Maximum Line-Pair Stabbing Problem and its Variations*

Sebastian Böcker Veli Mäkinen†

Abstract

We study the *Maximum Line-Pair Stabbing Problem*: Given a planar point set S, find a pair of parallel lines within distance ϵ from each others such that the number of points of S that intersect (stab) the area in between the two lines is maximized. There exists an algorithm that computes maximum stabbing in $O(|S|^2)$ time and space. We give a more space-efficient solution; the time complexity increases to $O(|S|^2 \log |S|)$, but the space reduces to $O(|S|)$. Our algorithm also extends to a dual problem where one searches for a line stabbing maximum number of variable size circles; as far as we know, this problem has previously been studied only on fixed size circles.

A variant of the stabbing problem equals a one-dimensional point set matching problem under translations, scalings, and errors. We study a version of this problem, where the matching has to be a one-to-one mapping. Existing techniques based on incremental maintenance of maximum matching using augmenting paths yield $O((mn)^3)$ time solution, where m and n are the sizes of the point sets to be matched. Our new algorithm achieves $O((mn)^2(m+n))$ time. The improvement is based on an observation that in our case the match-graph has a regular shape, and the maximum matching can be updated more efficiently.

1 Introduction

There are three dual ways to describe the *Maximum Line-Pair Stabbing Problem* (see Fig. 1):

- Given a set of points, find a pair of parallel lines within distance ϵ from each others such that the number of points in between the lines (including them) is maximum. For short, the resulting line-pair is said to stab maximum number of points.
- Given a set of circles with diameter ϵ, find a line that goes through maximum number of them.
- Given a set of lines, find a vertical line segment starting at point (x, y) of length $\delta(x)$ that crosses maximum number of lines. (Function δ will be defined later.)

*Technische Fakultät, Universität Bielefeld, Germany, {boecker,veli}@cebitec.uni-bielefeld.de. Supported by the ''Deutsche Forschungsgemeinschaft'' (BO 1910/1-3) within the Computer Science Action Program.

†Also partially supported by the Academy of Finland.

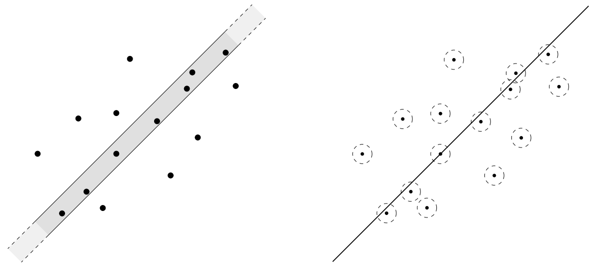

Figure 1: Stabbing points with a line-pair (left) and stabbing circles with a line (right).

Our motivation to study the problem comes from the calibration of Mass Spectrometry data [4]. There we basically have two sets of real values, one corresponding to the theoretical reference spectrum, and one for measured spectrum. We know that there is a linear function mapping the measured values close to the reference values. One way to find such transformation is to map each pair (reference value, measured value) into a plane, and find a line-pair stabbing maximum number of points.

The maximum stabbing problem has been studied earlier by Chin, Wang, and Wang [6]. They gave a quadratic time and space algorithm that is based on the third dual interpretation given above.

We use the same dual mapping in our algorithm as [6], but in a different way. The consequence is that we are able to manage with linear space, but the time complexity increases by a logarithm factor.

There is also an indirect way to solve the problem using geometric range query data structures. This solution is conceptually easiest to understand, but it also uses nearly quadratic space.

Returning back to the motivation [4] on two sets of real values, we note that the maximum stabbing problem is an over-generalization of that problem; instead of restricting by ϵ the Euclidean distance between the two lines, one can restrict the distance in the one-dimensional projection. As such the problem is not well-defined as it has a degenerate optimum solution. We study a robust variant of this problem, where we require that the linear function mapping one set to the other maximizes the number of one-to-one matches. This problem can be solved using existing techniques for incremental bipartite matching. We give a new, one order of magnitude faster, solution to the problem.

