Lower Bounds for Expected-Case Planar Point Location

Theocharis Malamatos[*]

Abstract

Given a planar polygonal subdivision S, the point location problem is to preprocess S into a data structure so that the cell of the subdivision that contains a given query point can be reported efficiently. Suppose that we are given for each cell $z \in S$ the probability p_z that a query point lies in z. The entropy H of the resulting discrete probability distribution is a lower bound on the expected-case query time. Further it is known that it is possible to construct a data structure that answers point-location queries in $H + 2\sqrt{2H} + o(\sqrt{H})$ expected number of comparisons. A fundamental question is how close to the entropy lower bound H the exact optimal expected query time can reach. In this paper we show that there exists a query distibution Q over S such that even when we are given complete information on Q, the optimal expected query time must be at least $H + \Omega(\sqrt{H})$, which differs just by a constant factor in the second order term from the best known upper bound.

1 Introduction

Planar point location is among the important two-dimensional search problems. Given a polygonal subdivision S of linear complexity in n, the goal is to preprocess S so that, given any query point q, the cell containing q can be computed efficiently. During the last twenty-five years a number of elegant techniques have been developed that solve the problem in asymptotically worst-case optimal $O(\log n)$ query time using $O(n)$ space [14]. In [7] Goodrich, Orletsky and Ramaiyer posed the question of determining the exact constant factor in the query time. The question was answered by Seidel and Adamy [13] who presented a method with $\log n + 2\sqrt{\log n} + o(\sqrt{\log n})$ time (where log denotes base-two logarithm) and $O(n)$ space and proved a nearly matching lower bound.

In many applications query points exhibit a highly non-uniform distribution among the cells. This raises the question of minimizing the expected-case query time. Suppose that we are given for each cell $z \in S$ the probability p_z that z contains a query point. (For simplicity, we assume that the probability that a query point lies on a segment or vertex of S is zero.) The entropy of S, denoted H throughout, is defined as $H = \sum_{z \in S} p_z \log(1/p_z)$. For the one-dimensional restriction of this problem, a classic result by Shannon implies that the expected number of comparisons for a query is at least as large as the entropy of the probability distribution [9]. Mehlhorn [12] showed that it is possible to construct a binary search tree whose expected query time is at most $H + 2$.

The entropy lower bound H clearly applies to the two-dimensional case as well, nonetheless only recently methods have been proposed whose query time can be upper bounded by a function of entropy. Arya *et al.* [1] showed that for subdivisions consisting of convex polygons, $O(H)$ expected query time can be achieved assuming a certain restricted class of query distributions. Arya, Malamatos, and Mount [2] for the case of polygonal subdivisions consisting of cells of constant combinatorial complexity and for any query distribution presented a method that answers queries in at most $H + 2\sqrt{2H} + o(\sqrt{H})$ time using $O(n^{1+\epsilon})$ space. The space of this method was subsequently improved to $O(n \log^* n)$ by the same authors [3] and eventually to the optimal $O(n)$ by Arya *et al.* [5] while preserving the $H + O(\sqrt{H} + 1)$ query time. In related work Arya, Malamatos and Mount [4] presented a simple and practical randomized algorithm with $O(H)$ time and $O(n)$ space and Iacono [8] developed a similar deterministic method achieving the same bounds.

It is natural to ask what the exact expected case query complexity of planar point location is. Can we achieve the entropy lower bound H (within some small additive constant) similarly to the one-dimensional case [12] or can we justify the presence of the $\sqrt{H}$ term in the upper bound of [2]? In this paper we present the following result: Let Q be a query distribution over some subdivision S and assume that Q is given completely at construction time. Then there exists a query distribution Q such that the expected query time is at least $H + \frac{1}{64}\sqrt{H} - O(1)$. This result shows that in two dimensions the entropy lower bound cannot be reached exactly and it implies that the upper bound in [2] is at most a multiplicative factor in the second order term far from optimal. We mention that a lower bound of $H + \sqrt{H} - O(1)$ has been shown in [11] assuming that the query distribution within the cells of S is unknown but the main (adversary) idea behind this bound does not apply when complete information on Q is available.

