Geometric Hitting Set for Segments of Few Orientations

Sándor P. Fekete[1], Kan Huang[2], Joseph S.B. Mitchell[2], Ojas Parekh[3], and Cynthia A. Phillips[3](✉)

[1] TU Braunschweig, Braunschweig, Germany
s.fekete@tu-bs.de

[2] Stony Brook University, Stony Brook, NY, USA
{khuang,jsbm}@ams.sunysb.edu

[3] Sandia National Labs, Albuquerque, NM, USA
{odparek,caphill}@sandia.gov

Abstract. We study several natural instances of the geometric hitting set problem for input consisting of sets of line segments (and rays, lines) having a small number of distinct slopes. These problems model path monitoring (e.g., on road networks) using the fewest sensors (the "hitting points"). We give approximation algorithms for cases including (i) lines of 3 slopes in the plane, (ii) vertical lines and horizontal segments, (iii) pairs of horizontal/vertical segments. We give hardness and hardness of approximation results for these problems. We prove that the hitting set problem for vertical lines and horizontal rays is polynomially solvable.

1 Introduction

The set cover problem is fundamental in combinatorial optimization. It is NP-hard and has an $O(\log n)$-approximation algorithm, which is best possible (unless $P = NP$, [13]). Equivalently, set cover can be cast as a hitting set problem: given a collection, $\mathcal{C}$, of subsets of set U, find a smallest cardinality set $H \subseteq U$ such that every set in $\mathcal{C}$ contains at least one element of H. Numerous special instances of set cover/hitting set have been studied. Our focus in this paper is on geometric instances that arise in covering (hitting) sets of (possibly overlapping) line segments using the fewest points ("hit points"). A closely related problem is the "Guarding a Set of Segments" (GSS) problem [3,5,6,25], in which the segments may cross arbitrarily, but do not overlap. Since this problem is strongly NP-complete [5] in general, our focus is on special cases, primarily those in which the segments come from a small number of orientations (e.g., horizontal, vertical). We provide several new results on hardness and approximation algorithms.

We also are motivated by the path monitoring problem: given a set of trajectories, each a path of line segments in the plane, place the fewest sensors (points) to observe (hit) all trajectories. To gain theoretical insight into this challenging problem, we examine cleaner, but progressively harder, versions of hitting trajectory/line-like objects with points. If the trajectories are on a Manhattan road network, the paths are (possibly overlapping) horizontal/vertical segments.

L. Sanità and M. Skutella (Eds.): WAOA 2015, LNCS 9499, pp. 145–157, 2015.
DOI: 10.1007/978-3-319-28684-6_13

Alternatively, one wishes to place the fewest vendors or service stations in a road network to service a set of customer trajectories.

Our Results. We give complexity and approximation results for several geometric hitting set problems on inputs S of line "segments" of special classes, mostly of fixed orientations. The segments are allowed to overlap arbitrarily. We consider various cases of "segments" that may be bounded (line segments), semi-infinite (rays), or unbounded in both directions (lines). Our results are:

(1) Hitting lines of 3 slopes in the plane is NP-hard (greedy is optimal for 2 slopes). For set cover with set size at most 3, standard analysis of the greedy algorithm gives an approximation factor of $H(3) = 1+(1/2)+(1/3) = (11/6)$, and there is a 4/3-approximation based on semi-local optimization [15]. We prove that the greedy algorithm in this special geometric case is a (7/5)-approximation.
(2) Hitting vertical lines and horizontal rays is polytime solvable.
(3) Hitting vertical lines and horizontal (even unit-length) segments is NP-hard. Our proof shows hitting horizontal and vertical unit-length segments is also NP-hard. We prove APX-hardness for hitting horizontal and vertical segments.
(4) Hitting vertical lines and horizontal segments has a (5/3)-approximation algorithm. (This problem has a straightforward 2-approximation).
(5) Hitting pairs of horizontal/vertical segments has a 4-approximation. Hitting pairs having one vertical and one horizontal segment has a (10/3)-approximation. These results are based on LP-rounding. More generally, hitting sets of k segments from r orientations has a $(k \cdot r)$-approximation algorithm.
(6) We give (in the full paper) a linear-time combinatorial 3-approximation algorithm for hitting triangle-free sets of (non-overlapping) segments. A 3-approximation for this version of GSS was recently given [25] using linear programming.

