Theoretical Computer Science 591 (2015) 28–59

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The searchlight problem for road networks

Dariusz Dereniowski [a,1], Hirotaka Ono [b], Ichiro Suzuki [c,2], Łukasz Wrona [d], Masafumi Yamashita [e], Paweł Żyliński [f,*,3]

[a] Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
[b] Department of Economic Engineering, Kyushu University, 6-19-1, Hakozaki, Fukuoka, 812-8581, Japan
[c] Department of Electrical Engineering and Computer Science, University of Wisconsin–Milwaukee, WI 53201-0784, USA
[d] Department of Algorithms and System Modeling, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 80-233 Gdańsk, Poland
[e] Department of Computer Science and Communication Engineering, Kyushu University, 744, Motooka, Fukuoka, 819-0395, Japan
[f] Institute of Informatics, University of Gdańsk, 80-952 Gdańsk, Poland

ARTICLE INFO

Article history:
Received 20 May 2013
Received in revised form 21 April 2015
Accepted 23 April 2015
Available online 5 May 2015
Communicated by D. Peleg

Keywords:
The searchlight problem
Graph searching
Lines
Line segments
Grids

ABSTRACT

We consider the problem of searching for a mobile intruder hiding in a road network given as the union of two or more lines, or two or more line segments, in the plane. Some of the intersections of the road network are occupied by stationary guards equipped with a number of searchlights, each of which can emit a single ray of light in any direction along the lines (or line segments) it is on. The goal is to detect the intruder, that is, to illuminate its location. Guards may alter the direction in which they aim a searchlight, but need to switch it off for some finite time interval to effect the change. In contrast, the intruder may move with arbitrary speed along the network (but cannot pass guards) and exploit this time interval to recontaminate previously illuminated sections of the network. For various classes of road networks characterized by the number n of lines (or line segments) comprising it and the number g ($\leq n-1$) of possible locations of guards (fixed in advance and guaranteed to give complete coverage), we present several upper and lower bounds on the worst-case number of searchlights, each placed at one of the guard positions, required to successfully search a given road network. In particular, we prove the following results:

1. $\min\{2g-1, n-2\}$ searchlights are sometimes necessary and $\min\{\frac{7}{3}g, n\}-1$ are always sufficient for searching a road network given as the union of n lines;
2. $\Omega(g \cdot \log \frac{n}{g})$ searchlights are sometimes necessary and $O(g^2 \cdot \log n)$ searchlights are always sufficient for searching a road network given as the union of n line segments, and
3. at most one searchlight per guard position, and hence a total of at most g searchlights, is always sufficient for searching a road network given as the union of axis-aligned lines or line segments.

* Corresponding author.
E-mail addresses: deren@eti.pg.gda.pl (D. Dereniowski), hirotaka@econ.kyushu-u.ac.jp (H. Ono), suzuki@uwm.edu (I. Suzuki), lukasz.wrona@eti.pg.gda.pl (Ł. Wrona), mak@csce.kyushu-u.ac.jp (M. Yamashita), zylinski@inf.ug.edu.pl (P. Żyliński).

[1] Partially supported by MNiSW Grant No. N206 379337 (2009–2011).
[2] Supported in part by UWM Research Growth Initiative.
[3] Partially supported by MNiSW Grant No. N516 196437 (2009–2012).

http://dx.doi.org/10.1016/j.tcs.2015.04.026

The proofs of the upper bounds induce algorithms for generating a search schedule for detecting the intruder using the claimed number of searchlights.

1. Introduction

Assume that a mobile intruder capable of moving continuously and arbitrarily fast is hiding in the streets of a dark city, where each street is modeled as a line or a line segment in the plane. Suppose that a number of stationary guards have been placed on the streets, each equipped with one or more searchlights, where a searchlight can be aimed in any street direction (from the guard position) and emit a single ray of light in that direction. The objective of the guards is to detect the intruder using the searchlights, where the intruder is considered *detected* at the moment he is illuminated by one of the searchlights or he reaches a position where a guard is located.

Formally, for $n \geq 2$, let $\mathcal{L} = \{L_1, L_2, \ldots, L_n\}$ be a set of n distinct lines, or n distinct (open or closed) line segments, in the plane, such that their union $\bar{\mathcal{L}} = L_1 \cup L_2 \cup \cdots \cup L_n$ is connected. In the former case we assume that no two collinear line segments in $\mathcal{L}$ intersect each other. For now, we continue the discussion assuming the L_i's are lines. All definitions and assumptions naturally carry over to the case of line segments without any change. We call $\mathcal{L}$ a *road network*. A point p is *visible* from a point q in $\bar{\mathcal{L}}$ if p and q lie on a common line in $\mathcal{L}$. A finite set of points $V \subseteq \bar{\mathcal{L}}$ is a *guard set* of $\mathcal{L}$ if every point in $\bar{\mathcal{L}}$ is visible from at least one point in V. The points in V are called *guards*. Hereafter we assume guards are distinct points and are always placed at intersections of lines in $\mathcal{L}$, since any guard set can be transformed into another without increasing the size by relocating every guard lying on a single line to a nearest intersection. The size γ of a smallest guard set of $\mathcal{L}$ is called the *guard number* of $\mathcal{L}$. Obviously, $1 \leq \gamma \leq n-1$ since $\bar{\mathcal{L}}$ is connected. Given a guard set V of $\mathcal{L}$, where $\gamma \leq |V| = g \leq n-1$, we call the pair $\mathcal{A} = (\mathcal{L}, V)$ an (n, g)-*arrangement*, or simply, an *arrangement*.

A guard $p \in V$, placed at an intersection of k lines, can have one or more *searchlight* that can emit a single ray of "light" that can be aimed in the direction of any of the $2k$ half-lines that meet at p and "illuminate" the points on it.[4] Each searchlight can be aimed in only one direction at a time, and we assume that the direction of the searchlight cannot be changed instantaneously. One way to interpret this assumption is that F has to be "turned off" while it changes directions. See Fig. 1 for an illustration of this assumption. In the following we often use the term "rotate" to refer to the action of changing direction of a searchlight.

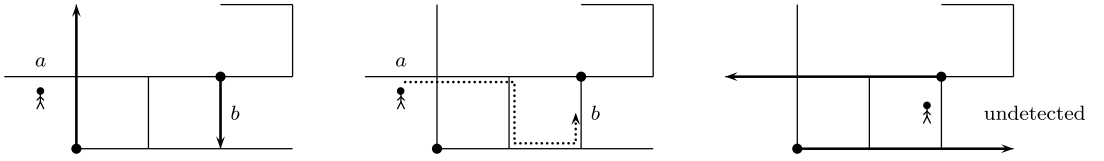

Fig. 1. Two searchlights shown on the left are rotated simultaneously as shown on the right. As is shown in the middle, while the searchlights are "turned off" during the rotation, an intruder hiding in a can move to b without being detected and "recontaminate" b.

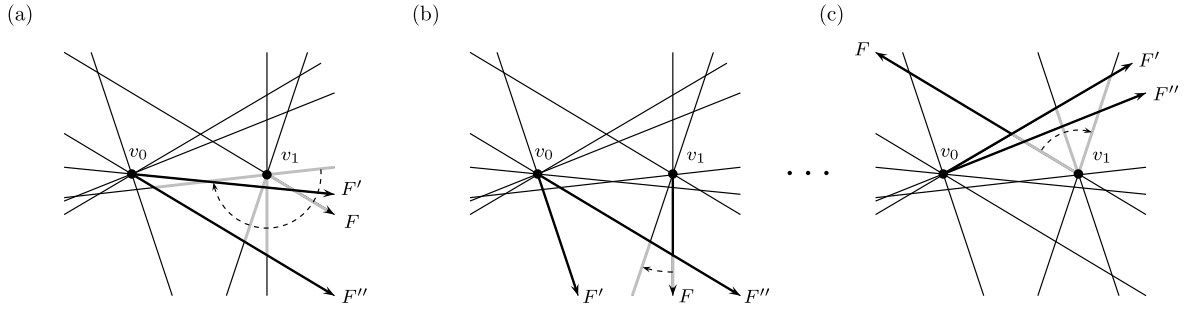

Fig. 2. A sample search strategy for an $(8, 2)$-arrangement. The gray line components searched in (a), together with the half-line illuminated with F', are not recontaminated during the rotation of F', due to the choice of the first wedge and the assumption that the intruder cannot pass through v_0 and v_1 without being detected.

A *searching strategy* for $\mathcal{A} = (\mathcal{L}, V)$ using a given set of searchlights placed at guard positions in V specifies, as a function of time, the directions in which the searchlights are aimed. A searching strategy *successfully searches* (or *clears*) $\mathcal{L}$ if it is not

[4] If $\mathcal{L}$ consists of line *segments*, then there will be *up to* $2k$ possible directions of the searchlight.

possible for the intruder in $\bar{\mathcal{L}}$ to avoid detection during its execution regardless of his moves, where *detection* occurs at the moment the intruder is illuminated by one of the searchlights or he reaches a guard. We denote by $s(\mathcal{A})$, or $s(\mathcal{L}, V)$, the smallest total number of searchlights necessary for the guards in V to successfully search $\mathcal{L}$.

See Fig. 2 for an example of successfully searching an $(8, 2)$-arrangement using three searchlights, F' and F'' placed at guard v_0 and F placed at v_1. The example illustrates a strategy that we often use, termed *rotational wedge sweep*, in which the rays of F' and F'' successively form wedges around v_0 not containing v_1 and the ray of F searches the segments and half-lines, referred to as *line components*, in the wedges visible from v_1.[5] First, as shown in Fig. 2(a), F' and F'' are aimed respectively at the first and second half-lines around v_0 clockwise from v_1 to form the first wedge not containing v_1, and F searches all line components between v_1 and F', and between F' and F''. Note that as long as F'' remains on, the line components just searched will remain *clear*, that is, guaranteed to be free of the intruder, even if F' is turned off, because the intruder cannot pass through v_1. F' is now rotated clockwise to illuminate the third half-line around v_0, so that F' and F'' form the second wedge around v_0, as shown in Fig. 2(b). The line components in that wedge are searched by F, and while F' remains on, F'' is rotated to illuminate the fourth half-line around v_0 so that F' and F'' form the third wedge around v_0 without recontaminating any of the line components searched so far between v_1 and F'. The process continues until the line components in the last wedge and the segments between that wedge and v_1 are searched by F, as shown in Fig. 2(c).

The objectives of the *searchlight problem* for an (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ are:

1. To determine $s(\mathcal{A})$.
2. To determine a placement of $s(\mathcal{A})$ searchlights in the guard positions in V and compute a searching strategy for $\mathcal{A}$ using these searchlights.

In this paper we mainly focus on the worst-case scenario for the problems above in the following sense. For $n \geq 2$ and $1 \leq g \leq n-1$, define $s(n, g)$ to be the maximum of $\mathcal{A}$, over all (n, g)-arrangements $\mathcal{A} = (\mathcal{L}, V)$. That is, $s(n, g)$ is the worst-case total number of searchlights necessary to successfully search any (n, g)-arrangement. Our main results include the following upper and lower bounds on $s(n, g)$.

1. If $\mathcal{L}$ is a set of lines, then $\min\{2g-1, n-2\} \leq s(n, g) \leq \min\{\frac{7}{3}g, n\} - 1$. (Section 2.)
2. If $\mathcal{L}$ is a set of line segments, then $s(n, g) = \Omega(g \cdot \log \frac{n}{g})$ and $s(n, g) = O(g^2 \cdot \log n)$. (Subsections 3.1 and 3.2, respectively.)
3. If $\mathcal{L}$ is a set of axis-aligned lines or line segments, then it is sufficient to have at most one searchlight per guard position, and hence, $s(n, g) \leq g$. (Subsections 2.2 and 3.3, respectively.)

