Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2424 – 2433

doi:10.3906/elk-1601-186

Rapidly converging solution for p-centers in nonconvex regions

Monika MANGLA[1,*], **Deepak GARG**[1,2]
[1]Department of Computer Science and Engineering, Thapar University, Patiala, India
[2]Department of Computer Science and Engineering, Bennett University, Greater Noida, India

Received: 18.01.2016 • **Accepted/Published Online:** 20.09.2016 • **Final Version:** 29.05.2017

Abstract: This paper aims to locate p resources in a nonconvex demand plane having n demand points. The objective of the location problem is to find the location for these p resources so that the distance from each of n demand points to its nearest resource is minimized, thus simulating a p-center problem. We employ various geometrical structures for solving this location problem. The suggested approach is also capable of finding the optimal value of p so that all demand points have at least one resource at a distance Δ, where Δ is the maximum permissible distance for emergency services. Finally, an implementation of the proposed approach is presented and it is observed that the suggested approach rapidly converges towards the optimal location.

Key words: Facility location, p-center, convex polygon, geodesic distance, nonconvex region, Delaunay triangulation

1. Introduction

The p-center location problem is considered to be an important variant of location problems. The objective here is to minimize the maximum distance for every demand point to its nearest facility. At the same time, it also ensures that all n demand points in the region should be served by at least one facility. The p-center problem has had fundamental applications in a wide range of areas for a long time. It can be used for finding the best location of emergency or business facilities such as industrial factories and ambulance or fire stations. Another application of p-center is to identify locations for building servers in telecommunication systems and computer networks [1]. Thus, this area has long been a prime focus of researchers [2,3].

As a p-center is an NP-complete problem [4], it has always been a prospective area for research. Various heuristics and approximation algorithms have been proposed over time to solve the problem. Research is still ongoing for solving multiple variants of the problem like continuous or discrete location problems [5]. The location problem is continuous if the set of candidate locations for the facility is infinite. On the other hand, selecting a location among finite candidate locations is called a discrete location problem [6,7].

If $D = \{p_1, p_2, \dots p_n\}$ is a set of n demand points in the plane, the objective is to find p centers $C = \{c_1, c_2, \dots c_p\}$ such that the maximum distance for all demand points to their closest resources is minimized [8]. The formal definition of the p-center problem is: given n demand points on a network and a weight w_i associated with each demand point for $i = \{1, 2 \dots n\}$, find p locations p_i for new facilities on the plane that minimize the maximum weighted Euclidean distance between each demand point and its closest

*Correspondence: manglamona@gmail.com

facility. The following mathematical formulation was given for the *p-center* problem [9]:

$$z(c) = \min_{1\leq j\leq p} \left\{ \max_{1\leq i\leq n} \left\{ \min_{1\leq j\leq p} d\left(p_i, c_j\right) \right\} \right\}, \tag{1}$$

where $d\left(p_i,\, c_j\right)$ represents the Euclidean distance between demand point p_i and facility c_j and is defined as $d\left(p,q\right) = \sqrt{\left(p_x - q_x\right)^2 + \left(p_y - q_y\right)^2}$.

For different variants of the p-center problem, various algorithms have been proposed by researchers. For the continuous p-center problem, the best algorithm of the order of $O(n^{\sqrt{p}})$ was proposed in [10]. The *1*-center and *2*-center problems were discussed in [10,11] and run in $O\left(n^2 log^2 n\right)$ time.

In a polygonal p-center, the demand plane is represented by a simple polygon that results in some restrictions for placing resources. These restrictions in the location of resources occur due to inconsistency in Euclidean distance and realistic distance. The convex polygonal demand plane can be treated in a similar manner to the basic p-center problem and therefore does not necessitate further elaboration. On the other hand, a polygon consisting of any nonconvex vertex requires a more focused approach for the p-center, thus making the polygonal p-center problem different from the p-center problem.

Computational geometry, since its inception, has been closely related to solving the location problem in alliance with other approaches and heuristics. Over time it has helped to solve all variations of the location problem and eventually it was accepted as an efficient and effective choice for solving spatial problems. In [12,13] the $p-$center was considered for demand planes depicted by graphs and trees, such as client/server problems. This paper attempts to solve the polygonal p-center problem with the help of various computational geometric structures. Here the terms 'resource' and 'facility' have been used interchangeably throughout the paper.