Related Work. There is a wealth of related work on geometric set cover and hitting set problems; we do not attempt here to give an exhaustive survey. The *point line cover* (PLC) problem (see [23,27]) asks for a smallest set of lines to cover a given set of points; it is equivalent, via point-line duality, to the hitting problem for a set of lines. The PLC (and thus the hitting problem for lines) was shown to be NP-hard [30]; in fact, it is APX-hard [7] and Max-SNP Hard [28]. The problem has an $O(\log OPT)$-approximation (e.g., greedy – see [26]); in fact, the greedy algorithm for PLC has worst-case performance ratio $\Omega(\log n)$ [16].

Hassin and Megiddo [22] considered hitting geometric objects with the fewest lines having a small number of distinct slopes. They observed that, even for covering with axis-parallel lines, the greedy algorithm has an approximation ratio that grows logarithmically. They gave approximations for the problem of hitting horizontal/vertical segments with the fewest axis-parallel lines (and, more generally, with lines of a few slopes). Gaur and Bhattacharya [19] consider covering points with axis-parallel lines in d-dimensions; they give a $(d-1)$-approximation based on rounding the corresponding linear program (LP) formulation. Many other stabbing problems (find a small set of lines that stab a given set of objects) have been studied; see, e.g., [14,17,20,21,26,29].

A recent paper [25] gives a 3-approximation for hitting sets of "triangle-free"segments. Brimkov et al. [3,5,6] have studied the hitting set problem on line segments, including various special cases; they refer to the problem as "Guarding a Set of Segments", or GSS. GSS is a special case of the "art gallery problem:" place a small number of "guards" (e.g., points) so that every point within a geometric domain is "seen" by at least one guard [32,34]. Brimkov et al. [4] provide experimental results for three GSS heuristics, including two variants of "greedy," showing that in practice the algorithms perform well and are often optimal or very close to optimal. They prove, however, that, in theory, the methods do not provide worst-case constant-factor approximation bounds. For the special case that the segments are "almost tree (1)" (a connected graph is an *almost tree (k)* if each biconnected component has at most k edges not in a spanning tree of the component), a $(2 - \varepsilon)$-approximation is known [3].

An important distinction between GSS and our problems is that allow *overlapping* (or partially overlapping) segments (rays, and lines), while, in GSS, each line segment is maximal in the input set of line segments (the union of two distinct input segments is not a segment). A special case of our problem is *interval stabbing* on a line: Given a set of segments (intervals), arbitrarily overlapping on a line, find a smallest hitting set of points that hit all segments. A simple sweep along the line solves this problem optimally: when a segment ends, place a point and remove all segments covered by that point.

If no point lies within three or more objects, then the hitting set problem is an edge cover problem in the intersection graph of the objects. In particular, if no three segments pass through a common point, the problem can be solved optimally in polynomial time. (This implies that in an arrangement of "random" segments, the GSS problem is almost surely polynomially solvable; see [3]).

Hitting axis-aligned rectangles is related to hitting horizontal and vertical segments. Aronov, Ezra, and Sharir [2] provide an $O(\log\log OPT)$-approximation for hitting set for axis-aligned rectangles (and axis-aligned boxes in 3D), by proving a bound of $O(\varepsilon^{-1}\log\log(\varepsilon^{-1}))$ on the ε-net size of the corresponding range space. The connection between hitting sets and ε-nets [8,11,12,18] implies a c-approximation for hitting set if one can compute an ε-net of size c/ε; recent major advances [1,33] on lower bounds on ε-nets imply that associated range spaces (rectangles and points, lines and points, points and rectangles) have ε-nets of size superlinear in $1/\varepsilon$. Remarkably, improved $(1+\varepsilon)$-approximation algorithms (i.e., PTASs) for certain geometric hitting set and set cover problems are possible with simple local search. For example, Mustafa and Ray [31] give a local search PTAS for computing a smallest subset, of a given set of disks, that covers a given set of points. Hochbaum and Maas [24] used grid shifting to obtain a much earlier PTAS for the minimum unit disk cover problem when disks can be placed anywhere in the plane, not restricted to a discrete input set.

2 Hitting Segments

Suppose S is a set of line segments in the plane. If all segments are horizontal, then we can compute an optimal hitting set by independently solving the interval stabbing problem along each of the horizontal lines determined by the input.