2 Maximum Stabbing using Half-Space Range Counting Queries

Let $S \subset \mathbb{R}^2$ be a point set and $\epsilon > 0$ a given parameter defining an instance of the maximum line-pair stabbing problem throughout this paper. In the following, we derive an $O(|S|^2 \log |S|)$ time solution to the problem using data structures for half-space range searching [1].

Let us denote by $\mathcal{L}$ the set of feasible line-pairs, i.e. those pairs of parallel lines within distance ϵ from each others. The solution is as follows:

(i) Build the data structure of Chazelle [5] as pointed out in [1, Theorem 4.4, p. 24] for half-space range counting queries on S. The data structure can be constructed in $O(|S|^2)$ time and it occupies $O(|S|^2/\log^2|S|)$ space. For any line, it gives the number of points on either side of it in $O(\log|S|)$ time.

(ii) Construct the set $L \subset \mathcal{L}$ of *representative* line-pairs as follows. On each pair of points of S, pair the line going through these points with the two parallel lines at ϵ distance from it. Add these two line-pairs to L. Also, on each pair (p, s) of points of S having distance greater or equal to ϵ, add the (two) line-pair(s) to L formed by the parallel lines at ϵ distance from each others such that one line goes through p and one through s.

(iii) Make two queries on the data structure on each line-pair in L, to compute how many points of S intersect the area in between the lines.

(iv) Choose the line-pair that obtains the maximum count.

The correctness of the method follows using a shifting argument to show that it is enough to consider the set of representative line-pairs.

The algorithm makes $O(|S|^2)$ queries each taking $O(\log|S|)$ time, giving overall running time $O(|S|^2 \log|S|)$. The space usage is $O(|S|^2/\log^2|S|)$.

3 Direct Solution to Maximum Stabbing Problem

We solve the problem using the third dual interpretation mentioned in the introduction. We map each point $p = (p_x, p_y) \in S$ into a line $p^* : y = p_x x - p_y$. A line $\ell : y = mx + b$ is mapped into a point $\ell^* = (m, -b)$. Consider another line $\kappa : y = mx + (b + \delta(m))$ parallel to ℓ, where $\delta(m)$ is defined so that the distance of κ and ℓ is ϵ. That is, $\delta(m) = \epsilon\sqrt{1^2 + m^2}$. This line is mapped into a point $\kappa^* = (m, -b - \delta(m))$. One easily notices, that the parallel lines in between κ and ℓ, are mapped into a line segment $m \times [-b - \delta(m), -b]$. Hence, our original problem equals that of finding a line segment $m \times [-b - \delta(m), -b]$ such that the number of lines it intersects is maximal, over all choices of m and $-b$. We are ignoring vertical line-pairs for the moment.

Let us now derive the algorithm that finds the optimum values for m and $-b$ in time $O(|S|^2 \log|S|)$ and space $O(|S|)$. Using again a shifting argument we notice that $(m, -b)$ can be chosen so that $-b = p_x m - p_y$ for some $(p_x, p_y) \in S$. Still our space of choices for m and $-b$ is infinite. To make it finite, we notice that fixing two lines partitions the range of m along one line into constant number of relevant ranges.

To be more precise, let us denote by S^* the set of lines that are images of points in S. Consider two lines $p^* : p_x x - p_y$ and $q^* : q_x x - q_y$ of S^*, and the case where $(m, -b)$ is chosen so that $-b = p_x m - p_y$. Equation

$$p_x m - p_y - \delta(m) = q_x m - q_y \tag{1}$$

has only constant number of solutions (if any), which means that there are only constant number of ranges $R \subseteq \mathbb{R}$ such that $m \in R$ if and only if point $(m, q_x m - q_y)$ is included in the line segment $m \times [p_x m - p_y - \delta(m), p_x m - p_y]$. Furthermore, the solutions to (1) can be found in constant time. Now, let $\mathbf{R}(p^*)$ be the multiset of ranges of line p^* formed by repeating the above process for each $q^* \in S^*$. After sorting the endpoints (y-coordinates) of the ranges in $\mathbf{R}(p^*)$ into ascending order, attaching to each endpoint value $+1$ or -1 depending on whether the endpoint is a start or end of a range, one can scan through the endpoints keeping a counter how many ranges are active at each phase. Endpoints with the same coordinates are sorted so that those associated with $+1$ precede those associated with -1. The endpoint associated with the largest count gives the optimal choice for $(m, -b)$ restricted to values such that $-b = p_x m - p_y$. The same process can be repeated constructing $\mathbf{R}(p^*)$, sorting it, and computing the largest count, for each $p^* \in S$. The optimum choice for $(m, -b)$ corresponds then to the overall largest count.

