Fast Approximation Algorithms for Stabbing Special Families of Line Segments with Equal Disks

Konstantin Kobylkin[1,2(✉)]

[1] Krasovsky Institute of Mathematics and Mechanics, Ural Branch of RAS, Sophya Kovalevskaya Street 16, 620108 Ekaterinburg, Russia
[2] Ural Federal University, Mira Street 19, 620002 Ekaterinburg, Russia

Abstract. An NP-hard problem is considered to stab a given set of n straight line segments on the plane with the smallest size subset of disks of fixed radii $r > 0$, where the set of segments forms a straight line drawing $G = (V, E)$ of a planar graph without proper edge crossings. To the best of our knowledge, only 100-approximation $O(n^4 \log n)$-time algorithm is known (Kobylkin, 2018) for this problem. Moreover, when segments of E are axis-parallel, 8-approximation is proposed (Dash et al., 2012), working in $O(n \log n)$ time. In this work another special setting is considered of the problem where G belongs to classes of special plane graphs, which are of interest in network applications. Namely, three fast $O(n^{3/2} \log^2 n)$-expected time algorithms are proposed: a 10-approximate algorithm for the problem, considered on edge sets of minimum Euclidean spanning trees, a 12-approximate algorithm for edge sets of relative neighborhood graphs and 14-approximate algorithm for edge sets of Gabriel graphs. The paper extends recent work (Kobylkin et al. 2019) where $O(n^2)$-time approximation algorithms are proposed with the same constant approximation factors for the problem on those three classes of sets of segments.

Keywords: Operations research · Computational geometry · Approximation algorithms · Straight line segment · Hippodrome

1 Introduction

Facility location problem represents an important application area for many combinatorial optimization problems. Usually, in facility location problems there

The work is carried out within the research, conducted at the Ural Mathematical Center. It is also supported by the Russian Foundation for Basic Research, project №-19-07-01243. The paper is a substantially extended version of the short paper Kobylkin, K., Dryakhlova, I.: Practical approximation algorithms for stabbing special families of line segments with equal disks. In: Kotsireas, I., Pardalos, P. (eds.) Learning and Intelligent Optimization, LION 2020. Lecture Notes in Computer Science, vol. 12096. Springer, Cham.

W. M. P. van der Aalst et al. (Eds.): AIST 2020, LNCS 12602, pp. 421–432, 2021.
https://doi.org/10.1007/978-3-030-72610-2_32

are objects of interest, e.g. customers, roads, offices or production units, facility objects, say, inventories, stores, markets, cellular base stations, petrol or charging stations, and the problem is to place facilities at the vicinity of objects of interest. Both types of objects are often implied to be geographically distributed. Here optimization is usually done over locations of facilities to achieve the minimum average (or maximal) distance from the placed facilities to the objects of interest. Alternatively, total number is minimized of the located facilities while providing the necessary degree of coverage of all objects of interest. The latter class of problems is called a class of coverage problems.

Both types of facility location problems can adequately be modeled by optimization problems from computational geometry. Here objects of interest are encoded by some simple geometric structures, e.g., points, straight line (or curvilinear) segments and rectangles etc. Besides, facility objects are modeled by translates of fixed objects like points, identical disks, axis-parallel squares or rectangles. The corresponding optimization problem from the class of coverage problems can look as follows: given a set $\mathcal{K}$ of geometric objects on the plane, the smallest cardinality set $\mathcal{C}$ of objects is to be found, chosen from a class $\mathcal{F}$ of simply shaped objects, such that each object from $\mathcal{K}$ is intersected by an object from $\mathcal{C}$ in some prescribed way.

In this paper, subquadratic time small constant factor approximation algorithms are designed for the following problem in which $\mathcal{F}$ is a set of radius r disks and $\mathcal{K}$ coincides with a finite set E of straight line segments on the plane. Intersecting Plane Graph with Disks (IPGD): Given a straight line drawing (or a plane graph) $G = (V, E)$ of an arbitrary simple planar graph without proper edge crossings and a constant $r > 0$, find the smallest cardinality set $\mathcal{C}$ of disks of radius r such that $e \cap \bigcup\limits_{C \in \mathcal{C}} C \neq \varnothing$ for each edge $e \in E$. Here each isolated vertex $v \in V$ is treated as a zero-length segment $e_v \in E$. Moreover, the vertex set V is assumed to be in general position, i.e. no triple of points of V lies on any straight line.

Below the term "plane graph" is used to denote any straight line embedding of a planar graph whose (straight line) edges intersect at most at their endpoints.

The IPGD problem finds its applications in sensor network deployment and facility location problems related to optimal coverage of objects of some infrastructure with sensors or facilities.