We prove the upper bounds constructively, by presenting in each case an algorithm for successfully searching any (n, g)-arrangement using a number of searchlights matching the bound.[6]

Since a (connected) arrangement of lines/line segments can be thought of as a polygon with holes, representing a region of intersecting thin corridors, our problem can be considered as a variant of the searchlight problem in polygons introduced by Sugihara et al. [25] and then studied in [10–12,18,22,26,27,32–34,37]. A wider perspective locates our problem as a variant of the art gallery problem, originally posed by Klee in 1973 as the question of determining the minimum number of guards sufficient to see every point of the interior of a simple polygon; for more details, see a book by O'Rourke [19], a survey article by Shermer [21], and a book chapter by Urrutia [31]. In particular, our searchlight problem is a variant of the art gallery problem for lines and line segments, which was first formulated by Ntafos [17], and then extensively studied in its several variants in [1–3,6,7,9,14,19,29,30,36].

Furthermore, our problem is also related to the searchlight problem for graphs introduced and studied in [38–40]. The difference between the two lies in the visibility model: in the searchlight problem for graphs visibility is restricted to within the neighborhood of a vertex where a searchlight is located, while visibility in our problem is based on a straight line of sight within a line or a line segment. An analogous vision-based pursuit–evasion problem in grids, where a number of pursuers have to capture an evader, has been studied in [4,5,16,23,24,28]. The pursuit–evasion problem in grids is itself a variant of the well-known *graph search problem* [13,15,20], to mention just a few; see [8] for a beautiful survey of known results and variations.

The rest of the paper is organized as follows. In Section 2, we provide lower and upper bounds on $s(n, g)$ for the case in which $\mathcal{L}$ is a set of lines. Next, in Section 3, we provide lower and upper bounds on $s(n, g)$ for the case in which $\mathcal{L}$ is a set of line segments. Finally, in Section 4, some open problems are discussed, in particular, for an (n, g)-arrangement $(\mathcal{L}, V)$ of *axis-aligned* line segments, called a *grid*, we introduce another parameter: the switch number $sn(\mathcal{L}, V)$, defined as the minimum number of steps required to search $(\mathcal{L}, V)$ by g guards with single searchlights located at guards in V.

For the standard terminology of graph theory used in this paper, we refer the reader to [35].

[5] There may be line components in a wedge not visible from v_1 if $|V| \geq 3$.

[6] It is out of the scope of the paper to develop efficient implementations of our algorithms or to investigate the time and space complexity of deciding whether the given arrangement can be searched using k searchlights. We leave these issues for future study.

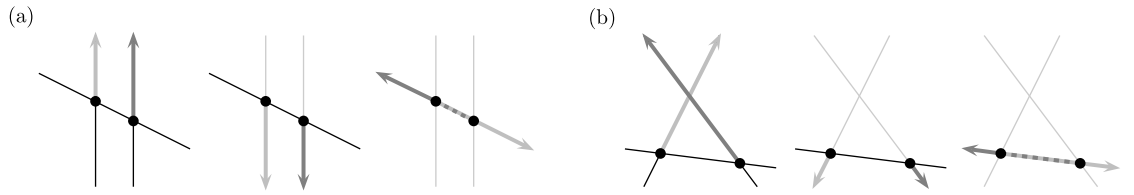

Fig. 3. Both $(3,2)$-arrangements can be searched using two searchlights, by illuminating all half-lines and the segment between the guards in any order, except in (b) the two half-lines that intersect each other must be illuminated simultaneously.

2. Bounds on $s(n, g)$ for regions composed of lines

In this section, we prove lower and upper bounds on $s(n, g)$ for the case in which $\mathcal{L}$ is a set of lines (as opposed to line segments). We start with a simple observation.

Observation 2.1. *$s(n, 1) = 1$, $s(3, 2) = 2$, and $s(n, n-1) \geq n-1$ for $n \geq 4$.*

Proof. In an $(n, 1)$-arrangement the n lines pass through a common intersection. So one full rotation of a searchlight placed at the intersection clears all $2n$ half-lines. A $(3, 2)$-arrangement consists of either a line intersecting two parallel lines with a guard placed at each intersection, or three non-parallel lines with guards placed at any two intersections. Both can be successfully searched using two searchlights as illustrated in Fig. 3. The third claim is established by an arrangement that consists of one horizontal line and $n-1$ vertical lines, and a guard at each of the $n-1$ intersections, where obviously, every guard needs a searchlight. □

2.1. Lower bounds

Consider the $(4, 2)$-arrangement $\mathcal{A}_2 = (\mathcal{L}_2, \{v_0, v_1\})$ illustrated in Fig. 4(a), where two lines, passing through v_0 and forming a wedge (i.e., an angular region) not containing v_1, intersect two lines passing through v_1 and forming a wedge not containing v_0. We call the part of $\bar{\mathcal{L}}_2$ lying in the shaded region and forming a four-edge cycle *block*. $\mathcal{A}_2$ is an instance of a more general construction, a $(2g, g)$-arrangement $\mathcal{A}_g = (\mathcal{L}_g, \{v_0, \ldots, v_{g-1}\})$, $g \geq 2$, illustrated in Fig. 4(b) where (i) two lines pass through each v_i, and (ii) for every pair of i and j, $i \neq j$, the two lines passing through v_i and the two lines passing through v_j form a block intersected by no other lines passing through any other guard.

Clearly $s(\mathcal{A}_2) \geq 3$, because we need at least one searchlight at each guard and the block cannot be cleared unless one additional searchlight is placed at v_0 or v_1. Similarly, to clear all blocks of $\mathcal{A}_g$ at least two searchlights must be placed at every guard except possibly one. Therefore $s(\mathcal{A}_g) \geq 2g-1$. Since we can add any number of lines passing through v_0 to $\mathcal{A}_g$ without reducing the number of searchlights necessary to search, $s(n, g) \geq 2g-1$ if $n \geq 2g$.

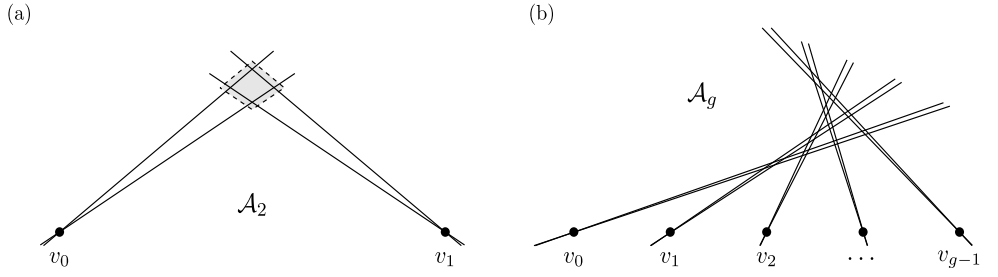

Fig. 4. (a) $(4, 2)$-Arrangement $\mathcal{A}_2 = (\mathcal{L}_2, \{v_0, v_1\})$ that requires at least three searchlights. (b) $(2g, g)$-Arrangement $\mathcal{A}_g = (\mathcal{L}_g, \{v_0, \ldots, v_{g-1}\})$, $g \geq 2$.

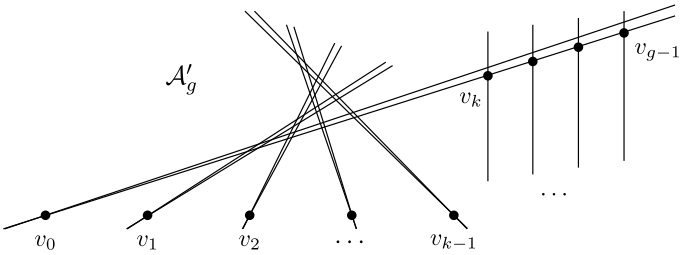

Fig. 5. An (n, g)-arrangement $\mathcal{A}'_g$ that requires at least $n-2$ searchlights, where $g+2 \leq n < 2g$ and $k = n-g$.

Assume now that $g+2 \leq n < 2g$, in particular, let $n = 2g - j$, for some j, $1 \leq j \leq g-2$. Then, we have $n = 2k + j$, and $g = k + j$, where $k = n - g \geq 2$. Now, we construct the $(2k, k)$-arrangement $\mathcal{A}_k = (\mathcal{L}_k, \{v_0, \ldots, v_{k-1}\})$, in the manner described above, which requires at least $2k-1$ searchlights. Next, we add j vertical lines that cross lines in $\mathcal{L}_k$ to the right of v_{k-1} and j guards $v_k, \ldots, v_{g-1}$ at their intersections with one of the lines passing through v_0. One can observe that for the resulting (n, g)-arrangement $\mathcal{A}'_g$ (see Fig. 5 for an illustration), the number $s(\mathcal{A}'_g)$ of required searchlights is at least $2k + j - 2 = n - 2$: at least one searchlight for each of $v_0, v_k, \ldots, v_{g-1}$ and two searchlights for all but one in $v_1, \ldots, v_{k-1}$.

Based on these observations and Observation 2.1, we obtain the following theorem.

Theorem 2.2. *For* $1 \leq g \leq n-1$,

$$s(n,g) \geq \begin{cases} 2g-1 & \text{if } 1 \leq g \leq \frac{n}{2}; \\ n-2 & \text{if } \frac{n}{2} < g \leq n-2; \\ n-1 & \text{if } g = n-1. \end{cases}$$

Thus in particular, $s(n, 2) \geq 3$ for $n \geq 4$. We also observe that the search strategy given earlier for the $(8, 2)$-arrangement in Fig. 2, where two searchlights placed at a guard successively form wedges and one searchlight placed at the other guard clears the segments in the wedges, can be used to successfully search any $(n, 2)$-arrangement for $n \geq 4$. Therefore we have:

Corollary 2.3. $s(n, 2) = 3$ *for* $n \geq 4$.

2.2. Upper bounds in terms of guard degrees

Let $\mathcal{A} = (\mathcal{L}, V)$ be an (n, g)-arrangement of lines. For any $v \in V$, we regard every line $L \in \mathcal{L}$ that v lies on as two half-lines that meet at v, and denote by $\mathcal{L}(v)$ the set of all such half-lines that meet at v. We call $\deg_{\mathcal{A}}(v) = |\mathcal{L}(v)|$ the *degree* of v. Since v is an intersection of two or more lines in $\mathcal{L}$, $\deg_{\mathcal{A}}(v)$ is even and at least 4. Next, let $\Delta(\mathcal{A}) = \max_{v \in V} \deg_{\mathcal{A}}(v)$; $\Delta(\mathcal{A})$ is called the *degree* of $\mathcal{A}$. Finally, let $\mathcal{P} \subset \mathcal{L}$ be a set of parallel lines in $\mathcal{L}$ that together have the largest number of guards on them among all sets of parallel lines; denote this number by $\Gamma(\mathcal{A})$. We have the following theorem.

Theorem 2.4. *For any* (n, g)*-arrangement* $\mathcal{A} = (\mathcal{L}, V)$ *of lines, we have*

$$s(\mathcal{A}) \leq \min\left\{\frac{1}{2}\sum_{v \in V} \deg_{\mathcal{A}}(v) - \Gamma(\mathcal{A}), n-1\right\}.$$

Before we present the proof, let us point out that if $\mathcal{L}$ is a set of axis-aligned lines[7] (and hence $\deg_{\mathcal{A}}(v) = 4$ for all $v \in V$ and $\Gamma(\mathcal{A}) = g$), then the inequality $s(\mathcal{A}) \leq \frac{1}{2}\sum_{v \in V} \deg_{\mathcal{A}}(v) - \Gamma(\mathcal{A})$ implies that it is sufficient to place one searchlight at each guard for searching $\bar{\mathcal{L}}$ (see Corollary 2.5(d)).