We have now found the optimum line segment $m \times [-b - \delta(m), -b]$ for the dual problem and hence we have found the line-pair stabbing the maximum number of points of S for the primal problem. The space requirement is that needed for storing set of ranges $\mathbf{R}(p^*)$ at each phase, i.e. $O(|S^*|) = O(|S|)$. The time requirement is that of sorting $\mathbf{R}(p^*)$ on each $p^* \in S^*$, i.e. $O(|S| \cdot |S| \log|S|) = O(|S|^2 \log|S|)$. Finally, we note that if the solution is a vertical line-pair, our algorithm does not find it because the mapping to the dual plane does not apply for such lines. However, in this case, a simple vertical sweep is enough to compute the maximum stabbing over the linear number of relevant vertical line-pairs. This will only take $O(|S| \log|S|)$ time, which is negligible.

In the full version, we show how to handle variable size circles in the second dual interpretation, giving:

Theorem 1 *Given a set C of variable size circles on the plane, one can find the line going through maximum number of them in time $O(|C|^2 \log |C|)$ and space $O(|C|)$. The maximum line-pair stabbing problem is a special case, and can be solved within the same time and space bounds.*

4 One-Dimensional Point Set Matching under Translations, Scalings, and Errors

As explained in the introduction, when restricting to ϵ distances measured in the one-dimensional projection, the stabbing problem has a simpler interpretation as a point set matching problem: Given two sets of real values, i.e. one-dimensional point sets, $A, B \subset \mathbb{R}$, find a linear function $f : \mathbb{R} \to \mathbb{R}$ such that $|M_f(A, B, \epsilon)|$ is maximum, where M_f is a matching of A and B such that for each $(a, b) \in M_f$ holds $|f(a) - b| \leq \epsilon$. We call this variant of the stabbing problem the *linear one-dimensional point set matching* problem.

To solve the problem, consider a set F of representative linear functions constructed as follows: Let $B(\epsilon) = \{p - \epsilon, p + \epsilon \mid p \in B\}$. For each quadruple (a', a, b', b) such that $a', a \in A$ with $a' < a$ and $b', b \in B(\epsilon)$ with $b' < b$, add function $f(x) = \frac{b-b'}{a-a'}(x - a') + b'$ to F. Each function in F defines a translation and scaling that maps two points of A into ϵ distance from some points of B. Conditions $a' < a$ and $b' < b$ prevent reflections. Again using the shifting argument, one observes that this is the sufficient set of transformations to be examined. The size of this set is $O((mn)^2)$, where $m = |A|$ and $n = |B|$. To find the optimum transformation f, we construct all M_f for $f \in F$ incrementally, and choose the f that corresponds to the largest M_f: For each representative translation $b' - a'$, where $a' \in A$ and $b' \in B(\epsilon)$, construct the set of scale ranges $R(a', b') = \{[\frac{b-\epsilon-b'}{a-a'}, \frac{b+\epsilon-b'}{a-a'}] \mid a \in A, b \in B\}$. Sort the endpoints of ranges in $R(a', b')$ into increasing order, and scan through them incrementing and decrementing a counter analogously as explained before. This gives the optimum scale for the fixed translation. Repeating the process for all representative translations gives the overall optimum transformation. Noticing that the scale ranges corresponding to a fixed $a \in A$ can be obtained in sorted order by scanning through sorted B, the algorithm can be implemented to run in $O((mn)^2 \log m)$ time by merging the m sorted lists at each phase.