The paper is divided into five sections. Section 2 discusses the p-center problem and a relevant literature survey. Section 3 focuses on the polygonal p-center problem and how various researchers have addressed it. Section 4 introduces various geometric structures and their association with location problems for the polygonal $p-$center. Section 5 discusses the proposed approach for polygonal p-center problems and illustrations of it are shown in Section 6. Section 7 concludes the paper and gives the future scope for research.

2. The *p*-center problem

The objective of the p-center problem is to minimize the coverage distance such that every demand node has at least one facility within a radius of threshold $\boldsymbol{\Delta}$ (distance). The p-center problem is also known as the minimax problem as the goal is to minimize the maximum distance between a demand node and its nearest facility, i.e. the p-radius. Figure 1 illustrates the 3-center problem, where asterisks and hexagons represent facilities and demand points, respectively. The circle represents the region/area within which all demand points are served by the corresponding facility.

Here, the objective is to minimize the maximum distance between a demand point and its closest facility represented by the value of variable z in Eq. (1). The most important aspect of the p-center problem is based on the findings of Drezner et al. [14]. As per [14], any distance beyond threshold Δ is assumed to be constant. The p-center finds its major applications in the area of emergency services. Consider a scenario of a fire brigade service; here the service provided is beneficial if and only if it is received within the stipulated time (represented by threshold Δ), failing which the service becomes ineffectual. This is demonstrated by considering the location of p circular disks of maximum radius z (threshold Δ) centered at p locations such that all n demand points are covered by at least one disk [15].

2.1. Polygonal p-center

In a polygonal *p-center*, the demand plane is assumed to be a simple polygon. When all the vertices of the polygon are convex then the polygonal p-center can be attempted in a similar manner to the basic p-center problem and hence no further elaboration is required. On the other hand, a polygon having even a single nonconvex vertex needs a more focused approach as the Euclidean distances may not be equivalent to the feasible distance whenever a nonconvex vertex appears in the path. In a polygonal p-center, the polygon may represent any region, e.g., an island or a boundary, where facilities are to be located.

The polygonal$p$-center problem can be represented by Figure 2; here the actual distance between demand point c_i and facility X is shown by a solid line, which is apparently different from the Euclidean distance shown by a dashed line. This feasible distance shown by the solid line in Figure 2 is called the geodesic distance or geodesic path. Several algorithms exist to find the geodesic path for a pair of vertices in a simple polygon consisting of nonconvex vertices [16,17].

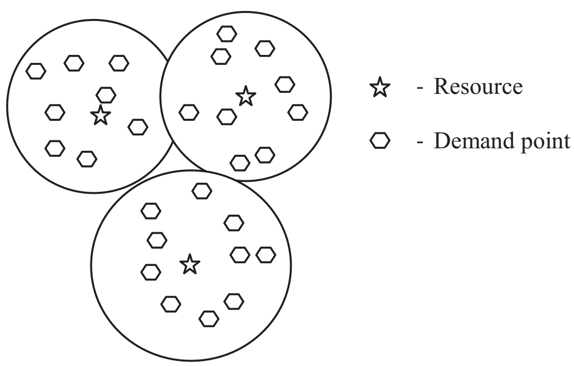

Figure 1. Illustrating the 3-center problem.

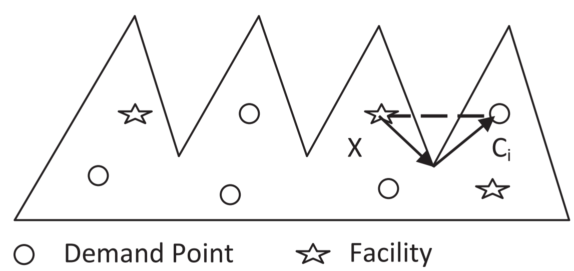

Figure 2. Illustration of polygonal p-center problem.

In this paper we use geodesic center Gc of a simple polygon. Gc is a point inside the polygon that minimizes the maximum distance to any point in the polygon. Gc is the midpoint of the geodesic diameter of a polygon, which is the length of the geodesic path of the two most distant vertices in the polygon [16]. As shown in Figure 3, the solid line represents the boundary of the simple polygon with some nonconvex vertices. The dotted line in Figure 3 represents the geodesic path between two pairs of the most distant vertices while the diamond represents the geodesic center of the polygon. According to the definition of the geodesic center of the polygon, all demand points in the polygon will lie at a maximum distance of the geodesic center of the polygon.

