# Separating points by axis-parallel lines

Gruia Călinescu *  Adrian Dumitrescu†  Peng-Jun Wan*


## Abstract

We study the problem of separating $n$ points in the plane, no two of which have the same $x$ or $y$-coordinate using a minimum number of vertical and horizontal lines avoiding the points, so that each cell of the subdivision contains at most one point. We prove that this problem and some variants of it are NP-complete. We give an approximation algorithm with ratio 2 for the planar problem, and a ratio $d$ approximation algorithm for the $d$-dimensional variant, in which the points are to be separated using axis-parallel hyperplanes. We reduce the problem to the rectangle stabbing problem studied by Gaur *et al* [4]. Their approximation algorithm uses LP-rounding. Our algorithm presents an alternative LP-rounding procedure which also works for the rectangle stabbing problem.


## 1 Introduction

Let $P$ be a set of $n$ points in the plane, no two of which have the same $x$ or $y$-coordinate. We consider the problem of finding a minimum set of axis-parallel lines that do not pass through any of the given points, such that each cell of the resulting subdivision contains at most one point. In other words, for each pair of points there is a line in our set which separates the two points. We refer to this problem as the *separation problem* SEPARATION. Its natural extension in higher dimensions, called the multi-modal sensor allocation problem in [9], asks for a minimum cardinality set of hyperplanes which separate $n$ given points. It has applications to fault-tolerant multi-modal sensor fusion in the context of embedded sensor networks [9]. It is also a natural problem to consider, from the perspective of computational geometry, and appears to be closely related to other problems of separating points or hitting objects studied recently in the CG literature [1, 2, 5, 6, 7].

When the number of dimensions is part of the input, the separation problem has been shown to be NP-complete [8]. However the proof of this result does not carry over to the case when the number of dimensions is small (e.g., in the plane). We prove that the separation problem is NP-complete. Our proof can be adapted to show that other variants of the problem are NP-complete as well (see below). We present two LP-based approximation algorithms with ratio 2 in the plane, respectively $d$ in $\mathbb{R}^d$: the first is obtained by casting the separation problem as a special case of the rectangle stabbing problem [5, 4] (see Section 2). The second uses a different — counting based — rounding procedure. We show that the second algorithm also works for the rectangle stabbing problem, with the same ratio 2. We exhibit an infinite sequence of examples in the plane having integrality gap $3/2$, for both problems. Our main result is

**Theorem 1** *There exists a ratio 2 approximation algorithm for the separation problem in the plane. The above problem is NP-complete. Moreover, assuming P $\neq$ NP, there is an absolute constant $\epsilon_S > 0$ such that no polynomial time algorithm has approximation guarantee $1 + \epsilon_S$.*

A natural variant of the above point separation problem is a *colored* version: the points are colored, and one has to find a minimum set of axis-parallel lines, such that the set of points (if non-empty) in each cell of the resulting subdivision is monochromatic. Clearly having each point colored by a different color is equivalent to the original problem. Thus when the numbers of colors is part of the input this problem is also NP-complete. We prove that it remains so for any number $k$ of colors, $k \geq 2$. This version also extends to higher dimensions, as the original problem does. Both our algorithms can be used to obtain a 2-approximate solution for the colored version in the plane, or $d$-approximate solutions for the colored version in $\mathbb{R}^d$.

## 2 Algorithms for the separation problem in the plane

Without loss of generality, we can restrict the set of vertical or horizontal separating lines to a set $\mathcal{L}$ of $2(n-1)$ *canonical lines*, one for each pair of consecutive points with respect to the $x$-coordinate, and one for each pair of consecutive points with respect to the $y$-coordinate (say, at the average coordinate value of two consecutive points).

We first give two lower bounds on $OPT$, the size of an optimal solution. Consider the complete geometric graph $G = (V, E)$ whose vertex set is the set $P$ of $n$ points. We say that two edges of $G$ are *independent*, if there is no vertical or horizontal line that intersects both in their interior. Let $I$ be a maximum independent set of edges of $G$. Then clearly, $OPT \geq |I|$, since each edge of $I$ requires a distinct separating line.

