Optimal Covering and Hitting of Line Segments by Two Axis-Parallel Squares

Sanjib Sadhu[1(✉)], Sasanka Roy[2], Subhas C. Nandy[2], and Suchismita Roy[1]

[1] Department of CSE, National Institute of Technology Durgapur, Durgapur, India
sanjibsadhu411@gmail.com
[2] Indian Statistical Institute, Kolkata, India

Abstract. This paper discusses the problem of covering and hitting a set of line segments $\mathcal{L}$ in $\mathbb{R}^2$ by a pair of axis-parallel squares such that the side length of the larger of the two squares is minimized. We also discuss the restricted version of covering, where each line segment in $\mathcal{L}$ is to be covered completely by at least one square. The proposed algorithm for the covering problem reports the optimum result by executing only two passes of reading the input data sequentially. The algorithm proposed for the hitting and restricted covering problems produces optimum result in $O(n \log n)$ time. All the proposed algorithms are in-place, and they use only $O(1)$ extra space. The solution of these problems also give a $\sqrt{2}$ approximation for covering and hitting those line segments $\mathcal{L}$ by two congruent disks of minimum radius with same computational complexity.

Keywords: Two-center problem · Covering line segments by squares · Two pass algorithm · Computational geometry

1 Introduction

Covering a point set by squares/disks has drawn interest to the researchers due to its applications in sensor network. Covering a given point set by k congruent disks of minimum radius, known as k-center problem, is NP-Hard [12]. For $k = 2$, this problem is referred to as the *two center problem* [3,5,6,8,9,13].

A line segment ℓ_i is said to be covered (resp. hit) by two squares if every point (resp. at least one point) of ℓ_i lies inside one or both of the squares. For a given set $\mathcal{L}$ of line segments, the objective is to find two axis-parallel congruent squares such that each line segment in $\mathcal{L}$ is covered (resp. hit) by the union of these two squares, and the size of the squares is as small as possible. There are mainly two variations of the covering problem: standard version and discrete version. In discrete version, the center of the squares must be on some specified points, whereas there are no such restriction in standard version. In this paper, we focus our study on the standard version of covering and hitting a set $\mathcal{L}$ of line segments in $\mathbb{R}^2$ by two axis-parallel congruent squares of minimum size.

As an application, consider a sensor network, where each mobile sensor is moving to and fro along different line segment. The objective is to place two

Y. Cao and J. Chen (Eds.): COCOON 2017, LNCS 10392, pp. 457–468, 2017.
DOI: 10.1007/978-3-319-62389-4_38

base stations of minimum transmission range so that each of mobile sensors are always (resp. intermittently) connected to any of the base stations. This problem is exactly same as to cover (resp. hit) the line segments by two congruent disks (in our case axis-parallel congruent squares) of minimum radius.

Most of the works on the *two center problem* deal with covering a given point set. Kim and Shin [11] provided an optimal solution for the *two center problem* of a convex polygon where the covering objects are two disks. As mentioned in [11], the major differences between the *two-center problem for a convex polygon* P and the *two-center problem for a point set* S are (i) points covered by the two disks in the former problem are *in convex positions* (instead of arbitrary positions), and (ii) the union of two disks should also cover the edges of the polygon P. The feature (i) indicates the problem may be easier than the standard two-center problem for points, but feature (ii) says that it might be more difficult. To the best of our knowledge, there are no works on covering or hitting a set of line segments by two congruent squares of minimum size.

Related Work: Drenzer [4] covered a given point set S by two axis-parallel squares of minimum size in $O(n)$ time, where where $n = |S|$. Ahn and Bae [10] proposed an $O(n^2 \log n)$ time algorithm for covering a given point set S by two disjoint rectangles where one of the rectangles is axis parallel and other one is of arbitrary orientation, and the area of the larger rectangle is minimized. Two congruent squares of minimum size covering all the points in S, where each one is of arbitrary orientation, can be computed in $O(n^4 \log n)$ time [1]. The best known deterministic algorithm for the standard version of two-center problem for a point set S is given by Sharir [13] that runs in $O(n \log^9 n)$ time. Eppstein [5] proposed a randomized algorithm for the same problem with expected time complexity $O(n \log^2 n)$. The standard and discrete versions of the two-center problem for a convex polygon P was first solved by Kim and Shin [11] in $O(n \log^3 n \log \log n)$ and $O(n \log^2 n)$ time respectively. Hoffmann [7] solved the rectilinear 3-center problem for a point set in $O(n)$ time. However none of the algorithms in [1,4,7] can handle the line segments.

Our Work: We propose in-place algorithms for covering and hitting n line segments in $\mathbb{R}^2$ by two axis-parallel congruent squares of minimum size. We also study the restricted version of the covering problem where each object needs to be completely covered by at least one of the reported squares. The time complexities of our proposed algorithms for these three problems are $O(n)$, $O(n \log n)$ and $O(n \log n)$ respectively, and these work using $O(1)$ extra work-space. The same algorithms work for covering/hitting a polygon, or a set of polygons by two axis-parallel congruent squares of minimum size. We show that the result of this algorithm can produce a solution for the problem of covering (resp. hitting) these line segments by two congruent disks of minimum radius in $O(n)$ (resp. $O(n \log n)$) time with an approximation factor $\sqrt{2}$.

1.1 Notations and Terminologies Used

Throughout this paper, unless otherwise stated a *square* is used to imply an axis-parallel square. We will use the following notations and definition.