If the segments are of two different orientations (slopes), then the problem becomes significantly harder. Without loss of generality, assume the segments are horizontal and vertical. We show the problem is hard even if the axis-parallel segments are all the same length. This result (Corollary 1) is a consequence of an even stronger result, Theorem 4, which we establish in Sect. 5.

We get an immediate 2-approximation algorithm by solving optimally each of the two orientations, and using the union of the hitting points for both. (This generalizes to a k-approximation for hitting sets of segments of k orientations).

3 Hitting Lines

When S is a set of n lines in the plane, greedy gives an $O(\log OPT)$ approximation factor; any approximation factor better than logarithmic would be quite interesting. (See [16,27].) If the lines have only 2 slopes, then greedy is optimal.

3.1 Hardness of Hitting Lines of 3 Slopes in 2D

We prove that the hitting set problem is NP-hard when lines have more than two orientations. Consider the dual formulation: (3-Slope-Line-Cover, 3SLC) Find a minimum-cardinality set of non-vertical lines to cover a set P of points (duals to the set S of lines), which are known to lie on three vertical lines.

We prove (in the full paper) that 3SLC is NP-hard from 3-SAT, using variable gadgets and clause gadgets that rely on carefully placed points on three vertical lines. "Propagation" of variable assignments is determined by triples of points on distinct vertical lines coverable by a single line.

Theorem 1. *The problem 3SLC is NP-complete.*

3.2 Analysis of the Greedy Hitting Set Algorithm for Lines of 3 Slopes in 2D

If no point lies in more than k sets, the greedy algorithm's approximation factor is $H(k) = \sum_{i=1}^{k}(1/i)$ [10]. This property holds for lines of 3 slopes with $k = 3$, giving a greedy approximation factor $H(3) = 11/6$. We give a new analysis, exploiting the special geometric structure of the hitting set problem for lines of 3 slopes, to obtain an approximation factor (7/5); see the full paper.

3.3 Axis-Parallel Lines in 3D

While in 2D the hitting set problem for axis-parallel lines is easily solved, in 3D we prove (in the full paper) that the corresponding hitting set problem is NP-hard, using a reduction from 3-SAT.

Theorem 2. *Hitting set for axis-parallel lines in 3D is NP-complete.*

4 Hitting Rays and Lines

Hitting rays is “harder” than hitting lines, since any instance of hitting lines has a corresponding equivalent instance as a hitting rays problem (place the apices of the rays far enough away that they are effectively lines). Unlike lines, there can be many different collinear rays. Divide collinear rays into two groups according to the direction they point along the containing line, ℓ; because of nesting, we need keep only one of the rays pointing in each of the two directions along ℓ.

We show that the special case with horizontal rays and vertical lines (abbreviated HRVL) is exactly solvable in polynomial time:

Theorem 3. *The hitting set problem for vertical lines and horizontal rays can be solved in $O(nT)$ time, where n is the number of entities and T is the time for computing a maximum matching in a bipartite graph with n nodes.*

We begin with a high-level overview of the algorithm. A point can cover at most 3 objects: a vertical line, a left-facing ray, and a right-facing ray. This requires the two rays to intersect in a segment, and the vertical line to intersect this segment. We call these points 3-hitters. We can compute the maximum possible number of 3-hitters via maximum matching in a bipartite graph, where edges represent intersections between vertical lines and horizontal segments. We prove there exists an optimal solution with this maximum number of 3-hitters. The algorithm performs a sweep inward from the left and right, finding a suitable set of 3-hitters, ensuring the remaining lines have the best possible chance to share a point with the remaining rays. Once everything that is 3-hit is removed, the remaining objects intersect in at most pairs. So we can finish the hitting by solving an edge cover problem between rays and lines. We prove this is optimal.

We now give additional algorithmic and proof details. We call a horizontal ray to the left (resp., right) an *l-ray* (resp. *r-ray*). In this section, all lines are vertical. If two collinear rays are disjoint, we shift one ray slightly up or down, so no two disjoint rays are collinear. These rays cannot be covered by a single point, so this does not fundamentally alter the optimal solution.

If a line only contains one ray, we add a ray to pair with it. For example, if an r-ray intersects no l-ray, we add an intersecting l-ray whose right endpoint is to the right of all vertical input lines. This additional ray won’t change the optimal solution. If an l-ray and r-ray intersect, their intersection is a segment. Since all rays intersect another ray, we represent each pair of rays by their segment.