Put $OPT = l$. The maximum number of cells induced by $l$ lines is attained when the lines are divided evenly into vertical and horizontal. Since each point requires a distinct cell of

*Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA; {calinesc,wan}@cs.iit.edu

†Computer Science, University of Wisconsin–Milwaukee, 3200 N. Cramer Street, Milwaukee, WI 53211, USA; ad@cs.uwm.edu

the arrangement of $l$ lines, we have $(\lfloor l/2 \rfloor + 1)(\lceil l/2 \rceil + 1) \geq n$, which implies

$$OPT \geq \lceil 2\sqrt{n} \rceil - 2. \quad (1)$$

In the *rectangle stabbing problem* [5, 4], we are given a set of (nondegenerate) axis-parallel rectangles in the plane, with the objective of stabbing all the rectangles with the minimum number of axis-parallel lines (a rectangle is said to be stabbed by line $\ell$, if $\ell$ intersects its interior). Gaur, Ibaraki and Krishnamurti have recently given a ratio 2 approximation algorithm for this problem [4].

Let us first see how the separation problem can be cast as a rectangle stabbing problem. For each pair of points $u, v \in P$, consider the rectangle $R_{uv}$, whose diagonal is $uv$. Then separating all the points in $P$ is equivalent to stabbing all rectangles $R_{uv}$, with $u, v \in P$. Note also that it is enough to restrict ourselves to empty rectangles, i.e., those that do not contain other points of $P$: stabbing all empty rectangles $R_{uv}$ guarantees that all rectangles are stabbed. However, in general this restriction may be not significant, as it is easy to construct examples with $\Omega(n^2)$ empty rectangles determined by the $n$ points.

Let $\mathcal{R}$ be the collection of rectangles in the rectangle stabbing problem. A set $\mathcal{L}$ of canonical lines is selected first, as in the separation problem (see [4] for details regarding this selection). The natural IP (integer program) with variables $X_L$, for $L \in \mathcal{L}$ is

$$\text{minimize} \sum_{L \in \mathcal{L}} x_L \quad (2)$$

$$\text{subject to} \sum_{L \text{ stabs } R} x_L \geq 1 \quad \forall R \in \mathcal{R} \quad (3)$$

$$x_L \in \{0, 1\} \quad \forall L \in \mathcal{L}. \quad (4)$$

The linear programming relaxation of IP is obtained by replacing the constraints $(4)$ by

$$x_L \geq 0 \quad \forall L \in \mathcal{L}. \quad (5)$$

Denote by $LP$ the value of the linear program in $(2)$. The algorithm of Gaur *et al.* solves the linear program and classifies rectangles as horizontal or vertical (with ties broken arbitrarily), depending on whether

$$\sum_{\substack{L:\text{ horizontal} \\ L \text{ stabs } R}} x_L \geq \frac{1}{2} \quad \text{or} \quad \sum_{\substack{L:\text{ vertical} \\ L \text{ stabs } R}} x_L \geq \frac{1}{2}.$$

It then solves optimally the problem of stabbing the horizontal rectangles by horizontal lines, and that of stabbing the vertical rectangles by vertical lines, by solving the corresponding linear programs $LP_H$ and $LP_V$. The solutions of these two linear programs are integral, a property that follows from the total unimodularity of their system matrices. Putting together the two sets of lines results in a 2-approximation algorithm, using again the total unimodularity property. We remark here, that instead of solving $LP_H$ and $LP_V$, one can solve directly the corresponding stabbing problems using the greedy algorithm, since these become interval stabbing problems on the line.

The formulation of the integer and linear programs for the separation problem is analogous. The IP with variables $X_L$, for $L \in \mathcal{L}$ is

$$\text{minimize} \sum_{L \in \mathcal{L}} x_L \quad (6)$$

$$\text{subject to} \sum_{L \text{ separates } uv} x_L \geq 1 \quad \forall (u, v) \quad (7)$$

$$x_L \in \{0, 1\} \quad \forall L \in \mathcal{L}. \quad (8)$$

The linear programming relaxation of IP is obtained by replacing the constraints $(8)$ by

$$x_L \geq 0 \quad \forall L \in \mathcal{L}. \quad (9)$$

The 2-approximate solution is obtained in the same way.