In [16], an algorithm was proposed to find the geodesic center in $O(nlogn)$ time. For the *1*-center problem in a polygonal plane, the facility must be located at the geodesic center of the polygon and thus the geodesic diameter is twice the p-radius.

3. Computational geometry in a p-center

Computational geometry is able to deal with spatial problems in an efficient and effective manner with the help of various geometric structures and thus has been extensively used for solving location problems since its inception. The utilization of computational geometry for solving different variations of location problems has escalated as a result of evolutions in computational geometric algorithms. Knowledge of the demand location allows using the geometric properties to identify a set of candidate facility sites in continuous space containing a subset of $p$sites that maximize coverage [17]. Although facilities are permitted to be located in continuous

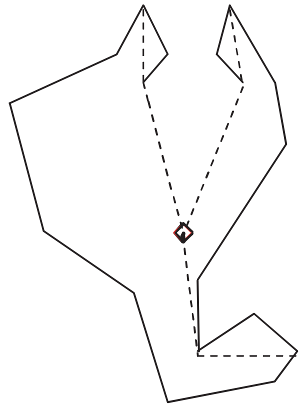

Figure 3. Geodesic diameter and geodesic center of a polygon.

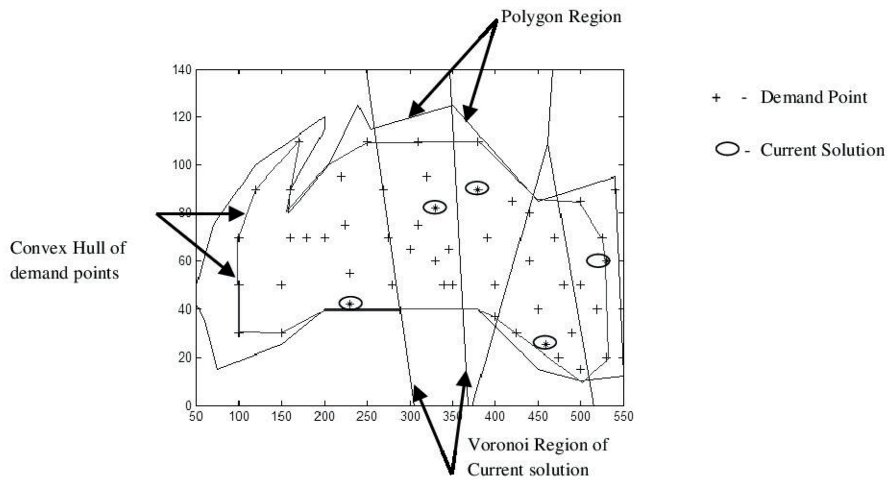

Figure 4. Illustration of demand points and Voronoi region for resources.

space, a discrete location model can also be implemented using spatial properties of demand locations. This results in geometric structures of demand points serving as the best choice for solving p-center problems. This section gives a brief overview of the Voronoi diagram, a popular computational geometry structure, followed by its dual Delaunay triangulation.

3.1. Voronoi diagram

This is one of the most popular and practical geometric structures that can be utilized for solving various location problems. The Voronoi diagram is based on set of demand points in a d-dimensional plane where these points are divided into various groups based on their distance to the nearest resource [18,19].

Definition 1 *Let $P = \{p_1, p_2 \ldots p_n\}$ be a set of n distinct points in the plane. The Voronoi diagram (VD) of P is the subdivision of the plane into n cells, one for each site. A point q lies in the Voronoi region VR_i corresponding to a site $p_i \in P$ if and only if distance $(q, p_i) <$ distance (q, p_j) for each $p_i \in P, j \neq i$ The Voronoi diagram of D, i.e. $VD(D)$, may also be defined as the union of Voronoi regions of all the points in P, i.e.*

