Noname manuscript No.
(will be inserted by the editor)

Linear Programming and Stabbing Problems on Line Segments and Moving Points

Gerardo González Martínez ·
Mazay Jiménez-Salinas · **Carlos Seara** ·
Jorge Urrutia

Received: date / Accepted: date

Abstract Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in the plane, and suppose that at time $t = 0$ they start moving in the vertical direction at unit speed; that is, if $p_i = (a_i, b_i)$ at time t, p_i has moved to the point $p_i(t) = (a_i, b_i + t)$. Let l_t^i be the closed line segment with end-points p_i and $p_i(t)$, $i = 1, \ldots, n$. In this short paper we address the following problem: Find the smallest time t such that there is a line ℓ that intersects all of the line segments $l_t^1, \ldots, l_t^n$. We prove that our problem can be solved in linear time. We also show that the same problem when the elements of P are points in the d-dimensional space $\mathbb{R}^d$ that move up, and at different speeds can also be solved in linear time for fixed d.

Keywords linear programming, linear time, stabbing objects, geometric separability.

1 Introduction

A classical problem in Computational Geometry that can be solved in linear time using linear programming is the so called *Separability Problem of Red and Blue Points*: Given n points on the plane, some of which are colored red, and the others blue, is there a line that separates the red from the blue points? In two well known papers by Megiddo[5, 6], several other problems in Computational Geometry, including the separability problem of red and blue points, the problem of finding the smallest circle containing a set of points on the plane, and the separability problem of sets of red and blue points in a fixed dimensional space and others, are solved in linear time. In Megiddo[6] it is proved that linear programming in $\mathbb{R}^d$

Gerardo González Martínez · Mazay Jiménez-Salinas
Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México City, México.
E-mail: {gmg_17, m.jimenez}@uxmcc2.iimas.unam.mx.

Carlos Seara
Matemàtiques, Universitat Politècnica de Catalunya, Spain.
E-mail: carlos.seara@upc.edu. Supported by projects Gen. Cat. DGR 2014SGR46, MINECO MTM2015-63791-R and H2020-MSCA-RISE project 734922 - CONNECT.

Jorge Urrutia
Instituto de Mateméticas, Universidad Nacional Autónoma de México, México City, México
E-mail: urrutia@matem.unam.mx. This research was partially supported by UNAM project PAPIIT-IN102117, and H2020-MSCA-RISE project 734922 - CONNECT.

can be solved in $O(f(d)) \times n$, which when d is constant, yields a linear time algorithm. The function $f(d)$ is exponential in d. It is then easy to see that the separability problem of red and blue points in $\mathbb{R}^d$ can be reduced to a linear program in $\mathbb{R}^d$.

Another well studied problem in Computational Geometry, is this: *Given a family of line segments* $\{l_1, \dots, l_n\}$ *find, if it exists, a line (stabber) that intersects all of* $\{l_1, \dots, l_n\}$. It is easy to see that if all of $\{l_1, \dots, l_n\}$ are vertical segments, deciding if they have a stabbing and finding one if it exists, can be solved in linear time. If the segments have different orientations, this problem can be solved in $O(n \log n)$ time, and this is optimal, see[3, 2].

In this short paper we introduce a new variation of stabbing problems arising from line segments generated by sets of moving points. These problems can be considered a hybrid problem combining separability of red and blue points, and stabbing of line segments. We start addressing the following problem:

Problem 1. Let $P = \{p_1, \dots, p_n\}$ be a set of n points in the plane, and suppose that the elements of P start moving up at time $t = 0$ at a constant unit speed, see Figure 1a. Suppose that $p_i = (a_i, b_i)$. As p_i moves up, at time t the point p_i has moved to the point $p_i(t) = (a_i, b_i + t)$, and has traversed a line segment l_t^i of length t, starting at p_i and ending at the point $p_i(t)$. Our problem is to find the smallest t such that there exists a line ℓ that stabs $l_t^1, l_t^2, \cdots, l_t^n$, see Figure 1b.