[*] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, tmalamat@mpi-inf.mpg.de

2 Preliminaries

Our results are established in the *trapezoidal search graph* model (TSG model, for short) introduced by Seidel and Adamy [13] which forms the basis of (nearly) all point-location algorithms. In this model two standard types of *comparisons* are used in order to locate a query point. The first type determines whether the query point lies to the left or right of a vertical line passing through a vertex in the subdivision S. The other determines whether the query point lies above or below a segment of S. This second type of comparison is only performed after we have determined that the x-coordinate of the query point lies between the two x-coordinates of the endpoints of the segment. (For simplicity we assume that no segment in S is vertical.) The query time is measured in terms of the total number of comparisons used.

For the purposes of analysing the query time, any comparison-based point location method can be represented as *binary space partition (BSP) tree* [6]. (Note that the search structure associated with a method may be a dag instead of a tree, but this only affects the space requirements and not the query time.) In the remainder we always describe a method in terms of its corresponding BSP tree. Given a BSP tree for S, answering a query translates to locating by a simple descent the leaf of the BSP tree which the query point lies in and reporting the cell in S associated with this leaf. Observe that in the TSG model each tree node is related to a planar region which is a vertically aligned trapezoid.

Seidel and Adamy [13] gave an example subdivision for which any method in the TSG model has worst-case query time at least $\log n + 2\sqrt{\log n} - (1/2)\log\log n - O(1)$. Our analysis for the expected-case has similarities to that in [13] but there is an important distinction. For the worst case, it suffices to show that the depth of some leaf in the BSP tree is large. However, this is not enough to give a good bound on the weighted external path length [9], which is the relevant quantity for expected query time. To establish a good lower bound on this quantity, we need to show the existence of leaves deep in the tree that have a large total weight.

3 The Lower Bound

Let S be a planar subdivision and Q be a query distribution over S. We assume that Q is fully known during preprocessing. Consider a point-location method for S in the TSG model. In this section we give a lower bound on the expected query time which the method must have.

Let $n = 2^k$ where $k \geq 0$ is any integer. Let S_n be the subdivision consisting of the segments in the sets $\{[(0,i),(i,i)]\,|\,0 < i \leq n\}$, $\{[(i,i),(n,i)]\,|\,0 \leq i < n\}$, $\{[(0,i),(0,i+1)]\,|\,0 \leq i < n\}$, $\{[(n,i),(n,i+1)]\,|\,0 \leq i < n\}$, and $\{[(i,i),(i+1,i+1)]\,|\,0 \leq i < n\}$. See Figure 3. Clearly there are $2n$ horizontally aligned trapezoidal cells in S_n. Note that the number of rows is n and that the number of segments in S_n is bounded by $O(n)$. We assume that no query point lies in the outer face of S_n and therefore we can ignore the vertical segments. We also assume wlog that no query point lies on the boundary of any column or row of S_n.

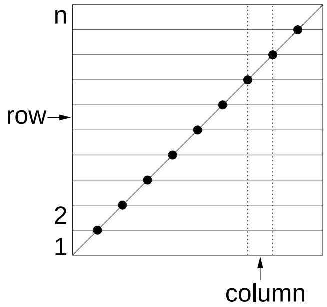

Figure 1: Subdivision S_n and its rows and columns.

Before describing the query distribution Q, we introduce some notations. Let r be a region in S_n. Throughout, we denote by $p(r)$ the probability that a query lies in r given that it lies in some region $r' \subseteq S_n$. When $r' = S_n$ we use $P(r)$ instead of $p(r)$. Assume that we are given that the query point q lies in a region r. Then we denote by $E(r)$ the minimum expected number of comparisons to locate the cell of S_n containing q. An axis-aligned square region b in S_n which has all its four corners at integer coordinates is called a *box*. We say that a box b has *size* n_b where n_b is the number of columns it intersects. A *diagonal* box is a box whose diagonal is contained in the diagonal of S_n.