Let H and V denote the number of segments and lines respectively. Any solution requires V points to cover the lines. Those points can help “hit” segments in two possible ways: (1) Place a point on the segment. We call the corresponding line a 3-hitter and say the segment is 3-hit by the line. (2) Hit each ray outside its intersecting segment. This requires two points. We call the left(right) line an l-hitter(r-hitter). We say the segment is *double-hit* by those two lines.

Let v_1 and v_2 be the number of segments hit by the V points in the first and second ways respectively. Then the number of points in the solution is $H + V - v_1 - v_2$. We must put a point on each line to hit as many segments as possible.

Given an instance of HRVL, we can calculate the maximum number of 3-hitters. We construct a bipartite graph G where one set of nodes is the lines and the other set of nodes is the segments. There is an edge between two nodes if and only if the line and segment they represent intersect. Maximum matching in a bipartite graph is solvable in polynomial time. A matching in the graph represents a set of independent intersections in the corresponding HRVL. That is, a set of M edges in a matching corresponds to a way to cover M segments and M lines with M points. These are coverages of type 1. The following intuitive lemma shows it is better to adopt the first way to hit segment.

Lemma 1. *For any instance of HRVL, there is a maximum matching between lines and segments that can be augmented to be an optimal solution.*

Proof Sketch. We use contradiction. Let v_1^* be the largest v_1 for any minimum hitting set. We assume that v_1^* is less than m, the cardinality of the maximum matching between lines and segments. Thus, there is an augmenting path in the bipartite graph G, such as the green path in Fig. 1. Because the current solution is optimal, any augmenting path cannot improve it. This allows us to infer some properties of the first segment and the last line on the augmenting path. We consider the augmenting path P with the shortest length and the shortest horizontal distance between the last two lines. Then by case analysis on path P, we argue there exists another augmenting path that increases v_1^* or violates a minimality condition of P. The proof appears in the full paper.

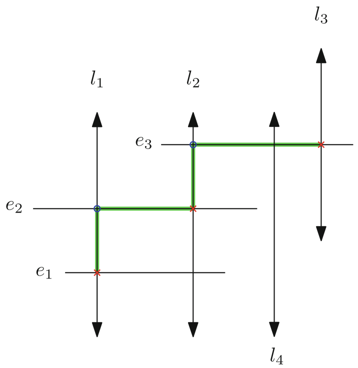

Fig. 1. A green augmenting path: the matching size increases by replacing blue circles with red crosses (Color figure online).

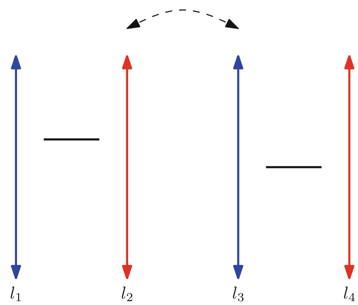

Fig. 2. Swapping l_2 and l_3 makes both of them more useful (Color figure online).

Lemma 2. *Given an optimal solution $\mathcal{S}$, there is an optimal solution $\mathcal{S}'$ that has the same set of 3-hitters as $\mathcal{S}$, with its l-hitters all left of its r-hitters.*

Proof. In Fig. 2 two segments are double-hit by two pairs of lines; the blue lines are l-hitters and the red lines are r-hitters. When we pair l_1 to l_3 and pair l_2 to l_4, the two segments are still double-hit, because this swap moves the l-hitter further left and the r-hitter further right. A sequence of such swaps moves all l-hitters to the left of all r-hitters. □

In the full paper, we give details of an algorithm for HRVL. The algorithm maximizes the number of 3-intersections and "balances" the remaining lines between the left and right sides as much as possible. In the algorithm, we test the criticality of a line: given the previous choices, if a *critical* line is not used as a 3-hitter, there is no way to extend the previous choices to a maximum matching.

We now argue that the left-right-balanced approach gives lines that obey Lemma 2. Let S be the solution given by our HRVL algorithm, and let S' be an optimal solution with the maximum set of 3-hitters. We know that S and S' have the same number of 3-hitters. Let D and D' denote the lines left behind (not 3-hitters) in S and S' respectively. We order lines in D and D' from left to right. Let k be $\lfloor \frac{|D|}{2} \rfloor$. Thus, there are at most k pairs of double-hitters in S and S'. Let lh_i (resp., lh'_i) be the ith line of D (resp., D').

