Applied Soft Computing 11 (2011) 963–969

Contents lists available at ScienceDirect

Applied Soft Computing

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

Solving the constrained coverage problem

Mansoor Davoodi*, Ali Mohades

Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 28 May 2008
Received in revised form 13 January 2010
Accepted 17 January 2010
Available online 25 January 2010

Keywords:
Coverage problem
p-Center problem
Voronoi diagram
Heuristic algorithms
Sensor networks

ABSTRACT

Coverage problem which is one of the challenging problems in facility location studies, is *NP*-hard. In this paper, we focus on a constrained version of coverage problem in which a set of demand points and some constrained regions are given and the goal is to find a minimum number of sensors which covers all demand points. A heuristic approach is presented to solve this problem by using the Voronoi diagram and *p*-center problem's solution. The proposed algorithm is relatively time-saving and is compared with alternative solutions. The results are discussed, and concluding remarks and future work are given.

1. Introduction

The coverage problem is among the most important practical subjects relating to the sensor networks and facility location that have been studied in the past years. Generally, it is referred to as the k-coverage form and it has been discussed in relation to the continuous vs. discrete, constrained vs. unconstrained and weighted vs. unweighted space. In the discrete space, there are n targets (or demand points) and the goal is to cover them by sensors that have an effective sense radius of r. Any sensor can be displayed as the center of a circle with a radius r (denoted by r-*sensor*). As such, any demand point is *covered* by a sensor if its distance from the sensor is equal to or less than r. In the k-coverage problem, each demand point is covered by at least k different r-sensors, where the number of sensors is minimized. Because this problem is generally speaking *NP*-hard [1], many approximate and heuristic algorithms have been suggested to solve it (for example, see [2–6]).

In the simplest case, when k equals one, the 1-coverage problem, which we will refer to in this paper as the coverage problem for the sake of simplicity, emerges. The goal is to determine the minimum number of r-sensors such that they cover all demand points. Current and O'Kelly [7] studied continuous space, where the demand points constitute an infinite set and are modeled by polygons. In general, the coverage problem in continuous space can be divided into complete and approximate coverage and it can be solved by using a grid-based or partitioning approach [7,8]. In the constrained versions of such an approach, which are one step closer to real-world problems, there are some constraints on the position of sensors. These constraints, which are usually shown by polygons, represent actual constraints like highways, private property, lakes, etc. [9].

Another type of facility location problems, of which various versions have been studied in recent years, is the p-center problem [10–13], which to a certain extent resembles the coverage problem identified above. Here, the goal is to find exact p-circles that cover all demand points and minimize the maximum radius of the circles. The p-center problem belongs to the *NP*-complete class of problems [14] and is applied in emergency facilities like police offices and hospitals.

Recently, Wei and Murray have used a heuristic algorithm to try and solve the constrained p-center problem in the continuous space [15]. Their method has been extended and was applied to a constrained space to determine location of some warning sirens [16]. Other locations where the coverage problem occurs, apart from in sensor networks and wireless antennas [17], include military applications, object tracking, data gathering, etc. [18–20]. With regard to the widespread occurrence of the coverage problem, the object of this paper is to present a heuristic algorithm called Heuristic Algorithm for the Coverage Problem (HACP) to solve the constrained coverage problem in discrete space, using the p-center solving algorithm. By using the output of the p-center problem and a binary search technique, HACP controls the number of centers (facilities) and is able to achieve the desired result.

This paper is structured as follows. In the first section, we discuss the coverage and p-center problems and provide a brief introduction of the contribution of this paper. In the second section, we propose a heuristic Voronoi diagram algorithm to solve the

* Corresponding author.
E-mail addresses: md_monfared@yahoo.com, mdmonfared@aut.ac.ir (M. Davoodi), Mohades@aut.ac.ir (A. Mohades).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.01.016

constrained p-center in discrete space on the basis of [15]. Our algorithm in greater detail is proposed in the third section, and its simulation results are provided in the fourth section. In section five, we discuss on the some issues of the algorithm, and finally conclusions and suggestions for future research are presented.

2. Heuristic Voronoi diagram algorithm to solve the *p*-center problem

In this section, we begin by discussing a theorem that describes the relationship between two facility location problems, the coverage problem and the p-center problem. Then we introduce Voronoi diagram and describe the compatible heuristic Voronoi diagram algorithm designed to solve the constrained p-center problem in discrete space.

2.1. Coverage and p-center problems

For the numbers $d \geq 2$ and $p \geq 1$ and a real value $r > 0$, the two above-mentioned problems are defined as follows:

p-Center problem: n demand points in d-dimensional space and parameter p are given. We want to determine p-center (facility) points such that the maximum distance of each demand point to the nearest center is minimized.

