Smallest Square Covering k Points for Large Value of k

Priya Ranjan Sinha Mahapatra

Abstract Given a set of n planar points, a positive integer $k (1 \leq k \leq n)$ and a geometric object, the objective of the k-cover problem is to find a smallest object such that it covers at least k input points. A deterministic algorithm is proposed to solve the k-cover problem when the object is an axis-parallel square and $k > \frac{n}{2}$. The time and space complexities of the algorithm are $O(n + (n-k)\log^2(n-k))$ and $O(n)$ respectively.

Keywords Computational geometry · Facility location · Minimum enclosing square · Matrix search

1 Introduction

Let S be a set of n planar input points. The points set S is said to be covered by a geometric object C is each point of P lies within the interior of C or on the boundary of C. The process of covering a set S by an object C is called full covering and the corresponding problem is known as full covering problem. The full covering problem has been well studied in different areas such as facility location, operation research, computational geometry etc. The covering objects used are a circle [1, 2], a rectangle [3], a square [4], a triangle [5], a circular annulus [6], a rectangular annulus [7], a rectilinear annuls [8] etc. Another purpose of covering is to cover the input points partially and corresponding problem is known as partial covering problem. Therefore, one objective of the partial covering problem is to find a smallest object of given type that covers at least k points of S. The partial covering problem is also well studied in theoretical computer science and the objects used to cover are a square [9, 10], a rectangle [11–13], and a circle [14]. In this paper we consider the partial covering problem where the covering object is an axis-parallel

P.R.S. Mahapatra (✉)
Department of Computer Science and Engineering, University of Kalyani, Kalyani, India
e-mail: priya@klyuniv.ac.in

J.K. Mandal et al. (eds.), *Information Systems Design and Intelligent Applications*,
Advances in Intelligent Systems and Computing 339,
DOI 10.1007/978-81-322-2250-7_33

square. A square is said to be axis-parallel if each of it's side is parallel to one of the coordinate axis. This problem is well studied in computational geometry with many variations [15–17]. Interested reader may read these papers [18–21] and the references therein for getting further variation of the partial covering problem. Another motivation of considering this problem comes from classification when the objective is to find a cluster of given shape that contains at least k points [22–24].

2 Preliminaries

The space of all candidate (or potential) solutions of a problem is called the *solution space*, or *search space*. There are some optimization problems for which these solution spaces are known. A well known such optimization problem is the *graph coloring problem* where the task is to find the minimum number of colors to color a graph, say G, and this number is called the *chromatic number* of G. Observe that the integral value of the chromatic number will very between 1 and n where n is the number of vertices of the graph G. This observation implies that the solution space for the graph coloring problem is known. A general approach to solve this problem is as follows. First solve the decision version of such problem and then use the solution of this decision problem to solve the original problem. Observe that the decision version of the graph coloring problem can be expressed as "*is the graph G is k colorable?*". The answer is always "Yes" or "No". In this case the decision problem is NP-hard [25] and therefore it implies that the original graph coloring problem is also NP-hard.

3 Solution for General *k*

In this work we first consider the optimization problem of finding an axis-parallel square for general values of k that covers at least k points of S and the area (perimeter) is minimum. From now onwards a square means an axis-parallel square. Let *size* denote the length of a square. A square S_k covering k points of S is said to be k-cover square if there does not exist another square having area less than that of S_k and covering k points from P [16]. We have the following result on the characteristics of a k-cover square.

Observation 1 Reference [17] *At least one pair of opposite sides of a k-cover square must contain points from P.*

This result implies that size of a k-cover square be determined from the set of horizontal and vertical distances generated from the pair of points in S. Therefore the solution space for this optimization problem is known. The decision version of this problem can be defined as "*given a length δ, does there exist a square of size δ*

that covers at least k points of S?". Let $MaxCover(\delta)$ denote the square of size δ that covers the maximum number of points of S. Note that the decision problem of the original problem is same with the problem of finding a location of $MaxCover(\delta)$. The following result in [18] is used to locate $MaxCover(\delta)$.