Proof. Let $\mathcal{P}$ be a set of parallel lines in $\mathcal{L}$ that together have $\Gamma(\mathcal{A})$ guards on them. Without loss of generality assume that the lines in $\mathcal{P}$ are vertical. At each guard $v \in V$, place $\deg_{\mathcal{A}}(v)/2 - 1 \geq 1$ searchlights if v belongs to a vertical line in $\mathcal{P}$, and place $\deg_{\mathcal{A}}(v)/2 \geq 2$ searchlights otherwise. Note that each v has as many searchlights as there are non-vertical lines passing through it, and hence, all half-lines extending toward the left from v can be illuminated simultaneously, and all half-lines extending toward the right from v can be illuminated simultaneously. The total number of searchlights is $\frac{1}{2}\sum_{v \in V} \deg_{\mathcal{A}}(v) - \Gamma(\mathcal{A})$. Let $v_0, v_1, \ldots, v_{g-1}$ be the guards in V sorted in left-to-right order (with ties broken arbitrarily). We search $\mathcal{A} = (\mathcal{L}, V)$ as follows. Imagine that we sweep the plane from left to right using a vertical line X, starting from a position to the left of v_0 and ending at a position to the right of v_{g-1}. Before the sweep starts, at every guard v_i aim all searchlights toward the left so that all half-lines extending toward the left from v_i are illuminated simultaneously. This ensures that all half-lines that cross X are illuminated simultaneously, as is shown in Fig. 6(a), and hence, no undetected intruder exists in the region to the left of X. During the sweep, each time X reaches a guard v_i, we rotate the searchlights at v_i to first (i) illuminate the vertical half-lines that meet at v_i if v_i lies on a vertical line in $\mathcal{P}$ (in order to detect an intruder hiding in the vertical line), and then (ii) illuminate simultaneously all half-lines extending toward the right from v_i. This, together with the assumption that no intruder can pass through a guard position, ensures that (i) no undetected intruder can exists on X, and (ii) no undetected intruder, who initially lies to the right of X, can cross X and enter the region to the left of X during the sweep. Note that at any time during the sweep, if no guard lies on X, then all half-lines that cross X are illuminated simultaneously as is shown in Fig. 6(b).

[7] The x- and y-axes are considered "horizontal" and "vertical", respectively. We use "left", "right", "down" and "up", respectively, to refer to the $-x$, $+x$, $-y$ and $+y$ directions in the plane.

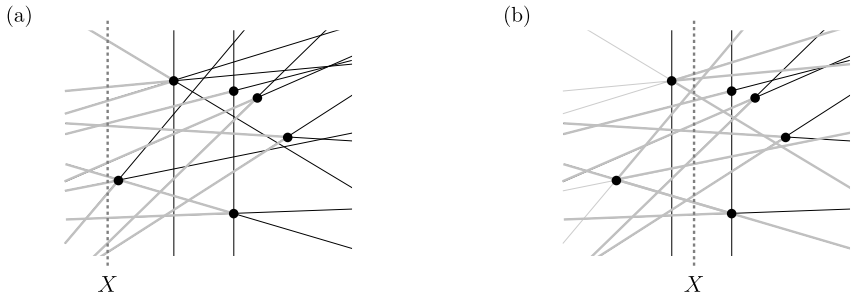

Fig. 6. All half-lines that cross X are illuminated simultaneously: (a) at the beginning and (b) during the sweep. Here $\frac{1}{2}\sum_{v\in V}\deg_{\mathcal{A}}(v) = 16$ and $\Gamma(\mathcal{A}) = 3$, and so 13 searchlights are used.

When X reaches a position to the right of v_{g-1}, no undetected intruder can exist on or to the left of X for the above reason, and furthermore, no undetected intruder can exist to the right of X because all half-lines that cross X are illuminated simultaneously. Therefore, $s(\mathcal{A}) \leq \frac{1}{2}\sum_{v\in V}\deg_{\mathcal{A}}(v) - \Gamma(\mathcal{A})$.

Now, let us prove $s(\mathcal{A}) \leq n-1$. Let the n lines be $L_1, L_2, \ldots, L_n$. Choose any guard $v \in V$, and choose any two lines that pass through v, say L_j and L_k. Place a searchlight F_j at v, and for each of the remaining $n-2$ lines L_i, $i \neq j, k$, place a searchlight F_i at any of the guards on it. (Some guards, including v, may have more than one searchlight.) Now, "rotate" the entire arrangement $\mathcal{A}$ so that L_k becomes vertical, and let $\mathcal{A}'$ be the resulting arrangement having $n-1$ searchlights. Note that in $\mathcal{A}'$, every line L_i except L_k has its "own" searchlight F_i on it, and hence, if the sweep-based strategy described above is applied to $\mathcal{A}'$, then (i) the two vertical half-lines that meet at v can be searched using F_j when X reaches v, and (ii) no undetected intruder, who initially lies to the right of X, can cross X and enter the region to the left of X during the sweep (due to the assumption that no intruder can pass through a guard position). Therefore, since all searchlight are aimed to the right at the end of the sweep, no undetected intruder can exist, and hence we have successfully searched $\mathcal{A}'$. This proves $s(\mathcal{A}) = s(\mathcal{A}') \leq n-1$. □

Observe that from Theorem 2.4 we obtain that for any (n, g)-arrangement $\mathcal{A}$ with no three lines having a point in common ($\Delta(\mathcal{A}) = 4$), $s(\mathcal{A}) \leq 2g-1$ holds. Furthermore, if additionally $\Gamma(\mathcal{A}) = |V|$, that is, all guards of $\mathcal{A}$ belong to some lines that are all parallel, then we obtain $s(\mathcal{A}) \leq g$. In other words, only one searchlight per each guard is enough. In particular, it follows that $s(\mathcal{A}) \leq g$ for any (n, g)-arrangement $\mathcal{A}$ of axis-aligned lines. Finally, one can easily observe that if V is a minimal guard set (not necessarily minimum), then each guard requires at least one searchlight, which immediately results in the exact formula of $s(n, g) = g$ for the class of arrangements of axis-aligned lines, established by the class of arrangements depicted in Fig. 7, for $\lfloor \frac{n}{2} \rfloor \leq g$, where $g \leq n-1$.[8]

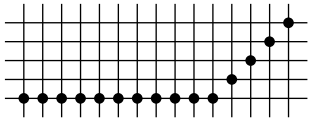

Fig. 7. The arrangement $\mathcal{A} = (\mathcal{L}, V)$ consists of $|V| = g = 15$ vertical and $n - g = 5 \geq 1$ horizontal lines; guards are marked with solid dots. We have $s(\mathcal{A}) = g$.

We summarize the above observations in the following corollary.

Corollary 2.5.

a) Let $\mathcal{A}$ be an (n, g)-arrangement of lines with $\Delta(\mathcal{A}) = 4$. Then, $s(\mathcal{A}) \leq \min\{2g, n\} - 1$.
b) Let $\mathcal{A}$ be an (n, g)-arrangement of lines such that all guards belong to some lines that are all parallel, with $\Delta(\mathcal{A}) = 4$. Then, $s(\mathcal{A}) \leq \min\{g, n-1\}$.
c) For the case of axis-aligned lines, if an arrangement $\mathcal{A}$ consists of $n' \geq 1$ horizontal and $n'' \geq 1$ vertical lines, then $s(\mathcal{A}) = \max\{n', n''\}$.
d) For the case of axis-aligned lines, $s(n, g) = g$ for any $\lfloor \frac{n}{2} \rfloor \leq g \leq n-1$.

[8] Notice that for any (n, g)-arrangement of axis-aligned lines or line segments, we have $\lfloor \frac{n}{2} \rfloor \leq g$.

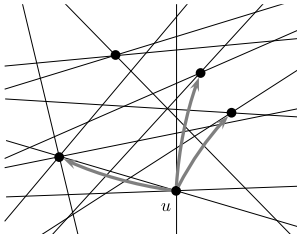

Fig. 8. An arrangement $\mathcal{A} = (\mathcal{L}, V)$ and the neighbors of a guard (vertex) u in the graph $N_{\mathcal{A}}$.

2.3. Upper bounds on $s(n, g)$ for lines

In this subsection, we provide upper bounds on $s(n, g)$, not based on the sum of guard degrees, but expressed in terms of the number of so-called source components in the directed graph $N_{\mathcal{A}}$ of an arrangement $\mathcal{A}$, a graph that represents the "neighbor relation" between guards (defined below). In particular, if this graph has exactly one connected component, we obtain $s(\mathcal{A}) \leq 2g$, even when $\Delta(\mathcal{A}) > 4$ (compare with Corollary 2.5(a)). The bounds are proved constructively, i.e., we provide searching strategies (algorithms) that search any given (n, g)-arrangement using a number of searchlights matching the bound.

Let $\mathcal{A} = (\mathcal{L}, V)$ be an arrangement of lines and let $\mathcal{L}(v_i) = \{L_0^i, L_1^i, \ldots, L_{d_i-1}^i\}$, where $d_i = |\mathcal{L}(v_i)|$. We assume that the half-lines $L_0^i, L_1^i, \ldots, L_{d_i-1}^i$ appear in clockwise order around vertex v_i, and denote by W_t^i the wedge between L_t^i and L_{t+1}^i, called the *t-th wedge* of v_i. (Indices are calculated modulo d_i.) We assume that W_t^i does not include its boundary $L_t^i \cup L_{t+1}^i$.

Let $u, v \in V$ be two distinct guards. We say that v is a *neighbor of* u if there exist two intersecting half-lines $L_u \in \mathcal{L}(u)$ and $L_v \in \mathcal{L}(v)$ such that for any $w \in V \setminus \{u, v\}$, no half-line in $\mathcal{L}(w) \setminus \{L_u, L_v\}$ intersects the open segment $\overline{vx}$ between v and the intersection x of L_u and L_v (we say L_u is *directly reachable* from v along *bridge* $\overline{vx}$); see Fig. 8. Intuitively, v is a neighbor of u if the first half-line hit by some half-line emanating from v belongs to $\mathcal{L}(u)$. By definition, if $v \in L_u$ then v is a neighbor of u. Note that this neighbor relation is not necessarily symmetric, i.e., u may not be a neighbor of v, while v is a neighbor of u. Now, define a directed graph $N_{\mathcal{A}} = (V, E)$ that represents the neighbor relation; $(u, v) \in E$ if and only if v is a neighbor of u. In the following, we shall use "guard" and "vertex" interchangeably.

Observation 2.6. *If $g \geq 2$ then for any $v \in V$, there exists a guard u such that $(u, v) \in E$.*

Proof. It follows from the fact that $\mathcal{L}$ is connected and V is a guard set for $\mathcal{L}$. □

Observation 2.7. *If u is a neighbor of v_i and belongs to wedge W_t^i of v_i, then L_t^i or L_{t+1}^i is directly reachable from u.*

Proof. The claim follows immediately from definition. □

Now, let $G = (\mathcal{V}, \mathcal{E})$ be the condensation graph of $N_{\mathcal{A}}$, where $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$ is the set of strongly connected components of $N_{\mathcal{A}}$ and $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$ if and only if there are vertices $u \in \mathcal{V}_i$ and $v \in \mathcal{V}_j$ such that $(u, v) \in E$. Obviously, sets $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m$ form a partition of V, and G is a directed acyclic graph (DAG). Every source vertex in G is called a *source component* of $N_{\mathcal{A}}$.

Observation 2.8. *For any source component $\mathcal{U}$ of N_A, we have $|\mathcal{U}| \geq 2$.*

Proof. The claim follows from Observation 2.6 and the definition of source component. □

Let $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h$ be the source components of $N_{\mathcal{A}}$ (i.e., the source vertices in G); recall that an *arborescence* is a directed rooted tree in which all edges point away from the root.

Observation 2.9. *For $i = 1, 2, \ldots, h$, let $u_i \in \mathcal{U}_i$ be any vertex. Then, there is a spanning forest of $N_{\mathcal{A}}$ that consists of arborescences $T_i = (V_i, E_i)$, $\mathcal{U}_i \subseteq V_i$, each with root u_i, for $i = 1, 2, \ldots, h$.*

In the following, we first provide an upper bound on $s(n, g)$ when the graph $N_{\mathcal{A}}$ has a single source component (and so a spanning forest of $N_{\mathcal{A}}$ consists of one arborescence). Next, we extend our approach to the general case of arbitrary number of source components. Finally, after introducing the concept of external guards, we prove that $s(n, g) \leq 7g/3 - 1$.

2.3.1. When graph $N_{\mathcal{A}}$ has a single source component

Let $\mathcal{A} = (\mathcal{L}, V)$ be an (n, g)-arrangement such that its graph $N_{\mathcal{A}}$ has exactly one source component $\mathcal{U}_1$. In this subsection, we shall prove the following theorem.

