount. We demonstrated this through the following experiment. For the 5-center problem in a square area, we ran Algorithms 1 (EPC-based) and 2 (Cluster-based) in two settings. In one setting (NoFix), the $(p+1)$ -dispersion solution points were not included in the initial X (thus $|X| = 14$) and no assignment fixing was applied; in the other setting, the $(p+1)$ -dispersion solution points were included in the initial X (thus $|X| = 14 + (6+1) = 21$) and applicable point-circle (cluster) assignment was fixed using the above method. Figure 6 shows the progress each run was able to make within one-hour run time. For instance, EPC NoFix started with 14 pivots and an initial gap of around 0.4, and ended having 34 pivots with an ending gap of 0.092 (or 9.2%). In comparison, EPC started with 21 pivots and an initial gap of 0.103 (determined by the 7-dispersion solution), and ended having 116 pivots and a gap of 0.024. Clearly, the EPC (with assignment fixes) solved faster and made better progress in the set time frame. For the Cluster

problem, fixing the assignment did not help much, but it did enable the algorithm to start with a smaller starting gap provided by the $(p + 1)$ -dispersion problem. Considering its overall positive effect on computational efficiency, we will employ the assignment fixes in subsequent experiments.

4.3 Case study: p -center solutions for regular polygons

We apply the Voronoi heuristics and Algorithm 1 and 2 to various stylized cases consisting of regular polygons and a range of p values. The purpose is to report the performances of the proposed algorithms. The test cases are easily reproducible and hence the experiment results can serve as a benchmark for future improvements.

We consider regular polygons with 3, 4, 5 and 6 equal-length sides in a 100×100 canvas area. The vertex coordinates are listed below.

```
triangle = np.array([[0,0], [100,0],
                    [50,86.6025]])
square = np.array([[0,0], [0,100],
                  [100,100], [100,0]])
pentagon = np.array([[19.0983,0],
                    [80.9017,0], [100,58.7785],
                    [50,95.1057], [0,58.7785]])
hexagon = np.array([[25,0], [75,0],
                   [100,43.3013], [75,86.6025],
                   [25,86.6025], [0,43.3013]])
```

We solve the p -center problem for each polygon for $p \in \{3, \dots, 10\}$. So a total of 32 cases are computed. In each

Table 1 Performance achieved by different methods in 2-hour run time for 32 stylized test cases

Region	p	UB	Gap			Med CPU (s)		Max CPU (s)		Pivots	
			Disp	Clus	EPC	Clus	EPC	Clus	EPC	Clus	EPC
triangle	3	28.868	0.00%	0.00%	0.00%	0.1	1.2	0.1	1.2	11	11
	4	26.795	6.70%	6.70%	0.01%	2.4	9.5	49.0	28.8	75	142
	5	25.000	0.00%	0.00%	0.00%	0.3	2.2	0.3	2.2	17	17
	6	19.245	4.90%	4.90%	0.35%	10.0	33.7	502.8	115.6	118	202
	7	18.525	7.40%	2.76%	2.10%	13.9	46.4	772.8	278.1	91	129
	8	17.699	5.83%	5.83%	5.83%	27.4	64.4	1273.7	1377.0	72	74
	9	16.667	0.00%	0.00%	0.00%	1.9	59.4	1.9	59.4	29	29
	10	14.434	4.65%	4.65%	4.65%	30.8	86.0	435.1	276.2	100	99
square	3	50.389	0.77%	0.01%	0.01%	0.3	4.7	1.6	14.8	93	160
	4	35.355	0.00%	0.00%	0.00%	0.2	1.6	0.2	1.6	14	14
	5	32.616	7.88%	1.93%	0.75%	18.4	34.6	300.0	169.8	136	187
	6	29.873	10.30%	3.77%	1.97%	16.7	33.9	810.7	268.6	85	136
	7	27.429	5.64%	2.18%	1.85%	10.8	46.9	1412.7	438.1	105	94
	8	26.099	4.21%	4.21%	4.21%	11.1	81.7	748.7	394.7	87	101
	9	23.064	8.67%	7.11%	7.90%	18.7	135.7	231.3	502.6	88	72
	10	21.823	8.83%	8.83%	8.83%	22.5	126.8	739.6	887.2	86	72
pentagon	3	38.790	14.40%	0.33%	0.09%	5.0	22.7	317.0	72.4	180	291
	4	32.771	5.70%	1.50%	0.65%	17.6	21.7	377.4	157.1	118	234
	5	26.287	0.00%	0.00%	0.00%	0.2	2.9	0.2	2.9	20	20
	6	24.345	8.78%	2.82%	1.96%	29.1	43.3	603.9	364.4	92	106
	7	22.402	11.34%	5.87%	5.98%	23.8	69.7	676.2	407.2	83	89
	8	20.183	9.19%	7.03%	5.75%	25.3	48.7	828.1	377.5	99	104
	9	19.377	12.72%	6.35%	10.34%	30.1	175.6	774.8	945.0	84	53
	10	18.338	11.01%	10.24%	11.01%	34.9	150.1	2126.4	733.8	82	56
hexagon	3	37.500	11.81%	0.05%	0.05%	5.8	24.3	157.1	144.8	269	271
	4	31.699	14.49%	0.83%	0.49%	17.3	20.4	447.2	198.8	131	214
	5	27.848	10.23%	6.35%	2.67%	14.1	27.5	1173.6	242.6	89	167
	6	25.000	0.00%	0.00%	0.00%	0.4	8.3	0.4	8.3	24	24
	7	21.651	9.60%	2.15%	2.88%	33.2	54.4	820.7	213.2	112	126
	8	19.973	12.38%	10.98%	6.39%	26.4	49.4	622.0	270.4	89	127
	9	18.852	10.54%	10.12%	10.54%	37.0	132.5	1408.4	1123.7	72	58
	10	18.105	12.46%	12.46%	12.46%	36.4	606.8	945.5	2126.0	78	44