Result 1 Reference [18] *Given the length* δ, *a location of* $MaxCover(\delta)$ *can be computed in* $O(n \log n)$ *time using* $O(n)$ *space.*

Let $S = \{p_1, p_2, \ldots, p_n\}$ be the set of points on the plane. Without loss of generality let these points of S be in non-decreasing order on x- coordinate. Here we use $(x(p), y(p))$ to denote a point of S. Note that the result in Observation 1 implies each of two horizontal sides (or two vertical sides) of a $MaxCover(\delta)$ must contains a point of S. Thus two types of solutions are required to consider to solve this problem. As these two types are symmetric, the techniques required to find of any one type can be easily extended for other. Therefore, without loss of generality, we assume that each horizontal side of the $MaxCover(\delta)$ is containing a point of S. The solution space of the original problem can be viewed as matrix M given below.

$$\begin{pmatrix}
x(p_2)-x(p_1) & x(p_3)-x(p_1) & \ldots & x(p_{n-1})-x(p_1) & x(p_n)-x(p_1) \\
x(p_3)-x(p_2) & x(p_4)-x(p_2) & \ldots & x(p_n)-x(p_2) & \\
\ldots & \ldots & \ldots & & \\
\ldots & \ldots & \ldots & & \\
x(p_{n-1})-x(p_{n-2}) & x(p_n)-x(p_{n-2}) & & & \\
x(p_n)-x(p_{n-1}) & & & &
\end{pmatrix}$$

We now conclude that the solution space of the problem can be stored in a lower (upper) triangular matrix of order $(n-1)\times(n-1)$. The first row of the matrix M contains $(n-1)$ horizontal distances, second row of M contains $(n-2)$ horizontal distances, and so on. This generalization implies that the ith row of M has $(n-i)$ horizontal distances for $i = 1, 2, \ldots, (n-1)$. We now compute the number of input points covered by the $MaxCover(\delta)$ using Result 1 for each value of $\delta(\in M)$. Moreover we store the current minimum value of δ for which the square $MaxCover(\delta)$ covers at least k points of S. Note that Result 1 can be used to find a placement of $MaxCover(\delta)$ in $O(n \log n)$ time and the value of δ is one among $O(n^2)$ values of matrix M. Thus we can derive the following straight forward result.

Result 2 *Let S be the set of n input points on the plane and* $k(\leq n)$ *be an integer constant. A smallest axis-parallel square covering at least k input points can be found in* $O(n^3 \log n)$ *time and* $O(n)$ *space.*

4 Improvement Using Matrix Search for $k > \frac{n}{2}$

We now improve Result 2 for large values of k $\left(k > \frac{n}{2}\right)$. Let S_1, S_2, S_3, S_4 and S_5 and S_5 be a partition of S such that $S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$ and all the partitions are not required to be pairwise disjoint. S_1 and S_2 are the $(n-k)$ left most and right most points of S respectively. S_3 and S_4 are the $(n-k)$ bottom most and top most points of S respectively. Moreover $S_5 = S - S'$ where $S' = S_1 \cup S_2 \cup S_3 \cup S_4$. Let the minimum enclosing rectangle (MER) R contain all points of S_5. The boundaries of R is closed in the sense that each side of R contains at least one point of S. The following result in [17] is used for $k > \frac{n}{2}$.

Observation 2 Reference [17] *For $k > n/2$, S_k must covers all the points of R.*

The length of the largest side of the minimum enclosing rectangle R is denoted by Δ. The following result is extended in [17] to compute a placement of $MaxCover(\delta)$ when $\delta > \Delta$.