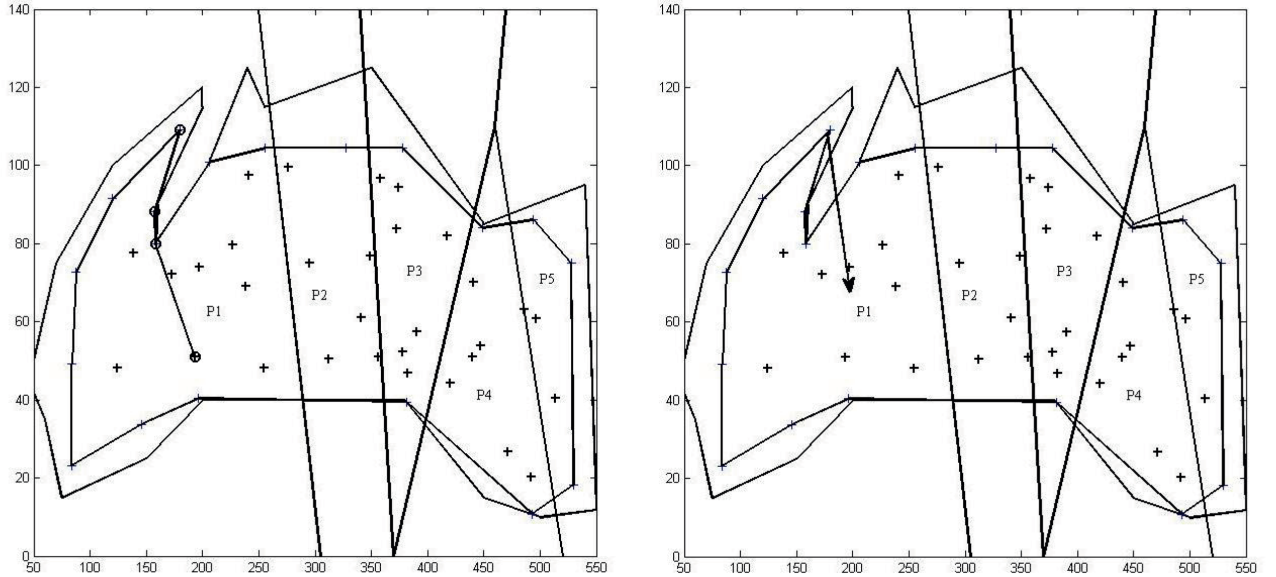

Figure 5. a) A case of the selected region having one Delaunay neighbor; b) moving towards a farther demand point along the geodesic path.

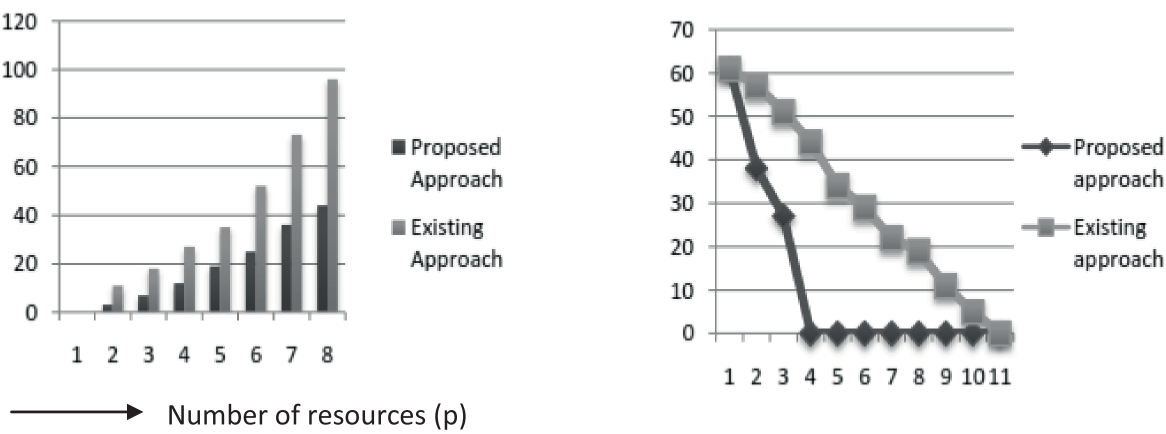

Figure 6. a) Number of iterations; b) rate of convergence.