Given a solution P and a line l, let $E(l, P)$ denote the number of segments on the left side of l not hit by 3-hitters in P. A line having more segments on its right side is more likely to be an l-hitter. We will show that line lh_i is at least as capable of being an l-hitter as is line lh'_i.

$$E(lh_i, S) \leq E(lh'_i, S'), \quad i = 1, 2, .., k \tag{1}$$

We split the proof of (1) into two lemmas; proofs appear in the full paper.

Lemma 3. *lh_i cannot be on the right side of lh'_i, $i = 1, 2, .., k$.*

An immediate result from this lemma is

$$E(lh_i, S') \leq E(lh'_i, S'). \tag{2}$$

Given a solution P and a line l, let $C(l, P)$ denote the number of segments on the left side of l that have been 3-hit in P. Let $N(l)$ be the total number of segments on the left of line l. The following lemma shows that the segments that S leaves to be used as 2-hitters are the segments that are easier to double-hit.

Lemma 4. *$C(lh_i, S) \geq C(lh_i, S')$, $i = 1, 2, .., k$.*

Therefore we obtain

$$\begin{aligned} E(lh_i, S) &= N(lh_i) - C(lh_i, S) \\ &\leq N(lh_i) - C(lh_i, S') = E(lh_i, S') \leq E(lh'_i, S'). \end{aligned}$$

.

5 Hitting Lines and Segments

5.1 Hardness

Theorem 4. *Hitting set for horizontal unit segments and vertical lines is NP-complete.*

Proof. The reduction is from 3SAT. See Fig. 3.

Each variable is represented by a collinear connected set of horizontal unit segments, and each clause is represented by a red vertical line that intersects appropriate pairs of horizontal variable segments (if that variable occurs in a clause) or just single segments (in case a variable does not occur in a clause). Setting appropriate parities for the literals in a clause is achieved by appropriate horizontal shifting of the segments, as shown in the figure. This results in a construction in which the only place where three of the elements (segments or lines) can be hit involves a vertical line representing a clause, corresponding

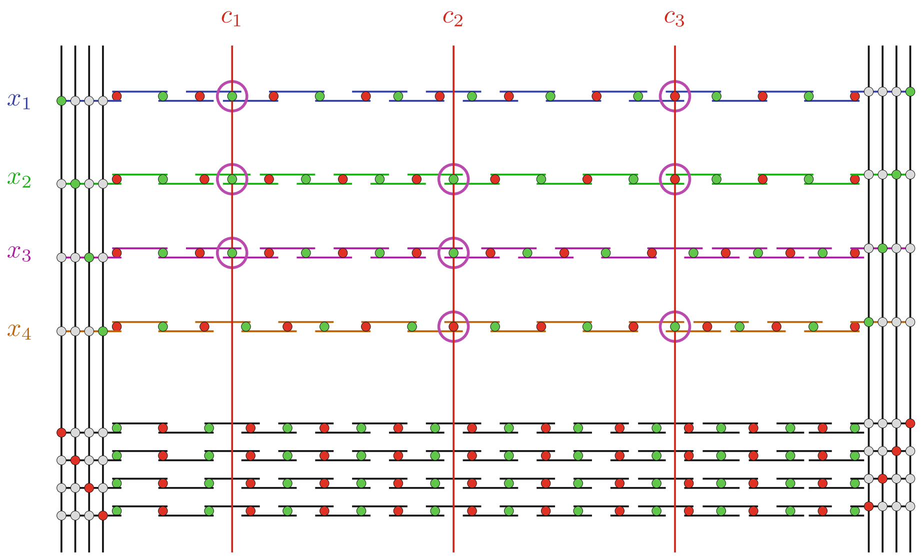

Fig. 3. A set of horizontal unit segments and vertical lines that represents the 3SAT instance $I = (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4)$. For better visibility, collinear segments are slightly shifted vertically, with red and green points indicating overlapping segments. In an optimal hitting set, the point covering a *labeled* horizontal segment induces a truth value for the corresponding variable: selecting one of its grey points (e.g., in the indicated green manner) assigns a value of "true"; selecting a red point, a value of "false". Overall, truth assignments for each variable correspond to a set of green or red points, respectively. Literals occurring in clauses are indicated by magenta circles; these are the only places where a point can hit three segments or lines at once (Color figure online).

to literals occurring in the respective clauses. (These are indicated by magenta circles in the figure). The N elements excluding the red vertical lines associated with clauses are called variable components.