Suppose one needs to provide a certain degree of coverage of a given road network with facility stations, which could be campings, petrol or charging stations, police precincts etc. Geometrically, the network roads can be modeled by piecewise linear arcs on the plane. One can split these arcs into chains of elementary straight line segments such that any two of the resulting elementary segments intersect at most at their endpoints. To cover the road network with facility stations to some extent, it might be reasonable to place the minimum number of stations such that each piece of every road (represented by an elementary segment) is within a given distance from some of the placed stations. This modeling approach gives a geometric combinatorial optimization model, which coincides with the IPGD problem.

The IPGD problem has close connections with the classical geometric Hitting Set problem on the plane. To describe a Hitting Set formulation of the IPGD problem, some notation is given below. Suppose $N_r(e) = \{x \in \mathbb{R}^2 : d(x,e) \leq r\}$, $\mathcal{N}_r(E) = \{N_r(e) : e \in E\}$ and $d(x,e)$ is Euclidean distance between a point $x \in \mathbb{R}^2$ and a segment $e \in E$, i.e. Euclidean distance between x and its projection on e; for a zero-length segment $x \in \mathbb{R}^2$ $N_r(x)$ denotes a radius r disk centered at x. Each object from $\mathcal{N}_r(E)$ is a Euclidean r-neighborhood of some segment of E also called *r-hippodrome* or *r-offset* in the literature [4].

Thus, the IPGD problem can equivalently be formulated as follows: given a set $\mathcal{N}_r(E)$ of r-hippodromes on the plane whose underlying straight line segments form an edge set of some plane graph $G = (V, E)$, find the minimum cardinality point set C such that $C \cap N \neq \varnothing$ for every $N \in \mathcal{N}_r(E)$. In fact, C represents a set of centers of radius r disks, forming a solution to the IPGD problem. In the sequel, a set $C_0 \subset \mathbb{R}^2$ is called a *piercing set* for $\mathcal{N}_r(E)$ when $C_0 \cap N \neq \varnothing$ for all $N \in \mathcal{N}_r(E)$.

1.1 Related Work and Our Results

Settings close to the IPGD problem are originally considered in [4]. Motivated by applications from sensor monitoring for urban road networks, they explore the case in which $\mathcal{F}$ contains equal disks and E consists of (generally properly overlapping) axis-parallel segments. Their algorithms can easily be extended to the case of sets E of straight line segments with bounded number of distinct orientations.

In [11] constant factor approximation algorithms are first proposed for the IPGD problem. Namely, a 100-approximate $O(n^4 \log n)$-time algorithm is given for the problem in its general setting where E is formed by an edge set of an arbitrary plane graph. Moreover, due to applications, 68- and 54-approximate algorithms are given in [12] for special cases where E is an edge set of a generalized outerplane graph and a Delaunay triangulation respectively as well as a 23-approximation algorithm is proposed under the assumption that all pairs of non-overlapping segments from E are at the distance more than r from each other.

Let us give some definitions. Let V be a finite point set in general position on the plane. Assuming that no 4 points of V lie on any circle, a plane graph $G = (V, E)$ is called a *Gabriel* graph [14] when $[u, v] \in E$ iff intersection of V is empty with interior of the disk with diameter $[u, v]$. Under the same assumption a plane graph $G = (V, E)$ is called a *relative neighborhood* graph [6] when $[u, v] \in E$ iff $\max\{d(u,w), d(v,w)\} \geq d(u,v)$ for any $w \in V \backslash \{u, v\}$. Both types of plane graphs defined above appear in a variety of network applications. They represent convenient network topologies, simplifying routing and control in geographical (e.g. wireless) networks. They can also be applied when approximating complex networks.

In [13] faster $O(n^2)$-time 10-, 12- and 14-approximate algorithms are designed for the NP-hard [10] IPGD problem when E is being an edge set of a minimum Euclidean spanning tree, a relative neighborhood graph and a Gabriel

graph respectively. This paper extends this latter work by presenting faster $O(n^{3/2}\log^2 n)$-expected time 10-, 12- and 14-approximation algorithms for the IPGD problem for classes of minimum Euclidean spanning trees, relative neighborhood graphs and Gabriel graphs respectively.

2 Our Approximation Algorithms

2.1 Some Preliminaries

Our subquadratic $O(1)$-approximation algorithms are improved versions of the $O(n^2)$-time $O(1)$-approximation algorithms, given in [13]. The latter algorithms operate on two concepts whose definitions are given below.

