Doctopic: Algorithms, automata, complexity and games ARTICLE IN PRESS TCS:10532

Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Optimizing squares covering a set of points

Sergey Bereg [a], Binay Bhattacharya [b], Sandip Das [c], Tsunehiko Kameda [d,*], Priya Ranjan Sinha Mahapatra [e], Zhao Song [f]

[a] *Department of Computer Science, Univ. of Texas at Dallas, Dallas, USA*
[b] *School of Computing Science, Simon Fraser University, Burnaby, Canada*
[c] *Advanced Computing and Microelectronics Unit, Indian Stat. Inst., Kolkata, India*
[d] *School of Computing Science, Simon Fraser University, Burnaby, Canada*
[e] *Department of Computer Science and Engineering, Univ. of Kalyani, Kalyani, India*
[f] *Department of Computer Science, Univ. of Texas at Austin, Austin, USA*

ARTICLE INFO

Article history:
Received 19 May 2015
Received in revised form 27 October 2015
Accepted 20 November 2015
Available online xxxx

Keywords:
Facility location
Point covering by squares
k-Center problem

ABSTRACT

We investigate three kinds of optimization problems regarding n points in the 2-dimensional plane that need to be enclosed by squares. (1) Find a given number of squares that enclose all the points, minimizing the size of the largest square used. (2) Problem (1) with the additional condition that the center of each enclosing square must lie on one of the two given axis-parallel lines, which are either parallel or perpendicular. (3) Enclose the maximum number of points, using a specified number of squares of a fixed size.
We propose different techniques to solve the above problems in cases where squares are axis-parallel or of arbitrary orientation, disjoint or overlapping. All the algorithms we use run in time that is a low-order polynomial in n, and improve upon the previous algorithms, if any.

1. Introduction

1.1. Problem definition and related work

Given a set $\mathcal{P}$ of n points, the enclosing problem in computational geometry is to find the smallest geometrical object of a given type that encloses all the points in $\mathcal{P}$. There are the problems of finding the minimum enclosing circle [30], the minimum area triangle [5,21,28], the minimum area rectangle [38], the minimum bounding box [27], the smallest ellipsoid [40], and the smallest width annulus [2]. As far as we are aware, there has been little work on finding the smallest enclosing square(s) with arbitrary orientation. Das et al. [9] presented an algorithm to identify the smallest square of arbitrary orientation, containing exactly k points in $O(n^2 \log n + kn(n-k)^2 \log n)$ time. It runs in $O(n^2 \log n)$ time when $n = k$.

Katz et al. [19] consider the problem of covering the points in $\mathcal{P}$ by two "constrained" squares whose centers must lie on some specified points. The two squares may or may not be disjoint. They presented an $O(n \log^2 n)$ time algorithm to find two constrained axis-parallel squares whose union covers $\mathcal{P}$ and the size of the larger square is minimized. They also presented an $O(n^2 \log^4 n)$ time algorithm to find two constrained parallel squares (with arbitrary orientation) whose union covers $\mathcal{P}$ and the size of the larger square is minimized. Jaromczyk and Kowaluk [18] solved the unconstrained

* Corresponding author.
E-mail address: tiko@sfu.ca (T. Kameda).

http://dx.doi.org/10.1016/j.tcs.2015.11.029
0304-3975/© 2015 Published by Elsevier B.V.

Table 1
Minimizing the size of the largest square. AP = "axis-parallel," AO = "arbitrary orientation," D = "disjoint," ND = "non-disjoint" (overlapping), and DC = "don't care" (the squares involved are not constrained to be D or ND).

Num.	D/ND/DC	AP/AO	Previous best	Our results	Section
1	–	AO	$O(n^2 \log n)$ [9]	$O(n \log n)$	2.1
1	–	AO[a]	–	$O(n \log n)$	2.2
2	D	AP	$O(n \log n)$ [18]	$O(n)$	2.3
2	ND	AP	–	$O(n)$	2.4
2	DC	AP	$O(n)$ [35]	$O(n)$[b]	–
2	D	AP–AO	$O(n^2 \log n)$ [20]	–	2.5
2	DC	AP–AO	–	$O(n^3 \log n)$	2.6
2	DC	AO	–	$O(n^4 \log n)$	2.7
3	D	AP	–	$O(n \log n)$	2.8
4	D	AP	–	$O(n^2 \log^2 n)$	2.10

[a] With a rectangular obstacle.
[b] Implied by the two previous results.

version of the above problem in $O(n^2)$ time. They also presented an $O(n^3 \log^2 n)$ time algorithm to find two constrained squares (where each square is allowed to rotate independently) whose union covers $\mathcal{P}$ and the size of the larger square is minimized.