Coverage problem: n demand points in d-dimensional space and parameter r are given. We want to find the minimum number of r-sensors which is needed to cover all demand points.

The p-center problem is equivalent to find p d-dimensional spheres with a radius of r^* such that they cover n demand points and r^* is minimum, while the coverage problem is equivalent to finding p^* spheres with radius r such that they cover n demand points and p^* is minimum.

Theorem: In a fixed-dimensional space, both p-center and coverage problems fall into the same complexity class of problems.

Proof: The fitness of the optimum p-center's solution, r^*, can be guessed; in d-dimensional space r^* belongs to a discrete set with cardinality $O(n^{d+1})$. For example, in a plane, r^* can be determined from the radius of all minimum circles that cover any two or three points in the demand set, which means that r^* belongs to a set with cardinality $O(n^2) + O(n^3) = O(n^3)$.

Let us assume that the coverage problem can be solved in $f(n,r)$ time. If R is the set of all possible radius solutions, r^*, since $|R| = O(n^{d+1})$, the p-center problem can be solved in $O(n^{d+1}f(n,r))$ time by using the coverage problem solution. Furthermore, In other hand, assume that the p-center problem can be solved in $f(n,p)$ time. The minimum number of r-spheres, p^*, always satisfies $1 \leq p^* \leq n$. This means that it is possible to use a sequential search to find the minimum number of spheres that have a radius smaller than or equals to r. Hence, the total time complexity for the coverage problem is $O(n\,f(n,r))$. It is clear that both cases above are reducible to each other in polynomial time for a fixed d, which means proof is complete.

2.2. Voronoi diagram

Voronoi diagram (VD) is one of the most practical geometric structures. VD is defined on the set of points in d-dimensional space, and includes all the nearest or the farthest neighborhood problems [21,22].

Definition: Let $D = \{p_1, p_2, \ldots, p_n\}$ be a set of n points (or sites) in the plane. The Voronoi region of any point, p_i, which is denoted by VR_i, is a convex polygon region contains all points in the plane which p_i is their nearest point respect to other points of D. The locus of the points that has more than one nearest site is called the Voronoi diagram of D and is denoted by VD(D).

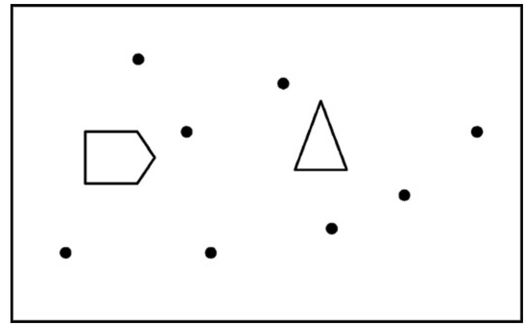

Fig. 1. Demand points with the constraints of the problem.

Fortune's algorithm can construct the VD of n points in the plane in $O(n \log n)$ time. The most important property of VD is that it can determine the nearest (or farthest) point in $O(\log n)$ time [22].

2.3. Heuristic algorithm for p-center problem

Let us assume that $D = \{p_1, p_2, \ldots, p_n\}$ 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 points (or centers) that must be positioned among the demand points. In the constrained version, there are some locations that are usually modeled by polygons in which the centers cannot be positioned. Indeed, a feasible solution to this problem is one where all the centers are located outside or on the boundary of such polygons.

Fig. 1 shows eight demand points and two constraints of the work area. In this example, the centers cannot be positioned inside the triangle or pentagon.

Suppose that s_1 and s_2 be the fitness of the p-center problem's solutions, before and after the considering the constraints, respectively. Clearly, $s_2 \geq s_1$. Indeed, in the 1-center problem, if the center of the minimum covering circle is a feasible solution, so $s_2 = s_1$; otherwise $s_2 > s_1$. It has been proved that the minimum feasible covering circle can be determined by one of the two following cases [15]:

- Closest points of constraint boundary segments to one of the demand points.
- Intersection point between bisector of two demand points and the constraint boundary segments.

Let h be the number of points on the convex hull and m be the number of constraints segments. So, the minimum feasible covering circle can be obtained in $O(mh^2)$ time. For example, Fig. 2 shows two solutions to the unconstrained p-center problem for p = 1 and 2. If we consider the constraints, the solution to case p = 1 (left hand side) remains a feasible center ($s_2 = s_1$), but in the case of p = 2, one of the two centers becomes infeasible. In this case, it is sufficient to move the infeasible center along the bisector of the two points a and b.