$$VD\,(P) = \cup\{VR_i|\ \text{for all}\ \ p_i \in P\}\,.$$

Fortune's algorithm [20] can construct the VD of n points in $O(n\,log\,n)$ time complexity. The same can be used to determine the nearest and reverse nearest query point in $O(log\,n)$ time. The Voronoi diagram, being a spatial decomposition based on locations of resources, can be used for allocation of demand points to the nearest facilities. Demand points in Voronoi region VR_i can be used for the capacitated facility location as facility P_i should be capable of providing service to all demand points in VR_i. In particular, Suzuki and Okabe [21] proposed a Voronoi diagram-based heuristic to solve the continuous p-center problem. The proposed Voronoi diagram-based heuristic (VDH) for attempting the continuous space p-center problem consists of the following steps [21]:

1. Generate p centers randomly in A as an initial configuration of location sites.
2. Construct the Voronoi diagram generated by the p centers.
3. Compute the center of each Voronoi polygon (a *1*-center problem).
4. If no center has moved more than a prespecified tolerance, or the maximum number of iterations is exceeded, stop. Otherwise, go to Step 2.

Furthermore, in basic p-center problems, it is obvious that all demand points in the i^{th} Voronoi region, VR_i, will definitely use the corresponding i^{th} facility, p_i, because it is the closest facility. Since VR_i is convex, the farthest demand point from facility p_i will lie at its boundary, thus determining the radius of Voronoi region VR_i. This radius of VR_i is minimized by placement of a facility near the center of the corresponding Voronoi region. On the contrary, a polygonal p-center is a nonconvex optimization problem. Therefore, it still remains a challenge to find the global optimal solution [22,23].

Here, due to the nonconvexity of Voronoi regions, a large number of vertices may be encountered during the iterations of the VDH. The repeated execution of the *1*-center significantly increases the computation time of the VDH for nonconvex polygons [24]. This necessitates the *1*-center algorithm for nonconvex regions to efficiently execute the polygonal p-center problem and this paper proposes an approach for the same. The proposed approach uses geometric structures of the demand points present in the demand plane. The considered demand plane is a nonconvex polygon without holes. Therefore, facilities can be located anywhere in the demand plane, which is a nonconstrained version of the problem.

3.2. Delaunay triangulation

A Delaunay triangulation is a dual of a Voronoi diagram. In a Delaunay triangulation, two Voronoi sites p_i and p_j are connected by an arc if and only if $VR(p_i)$ and $VR(p_j)$ are bounded by a common Voronoi edge. The proposed approach uses Delaunay triangulation to improve initial solutions during iterations of the algorithm. Usage of Delaunay triangulation helps in significantly reducing the number of iterations.

4. Proposed approach

This section focuses on the main part of the paper. Here an efficient approach has been proposed for solving the polygonal p-center problem using spatial structures of the demand points. The proposed approach uses the Voronoi diagram and Delaunay triangulation of the demand points to optimally locate p resources in a polygonal region during iterations of the algorithm. As already discussed, the distance will definitely be the geodesic distance Gc in contrast to the Euclidean distance. Here Gc coincides with the *1*-center problem and therefore this radius is equivalent to the geodesic radius.

In the literature, it is apparent that each resource lies within the convex hull of the demand point $(CH(D))$ for all variants of the location problem [8]. Usage of the convex hull limits the possibilities for locations of resources and thus reduces the complexity. The convex hull of the demand points is the minimum enclosing polygon that encloses all demand points. The proposed polygonal p-center approach is as follows:

In the proposed approach Step 1 constructs the convex hull C of the demand points as no resource should lie outside C. Step 2 randomly selects initial locations for p resources. The Voronoi diagram and Delaunay triangulations of these p locations are constructed in Step 3 and Step 4, respectively. Step 5 finds the demand points in the Voronoi region for each resource. It helps to find the geodesic radius for each resource in Step 6.