We show that any feasible hitting set with exactly $N/2$ points induces a truth assignment and vice versa. There is no point that hits more than two of the variable components at once. Therefore, stabbing all N of them requires at least $N/2$ points, and any solution consisting of exactly $N/2$ points must hit each variable component exactly once. Consequently, the black vertical lines and the black collinear sets of connected horizontal segments in Fig. 3 may be reordered so that, without loss of generality, a solution with exactly $N/2$ points does not pick any of the gray points. Eliminating the gray points results in a natural partition of the instance into point-disjoint even-cardinality loops of variable components for each variable, where the points in each loop alternate between red and green. Thus any solution of size $N/2$ hitting the variable components must select all red or all green points from each variable's loop, corresponding to a truth assignment. We get an overall feasible hitting set if and only if the points also stab the vertical clause lines, corresponding to a satisfying truth assignment. □

After appropriate vertical scaling, we can replace the vertical lines by vertical unit segments, immediately giving the following corollary.

Corollary 1. *Deciding if there exists a set of k points in the plane that hit a given set S of unit-length axis-parallel segments is NP-complete.*

We show in the full paper, using a reduction from MAX-2SAT(3), that the hitting set problem is APX-hard for vertical lines and horizontal segments.

5.2 Approximation

We give a 5/3-approximation for hitting a set V of vertical lines and a set H of horizontal segments. We start by looking at the lower bounds: $v = |V|$ is the number of vertical lines. It is a lower bound. Let h be the lower bound on hitting horizontal segments only. We can compute h exactly; it is the minimum number of hit points for the horizontal segments (computed on each horizontal line). At any stage of the algorithm, we let h and v be the current values of these lower bounds for hitting the current (remaining unhit) sets H and V.

In stage 1, we place two kinds of points:

(a) We place hitting points on vertical lines that reduce h (and v) by one. These points are "maximally productive" since no single hitting point can do more than to reduce h and v each by one. As vertical lines are hit, we remove them from V. Similarly, as horizontal segments are hit, we remove them from H.
(b) Look for pairs (if any) of points, on the same horizontal line and on two vertical lines (from among the current set V), that decrease h by one.

Let k_1 and k_2 be the number of type(a) and type(b) points placed in this stage, respectively. Therefore, for the remaining instance, the lower bound h decreases by $k_1 + k_2/2$, and v decreases by $k_1 + k_2$.

In stage 2, we now have a set of vertical lines V and horizontal segments H such that no single point at the intersection of a vertical line and a horizontal segment (or segments) reduces h, and no pair of points on two distinct vertical lines reduces h.

Lemma 5. *For such sets V and H as in stage 2, an optimal hitting set has size at least $v + h$, where $v = |V|$ and h is the minimum number of points to hit H.*

Proof. The hit points we place on V (one per line) might conceivably decrease h. We claim that this cannot happen. Assume to the contrary that it happens. Let $\{q_1, q_2, \ldots, q_K\}$ be a minimum-cardinality set such that each of them is on some line of V from left to right and h is decreased by 1 after placing the set. Since the set is minimum, the points in it should be on a horizontal line L.

Since we have found all productive points and pairs of points in stage 1, K should be at least 3. Consider the hit point q_2. The segments on L that are not hit by q_2 are either completely left or right of q_2; let H_l and H_r be the corresponding sets. Points to the left of q_2 do not hit H_r, and points to the right of q_2 to not hit H_l. If adding q_1 decreases H, that means q_1 and q_2 is a productive pair, which should be found in stage 1; otherwise this means that the point q_1 is unnecessary, contradicting the minimality of K.

Theorem 5. *There is a polynomial-time $5/3$-approximation algorithm for geometric hitting set for a set of vertical lines and horizontal segments.*

Proof. The total number of points selected by our algorithm is $k_1 + k_2$ from the first stage and $h - k_1 - k_2/2 + v - k_1 - k_2$ from the second stage. By Lemma 5, the points chosen in stage 2 is a lower bound on the cost of an optimal solution:

$$h - k_1 - k_2/2 + v - k_1 - k_2 \leq OPT. \tag{3}$$

We also have $h \leq OPT$ and $v \leq OPT$. There are two cases.

(i) $k_1 + k_2 \leq 2/3 \cdot OPT$: In this case we select at most $2/3 \cdot OPT$ points in Stages 1, and we use (3) to bound the number of points selected in Stage 2. We conclude that our algorithm selects at most $5/3 \cdot OPT$ points.

