Applied Soft Computing 11 (2011) 3321–3328

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Solving the constrained p-center problem using heuristic algorithms

Mansoor Davoodi [a,*], Ali Mohades [a], Jafar Rezaei [b]

[a] Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran
[b] Department of Technology, Strategy and Entrepreneurship, Faculty of Technology, Policy and Management, Delft University of Technology, P.O. Box 5015, 2600 GA Delft, The Netherlands

ARTICLE INFO

Article history:
Received 27 April 2010
Received in revised form 4 November 2010
Accepted 3 January 2011
Available online 8 January 2011

Keywords:
p-Center problem
Voronoi diagram
Heuristic algorithms
Optimization

ABSTRACT

The p-center problem is one of the location problems that have been studied in operations research and computational geometry. This paper describes a compatible discrete space version of the heuristic Voronoi diagram algorithm. Since the algorithm gets stuck in local optimums in some cases, we apply a number of changes in the body of the algorithm with regard to the geometry of the problem, in a way that it can reach the global optimum with a high probability. Finally, a comparison between the results of these two algorithms on several test problems and a real-world problem are presented.

1. Introduction

One of the most important types of facility location problems is the p-center problem. The objective is to minimize the maximum distance of any demand points from the nearest facility center. Fowler et al. [1] have shown that, when the parameter p is a part of the input, the problem is *NP*-complete. The problem can be discussed in continuous or discrete space. Usually (though not always) in operations research literature, when the demand set is infinite, the location problem is called a continuous problem [2], while in computational geometry a location problem is considered continuous if the candidate location set to facility is infinite [3–5]. Agarwal and Sharir [6] studied many versions of the p-center problem with regard to the geometry of demand set and space metrics. In addition to exact algorithms, many approximation and heuristic algorithms have been suggested to solve the p-center problem [4,7–9]. Caruso et al. [10] and Cheng et al. [11] have discussed the p-center on the graphs and trees with the aim of to solve a variety of problems, such as the location of client/server problems or the location on the street line of a city. On the other hand, some papers discuss continuous p-center problems. We can divide these works into two sub-categories: (i) those that consider the demand set as mid-discrete [3,12], like location problems concerning technical supporting centers or the warehouse location between some customers, and (ii) those that consider the demand set as mid-continuous [4,13], like wireless antennas or bulb location. This paper focuses primarily on the p-center problems with a discrete demand set, formulated as follows:

If $D = \{p_1, p_2, \ldots, p_n\}$ be a set of n demand points in the plane, the goal is to find p centers (denoted by $C = \{c_1, c_2, \ldots, c_p\}$) such that the maximum distance of the demand points from their closest center is minimized:

$$Z(C) = \underset{1\le j\le p}{Min}\{\underset{1\le i\le n}{Max}\{\underset{1\le j\le p}{Min}\ d(p_i, c_j)\}\}$$
$$d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \quad (1)$$

Notice that, in the formulation of the problem presented above, the candidate positions to the location of the centers is an infinite set. The best exact algorithm that has been proposed for this problem runs in $O(n^{\sqrt{p}})$ time [14], although there are efficient algorithms for small ps. If p equals one, the minimum covering circle (MCC) problem is put forward [15,16]. After many studies, Megiddo [17] solved MCC problem in linear time using a parametric searching algorithm. In the plane and for two-center problem relatively optimal algorithms have been presented [18,19]. Likewise, an attractive version of two-center problem, *alpha*-connected two-center problem has been suggested by Huang et al. [20,21]. *Alpha*-connected two-center problem are solved by using farthest point Voronoi diagram in $O(n^2 \log^2 n)$ time.

In the constrained space, there are some constraints on the location of the facility centers, and they cannot be situated in arbitrary locations [13,22–24]. Indeed, in this model, which is one step closer to real world problems, there are some areas, usually displayed by polygons, that show some constraints, like highways, lakes, private property, etc. In addition to the applications of p-center problem

* Corresponding author.
E-mail addresses: md_monfared@yahoo.com, mdmonfared@aut.ac.ir (M. Davoodi), j.rezaei@tudelft.nl (J. Rezaei).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.01.001

in emergency facility location, like hospitals, police offices, etc., it has also been used in clustering problems, data mining, image processing, transport networks, etc. [25,26].

One of the geometric objects that have been applied with great success in location problems is the Voronoi diagram (*VD*). By using *VD*, Drezner [3] has presented two similar algorithms, which can be used for discrete demand set cases where the *VD* is only a tool for reducing the computational time. However, in continuous demand set cases, *VD* is a main tool of the algorithms. In [27,28] a heuristic *VD* method has been proposed for solving the continuous space *p*-center problem, after which it has been extended by Plastria [29] and Wei et al. [13] to solve the constrained space. Unfortunately, all heuristic Voronoi algorithms mentioned here get stuck in local optimums and have not proved helpful in overcoming this problem, except repeating the algorithm.

In this paper, we begin with the description of a heuristic algorithm [23] that is able to solve the *p*-center problem in a constrained space with discrete demand set. We then develop the algorithm in such a way that it reaches to the global optimum with a high probability. This approach, which results in a non-deterministic algorithm, is much more powerful than the original algorithm. We compare the results of the two algorithms by solving several test problems.