Symbols used	Meaning
$\overline{pq}$ and $\|pq\|$	the line segment joining two points p and q, and its length
$x(p)$ (resp. $y(p)$)	x- (resp. y-) coordinates of the point p
$\|x(p) - x(q)\|$	*horizontal distance* between a pair of points p and q
$\|y(p) - y(q)\|$	*vertical distance* between a pair of points p and q
$s \in \overline{pq}$	the point s lies on the line segment $\overline{pq}$
$\square efgh$	an axis-parallel rectangle with vertices at e, f, g and h
$size(\mathcal{S})$	size of square $\mathcal{S}$; it is the length of its one side
$LS(\mathcal{S})$, $RS(\mathcal{S})$	Left-side of square $\mathcal{S}$ and right-side of square $\mathcal{S}$
$TS(\mathcal{S})$, $BS(\mathcal{S})$	Top-side of square $\mathcal{S}$ and bottom-side of square $\mathcal{S}$

Definition 1. *A square is said to be* **anchored** *with a vertex of a rectangle $\mathcal{R} = \square efgh$, if one of the corners of the square coincides with that vertex of $\mathcal{R}$.*

2 Covering Line Segments by Two Congruent Squares

LCOVER problem: Given a set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of n line segments (possibly intersecting) in $\mathbb{R}^2$, the objective is to compute two congruent squares $\mathcal{S}_1$ and $\mathcal{S}_2$ of minimum size whose union covers all the members in $\mathcal{L}$.

In the first pass, a linear scan is performed among the objects in $\mathcal{L}$, and four points a, b, c and d are identified with minimum x-, maximum y-, maximum x- and minimum y-coordinate respectively among the end-points of $\mathcal{L}$. This defines an axis-parallel rectangle $\mathcal{R} = \square efgh$ of minimum size that covers $\mathcal{L}$, where $a \in \overline{he}$, $b \in \overline{ef}$, $c \in \overline{fg}$ and $d \in \overline{gh}$. We use $L = |x(c) - x(a)|$ and $W = |y(b) - y(d)|$ as the length and width respectively of the rectangle $\mathcal{R} = \square efgh$, and we assume that $L \geq W$. We assume that $\mathcal{S}_1$ lies to the left of $\mathcal{S}_2$. $\mathcal{S}_1$ and $\mathcal{S}_2$ may or may not overlap (see Fig. 1). We use $\sigma = size(\mathcal{S}_1) = size(\mathcal{S}_2)$.

Lemma 1

(a) There exists an optimal solution of the problem where $LS(\mathcal{S}_1)$ and $RS(\mathcal{S}_2)$ pass through the points a and c respectively.

(b) The top side of at least one of $\mathcal{S}_1$ and $\mathcal{S}_2$ pass through the point b, and the bottom side of at least one of $\mathcal{S}_1$ and $\mathcal{S}_2$ pass through the point d.

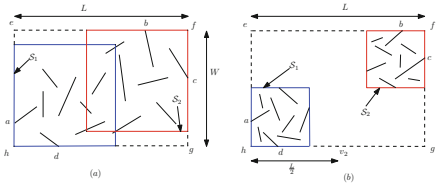

Fig. 1. Squares $\mathcal{S}_1$ and $\mathcal{S}_2$ are (a) overlapping, (b) disjoint

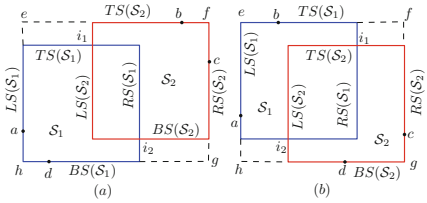

Fig. 2. (a) **Configuration 1** and (b) **Configuration 2** of squares $\mathcal{S}_1$ and $\mathcal{S}_2$

Thus in an optimal solution of the **LCOVER** problem, $a \in LS(\mathcal{S}_1)$ and $c \in RS(\mathcal{S}_2)$. We need to consider two possible configurations of an optimum solution (i) $b \in TS(\mathcal{S}_2)$ and $d \in BS(\mathcal{S}_1)$, and (ii) $b \in TS(\mathcal{S}_1)$ and $d \in BS(\mathcal{S}_2)$. These are named as **Configuration 1** and **Configuration 2** respectively (see Fig. 2).

Observation 1

(a) If the optimal solution of LCOVER problem satisfies **Configuration 1**, *then the bottom-left corner of $\mathcal{S}_1$ will be anchored at the point h, and the top-right corner of $\mathcal{S}_2$ will be anchored at the point f.*

(b) If the optimal solution of LCOVER problem satisfies **Configuration 2**, *then the top-left corner of $\mathcal{S}_1$ will be anchored at the point e, and the bottom-right corner of $\mathcal{S}_2$ will be anchored at the point g.*

We consider each of the configurations separately, and compute the two axis-parallel congruent squares $\mathcal{S}_1$ and $\mathcal{S}_2$ of minimum size whose union covers the given set of line segments $\mathcal{L}$. If σ_1 and σ_2 are the sizes obtained for **Configuration 1** and **Configuration 2** respectively, then we report $\min(\sigma_1, \sigma_2)$.

Consider the rectangle $\mathcal{R} = \square efgh$ covering $\mathcal{L}$, and take six points k_1, k_2, k_3, k_4, v_1 and v_2 on the boundary of $\mathcal{R}$ satisfying $|k_1 f| = |ek_3| = |hk_4| = |k_2 g| = W$ and $|ev_1| = |hv_2| = \frac{L}{2}$ (see Fig. 3). Throughout the paper we assume h as the origin in the co-ordinate system, i.e. $h = (0, 0)$.