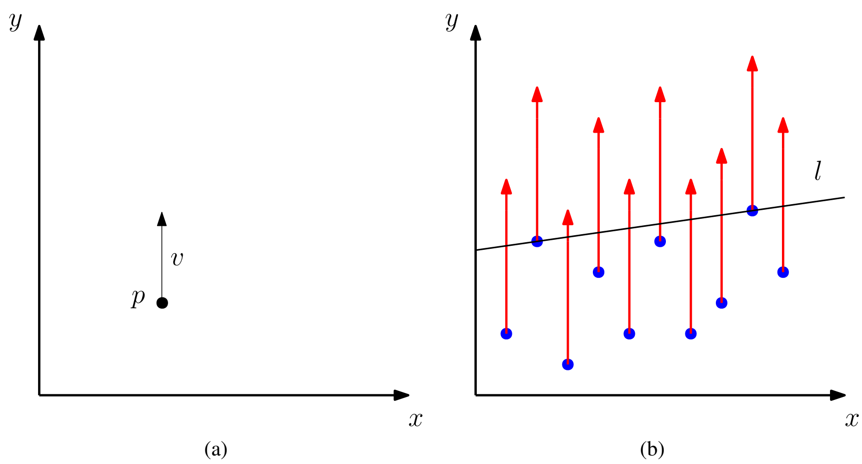


Fig. 1 *a*) Representation of a point and its velocity vector associated to it. *b*) Set of n points in the plane moving vertically at the same speed.

We will show that Problem 1 can solved in $O(n)$ time. We will also prove that the following variations to the previous problem, can also be solved in linear time:

Problem 2. Each point p_i moves vertically at its own speed.
Problem 3. Same problems as above for $p_i \in \mathbb{R}^d$ where d is fixed. In this case, we want to find a hyperplane that intersects the segments traced by the moving points.

We will show that all of the above problems can be solved using linear programming in $\mathbb{R}^{d+1}$, and thus can be solved in linear time for constant d.

1.1 **Related work.**

Other problems solved using linear programming in linear time, include Circular separability of two sets of points R and B was studied by O'Rourke et al. [7], they showed that determining if there is a circle that contains the points of R in its interior while leaving the points of B in its exterior can be done in linear time. For line segments in the plane Edelsbrunner *et al.* [3] presented an algorithm to compute stabbing lines of a set of n line segments (if they exist) in $O(n \log n)$. Later Avis *et al.* [2] presented an $\Omega(n \log n)$ fixed order algebraic decision tree lower bound for determining the existence of a line stabber for a family of n line segments in the plane. For $\mathbb{R}^d$ a hyperplane stabber for n segments can be found in $O(n^d)$ [1], in the same paper an algorithm in $O(n^{d-1}m)$ is presented to find a plane stabber for a set of m polyhedra with a total of n edges. For objects other than lines, Rappaport in [8] presented an algorithm for finding the stabbing region of a set of n disks in $O(n)$ time. In [5] Megiddo presented a set of problems of linear separability that can be solved in linear time using linear programming. In all of the papers related to stabbers and separability, the sets of segments, or points are static. Our contribution is that of introducing a new approach that deal with moving points, and segments that grow with time.

In Section 2, we show how to transform a geometric problem into a linear programming problem and we define some concepts for point to line transformations. In Section 3, we demonstrate that all the described problems can be solved in $O(n)$ time, when d is fixed.

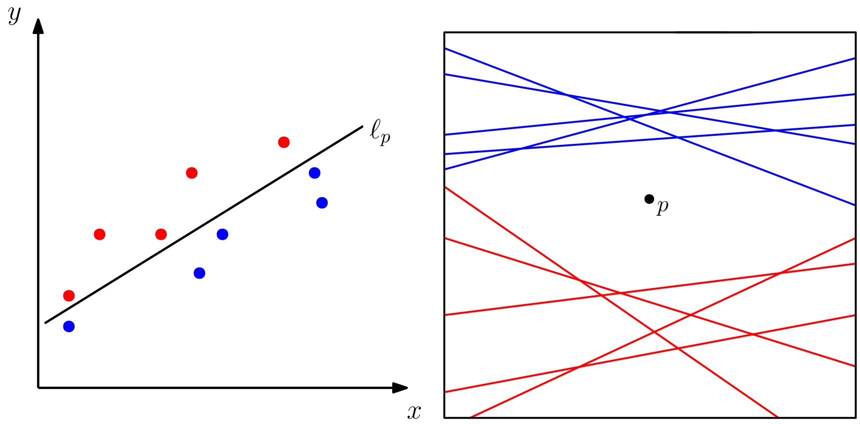
2 Preliminaries

We will start by showing the transformation used by Megiddo to reduce the separability problem of sets of red and blue points to a linear programming problem. The solutions to the problems presented here use similar reductions.