(ii) $k_1 + k_2 > 2/3 \cdot OPT$: The total number of points selected by our algorithm is $h - k_1 - k_2/2 + v \leq 2 \cdot OPT - (k_1 + k_2/2)$. Since $k_1 + k_2/2 \geq k_1/2 + k_2/2 > 1/3 \cdot OPT$, we obtain a 5/3-approximation in this case as well. □

Theorem 6. *There is a polynomial-time $5/3$-approximation algorithm for geometric hitting set for a set of vertical (downward) rays and horizontal segments.*

Proof. The 2-stage approximation algorithm described above works for this case as well. The key observation is that among any set of collinear downward rays, we may remove all but the one with the lowest apex from the instance. Therefore after Stage 1, the hitting points we place on the rays not yet hit will not decrease h. The argument is analogous to that in Lemma 5.

6 Hitting Pairs of Segments

We consider now the hitting set problem for inputs that are *unions* of two segments, one horizontal and one vertical. While we are motivated by pairs (and larger sets) of segments that form paths, our methods apply to general pairs of segments, which might meet to form an "L" shape, a "+", or a "T" shape, or they may be disjoint. This hitting set problem is NP-hard, since it generalizes the case of horizontal and vertical segments.

Theorem 7. *For objects that are unions of a horizontal and a vertical segment, the hitting set problem has a polynomial-time 4-approximation.*

Proof Sketch. For ease of discussion, we call the union of two segments an "L." We use a method similar to those used in [9,20]. Solve the natural set-cover linear programming (LP) relaxation. Create two new problems: one that has only the horizontal piece of some of the Ls and another that has only the vertical pieces of the remaining Ls. Place an L into the vertical problem if the LP vertical segment has value at least 1/2, and into the horizontal problem otherwise. Solve the two new problems in polynomial time using the combinatorial method for the 1D problem, or solving the LPs, which are totally unimodular, and thus will return integer solutions. Take all the points selected by either new problem. We prove in the full paper that these points are a 4-approximation.

The above idea naturally extends to a 4-approximation for the weighted version of the problem. For unions consisting of at most k segments drawn from r orientations, the approach yields a $(k \cdot r)$-approximation. Using similar methods and a stronger version of Theorem 5, we also have the following (see full paper):

Theorem 8. *For objects that are unions of a horizontal segment and a vertical line, the hitting set problem has a polynomial-time 10/3-approximation.*

Acknowledgment. This work is supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. J. Mitchell acknowledges support from the US-Israel Binational Science Foundation (grant 2010074) and the National Science Foundation (CCF-1018388, CCF-1526406).

References

1. Alon, N.: A non-linear lower bound for planar epsilon-nets. Discrete Comput. Geom. **47**, 235–244 (2012)
2. Aronov, B., Ezra, E., Sharir, M.: Small-size ε-nets for axis-parallel rectangles and boxes. SIAM J. Computing **39**, 3248–3282 (2010)
3. Brimkov, V.E.: Approximability issues of guarding a set of segments. Int. J. Comput. Math. **90**, 1653–1667 (2013)