We now provide a new, conceptually simpler, LP-based algorithm that only solves the linear program $(6), (7), (9)$, and directly rounds the solution. Go through the horizontal lines in order of their $y$-coordinates, adding up their fractions. Whenever the total reaches $1/2$, pick that line, reset the total to $0$, and keep going. Do the same with the vertical lines. The picked lines cannot miss any edge! Since $LP \leq OPT$, the approximation ratio is $2$. It is easy to see that this algorithm works for the rectangle stabbing problem as well, with the same ratio of 2.

Let $\mathcal{R}$ be the set of empty rectangles $R_{uv}$. We have $|\mathcal{R}| = \Omega(n)$ and $|\mathcal{R}| = O(n^2)$. Denote by $\mathcal{R}_u$ the set of empty rectangles $R_{uv}$, where $u$ lies to the left of $v$. $\mathcal{R}$ can be computed in $O(n^2)$ time, by computing $\mathcal{R}_u$ in $O(n)$ time for each $u \in P$. The details are omitted.

We finally remark that both algorithms can be used to solve the colored version of the separation problem in the plane with the same ratio of 2: write linear constraints only for the set of bichromatic edges, i.e., those whose endpoints have different colors.

### 2.1 Integrality gap

We now show an infinite sequence of examples in the plane having integrality gap $3/2$, for both the rectangle stabbing, and the separation problem. It is enough to do this for the separation problem (as a special case of the rectangle stabbing problem).

**Lemma 2** *The integrality gap of the linear program* $(6), (7), (9)$ *is* $3/2$ *for an infinite class of examples.*

**Proof.** Consider the five-point configuration in Fig. 1 (left), that we call an $X$. The points can be fractionally separated with weights $1/2$ on each of the four canonical lines shown in the figure. Thus $LP \leq 4/2 = 2$. Using the trivial lower bound (1) (or by inspection) gives $OPT \geq \lceil 2\sqrt{5} \rceil - 2 = 3$, and it is easy to see that this is tight.

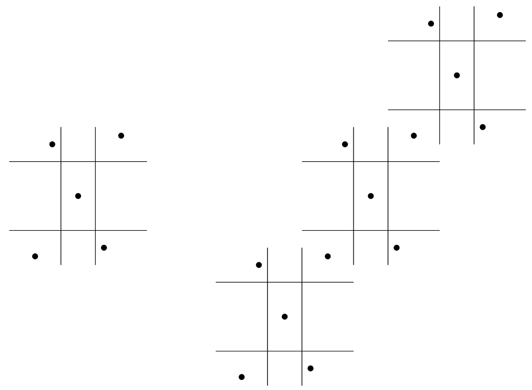

Figure 1: A class of examples with integrality gap $3/2$.

By repeating $k$ times the $X$ diagonally, such that two adjacent $X$s share one point, we obtain a configuration with $4k+1$ points, as in Fig. 1 (right), for $k=3$. One can think of the points being placed on an (infinite) chessboard. Observe that in each row or column of the board the points have increasing $x$ and $y$-coordinates. Again, the points can be fractionally separated with weights $1/2$ on each of the canonical lines shown in the figure. Thus $LP \leq 4k/2 = 2k$. To separate the points of each $X$ requires three lines, and since the points have increasing $x$ and $y$-coordinates in each row or column, no line used to separate one $X$ is of any help in separating other $X$s, thus $OPT \geq 3k$. It is easy to see that $3k$ lines are also enough, and the lemma follows. $\square$

## 3 Hardness results

### 3.1 NP-completeness

The decision version of the separation problem is clearly in NP, so we only have to prove its NP-hardness. Inspired by the reduction from Proposition 6.2 of [5], we reduce the satisfiability problem 3-SAT to the separation problem in the plane (SEPARATION). The input to 3-SAT is a boolean formula $\phi$ in 3-CNF form, i.e., each clause has exactly three literals. The problem asks whether $\phi$ is satisfiable. 3-SAT is known to be NP-complete [3]. Let $\phi$ have $n$ variables and $m$ clauses. The reduction constructs a set $P_\phi$ of $4n+12m+2$ points in the plane, no two of which have the same $x$ or $y$-coordinate. The construction is illustrated in Figure 2 for $\phi = (t+y+\overline{z})(\overline{x}+\overline{y}+z)(\overline{x}+\overline{y}+\overline{z})$. Here $n=4$ and $m=3$; the three clauses are denoted $C_1, C_2, C_3$.