By using two approaches which have been suggested in [14,15], and combining them, we propose an iterative algorithm called the heuristic Voronoi diagram algorithm (HVDA) to solve the constrained p-center problem.

HVDA

Inputs: Demand set D; number of centers, p; and constraint polygons.

Outputs: A solution to the discrete p-center problem and the maximum radius of circles.

Step 1 Compute the convex hull of D.

Step 2 Generate a random solution, $C = \{c_1, c_2, \ldots, c_p\}$, to the problem with regard to the convex hull of D.

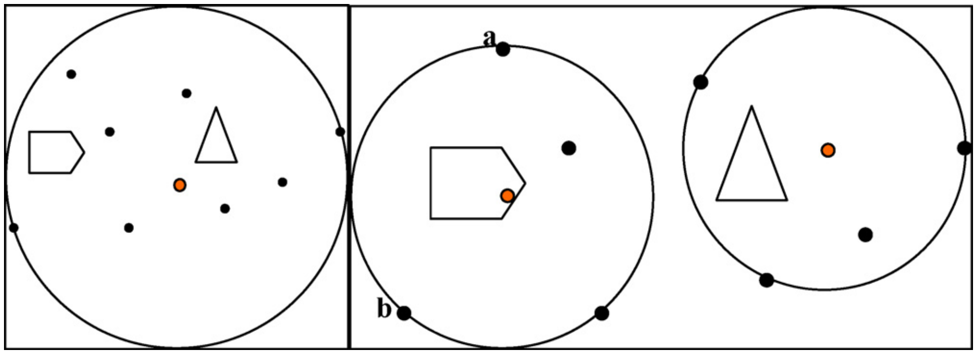

Fig. 2. Solving the p-center problem for two cases $p = 1$ and 2.

Step 3 Compute the Voronoi diagram of the solution, VD(C).
Step 4 Find all demand points in the Voronoi region of c_j and put them in S_j, for $j = 1, 2, \ldots, p$.
Step 5 Compute the minimum feasible covering circle of S_j and set its center as the new c_j, for $j = 1, 2, \ldots, p$. Set $r_{\max}$ as the maximum radius of circles.
Step 6 If the termination condition holds, return set C and value $r_{\max}$ and finish the algorithm; otherwise go back to step 3.

If, at step 4, a demand point lies on the Voronoi diagram of D, assign it arbitrarily to one of the S_js.

2.4. Analysis of algorithm

HVDA has a main loop (steps 3–6) and two preliminary steps. Step 1 computes the convex hull of D and runs in $O(n \log n)$ time for $n = |D|$ points. Step 2 determines a random solution (feasible or infeasible) in the convex hull of D. If the solution is infeasible, it is changed to a feasible solution in step 5. If the initial solution is selected totally at random, it is possible that one of the S_js is an empty set in step 4. Thus, we can select randomly p points from n demand points in step 2, or use one of the seed point algorithms of the p-center problem [23]. In the main loop of the algorithm, step 3 contains $O(p \log p)$ computations by Fortune's algorithm. Each nearest query point needs to $O(\log p)$ computations by using a VD(C), which means that the total complexity of step 4 is $O(n \log p)$. The minimum covering circle of set S_j can be computed in $O(n_j)$ computation by using Megiddo's algorithm [24], where $n_j = |S_j|$. If the center lies on one of the constraint polygons, we can find a feasible solution in $O(m\, n_j^2)$, where m is the number of segments of the constraint. Consequently, step 5 runs in $\sum_{j=1}^{p} O(mn_j^2) = O(mn^2)$. The algorithm can terminate after a predefined maximum number of iterations or when set C remains unchanged in two successive iterations. As a result, the total complexity of the algorithm in each iteration is:

$$O(p \log p + n \log p + mn^2 + p) \overset{n \geq p}{=} O(m\, n^2)$$

Because of the approach HVDA adopts to the problem, it has a high convergence rate, even for large-sized of inputs, and it can reach the desired result in a small number of iterations. Since the efficiency of HVDA depends on the initial centers (like most heuristic algorithms), it may sometimes become stuck in a local optimum. A natural proposed approach for this problem is that the algorithm is repeated with several random seed points and finally returns the best solution.

3. Heuristic algorithm to solve the coverage problem

As mentioned in the previous section, the optimal number of circles in the coverage problem is always between one and the number of demand points. Since the goal is to determine the minimum number of circles, a binary search on the number of circles can be used to find the optimal number. In general, HVDA starts with $p = \lfloor n/2 \rfloor$ at the first time. If the minimum radius of the largest circle, $r_{\max}$, is greater than r (radius of sensors), some centers should be added, Otherwise, we can reduce the number of centers. In the following, a step-by-step form of the algorithm is described.