4. Brimkov, V.E., Leach, A., Mastroianni, M., Wu, J.: Experimental study on approximation algorithms for guarding sets of line segments. In: Bebis, G., et al. (eds.) ISVC 2010, Part I. LNCS, vol. 6453, pp. 592–601. Springer, Heidelberg (2010)
5. Brimkov, V.E., Leach, A., Mastroianni, M., Wu, J.: Guarding a set of line segments in the plane. Theoret. Comput. Sci. **412**, 1313–1324 (2011)
6. Brimkov, V.E., Leach, A., Wu, J., Mastroianni, M.: Approximation algorithms for a geometric set cover problem. Discrete Appl. Math **160**, 1039–1052 (2012)
7. Brodén, B., Hammar, M., Nilsson, B.J.: Guarding lines and 2-link polygons is APX-hard. In: Proceedings of 13th Canadian Conference on Computational Geometry, pp. 45–48 (2001)
8. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. Discrete Comput. Geom. **14**, 263–279 (1995)
9. Carr, R., Fujito, T., Konjevod, G., Parekh, O.: A 2 1/10-approximation algorithm for a generalization of the weighted edge-dominating set problem. In: Paterson, M. (ed.) ESA 2000. LNCS, vol. 1879, pp. 132–142. Springer, Heidelberg (2000)
10. Chvátal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**, 233–235 (1979)
11. Clarkson, K.L.: Algorithms for polytope covering and approximation. In: Dehne, F., Sack, J.-R., Santoro, N., Whitesdes, S. (eds.) Proceedings of 3rd Workshop Algorithms and Data Structures. LNCS, vol. 709, pp. 246–252. Springer, Heidelberg (1993)
12. Clarkson, K.L., Varadarajan, K.: Improved approximation algorithms for geometric set cover. Discrete Comput. Geom. **37**, 43–58 (2007)
13. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: Proceedings of the 46th ACM Symposium on Theory of Computing, pp. 624–633 (2014)
14. Dom, M., Fellows, M.R., Rosamond, F.A., Sikdar, S.: The parameterized complexity of stabbing rectangles. Algorithmica **62**, 564–594 (2012)
15. Duh, R.-C., Fürer, M.: Approximation of k-set cover by semi-local optimization. In: Proceedings of the 29th ACM Symposium on Theory of Computing, pp. 256–264 (1997)
16. Dumitrescu, A., Jiang, M.: On the approximability of covering points by lines and related problems. Comput. Geom. **48**, 703–717 (2015)
17. Even, G., Levi, R., Rawitz, D., Schieber, B., Shahar, S.M., Sviridenko, M.: Algorithms for capacitated rectangle stabbing and lot sizing with joint set-up costs. ACM Trans. Algorithms **4**, 34:1–34:17 (2008)
18. Even, G., Rawitz, D., Shahar, S.M.: Hitting sets when the VC-dimension is small. Inf. Proc. Letters **95**, 358–362 (2005)
19. Gaur, D.R., Bhattacharya, B.: Covering points by axis parallel lines. In: Proceedings of 23rd European Workshop on Computational Geometry, pp. 42–45 (2007)
20. Gaur, D.R., Ibaraki, T., Krishnamurti, R.: Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. J. Algorithms **43**, 138–152 (2002)
21. Giannopoulos, P., Knauer, C., Rote, G., Werner, D.: Fixed-parameter tractability and lower bounds for stabbing problems. Comput. Geom. **46**, 839–860 (2013)
22. Hassin, R., Megiddo, N.: Approximation algorithms for hitting objects with straight lines. Discrete Appl. Math. **30**, 29–42 (1991)
23. Heednacram, A.: The NP-hardness of covering points with lines, paths and tours and their tractability with FPT-algorithms. Ph.D. Thesis, Griffith University (2010)
24. Hochbaum, D.S., Maas, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM **32**, 130–136 (1985)

25. Joshi, A., Narayanaswamy, N.S.: Approximation algorithms for hitting triangle-free sets of line segments. In: Ravi, R., Gørtz, I.L. (eds.) SWAT 2014. LNCS, vol. 8503, pp. 357–367. Springer, Heidelberg (2014)
26. Kovaleva, S., Spieksma, F.C.: Approximation algorithms for rectangle stabbing and interval stabbing problems. SIAM J. Discrete Math. **20**, 748–768 (2006)
27. Kratsch, S., Philip, G., Ray, S.: Point line cover: the easy kernel is essentially tight. In: Proceedings of 25th ACM-SIAM Symposium on Discrete Algorithms, pp. 1596–1606 (2014)
28. Kumar, V.S.A., Arya, S., Ramesh, H.: Hardness of set cover with intersection 1. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 624–635. Springer, Heidelberg (2000)
29. Langerman, S., Morin, P.: Covering things with things. Discrete Comput. Geom. **33**, 717–729 (2005)
30. Megiddo, N., Tamir, A.: On the complexity of locating linear facilities in the plane. Oper. Res. Lett. **1**, 194–197 (1982)
31. Mustafa, N.H., Ray, S.: Improved results on geometric hitting set problems. Discrete Comput. Geom. **44**, 883–895 (2010)
32. O'Rourke, J.: Art Gallery Theorems and Algorithms. The International Series of Monographs on Computer Science. Oxford University Press, New York (1987)
33. Pach, J., Tardos, G.: Tight lower bounds for the size of epsilon-nets. J. Am. Math. Soc. **26**, 645–658 (2013)
34. Urrutia, J.: Art Gallery and Illumination Problems. Handbook of Computational Geometry. Elsevier, Amsterdam (1999). Chap. 22