Theorem 2.10. *If $N_{\mathcal{A}}$ has exactly one source component, then $s(\mathcal{A}) \leq 2g$.*

Since for $g \leq 2$, we have $s(\mathcal{A}) < 2g$ by Observation 2.1, we may assume that $g \geq 3$. Let $T = (V, E_T)$ be a spanning arborescence of $N_{\mathcal{A}}$, rooted at a vertex in $\mathcal{U}_1$ (such an arborescence exists by Observation 2.9). Without loss of generality, assume that the root of T is v_0 and let $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ be a topological ordering of T.

Observation 2.11. *For any l, $0 \leq l \leq g-2$, if v_{l+1} is in the intersection of wedges of $v_0, v_1, \ldots, v_l$, i.e.,*

$$v_{l+1} \in W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$$

for some $t_0, t_1, \ldots, t_l$, then for some j, $0 \leq j \leq l$, $L^j_{t_j}$ or $L^j_{t_j+1}$ is directly reachable from v_{l+1} along some bridge $\overline{v_{l+1}x}$, where x is on the boundary of $W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$ and the segment $\overline{v_{l+1}x}$ except point x lies in $W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$.

Proof. Since $\langle v_0, v_1, \ldots, v_l\rangle$ is a prefix of the topological order $\langle v_0, v_1, \ldots, v_{g-1}\rangle$, one of $v_0, v_1, \ldots, v_l$, say v_j, is the parent of v_{l+1} in T. Since v_{l+1} is a neighbor of v_j, by Observation 2.7 there exists a bridge $\overline{v_{l+1}x}$ between v_{l+1} and a point x on the boundary $L^j_{t_j} \cup L^j_{t_j+1}$ of $W^j_{t_j}$. Since $v_{l+1} \in W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$ and no half-line in $\mathcal{L}$ intersects the interior of $\overline{v_{l+1}x}$, x must lie on the boundary of $W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$. Since $W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$ is a convex region, the segment $\overline{v_{l+1}x}$ except point x lies in $W^0_{t_0} \cap W^1_{t_1} \cap \cdots \cap W^l_{t_l}$. □

We place $2g$ searchlights as follows; three searchlights F^0, F^0_0, F^0_1 at v_0, two searchlights F^i_0, F^i_1 at each guard $v_i \in V \setminus \{v_0, v_{g-1}\}$, and one searchlight F^{g-1}_0 at v_{g-1}. For each i, $i = 0, 1, \ldots, g-2$, we use F^i_0 and F^i_1 to successively illuminate the boundary of the wedges of v_i (or in short, *support* the wedges one by one), to perform a (clockwise) rotational wedge sweep around v_i. Such wedge sweeps are simultaneously executed at multiple guards systematically, so that the intersection of such wedges can further be subdivided by a wedge sweep of some other guard using its two searchlights. As illustrated below, the order in which the guards are "activated" to perform a wedge sweep is crucial for a successful completion of the entire strategy.

Example Before we describe our algorithm formally, let us provide some intuition and an outline of our technique with an example. Consider an arrangement $\mathcal{A} = (\mathcal{L}, V)$, with $V = \{v_0, v_1, v_2, v_3, v_4, v_5\}$, and the arborescence T of the graph $N_{\mathcal{A}}$, both depicted in Fig. 9(a), with a topological ordering $O = \langle v_0, v_1, v_2, v_3, v_4, v_5, v_6\rangle$.

Suppose we attempt to search the intersection of the wedge W^0 of v_0 supported by F^0_0, F^0_1 and the wedge W^1 of v_1 supported by F^1_0, F^1_1; see Fig. 9(b). Since there is a non-illuminated cycle C around v_3 in $W^0 \cap W^1$ (Fig. 9(b)), due to unavoidable "recontamination" (Fig. 9(c)) it is not possible to perform a wedge sweep around v_3 using only two searchlights F^3_0 and F^3_1. This difficulty can be avoided if the searchlights are activated according to the ordering of the guards in O. Specifically, since v_2 follows v_0 and precedes v_3 in O, we activate v_2 next, instead of v_3. Since v_2 is a neighbor of v_0 and v_2 is in $W^0 \cap W^1$, by Observation 2.11 there is a bridge from v_2 to the boundary of $W^0 \cap W^1$. So we illuminate the bridge by F^2_0 and start a wedge sweep around v_2 using F^2_0 and F^2_1, as shown in Fig. 9(d). Now, since v_3 follows v_2 in O and v_3 is a neighbor of v_2, again by Observation 2.11, we can start a wedge sweep around v_3 within $W^0 \cap W^1 \cap W^2$ using searchlights F^3_0 and F^3_1, by first illuminating a bridge from v_3 to the boundary of $W^0 \cap W^1 \cap W^2$ by searchlight F^3_0; see Fig. 9(e). Note that by illuminating the bridge, we are able to search all cycles around v_3 within $W^0 \cap W^1 \cap W^2$. Since no such bridge is available for v_0, the first guard that performs a wedge sweep, we place three searchlights F^0, F^0_0, F^0_1 there and use F^0 as a "pseudo bridge."[9] The last guard to be activated, v_{g-1}, needs only one searchlight F^{g-1}_0, since no further wedge intersection subdivision should be necessary.

In summary, our strategy is to activate guards/searchlights according to a topological order induced by some arborescence of the graph $N_{\mathcal{A}}$ (though, as explained later, searchlights of active guards that cannot subdivide the current wedge intersection further need not be turned on). Let us now turn the above discussion into a formal description of the algorithm.

Configuration and subarrangement As outlined above, at any moment during a search of $\mathcal{A} = (\mathcal{L}, V)$, some guards v_i, except the last one v_{g-1} in the order $\langle v_0, v_1, \ldots, v_{g-1}\rangle$, are aiming the two searchlights F^i_0 and F^i_1 at some half-lines $L^i_{t_i}$ and $L^i_{t_i+1}$

[9] If v_0 has a half-line that does not intersect any other line at any point other than v_0, then that half-line works as a bridge and we can do without F^0.

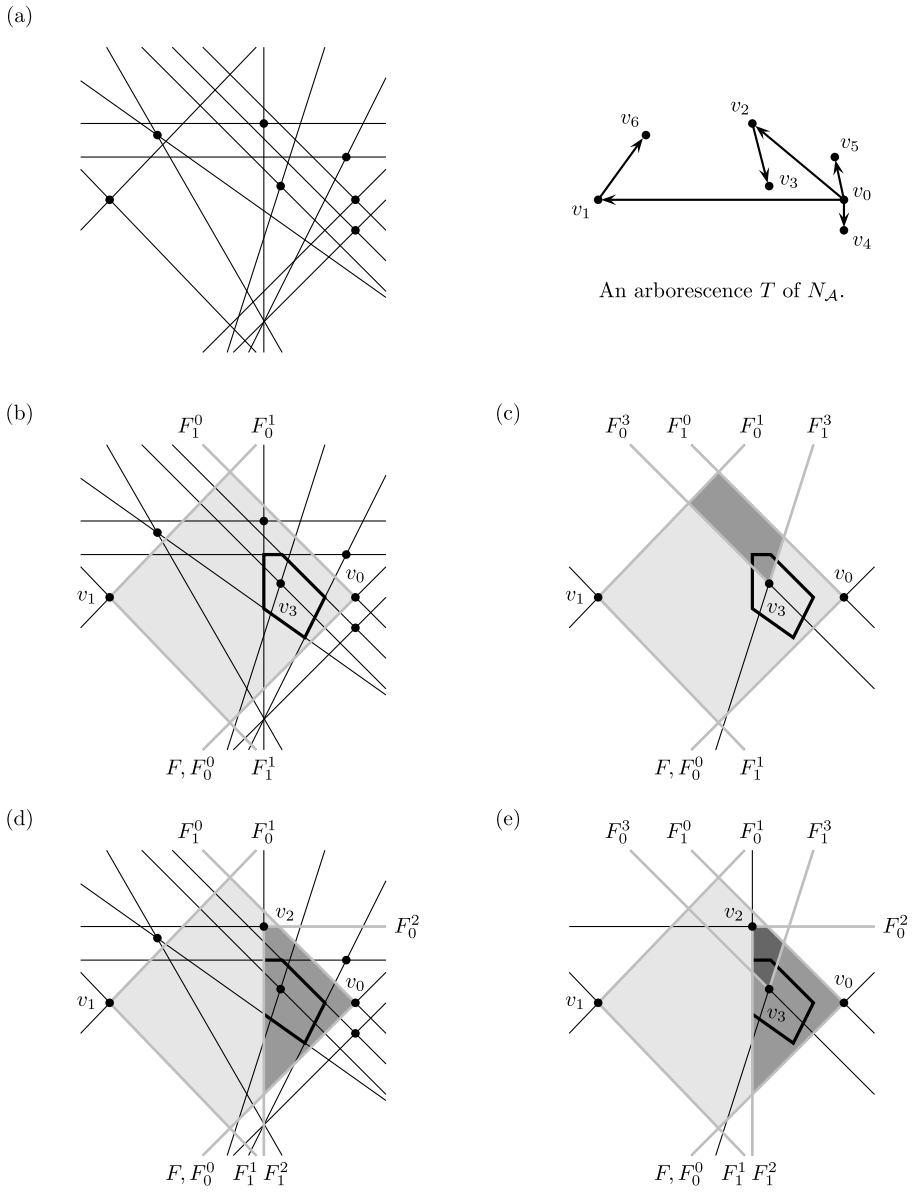

Fig. 9. Topological ordering of an arborescence T and the order of activating searchlights.

in $\mathcal{L}(v_i) = \{L_0^i, L_1^i, \ldots, L_{d_i-1}^i\}$, supporting the t_i-th wedge $W_{t_i}^i$ of v_i. Such guards are said to be *active*. A guard who is not currently supporting any of its wedges is said to be *inactive*. We represent the status of all guards by a mapping, called *configuration*,

$$c : V \to \mathbb{N} \cup \{\perp\},$$

where $\mathbb{N}$ is the set of non-negative integers and $\perp$ represents the "inactive" state of a guard. In particular, we set $c(v_i) = t_i$, $0 \le t_i \le d_i - 1$, if v_i is active and aiming F_0^i and F_1^i at $L_{t_i}^i$ and $L_{t_i+1}^i$, and $c(v_i) = \perp$ if v_i is inactive. For a technical reason, we always set $c(v_{g-1}) = \perp$.

Given a configuration c, we can uniquely identify the wedge intersection that is currently being searched.[10] First, let $\mathrm{act}(c) = \{v \in V : c(v) \in \mathbb{N}\}$ be the set of active guards. For each active guard $v_i \in \mathrm{act}(c)$, its searchlights F_0^i and F_1^i currently support wedge $W_{c(v_i)}^i$. Therefore,

[10] To be precise, it is the segments and half-lines within the wedge intersection that are being searched.

$$R(c) = \bigcap_{v_i \in \mathrm{act}(c)} W^i_{c(v_i)}$$

is the wedge intersection that is currently being searched. By definition, $R(c)$ is an open convex region, and we denote its boundary by $\partial R(c)$. The segments and half-lines within $R(c)$ are identified in

$$\mathcal{L}(c) = \{L \cap R(c) : L \in \mathcal{L}\}.$$

Finally, since $\mathcal{L}(c)$ must be searched using searchlights of the guards that are currently inactive (as $\mathcal{L}(c)$ only contains line components visible to inactive guards), we define the *subarrangement* representing the task of searching $\mathcal{L}(c)$ by

$$\mathcal{A}(c) = (\mathcal{L}(c), V(c)),$$

where $V(c) = \{v \in V : c(v) = \perp\}$ is the set of inactive guards in c. Note that since V is a guard set of $\mathcal{L}$ in arrangement $\mathcal{A} = (\mathcal{L}, V)$, for every segment or half-line in $\mathcal{L}(c)$, there exists a guard in $V(c)$ that can illuminate it.

Remark Subarrangement $\mathcal{A}(c) = (\mathcal{L}(c), V(c))$ is not necessarily an arrangement defined earlier, in two aspects: (1) $\overline{\mathcal{L}(c)}$ is not necessarily a connected region. (2) There may be guards in $V(c)$ that cannot illuminate any segment or half-line in $\mathcal{L}(c)$.