Heuristic Algorithm designed to solve the Coverage Problem (**HACP**)

Inputs: Demand set D; radius of sensors, r; and constraint polygons.
Output: A solution to the discrete coverage problem.
Step 1 Set the value of *low* to 1 and *high* to n, and set $p = \lfloor (low + high)/2 \rfloor$.
Step 2 Compute the convex hull of D.
Step 3 Generate a random solution, $C = \{c_1, c_2, \ldots, c_p\}$, to the p-center problem inside the convex hull of D.
Step 4 Compute the Voronoi diagram of the solution, VD(C).
Step 5 Determine all demand points in the Voronoi region of c_j and put them in S_j, for $j = 1, 2, \ldots, p$.
Step 6 Compute the minimum feasible covering circle of S_j and set its center as the new c_j, for $j = 1, 2, \ldots, p$. Set $r_{\max}$ as the maximum radius of circles.
Step 7 If the termination condition of the p-center algorithm holds, go to step 8. Otherwise go to step 4.
Step 8 If *low* equals p, return set C; otherwise, regarding the value of $r_{\max}$, run one of the step 9 or step 10.
Step 9 If $r_{\max} > r$ run the following two substeps.
Step 9-1 Set $low = p$ and $p' = \lfloor (low + high)/2 \rfloor$.
Step 9-2 Add $(p' - p)$ circles to C and set $p = p'$.

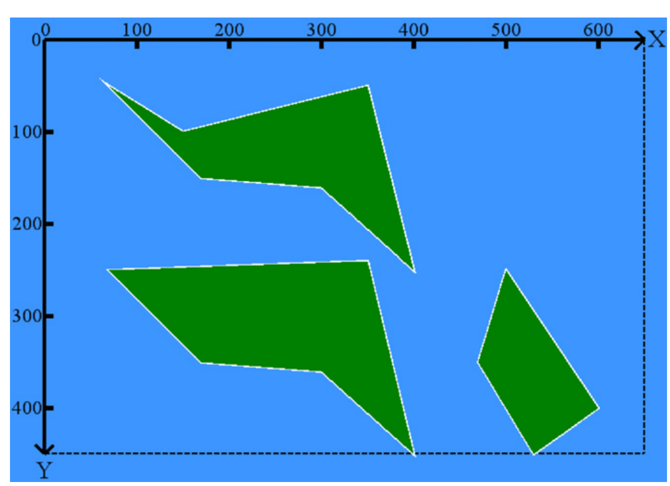

Fig. 3. Work space of the test problems.

Step 10 If $r_{\max} \leq r$ run the following two substeps.
Step10-1 Set $high = p$ and $p' = \lfloor (low + high)/2 \rfloor$.
Step10-2 Delete $(p' - p)$ circles from C and set $p = p'$.
Step 11 Go back to step 4.

The algorithm described above is based on a binary search on the maximum radius of circles reported by HVDA. The main loop (steps 4–11) repeats exactly $(\log n)$ times and returns the smallest set of circles covering n demand points. HVDA is implemented in step 4