Fig. 7 Solution progress made in a 2-hr time frame on different test cases

case, the initial upper bound is obtained by applying the Voronoi heuristics 20 times with random starting points and picking the best one. The initial set of pivot points X is the combination of the solution points for the $(p+1)$ -dispersion problem and the points identified by the Voronoi heuristic solution (see Section 3.4). The variable-fixing strategy discussed in Section 4.2 is also applied to the EPC(X) and Cluster(X) problems in Algorithms 1 and 2, respectively. A global timer was used and the total time allowed for each run was limited to 2 hours. The solutions are reported in Table 1 and Fig. 7. The best solutions found for all the cases are illustrated in Figs. 8 and 9.

Table 1 lists the upper bound (UB) of $D(R, p)$, the percent Gap (i.e., $(UB-LB)/UB \times 100\%$) achieved by the lower bound (LB) found by the $(p+1)$ -dispersion problem, Algorithm 2 (Cluster-based) and Algorithm 1 (EPC-based). Also listed in the table include the median and maximum solution time (Med and Max CPU) for the Cluster and EPC problems, and the number of pivots in set X upon the termination of the solution process. For EPC, the solution process terminates either when the gap is reduced

to below 0.001% or when the 2-hr time limit is reached. For Cluster, a third termination scenario is possible, that is, when the algorithm fails to identify a new pivot point based on the current solution. Only one case, i.e., a triangle with $p = 4$, terminated in this way, indicating that for this particular case there is an intrinsic gap between the minimum cluster radius (even when X includes all points in R) and $D(R, p)$. On the other hand, there are some “easy” cases where both the Cluster and the EPC based algorithms were able to find the best LB (which is equal to the UB) in one iteration using the initial set of pivot points. These cases include triangle with $p = 3, 5, 9$, square with $p = 4$, pentagon with $p = 5$ and hexagon with $p = 6$. The solution process for all the remaining cases was terminated due to the time limit.

In terms of computing time, we can see that in all cases the median time to solve a Cluster problem is shorter than the median time to solve an EPC problem over the 2-hr solution process, indicating better scalability of the Cluster-based algorithm. For both algorithms the median solution time generally increases as p increases, while this

Fig. 8 Best solutions for regular triangular and square areas covered by 3 to 10 centers. The solution may not be unique due to the symmetry of the area shape

Fig. 9 Best solutions for the regular pentagon and regular hexagon areas covered by 3 to 10 centers. The solution may not be unique due to the symmetry of the area shape

trend is more pronounced for the EPC-based algorithm. The maximum solution time is too volatile to reveal much general insight. One might wonder why for many cases the Cluster-based algorithm terminated with fewer pivot points than the EPC-based algorithm, given that the Cluster(X) problem solves faster. See the square case with $p = 3$ for example. This phenomenon is due to frequent invocations of solving the NEWP(e) model in Algorithm 2. This extra step only occurs sporadically for certain cases, but more frequently for other cases (e.g., simple shapes with small p values).