4.1 One-to-one Mapping Case

The solution M_f obtained with the above algorithm does not necessarily define a proper mapping between A and B; a point of A may be mapped into ϵ distance from several points of B, and vice versa. In fact, on most instances there is a degenerate optimum solution mapping all points of A into ϵ-distance from one point of B. In applications, such degenerated cases can be avoided by restricting the search space. However, a more rigorous way to define the problem is to search for M_f that contains the largest one-to-one mapping.

A brute-force algorithm to solve the one-to-one mapping case is as follows: At each phase of the previous algorithm that constructs sets M_f incrementally, construct a bipartite graph G_f having edges between $a \in A$ and $b \in B$ if and only if $(a, b) \in M_f$. Solve the maximum matching problem on each G_f, and choose f corresponding to the overall largest maximum matching.

Notice that the graphs G_f change only by one edge at each incremental step. Alt et al. [3, p. 246] describe a solution to a similar geometric problem exploiting this fact: As the maximum matching can only change by one at each phase, it is enough to check whether the new graph has an augmenting path. If so, use it to produce the new maximum matching. Otherwise the maximum matching does not change. Checking for an augmenting path takes $|M_f|$ time. In the worst case each M_f is of size $O(mn)$, and the whole algorithm takes $O((mn)^3)$ time.

The above technique yields the best known upper bound for the problem studied in [3]. Many consequent papers have studied different variations of the problem to avoid the costly maximum matching computation, like assuming disjoint error regions, transformations minimizing the Hausdorff distance, etc. For references, see survey [2].

Our problem, however, has an extra property that allows a more efficient way to find the maximum matchings. We will next show how to find the maximum matching in $O(m + n)$ time using a greedy method. To describe this solution, we first introduce some helpful notions to characterize our problem.

We say that a binary matrix $B(1 \ldots m, 1 \ldots n)$ containing values **0** and **1** is a *staircase matrix* if the following conditions hold:

(i) Each row of the matrix contains at most one *run* of **1**s, i.e. a maximal range of consecutive cells each containing value **1**.

(ii) Let i' and i, $i' < i$ be two rows containing a run of **1**s. Let the run at row i' cover indexes $c_{i'}, c_{i'} + 1, \ldots, d_{i'}$ and the run at row i cover indexes $c_i, c_i + 1, \ldots, d_i$. Then $c_{i'} \leq c_i$ and $d_{i'} \leq d_i$.

Notice that from (i) and (ii) follows identical conditions on columns, i.e. B is a staircase matrix if and only if B^{T} is staircase matrix.

Let $A = a_1 a_2 \cdots a_m$ and $B = b_1 b_2 \cdots b_n$ be the two point sets to be matched. We assume that the point sets are given sorted in ascending order. Let us consider a fixed M_f such that $f(x) = s(x - a') + b'$, where $s = \frac{b-b'}{a-a'} \geq 0$. Consider a *match matrix* $M(1 \dots m, 1 \dots n)$ having $M(i,j) = \mathbf{1}$ if $(a_i, b_j) \in M_f$, otherwise $M(i,j) = \mathbf{0}$. It follows easily from definitions (proof omitted):

Lemma 2 *The match matrix M is a staircase matrix.*

On fixed translation and scale, our problem reduces to finding a maximum size one-to-one matching R of rows and columns of M such that for each $(i,j) \in R$ holds $M(i,j) = \mathbf{1}$. Let R^* be one such maximum matching. We next show that one can obtain in $O(m+n)$ time a matching R that is as good as any R^*. To show this, we first prove (in Lemma 3) that there is always an *order-preserving* matching $\hat{R}$ that is as good as R^*; matching R is order-preserving if, for every pair $(i',j'), (i,j) \in R$, we have $i' \leq i$ if and only if $j' \leq j$. Then Lemma 4 constructs the algorithm.