Observation 2

(i) *The Voronoi partitioning line λ_1 of the corners f and h of $\mathcal{R} = \square efgh$ with respect to the L_∞ norm*[1] *is the polyline $k_1 z_1 z_2 k_4$, where the coordinates of its defining points are $k_1 = (L - W, W)$, $z_1 = (L/2, L/2)$, $z_2 = (L/2, W - L/2)$ and $k_4 = (W, 0)$ (see Fig. 3(a)).*

(ii) *The Voronoi partitioning line λ_2 of e and g of $\mathcal{R} = \square efgh$ in L_∞ norm is the polyline $k_3 z_1 z_2 k_2$ where $k_3 = (W, W)$ and $k_2 = (L - W, 0)$ (see Fig. 3(b)).*

[1] L_∞ distance between two points a and b is given by $\max(|x(a) - x(b)|, |y(a) - y(b)|)$.

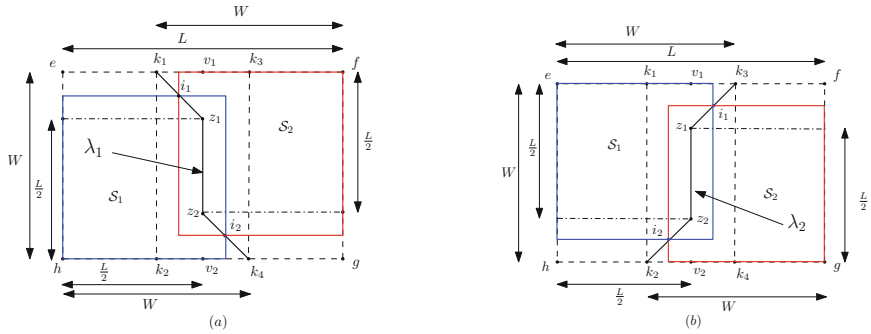

Fig. 3. Voronoi partitioning line (a) $\lambda_1 = k_1z_1z_2k_4$ of f and h in **Configuration 1** (b) $\lambda_2 = k_3z_1z_2k_2$ of e and g in **Configuration 2**

Note that, if $W \leq \frac{L}{2}$*, then the voronoi partitioning lines* λ_1 *and* λ_2 *for both the pairs* $(f,\ h)$ *and* $(e,\ g)$ *will be same, i.e.,* $\lambda_1 = \lambda_2 = \overline{v_1v_2}$*, where* $v_1 = (\frac{L}{2}, 0)$ *and* $v_2 = (\frac{L}{2}, W)$*.*

Lemma 2

(a) For **Configuration 1***, All the points* p *inside the polygonal region* $ek_1z_1z_2k_4h$ *satisfy* $d_\infty(p,h) < d_\infty(p,f)$*, and all points* p *inside the polygonal region* $k_1fgk_4z_2z_1$ *satisfy* $d_\infty(p,f) < d_\infty(p,h)$ *(see Fig. 3(a)).*
(b) Similarly for **Configuration 2***, all points* p *inside polygonal region* $ek_3z_1z_2k_2h$*, satisfy* $d_\infty(p,e) < d_\infty(p,g)$*, and all points* p *that lie inside the polygonal region* $k_3fgk_2z_2z_1$*, satisfy* $d_\infty(p,g) < d_\infty(p,e)$ *(see Fig. 3(b)).*

Lemma 3. *If* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *intersect, then the points of intersection* i_1 *and* i_2 *will always lie on voronoi partitioning line* $\lambda_1 = k_1z_1z_2k_4$ *(resp.* $\lambda_2 = k_3z_1z_2k_2$*) depending on whether* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *satisfy* **Configuration 1** *or* **Configuration 2***.*

Our algorithm consists of two passes. In each pass we sequentially read each element of the input array $\mathcal{L}$ exactly once. We consider $W > \frac{L}{2}$ only. The other case i.e. $W \leq \frac{L}{2}$ can be handled in the similar way.

Pass-1: We compute the rectangle $\mathcal{R} = \square efgh$, and the voronoi partitioning lines λ_1 and λ_2 (see Fig. 3) for handling **Configuration 1** and **Configuration 2**. We now discuss Pass 2 for **Configuration 1**. The same method works for **Configuration 2**, and for both the configurations, the execution run simultaneously keeping a $O(1)$ working storage.

Pass-2: λ_1 splits $\mathcal{R}$ into two disjoint parts, namely $\mathcal{R}_1 =$ region $ek_1z_1z_2k_4h$ and $\mathcal{R}_2 =$ region $fk_1z_1z_2k_4g$. We initialize $\sigma_1 = 0$. Next, we read elements in the input array $\mathcal{L}$ in sequential manner. For each element $\ell_i = [p_i, q_i]$, we identify

its portion lying in one/both the parts $\mathcal{R}_1$ and $\mathcal{R}_2$. Now, considering Lemma 2 and Observation 1, we execute the following:

ℓ_i **lies inside** $\mathcal{R}_1$**:** Compute $\delta = \max(d_\infty(p_i, h), d_\infty(q_i, h))$.
ℓ_i **lies inside** $\mathcal{R}_2$**:** Compute $\delta = \max(d_\infty(p_i, f), d_\infty(q_i, f))$.
ℓ_i **is intersected by** λ_1**:** Let θ be the point of intersection of ℓ_i and λ_1, $p_i \in \mathcal{R}_1$ and $q_i \in \mathcal{R}_2$. Here, we compute $\delta = \max(d_\infty(p_i, h), d_\infty(\theta, h), d_\infty(q_i, f))$.

