ON THE EQUIVALENCE OF SOME RECTANGLE PROBLEMS

Herbert EDELSBRUNNER
Institut für Informationsverarbeitung, Technical University of Graz, A-8010 Graz, Austria

Mark H. OVERMARS *
Department of Computer Science, University of Utrecht, 3508 TA Utrecht, The Netherlands

Received 15 October 1981; revised version received 19 January 1982

Rectangle intersection searching, rectangle enclosure searching, rectangle containment searching, dominance searching, equivalence, computational geometry

1. Introduction

A *rectangle* in a d-dimensional space is the Cartesian product of one interval on each of the d coordinate-axes. Hence, a rectangle is assumed to have its sides parallel to the coordinate-axes. A rectangle R *encloses* a rectangle R′ if every point of R′ is also a point of R, R is *contained* in R′ if R′ encloses R and R *intersects* R′ if R and R′ have at least one point in common (see Fig. 1 for an example). The rectangle R is enclosed by R_1, contains R_2 and R_4 and intersects R_1, R_2, R_3, R_4 and R_5.

Given a set V of rectangles in a d-dimensional space and another such rectangle R, the *rectangle enclosure searching problem* asks for all rectangles in V that enclose R, the *rectangle containment searching problem* asks for all rectangles in V that are contained in R, and the *rectangle intersection searching problem* asks for all rectangles in V that intersect R. Often, we are merely interested in the number of rectangles that enclose, are contained in or intersect the given query rectangle R. In this case, we call the problems, respectively, the rectangle enclosure, containment and intersection *counting* problems.

* The work of the second author was supported by the Netherlands Organization for the Advancement of Pure Research (ZWO).

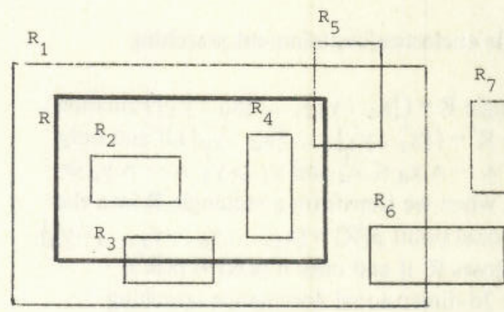

Fig. 1.

In particular the rectangle intersection searching (counting) problem has received considerable attention during the past few years (see e.g. [2,3,6,7,10]). The rectangle enclosure and containment searching (counting) problems were treated only recently by Lee and Wong [6] and McCreight [8]. Overmars and van Leeuwen [9] presented a general result by which some of the known solutions could be improved.

In this paper we will show that all three problems are in some sense equivalent to dominance searching: Given a set V of points in a d-dimensional space and another point $x = (x_1, ..., x_d)$, the *dominance searching problem* asks for all points $p = (p_1, ..., p_d)$ in V such that $p \leqslant x$, i.e., $p_1 \leqslant x_1 \wedge p_2 \leqslant x_2 \wedge \cdots \wedge p_d \leqslant x_d$. The counting variant of the problem that asks for the number of points in V that are dominated by x is the well-known ECDF-searching problem (where

0020-0190/82/0000–0000/$02.75

ECDF stands for Empirical Cumulative Distribution Function, see e.g. [1]).

In Section 2 we will show that the d-dimensional rectangle enclosure and containment searching (counting) problems are equivalent to the 2d-dimensional dominance searching (counting) problem. In Section 3 we show that the d-dimensional rectangle intersection counting problem is equivalent to the d-dimensional dominance counting problem. It follows that an improvement in the bounds for one of the problems immediately results in an improvement for the other problems. In Section 4 we briefly mention how the results can be generalized to other rectangle searching problems. This establishes an additional step towards a unified view of problems involving rectangles that were dealt with separately in the past.

2. Rectangle enclosure/containment searching

A rectangle $R = ([x_1 : y_1], ..., [x_d : y_d])$ encloses a rectangle $R' = ([x'_1 : y'_1], ..., [x'_d : y'_d])$ if and only if $x_1 \leqslant x'_1 \wedge \cdots \wedge x_d \leqslant x'_d$ and $y_1 \geqslant y'_1 \wedge \cdots \wedge y_d \geqslant y'_d$. Hence, when we transform a rectangle R into the 2d-dimensional point $p(R) = (x_1, ..., x_d, -y_1, ..., -y_d)$, then R encloses R′ if and only if $p(R) \leqslant p(R')$. Hence, the 2d-dimensional dominance searching problem can be used to solve the d-dimensional rectangle enclosure problem. Similarly, transforming the rectangle R into the point $p(R) = (-x_1, ..., -x_d, y_1, ..., y_d)$, R is contained in R′ if and only if $p(R) \leqslant p(R')$. Hence, also the d-dimensional rectangle containment searching problem can be solved using the 2d-dimensional dominance searching problem.