We define the query distribution $Q = Q(n, \rho)$ over S_n where $\rho > 0$ is a real parameter. First we will form a hierchical partition $\mathcal{P}$ of S_n into certain regions and then we will specify the query distribution over each of these regions. Place a 2×2 grid over S_n. This generates four identical boxes. Assign these boxes to level zero. We call as *D-box* any of the boxes in $\mathcal{P}$ that intersects the diagonal of S_n. (Note that a D-box is also a diagonal box.) Set $n \leftarrow n/2$ and increase the level by one. Recurse this process on each of the two D-boxes, unless $n = 1$ in which case we stop. This completes the partition $\mathcal{P}$ of S_n into a number of boxes. We describe now how the queries are distributed in S_n. Let κ and λ be two real numbers, such that $\kappa + \lambda = 1$ and $\lambda/\kappa = \rho$. To each of the n D-boxes at the last level in $\mathcal{P}$, we assign a query probability equal to κ/n. (For the next lemma, within such a D-box we may choose any arbitrary query distribution.) For $0 \leq i < \log n$, let F_i be the set of non-diagonal boxes at level i. We set $P(F_0) = 0$. Within each F_i for $i \geq 1$ we set the query point to be uniformly distributed with probability $P(F_i) = \lambda/\log(n/2)$. It is easy to see that $P(S_n) = 1$. We now state the main lemma of this section on which Theorem 5 is based.

Lemma 1 *Let $S = S_n$ be a planar subdivision and let $Q = Q(n, \rho)$ be the query distribution over S, where $n = 2^k$ for any integer $k \ge 1$ and $0 < \rho \le \frac{1}{64}$. Let $R \subseteq S$ be a diagonal box of size n' where $2^{k-1} < n' \le 2^k$. Consider a point-location method in the TSG model. Then any such method for R must have expected query time at least*

$$\log n' + \frac{1}{8}\sqrt{\rho \cdot \log n'} - 1.$$

Proof. The proof is by induction on k. Clearly, for $k \le 4$ the induction basis is true if the negative term in the hypothesis is at least 5. With some more care, we can show that in fact 1 suffices. (We omit this proof here.) So let $k > 4$. Let $\mathcal{T}$ be the BSP tree constructed by the method on box R. Let $\mathcal{T}'$ be the subtree of $\mathcal{T}$ consisting of all nodes that can be reached from the root of $\mathcal{T}$ using only vertical comparisons. The leaves of $\mathcal{T}'$ partition the box R into a number of vertical slabs. Let X denote the set of these slabs. (Note that for $n' > 1$ the first comparison on R must be a vertical one.)

Let n_s denote the number of columns in a slab $s \in X$. We have $\sum_{s \in X} n_s = n'$. Let $p(s)$ denote the probability of the query point lying in s given that it lies in R. Let $H_{\mathcal{T}'}$ denote the entropy of the leaves in $\mathcal{T}'$. By linearity of expectation, the expected query time $E(R)$ using $\mathcal{T}$ satisfies

$$E(R) \ge H_{\mathcal{T}'} + \sum_{s \in X} p(s)E(s),$$

where $E(s)$ is the expected query time of locating the query point given that it lies in s. Note that $H_{\mathcal{T}'} = \sum_{s \in X} p(s)\log(1/p(s))$ and thus

$$E(R) \ge \sum_{s \in X} p(s)\log(1/p(s)) + \sum_{s \in X} p(s)E(s). \qquad (1)$$

We show next how to compute a lower bound for $p(s)E(s)$. Here is a brief overview. For each such slab s, we find a region $v_s \subseteq s$ where it is possible to use the induction hypothesis. Then we find a second region $u_s \subseteq s$, disjoint from v_s, which we analyse directly. After computing independently lower bounds for v_s and u_s, by linearity of expectation we combine them to get a lower bound for s.