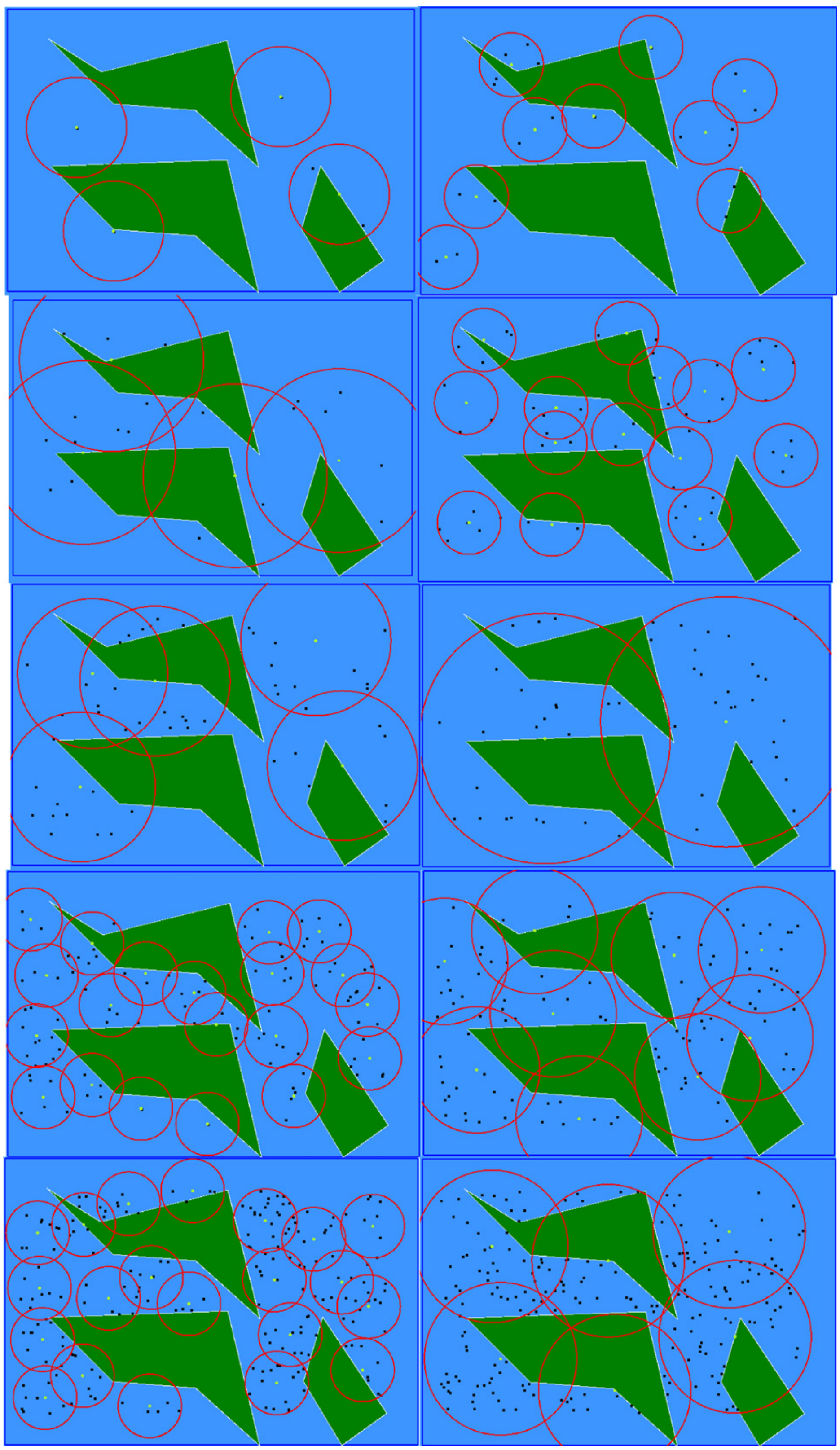

Fig. 4. Results of HACP (see Table 1 for more descriptions).

Table 1
The number of demand points (n) and size of radius (r) in each test.

	[Row, column]									
Position	[1,1]	[1,2]	[2,1]	[2,2]	[3,1]	[3,2]	[4,1]	[4,2]	[5,1]	[5,2]
n	5	20	20	50	50	50	100	100	200	200
r	80	50	150	50	120	200	50	100	50	120
Time (s)	0.02	0.08	0.03	0.37	0.20	0.03	2.08	1.35	16.43	12.90
#Repeat	2	6	4	15	7	3	27	18	48	34

upto step 7, and returns the maximum radius of p-circles, $r_{\max}$, and their positions, or set C. Taking into account the mentioned theorem, if HVDA solves the p-center problem optimally, HACP solve the coverage problem optimally, too. The time complexity analysis of HACP is similar to HVDA, and in fact, HACP is $O(\log n)$ times as expensive as HVDA. Due to the high speed of HVDA, HACP is fast, as well.

If we ignore the initial solution step in HVDA, because the Voronoi diagram and the minimum covering circle are unique, HVDA is a deterministic algorithm. In each iteration, HACP runs either one of the two steps 9 or 10. In step 9, some centers are added to set C, while step10 deletes some of them. The maximum of these additions or deletions, which is $n/4$, occurs in the first iteration. HACP behaves in a non-deterministic manner inasmuch as centers can be randomly added to the convex hull of D in step 9, while they can be randomly deleted in step10. This non-deterministic approach can help HACP obtain the global optimum even if, in some cases, HVDA is stuck in a local optimum. However, it is possible to allow for some additional complexity and run these steps using an effective approach. For example, if k circles need to be added to set C, by using *MaxHeap* data structure, we can find k largest circles in $O(k \log p)$ time. We then add these k new circles to the Voronoi region of the centers of them. Similarly, to delete the circles, it is possible to perform the following process k times:

1. Find the two nearest centers in C, and denote them by c_1 and c_2.
2. Merge the Voronoi regions of c_1 and c_2 and set $S = s_1 \cup s_2$, where s_i denotes the set of all demand points in the Voronoi region of c_i.
3. Delete c_1 and c_2 and add the center of the minimum feasible covering circle S of to C.