The algorithms' progression is visualized in Fig. 7. Each subplot represents a row in Table 1 and plots the trace of the gap reduction as more and more pivots are added to X . In each subplot, the horizontal axis represents the iteration number, the vertical axis represents the optimality gap, and the number in the title line represents the number of centers. Note that, although X grows over time, the horizontal axis does not exactly represent time, since new pivots are identified at random discrete instants in time and it is difficult to align the progression of the two

algorithms by time. Instead, the last point plotted for each algorithm is the point at which the algorithm terminates (mostly due to reaching the 2-hr time limit). As expected, for both algorithms the gap reduction is not a smooth process - not every new pivot point can immediately trigger a gap reduction. However, we cannot say that those pivot points that did not lead to an immediate gap reduction were wasted, as their presence played a role in future iterations.

For several cases, e.g., square with $p = 8$ and 10, both algorithms failed to improve the lower bound within the 2-hr time framework. Upon examining the solution process we have not identified any locking or cycling issue. When more time was allowed (e.g., set the time limit to 5 hours), gap reduction was observed for these cases. Therefore, we conclude that the observation was due to the intrinsic difficulty of these cases. In other words, the pivot points identified in a two-hour period were not enough to improve the lower bound upon what was identified by the $(p + 1)$ -dispersion solution.

4.4 Discussion

As noted in Section 1, we have assumed that the target region R is a convex polygon. In real-world applications, however, the service area to be covered by p hubs may be of an irregular shape, possibly having holes (infeasible locations for a center, such as lakes). In such cases, the Voronoi heuristics could not guarantee a feasible solution, or a valid upper bound. Instead, the algorithm proposed by Wei et al. (2006) and subsequently improved by Liu (2021a) should be used for upper bounding. The lower bounding procedure proposed in this paper could be then applied to the convex hull of the area. The lower bound thus obtained would remain valid, but would not be as tight. Several enhancements could be explored, for example, discard the pivot points that lie outside the actual service area, or introduce additional constraints to the EPC problem to prevent any center from dropping into a hole, etc. Such exploration is beyond the scope of this paper and we leave it for future work.

5 Conclusion

Covering an area with p minimum-radius circles is an intriguing mathematical problem that has a broad and growing base of applications. Contemporary applications include the location of base stations for 5G cellular networks, and the location of battery depots for delivery drones. Even though the Voronoi heuristics method is quite effective at finding good feasible solutions, there has been a lack of investigation for lower bounding methods, without which the solution quality cannot be quantified. In this paper, we developed an iterative framework for solving the p -center problem with a focus on computing and improving its lower bound. Several formulations and algorithmic enhancements were proposed, and computational results on several benchmark problems were reported.

From the mathematical programming point of view, the p -center problem in an area is such an extremely difficult problem to the extent that its reduced version, i.e., the Euclidean p -center (EPC) problem covering a discrete set of points, is an NP-hard problem. To enhance the computational efficiency of the algorithmic framework, we evaluated different formulations and solvers for globally solving the EPC problem, developed a clustering subproblem, and proposed innovative ways of utilizing the Voronoi heuristics and the p -dispersion problem to improve the computational efficiency of the difficult EPC and clustering subproblems.

Except for a few special cases, the lower bound computation is a continuous improvement process that gets harder and harder as the size of the subproblem increases. This is a major shortcoming of the proposed framework. More ingenious and scalable methods need to be invented to achieve better performances. Another possible route

for globally solving the p -center problem is finding a way to systematically enumerate all solutions by the Voronoi heuristics and prove the conjecture that the global solution is one of the Voronoi heuristics solutions. That would be a remarkable breakthrough in computational geometry and would provide insights in solving NP-hard problems as well.

Acknowledgements

The author would like to thank the two anonymous referees whose constructive comments helped improve the quality of the paper. The work is supported by the U.S. National Science Foundation under the award 1944068.

Author's contribution

The author claims all contribution in the present work. The author read and approved the final manuscript.

Funding

The research is supported by the National Science Foundation under grant number 1944068.

Availability of data and materials

The data used in the experiments will be made available under the author's GitHub repository.