Proposed algorithm for polygonal p-center problem
1. *Construct convex hull C of the demand points in the region* 2. *Randomly select p demand points as initial solution for the resources* $$C = \{c_1, c_2 \ldots c_p\}$$ 3. *Generate Voronoi diagram $VD(C)$ in the constrained region* 4. *Generate Delaunay triangulation $T(C)$* 5. *For each facility i*   *Assign all demand points in Voronoi region $VR(c_i)$ to set S_i* 6. *For each set S_i* *$Dist_i = \max\{r_{min_i}$ for all members in the set $S_i\}$* 7. *$DIST = \max\{Dist_i\}$ for all facilities* 8. *if $DIST >$ threshold then $Diff = DIST -$ threshold* 9. *For $p = 1$ or 2 move resource c_i having $Dist_i = DIST$ towards Gc by*   *$Diff$ provided it does not leave C* *else*   *Move resource C_i having $Dist_i = DIST$ towards longest edge in $T(C)$ by $Diff$ provided it does not leave C* 10. *If resource C_i has only one edge in $T(C)$ then move resource C_i towards farthest demand point by $Diff$ provided it does not leave C Reconstruct the Voronoi diagram VD and Delaunay triangulation T* 11. *Repeat Step 5 until termination condition holds* 12. *If termination condition does not hold for a specific number of iterations, increment the value of p and repeat Step 2*

Further steps ensure that the geodesic radius for any resource does not exceed the threshold Δ. Any resource having its geodesic radius more than the threshold Δ is chosen for relocation. Delaunay triangulation is then used to find the direction of relocation of the selected resource. The magnitude of relocation is obtained using Δ in Step 8. The algorithm iterates until termination conditions are encountered. Furthermore, if the algorithm terminates with any demand point having no resource in the circle of radius threshold Δ, the value of p is then incremented and the process is repeated for the modified value of p.

Thus, the proposed algorithm can also be used to find minimal p as it is a significant cost-influencing factor. The initial value of p can be set to any arbitrary value, even to 1, and can be further increased if required, thus finding an optimal number of resources.

In the proposed approach, Step 1 constructs the convex hull of the demand points. Well-known methods for the convex hull in $\mathbf{\Theta}(\mathbf{n}\log\mathbf{n})$ are available in the literature. It is then followed by random selection of $\mathbf{p}$ points as an initial solution, which is used for construction of the Voronoi diagram. The complexity of the Voronoi diagram for $\mathbf{p}$ points is $\mathbf{\Theta}(\mathbf{p}\log\mathbf{p})$ where $\mathbf{p} \ll \mathbf{n}$. As suggested in the approach, these initial solutions are improved using Delaunay triangulations during iterations of the approach. The Delaunay triangulation used in Step 4 of the algorithm is the dual of the Voronoi diagram and thus needs no extra computation.

As demand points in the region are static, this static nature of demand points can be used to generate the range tree. Generation of the range tree helps in finding all demand points in $\mathbf{VR_i}$ for each resource $\mathbf{c_i}$ in $\mathbf{O}(\log \mathbf{n})$ time. According to the suggested approach, shifting of any resource necessitates regeneration of the Voronoi diagram. It does not require regenerating the complete Voronoi diagram. The reason is that it only affects the Voronoi region found in Step 6 and its neighboring Voronoi regions.

Numbers of iterations here are reasonably small due to the higher rate of convergence. This higher rate of convergence is achieved using **Diff** as the magnitude of movement for the selected resource. Global selection of candidate resources for movement also helps in achieving higher rates of convergence. Thus, the proposed approach is a polynomial time method for the p-center in nonconvex polygons.

4.1. Illustration of the proposed algorithm

This section illustrates the proposed algorithm for a nonconvex region. Here demand points ($\mathbf{n}$) and resources ($\mathbf{p}$) are considered to be 50 and 5, respectively, as shown in Figure 4. The outer polygon represents the nonconvex polygon under consideration while the inner polygon represents the convex hull of the demand points. As already discussed, no resource lies outside the convex hull. Demand points are shown by the + symbol and a + symbol enclosed by an ellipse represents the initial location of resources in Figure 4. As discussed above, chosen $\mathbf{p}$ locations are used to construct the Voronoi diagram ($\mathbf{VD}$). This Voronoi diagram is represented by spatial decomposition in Figure 4. All demand points in $\mathbf{VR_i}$ will be allocated to resource $\mathbf{p_i}$ as it is their nearest facility.

Figure 5a represents the farthest demand point for $\mathbf{p_1}$ using a circle and the corresponding geodesic radius. Now, as stated in the algorithm, the resource having maximum geodesic radius is selected for movement; thus, resource $\mathbf{p_1}$ is selected during the current iteration as shown in Figure 5a. Now Delaunay triangulation $\mathbf{T}$ is used for movement of the selected resource. As per the spatial decomposition, selected resource $\mathbf{p_1}$ has only one neighboring Voronoi region and therefore $\mathbf{p_1}$ has only one edge in the corresponding $\mathbf{T}$. Hence, it is moved towards the farthest demand point in $\mathbf{VR_1}$ by a magnitude equal to the difference of the geodesic radius and threshold $\boldsymbol{\Delta}$ (VAL). This movement takes place along the geodesic path, as shown in Figure 5b.