It remains to be shown that the 2d-dimensional dominance searching problem can be solved by the d-dimensional rectangle enclosure problem and by the containment problem as well. Let us first use the enclosure problem to solve the dominance problem. We want to map a given point $p = (x_1, ..., x_d, y_1, ..., y_d)$ into a rectangle $R(p) = ([f_1(x_1) : f_2(y_1)], [f_1(x_2) : f_2(y_2)], ..., [f_1(x_d) : f_2(y_d)])$ such that

(i) $f_1(x) \leqslant f_2(y)$ for all x, y,

(ii) $x \leqslant x' \Leftrightarrow f_1(x) \leqslant f_1(x')$, and $y \leqslant y' \Leftrightarrow f_2(y) \geqslant f_2(y')$.

(i) guarantees that points are mapped into rectangles and (ii) guarantees that p′ dominates p if and only if R(p) encloses R(p′). A possible solution to (i) and (ii) is

$$f_1(x) = \begin{cases} x - 2 & \text{if } x \leqslant 1, \\ -1/x & \text{if } x \geqslant 1, \end{cases}$$

$$f_2(y) = \begin{cases} -y + 2 & \text{if } y \leqslant 1, \\ 1/y & \text{if } y \geqslant 1. \end{cases}$$

To transform the 2d-dimensional dominance searching problem into the d-dimensional containment searching problem, we like to map a point $p = (x_1, ..., x_d, y_1, ..., y_d)$ into the rectangle $R(p) = ([f_1(x_1) : f_2(y_1)], ..., [f_1(x_d) : f_2(y_d)])$ under the following conditions:

(i) $f_1(x) \leqslant f_2(y)$ for all x, y;

(ii) $x \leqslant x' \Leftrightarrow f_1(x) \geqslant f_1(x')$, and $y \leqslant y' \Leftrightarrow f_2(y) \leqslant f_2(y')$.

One can choose, for instance,

$$f_1(x) = \begin{cases} 1/x & \text{if } x \leqslant -1, \\ -x - 2 & \text{if } x \geqslant -1, \end{cases}$$

$$f_2(y) = \begin{cases} -1/y & \text{if } y \leqslant -1, \\ y + 2 & \text{if } y \geqslant -1. \end{cases}$$

The transformations clearly hold for the counting variants also.

Theorem 2.1. The d-dimensional rectangle enclosure and containment searching/counting problems are both equivalent to the 2d-dimensional dominance searching/counting problem.

3. Rectangle intersection counting

The d-dimensional dominance searching problem is a special case of the d-dimensional rectangle intersection searching problem in which the rectangles in the set are degenerated to points and the query rectangle is $([-\infty, x_1], ..., [-\infty, x_d])$. Hence it only needs to be shown that d-dimensional rectangle intersection can be solved using the d-dimensional dominance problem. We are only able to show this for the counting variants.

Lemma 3.1. The d-dimensional rectangle intersection counting problem can be solved using the d-dimensional dominance counting problem.

Proof. Let us first consider the 1-dimensional case. So we are given a set V of intervals. To determine the number of intervals that intersect a query interval [a : b], we count the number, n_0, of intervals that do not intersect [a : b]. Clearly, the number of intersecting intervals is $n - n_0$, where n is the total number of intervals in the set. Note that an interval [x : y] in the set does not intersect [a : b] if either $x > b$ or $y < a$. To determine the number n_1 of intervals [x : y] in the set with $x > b$ we perform a dominance counting query with $-b$ on the set V_1 that contains $-x$ for each begin point x of a segment in V, and to determine the number n_2 of intervals with $y < a$ we perform a dominance counting query with *a* on the set V_2 that contains all endpoints y of segments in the set. Clearly $n_0 = n_1 + n_2$. Hence, a 1-dimensional rectangle intersection counting query can be solved using two 1-dimensional dominance counting queries.