The algorithm Our successive wedge sweep algorithm can best be described as a recursive procedure. Since the top-most level of recursion that describes guard v_0's actions is slightly different from the subsequent recursive calls, we describe it separately here as ALGORITHM1 for $\mathcal{A} = (\mathcal{L}, V)$. Presenting ALGORITHM1 also helps us clarify the concept of rotational wedge sweep.

ALGORITHM1 uses a recursive search procedure SEARCH$(\mathcal{A}, c)$, where c is a configuration, that (as explained later) clears the subarrangement $\mathcal{A}(c) = (\mathcal{L}(c), V(c))$ using the searchlights of the guards in $V(c)$. We defer the description of SEARCH$(\mathcal{A}, c)$ until later. Again, we assume that a topological order $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ of the guards in V has already been chosen using a spanning arborescence of $N_{\mathcal{A}}$.

Algorithm ALGORITHM1$(\mathcal{A})$
/* v_0 performs a rotational wedge sweep */
Step 1: /* initialization */
  1.1 Aim F^0 at L^0_0. /* F^0's direction never changes during the search */
  1.2 Aim F^0_0 at L^0_0 and aim F^0_1 at L^0_1.
Step 2: /* Support the wedges $W^0_0, W^0_1, \ldots, W^0_{d_0-1}$ one by one and call SEARCH to clear the segments in them */
  2.1 For $j = 0$ to $d_0 - 1$ do
  2.2 Rename the searchlights so that F^0_0 is aimed at L^0_j and F^0_1 is aimed at L^0_{j+1}.
  2.3 Set configuration c by $c(v_0) = j$ and $c(v) = \perp$ for all $v \neq v_0$.
  2.4 Call SEARCH$(\mathcal{A}, c)$.
  2.5 Rotate F^0_0 clockwise and aim it at L^0_{j+2}.

Even without the details of SEARCH, the following lemma is obvious.

Lemma 2.12. *If, in Step 2.4,* SEARCH$(\mathcal{A}, c)$ *successfully searches $\mathcal{L}(v)$ of subarrangement $\mathcal{A}(c) = (\mathcal{L}(c), V(c))$ using the searchlights of the guards in $V(c) = \{v_1, v_2, \ldots, v_{g-1}\}$, then* ALGORITHM1$(\mathcal{A})$ *successfully searches arrangement $\mathcal{A}$.*

Proof. Given the assumption above, the first call of SEARCH clears all segments and half-lines in the 0-th wedge W^0_0 of v_0. When F^0_0 is rotated clockwise to illuminate L^0_2 in Step 2.5, W^0_0 remains clear because F^0 and F^0_1 illuminate its boundary $L^0_0 \cup L^0_1$. Again, by assumption the second call of SEARCH clears all segments and half-lines in the 1-st wedge W^0_1 of v_0 that are being supported by F^0_0 and F^0_1. When F^0_0 is rotated clockwise to illuminate L^0_3 in Step 2.5, $W^0_0 \cup W^1_0$ remains clear because F^0 and F^0_1 illuminate its boundary $L^0_0 \cup L^0_2$. Continuing this argument, we can conclude that all lines in $\mathcal{A}$ are clear at the termination of ALGORITHM1$(\mathcal{A})$, provided that SEARCH$(\mathcal{A}, c)$ successfully searches $\mathcal{L}(v)$ using the searchlights of the guards in $V(c)$. □

We now present the details of the recursive procedure SEARCH$(\mathcal{A}, c)$ that activates the guards in the order $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ and searches the subarrangement $\mathcal{A}(c) = (\mathcal{L}(c), V(c))$ using the searchlights of the guards in $V(c)$. The command "**return**" means the current execution of SEARCH is terminated and the control is transferred back to the calling process.

Procedure SEARCH($\mathcal{A}, c$)
Let v_k be the first guard in $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ that is inactive in c.
Case 1: $\mathcal{L}(c) = \emptyset$
  1.1 return.
Case 2: $\mathcal{L}(c) \neq \emptyset$ and $v_k = v_{g-1}$
  2.1 Clear $\mathcal{L}(c)$ by rotating searchlight F_0^{g-1} at v_{g-1} around v_{g-1}.
  2.2 return.
Case 3: $\mathcal{L}(c) \neq \emptyset$ and $v_k \neq v_{g-1}$
  Subcase 3.1: $v_k \notin R(c)$ /* see Fig. 10(a) */
    3.1.1 Let $W_m^k, W_{m+1}^k, \ldots, W_{m+t-1}^k$ be the wedges of v_k that $R(c)$ intersects.
    /* $1 \leq t < d_k$; L_m^k does not intersect $R(c)$ */
    3.1.2 Set $first = m$ and $last = m + t - 1$.
    3.1.3 Go to SWEEP.
  Subcase 3.2: $v_k \in R(c)$ /* $R(c)$ intersects all $W_0^k, W_1^k, \ldots, W_{d_k-1}^k$; see Fig. 10(b) */
    3.2.1 Let L_m^k be the half-line in $\mathcal{L}(v_k)$ that contains the bridge from v_k to $\partial R(c)$.
    3.2.2 Set $first = m$ and $last = m + d_k - 1$.
    3.2.3 Go to SWEEP.
  SWEEP: /* Perform wedge sweep over wedges $W_{first}^k, W_{first+1}^k, \ldots, W_{last}^k$ */
    S.1 Aim F_0^k at L_{first}^k and aim F_1^k at $L_{first+1}^k$.
    S.2 For $j = first$ to $last$ do
    S.3 Rename the searchlights so that F_0^k is aimed at L_j^k and F_1^k is aimed at L_{j+1}^k.
    S.4 Set configuration c' by $c'(v_k) = j$ and $c'(v) = c(v)$ for all $v \neq v_k$.
    S.5 Call SEARCH($\mathcal{A}, c'$).
    S.6 Rotate F_0^k clockwise and aim it at L_{j+2}^k.
    S.7 return.

Remarks

1. In Step 3.1.1, $1 \leq t < d_k$ and L_m^k does not intersect $R(c)$, because $R(c)$ is convex, $d_k \geq 4$, and all wedges have an apex angle of less than π.
2. In Step 3.1.1, if $t = 1$ then $R(c)$ is fully contained in wedge W_m^k. In this case the loop in SWEEP is executed only once for $j = m$, and searchlights F_0^k and F_1^k do not subdivide $R(c)$ any further. Therefore, Steps S.1, S.3 and S.6 can be omitted without affecting the rest of the search of $R(c)$.

The next two lemmas establish the correctness of procedure SEARCH.

Lemma 2.13. *In any invocation of* SEARCH($\mathcal{A}, c$), *from within* ALGORITHM1($\mathcal{A}$) *or from within* SEARCH, *the configuration c passed has the form in which for some l,* $0 \leq l \leq g-2$, $c(i) \in \mathbb{N}$ *for* $i = 0, 1, \ldots, l$ *and* $c(i) = \perp$ *for* $i = l+1, l+2, \ldots, g-1$.

Proof. Configuration c has the above format when first generated in Step 2.3 of ALGORITHM1($\mathcal{A}$). Subsequently, new configurations c' are generated in SEARCH from this c by always changing the first entry of $\perp$ to an integer. Furthermore, $c(v_{g-1}) = \perp$ is never changed. This completes the proof. □

Lemma 2.14. *When called in Step 2.4 of* ALGORITHM1($\mathcal{A}$), SEARCH($\mathcal{A}, c$) *successfully searches subarrangement* $\mathcal{A}(c) = (\mathcal{L}(v), V(c))$ *using the searchlights of the guards in* $V(c)$.

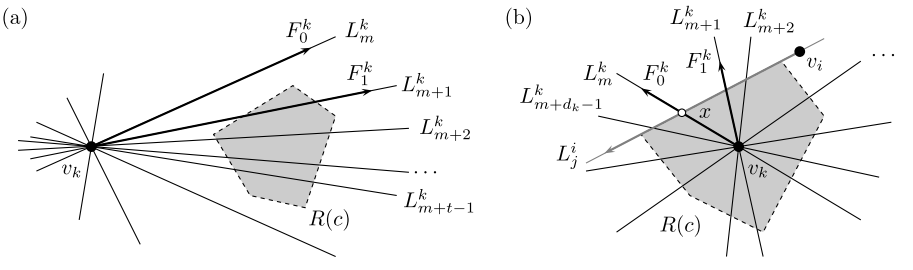

Fig. 10. Step 3 of Procedure SEARCH: (a) $v_k \notin R(c)$; (b) $v_k \in R(c)$: L_m^k contains the bridge $v_k x$.

Proof. There are no segments or half-lines to search in Case 1. For Case 2, by Lemma 2.13 $v_k = v_{g-1}$ implies guards $v_0, v_1, \ldots, v_{g-2}$ are all active, each supporting one of its wedges. Thus, all segments and half-lines in $\mathcal{L}(c)$ are visible from v_{g-1}, and they do not intersect with any half-line that does not emanate from v_{g-1}. Thus, v_{g-1} can clear them simply by illuminating them by F_0^{g-1} in any order. In Subcase 3.1, $R(c)$ is subdivided into smaller wedge intersections by searchlights F_0^k and F_1^k of v_k, and each is searched recursively by SEARCH in Step S.5 without using the searchlights of $v_0, v_1, \ldots, v_k$. Therefore, assuming (inductively) that each of these searches ends successfully, the search of entire $R(c)$ will successfully complete, since the wedge sweep starting with W_m^k that partially intersects $R(c)$ prevents recontamination of cleared wedge intersections. Finally, for Subcase 3.2, first note that by Lemma 2.13 guards $v_0, v_1, \ldots, v_{k-1}$ are all active, and hence $R(c)$ is the intersection of wedges of *all* guards $v_0, v_1, \ldots, v_{k-1}$. Thus by Observation 2.11, $v_k \in R(c)$ implies that there exists a bridge between v_k and $\partial R(c)$. Therefore, again assuming (inductively) that each of the searches in Step S.5 ends successfully, we can conclude that the search of entire $R(c)$ will successfully complete, because (i) the wedge sweep starts by first searching wedge W_m^k that has the bridge on its boundary, and (ii) the bridge, which has no intersection with other lines, will not be recontaminated during the sweep once it is searched. □

Theorem 2.10 follows from Lemmas 2.12 and 2.14.

2.3.2. When graph $N_{\mathcal{A}}$ has more than one source component

Let us extend the discussion above, so that we can handle, in a unified manner, all (n, g)-arrangements $\mathcal{A}$ whose graph $N_{\mathcal{A}}$ has h source components, $h \geq 1$. We start by modifying ALGORITHM1 slightly so that the resulting algorithm, ALGORITHM2, successfully searches arrangement $\mathcal{A} = (\mathcal{L}, V)$ regardless of the number of source components in $N_{\mathcal{A}}$, using *any* given ordering $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ of the guards in V and the following placement of searchlights based on the order $\langle v_0, v_1, \ldots, v_{g-1}\rangle$:

1. Three searchlights F^0, F_0^0, F_1^0 are placed at v_0.
2. For $i = 1, 2, \ldots, g-2$,
   (a) three searchlights F^i, F_0^i, F_1^i are placed at v_i if v_i is not a neighbor of any of $v_0, v_1, \ldots, v_{i-1}$, and
   (b) two searchlights F_0^i, F_1^i are placed at v_i otherwise.
3. One searchlight F_0^{g-1} is placed at v_{g-1}.

The total number of searchlights is $2g$ plus the number of guards v_i, $2 \leq i \leq g-2$, that is not a neighbor of any of $v_0, v_1, \ldots, v_{i-1}$. ALGORITHM2($\mathcal{A}$) is identical to ALGORITHM1($\mathcal{A}$), except that Steps 3.2.1 and S.1 of SEARCH($\mathcal{A}, c$) are replaced by the following Steps 3.2.1′ and S.1′, respectively[11]:

3.2.1′ If there exists a bridge from v_k to $\partial R(c)$, then let L_m^k be the half-line in $\mathcal{L}(v_k)$ that contains the bridge. Otherwise set $m = 0$.

S.1′ Aim F_0^k at L_{first}^k and aim F_1^k at $L_{first+1}^k$. In addition, aim F^k at L_{first}^k if v_k has F^k.

By construction, guard v_k has a third searchlight F^k and aims it at L_{first}^k in Step S.1′ precisely when no bridge exists from v_k to $\partial R(c)$ in Step 3.2.1′.