Declarations

Competing interests

The author declares that he has no competing interests.

Received: 12 November 2021 Accepted: 13 December 2021

Published online: 24 January 2022

References

- Achterberg, T. (2009). SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1), 1–41. <https://doi.org/10.1007/s12532-008-0001-1>.
- Aikens, C.H. (1985). Facility location models for distribution planning. *European Journal of Operational Research*, 22(3), 263–279. [https://doi.org/10.1016/0377-2217\(85\)90246-2](https://doi.org/10.1016/0377-2217(85)90246-2).
- Aronov, B., Carmi, P., Katz, M.J. (2009). Minimum-cost load-balancing partitions. *Algorithmica*, 54(3), 318–336.
- Aurenhammer, F. (1991). Voronoi diagrams - a survey of a fundamental data structure. *ACM Computing Surveys*, 23(3), 345–405.
- Baron, O., Berman, O., Krass, D., Wang, Q. (2007). The equitable location problem on the plane. *European Journal of Operational Research*, 183(2), 578–590.
- Berman, O., Drezner, Z., Tamir, A., Wesolowsky, G.O. (2009). Optimal location with equitable loads. *Annals of Operations Research*, 167(1), 307–325.
- Blanco, V., Puerto, J., Ben-Ali, S.E.-H. (2014). Revisiting several problems and algorithms in continuous location with L_r norms. *Computational Optimization and Applications*, 58(3), 563–595.
- Blanco, V., Puerto, J., Ben-Ali, S.E.-H. (2016). Continuous multifacility ordered median location problems. *European Journal of Operational Research*, 250(1), 56–64.
- Brimberg, J., Hansen, P., Mladenović, N., Taillard, E.D. (2000). Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem. *Operations Research*, 48(3), 444–460.
- Brimberg, J., & Salhi, S. (2005). A continuous location-allocation problem with zone-dependent fixed cost. *Annals of Operations Research*, 136(1), 99–115.
- Brown, G., Carlyle, M., Diehl, D., Kline, J., Wood, K. (2005). A two-sided optimization for theater ballistic missile defense. *Operations Research*, 53(5), 745–763.
- Callaghan, B., Salhi, S., Nagy, G. (2017). Speeding up the optimal method of dreznar for the p-centre problem in the plane. *European Journal of Operational Research*, 257(3), 722–734.
- Carlsson, J.G., Behrooz, M., Li, X. (2016). Geometric partitioning and robust ad-hoc network design. *Annals of Operations Research*, 238(1), 41–68.