Definition 1. *A subset $\mathcal{I} \subseteq \mathcal{N}_r(E)$ is called a maximal (with respect to inclusion) independent set in $\mathcal{N}_r(E)$, if $I \cap I' = \varnothing$ for any $I, I' \in \mathcal{I}$, and for any $N \in \mathcal{N}_r(E)$ there is some $I \in \mathcal{I}$ with $N \cap I \neq \varnothing$.*

Given $N \in \mathcal{N}_r(E)$, let $e(N)$ be a straight line segment such that $N_r(e(N)) = N$. Let also $\mathcal{N}_{r,e}(E) = \{N \in \mathcal{N}_r(E) : N \cap N_r(e) \neq \varnothing\}$ for $e \in E$. Of course, each maximal independent set $\mathcal{I}$ in $\mathcal{N}_r(E)$ defines a (possibly non-unique) partition of the form $\mathcal{N}_r(E) = \bigcup\limits_{I\in\mathcal{I}} \mathcal{N}_I$, where $\mathcal{N}_I \subseteq \mathcal{N}_{r,e(I)}(E)$, $I \in \mathcal{I}$; moreover, families $\mathcal{N}_I$ and $\mathcal{N}_{I'}$ are non-intersecting for distinct $I, I' \in \mathcal{I}$. It is easy to see that each maximal independent set in $\mathcal{N}_r(E)$ also induces the corresponding partition of E.

Definition 2. *Let $G = (V, E)$ be a plane graph and $\alpha > 0$ be some (r-independent) absolute constant. An edge $e \in E$ is called α-coverable with respect to E, if for any constant $\rho > 0$ one can construct at most α-point piercing set $U(\rho, e, E) \subset \mathbb{R}^2$ for $\mathcal{N}_{\rho,e}(E)$ in polynomial time with respect to $|\mathcal{N}_{\rho,e}(E)|$.*

It turns out (see Lemmas 1 and 5 in [13] for proof) that any edge of every Gabriel and relative neighborhood graph is α-coverable for some suitable positive integer α.

Lemma 1. *Any edge $e \in E$ is 12-coverable of an arbitrary subgraph $G = (V, E)$ of a relative neighborhood graph. More precisely, for any $\rho > 0$ respective piercing set $U(\rho, e, E)$ for $\mathcal{N}_{\rho,e}(E)$ can be found in $O(1)$ time.*

Lemma 2. *Any edge $e \in E$ is 14-coverable of an arbitrary subgraph $G = (V, E)$ of a Gabriel graph. Namely, for any $\rho > 0$ respective piercing set $U(\rho, e, E)$ for $\mathcal{N}_{\rho,e}(E)$ can be found in $O(1)$ time.*

Let $G = (V, E)$ be a Euclidean minimum spanning tree with a root $v_0 \in V$, $\mathrm{depth}(u) = \mathrm{depth}(u|v_0, G)$ be the (graph-theoretic) distance in G from v_0 to an arbitrary $u \in V$ and $V(u|v_0)$ be the subset of those vertices $w \in V$ such that the shortest path in G from w to v_0 (with respect to the number of its edges) passes through u. If an edge $e = [u_1, u_2] \in E$ is such that $\mathrm{depth}(u_1) = \mathrm{depth}(u_2) - 1$,

then $\operatorname{int} N_{2\Delta}(u_2) \cap (V \backslash V(u_2|v_0)) = \varnothing$, where $\Delta = d(u_1, u_2)/2$ and $\operatorname{int} N$ denotes interior of a set $N \subset \mathbb{R}^2$.

It can also be proved (see the lemma 3 from [13]) that in any subgraph of a minimum Euclidean spanning tree there always exists a 10-coverable edge.

Lemma 3. *Let $G_0 = (V_0, E_0)$ be a subgraph without isolated vertices of a Euclidean minimum spanning tree $G = (V, E)$. Let $\operatorname{depth}(\cdot|v_0)$ be a distance function on V with respect to a chosen $v_0 \in V$ as defined above. Then an edge $e = [u_1, u_2] \in E_0$ is 10-coverable with respect to E_0, if $u_2 \in \operatorname{Arg}\max_{u \in V_0} \operatorname{depth}(u)$. Besides, for any constant $\rho > 0$ the corresponding piercing set $U(\rho, e, E_0)$ of size at most 10 can be found in $O(1)$ time.*

Roughly, $O(1)$-approximate algorithms from [13] compute a partition $\mathcal{N}_r(E) = \bigcup_{I \in \mathcal{I}} \mathcal{N}_I$, defined by some maximal independent set $\mathcal{I}$ for $\mathcal{N}_r(E)$. Moreover, on the way of constructing $\mathcal{I}$, a constant sized piercing set $U(I)$ is computed for each $\mathcal{N}_I \subseteq \mathcal{N}_{r,e(I)}(E)$, $I \in \mathcal{I}$, as described in proofs of lemmas 1, 3 and 5 from [13]. Finally, it turns out that $\left\{ N_r(u) : u \in \bigcup_{I \in \mathcal{I}} U(I) \right\}$ is an $O(1)$-approximate solution to the IPGD problem.