We distinguish two cases depending on the size of s and its placement in S_n. The first case is when s intersects at most two 1-level D-boxes. Let w_s be the diagonal box of size n_s that is contained in s. (See Figure 2.) Let n'_s be the smallest power of 2 which is at least equal to n_s. (Note that $n_s \le n'_s < 2n_s$.) We define v_s to be the intersection of w_s with the union of all D-boxes of size $n'_s/2$.

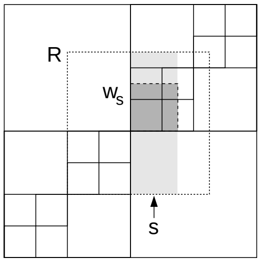

Figure 2: Regions R, s, w_s and boxes in $\mathcal{P}$ at levels 0, 1, and 2. For clarity, the segments of S_n have been omitted.

By definition of $\mathcal{T}'$, the next comparison after reaching slab s must be a comparison with a horizontal segment crossing s. Note that this segment lies outside region v_s. Thus, by linearity of expectation, we can write

$$p(s)E(s) \ge p(v_s)(1 + E(v_s)) + p(u_s)E(u_s),$$

and by summing up over all slabs in X,

$$\sum_{s \in X} p(s)E(s) \ge \sum_{s \in X} p(v_s)(1 + E(v_s)) + \sum_{s \in X} p(u_s)E(u_s). \qquad (2)$$

Since by definition any 0-level non-diagonal box contains q with zero probability, we may consider v_s as a diagonal box R' of size n_s over a subdivision $S_{n'_s}$ where the query distribution is $Q(n'_s, \rho')$ with $\rho' = (\lambda \log(n'_s/2)/\log(n/2))(1/\kappa) = \rho \log(n'_s/2)/\log(n/2)$. (Note that $n'_s/2 < n_s \le n/2$.) Thus, by induction we get that $E(v_s) \ge \log n_s + \frac{1}{8}\sqrt{\rho' \log n_s} - 1$.

Next we compute a lower bound on $p(v_s)$. Let $P(R)$ be the probability that q lies in R given that it lies in S_n. Set $a = 1/P(R)$.

Lemma 2 $p(v_s) \ge (an_s/n)\left(1 - \frac{\lambda \log(8n/n_s)}{\log(n/2)}\right)$.

Lemma 3 *The contribution to the expected query time from region v_s is at least*

$$p(v_s)(1 + E(v_s)) \ge \frac{an_s}{n}\left(\log n_s + \frac{1}{8}\sqrt{\rho \log n'} - \frac{1}{14} - \frac{\lambda \log(n'/n_s)}{\log(n/2)}\log n_s - \frac{1+\lambda}{8}\sqrt{\frac{\rho}{\log(n/2)}}\log(n'/n_s)\right).$$

The proof of Lemma 3 follows from the above bound on $E(v_s)$ and Lemma 2.

The second case is when a slab s intersects three or four 1-level D-boxes (s has large width). In this case we can also show a similar bound with that of Lemma 3. Due to lack of space most proofs of the paper including this one are given in [10].

We now focus on region u_s. We select a set I of D-boxes lying inside R. This set is chosen as follows. Traverse the tree corresponding to the hierachical partition $\mathcal{P}$ top-down visiting only nodes that are associated with D-boxes. If a D-box which lies completely in R is visited then we include it in set I and backtrack. Clearly this process gives a set of disjoint D-boxes whose union covers the diagonal of R.

In the following we assume that a slab s intersects at most one box $b \in I$. (The case where a slab intersects two or more such boxes is handled in [10].) We define $u_s = \{s \cap F_i | \log(n/n_b) \le i \le \log(n/2n'_s)\}$. Note that u_s and v_s are disjoint.