The dual line of a point $p = (a,b)$ in the plane, denoted by ℓ_p, is the non-vertical line with equation $y = ax - b$. The dual of ℓ_p is p. Consider a set $B = \{p_1, \ldots, p_s\}$ of s blue points, and a set $R = \{q_1, \ldots, q_t\}$ of t red points. The problem of deciding if there exists a line ℓ that leaves all of the red points above it, and all of the blue points below it, is then transformed into the following linear programming problem:

$$\begin{aligned} \text{minimize} \quad & y \\ \text{subject to} \quad & \\ & a_i x - y - b_i \leq 0 \\ & a'_j x - y - b'_j \geq 0 \end{aligned}$$

Where $p_i = (a_i, b_i)$, $1 \leq i \leq s$; and $q_j = (a'_j, b'_j)$, $1 \leq j \leq t$. See Figure 2a. Note that if there is a line that separates the set of blue points from the red points, then linear programming problem stated above has a feasible solution, any point (if it exists) in the dual space that is below all of the blue lines, and above the red lines in the dual space, corresponds to a line that separates the blue points from the red points, see Figure 2b. For more details, the reader is referred to Matousek [4], page 21.



(a) (b)

Fig. 2 *a*) Two sets of points red and blue. *b*) Dual plane showing the transformation of red and blue points.

3 Stabbing line segments generated by moving points

We now show how to transform Problem 1 to a linear programming problem. Consider a set of points $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$, and suppose that at time $t > 0$, they have moved to the set of points $P(t) = \{p_1(t), \ldots, p_n(t)\}$ as defined above. We may think of P as a set of red points that are fixed, and $P(t)$ as a set of blue points that are moving upwards. In the dual space, each red point $p_i = (a_i, b_i)$ will be transformed to the line ℓ_i with equation $y = a_i x - b_i$, and each point $p_i(t) \in P(t)$ to the line ℓ'_i with equation $y = a_i x - b_i - t$.

It is now straightforward to see that solving Problem 1 is equivalent to solving the following linear programming problem in $\mathbb{R}^3$ with variables x, y and t:

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & \\ & a_i x - y - b_i \leq 0 \\ & a_i x - y - t - b_i \geq 0 \end{aligned}$$

Thus we have the following result:

Theorem 1 *The smallest time t such that a line ℓ stabs the line segments $l_t^1, \ldots, l_t^n$ can be calculated in $O(n)$ time.*

3.1 Variants on moving points

Let us consider now Problem 2. As above, we can think of the set elements of P as a set of fixed red points, and $P(t)$ as a set of blue points that move upwards at different speeds.

Suppose then that each point p_i moves upwards at speed s_i, thus at time t, point p_i has moved to the point $p_i(t) = (a_i, b_i + s_i t)$.

Then in the dual space every red point $p_i = (a_i, b_i)$ is mapped to the line $y = a_i x - b_i$, and each blue point to the line $y = a_i x - b_i - s_i t$. Problem 2 can be stated as the following linear programming problem in $\mathbb{R}^3$ with variables x, y and t:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \\ & a_i x - y - b_i \leq 0 \\ & a_i x - y - b_i - s_i t \geq 0 \end{array}$$

3.2 **Fixed dimension**

The d-dimensional case can be solved in linear time. Let $\mathbb{R}^d$ be the d-dimensional real space and let $p = (a_1, \ldots, a_d)$ be a point in $\mathbb{R}^d$. As before, in the dual space of $\mathbb{R}^d$, the point $p = (a_1, \ldots, a_d)$ is mapped to the hyperplane with equation $x_d = a_1 x_1 + \cdots + a_{d-1} x_{d-1} - a_d$.

Consider a point set $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, where $p_i = (a_1^i, \ldots, a_d^i)$, $i = 1, \ldots, n$. Suppose that at time $t = 0$ they start moving upwards, each p_i moving at its own speed s_i. Thus at time t, p_i has moved to the point $p_i(t) = (a_1^i, \ldots, a_d^i + s_i t)$, $i = 1, \ldots, n$. Let $P(t) = \{p_1(t), \ldots, p_n(t)\}$.

In the dual space, each point $p_i \in P$ will be transformed to the hyperplane $x_d = a_1^i x_1 + \cdots + a_{d-1}^i x_{d-1} - a_d^i$ and each point $p_i \in P(t)$ is mapped to the hyperplane $x_d = a_1^i x_1 + \cdots + a_{d-1}^i x_{d-1} - a_d^i - s_i t$.