The process described above can be handled in $O(p \log p)$ time, where $p = |C|$ using the Voronoi diagram. Consequently, the total deletion process takes $O(k p \log p)$ time. Although, different methods can be used in the addition and deletion process, with the heuristic algorithms, and increase in the convergence rate diminishes the fitness of the ultimate solution.

4. Simulation result

In this section, we apply HACP on the several random test problems using the above-mentioned addition and deletion rule. All test problems are dealt with in a X–Y rectangular shape of size 650×450 (X is left-right and Y is top-down axis) and have three constraint polygons. Fig. 3 shows the work space. Constrains are indicated in by green.

The algorithm is tested with various numbers of demand points, n, and sensor radius, r. In each run, n random demand points are generated in work space first, after which HACP is run with several predefined sensor radii. In all simulations, a hybrid terminating condition of a maximum of 100 iterations and solution remaining unchanged in three consecutive iterations of the p-center algorithm is used. Table 1 shows the values of n and r in each of the subfigures in Fig. 4, in which the subfigures are addressed in a matrix form. For example, [1,1] represents the topmost left subfigure, while [5,2] represents the bottommost right subfigure. The black points are demand points, and sensors are indicted by yellows center and red boundaries in the subfigures. The third row of table shows runtime of each simulation. In each simulation coverage algorithm using HVDA several times, the average number of repetitions of HVDA in each simulation reported in the last row of Table 1.

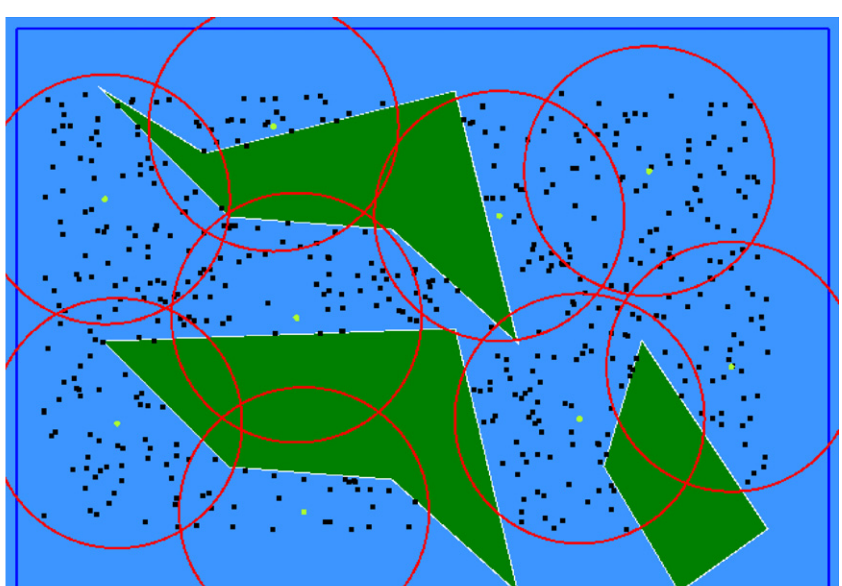

Fig. 5. High density demands, $n = 500$ and $r = 100$.