- Chandrasekaran, R., & Daughety, A. (1981). Location on tree networks: P-centre and n-dispersion problems. *Mathematics of Operations Research*, 6(1), 50–57.
- Chen, P.-C., Hansen, P., Jaumard, B., Tuy, H. (1992). Weber's problem with attraction and repulsion*. *Journal of Regional Science*, 32(4), 467–486. <https://doi.org/10.1111/j.1467-9787.1992.tb00200.x>.
- CONOPT Solver Manual in GAMS Documentation (2020). https://www.gams.com/latest/docs/S_CONOPT.html. Accessed 1 Aug 2021.
- Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6(1), 37–53. <https://doi.org/10.1137/1006005>.
- Cooper, L., & Katz, I.N. (1981). The weber problem revisited. *Computers & Mathematics with Applications*, 7(3), 225–234. [https://doi.org/10.1016/0898-1221\(81\)90082-1](https://doi.org/10.1016/0898-1221(81)90082-1).
- Daskin, M.S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, Inc.
- de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008). *Computational Geometry Algorithms and Applications*, 3rd edn. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.
- Dinler, D., Tural, M.K., Iyigun, C. (2015). Heuristics for a continuous multi-facility location problem with demand regions. *Computers and Operations Research*, 62, 237–256.
- Drezner, Z. (1984). The p-centre problem-heuristic and optimal algorithms. *The Journal of the Operational Research Society*, 35(8), 741–748.
- Drezner, T., Drezner, Z., Suzuki, A. (2019). A cover based competitive facility location model with continuous demand. *Naval Research Logistics (NRL)*, 66(7), 565–581.
- Drezner, Z., & Erkut, E. (1995). Solving the continuous p-dispersion problem using non-linear programming. *The Journal of the Operational Research Society*, 46(4), 516–520.
- Drud, A. (1985). CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, 31(2), 153–191. <https://doi.org/10.1007/BF02591747>.
- Du, H., & Xu, Y. (2014). An approximation algorithm for k-center problem on a convex polygon. *Journal of Combinatorial Optimization*, 27, 504–518.
- Dyer, M.E. (1986). On a multidimensional search technique and its application to the euclidean one-centre problem. *SIAM Journal on Computing*, 15(3), 725–738. <https://doi.org/10.1137/0215052>.
- Elzinga, J., & Hearn, D. (1972). Geometrical solutions for some minimax location problems. *Transportation Science*, 6, 379–394. <https://doi.org/10.1287/trsc.6.4.379>.
- Elzinga, D.J., & Hearn, D.W. (1972). The minimum covering sphere problem. *Management Science*, 19(1), 96–104.
- Farahani, R.Z., Asgari, N., Heidari, N., Hosseini, M., Goh, M. (2012). Covering problems in facility location: A review. *Computers & Industrial Engineering*, 62, 368–407.
- Gamal, M.D.H., & Salhi, S. (2003). A cellular heuristic for the multisource weber problem. *Computers and Operations Research*, 30(11), 1609–1624.
- Ganame, H., Yingzhuang, L., Hamrouni, A., Ghazzai, H., Chen, H. (2021). Evolutionary algorithms for 5g multi-tier radio access network planning. *IEEE Access*, 9, 30386–30403. <https://doi.org/10.1109/ACCESS.2021.3058619>.
- Gonzalez-Brevis, P., Gondzio, J., Fan, Y., Poor, H.V., Thompson, J., Krikidis, I., Chung, P. (2011). Base station location optimization for minimal energy consumption in wireless networks. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)* (pp. 1–5). Budapest, Hungary.
- Gurobi Optimization, LLC (2020). Gurobi Optimizer Reference Manual. <http://www.gurobi.com>.
- Hansen, P., Mladenović, N., Moreno Pérez, J.A. (2008). Variable neighbourhood search: methods and applications. *4OR*, 6(4), 319–360. <https://doi.org/10.1007/s10288-008-0089-1>.
- Hansen, P., Mladenović, N., Taillard, É. (1998). Heuristic solution of the multisource Weber problem as a p-median problem. *Operations Research Letters*, 22(2), 55–62.
- Hansen, P., Mladenović, N., Taillard, É. (1998). Heuristic solution of the multisource weber problem as a p-median problem. *Oper. Res. Lett.*, 22(2-3), 55–62. [https://doi.org/10.1016/S0167-6377\(98\)00004-2](https://doi.org/10.1016/S0167-6377(98)00004-2).
- Hochbaum, D.S. (1997). Approximation algorithms for np-hard problems. *SIGACT News*, 28(2), 40–52. <https://doi.org/10.1145/261342.571216>.
- Klose, A., & Drexler, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, 162(1), 4–29. Logistics: From Theory to Application.
- Kuenne, R.E., & Soland, R.M. (1972). Exact and approximate solutions to the multisource weber problem. *Mathematical Programming*, 3(1), 193–209. <https://doi.org/10.1007/BF01584989>.
- Liu, Y. (2019). An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research*, 111, 1–20.
- Liu, Y. (2021). A faster algorithm for the constrained minimum covering circle problem to expedite solving p-center problems in an irregularly shaped area with holes. *Naval Research Logistics*. <https://doi.org/10.1002/nav.22023>.