This paper is divided into five sections. Section 1 reviews the *p*-center literature and introduces the main goal of the paper. Section 2 describes *VD* and a heuristic Voronoi diagram algorithm for solving the constrained *p*-center problem, while Section 3 contains a non-deterministic version of the algorithm that makes it possible to find the global optimum solution with a high probability. The simulation results of the algorithms and a real-world case problem are presented in Section 4, while the conclusions of this paper and possible avenues for future research are discussed in Section 5.

2. Heuristic algorithm for solving the *p*-center problem

This section begins with a brief introduction of the Voronoi diagram, after which a compatible Voronoi diagram algorithm as presented in [23] to solve the constrained version of *p*-center problem is described.

2.1. Voronoi diagram

One of the most practical geometric structures is the Voronoi diagram (*VD*), which is based on the set of points in the *d*-dimensional. *VD* includes all problems in nearest or farthest neighborhood search problems (for more information, see [30,31]).

Definition. If $D = \{p_1, p_2, \ldots, p_n\}$ is a set of *n* points (or sites) in a given plane, the Voronoi region of any point, p_i, denoted by VR_i, is a convex polygon such that all points in it are closer to p_i than other points of *D*. Also, the *Voronoi diagram* of *D* (denoted by *VD*(*D*)) is defined as the set of all points in the plane that have more than one nearest site.

Fortune's algorithm [32] can construct the *VD* of *n* points in the plane in O(*n* log *n*) time complexity. The most important property of *VD* is that it can answer the nearest (or farthest) query point in O(log *n*) time.

2.2. Heuristic algorithm for p-center problem

If $D = \{p_1, p_2, \ldots, p_n\}$ is a set of *n* demand points in the plane and $C = \{c_1, c_2, \ldots, c_p\}$ is a set of *p* facility centers that must be positioned among the points of *D*, there exist some locations in the constrained space that are usually modeled by polygons in which the centers cannot be positioned. Indeed, a feasible solution of these problems

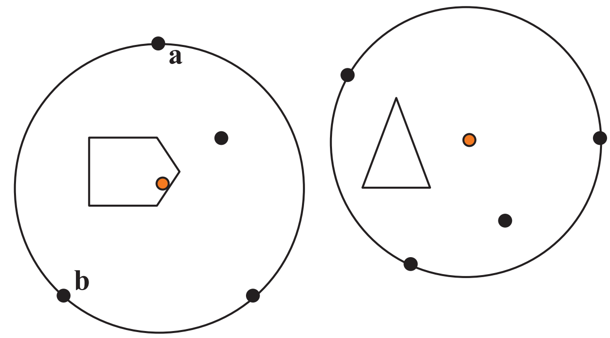

Fig. 1. An infeasible solution for a 2-center problem with 8 demand points and two constrained polygons. The left center lies in the pentagon.

is one such that all centers are located outside or on the boundary of such polygons.

If s_1 is the fitness of the solution of *p*-center, (or *Z*(*C*) in Eq. (1)), without considering the constraints, and s_2 is the fitness of solution of the constrained *p*-center problem, it is clear that $s_2 \geq s_1$. Indeed, in the 1-center problem, if the center of the minimum covering circle is a feasible solution, then $s_1 = s_2$; otherwise $s_1 > s_2$. It has been proven that the minimum feasible covering circle can be determined by one of two possible cases [13]:

- The feasible center is one of the points of the boundary line segments of the constraints closest to one of the demand points.
- The feasible center is one of the points of the intersection of the bisector of two demand points with the boundary of the constraints.

If *h* is the number of points on the convex hull and *m* is the number of segments of constraints, the minimum feasible covering circle can be obtained in O(mh^2) time. For example, Fig. 1 shows an infeasible solution for constrained 2-center problems. In this example, the left hand center lies in the constrained pentagon (case $s_2 \geq s_1$). To find a feasible solution, it is sufficient to move the infeasible center along the bisector of the two points (*a*) and (*b*).

In the following section, we briefly describe an iterative algorithm called a heuristic Voronoi diagram algorithm 1 (*HVDA1*) for solving the constrained *p*-center problem [23].

Heuristic Voronoi diagram algorithm 1 (*HVDA1*)
Inputs: Demand set *D*; number of centers, *p*; and constraint polygons.
Outputs: A solution for the *p*-center problem.
Step 1 Compute the convex hull of *D*.
Step 2 Generate a random solution, $C = \{c_1, c_2, \ldots, c_p\}$, for the problem in the convex hull of *D*.
Step 3 Compute the Voronoi diagram of the solution, *VD*(*C*).
Step 4 For $j = 1, 2, \ldots, p$; find all demand points in the Voronoi region of c_j and put them in a new set S_j.
Step 5 For $j = 1, 2, \ldots, p$; Compute the minimum feasible covering circle of S_j and set its center as the new c_j.
Step 6 If the termination conditions hold, return the set *C* and finish the algorithm; otherwise go to Step 3.

If a point lies on the *VD*(*D*) in step 4, assign it arbitrarily to a related Voronoi regions.

If the initial solution is selected totally at random, it is possible that some of the S_js are an empty set, which is why, in step 2, we can select random *p* points from *n* demand points, or use one of the seed points algorithms of the *p*-center problem that have been suggested by Pelegrin and Canovas [33]. As shown in [23], the total complexity of the algorithm in each iteration is $O(mn^2)$, where *n* is the number of demand points, and *m* is the number of constraints segments.