The key to the performance gain, achieved in our algorithms over the algorithms of [13], lies in the efficient way of constructing a partition of $\mathcal{N}_r(E)$, which is induced by $\mathcal{I}$. Here, our algorithms additionally maintain a special data structure. For a given set F of non-intersecting straight line segments and a point $x \in \mathbb{R}^2$, this data structure allows to efficiently compute the segment $f \in F$, which is the nearest to x with respect to Euclidean distance. In fact, it implicitly implements a Euclidean Voronoi diagram for F. Voronoi diagram for sets of pairwise non-overlapping straight line segments is a generalization of the well-known Euclidean Voronoi diagram for point sets on the plane.

Definition 3. *Let F be a set of pairwise non-intersecting straight line segments. A Voronoi diagram $\mathcal{V}(F)$ for F is a partition of the plane into a set of open regions and their boundaries where each region represents a locus of those points, which are closer to a particular segment $f \in F$ than to any other segment of $F \backslash \{f\}$. Boundary of each region is composed of curvilinear edges and vertices. Each (relatively open) edge is the locus of points, which are equidistant from two distinct segments $f, f' \in F$ while being at the larger distance from segments of $F \backslash \{f, f'\}$. Each vertex represents a endpoint of an edge of $\mathcal{V}(F)$, which is equidistant from more than two distinct segments of F.*

Open regions, edges and vertices of $\mathcal{V}(F)$ are called Voronoi cells, edges and vertices respectively.

An additional assumption is imposed below on sets of segments for simplicity.

Definition 4. *A set F of pairwise non-overlapping straight line segments is called to be in general position if:*

1. no quadruple exists of segments from F which is touched by any single disk;
2. the set is in general position of endpoints of segments from F.

Generality of position can be achieved by a small perturbation of endpoints of E.

2.2 Implementation of Algorithms

Let $G = (V, E)$ be a plane graph and $r > 0$ be a constant, forming an input of the IPGD problem. Work of our algorithms can be split into two stages. At their first stage a partition $\mathcal{N}_r(E) = \bigcup_{I \in \mathcal{I}} \mathcal{N}_I$ is efficiently extracted, which is induced by a maximal independent set $\mathcal{I}$ in $\mathcal{N}_r(E)$. Then, during the second stage, another pass is performed over the built set $\mathcal{I}$ to construct a piercing set $U(I)$ of $\mathcal{N}_I$ for each $I \in \mathcal{I}$. Here, $U(I)$ is built in the analogous way to that done in the algorithms from [13]. Merging those piercing sets together into a point set $C = \bigcup_{I \in \mathcal{I}} U(I) \subset \mathbb{R}^2$, a set $\mathcal{C} = \{N_r(c) : c \in C\}$ is yielded as an approximate solution to the IPGD problem instance, defined by G and r. In our algorithms below, the sought partition $\mathcal{N}_r(E) = \bigcup_{I \in \mathcal{I}} \mathcal{N}_I$ is found implicitly in the form of constructing the corresponding partition of E, induced by $E' = \{e(I) : I \in \mathcal{I}\}$.

Algorithmic work at the first stage is split into n_1 phases, where $n_1 \leq \sqrt{n}$. During the ith phase a pass is performed over some part of E to iteratively grow a subset $F \subseteq E$ by adding segments from E into F one by one such that:

1. $\mathcal{N}_r(F)$ contains pairwise non-overlapping r-hippodromes;
2. the upper bound $|F| \leq \sqrt{n}$ holds.

During ith phase a special incremental data structure is applied. It allows to do the following two operations:

1. query: given a straight line segment $e \notin F$, return a segment $f \in F$ such that $N_r(e) \cap N_r(f) \neq \varnothing$ or report that $N_r(e) \cap N_r(f) = \varnothing$ for all $f \in F$; namely, in the former case the segment f is returned along with the truth value of a special indicator variable named $flag$; otherwise, $flag = False$ is returned;
2. insertion: insert a segment $e \notin F$ into F.

Here the segment e is allowed to intersect segments of F at most at their endpoints. When $F = \varnothing$, $flag = False$ is returned.

Below a pseudo-code is presented of two of our algorithms. Their input is formed by a graph G which is either a Gabriel or a relative neighborhood graph. Moreover, they contain a constant parameter $\alpha > 0$, which is specific to the class of plane graphs from which G is chosen. Here $\alpha = 14$ for the case where G is a Gabriel graph whereas $\alpha = 12$, when G is a relative neighborhood graph.

COVERING SEGMENTS WITH r-DISKS.

Input: a constant $r > 0$ and a plane graph $G = (V, E)$;
Output: an α-approximate solution $\mathcal{C}$ of radius r disks for the IPGD problem instance, defined by G and r.