If $\delta > \sigma_1$, we update σ_1 with δ. Similarly, σ_2 is also computed in this pass considering the pair$(e,\ g)$ and their partitioning line λ_2. Finally, $\min(\sigma_1, \sigma_2)$ is returned as the optimal size along with the centers of the squares $\mathcal{S}_1$ and $\mathcal{S}_2$.

Theorem 1. *Given a set of line segments $\mathcal{L}$ in $\mathbb{R}^2$ in an array, one can compute two axis-parallel congruent squares of minimum size whose union covers $\mathcal{L}$ by reading the input array only twice in sequential manner, and maintaining $O(1)$ extra work-space.*

3 Hitting Line Segments by Two Congruent Squares

Definition 2. *A geometric object Q is said to be* hit *by a square $\mathcal{S}$ if at least one point of Q lies inside (or on the boundary of) $\mathcal{S}$.*

Line segment hitting (LHIT) problem: Given a set $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of n line segments in $\mathbb{R}^2$, compute two axis-parallel congruent squares $\mathcal{S}_1$ and $\mathcal{S}_2$ of minimum size whose union hits all the line segments in $\mathcal{L}$.

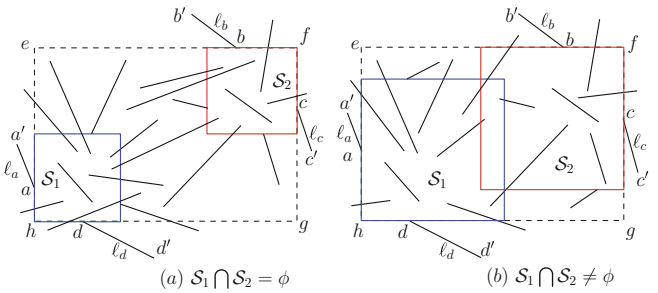

Fig. 4. Two axis-parallel congruent squares $\mathcal{S}_1$ and $\mathcal{S}_2$ hit line segments in $\mathcal{L}$

The squares $\mathcal{S}_1$ and $\mathcal{S}_2$ may or may not be disjoint (see Fig. 4). We now describe the algorithm for this **LHIT** problem.

For each line segment ℓ_i, we use $LP(\ell_i)$, $RP(\ell_i)$, $TP(\ell_i)$ and $BP(\ell_i)$ to denote its left end-point, right end-point, top end-point and bottom end-point using the relations $x(LP(\ell_i)) \leq x(RP(\ell_i))$ and $y(BP(\ell_i)) \leq y(TP(\ell_i))$. Now we compute four line segments $\ell_a,\ \ell_b,\ \ell_c$, and $\ell_d \in \mathcal{L}$ such that one of their end-points a, b, c and d, respectively satisfy the following

$$a = \min_{\forall \ell_i \in \mathcal{L}} x(RP(\ell_i)), b = \max_{\forall \ell_i \in \mathcal{L}} y(BP(\ell_i)), c = \max_{\forall \ell_i \in \mathcal{L}} x(LP(\ell_i)), d = \min_{\forall \ell_i \in \mathcal{L}} y(TP(\ell_i))$$

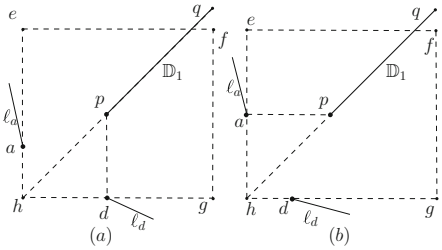

Fig. 5. $\mathbb{D}_1$ for $y(LP(\ell_a)) \geq y(RP(\ell_a))$ and $x(TP(\ell_d)) < x(BP(\ell_d))$

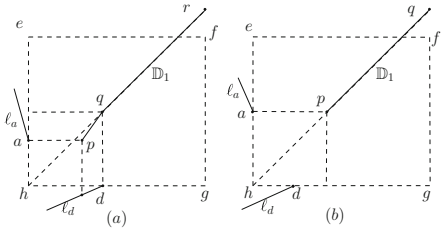

Fig. 6. $\mathbb{D}_1$ for $y(LP(\ell_a)) \geq y(RP(\ell_a))$ and $x(TP(\ell_d)) \geq x(BP(\ell_d))$

We denote the other end point of ℓ_a, ℓ_b, ℓ_c and ℓ_d by a', b', c' and d', respectively. The four points a, b, c, d define an axis-parallel rectangle $\mathcal{R} = \Box efgh$ of minimum size that hits all the members of $\mathcal{L}$ (as per Definition 2), where $a \in \overline{he}$, $b \in \overline{ef}$, $c \in \overline{fg}$ and $d \in \overline{gh}$ (see Fig. 4). We use $L = |x(c) - x(a)|$ and $W = |y(b) - y(d)|$ as the length and width of the rectangle $\mathcal{R}$, and assume $L \geq W$. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the two axis-parallel congruent squares that hit the given line segments $\mathcal{L}$ optimally, where $\mathcal{S}_1$ lies to the left of $\mathcal{S}_2$.

Observation 3. *(a) The left side of $\mathcal{S}_1$ (resp. right side of $\mathcal{S}_2$) must not lie to the right of (resp. left of) the point a (resp. c), and (b) the top side (resp. bottom side) of both $\mathcal{S}_1$ and $\mathcal{S}_2$ cannot lie below (resp. above) the point b (resp. d).*

For the **LHIT** problem, we say $\mathcal{S}_1$ and $\mathcal{S}_2$ are in **Configuration 1**, if $\mathcal{S}_1$ hits both ℓ_a and ℓ_d, and $\mathcal{S}_2$ hits both ℓ_b and ℓ_c. Similarly, $\mathcal{S}_1$ and $\mathcal{S}_2$ are said to be in **Configuration 2**, if $\mathcal{S}_1$ hits both ℓ_a and ℓ_b, and $\mathcal{S}_2$ hits both ℓ_c and ℓ_d.