Fix a subregion $C_i = s \cap F_i$ for some value i in the previous range. Observe that any column in C_i is the same point-location subdivision (and has the same query distribution) with any other column in C_i, expect that it is permutated. Also for a single column of C_i, point location reduces to searching in the one-dimensional case. Since each column in C_i covers $n/2^{i+1}$ rows of S_n and the query distribution within F_i is uniform, it follows that $E(C_i) \ge \log\left(n/2^{i+1}\right) \ge \log n - i - 1$. Clearly $p(C_i) = (an_s/n)(\lambda/\log(n/2))$. Now by linearity of expectation for C_i's we can compute a lower bound on $p(u_s)E(u_s)$.

Lemma 4 *The contribution to expected query time from regions u_s for $s \in X$ is at least*

$$\sum_{s\in X} p(u_s)E(u_s) \ge \sum_{s\in X}\left(\frac{a\lambda n_s}{2n\log(n/2)}\right)\log(n'/n_s) \cdot(\log n + \log n_s) - \frac{1}{6}.$$

Finally to show the induction we apply Lemmas 3 and 4 in Eq. (2) and then we substitute the result in Eq. (1). Using some simple bounds for $p(s)$, a and λ, and simplifying completes the proof. (See in [10].)

□

Let $S = S_n$, $Q = Q(n, \frac{1}{64})$ and adjust the query distribution at the last level of $\mathcal{P}$ so that each cell receives $(1/2n)$ probability. It follows that $H = \log n + 1$. Also note that for any subdivision of $O(n)$ size, we always have $H \le \log n + O(1)$. By applying Lemma 1 for $R = S_n$ we can easily obtain the following theorem:

Theorem 5 *For any $n \ge 2$ there is a subdivision S consisting of n cells of bounded complexity and a query distribution Q over S which is fully known such that any point-location method for S in the TSG model must use at least $H + \frac{1}{64}\sqrt{H} - O(1)$ expected number of comparisons, where H is the entropy of S.*

References

[1] S. Arya, S.-W. Cheng, D. M. Mount, and H. Ramesh. Efficient expected-case algorithms for planar point location. In *Proc. 7th Scand. Workshop Algorithm Theory*, volume 1851 of *Lecture Notes Comput. Sci.*, pages 353–366. Springer-Verlag, 2000.

[2] S. Arya, T. Malamatos, and D. M. Mount. Nearly optimal expected-case planar point location. In *Proc. 41 Annu. IEEE Sympos. Found. Comput. Sci.*, pages 208–218, 2000.

[3] S. Arya, T. Malamatos, and D. M. Mount. Entropy-preserving cuttings and space efficient planar point location. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 256–261, 2001.

[4] S. Arya, T. Malamatos, and D. M. Mount. A simple entropy-based algorithm for planar point location. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 262–268, 2001.

[5] S. Arya, T. Malamatos, D. M. Mount, and K. C. Wong. Optimal expected-case planar point location. Technical Report HKUST-TCSC-2004-09, Dept. of Computer Science, Hong Kong University of Science and Technology, 2004. `http://www.cs.ust.hk/tcsc/RR/`.

[6] M. D. Berg, M. V. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.

[7] M. T. Goodrich, M. Orletsky, and K. Ramaiyer. Methods for achieving fast query times in point location data structures. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 757–766, 1997.

[8] J. Iacono. Expected asymptotically optimal planar point location. *Comput. Geom. Theory Appl.*, 29:19–22, 2004.

[9] D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 2nd edition, 1998.

[10] T. Malamatos. Lower bounds for expected-case planar point location. Full version, 2005. `http://www.mpi-inf.mpg.de/~tmalamat`.

[11] T. Malamatos. *Expected-case planar point location*. PhD thesis, Dept. of Computer Science, Hong Kong University of Science and Technology, 2002.

[12] K. Mehlhorn. Best possible bounds on the weighted path length of optimum binary search trees. *SIAM J. Comput.*, 6:235–239, 1977.

[13] R. Seidel and U. Adamy. On the exact worst case query complexity of planar point location. *J. Algorithms*, 37:189–217, 2000.

[14] J. Snoeyink. Point location. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 2nd edition, 2004.