Result 3 Reference [17] *A placement of $MaxCover(\delta)$ for a given $\delta > \Delta$ can be found in $O((n-k)\log(n-k))$ time using $O(n)$ space.*

Define Q be the subset of S such that Q is the points of S' and the points on the boundary of the minimum enclosing rectangle R. Let $= \{q_1, q_2, \ldots, q_r\}$ denote the non decreasing order of the points in Q on x-coordinate. Observe that the number of points in Q is r and r is of $O(n-k)$. Moreover the value of r is at most $4(n-k)$. We use q_α and q_β to denote the points of S those lie on the right and left side of the minimum enclosing rectangle R respectively. The results in Observation 2 and Observation 1 imply that the solution space for finding S_k for $k > \frac{n}{2}$ is now reduced from the above matrix M to the following matrix N.

$$\begin{pmatrix} x(q_\alpha) - x(q_1) & x(q_{\alpha+1}) - x(q_1) & \ldots & x(q_r) - x(q_1) \\ x(q_\alpha) - x(q_2) & x(q_{\alpha+1}) - x(q_2) & \ldots & x(q_r) - x(q_2) \\ \ldots & \ldots & \ldots & \\ x(q_\alpha) - x(q_i) & x(q_{\alpha+1}) - x(q_i) & \ldots & x(q_r) - x(q_i) \\ \ldots & \ldots & \ldots & \\ x(q_\alpha) - x(q_\beta) & x(q_{\alpha+1}) - x(q_\beta) & \ldots & x(q_r) - x(q_\beta) \end{pmatrix}$$

Note that the number of elements in the matrix N is $O((n-k)^2)$. We can now use Result 3 for each entry of N. Thus the following result can be found like earlier.

Result 4 *Let S be the set of n input points on the plane and $k(> \frac{n}{2})$ be an integer constant. A smallest axis-parallel square covering at least k points of S can be found in $O((n-k)^3 \log(n-k))$ time using $O(n)$ space.*

It is now shown that the standard *sorted matrices search* by Frederickson and Johnson [26] can be used to improve Result 4. Sorted matrices search is basically a prune and search technique. The technique has been demonstrated in the last decade in various works dealing with (not only) covering problems and facility location [27]. The idea is to define a decision problem of the original optimization problem and then perform a kind of binary search in order to determine the optimal (in this case, smallest area) value. Recall that in our case, the objective of the decision version of the problem is to find a placement of the square $MaxCover(\delta)$ where $\delta \in N$. Note that elements in each row are in non-increasing order. The same ordering is also true for elements in each column of N. This implies that N is a sorted matrix [26, 28]. Using our decision problem we can make a kind of binary search using the sorted matrix N obtaining running time of $O(T_d * \log(n-k) + n)$, where T_d is the running time of the decision algorithm. Here the task of the decision algorithm is to find a placement of the square $MaxCover(\delta)$ for $\delta > \Delta$. It follows from Result 3 that $T_d = O((n-k)\log(n-k))$. In a similar pass we can find another potential solution keeping time and space complexities unchanged. Thus we have the following result.

Result 5 *Let S be the set of n input points on the plane and $k(>\frac{n}{2})$ be an integer constant. A smallest area square covering at least k points of S can be found in $O(n + (n-k)\log^2(n-k))$ time using $O(n)$ space.*

5 Conclusion

An $O(n + (n-k)\log^2(n-k))$ time algorithm is proposed to identify a smallest axis-parallel square that covers at least k points among a set of n points. It is shown that the search space of the decision problem of the original optimization problem is a sorted matrix when $k > \frac{n}{2}$. However, the solution space of the decision problem is not a sorted matrix for general k. This observation mainly forces the authors in [18] to use another prune and search technique other than matrix search to solve the optimization problem for general k and their solution requires $O(n\log^2 n)$ time and $O(n)$ space. It would be interesting to investigate the possibility of finding an efficient search technique that can reduce the complexity of the optimization problem for general k other than the method used in [18].

References

1. Megiddo, N.: Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. SIAM J. Comput. **12**, 759–776 (1995)
2. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. Springer, Berlin (1990)
3. Toussaint, G.T.: Solving geometric problems with the rotating calipers. In: Proceedings of IEEE MELECON (1983)
4. Katz, M.J., Kedem, K., Segal, M.: Discrete rectilinear 2-center problems. Comput. Geom. **15**, 203–214 (2000)