---

1. set $n = |E|$, $E' := \varnothing$, $E'' := \varnothing$, $E_0 := E$ and $C := \varnothing$; // *stage 1*
2. while $|E''| \leq \sqrt{n}$, process edges of E_0 one by one: do steps 3–5 //*phase begins*
3. choose $e \in E_0$ and perform a query to the data structure built for $F = E''$;
4. if $flag = True$ and a segment $g \in E''$ are returned, set $E_g := E_g \cup \{e\}$; otherwise, when $flag = False$ is returned, insert e into E'' and set $E_e := \varnothing$;
5. set $E_0 := E_0 \backslash \{e\}$;
6. if $E_0 = \varnothing$, go to step 9; otherwise, process edges of E_0 one by one: do steps 7–8
7. choose $e \in E_0$ and perform a query to the data structure built for $F = E''$;
8. if $flag = True$ and $g \in E''$ are returned, set $E_g := E_g \cup \{e\}$ and $E_0 := E_0 \backslash \{e\}$;
9. set $E' := E' \cup E''$ and $E'' := \varnothing$;
10. if $E_0 \neq \varnothing$, go to step 2; //*phase finishes*
11. for each $e' \in E'$ repeat steps 12–13 // *stage 2*
12. construct a piercing set $U(e')$ of at most α points for $\mathcal{N}_r(E_{e'})$, applying e.g. the corresponding $O(1)$-time procedure, mentioned in lemmas 1 and 2 above;
13. set $C := C \cup U(e')$;
14. return $\mathcal{C} := \{N_r(c) : c \in C\}$ as an α-approximate solution.

As the pseudo-code above shows, in accordance with the basic algorithm of [13], the COVERING SEGMENTS WITH r-DISKS algorithm computes a maximal independent set in $\mathcal{N}_r(E)$ by iteratively growing a subset $E' \subseteq E$ such that r-hippodromes of $\mathcal{N}_r(E')$ are pairwise non-overlapping. The algorithm grows the set E' by chunks E'' of size $\sqrt{n}$ except the last chunk, where each chunk is computed in a single phase. While $|E''| \leq \sqrt{n}$, each chunk E'' is grown by performing query and insertion operations of the data structure, built on top of E'', for the remaining unprocessed segments of E_0, which are tried one by one. When $|E''| > \sqrt{n}$, those remaining segments $e \in E_0$ are removed from E_0 for which query operation returns $flag = True$. As shown below, the used randomized data structure allows to perform query operation in $O\left(\log^2 |E''|\right)$ expected time. Besides, the maintained bound $|E''| \leq \sqrt{n}$ helps keep total expected time complexity of insertion operations as low as $O(n \log n)$ in each phase.

The described organization of processing of segments of E and favourable query times of the used data structure allow to formulate the following

Theorem 1. *Suppose $G = (V, E)$ is a plane graph whose edge set is in general position. The* COVERING SEGMENTS WITH r-DISKS *algorithm is*

1. 12*-approximate when G is a subgraph of a relative neighborhood graph;*
2. 14*-approximate if G is a subgraph of a Gabriel graph.*

It admits an implementation, which works in $O\left((n + \mathrm{OPT}\sqrt{n}) \log^2 \mathrm{OPT}\right)$ expected time and $O(n)$ expected space, where OPT *is the problem optimum.*

Theorem 2. *The* COVERING SEGMENTS WITH r-DISKS *algorithm can be slightly modified to become a* 10*-approximate $O\left((n + \mathrm{OPT}\sqrt{n}) \log^2 n\right)$-expected time and $O(n)$-expected space algorithm when G is a subgraph of a minimum Euclidean spanning tree whose edge set is in general position.*

Proofs of Theorems 1 and 2 are given in Sect. 4.

It can be seen that expected complexity of the Covering segments with r-disks algorithm depends on OPT. For example, if OPT $< \sqrt{n}$, then this algorithm is of almost linear expected time complexity. Due to an obvious bound OPT $\leq n$, the algorithm has at most $O\left(n^{3/2}\log^2 n\right)$ expected time complexity.

To compare performance of the Covering segments with r-disks algorithm with the basic algorithm from [13] it is enough to observe that the former algorithm would coincide with the latter one if the restriction $|E''| = 1$ was imposed on all chunks E''. This restriction leads to the $O(n\text{OPT})$ expected and worst case time complexity: here one has at most OPT singleton chunks as well as constant query and insertion times.

Key to the achieved performance gain in our algorithms over the related work lies in efficient implementation of query operations of the data structure, built on top of E'', within the Covering segments with r-disks algorithm. This core data structure is described in the section below.

