Theoretical Computer Science 634 (2016) 120–126

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Efficiently approximating color-spanning balls

Payam Khanteimouri [a], Ali Mohades [a,*], Mohammad Ali Abam [b], Mohammad Reza Kazemi [a]

[a] *Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran*
[b] *Sharif University of Technology, Tehran, Iran*

ARTICLE INFO

Article history:
Received 11 October 2015
Received in revised form 13 February 2016
Accepted 18 April 2016
Available online 10 May 2016
Communicated by T. Erlebach

Keywords:
Approximation algorithms
Color-spanning objects
Complexity

ABSTRACT

Suppose n colored points with k colors in $\mathbb{R}^d$ are given. The Smallest Color-Spanning Ball (SCSB) is the smallest ball containing at least one point of each color. As the computation of the SCSB in L_p metric ($p \geq 1$) is time-consuming, we focus on approximately computing the SCSB in near-linear time. Initially, we propose a 3-approximation algorithm running in $O(n \log n)$ time. This algorithm is then utilized to present a $(1+\varepsilon)$-approximation algorithm with the running time of $O((\frac{1}{\varepsilon})^d n \log n)$. We improve the running time to $O((\frac{1}{\varepsilon})^d n)$ using randomized techniques. Afterward, spanning colors with two balls is studied. For a special case where $d = 1$, there is an algorithm with $O(n^2)$ time. We demonstrate that for any $\varepsilon > 0$ under the assumption that SETH is true, no approximation algorithm running in $O(n^{2-\varepsilon})$ time exists for the problem even in one-dimensional space. Nevertheless, we consider the L_∞ metric where a ball is an axis-parallel hypercube and present a $(1+\varepsilon)$-approximation algorithm running in $O\big((\frac{1}{\varepsilon})^{2d}(\frac{n^2}{k})\log^2 n\big)$ time which is remarkable when k is large. This time can be reduced to $O\big((\frac{1}{\varepsilon})\frac{n^2}{k}\log n\big)$ when $d = 1$.

1. Introduction

Background. In computational geometry, most of the time we assume that the points are exact, meaning the exact locations of points are known. But in practice, location of input points is imprecise due to several reasons such as device errors, network latency, etc. Thus, each *imprecise point* is defined by a continuous or discrete range which specifies all the possible locations [15]. In case of an imprecise point that is modeled with a discrete range such as a point set, the location of that point is obtained from any point within its defining set. Therefore, by assigning a distinct color to the points included in each defining set, a set of k imprecise points can be modeled by a set of n points with k colors, where n is the total number of all possible locations. In this framework, one usually asks for choosing one point from each color to minimize or maximize a geometric function such as enclosing ball, closest pair, diameter or the area of convex hull. Similar problems may appear in other areas like facility location, statistical clustering, pattern recognition and generalized range searching [1,7,17].

Related works. To summarize, n points with k colors in $\mathbb{R}^d$ are given; the goal is to select a point from each color such that a geometric function is minimized or maximized. The *Smallest Color-Spanning Ball* (SCSB) is one of the several functions that has been studied recently. For the case of $d = 2$, the SCSB can be computed in $O(kn \log n)$ time using the upper envelope of

* Corresponding author.
E-mail address: mohades@aut.ac.ir (A. Mohades).

http://dx.doi.org/10.1016/j.tcs.2016.04.022
0304-3975/© 2016 Elsevier B.V. All rights reserved.

Voronoi surfaces [8]. Moreover, Khanteimouri et al. [12] considered the problem in L_∞ metric and presented an $O(n\log^2 n)$ time algorithm to solve the problem.

As a generalization, the *2-center problem*[1] is naturally studied in the context of colored points (the so-called *chromatic 2-center problem*). Khanteimouri et al. [13] considered the simplest case where the points are on $\mathbb{R}^1$. The problem is to compute two smallest intervals with equal length in such a way that together, they span all colors. They presented an $O(n^2\log n)$ time algorithm which has been recently improved to $O(n^2)$ time by Jiang and Wang [11]. Moreover, Jiang and Wang showed that approximating *the smallest color-spanning t-intervals* (similarly, t smallest intervals spanning all colors together) is W[2]-hard when t is a parameter—see [11] for more related results.

Our results. In Section 2, we focus on spanning colors with one ball (SCSB problem) for a given set of n points with k colors in $\mathbb{R}^d$ where d is fixed. Since computing the SCSB is time-consuming, we donate our attention to approximate it. We start by proposing an $O(n\log n)$-time algorithm to approximately compute the SCSB up to a factor of 3. Then, we use it to present a $(1+\varepsilon)$-approximation algorithm running in $O\big((\frac{1}{\varepsilon})^d n\log n\big)$ time. Moreover, we apply the sieve technique [14] and present a randomized $(1+\varepsilon)$-approximation algorithm to improve the running time to $O\big((\frac{1}{\varepsilon})^d n\big)$ expected time.