Also in the 2-dimensional case we will solve the problem by determining the number n_0 of rectangles that do not intersect the query rectangle. To this end we first compute the number n_a of rectangles in the set that lie completely above the query rectangle. This can be done using a 1-dimensional dominance counting query, by considering the projections of the rectangles on the y-axis. Similarly, we can determine the numbers n_b, n_l and n_r of rectangles that lie below, to the left and to the right of the query rectangle, respectively. In this way the rectangles that lie completely in one of the areas A, B, C or D of Fig. 2 have been counted twice. Let us only consider area D. The number n_D of rectangles that lie completely in area D can be determined by performing a 2-dimensional dominance counting query with the lower left point of the query rectangle on the set V_{ur} that contains the upper right point of each rectangle in the set. In a similar way one can compute the numbers n_A, n_B and n_C of rectangles that lie in A, B and C, respectively, using 2-dimensional dominance counting. Clearly, $n_0 = n_a + n_b + n_l + n_r - n_A - n_B - n_C - n_D$. Hence the 2-dimensional rectangle intersection counting problem can be solved using 8 instances of the 2-dimensional dominance counting problem.

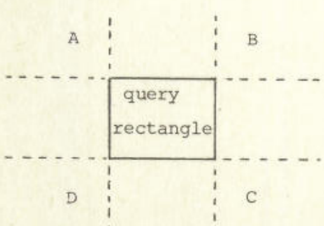

Fig. 2.

The generalizations to the d-dimensional case are straightforward and left as an easy exercise to the reader. It follows that the d-dimensional rectangle intersection counting problem can be solved using a number of instances of the d-dimensional dominance counting problem. □

We have shown the following.

Theorem 3.2. The d-dimensional rectangle intersection counting problem and the d-dimensional dominance counting problem are equivalent.

4. Extensions

Beside the three rectangle searching problems considered in the previous sections, numerous other rectangle searching problems can be defined. For example, one may ask for all rectangles that lie completely to the right of a given rectangle x or for those rectangles whose boundaries do not intersect the boundary of x (i.e., those rectangles that enclose x, are contained in x or do not intersect x). In the 1-dimensional case, i.e., when rectangles are intervals on a line, one can define 64 different types of rectangle searching problems. By an exhaustive case study it can be shown that the counting variant of each of these 64 problems is equivalent to the 0-, 1- or 2-dimensional dominance counting problem (see [4]). Defining a d-dimensional rectangle searching problem as being the Cartesian product of 1-dimensional rectangle problems, there are 64^d different d-dimensional rectangle searching problems. Edelsbrunner and Overmars [4] show the following theorem.

Theorem 4.1 ([4]). The counting variant of each d-dimensional rectangle searching problem is equivalent to the d′-dimensional dominance counting problem for some d′ with $0 \leqslant d' \leqslant 2d$.

For some of the problems equivalence can even be shown for the searching versions of the problems but in most cases it remains an open question whether or not the searching versions are equivalent.

Finally, some remarks on lowerbounds. Fredman [5] proved lowerbounds for the range searching problem (a generalization of the dominance problem in which we ask for those elements in a d-dimensional pointset that lie within a given rectangle) but these do not apply to the dominance searching problem. A lowerbound for the dominance searching/counting problem would immediately lead to lowerbounds for all reactangle searching/counting problems, due to the equivalences described above.

References

[1] J.L. Bentley, Multidimensional divide and conquer, Comm. ACM 23 (1980) 214–229.

[2] H. Edelsbrunner, Dynamic rectangle intersection searching, Rept. F47, Institut für Informationsverarbeitung, Technical University of Graz (1980)

[3] H. Edelsbrunner and H.A. Maurer, On the intersection of orthogonal objects, Inform. Process. Lett., to appear.

[4] H. Edelsbrunner and M.H. Overmars, Equivalences between rectangle problems, unpublished notes, 1981.

[5] M.L. Fredman, A lowerbound of the complexity of orthogonal range queries, J. ACM 28 (1981) 696–705.

[6] D.T. Lee and C.K. Wong, Finding intersections of rectangles by range search, J. Algorithms 2 (1981) 337–347.

[7] E.M. McCreight, Efficient algorithms for enumerating intersecting intervals and rectangles, Rept. CSL-80-9, XEROX Parc (1980).

[8] E.M. McCreight, Priority search trees, Rept. CSL-81-5, XEROX Parc (1981).

[9] M.H. Overmars and J. van Leeuwen, Worst-case optimal insertion and deletion methods for decomposable searching problems, Inform. Process. Lett. 12(4) (1981) 168–173.

[10] V.K. Vaishnavi and D. Wood, Rectilinear line segment intersection, layered segment trees and dynamization, Rept. 80-CS-8, Unit for Computer Science, McMaster University (1980).