Without loss of generality, we assume that $\mathcal{S}_1$ and $\mathcal{S}_2$ are in **Configuration 1**. We compute the reference (poly) line $\mathbb{D}_1$ (resp. $\mathbb{D}_2$) on which the top-right corner of $\mathcal{S}_1$ (resp. bottom-left corner of $\mathcal{S}_2$) will lie. Let $\mathbb{T}_1$ (resp. $\mathbb{T}_2$) be the line passing through h (resp. f) with slope 1. Our algorithm consists of the following phases:

1. Computation of the reference lines $\mathbb{D}_1$ and $\mathbb{D}_2$.
2. Computation of event points for the top-right (resp. bottom-left) corner of $\mathcal{S}.1$ (resp. $\mathcal{S}_2$) on $\mathbb{D}_1$ (resp. $\mathbb{D}_2$).
3. Searching for pair ($\mathcal{S}_1$, $\mathcal{S}_2$) that hit all the line segments in $\mathcal{L}$ and max($size(\mathcal{S}_1)$, $size(\mathcal{S}_2)$) is minimized.

Computation of the reference lines $\mathbb{D}_1$ and $\mathbb{D}_2$: The reference line $\mathbb{D}_1$ is computed based on the following four possible orientations of ℓ_a and ℓ_d

(i) $\boldsymbol{y(LP(\ell_a)) \geq y(RP(\ell_a))}$ and $\boldsymbol{x(TP(\ell_d)) < x(BP(\ell_d))}$: Here $\mathbb{D}_1$ is the segment $\overline{pq}$ on $\mathbb{T}_1$ where p is determined (i) by its x-coordinate i.e. $x(p) = x(d)$, if $|ha| < |hd|$ (see Fig. 5(a)), (ii) by its y-coordinate i.e. $y(p) = y(a)$, if $|ha| \geq |hd|$ (see Fig. 5(b)). The point q on $\mathbb{T}_1$ satisfy $x(q) = x(f)$.

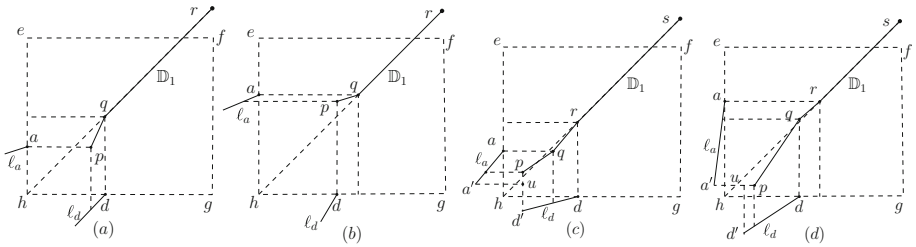

Fig. 7. $\mathbb{D}_1$ for $y(LP(\ell_a)) < y(RP(\ell_a))$ and $x(TP(\ell_d)) > x(BP(\ell_d))$

(ii) $\boldsymbol{y(LP(\ell_a)) \geq y(RP(\ell_a))}$ and $\boldsymbol{x(TP(\ell_d)) \geq x(BP(\ell_d))}$: Here,
if $|ha| < |hd|$ (see Fig. 6(a)), then the reference line $\mathbb{D}_1$ is a polyline $\overline{pqr}$, where (i) $y(p) = y(a)$ and $x(p)$ satisfies $|x(p) - x(a)|$ = vertical distance of p from the line segment ℓ_d, (ii) the point q lies on $\mathbb{T}_1$ satisfying $x(q) = x(d)$ and (iii) the point r lies on $\mathbb{T}_1$ satisfying $x(r) = x(f)$.
If $|ha| \geq |hd|$ (see Fig. 6(b)), then the reference line $\mathbb{D}_1$ is a line segment $\overline{pq}$, where p, q lies on $\mathbb{T}_1$, and p satisfies $y(p) = y(a)$ and q satisfies $x(q) = x(f)$.

(iii) $\boldsymbol{y(LP(\ell_a)) < y(RP(\ell_a))}$ and $\boldsymbol{x(TP(\ell_d)) \leq x(BP(\ell_d))}$: This case is similar to case (ii), and we can compute the respective reference lines.

(iv) $\boldsymbol{y(LP(\ell_a)) < y(RP(\ell_a))}$ and $\boldsymbol{x(TP(\ell_d)) > x(BP(\ell_d))}$: There are two possible subcases:
(A) If ℓ_a and ℓ_d are parallel or intersect (after extension) at a point to the right of $\overline{he}$ (Fig. 7(a,b)), then the reference line $\mathbb{D}_1$ is a polyline $\overline{pqr}$, where **(a) if** $|\mathbf{ha}| < |\mathbf{hd}|$ (Fig. 7(a)), then (1) $y(p) = y(a)$ and $|x(p) - x(a)|$ = the vertical distance of p from ℓ_d, (2) the points q and r lie on $\mathbb{T}_1$ satisfying $x(q) = x(d)$ and $x(r) = x(f)$, **(b) if** $|\mathbf{ha}| > |\mathbf{hd}|$ (Fig. 7(b)), then (1) $x(p) = x(d)$ and $|y(p) - y(d)|$ = the horizontal distance of p from ℓ_a, (2) the points q and r lie on $\mathbb{T}_1$ satisfying $y(q) = y(a)$ and $x(r) = x(f)$.