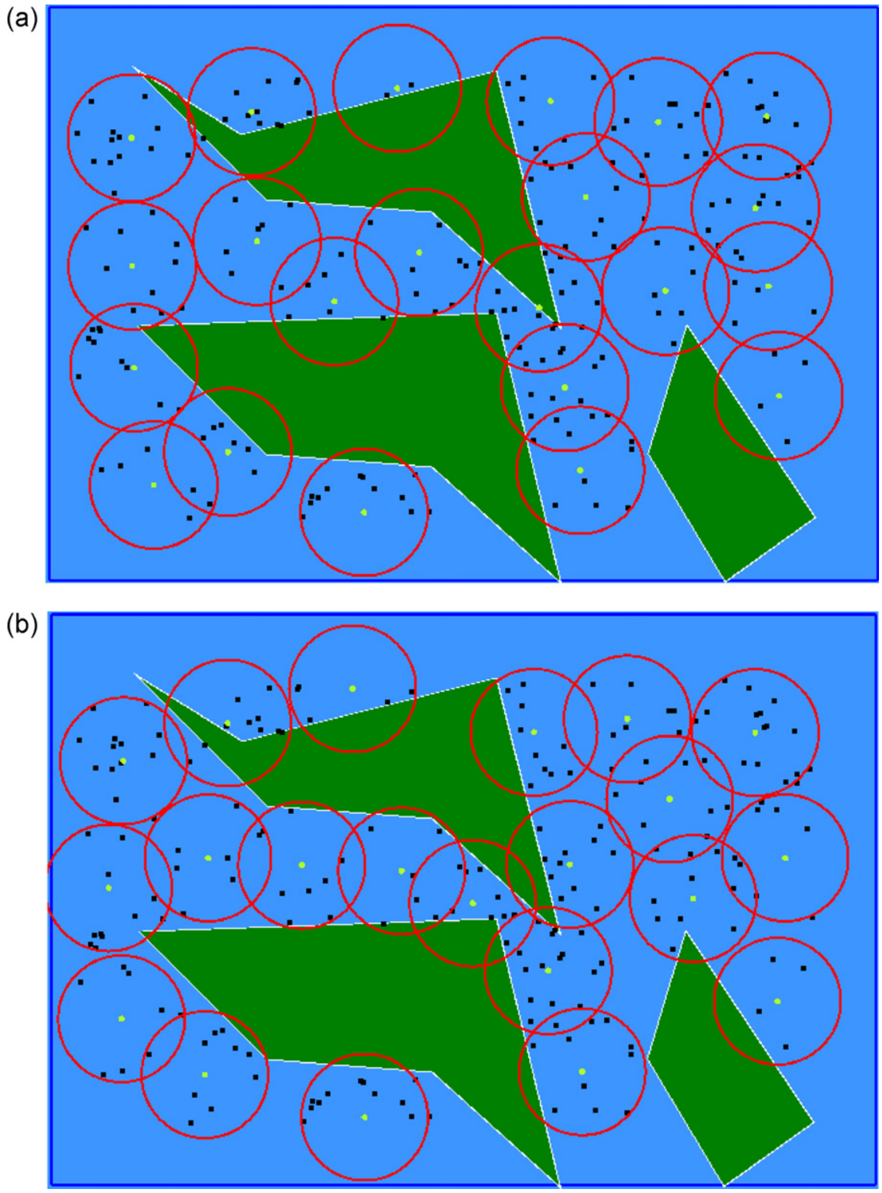

Fig. 6. Two solutions of a problem with 200 demand points and 50 for sensor radius. The number of sensors in figure (a) is 22 and in figure (b) is 21.

Although, HACP is able to determine the global minimum solution in most tests, it gets stuck in a few cases, especially in problems with large n and small r. of course, when the density of demand points is very large and uniformly distributed, in which the problem is similar to the continuous version of the coverage problem. Fig. 5 shows result of algorithm for 500 demand point with r equals 100. This result obtained in about 5 min.

As mentioned above in some cases algorithm get stuck in local optimum likes most heuristic algorithms. Generally, in such case, for tremendous confidence in global optimality of solution, we can repeat algorithm several times. Fig. 6 shows two result of algorithm for a set of 200 demand points with r equals 50. In first (Fig. 6(a)) run the minimum number of reported for p is 22 and in second (Fig. 6(b)) is 21. The authors propose other way to improve the quality of solution in [26].

The simulations were implemented using C# programming language on a 1.8 GHz CPU and 512 MB RAM computer under the Windows *XP* operating system. To compute MCC and VD, Elizinga and Hearn [25], and Fortune's algorithms [22] are used.

5. Discussion

The presented algorithm can obtain the global optimum coverage with a high degree of probability. Since the algorithm is based on the p-center, if the solution to p-center is optimum, then so is the coverage solution. On the other hand, if the solution to the p-center be a local optimum in some iterations of the algorithm, it is possible that the solution to the coverage problem is not a global optimum. A general approach to adjusting this problem in heuristic algorithms is to repeat the algorithm and select the best solution. We point out that it is not necessary for the p-center algorithm to determine the global solution in all iterations to obtain the optimum global solution to the coverage problem. This means a slight improvement in the p-center algorithm resulted to a considerable improvement in coverage algorithm. Since all steps of p-center algorithm have deterministic manner except the initial step, it can be improved by adding a random step to increase the amount of explorers in the search space [26] such as mutation operation in genetic algorithms.

6. Conclusion

In this paper, we have proposed a fast and fairly powerful heuristic Voronoi diagram algorithm to solve the constrained coverage problem in the discrete space. We have shown that the complexity of the algorithm is low degree polynomial time, and we have tested it on some problems involving large numbers of demand points.

We have used Euclidean metric in distance function. However, in some real-world coverage problems, combining the shortest geodesic path distance with Euclidean distance is more practical, as becomes clear from studies like [27] on the *p*-center problem. This combinatorial metric indicates an interesting problem may arise with regard to the constrained coverage problem. Also, with respect to the improvements we have made to the weighted Voronoi diagram, one future area for research may be to create a compatible version of these algorithms involving weighted demand points or non-uniform density demand space.