5. O'Rourke, J., Aggarwal, A., Maddila, S., Baldwin, M.: An optimal algorithm for finding minimal enclosing triangles. J. Algorithms **7**, 258–269 (1986)
6. Agarwal, P.K., Sharir, M., Toledo, S.: Applications of parametric searching in geometric optimization. J. Algorithms **17**, 292–318 (1994)
7. Mukherjee, J., Mahapatra, P.R.S., Karmakar, A., Das, S.: Minimum-width rectangular annulus. Theoret. Comput. Sci. **508**, 74–80 (2013)
8. Gluchshenko, O., Hamacher, H.W., Tamir, A.: An optimal $O(n \log n)$ algorithm for finding an enclosing planar rectilinear annulus of minimum width. Oper. Res. Lett. **37**, 168–170 (2009)
9. Datta, A., Lenhof, H.E., Schwarz, C., Smid, M.: Static and dynamic algorithms for k-point clustering problems. J. Algorithms **19**, 474–503 (1995)
10. Smid, M.: Finding k points with a smallest enclosing square. Report MPI-92-152, Max-Plank-Institute fur Informatik, Saarbriucken (1992)
11. Aggarwal, A., Imai, H., Katoh, N., Suri, S.: Finding k points with minimum diameter and related problems. J. Algorithms **12**, 38–56 (1991)
12. Ahn, H.K., Bae, S.W., Demaine, E.D., Demaine, M.L., Kim, S.S., Korman, M., Reinbacher, I., Son, W.: Covering points by disjoint boxes with outliers. Comput. Geom. Theory Appl. **44**, 178–190 (2011)
13. Segal, M., Kedem, K.: Enclosing k points in the smallest axis parallel rectangle. Inf. Process. Lett. **65**, 95–99 (1998)
14. Matoušek, J.: On geometric optimization with few violated constraints. Discrete Comput. Geom. **14**, 365–384 (1995)
15. Chan, T.M.: Geometric applications of a randomized optimization technique. Discrete Comput. Geom. **22**, 547–567 (1999)
16. Das, S., Goswami, P.P., Nandy, S.C.: Smallest k-point enclosing rectangle and square of arbitrary orientation. Inf. Process. Lett. **94**, 259–266 (2005)
17. Mahapatra, P.R.S., Karmakar, A., Das, S., Goswami, P.P.: k-enclosing axis-parallel square. LNCS 6784, pp. 84–93. Springer, New York (2011)
18. Mahapatra, P.R.S., Goswami, P.P., Das, S.: Covering points by isothetic unit squares. In: Proceeding of the 19th Canadian Conference on Computational Geometry (CCCG 2007), pp. 169–172 (2007)
19. Mahapatra, P.R.S., Goswami, P.P., Das, S.: Maximal covering by two isothetic unit squares. In: Proceeding of the 20th Canadian Conference on Computational Geometry (CCCG 2008), pp. 103–106 (2008)
20. Atta, S., Mahapatra, P.R.S.: Genetic Algorithm Based Approach for Serving Maximum Number of Customers Using Limited Resources, vol. 10, pp. 492–497. Elsevier, Amsterdam (2013)
21. Atta, S., Mahapatra, P.R.S.: Genetic Algorithm Based Approaches to Install Different Types of Facilities, vol. 248, pp. 195–203, AISC (Springer) (2014)
22. Andrews, H.C.: Introduction to Mathematical Techniques in Pattern Recognition. Wiley-Intersciences, New York (1972)
23. Asano, T., Bhattacharya, B., Keil, M., Yao, F.: Clustering algorithms based on maximum and minimum spanning trees. In: Proceeding of the 4th Annual Symposium on Computational Geometry, pp. 252–257 (1988)
24. Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)
25. Cook, S.: The complexity of theorem proving procedures. In: Proceeding of the 3rd Annual ACM Symposium on Theory of Computing, pp. 151–158 (1971)
26. Frederickson, G.N., Johnson, D.B.: Generalized selection and ranking: sorted matrices. SIAM J. Comput. **13**, 14–30 (1984)
27. Drezner, Z., Hamacher, H.: Facility Location: Applications and Theory. Springer, Berlin (2002)
28. Glozman, A., Kedem, K., Shpitalnik, G.: On some geometric selection and optimization problems via sorted matrices. Comput. Geom. Theory Appl. **11**, 17–28 (1998)