If the "constraints" are given in the form of a line, this problem is equivalent to the unweighted 1-dimensional k-center problem, which was solved by Frederickson in linear time [11]. In other words, if the points are presorted by their x-coordinates, then the unweighted line-constrained k-center problem can be solved in $O(n)$ time. A recent paper by Wang and Zhang [39] also discusses this problem, citing Fournier and Vigneron's result [10].

Saha and Das [32] proposed an algorithm to find two parallel rectangles with arbitrary orientation, which may or may not be disjoint, whose union covers $\mathcal{P}$ and the size of the larger rectangle is minimized. Their algorithm runs in $O(n^3)$ time and $O(n^2)$ space. Kim et al. [20] considered two variants of the optimization problem for disjoint *rectangles*: (1) the rectangles are allowed to be oriented freely while being restricted to be parallel to each other, and (2) one rectangle is restricted to be axis-parallel but the other rectangle is allowed to be oriented freely. For both of the problems, they presented $O(n^2 \log n)$ time algorithm using $O(n)$ space. Segal [34] discusses lower bounds on some covering problems by squares, and, in particular, shows that it takes $\Omega(n \log n)$ time to find the smallest square with a given center that encloses all the points.

Researchers have also considered the problem of maximizing the number of points in $\mathcal{P}$ that can be enclosed by different shapes. Overmars and Yap [29] presented an $O(n \log n)$ time algorithm to solve this problem for a rectangle of fixed length and width. Their $O(n \log n)$ time algorithm also works for an axis-parallel square of fixed size. Sinha Mahapatra et al. proposed an $O(n^2)$ time and space algorithm for computing two axis-parallel unit squares which may be either disjoint or overlapping such that they together cover the maximum number of points [36]. In case the two such squares intersect, the interior of their intersection is not allowed to contain any point of $\mathcal{P}$. They designed an $O(n^2 \log^2 n)$ time and $O(n \log n)$ space algorithm to find two overlapping axis-parallel unit squares such that they together cover the maximum number of points [37]. They introduced an $O(k^2 n^5)$ time and $O(kn^4)$ space algorithm to find k disjoint axis-parallel unit squares, so as to maximize the number of points covered by them [36].

Throughout the paper, we always use the l_∞ distance metric, or equivalently the l_1 distance metric. Other researchers have considered the Euclidian or l_2 distance metric (namely, covering by circles), e.g., [7,16,17,24,31,39]. Hurtado et al. used a convex polygon, as well as a line, to constrain the center of the enclosing circle [16,17]. Chazelle and Lee [7] used a fixed radius circle to cover the maximum number of points from $\mathcal{P}$.

1.2. Our contributions

We assume that no three points in $\mathcal{P}$ lie on the same line. The objective of the first group of problems that we solve is to minimize the size of the largest square used. Table 1 summarizes the time complexities of our algorithms presented in this paper, and compares them with the previously best results. The first column indicates the number of squares used. Some remarks are in order regarding Table 1. Sharir and Welzl introduced the concept of the *LP-type* problem [35] and applied it to solve some geometric problems in linear time. For example, when $k = 3$, they showed that the enclosing problem, considered as the rectilinear 3-center problem, is of LP-type, thus can be solved in linear time using the algorithms of Matoušek et al. [23] and Chazelle and Matoušek [6]. However, our result in Section 2.8, shown in Table 1, has the condition that the enclosing squares be disjoint (D), while Sharir and Welzl [35] allow the enclosing squares to intersect (DC). In the DC case, Nussbaum [26] and Segal [33] present $O(n \log n)$ time algorithms for $k = 4$ and $k = 5$. We also note that the problems discussed in Secs. 2.1 and 2.7, which involve squares of arbitrary orientation, are not of LP-type.

We now impose the constraint that the center of each enclosing square must lie on one of the two given axis-parallel or perpendicular lines. We show that the k min-sized enclosing squares under this condition can be found in $O(n \log n)$ time, independently of k.