Acknowledgment

We are grateful Jafar Rezaei and unknown reviewers for their helps and useful suggestions.

References

[1] S. Yang, F. Dai, M. Cardei, J. Wu, On connected multiple point coverage in wireless sensor networks, Journal of Wireless Information Networks 13 (4) (2006) 289–301.
[2] M. Bagheri, Efficient k-coverage algorithms for wireless sensor networks and their applications to early detection of forest, Master thesis, School of Computing Science, Simon Fraser University, 2007.
[3] A. So, Y. Ye, On solving coverage problems in a wireless sensor network using Voronoi diagrams, in: Proceedings of Workshop on Internet and Network Economics (WINE'05), Hong Kong, December, 2005.
[4] F. Ye, G. Zhong, S. Lu, L. Zhang, PEAS: a robust energy conserving protocol for long-lived sensor networks, in: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), Rhode Island, 2003.
[5] D.S. Hochbaum, W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, Journal of ACM 32 (1985) 130–136.
[6] D.S. Hochbaum, W. Maass, Fast approximation algorithms for a nonconvex covering problem, Journal of Algorithms 8 (1987) 305–323.
[7] J. Current, M. O'Kelly, Locating emergency warning sirens, Decision Science 23 (1992) 221–234.
[8] A.T. Murray, M.E. O'Kelly, Assessing representation error in point-based coverage modeling, Journal of Geographical Systems 4 (2002) 171–191.
[9] M. Davoodi, A. Mohades, J. Rezaei, A Genetic Algorithm For the Constrained Coverage Problem, Applications of Soft Computing, AISC 58, Springer-Verlag, Berlin–Heidelberg, 2009, pp. 347–356.
[10] N. Mladenović, M. Labbe, P. Hansen, Solving the p-center problem with tabu search and variable neighborhood search, Networks 42 (2003) 48–64.
[11] S. Elloumi, M. Labbe, Y. Pochet, A new formulation and resolution method for the p-center problem, INFORMS Journal on Computing 16 (1) (2004) 84–94.
[12] M.S. Daskin, A new approach to solving the vertex p-center problem to optimality: algorithm and computational results, Communications of the Operations Research Society of Japan 45 (2000) 428–436.
[13] Z. Drezner, The p-center problem—heuristics and optimal algorithms, Journal of the Operational Research Society 35 (1984) 741–748.
[14] R.J. Fowler, M.S. Paterson, S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete, Information Processing Letters 12 (1987) 133–137.
[15] H. Wei, A.T. Murray, Solving the continuous space p-centre problem: planning application issues, IMA Journal of Management Mathematics 17 (2006) 413–425.
[16] A. Suzuki, A. Okabe, Using Voronoi diagrams, in: Z. Drezner (Ed.), Facility Location: a Survey of Applications and Methods, Springer, New York, 1995, pp. 103–118.
[17] C. Huang, Y. Tseng, The coverage problem in a wireless sensor network, Mobile Networks and Applications 10 (2005) 519–528.
[18] D.L. Hall, J. Llinas, Handbook of Multisensor Data Fusion, CRC Press, New York, 2001.
[19] S. Kumar, T.H. Lai, J. Balogh, On k-coverage in a mostly sleeping sensor network, in: Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM'04), Philadelphia, PA, USA, 2004, pp. 144–158.
[20] Z. Zhao, R. Govindan, Understanding packet delivery performance in dense wireless sensor networks, in: Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (Sensys'03), Los Angeles, CA, 2003, pp. 1–13.
[21] J. O'Rourke, Computational Geometry in C, 2nd ed., Cambridge U. Press, New York, 1998.
[22] M. De Berg, M. van Kreveld, M. Overmans, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, 2nd rev. ed., Springer-Verlag, Berlin, 2000.
[23] B. Pelegrin, L. Canovas, A new assignment rule to improve seed points algorithms for the continuous k-center problem, European Journal of Operational Research 104 (1998) 366–374.
[24] N. Megiddo, Linear-time algorithms for linear programming in R^3 and related problems, SIAM Journal of Computing 12 (1983) 759–776.
[25] J. Elizinga, D.W. Hearn, Geometrical solution to some minimax location problems, Transport Science 6 (1972) 379–394.
[26] M. Davoodi, A. Mohades, J. Rezaei, Solving the constrained p-center problem using heuristic algorithms, Journal of the Operational Research Society (submitted for publication).
[27] J.M. Robert, G. Toussaint, Computational geometry and facility location, Technical Report SOCS 90.20, in: Proceedings of the International Conference on Operations Research and Management Science, McGill Univ., Montreal, PQ, 1990, pp. B1–B19.