There are three types of points: *variable* points, *clause* points and *control* points. The control points come in pairs and have increasing $y$-coordinates when scanned from left to right: denoted $q_1, \ldots, q_{4n+2m+2}$. For $1 \leq i \leq n+1$, the pair $q_{2i-1}, q_{2i}$ "forces" a horizontal line (which is more useful than the vertical line separating the pair), and for $n+2 \leq i \leq 2n+m+1$, the pair $q_{2i-1}, q_{2i}$ "forces" a vertical line. We call these lines *grid lines*, and we denote by $h$ the lowest horizontal grid line. There are three variable points

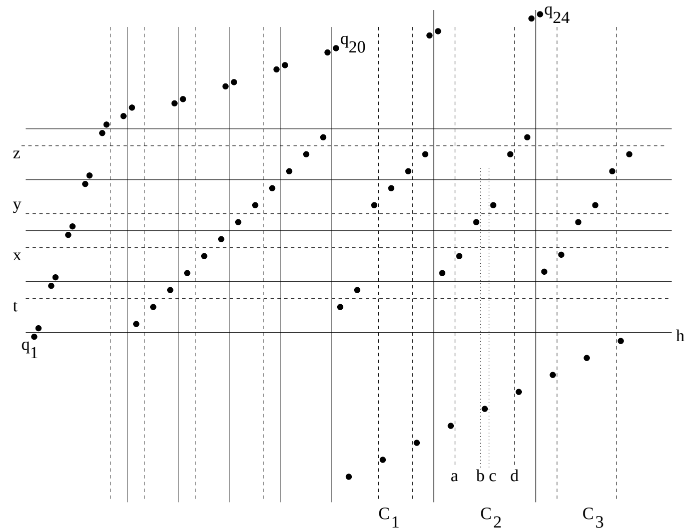


Figure 2: The point set $P_\phi$ corresponding to $\phi = (t+y+\overline{z})(\overline{x}+\overline{y}+z)(\overline{x}+\overline{y}+\overline{z})$. The solution (set of separating lines) corresponding to the truth assignment $t=1, x=1, y=0, z=1$ is shown; the grid lines are solid, while the other separating lines are dashed.

for each variable, and nine clause points for each clause. The nine points of each clause $C$ are made up of six points that appear in the rows of the variables that appear in $C$ (above the horizontal line $h$), and three points below $h$. We have a pair of points in the grid cell given by each variable-clause pair $(x, C)$, where the variable $x$ appears in $C$, thus six points per clause above line $h$. The three points of each variable require two separating lines. Every optimal solution can be assumed to use exactly one vertical line, as one vertical line also separates two control points and a second one is not needed. The choice of the higher (resp. lower) horizontal line corresponds to setting of the variable to true (resp. false). If $x$ appears unnegated in $C$, the pair of points is separated by the higher horizontal line, whereas if $x$ appears negated in $C$, the pair of points is separated by the lower horizontal line.

Clearly, constructing $P_\phi$ can be accomplished in polynomial time. The result follows once we establish the following claim (proof omitted).

**Claim 1** *$\phi$ is satisfiable if and only if $P_\phi$ can be separated using $4n+3m+2$ lines.*

We can use the same reduction to show that the separation problem with colored points is also NP-complete. The 2-coloring that we use has the property that all the edges specified in the above proof are bichromatic (i.e., their endpoints have different colors). We omit the details for lack of space. We thus have

**Corollary 1** *The separation problem in the plane with colored points is NP-complete.*

### 3.2 APX-hardness

The *maximum* 3-*satisfiability* problem MAX-3SAT is that of finding a truth value assignment which satisfies the maximum number of clauses in a Boolean formula in 3-CNF form. For each fixed $k$, define MAX-3SAT($k$) to be the restriction of MAX-3SAT to Boolean formulae in which each variable occurs at most $k$ times. Theorem 3 below is immediate from Theorems 29.7, 29.11, and Corollary 29.8 in [10].