Note that if $N_{\mathcal{A}}$ has exactly one source component and $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ is a topological order of the guards in V induced by a spanning arborescence of $N_{\mathcal{A}}$, then a total of $2g$ searchlights are placed and ALGORITHM2 reduces to ALGORITHM1.

The correctness of ALGORITHM2 is proved in the next lemma.

Lemma 2.15. *Given an arbitrary ordering $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ of the guards in V,* ALGORITHM2($\mathcal{A}$) *successfully searches arrangement $\mathcal{A} = (\mathcal{L}, V)$ using the searchlights placed as above.*

Proof. Let us focus on the difference between ALGORITHM1 and ALGORITHM2. In Case 1, Case 2 and Subcase 3.1 the two algorithms behave identically. Consider Subcase 3.2, where v_k, the guard to be activated, is in the intersection $R(c)$ of wedges of guards $v_0, v_1, \ldots, v_{k-1}$. If v_k is a neighbor of some v_j, $1 \leq j \leq k-1$, then the argument in the proof of Observation 2.11 carries through (even though $\langle v_0, v_1, \ldots, v_{g-1}\rangle$ may not be a topological order of any arborescence of $N_{\mathcal{A}}$) and shows that a bridge exists from v_k to $\partial R(c)$. Thus, as in ALGORITHM1, a wedge sweep around v_k can be performed using two searchlights F_0^k and F_1^k starting with the wedge W_{first}^k that has the bridge on its boundary. If, on the other hand, v_k is not a neighbor of any of v_j, $1 \leq j \leq k-1$, then such a bridge does not exist. In this case v_k has a third searchlight F^k, so as in Step 1.1 of ALGORITHM1, a wedge sweep around v_k can be performed using F_0^k and F_1^k after setting F^k as a "pseudo bridge" in Step S.1′. This completes the proof of the lemma. □

[11] Technically, it is the procedure SEARCH that is different in ALGORITHM1 and ALGORITHM2.

We are now ready to prove an upper bound on $s(\mathcal{A})$ in terms of the number h of source components in $N_{\mathcal{A}}$. Theorem 2.16 generalizes Theorem 2.10.

Theorem 2.16. $s(\mathcal{A}) \leq 2g + (h-1)$.

Proof. Let $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h$ be the source components of $N_{\mathcal{A}}$. By Observation 2.9 there exists a spanning forest of $N_{\mathcal{A}}$ consisting of h arborescences $T_j = (V_j, E_j)$, $1 \leq j \leq h$, each with root $u_j \in \mathcal{U}_j$, where $V_1, V_2, \ldots, V_k$ form a partition of V. For $1 \leq j \leq h$, let σ_j be a topological order of the guards in V_j induced by T_j, and let $\langle v_0, v_1, \ldots, v_{g-1}\rangle = \sigma_1\sigma_2\ldots\sigma_h$ be their concatenation. Note that each σ_j starts with u_j, $v_0 = u_1$, and $v_{g-1} \neq u_h$ by Observation 2.8. In this sequence, there are exactly $h-1$ guards, $u_2, u_3, \ldots, u_h$, among $v_1, \ldots, v_{g-2}$ that are not a neighbor of any guard preceding them. Therefore, the total number of searchlights placed at the guards to run ALGORITHM2 is $2g + (h-1)$ (three at each of $u_1, u_2, \ldots, u_h$, one at v_{g-1}, and two at each of the remaining $g-h-1$ guards). The theorem follows from this observation and Lemma 2.15. □

We observe that the orderings $\sigma_1, \sigma_2, \ldots, \sigma_h$ induced by the h arborescences $T_j = (V_j, E_j)$, $1 \leq j \leq h$, can be chosen independently of one another in the proof of Theorem 2.16, without affecting the correctness of ALGORITHM2. This fact allows us to "dynamically" modify σ_j for some j during the execution of ALGORITHM2 to further reduce the total number of searchlights.

Specifically, suppose ALGORITHM2 is being executed using the ordering $\langle v_0, v_1, \ldots, v_{g-1}\rangle = \sigma_1\sigma_2\ldots\sigma_h$ set in the proof of Theorem 2.16, where for each j, $1 \leq j \leq h$, σ_j starts with root $u_j \in \mathcal{U}_j$ of arborescence $T_j = (V_j, E_j)$. Suppose SEARCH$(\mathcal{A}, c)$ is being called with current configuration c. The only situation in which the next guard v_k to be activated needs three searchlights is where $v_k = u_j$ (i.e., v_k is the first guard among $v_0, v_1, \ldots, v_{k-1}, v_k$ that belongs to V_j) for some j, $2 \leq j \leq h$, and $u_j \in R(c)$ (Subcase 3.2). Now, suppose the following condition holds for $\mathcal{U}_j$:

Condition A: There exists a guard v_i, $0 \leq i \leq k-1$, such that none of the wedges of v_i contains all guards in $\mathcal{U}_j$.

Condition A ensures that there exists another guard $u'_j \in \mathcal{U}_j$ that is not in the wedge of v_i containing u_j, and hence, $u'_j \notin R(c)$. (Recall that $|\mathcal{U}_j| \geq 2$ by Observation 2.8.) So, if we modify the ordering $\sigma_1\sigma_2\ldots\sigma_j\ldots\sigma_k$ at this moment by replacing σ_j by σ'_j, where σ'_j is a topological ordering of the guards in V_i starting with u'_j induced by an arborescence $T'_j = (V_j, E'_j)$ with root u'_j, then $v_k = u'_j \notin R(c)$ holds (Subcase 3.1) and v_k needs only two searchlights to perform a wedge sweep. Of course, the correctness of ALGORITHM2 is not affected by this modification. In summary, if Condition A holds for $\mathcal{U}_j$, then we can always ensure that $v_k \notin R(c)$, by using either σ_j (if $u_j \notin R(c)$) or σ'_j (if $u_j \in R(c)$). Let us now formalize this observation.

Given an arrangement $\mathcal{A} = (\mathcal{L}, V)$ whose graph $N_{\mathcal{A}}$ has h source components $\mathcal{U}_1, \ldots, \mathcal{U}_h$, a component $\mathcal{U}_i$ is called a *splitter* of $\mathcal{U}_j$, $i \neq j$, if there exists a guard $v \in V[\mathcal{U}_i]$ such that none of the wedges of v fully contains $\mathcal{U}_j$, where $V[\mathcal{U}_i]$ is the set of guards reachable from the guards in $\mathcal{U}_i$ in $N_{\mathcal{A}}$. Construct now a directed graph $G^*_{\mathcal{A}} = (\mathcal{V}^*, \mathcal{E}^*)$ such that $\mathcal{V}^* = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h\}$ and, for $1 \leq i, j \leq h$, $(\mathcal{U}_i, \mathcal{U}_j) \in \mathcal{E}^*$ if and only if $i \neq j$ and $\mathcal{U}_i$ is a splitter of $\mathcal{U}_j$. Note that $(\mathcal{U}_i, \mathcal{U}_j) \in \mathcal{E}^*$ implies that if the searchlights at the guards in $V[\mathcal{U}_i]$ are activated before those in $\mathcal{U}_j$, then $\mathcal{U}_j$ satisfies Condition A and hence none of the guards in $\mathcal{U}_j$ needs three searchlights.

Let h^* be the minimum number of arborescences in a spanning forest of $G^*_{\mathcal{A}}$. (Equivalently, h^* is the number of source vertices in the condensation graph of $G^*_{\mathcal{A}}$.) Since $h^* \leq h$, the following theorem improves the upper bound on $s(\mathcal{A})$ proved in Theorem 2.16.

Theorem 2.17. $s(\mathcal{A}) \leq 2g + (h^* - 1)$.

Proof. Choose a spanning forest of $G^*_{\mathcal{A}}$ consisting of h^* arborescences. Pick an ordering of $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h$ consistent with topological orderings induced by the arborescences. Without loss of generality, rename the source components as necessary and let $\langle \mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h\rangle$ be such an ordering. Next, choose a spanning forest of $N_{\mathcal{A}}$ consisting of h arborescences $T_j = (V_j, E_j)$, $1 \leq j \leq h$, each with root $u_j \in \mathcal{U}_j$, where $V_1 = V[\mathcal{U}_1]$ and for $2 \leq j \leq h$, $V_j = V[\mathcal{U}_j] \setminus (V[\mathcal{U}_1] \cup V[\mathcal{U}_2] \cup \cdots \cup V[\mathcal{U}_{j-1}])$. (Such a spanning forest always exists, by Observation 2.9 and the fact that, for each j, every guard in V_j is reachable from u_j in the subgraph of $N_{\mathcal{A}}$ induced by V_j.) As in the proof of Theorem 2.16, let σ_j be a topological order of the guards in V_j induced by T_j, $1 \leq j \leq h$, and let $\langle v_0, v_1, \ldots, v_{g-1}\rangle = \sigma_1\sigma_2\ldots\sigma_h$ be their concatenation, where each σ_j starts with u_j, $v_0 = u_1$, and $v_{g-1} \neq u_h$ by Observation 2.8. In this sequence, there are exactly $h-1$ guards, $u_2, u_3, \ldots, u_h$, among $v_1, \ldots, v_{g-2}$ that are not a neighbor of any guard preceding them. Now, as the discussion preceding the theorem shows, for $2 \leq j \leq h$, the number of searchlights u_j requires to execute a wedge sweep is two if Condition A holds for $\mathcal{U}_j$, and three otherwise. Obviously, by the choice of the sequence $\langle \mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h\rangle$ that reflect the "splitter" relation, Condition A does *not* hold for exactly h^* components among $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h$ (including $\mathcal{U}_1$) that are the roots of the h^* arborescences in the spanning forest of $G^*_{\mathcal{A}}$. Therefore, the total number of searchlights necessary is $2g + (h^* - 1)$ (three each at u_1 and additional $h^* - 1$ guards among $u_2, \ldots, u_h$, one at v_{g-1}, and two each at the remaining $g - h^* - 1$ guards). The theorem follows from this observation and Lemma 2.15. □

As we observed earlier in Footnote 9, a guard needs only two searchlights to perform a wedge sweep if it has a half-line that does not intersect any other line. We can strengthen this observation as follows. Given an (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$, call a guard $v \in V$ *external* if it lies on the boundary of an unbounded face of the planar subdivision determined by $\bar{\mathcal{L}}$. A guard that has a half-line that does not intersect any other line is external. The following observation generalizes the remark in Footnote 9.

Observation 2.18. *An external guard v needs only two searchlights to perform a wedge sweep.*

Proof. Let f be an unbounded face whose boundary v is on. Draw an imaginary half-line L emanating from v that stays within f. Since L does not intersect any other line, as noted in Footnote 9, v can perform a wedge sweep using two searchlights, using L as a pseudo bridge. This means that in the absence of L, v can still perform a wedge sweep using two searchlights by simply ignoring all instructions to illuminate L. □

As shown below, Observation 2.18 allows us to obtain another upper bound on $s(\mathcal{A})$ in the presence of external guards. Let X be the set of external guards of an (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$, and set $\mathcal{U}_0 = V[X] \subseteq V$, where $V[X]$ is the set of vertices of $N_{\mathcal{A}} = (V, E)$ that are reachable from some vertex in X. Consider the pair $\mathcal{A}' = (\mathcal{L}', V')$, where $V' = V \setminus \mathcal{U}_0$ and $\mathcal{L}'$ is the set of all lines in $\mathcal{L}$ that cannot be illuminated from any guard in $\mathcal{U}_0$, obtained from $\mathcal{A}$ by removing the guards in $\mathcal{U}_0$ (and the half-lines emanating from them).