Lemma 3 *There is a maximum matching $\hat{R}$ that is order-preserving.*

Proof. We show that there is an algorithm to convert any maximum matching R^* into an equally good order-preserving matching in finite number of steps: Let $\check{R}$ be the set of all pairs in R^* that have at least one *conflict* with a pair in R^*: $(i',j') \in \check{R}$ if and only if there is $(i,j) \in R^*$ such that pairs (i',j') and (i,j) conflict the order-preserving condition. Let $(i_{min}, j') \in \check{R}$ be the pair having the smallest index i_{min}, and (i, j_{min}) the conflicting pair for (i_{min}, j') having the smallest index j_{min}. Since $M(i_{min}, j') = M(i, j_{min}) = \mathbf{1}$, we infer $M(i_{min}, j_{min}) = M(i, j') = \mathbf{1}$ from the staircase property. Hence, we can exchange the pairs to remove the conflict, i.e. $R^* \leftarrow (R^* \setminus \{(i_{min}, j'), (i, j_{min})\}) \cup \{(i_{min}, j_{min}), (i, j')\}$. After the exchange, R^* is still a maximum matching, and contains one more pair, namely (i_{min}, j_{min}), that is not conflicting with any other pair. Moreover, when the process is repeated with the new R^*, the new set $\check{R}$ contains only pairs (i,j) such that $i > i_{min}$, where (i_{min}, j') is the pair selected in the previous step. This follows from the fact that R^* is a one-to-one matching. Thus, after at most $|A|$ rounds, set $\check{R}$ is empty, and R^* is order-preserving and has the same matching score as the original one. □

Lemma 4 *A maximum matching R can be obtained greedily in $O(m+n)$ time.*

Proof. Let $\hat{R}$ be any a order-preserving maximum matching. Such $\hat{R}$ exists by Lemma 3. Let (i',j') be the first pair in $\hat{R}$, having the smallest index i'. We can replace it by $(i', c_{i'})$ where c_x denotes the first column having $M(x, c_x) = \mathbf{1}$ at row x. Let (i,j) be the second pair in $\hat{R}$. We can replace it by $(i, \max(c_{i'}+1, c_i))$. Such replacement is always possible due to the staircase property. The same replacement can be applied inductively, obtaining a new order-preserving maximum matching, where the matching column for each row in the originating matching $\hat{R}$ is picked greedily. To complete the argument, we note that one can pick the rows also greedily due to the staircase property without changing the cardinality of the matching: We can traverse row by row choosing the first available matching column, comparing three values: previous matched column, start of the run of $\mathbf{1}$s at the row, and end of the run of $\mathbf{1}$s at the row. This takes overall $O(m+n)$ time, and we obtain a maximum matching R. □

Recall the incremental algorithm that updates the graph G_f scanning scales from left to right for a fixed translation. We can instead represent the graph G_f as a match matrix M. As proven before, M is a staircase matrix in each scale. Deleting or inserting an edge in G_f corresponds to updating the value of a cell in M. Each update extends or reduces a run of $\mathbf{1}$s at some row, and hence we can maintain for each row pointers to the start and end of the run in constant time. We can repeat the greedy algorithm of Lemma 4 at each scale. We conclude:

Theorem 5 *The linear one-dimensional one-to-one point set matching problem on two point sets A and B of sizes m and n, respectively, can be solved in $O((mn)^2(m+n))$ time.*

Acknowledgment

We wish to thank the anonymous referee pointing out the reference [6].

References

[1] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In *Advances in Discrete and Computational Geometry*, Contemporary Mathematics 223, Am. Math. Soc. Press, 1999, pp. 1–56.

[2] H. Alt and L. Guibas. Discrete Geometric Shapes: Matching, Interpolation, and Approximation, In *Handbook of Computational Geometry*, Elsevier, pp. 121–153, 1999.

[3] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, Symmetry, and Similarity of Geometric Objects. *Discr. & Comp. Geom.*, 3(1988):237–256.

[4] S. Böcker. Calibrating Time-of-Flight Mass Spectra. Manuscript, October 2004.

[5] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discr. & Comp. Geom.*, 9(1993), 145–158.

[6] F. Y. L. Chin, C. A. Wang, and F. L. Wang. Maximum Stabbing Line in 2D Plane. In *Proc. COCOON*, Springer-Verlag LNCS 1627, pp. 379–388, 1999.