3 Description of the Core Data Structure

3.1 Implementing the Query Operation with a Few Nearest Neighbor Queries

Suppose a set F is given of pairwise non-intersecting straight line segments on the plane. Besides, let e be a straight line segment such that its intersection with each segment of F (if it is nonempty) can only be at the common endpoint. The query operation for e on F can in theory be implemented by computing the segment $f \in F$, being the closest to e, and checking if $N_r(e) \cap N_r(f) \neq \varnothing$. It means that the query operation admits its implementation by performing a segment nearest neighbor query operation on F.

Unfortunately, this approach fails to efficiently work within the Covering segments with r-disks algorithm. In fact, there is a variety of incremental algorithms to maintain efficient point nearest neighbor queries for sets of point sites. The most efficient known data structure [2] provides $O(\log^2 |F|)$ expected query time and $O(\log^4 |F|)$ expected insertion time. However, there is a lack of available incremental data structures to maintain segment nearest neighbor queries. The only available incremental randomized data structure to work with straight line segment sites and segment queries, which we are able to find, implicitly maintains their segment Voronoi diagram [7][1].

In this data structure segment nearest neighbor queries cost $O(\log^2 |F| + t)$ expected time, where t is the number of segments of F, sharing a Voronoi edge with e in the Voronoi diagram $\mathcal{V}(F \cup \{e\})$. Thus, t can be $\Omega(|F|)$ on average in general. Therefore with this idea one can not guarantee for the query to have sublinear expected time complexity and, as a consequence, for the Covering

[1] Its C++ implementation is built in the CGAL library (see https://www.cgal.org/), providing robust geometric computations.

SEGMENTS WITH r-DISKS algorithm to have subquadratic expected time complexity.

Luckily, query operations, performed in the COVERING SEGMENTS WITH r-DISKS algorithm, can efficiently be implemented using a constant number of nearest neighbor queries of the special type in which sites are straight line segments whereas query objects are points.

Nearest Neighbor Query. Given a point $x \in \mathbb{R}^2$, find a segment $f \in F$, being the closest to x among segments of F with respect to Euclidean distance between points and segments.

Lemma 4. *Let e be a query straight line segment in a query operation for a set F of pairwise non-intersecting striaght line segments on the plane, being in general position. Suppose the following assumptions are hold:*

1. *$F \cup \{e\}$ is a subset of edges of a Gabriel graph;*
2. *$N_r(f) \cap N_r(g) = \varnothing$ for any distinct $f, g \in F$.*

Then, the query operation can be implemented using at most 14 nearest neighbor query operations and at most 14 operations to check if two r-hippodromes intersect.

Proof. Let $\Delta = d(x, y)/2$, where x and y are endpoints of e. First, nearest neighbor query operation is applied for both x and y, returning segments f_x and f_y of F respectively. If either $N_r(f_x) \cap N_r(e) \neq \varnothing$ or $N_r(f_y) \cap N_r(e) \neq \varnothing$, it is done. Otherwise, consider three cases.

CASE 1. Let $\Delta \leq r$. Let also z_1 and z_2 be points at the intersection $\operatorname{bd} N_{2r}(x) \cap \operatorname{bd} N_{2r}(y)$, where $\operatorname{bd} N$ denotes boundary of a set $N \subset \mathbb{R}^2$. Perform another two nearest neighbor queries for z_1 and z_2. Let $f_{z_i} \in F$ be closest to z_i. If $N_r(e) \cap N_r(f_{z_{i_0}}) \neq \varnothing$ for some $i_0 \in \{1, 2\}$, then it is done. If not, it can be proved that $N_r(e) \cap N_r(f) = \varnothing$ for all $f \in F$.

Indeed, suppose, in contrary, that $f_0 \in F$ exists such that $N_r(e) \cap N_r(f_0) \neq \varnothing$ or, equivalently, f_0 intersects $N_{2r}(e)$. Obviously, the case $f_0 \cap \bigcup_{u \in \{x,y\}} N_{2r}(u) \neq \varnothing$ is impossible. Therefore f_0 must intersect $N_{2r}(e) \setminus \bigcup_{u \in \{x,y\}} N_{2r}(u)$. Of course, if f_0 intersects $N_{2r}(e)$ inside the same half-plane H_{i_0}, bounded by the straight line through e, as that in which the point z_{i_0} lies for some $i_0 \in \{1, 2\}$, then both f_0 and $f_{z_{i_0}}$ must intersect $N_{\sqrt{8r^2 - 4r\sqrt{4r^2 - \Delta^2}}}(z_{i_0})$. As $\Delta \leq r$, it implies that $H_{i_0} \cap N_{2r}(e) \setminus \bigcup_{u \in \{x,y\}} N_{2r}(u)$ is covered by $N_{2r}(f_{z_{i_0}})$ and, therefore $N_r(f_{z_{i_0}}) \cap N_r(f_0) \neq \varnothing$, a contradiction with the assumption 2. Thus, at most 4 nearest neighbor query operations are enough in the considered case.

