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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Planar maximum-box problem revisited

Farnaz Sheikhi, Ali Mohades *

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

ARTICLE INFO

Article history:
Received 10 February 2016
Received in revised form 31 March 2017
Accepted 27 December 2017
Available online xxxx
Communicated by M.J. Golin

Keywords:
Discrete optimization
Computational geometry
Plane sweep algorithms

ABSTRACT

Let B be a set of b blue points and R be a set of r red points in the plane. In this paper we study the problem of finding rectangles that contain the maximum number of blue points without containing any red points, known as the *maximum-box problem*. First we study this problem for axis-aligned rectangles, and propose an exact worst-case optimal $O(r^2 + rb + b\log b)$ time algorithm using $O(r + b)$ space to find all maximum boxes. We also provide a 2-approximation algorithm running in $O((r + b)\log(r + b))$ time and using $O(r + b)$ space to find a single maximum box in the axis-aligned case. Then we generalize the exact algorithm for the axis-aligned case to find all arbitrarily oriented maximum boxes leading to a worst-case optimal $O((r + b)^2(r + \log b))$ time algorithm using $O((r + b)^2)$ space to solve the problem. We conclude the paper by discussing time and space trade-offs. Our results improve the previously best known solutions to the maximum-box problem.

1. Introduction

Given a set B of b blue points and a set R of r red points, in the maximum-box problem the goal is to find a rectangular box $\mathcal{BX}$ so that $\mathcal{BX} \cap R = \emptyset$ and the cardinality of $\mathcal{BX} \cap B$ is maximized. Each box $\mathcal{BX}$ satisfying this condition can be enlarged so that its sides touch red points or reach the boundary of the bounding box of $B \cup R$. Hence, the maximum-box problem can be seen as the problem of finding rectangles that contain the maximum number of blue points, and can contain red points only on the boundary. From now on, we call such rectangles *maximum blue rectangles* or MBRs for short. The maximum-box problem finds applications in data analysis [8], where the goal is to find patterns that intersect exactly one of the given two sets. A study of criteria for selecting the most suitable patterns for classification of data using logical analysis has shown that the best results are achieved by maximal boxes [9]. This problem has also attracted attentions from the view point of computer graphics [7]. The maximum box problem has been studied for axis-aligned and arbitrarily oriented rectangles.

For axis-aligned rectangles, the maximum-box problem was first studied by Eckstein et al. [8] for arbitrary dimension d. They proved that the problem is NP-hard when d is a part of input. For the maximum-box problem in the plane, which is the topic of our paper, the following results are known. Segal [11] gave an $O(b^2(b + r)(\log^4 b + \log^3 r \log b))$ time algorithm, using $O(r + b)$ space, to find the smallest area MBR. Later, Liu and Nediak [10] proposed an $O(r^2 \log r + rb + b\log b)$ time algorithm, using $O(r + b)$ space, to compute all MBRs. They also provided a 2-approximation algorithm (the number of blue points is approximated) running in $O((r + b)\log^2(r + b))$ time and using $O(r + b)$ space to find an MBR. Backer and

* Corresponding author.
E-mail addresses: f.sheikhi@aut.ac.ir (F. Sheikhi), mohades@aut.ac.ir (A. Mohades).

https://doi.org/10.1016/j.tcs.2017.12.038
0304-3975/

Please cite this article in press as: F. Sheikhi, A. Mohades, Planar maximum-box problem revisited, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.12.038