The d-dimensional case can be stated as the following linear programming problem, which can be solved in linear time when d is a constant:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \\ & a_1^i x_1 + \cdots + a_{d-1}^i x_{d-1} - x_d - a_d^i \leq 0 \\ & a_1^i x_1 + \cdots + a_{d-1}^i x_{d-1} - x_d - a_d^i - s_i t \geq 0 \end{array}$$

Theorem 2 *For any fixed dimension d, Problems 1, 2, and 3 can be solved in $O(n)$ time.*

4 Conclusions and final remarks

In this short paper we have introduced a new type of problems for line segments generated by moving points. We proved how these problems can be converted to linear programming problems and can be solved in linear time. Some open problems remain. One of the most interesting ones is the following: As in Problem 1, consider a set of points in the plane that start moving at time $t = 0$, but now each point moves at its own fixed direction and speed.

Find the smallest t for which there is (if it exists) a line that stabs the line segments traced by the moving points. It is not hard to get an algorithm to solve this problem in $O(n^2 \log n)$ time. Can this problem be solved in $O(n \log n)$ time? A way to prove that we cannot solve this problem in better than $O(n \log n)$ time is as follows: As mentioned before, it was proved by Avis *et al* [2] that the problem of deciding wherter there exits a line that stabs a set of line segments cannot be solved in better than $O(n \log n)$ time in the algebraic decision tree model. Using this result, it is not hard to see that when our points move on different directions, our problem solves the stabbing problem of a set of line segments as follows: Let $L = \{l_1, \ldots, l_n\}$ be a set of segments, none of which is horizontal (this restriction can be eliminated easily by applying a rotation). For each segment l_i let p_i and q_i be the lowest and the highest end point of l_i respectively. Suppose that at time $t = 0$ each point p_i start moving towards q_i at a speed proportional to the length of l_i, and thus at time $t = 1$ the point p_i has moved to the position of point q_i, $i = 1, \ldots, n$. As before let $p_i(t)$ be the position of p_i, and let l_t^i be the segment with endpoints p_i and $p_i(t)$. Observe that if the smallest t for which there is a stabber of $L(t) = \{l_t^1, \ldots, l_t^n\}$ is smaller than or equal to 1, then the set L has a stabber. Thus finding such a t cannot be done in better than $O(n \log n)$ time. What about the complexity of the last problem in higher dimensions?

References

1. Avis, D., Doskas, M.: Algorithms for High Dimensional Stabbing Problems, pp. 199–210. Springer Berlin Heidelberg, Berlin, Heidelberg (1988). DOI 10.1007/978-3-642-83539-1_7. URL `https://doi.org/10.1007/978-3-642-83539-1_7`
2. Avis, D., Robert, J., Wenger, R.: Lower bounds for line stabbing. Information Processing Letters **33**(2), 59 – 62 (1989). DOI https://doi.org/10.1016/0020-0190(89)90155-5. URL `http://www.sciencedirect.com/science/article/pii/0020019089901555`
3. Edelsbrunner, H., Maurer, H.A., Preparata, F.P., Rosenberg, A.L., Welzl, E., Wood, D.: Stabbing line segments. BIT Numerical Mathematics **22**(3), 274–281 (1982). DOI 10.1007/BF01934440. URL `https://doi.org/10.1007/BF01934440`
4. Matousek, J., Gärtner, B.: Understanding and using linear programming. Springer Science & Business Media (2007)
5. Megiddo, N.: Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. SIAM Journal on Computing **12**(4), 759–776 (1983). DOI 10.1137/0212052. URL `https://doi.org/10.1137/0212052`
6. Megiddo, N.: Linear programming in linear time when the dimension is fixed. J. ACM **31**(1), 114–127 (1984). DOI 10.1145/2422.322418. URL `http://doi.acm.org/10.1145/2422.322418`
7. O'Rourke, J., Rao Kosaraju, S., Megiddo, N.: Computing circular separability. Discrete & Computational Geometry **1**(2), 105–113 (1986). DOI 10.1007/BF02187688. URL `https://doi.org/10.1007/BF02187688`
8. Rappaport, D.: A convex hull algorithm for discs, and applications. Computational Geometry **1**(3), 171 – 187 (1992). DOI https://doi.org/10.1016/0925-7721(92)90015-K. URL `http://www.sciencedirect.com/science/article/pii/092577219290015K`