- Liu, Y. (2021). A multi-agent semi-cooperative unmanned air traffic management model with separation assurance. *EURO Journal on Transportation and Logistics*, 10, 100058. <https://doi.org/10.1016/j.ejtl.2021.100058>.
- Maranas, C.D., & Floudas, C.A. (1994). A global optimization method for weber's problem with attraction and repulsion. In W.W. Hager, D.W. Hearn, P.M. Pardalos (Eds.), *Large Scale Optimization: State of the Art*. https://doi.org/10.1007/978-1-4613-3632-7_14 (pp. 259–285). Boston: Springer.
- McGregor, P., & Shen, D. (1977). Network design: An algorithm for the access facility location problem. *IEEE Transactions on Communications*, 25(1), 61–73.
- Megiddo, N. (1983). Linear-time algorithms for linear programming in r^3 and related problems. *SIAM Journal on Computing*, 12(4), 759–776. <https://doi.org/10.1137/0212052>.
- Megiddo, N., & Supowit, K. (1984). On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1), 182–196.
- Misener, R., & Floudas, C.A. (2013). Glomiko: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1), 3–50. <https://doi.org/10.1007/s10898-012-9874-7>.
- MOSEK (2019). MOSEK Modeling Cookbook. <https://docs.mosek.com/modeling-cookbook/index.html>. Accessed 1 Aug 2021.
- Murray, A.T., O'Kelly, M.E., Church, R.L. (2008). Regional service coverage modeling. *Computers & Operations Research*, 35(2), 339–355. <https://doi.org/10.1016/j.cor.2006.03.004>. Part Special Issue: Location Modeling Dedicated to the memory of Charles S. ReVelle.
- Murray, A.T., & Wei, R. (2013). A computational approach for eliminating error in the solution of the location set covering problem. *European Journal of Operational Research*, 224(1), 52–64. <https://doi.org/10.1016/j.ejor.2012.07.027>.
- Nurmela, K.J., & Ostergård, P.R.J. (1999). More optimal packings of equal circles in a square. *Discrete & Computational Geometry*, 22, 439–457.
- Ohlemüller, M. (1997). Tabu search for large location-allocation problems. *Journal of the Operational Research Society*, 48, 745–750. <https://doi.org/10.1038/sj.jors.2600409>.
- Raeisi Dehkordi, A. (2019). The optimal solution set of the multi-source weber problem. *Bulletin of the Iranian Mathematical Society*, 45(2), 495–514.
- Revelle, C.S., & Laporte, G. (1996). The plant location problem: New models and research prospects. *Operations Research*, 44(6), 864–874.
- ReVelle, C., Toregas, C., Falkson, L. (1976). Applications of the location set-covering problem. *Geographical Analysis*, 8(1), 65–76.
- Rosing, K.E. (1992). An optimal method for solving the (generalized) multi-Weber problem. *European Journal of Operational Research*, 58(3), 414–426. [https://doi.org/10.1016/0377-2217\(92\)90072-H](https://doi.org/10.1016/0377-2217(92)90072-H).
- Sahinidis, N.V. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2), 201–205.
- Salhi, S., & Gamal, M.D.H. (2003). A genetic algorithm based approach for the uncapacitated continuous location-allocation problem. *Annals of Operations Research*, 123(1), 203–222. <https://doi.org/10.1023/A:1026131531250>.
- Shamos, M.I., & Hoey, D. (1975). Closest-point problems, In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. <https://doi.org/10.1109/SFCS.1975.8> (pp. 151–162).
- Shier, D.R. (1977). A min-max theorem for p-center problems on a tree. *Transportation Science*, 11(3), 243–252.
- Suzuki, A., & Drezner, Z. (1996). The p-center location problem in an area. *Location Science*, 4(1), 69–82.
- Tellier, L.-N. (1972). The weber problem: Solution and interpretation. *Geographical Analysis*, 4(3), 215–233. <https://doi.org/10.1111/j.1538-4632.1972.tb00472.x>.
- Wang, C.-H., Lee, C.-J., Wu, X. (2020). A coverage-based location approach and performance evaluation for the deployment of 5g base stations. *IEEE Access*, 8, 123320–123333. <https://doi.org/10.1109/ACCESS.2020.3006733>.
- Wei, R., & Murray, A.T. (2015). Continuous space maximal coverage: Insights, advances and challenges. *Computers & Operations Research*, 62, 325–336.

- Wei, H., Murray, A.T., Xiao, N. (2006). Solving the continuous space p-centre problem: planning application issues. *IMA Journal of Management Mathematics*, 17(4), 413–425.
- Wesolowsky, G.O. (1993). The Weber problem: History and perspectives. *Location Science*, 1, 5–23.
- Wolsey, L. (2020). *Integer Programming*, 2nd edn. Hoboken: John Wiley & Sons, Ltd.
- Yin, P., & Mu, L. (2015). An empirical comparison of spatial demand representations in maximal coverage modeling. *Environment and Planning B: Planning and Design*, 42(4), 574–592.
- Çalik, H., Labbé, M., Yaman, H. (2019). p-center problems. In G Laporte, S Nickel, F Saldanha da Gama (Eds.), *Location Science* (pp. 51–65). Cham: Springer.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.