(B) If extended ℓ_a and ℓ_d intersect at a point to the left of $\overline{he}$ (Fig. 7(c,d)), then $\mathbb{D}_1$ is a polyline $\overline{pqrs}$, where
(i) the line segment $\overline{pq}$ is such that for every point $\theta \in \overline{pq}$, the horizontal distance of θ from ℓ_a and the vertical distance of θ from ℓ_d are same.
(ii) the line segment $\overline{qr}$ is such that for every point $\theta \in \overline{qr}$, we have
if $|\mathbf{ha}| < |\mathbf{hd}|$ then $|x(\theta) - x(a)|$ = vertical distance of θ from ℓ_d (Fig. 7(c)),
else $|y(\theta) - x(d)|$ = horizontal distance of θ from ℓ_a, (Fig. 7(d))
(iii) the point s lies on $\mathbb{T}_1$ satisfying $x(s) = x(f)$.

In the same way, we can compute the reference line $\mathbb{D}_2$ based on the four possible orientations of ℓ_b and ℓ_c. The break points/end points of $\mathbb{D}_2$ will be referred to as p', q', r', s' depending on the appropriate cases. From now onwards, we state the position of square $\mathcal{S}_1$ (resp. $\mathcal{S}_2$) in terms of the position of its top-right corner (resp. bottom-left corner).

Observation 4. *The point $p \in \mathbb{D}_1$ (resp. $p' \in \mathbb{D}_2$) gives the position of minimum sized axis-parallel square $\mathcal{S}_1$ (resp. $\mathcal{S}_2$) that hit ℓ_a and ℓ_d (resp. ℓ_b and ℓ_c).*

Computation of discrete event points on $\mathbb{D}_1$and $\mathbb{D}_2$: Observe that the line segments in $\mathcal{L}$ that hits the vertical half-line below the point $p \in \mathbb{D}_1$ (resp. above the point $p' \in \mathbb{D}_2$), or the horizontal half-line to the left of the point $p \in \mathbb{D}_1$ (resp. to the right of the point $p' \in \mathbb{D}_2$) will be hit by any square that hits ℓ_a and ℓ_d (resp. ℓ_b and ℓ_c), and these line segments need not contribute any event point on $\mathbb{D}_1$ (resp. $\mathbb{D}_2$). For each of the other segments $\ell_i \in \mathcal{L}$, we create an event point e_i^1 (resp. e_i^2) on $\mathbb{D}_1$ (resp. $\mathbb{D}_2$) as follows:

(i) p is an event point on $\mathbb{D}_1$ and p' is an event point on $\mathbb{D}_2$ (see Observation 4)
(ii) If ℓ_i lies completely above $\mathbb{D}_1$ (resp. $\mathbb{D}_2$), then we compute an event point $e_i^1 = (x_{i_1}, y_{i_1})$ on $\mathbb{D}_1$ (resp. $e_i^2 = (x_{i_2}, y_{i_2})$ on $\mathbb{D}_2$) where $y_{i_1} = y(BP(\ell_i))$ (resp. $x_{i_2} = x(RP(\ell_i))$). (e.g. e_1^1 for ℓ_1 and e_4^2 for ℓ_4 in Fig. 8).
(iii) If ℓ_i lies completely below $\mathbb{D}_1$ (resp. $\mathbb{D}_2$), we compute an event point $e_i^1 = (x_{i_1}, y_{i_1})$ on $\mathbb{D}_1$ (resp. $e_i^2 = (x_{i_2}, y_{i_2})$ on $\mathbb{D}_2$) where $x_{i_1} = x(LP(\ell_i))$ (resp. $y_{i_2} = y(TP(\ell_i))$). (e.g. e_3^1 for ℓ_3 and e_6^2 for ℓ_6 in Fig. 8).
(iv) If ℓ_i intersects with $\mathbb{D}_1$ (resp. $\mathbb{D}_2$) at point p_1 (resp. q_1), then we create the event point e_i^1 on $\mathbb{D}_1$ (resp. e_i^2 on $\mathbb{D}_2$) according to the following rule:
(a) If the $x(BP(\ell_i)) > x(p_1)$ (resp. $x(TP(\ell_i)) < x(q_1)$), then we take p_1 (resp. q_1) as the event point e_1^i (resp. e_2^i). (e.g. e_4^1 for ℓ_4 in Fig. 8).
(b) If $x(BP(\ell_i)) < x(p_1)$ then if $BP(\ell_i)$ lies below $\mathbb{D}_1$ then we consider the point of intersection by $\mathbb{D}_1$ with the vertical line passing through the $BP(\ell_i)$ as the event point e_i^1 (see e_2^1 for ℓ_2 in Fig. 8), and if $BP(\ell_i)$ lies above $\mathbb{D}_1$ then we consider the point of intersection $\mathbb{D}_1$ with the horizontal line passing through $BP(\ell_i)$ as the event point e_i^1 (see e_5^1 for ℓ_5 in Fig. 8).
(c) If $x(TP(\ell_i)) > x(q_1)$ then if $TP(\ell_i)$ lies above $\mathbb{D}_2$ then we consider the point of intersection by $\mathbb{D}_2$ with the vertical line passing through $TP(\ell_i)$ as the event point e_i^2, and if $TP(\ell_i)$ lies below $\mathbb{D}_2$ then we consider the point of intersection $\mathbb{D}_2$ with the horizontal line passing through $TP(\ell_i)$ as the event point e_i^2.