Lemma 2.19. *If $\mathcal{L}'$ is not empty, then $\mathcal{A}' = (\mathcal{L}', V')$ is an arrangement.*

Proof. Suppose $\mathcal{L}'$ is not empty. We need to show that V' is a guard set of $\mathcal{L}'$ and $\bar{\mathcal{L}}'$ is connected. The former holds by definition, and we prove the latter by showing that every guard in V' is at the intersection of two lines in $\mathcal{L}'$, which implies that not all lines in $\mathcal{L}'$ are parallel. Let v be any guard in V'. Suppose v is on just one line L in $\mathcal{L}'$, while, by assumption on $\mathcal{A}$, v is at the intersection of two lines L and L' in $\mathcal{L}$. Then, it must be the case that $L' \in \mathcal{L}(u)$ for some guard $u \in \mathcal{U}_0$. This implies that v is a neighbor of u and hence $v \in V[X] = \mathcal{U}_0$, which contradicts $v \in V' = V \setminus \mathcal{U}_0$. □

Based on Lemma 2.19, let $\mathcal{U}_1, \ldots, \mathcal{U}_{h'}$ be the source components of the directed graph $N_{\mathcal{A}'}$ of arrangement $\mathcal{A}'$. Define a directed graph $G^+_{\mathcal{A}} = (\mathcal{U}^+, \mathcal{E}^+)$ where $\mathcal{U}^+ = \{\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{h'}\}$, and for $1 \le i, j \le h'$, $(\mathcal{U}_i, \mathcal{U}_j) \in \mathcal{E}^+$ if and only if $i \ne j$ and $\mathcal{U}_i$ is a splitter of $\mathcal{U}_j$. In addition, for $1 \le j \le h'$, $(\mathcal{U}_0, \mathcal{U}_j) \in \mathcal{E}^+$ if and only if there exists a guard $v \in \mathcal{U}_0$ such that none of the wedges of v fully contains $\mathcal{U}_j$.[12] Note that, by definition, $\mathcal{U}_0$ is always a source vertex of $G^+_{\mathcal{A}}$. Let h^+ be the minimum number of arborescences in a spanning forest of $G^+_{\mathcal{A}}$. (Equivalently, h^+ is the number of source vertices in the condensation graph of $G^+_{\mathcal{A}}$.) We have the following theorem.

Theorem 2.20. *If $X \ne \emptyset$, then $s(\mathcal{A}) \le 2g + (h^+ - 2)$.*

Proof. As in the proof of Theorem 2.17, it suffices to give an ordering $\langle v_0, v_1, \ldots, v_{g-1} \rangle$ of the guards in V for which $2g + (h^+ - 2)$ searchlights are sufficient for executing ALGORITHM2. Such an ordering can easily be obtained by concatenating the following two orderings in the given order:

1. Any ordering of the guards in X followed by an ordering of the guards in $\mathcal{U}_0 \setminus X$, such that, every guard $v \notin X$ is a neighbor of some guard preceding it in the sequence. (Such an ordering always exists because $\mathcal{U}_0 = V[X]$.)
2. An ordering of the guards in V' $(= V \setminus \mathcal{U}_0)$ constructed for $N_{\mathcal{A}'}$ from $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{h'}$ based on a spanning forest of $N_{\mathcal{A}'}$ consisting of h^+ arborescences, as was done in the proof of Theorem 2.17 for $N_{\mathcal{A}}$ from $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_h$.

Since every guard in X (including the guard that is now v_0) needs only two searchlights by Observation 2.18, the total number of searchlights is $2g + (h^+ - 2)$ (three at each first guard of the $h^+ - 1$ source vertices of $G^+_{\mathcal{A}}$ excluding $\mathcal{U}_0$, one at v_{g-1}, and two at each of the remaining $g - h^+$ guards). This completes the proof. □

We now establish an upper bound of $7g/3 - 1$ on $s(n, g)$ based on estimates of h^* and h^+ and the bounds given in Theorems 2.17 and 2.20. First, we have:

Lemma 2.21. *Let $\mathcal{U}$ be an arbitrary source component of an arrangement. If $\mathcal{U}$ does not contain an external guard, then $|\mathcal{U}| \ge 3$.*

Proof. Assume that $\mathcal{U}$ does not contain an external guard. By Observation 2.8, $|\mathcal{U}| \ge 2$. Suppose for a contradiction that $|\mathcal{U}| = 2$ and denote $\mathcal{U} = \{x_1, x_2\}$. Since neither x_1 nor x_2 is external and neither is a neighbor of any other guard, every

[12] Since $\mathcal{U}_0 = V[\mathcal{U}_0]$ holds, $(\mathcal{U}_0, \mathcal{U}_j) \in \mathcal{E}^+$ effectively means $\mathcal{U}_0$ is a "splitter" of $\mathcal{U}_j$. However, the splitter relation has been defined over the set of source components, to which $\mathcal{U}_0$ does not belong.

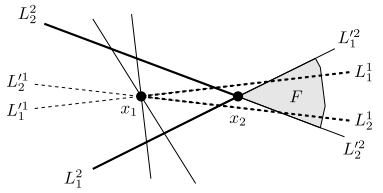

Fig. 11. Illustration for the proof of Lemma 2.21.

half-line in $\mathcal{L}(x_1)$ is intersected first by a half-line in $\mathcal{L}(x_2)$, and every half-line in $\mathcal{L}(x_2)$ is intersected first by a half-line in $\mathcal{L}(x_1)$. Thus, x_1 and x_2 cannot lie on the same line. Let L_1^1, L_2^1 be the half-lines in $\mathcal{L}(x_1)$ defining the smallest wedge of x_1 containing x_2. Similarly, let L_1^2, L_2^2 be the half-lines in $\mathcal{L}(x_2)$ defining the smallest wedge of x_2 containing x_1. See Fig. 11. For each of these half-lines L_q^p, let us denote by $L_q'^p$ the half-line in $\mathcal{L}(x_p)$ such that L_q^p and $L_q'^p$ are the two halves of a single line passing through x_p. Then, either $L_1'^2$ or $L_2'^2$ is not intersected by any half-line in $\mathcal{L}(x_1)$, or either $L_1'^1$ or $L_2'^1$ is not intersected by any half-line in $\mathcal{L}(x_2)$ — a contradiction. □

Lemma 2.21 implies that if $X = \emptyset$ then $h^* \leq g/3$, and if $X \neq \emptyset$ then $h^+ \leq (g - |\mathcal{U}_0|)/3 \leq (g-1)/3$. By these observations and Theorems 2.17 and 2.20, we obtain:

Theorem 2.22. *For each $n \geq 1$ and $g \geq 1$, $s(n, g) \leq 7g/3 - 1$.*

It is worth pointing out that the result of Lemma 2.21 cannot be improved since for every $k \geq 1$, there exists an arrangement without external guards having exactly k source components, each of size three: see Fig. 12(a, b and c), respectively, for such an arrangement for $k = 1$, its schematic drawing, and a schematic drawing of such an arrangement for $k = 2$. On the other hand, they satisfy $h^+ = 1$ because the most "internal" source component is a splitter for all other source components, and hence by Theorem 2.20, they require only $2g$ searchlights.

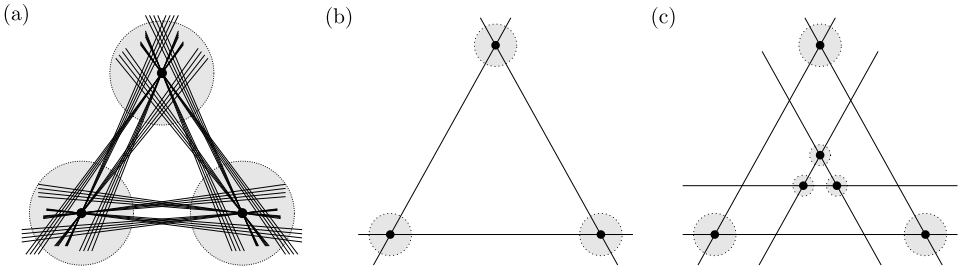

Fig. 12. (a) An arrangement $\mathcal{A} = (\mathcal{L}, V)$ with no external guards whose directed graph $N_{\mathcal{A}}$ is a 3-vertex bidirectional cycle. The three guards in the cycle form a single connected component. (b) For any guard $v \in V$, we "bundle" the lines in $\mathcal{L}(v)$ tightly and use the bundles to form a triangular arrangement. (c) By repeatedly embedding a triangular arrangement in the "interior" of an existing arrangement, we obtain an arrangement with no external guards, having k source components of size three each; here $k = 2$.

3. The searchlight problem for line segments

In this section, we discuss the bounds on the number of searchlights that are sometimes necessary but always sufficient to successfully search an arbitrary (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ of line segments. We note that the problem of determining the minimum guard set for a set of line segments is NP-hard [2].

In Section 3.1, we show a lower bound of $\Omega(g \cdot \log \frac{n}{g})$ on $s(n, g)$, that is, we prove that there are (n, g)-arrangements of line segments that require $\Omega(g \cdot \log \frac{n}{g})$ searchlights. Next, in Section 3.2, we establish an upper bound of $O(g^2 \cdot \log n)$ on $s(n, g)$, that is, we prove that any (n, g)-arrangement of line segments can be searched using $O(g^2 \cdot \log n)$ searchlights. Finally, following the approach discussed in Section 2.2, we provide an upper bound on $s(n, g)$ in terms of the sum of guard degrees.

As before, $\Delta(\mathcal{A}) = \max_{v \in V} \deg_{\mathcal{A}}(v)$ is the degree of $\mathcal{A} = (\mathcal{L}, V)$, where $\deg_{\mathcal{A}}(v) = |\mathcal{L}(v)|$ and $\mathcal{L}(v)$ is the set of all maximal subsegments of line segments in $\mathcal{L}$ that meet at v.

3.1. The lower bound

Consider the arrangement $\mathcal{A}_2 = (\mathcal{L}_2, V_2)$ depicted in Fig. 13, with $\Delta(\mathcal{A}_2) = \Delta_2 = 4$, $|\mathcal{L}_2| = 2 \cdot \Delta_2 = 8$ and $V_2 = \{v_0, v_1\}$.

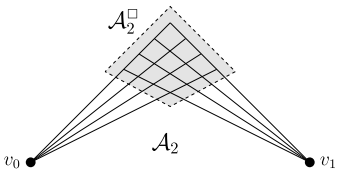

Fig. 13. Arrangement $\mathcal{A}_2 = (\mathcal{L}_2, V_2)$ and its subarrangement $\mathcal{A}_2^{\square}$.

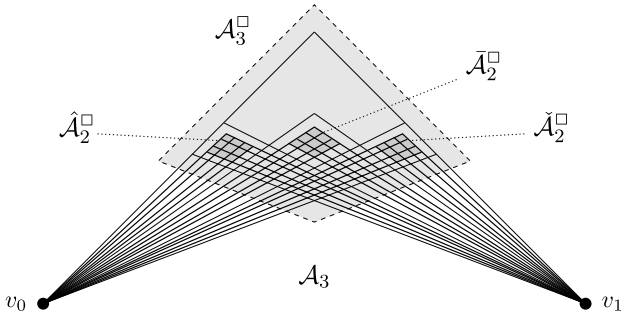

Fig. 14. Arrangement $\mathcal{A}_3$ constructed from three copies of $\mathcal{A}_2$: $\hat{\mathcal{A}}_2$, $\bar{\mathcal{A}}_2$ and $\check{\mathcal{A}}_2$, and eight additional line segments.

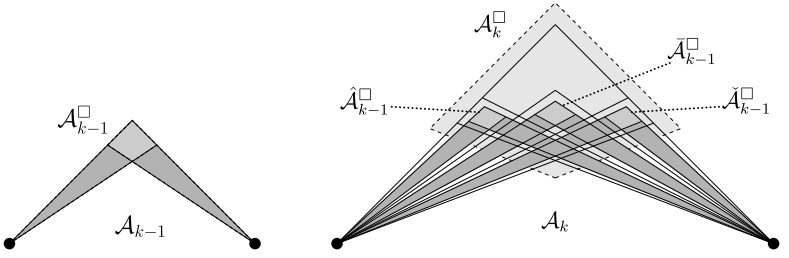

Fig. 15. Arrangement $\mathcal{A}_k$ constructed from three copies of $\mathcal{A}_{k-1}$ and eight additional line segments.

We now construct an arrangement $\mathcal{A}_3 = (\mathcal{L}_3, V_3)$ and its subarrangement $\mathcal{A}_3^{\square}$ as depicted in Fig. 14, where

$$\Delta(\mathcal{A}_3) = \Delta_3 = 4 + 3 \cdot \Delta_2 = 16 \leq 4 \cdot \Delta_2.$$

$\mathcal{A}_3$ is constructed from three "copies" of $\mathcal{A}_2$ and eight additional line segments; observe that $|\mathcal{L}_3| = 2 \cdot \Delta_3$ and $|V_3| = |V_2| = 2$.