The process continues until any resource has its geodesic radius not exceeding threshold Δ. This process will be executed until termination conditions hold.

5. Simulation and results

The above algorithm has been simulated to explain its working for $\mathbf{n} = \mathbf{80}$ demand points and $\mathbf{p} = \mathbf{5}$ resources. The threshold value $\boldsymbol{\Delta}$ has been set to $\mathbf{170}$. The randomly generated initial solution is shown in Figure 4. The initial solution converges towards optimal locations during iterations of the algorithm.

During an iteration of the algorithm, the radius of each existing resource in $\mathbf{C}$ is calculated. This radius $\mathbf{dist_i}$ of resource $\mathbf{c_i}$ represents the distance from $\mathbf{c_i}$ to the farthest demand point in $\mathbf{VR_i}$. The radius $\mathbf{dist_i}$ for $\mathbf{i} = \{\mathbf{1}, \mathbf{2} \ldots \mathbf{p}\}$ is shown in Table 1 during iterations of the algorithm. From Table 1, we see that during the first iteration $\mathbf{dist_5}$ is largest, which requires moving resource $\mathbf{p_5}$. Finding the direction of movement requires determining the longest edge in Delaunay triangulation. The length of Delaunay edges is given in Table 2. Using Tables 1 and 2, it is determined that $\mathbf{p_5}$ should be moved towards $\mathbf{p_3}$ by 61 (*Diff* for the first iteration).

The numbers marked in bold in Table 1 thus determine the resource to be moved while bolded numbers in Table 2 determine the direction of movement for the selected resource. The values in parentheses in Table 1 represent the number of demand points in $\mathbf{VR_i}$ of $\mathbf{p_i}$ during that iteration. This represents the number of

Table 1. Radius for each resource in the 5-center problem and number of demand points allocated to each resource.

Iter.	$Dist_1$	$Dist_2$	$Dist_3$	$Dist_4$	$Dist_5$	DIST	Diff
1	125.92(12)	108.78(15)	123.49(19)	197.04(11)	**231(23)**	231	61
→Number of resources (p)					Resource p5 needs to be moved		
2	125.92(17)	108.78(14)	140.13(20)	197.04(11)	**208.03(25)**	208.03	38.03
					Resource p5 needs to be moved		
3	125.92(20)	107.32(15)	140.13(24)	**197.04(11)**	134.43(20)	197.04	27.04
					Resource p4 needs to be moved		
4	125.92(20)	107.32(15)	140.13(23)	**170(12)**	134.43(20)	170	0
					No further movements are required		

Table 2. Length of edges in Delaunay triangulations, thus determining the direction of movement for resources obtained from Table 1.

Iter.	What to move	P_1	P_2	P_3	P_4	P_5
1	$\mathbf{P_5}$	179.06	151.82	**207.32**	X	X
	Resource p5 moved in the direction of p3					
2	$\mathbf{P_5}$	**193.17**	105.06	110.44	X	X
	Resource p5 moved in the direction of p1					
3	$\mathbf{P_4}$	X	X	56.32*	X	X
	Resource p4 moved towards farthest demand point					

*Represents only one edge in the Delaunay triangulation.

demand points allocated to each resource. During iteration of the algorithms, the resources continuously change their locations, thus changing the number of demand points in $\mathbf{VR_i}$.

The same process is repeated during subsequent iterations of the algorithm. Using Tables 1 and 2, the selected resource is moved towards a particular direction by $Diff$. It is observed from the illustration that $Diff$ significantly converges to zero, thus reducing the number of iterations. As shown, $Diff$ converges from 61 to 0 in four iterations only.

Thus, we have successfully simulated the proposed algorithm. Contrary to existing approaches, it is seen that the initial solution rapidly converges towards an optimal solution in the proposed approach [9,13]. During comparative analysis, it is observed that the proposed approach outperforms the existing approach [9]. The results of comparison are shown in Figure 6. Figures 6a and 6b illustrate the number of iterations and rate of convergence, respectively.