CASE 2. If $r < \Delta \leq 2r$, split e into two subsegments of length Δ and apply the same technique for each subsegment as in the previous case. Here at most 7 nearest neighbor query operations are enough.

CASE 3. Suppose that $\Delta > 2r$. Let z be the midpoint of e. Recall that $N_\Delta(z)$ does not contain endpoints of segments of $F \cup \{e\}$ in its interior according to the

assumption 1. Let z_1 and z_2 be projections of 4 points from $\mathrm{bd}\, N_\Delta(z) \cap \mathrm{bd}\, N_{2r}(e)$ onto e, where $z_1 \in [x, z]$ and $z_2 \in [y, z]$. Below it is proved that length of $[z_1, x]$ is less than $2r$. Indeed, $d(x, z_1) = \Delta - \sqrt{\Delta^2 - 4r^2} \leq 2r$. Moreover, $d(x, z_1) \leq r$ when $\Delta > \frac{5r}{2}$.

From [13] (see Sect. 2 and the Lemma 5 therein) it follows that if some $f_0 \in F$ intersects $N_{2r}(e)$, then f_0 must intersect $N_{2r}(e) \backslash N_\Delta(z)$. Therefore at most 14 nearest neighbor query operations are enough for $2r < \Delta \leq \frac{5r}{2}$ and 8 nearest neighbor query operations are sufficient when $\Delta > \frac{5r}{2}$.

Any relative neighborhood graph and minimum Euclidean spanning tree is a subgraph of a Gabriel graph. Therefore the following corollary holds.

Corollary 1. *Let e be a query straight line segment. Suppose the assumptions are hold:*

1. *$F \cup \{e\}$ is a subset of edges of either a relative neighborhood graph or a minimum Euclidean spanning tree;*
2. *$N_r(f) \cap N_r(g) = \varnothing$ for any distinct $f, g \in F$.*

Then, the query operation for e on F can be implemented using at most 14 *nearest neighbor query operations and at most* 14 *operations to check if two r-hippodromes intersect.*

Due to the Lemma 4 and the Corollary 1, the data structure from [7] can be used to implement query and insertion operations, performed within the Covering segments with r-disks algorithm. To the best of our knowledge, this data structure has the most efficient implementation of point nearest neighbor queries for segment sites. Being incorporated into our algorithm, it implicitly stores a Voronoi diagram $\mathcal{V}(E'')$ of the set E''. Its performance is summarized in the following lemma (see works [1,7–9,15] for proofs).

Lemma 5. *Let F be a set of pairwise non-intersecting straight line segments in general position on the plane. A randomized data structure can be built incrementally in $O(|F|^2 \log |F|)$ expected time and $O(|F|)$ expected space cost such that:*

1. *given a point $x \in \mathbb{R}^2$, nearest neighbor query for x and F can be performed in $O(\log^2 |F|)$ expected time;*
2. *given a segment $e \notin F$ such that $\mathcal{N}_r(F \cup \{e\})$ contains only pairwise non-overlapping r-hippodromes, insertion of e into F can be done in $O(|F| \log |F|)$ expected time.*

In distinction to the data structure from [7] only a single type of randomization is applied in the implementation of the used data structure, which is related to generating a random hierarchy of nested subsets of F.

In [7] another type of randomization is also used implied by a random order of insertion of segments into F : i.e. it is assumed that the order of insertion of segments into F is a random permutation on F and all insertion orders are equally likely. Applying both randomization types allows to reduce expected time

complexity of insertion operations to $O(1)$. In the COVERING SEGMENTS WITH r-DISKS algorithm this favourable symmetry of insertion of segments into F can not be guaranteed even if E is preliminarily randomly shuffled at the algorithm step 1.

4 Proofs of Theorems 1 and 2

4.1 Proof of the Theorem 1

Proof. In each phase (steps 2–10 of the COVERING SEGMENTS WITH r-DISKS algorithm) query operations are performed for at most n segments of E. Therefore running these operations takes $O(n \log^2 \mathrm{OPT})$ expected time in each phase due to the Lemma 4, the Corollary 1 and the Lemma 5. As $|E''| \leq \min\{\sqrt{n}, \mathrm{OPT}\}$, at most $\min\{\sqrt{n}, \mathrm{OPT}\}$ insertions are done in each phase. Thus, insertion operations take $O(\min\{n, \mathrm{OPT}^2\} \log \mathrm{OPT})$ expected time, again, due to the Lemma 5. There are at most $\frac{\mathrm{OPT}}{\sqrt{n}} + 1$ phases in the algorithm as $\mathrm{OPT} \leq n$. Steps 11–13 require $O(\mathrm{OPT})$ time in view of Lemmas 1 and 2. Thus, total expected complexity of the algorithm is of the order

$$O\left(n\left(\frac{\mathrm{OPT}}{\sqrt{n}} + 1\right) \log^2 \mathrm{OPT}\right),$$

or of the order $O\left((n + \mathrm{OPT}\sqrt{n}) \log^2 \mathrm{OPT}\right)$. Its expected space cost is $O(n)$.