Thereafter, we consider the problem of spanning colors with two balls in Section 3. As a lower bound, we show that under the SETH assumption for any constant $\varepsilon > 0$, no approximation algorithm running in $O(n^{2-\varepsilon})$ time exists. This lower bound is held even for the case of $d = 1$ where a ball is an interval. Finally, we consider the L_∞ metric and propose a $(1+\varepsilon)$-approximation algorithm in $O\big((\frac{1}{\varepsilon})^{2d}(\frac{n^2}{k})\log^2 n\big)$ time which is remarkable when k is large. When $d = 1$, the problem can be solved in $O\big((\frac{1}{\varepsilon})\frac{n^2}{k}\log n\big)$ time.

2. Spanning colors with one ball

To our best knowledge, the best known algorithm computing SCSB for n given colored points in $\mathbb{R}^d$ runs in $O(n^{d+2})$ time using a brute-force method which of course is time-consuming—we recall that when $d = 2$, the SCSB can be computed faster. Therefore, approximating SCSB is worth to be considered. In this section, we focus our attention to approximate the SCSB in L_2 metric. Nevertheless, our results are generally extendible to L_p metric for any $p \geq 1$.

2.1. Deterministic algorithms

First, we give a constant factor approximation for the SCSB problem and then, we apply it to obtain a $(1+\varepsilon)$-approximation algorithm. We start with some definitions.

Definitions. Let $\mathcal{P}$ be the given set of n points with k colors in $\mathbb{R}^d$. We know, by the pigeonhole principle, there is a color c with at most n/k occurrences. Let $\mathcal{P}^c$ be the set of points that are colored by color c. We denote the smallest color-spanning ball of $\mathcal{P}$ by B^* and its radius by r^*. In addition, let $B(p)$ be the smallest color spanning ball centered at a point p. Finally, $H(p,l)$ is the d-dimensional axis-parallel hypercube centered at a point p whose side length is l.

3-approximation. As the main ingredient of our 3-approximation algorithm, we use the data structure given by Arya et al. [2] that reports an approximate nearest neighbour (ANN) in an input set for a query point. Their data structure is a balanced compressed quadtree which reports a point whose distance to the query point is not more than $(1+\varepsilon)$ times the distance of the exact nearest neighbour in $O(\frac{1}{\varepsilon^d}+\log n)$ time. We first construct ANN data structures for each color separately in the total time of $O(n\log n)$. Using these ANN data structures, one can compute the $(1+\varepsilon)$-approximation of the smallest color-spanning ball with the known center $p \in \mathbb{R}^d$ in $O\big(k(\frac{1}{\varepsilon^d}+\log n)\big)$ time for some constant $\varepsilon > 0$. B^* contains at least a point $p \in \mathcal{P}^c$ as it must contain every color. Furthermore, for any point p inside B^*, $B(p)$ can not have radius greater than $2r^*$. This guides us to take the minimum over balls $B(p)$ for all $p \in \mathcal{P}^c$. Since computing the approximation is faster, we compute the $(1+\varepsilon)$-approximation of $B(p)$ for all $p \in \mathcal{P}^c$ and then we take the minimum. These together can be done in $O\big(\frac{n}{k}\times k(\frac{1}{\varepsilon^d}+\log n)\big) = O\big(n(\frac{1}{\varepsilon^d}+\log n)\big)$ time. Therefore, we can compute a $(2+\varepsilon)$-approximation of B^* in $O\big(n(\frac{1}{\varepsilon^d}+\log n)\big)$ time. Setting ε to 1 gives us a 3-approximation algorithm with the running time of $O(n\log n)$. The following theorem concludes the above.

Theorem 1. *For a set $\mathcal{P}$ of n points in $\mathbb{R}^d$ where each point is colored with one of k colors, a 3-approximation of B^* can be computed in $O(n\log n)$ time.*

$(1+\varepsilon)$-approximation. To present a $(1+\varepsilon)$-approximation of B^* for any constant $\varepsilon > 0$, we first use the above algorithm to compute the radius $r \leq 3r^*$. Then, we consider a grid Z over $\mathcal{P}$ and round all points of $\mathcal{P}$ to grid points in the following way. Each cell of the grid is incident to 2^d grid points. For a point $p \in \mathcal{P}$, we denote by $g(p)$ the grid point incident to the cell containing p whose coordinates in any dimension are the lowest ones. We set the grid-cell side length of

[1] The standard 2-center problem is to compute two smallest disks of equal radii such that they cover all the given points in the plane together. It has been widely studied under L_1, L_∞, L_2 metrics—see the main results in [4,5,19].