**Theorem 3** [10] *Assuming P ≠ NP, there is an absolute constant $\epsilon_M > 0$ such that no polynomial time algorithm for MAX-3SAT(5) satisfies at least $(1 - \epsilon_M)m$ clauses for every satisfiable formula $\phi$ with $m$ clauses.*

To prove the approximation hardness stated in Theorem 1, we use the same reduction, and Theorem 3. Calculations show that one can take $\epsilon_S = \epsilon_M/75$. We omit the details.

## 4 Remarks

### 4.1 Higher dimensions

Following [4], it is now straightforward to observe that both our algorithms yield a factor $d$ approximation for the separation problem in $\mathbb{R}^d$. This holds for the colored version as well. One has to replace $1/2$ with $1/d$ in the corresponding places. In the first phase, after solving the linear program, edges are classified into $d$ types, depending on the coordinate for which the sum of fractional weights is at least $1/d$. In the second phase, the first algorithm solves $d$ linear programs (as in [4]), or solves $d$ interval stabbing problems on the line (as in Section 2). The second algorithm cycles through all coordinates and for each coordinate, goes through the hyperplanes in order, and picks a hyperplane when the total sum of the fractional weights reaches $1/d$. The total is then reset to 0, and the process continues.

### 4.2 Concluding remarks

Several interesting questions regarding the separation problem in the plane remain, such as: Is it possible to improve the ratio 2 approximation? Do special cases, e.g., points in convex position, admit better approximation ratios, or even exact solutions? Are there planar examples having integrality gap larger than $3/2$? We have examples for which every LP solution is not half integral. Using rounding, one can obtain a ratio $3/2$ approximation for instances whose LP solution is half integral. One can potentially strengthen the LP by adding constraints requiring that each triplet of points is fractionally separated by at least 2, but we did not find yet any benefit in doing that.

## References


[1] E. Arkin, F. Hurtado, J. S. B. Mitchell, C. Seara and S. Skiena, Some separability problems in the plane, *Proceedings of the 16-th European Workshop on Computational Geometry*, 2000, 51–54.

[2] O. Devillers, F. Hurtado, M. Mora and C. Seara, Separating several point sets in the plane, *Proceedings of the 13th Canadian Conference on Computational Geometry*, (CCCG '01), 2001, 81–84.

[3] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.*, W.H. Freeman and Company, 1979.

[4] D. R. Gaur, T. Ibaraki and R. Krishnamurti, Constant ratio approximation algorithm for the rectangle stabbing problem and the rectilinear partitioning problem, *Journal of Algorithms*, **43** (2002), 138–152.

[5] R. Hassin and N. Megiddo, Approximation algorithms for hitting objects with straight lines, *Discrete Applied Mathematics*, **30(1)** (1991), 29–42.

[6] F. Hurtado, M. Mora, P. A. Ramos and C. Seara, Two problems on separability with lines and polygonals, *Proceedings of the 15-th European Workshop on Computational Geometry*, 1999, 33–35.

[7] F. Hurtado, M. Noy, P. A. Ramos and C. Seara, Separating objects in the plane with wedges and strips, *Discrete Applied Mathematics*, **109** (2001), 109–138.

[8] F. Koushanfar, M. Potkonjak and A. Sangiovanni-Vincentelli, Fault-tolerance techniques in wireless ad-hoc sensor networks, UC Berkeley technical reports, 2002.

[9] F. Koushanfar, S. Slijepcevic, M. Potkonjak and A. Sangiovanni-Vincentelli, Error-tolerant multi-modal sensor fusion, *IEEE CAS Workshop on Wireless Communication and Networking*, Pasadena, 2002.

[10] V. Vazirani, *Approximation Algorithms*, Springer-Verlag, New York, 2001.