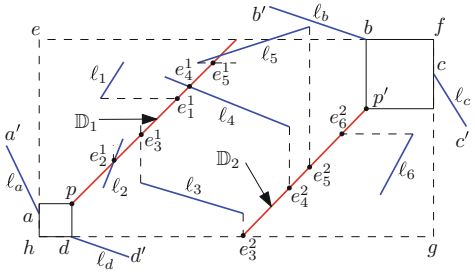

Fig. 8. Event points for **LHIT** problem under **Configuration 1**

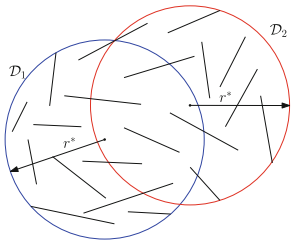

Fig. 9. Covering $\mathcal{L}$ by two disks $\mathcal{D}_1$ & $\mathcal{D}_2$

Observation 5

(i) An event e_i^1 on $\mathbb{D}_1$ shows the position of the top-right corner of the minimum sized square $\mathcal{S}_1$ that hits ℓ_a, ℓ_d and ℓ_i, and an event e_i^2 on $\mathbb{D}_2$ shows the position of the bottom-left corner of the minimum sized square $\mathcal{S}_2$ that hits ℓ_b, ℓ_c and ℓ_i.

(ii) The square $\mathcal{S}_1$ whose top-right corner is at e_i^1 on $\mathbb{D}_1$ hits all those line segments ℓ_j whose corresponding event points e_j^1 on $\mathbb{D}_1$ satisfies $x(h) \leq x(e_j^1) \leq x(e_i^1)$. Similarly, the square $\mathcal{S}_2$ whose bottom-left corner is at e_2^i on $\mathbb{D}_2$ hits all those line segments ℓ_j whose corresponding event point e_j^2 on $\mathbb{D}_2$ satisfies $x(e_i^1) \leq x(e_j^1) \leq x(f)$.

Let us consider an event point e_i^1, and the corresponding square $\mathcal{S}_1$. We can identify the size and position of the other square $\mathcal{S}_2$ that hit the line segments ℓ_j which were not hit by $\mathcal{S}_1$ in linear time. Observe that, as the size of $\mathcal{S}_1$ increases, the size of $\mathcal{S}_2$ either decreases or remains same. Thus $\max(size(\mathcal{S}_1),\ size(\mathcal{S}_2))$ is a convex function, and we can compute the minimum value of this function in $O(\log n)$ iterations.

We initially set $\alpha = 1$ and $\beta = n$. In each iteration of our *in-place algorithm*, we compute $\mu = \lfloor \frac{\alpha+\beta}{2} \rfloor$, and compute the μ-th smallest element e^* among $\{e_i^1, i = 1, 2, \ldots, n\}$, and define $\mathcal{S}_1$ in $O(n)$ time [2]. Next, in a linear pass, we compute $\mathcal{S}_2$ for hitting the line segments ℓ_j that are not hit by $\mathcal{S}_1$. If $size(\mathcal{S}_1) < size(\mathcal{S}_2)$ then we set $\alpha = \mu$, otherwise we set $\beta = \mu$ to execute the next iteration. If in two consecutive iterations we get the same μ, the process terminates.

Similarly, we can determine the optimal size of the congruent squares $\mathcal{S}_1$ and $\mathcal{S}_2$ in **Configuration 2**. Finally we consider that configuration for which the size of the congruent squares is minimized. Thus we get the following result:

Theorem 2. *The* **LHIT** *problem can be solved optimally in $O(n \log n)$ time using $O(1)$ extra work-space.*

4 Restricted Version of LCOVER Problem

In restricted version of the **LCOVER** problem, each line segment in $\mathcal{L}$ is to be covered completely by atleast one of the two congruent axis-parallel squares $\mathcal{S}_1$ and $\mathcal{S}_2$. We compute the axis-parallel rectangle $\mathcal{R} = \Box efgh$ passing through the four points a, b, c and d as in our algorithm for **LCOVER** problem. As in the **LCOVER** problem, here also we have two possible configurations for optimal solution. Without loss of generality, we assume that $\mathcal{S}_1$ and $\mathcal{S}_2$ satisfy **Configuration 1**. We consider two reference lines $\mathbb{D}_1$ and $\mathbb{D}_2$, each with unit slope that passes through h and f, respectively. These reference lines $\mathbb{D}_1$ and $\mathbb{D}_2$ are the locus of the top-right corner of $\mathcal{S}_1$ and bottom-left corner of $\mathcal{S}_2$, respectively. For each line segment ℓ_i, we create an event point $e_i^1 = (x_{i_1}, y_{i_1})$ on $\mathbb{D}_1$ (resp. $e_i^2 = (x_{i_2}, y_{i_2})$ on $\mathbb{D}_2$) as follows:

(i) If ℓ_i lies completely above $\mathbb{D}_1$ (resp. $\mathbb{D}_2$), then the event point e_i^1 on $\mathbb{D}_1$ (resp. e_i^2 on $\mathbb{D}_2$) will satisfy $y_{i_1} = y(TP(\ell_i))$ (resp. $x_{i_2} = x(LP(\ell_i))$).
(ii) If ℓ_i lies completely below $\mathbb{D}_1$ (resp. $\mathbb{D}_2$) then the event point e_i^1 on $\mathbb{D}_1$ (resp. e_i^2 on $\mathbb{D}_2$) will satisfy $x_{i_1} = x(RP(\ell_i))$ (resp. $y_{i_2} = y(BP(\ell_i))$).
(iii) If ℓ_i intersects with $\mathbb{D}_1$ then we create the event point e_i^1 on $\mathbb{D}_1$ as follows: Let the horizontal line through $TP(\ell_i)$ intersect with $\mathbb{D}_1$ at point p, and the vertical line through $BP(\ell_i)$ intersect with $\mathbb{D}_1$ at point q. If $x(p) > x(q)$, then we take p (else q) as the event point on $\mathbb{D}_1$.
(iv) If ℓ_i intersects with $\mathbb{D}_2$, then we create the event point e_i^2 on $\mathbb{D}_2$ as follows: Let the vertical line through $BP(\ell_i)$ intersect with $\mathbb{D}_2$ at point p, and the horizontal line through $TP(\ell_i)$ intersect with $\mathbb{D}_2$ at point q. If $x(p) > x(q)$, then we take q (else p) as the event point on $\mathbb{D}_2$.

Observation similar to Observation 5 in LHIT problem also holds for this problem where $\mathcal{S}_1$ and $\mathcal{S}_2$ cover $\mathcal{L}$ with restriction. Thus, here we can follow the same technique as in LHIT problem to obtain the following result:

Theorem 3. *The restricted version of* **LCOVER** *problem can be solved optimally in* $O(n \log n)$ *time using* $O(1)$ *extra work-space.*

5 Covering/Hitting Line Segments by Two Congruent Disks

In this section, we consider problems related to **LCOVER**, **LHIT** and **restricted LCOVER** problem, called *two center problem*, where the objective is to cover, hit or restricted-cover the given line segments in $\mathcal{L}$ by two congruent disks so that their (common) radius is minimized. Figure 9 demonstrates a covering instance of this *two center problem*. Here, we first compute two axis-parallel squares $\mathcal{S}_1$ and $\mathcal{S}_2$ whose union covers/ hits all the members of $\mathcal{L}$ optimally as described in the previous section. Then we report the circum-circles $\mathcal{D}_1$ and $\mathcal{D}_2$ of $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively as an approximate solution of the *two center problem*.

Lemma 4. *A lower bound for the optimal radius of two center problem for* $\mathcal{L}$ *is the radius* r' *of in-circle of the two congruent squares* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *of minimum size that cover/ hit/ restricted-cover* $\mathcal{L}$*; i.e.* $r' \leq r^*$.

The radius r of the circum-circle $\mathcal{D}_1$ and $\mathcal{D}_2$ of the squares $\mathcal{S}_1$ and $\mathcal{S}_2$ is $\sqrt{2}$ times of the radius r' of their in-circles. Lemma 4 says that $r' \leq r^*$. Thus, we have

Theorem 4. *Algorithm* **Two center** *generates a* $\sqrt{2}$ *approximation result for* LCOVER*,* LHIT *and* restricted LCOVER *problems for the line segments in* $\mathcal{L}$.

References

1. Bhattacharya, B., Das, S., Kameda, T., Sinha Mahapatra, P.R., Song, Z.: Optimizing squares covering a set of points. In: Zhang, Z., Wu, L., Xu, W., Du, D.-Z. (eds.) COCOA 2014. LNCS, vol. 8881, pp. 37–52. Springer, Cham (2014). doi:10.1007/978-3-319-12691-3_4
2. Carlsson, S., Sundström, M.: Linear-time in-place selection in less than $3n$ comparisons. In: Staples, J., Eades, P., Katoh, N., Moffat, A. (eds.) ISAAC 1995. LNCS, vol. 1004, pp. 244–253. Springer, Heidelberg (1995). doi:10.1007/BFb0015429
3. Chan, T.M.: More planar two-center algorithms. Comput. Geom. Theory Appl. **13**(3), 189–198 (1999)
4. Drezner, Z.: On the rectangular p-center problem. Naval Res. Log. **34**(2), 229–234 (1987)
5. Eppstein, D.: Faster construction of planar two-centers. In: 8th ACM-SIAM Symposium On Discrete Algorithms (SODA), pp. 131–138 (1997)
6. Hershberger, J.: A fast algorithm for the two-Center decision Problem. Inf. Process. Lett. (Elsevier) **47**(1), 23–29 (1993)
7. Hoffmann, M.: A simple linear algorithm for computing rectilinear 3-centers. Comput. Geom. **31**(3), 150–165 (2005)
8. Jaromczyk, J.W., Kowaluk, M.: An efficient algorithm for the euclidean two-center problem. In: Mehlhorn, K. (ed.) Symposium on Computational Geometry, pp. 303–311. ACM (1994)
9. Katz, M.J., Kedem, K., Segal, M.: Discrete rectilinear 2-center problems. Comput. Geom. **15**(4), 203–214 (2000)
10. Kim, S.S., Bae, S.W., Ahn, H.K.: Covering a point set by two disjoint rectangles. Int. J. Comput. Geometry Appl. **21**(3), 313–330 (2011)
11. Kim, S.K., Shin, C.-S.: Efficient algorithms for two-center problems for a convex polygon. In: Du, D.-Z.-Z., Eades, P., Estivill-Castro, V., Lin, X., Sharma, A. (eds.) COCOON 2000. LNCS, vol. 1858, pp. 299–309. Springer, Heidelberg (2000). doi:10.1007/3-540-44968-X_30
12. Marchetti-Spaccamela, A.: The p center problem in the plane is NP complete. In: Proceedings of the 19th Allerton Conference on Communication, Control and Computing, pp. 31–40 (1981)
13. Sharir, M.: A near-linear algorithm for the planar 2-center problem. Discrete Comput. Geom. **18**(2), 125–134 (1997)