6. Conclusion and future work

In this paper, we have implemented the p-center for the nonconvex demand region. The proposed approach utilizes geometric structures like the Voronoi diagram and Delaunay triangulation of the demand points. An algorithm has been proposed and illustrated for the same. It is observed that the proposed algorithm outperforms the existing approach. It is also observed that using **Diff** as the magnitude of movement results in a higher convergence rate. The proposed algorithm is also capable of estimating an optimal value of **p** for p-center problems. The work can be further extended to reduce the number of iterations by estimating initial solutions, thus further reducing the execution time of the algorithm. The proposed approach can also be extended in the direction of solving the polygonal p-center with constraints.

References

[1] Yen WCK. The connected p-center problem on block graphs with forbidden vertices. Theor Comput Sci 2012; 426: 13-24.

[2] Frederickson G. Parametric search and locating supply centers in trees. Lect Notes Comp Sci 1991; 519: 299-319.

[3] Foul A. A 1-center problem on the plane with uniformly distributed demand points. Oper Res Lett 2006; 34: 264-268.

[4] Fowler R, Paterson M, Tanimoto S. Optimal packing and covering in the plane are NP-complete. Inform Process Lett 1981; 12: 133-137.

[5] Drezner Z. The p-centre problem–heuristic and optimal algorithms. J Oper Res Soc 1984; 35: 741-748.

[6] Elshaikh A, Salhi S, Nagy G. The continuous p-centre problem: an investigation into variable neighbourhood search with memory. Eur J Oper Res 2015; 241: 606-621.

[7] Suzuki A, Drezner Z. The p-center location problem in an area. Locat Sci 1996; 4: 69-82.

[8] Davoodi M, Mohades A, Rezaei J. Solving the constrained p-center problem using heuristic algorithms. Appl Soft Comput 2011; 11: 3321-3328.

[9] Rayco M, Francis R, Tamir A. A p-center grid-positioning aggregation procedure. Comput Oper Res 1999; 26: 1113-1124.

[10] Hwang R, Lee R, Chang R. The slab dividing approach to solve the Euclidean P-center problem. Algorithmica 1993; 9: 1–22.

[11] Elizinga J, Hearn D. Geometrical solution for some minimax location problems. Transport Sci 1972; 6: 379-394.

[12] Cheng T, Kang L, Ng C. An improved algorithm for the p-center problem on interval graphs with unit lengths. Comput Oper Res 2007; 34: 2215-2222.

[13] Dyer M, Frieze A. A simple heuristic for the p-center problem. Oper Res Lett 1985; 3: 285-288.

[14] Drezner Z. Conditional p-center problems. Transport Sci 1989; 23: 51-53.

[15] Dantrakul S, Likasiri C, Pongvuthithum R. Applied p-median and p-center algorithms for facility location problems. Expert Syst Appl 2014; 41: 3596-3604.

[16] Pollack R, Sharir M, Rote G. Computing the geodesic center of a simple polygon. Discrete Comput Geom 1989; 4: 611-626.

[17] Zarandi MHF, Davari S, Sisakht SAH. The large scale maximal covering location problem. Sci Iran 2011; 18: 1564-1570.

[18] Rourke J. Computational Geometry in C. 2nd ed. New York, NY, USA: Cambridge University Press, 1998.

[19] Berg M, Cheong O, Kreveld M, Overmars M. Computational Geometry: Algorithms and Applications. 3rd ed. Berlin: Springer-Verlag, 2008.

[20] Fortune S. A sweepline algorithm for Voronoi diagrams. Algorithmica 1987; 2: 153-174.

[21] Suzuki A, Okabe A. Using Voronoi diagrams. In: Drezner Z, editor. Facility Location: A Survey of Applications and Methods. New York, NY, USA: Springer, 1995. pp. 103-118.

[22] Huang P, Tsai Y, Tang C. A fast algorithm for the alpha-connected two center decision problem. Inform Process Lett 2003; 85: 205-210.

[23] Huang P, Tsai Y, Tang C. A near-quadratic algorithm for the alpha connected two-center problem. J Inf Sci Eng 2006; 22: 1317-1324.

[24] Davoodi M, Mohades A. Solving the constrained coverage problem. Appl Soft Comput 2011; 11: 963-969.