4.2 Proof of the Theorem 2

Proof. One can maintain an $O(n)$ sized search tree (see e.g. the Chap. 13 of [3]) to report an edge of E_0 in $O(\log n)$ time at the algorithm step 3, which is incident to a vertex, being the most distant from some fixed vertex of G. This search tree can be preliminary created in $O(n \log n)$ time by performing a breadth-first search over G. Each time when a segment is removed from E_0 at steps 5 and 8 of the COVERING SEGMENTS WITH r-DISKS algorithm, the corresponding node is removed from the tree in $O(\log n)$ time. Thus, first, n insertions in a row are performed into the tree; second, n consecutive deletions are done.

5 Conclusion

The paper presents randomized subquadratic small constant factor approximation algorithms for three special cases of the problem of intersecting a given set of straight line segments on the plane with the least number of identical disks of potential interest in facility location. When $\mathrm{OPT} = O(\sqrt{n})$, these algorithms have almost linear expected time complexity of $O(n \log^2 n)$, where OPT is the problem optimum. In the general case their expected complexity is $O\left((n + \mathrm{OPT}\sqrt{n}) \log^2 n\right)$, i.e. with $\frac{\mathrm{OPT}}{\sqrt{n}}$ superlinear multiplicative overhead. Approximation factors of the proposed algorithms are still prohibitively high to be practical as being only theoretically guaranteed upper bounds on approximation ratios of the algorithms in the worst case. In the follow-up paper their actual approximation factors will be explored for real-world facility location problems.

References

1. Boissonnat, J.D., Wormser, C., Yvinec, M.: Curved Voronoi diagrams. In: Boissonnat, J.D., Teillaud, M. (eds.) Effective Computational Geometry for Curves and Surfaces, pp. 67–116. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-33259-6_2
2. Chan, T.M.: Dynamic geometric data structures via shallow cuttings. In: Proceedings of the 35th International Symposium on Computational Geometry, SoCG 2019, pp. 24:1–24:13 (2019)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press, Cambridge (2009)
4. Dash, D., Bishnu, A., Gupta, A., Nandy, S.: Approximation algorithms for deployment of sensors for line segment coverage in wireless sensor networks. Wirel. Netw. **19**(5), 857–870 (2012)
5. Devillers, O.: The Delaunay hierarchy. Int. J. Found. Comput. Sci. **13**, 163–180 (2002)
6. Jaromczyk, J., Toussaint, G.: Relative neighborhood graphs and their relatives. Proc. IEEE **80**(9), 1502–1517 (1992)
7. Karavelas, M.I.: A robust and efficient implementation for the segment Voronoi diagram. In: Proceedings of the 1st International Symposium on Voronoi Diagrams in Science and Engineering, Tokyo, pp. 51–62 (2004)
8. Karavelas, M., Yvinec, M.: The Voronoi Diagram of Convex Objects in the Plane. Research report. RR-5023, INRIA. 27 p. (2003)
9. Klein, R., Mehlhorn, K., Meiser, S.: Randomized incremental construction of abstract Voronoi diagrams. Comput. Geom. **3**(3), 157–184 (1993)
10. Kobylkin, K.: Stabbing line segments with disks: complexity and approximation algorithms. In: van der Aalst, W.M.P., et al. (eds.) AIST 2017. LNCS, vol. 10716, pp. 356–367. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73013-4_33
11. Kobylkin, K.: Constant factor approximation for intersecting line segments with disks. In: Battiti, R., Brunato, M., Kotsireas, I., Pardalos, P.M. (eds.) LION 12 2018. LNCS, vol. 11353, pp. 447–454. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05348-2_39
12. Kobylkin, K.: Efficient constant factor approximation algorithms for stabbing line segments with equal disks, CoRR abs/1803.08341, 31 p. (2018). https://arxiv.org/pdf/1803.08341.pdf
13. Kobylkin, K., Dryakhlova, I.: Approximation algorithms for piercing special families of hippodromes: an extended abstract. In: Khachay, M., Kochetov, Y., Pardalos, P. (eds.) MOTOR 2019. LNCS, vol. 11548, pp. 565–580. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22629-9_40
14. Matula, D., Sokal, R.: Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. Geogr. Anal. **12**(3), 205–222 (1980)
15. Motwani, R., Raghavan, P.: Randomized Algorithms, 476 p. Cambridge University Press, Cambridge (1995)