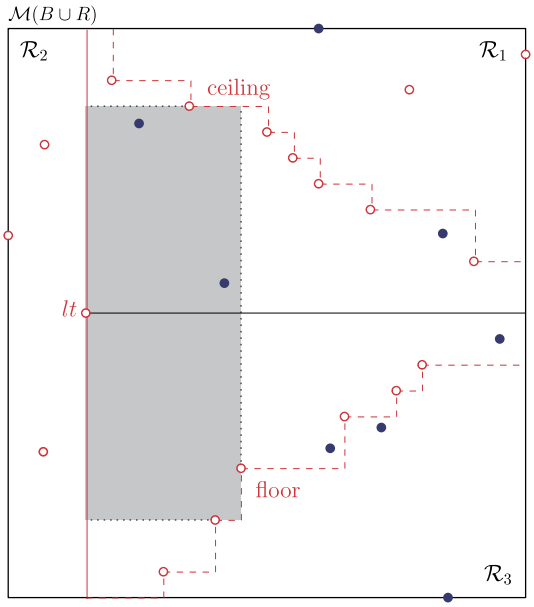

Fig. 1. Partitioning $\mathcal{M}(B \cup R)$ into three regions $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, and the corresponding staircases. The shaded region shows a $PMBR$ that has lt on the left side and is constructed by extension of a step of the floor. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Keil [1], presenting an $\Omega(r^2)$ lower bound on the number of axis-aligned MBRs, proposed an $O((r+b)\log^3(r+b))$ time algorithm, using $O((r+b)\log(r+b))$ space, to compute a single MBR. However, reducing the logarithmic gap either in the space complexity of the algorithm of Backer and Keil [1] and proposing a linear-space algorithm to compute an MBR, or in the time complexity of the algorithm of Liu and Nediak [10] and proposing a worst-case optimal quadratic time algorithm to compute all axis-aligned MBRs was left open. Recently, Barbay et al. [3] have proposed a divide-and-conquer algorithm that can find an axis-aligned MBR in $O((r+b)^2)$ time and $O(r+b)$ space.

For arbitrarily oriented rectangles, Bereg et al. [4] have recently proposed an $O((r+b)^2(r\log r + r\log b))$ time algorithm, using $O(r^2+rb)$ space, to compute all arbitrarily oriented MBRs in the plane. They have also provided a 2-approximation algorithm to find a single arbitrarily oriented MBR.

In this paper, first we study the maximum-box problem in the plane for axis-aligned rectangles. We propose a worst-case optimal $O(r^2+rb+b\log b)$ time algorithm, using $O(r+b)$ space, to compute all MBRs in this case, reducing the logarithmic gap that exists in the literature [10]. In comparison with the divide-and-conquer algorithm of Barbay et al. [3], our solution uses a simple plane sweep algorithm. Rather than computing a single MBR our algorithm finds all MBRs. It provides an improved approximation algorithm to find an axis-aligned MBR, and can be used as a basis to design an improved algorithm to compute all arbitrarily oriented MBRs. Hence, based on this algorithm we provide a 2-approximation algorithm running in $O((r+b)\log(r+b))$ time and using $O(r+b)$ space to find an axis-aligned MBR, improving the result of [10]. Moreover, we generalize our algorithm for the axis-aligned case to the arbitrarily oriented case, and propose an $O((r+b)^2(r+\log b))$ time algorithm, using $O((r+b)^2)$ space, to find all arbitrarily oriented MBRs, improving the result of [4]. We present a lower bound of $\Omega(r^3)$ on the number of MBRs in the arbitrarily oriented case, proving that the running time of the proposed algorithm is worst-case optimal. Furthermore, reducing the space complexity we propose an algorithm running in $O(r(r+b)^2\log(r+b))$ time, using $O(r+b)$ space, to compute all arbitrarily oriented MBRs. We conclude the paper by discussing time and space trade-offs.

2. Preliminaries

We start by defining some terminology and notation. Let $\mathcal{M}(B \cup R)$ denote the axis-aligned bounding box of $B \cup R$, and let x_p and y_p be the x- and y-coordinate of a point p, respectively. As explained earlier, each side of an MBR touches either a red point or the boundary of $\mathcal{M}(B \cup R)$. Hence, to compute MBRs first we describe how to compute MBRs that touch a red point on the left side. Then, computing MBRs that touch the boundary of $\mathcal{M}(B \cup R)$ on the left side can be done easily, as will be explained later. Thus, for now, for each red point we compute all potential MBRs (or $PMBR$s for short) that have the red point on their left side. Let lt denote this red point. To this aim we draw a vertical line and a horizontal half-line rightward through lt. This partitions $\mathcal{M}(B \cup R)$ into three regions: $\mathcal{R}_1$ (the upper right region), $\mathcal{R}_2$ (the left region), and $\mathcal{R}_3$ (the lower right region), as illustrated in Fig. 1.

We say that a red point $r^* \in \mathcal{R}_1$ is *upper maximal* if there does not exist another red point $r' \in \mathcal{R}_1$ such that $x_{r'} < x_{r^*}$ and $y_{r'} < y_{r^*}$. Connecting the upper maximal points in $\mathcal{R}_1$ we get a staircase which we call the *ceiling*. Further, we say that

Please cite this article in press as: F. Sheikhi, A. Mohades, Planar maximum-box problem revisited, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.12.038