We continue with an analogous construction for any $k \geq 4$. Namely, the arrangement $\mathcal{A}_k = (\mathcal{L}_k, V_k)$, with $|\mathcal{L}_k| = 2 \cdot \Delta(\mathcal{A}_k)$, $V_k = \{v_0, v_1\}$ and

$$\Delta(\mathcal{A}_k) = \Delta_k = 4 + 3 \cdot \Delta_{k-1} \leq 4 \cdot \Delta_{k-1} \leq 4^{k-1} \tag{1}$$

is constructed from three "copies" of the arrangement $\mathcal{A}_{k-1}$ with disjoint sets $\mathcal{L}_{k-1}$ and eight additional line segments, as illustrated in Fig. 15. One can observe that the arrangement $\mathcal{A}_k$ has the following key property: *Between any two copies of $\mathcal{A}_{k-1}^{\square}$, there is a path that avoids the line segments of the third copy of $\mathcal{A}_{k-1}$.*

Lemma 3.1. *For each $k \geq 2$, $s(\mathcal{A}_k) = \Omega(\log \Delta_k)$.*

Proof. If, in a searching strategy for $\mathcal{A}_k$, $i \geq 1$ searchlights simultaneously illuminate some segments in $\mathcal{L}_k$ at time t, each containing a segment from $\mathcal{A}_k^{\square}$, then we say that these i searchlights are *involved in searching* $\mathcal{A}_k^{\square}$ at time t. Following this definition, for a searching strategy A for $\mathcal{A}_k$, let $x_A(\mathcal{A}_k)$ denote the maximum number of searchlights involved in searching $\mathcal{A}_k^{\square}$ by A. Define

$$x(\mathcal{A}_k) := \min_A x_A(\mathcal{A}_k),$$

where the minimum is taken over all searching strategies A for $\mathcal{A}_k$. For the purpose of the proof, let $\mathcal{A}_1 = (\emptyset, \{v_0, v_1\})$. Observe that $s(\mathcal{A}_1) = x(\mathcal{A}_1) = 0$ and we have $s(\mathcal{A}_k) \geq x(\mathcal{A}_k)$ for each $k \geq 2$.

Now, we prove by induction on $k \geq 2$ that $x(\mathcal{A}_k) \geq x(\mathcal{A}_{k-1}) + 1$ holds. For $k = 2$, the claim follows from the fact that $x(\mathcal{A}_1) = 0$ and we need to place at least one searchlight at each guard in $\mathcal{A}_2$ to search $\mathcal{A}_2^{\square}$ (and so $x(A_2) \geq 2$).

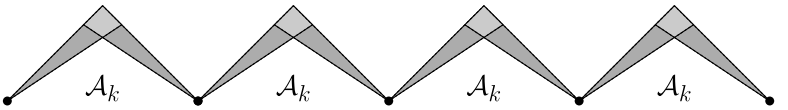

Fig. 16. Arrangement $\mathcal{A}$ with $s(\mathcal{A}) = \Omega(g \cdot \log \frac{n}{g})$; here $g = 5$.

Suppose now that the claim holds for some $k-1 \geq 2$ and we prove it for k. Let $\hat{\mathcal{A}}^{\square}_{k-1}$, $\bar{\mathcal{A}}^{\square}_{k-1}$ and $\check{\mathcal{A}}^{\square}_{k-1}$ be the three copies of $\mathcal{A}^{\square}_{k-1}$ included in $\mathcal{A}^{\square}_{k}$, as shown in Fig. 15. By definition, since $\mathcal{A}^{\square}_{k}$ includes a copy of $\mathcal{A}^{\square}_{k-1}$, we have $x(\mathcal{A}_k) \geq x(\mathcal{A}_{k-1})$. Consider now any searching strategy A for $\mathcal{A}_k$, and suppose for a contradiction that A involves only $x(\mathcal{A}_{k-1})$ searchlights in searching $\mathcal{A}^{\square}_{k}$. Note that A clears the entire $\mathcal{A}^{\square}_{k}$, and $\hat{\mathcal{A}}^{\square}_{k-1}$, $\bar{\mathcal{A}}^{\square}_{k-1}$ and $\check{\mathcal{A}}^{\square}_{k-1}$ are pairwise disjoint copies of $\mathcal{A}^{\square}_{k-1}$. Thus, by the induction hypothesis, there exist three time moments $\hat{t}$, $\bar{t}$ and $\check{t}$ in which at least $x(\mathcal{A}_{k-1})$ searchlights are involved in searching $\hat{\mathcal{A}}^{\square}_{k-1}$, $\bar{\mathcal{A}}^{\square}_{k-1}$ and $\check{\mathcal{A}}^{\square}_{k-1}$, respectively. Take $\hat{t}$, $\bar{t}$ and $\check{t}$ to be maximal, i.e., $\hat{\mathcal{A}}^{\square}_{k-1}$, $\bar{\mathcal{A}}^{\square}_{k-1}$ and $\check{\mathcal{A}}^{\square}_{k-1}$ remain clear after the time moments $\hat{t}$, $\bar{t}$ and $\check{t}$, respectively. Moreover, since $\hat{\mathcal{A}}^{\square}_{k-1}$, $\bar{\mathcal{A}}^{\square}_{k-1}$ and $\check{\mathcal{A}}^{\square}_{k-1}$ are pairwise disjoint, we have that the three time moments are pairwise different.

Assume without loss of generality that $\hat{t} < \bar{t} < \check{t}$. Consider the time moment $\bar{t}$ when all $x(\mathcal{A}_{k-1})$ searchlights are involved in searching $\bar{\mathcal{A}}^{\square}_{k-1}$. Now, since there is a non-illuminated path in $\mathcal{A}^{\square}_{k}$ (i.e., a path disjoint from all segments of $\bar{\mathcal{A}}_{k-1}$) from the contaminated $\check{\mathcal{A}}^{\square}_{k-1}$ to $\hat{\mathcal{A}}^{\square}_{k-1}$, the intruder may at time moment $\bar{t}$ get from $\check{\mathcal{A}}^{\square}_{k-1}$ to $\hat{\mathcal{A}}^{\square}_{k-1}$ without being detected. This contradicts the maximality of $\hat{t}$, i.e., it contradicts the assumption that $\hat{\mathcal{A}}^{\square}_{k-1}$ remains clear at any time moment $t \geq \hat{t}$. Consequently, the searching strategy A cannot use only $x(\mathcal{A}_{k-1})$ searchlights to search $\mathcal{A}^{\square}_{k}$ (and thus to search $\mathcal{A}_k$). This proves that $x(\mathcal{A}_k) \geq x(\mathcal{A}_{k-1}) + 1$.

Now we are ready to conclude the proof of the lemma. Note that for $k \geq 2$, $s(\mathcal{A}_k) \geq x(\mathcal{A}_k) \geq 2$ and a simple induction on k together with the above claim imply $s(\mathcal{A}_k) \geq k$. Thus, by (1), $s(\mathcal{A}_k) = \Omega(\log |\mathcal{L}_k|) = \Omega(\log \Delta_k)$ as required. □

Now, by adjoining $g-1$ copies of $\mathcal{A}_k$, as shown in Fig. 16, we construct an (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ with $n = (g-1) \cdot \Delta(\mathcal{A})$ and $\Delta(\mathcal{A}) = 2 \cdot \Delta(\mathcal{A}_k)$. Each guard in V requires $\Omega(\log \Delta(\mathcal{A}))$ searchlights to search $\mathcal{A}$, and hence $s(\mathcal{A}) = \Omega(g \cdot \log \Delta(\mathcal{A})) = \Omega(g \cdot \log \frac{n}{g})$.

Of course, the above construction is derived for a particular choice of g and n. However, to derive asymptotically the same lower bound for any choice of g and n, all we need is to consider the largest k such that $(g-1) \cdot 2 \cdot 4^{k-1} \leq n$. Then our construction provides the arrangement $\mathcal{A}$ such that $s(\mathcal{A}) = \Omega(g \cdot k)$. Since $k = \Omega(\log \frac{n}{g})$ by the maximality of k, we also obtain $s(\mathcal{A}) = \Omega(g \cdot \log \frac{n}{g})$, and hence we may conclude with the following theorem.

Theorem 3.2. *For arrangements of line segments, we have* $s(n, g) = \Omega(g \cdot \log \frac{n}{g})$.

3.2. The upper bound

In this section we prove that $O(g^2 \cdot \log n)$ searchlights are always sufficient to search any (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ of line segments. Briefly, the idea is as follows. First, by permanently illuminating a set $\mathcal{I} \subseteq \mathcal{L}$ of segments using at most $2g-1$ searchlights (Section 3.2.1), we divide $\mathcal{A}$ into k "nice" subarrangements $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ such that each satisfies three "nice" properties defined below. Next, we search these subarrangements one by one using for each of them a recursive divide-and-conquer procedure based on the concept of "balanced splitters" and "separators" (Section 3.2.3 and Section 3.2.4, respectively).

3.2.1. Partitioning into nice subarrangements

Let $\mathcal{I} \subseteq \mathcal{L}$ be a set of segments that satisfies the following *nice* properties.

1. Every guard in V lies on some segment in $\mathcal{I}$,
2. the union $\bar{\mathcal{I}}$ of segments in $\mathcal{I}$ is connected, and
3. there is a segment $L \in \mathcal{I}$ incident to the unbounded face F of the planar subdivision formed by the segments in $\mathcal{L}$ (in other words, L has an endpoint on the boundary of F).

Note that such set $\mathcal{I}$ always exists (one may take $\mathcal{I} = \mathcal{L}$); in Lemma 3.4 we will establish the existence of a set $\mathcal{I}$ of "small" cardinality. Now, if we place a sufficient number of searchlights at the guard locations and simultaneously and permanently illuminate all segments in $\mathcal{I}$, then the segments in $\mathcal{L} \setminus \mathcal{I}$ are partitioned into a number of sets $\mathcal{L}_1, \ldots, \mathcal{L}_k$ of non-illuminated subsegments (to which we shall simply refer as "segments" hereafter), where each $\bar{\mathcal{L}}_i$ is a maximal connected set in $\bar{\mathcal{L}} \setminus \bar{\mathcal{I}}$; see Fig. 17. Each $\mathcal{L}_i$, $i = 1, \ldots, k$, together with $\mathcal{I}$, constitutes a *nice subarrangement of* $\mathcal{A}$ *with respect to* $\mathcal{I}$, denoted by $\mathcal{A}_i = (\mathcal{L}_i, V, \mathcal{I})$. We call $\mathcal{I}$ the *boundary* of $\mathcal{A}_i$. The above properties of $\mathcal{I}$ guarantee the following.

Observation 3.3. *For each* $i \in \{1, \ldots, k\}$, *every segment in* $\mathcal{L}_i$ *has an endpoint in* $\mathcal{I}$.

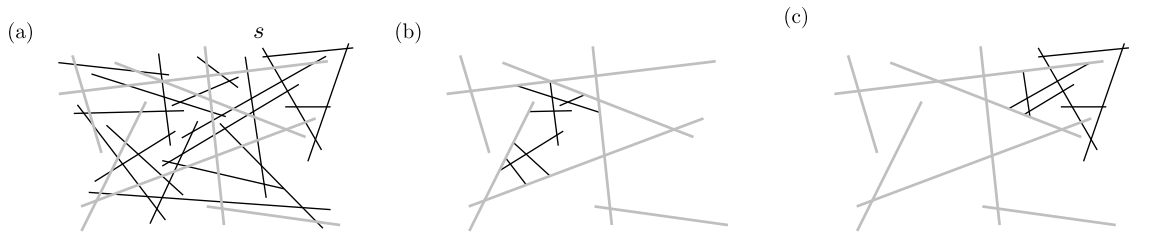

Fig. 17. (a) An arrangement $\mathcal{A} = (\mathcal{L}, V)$ with a subset $\mathcal{I} \subset \mathcal{L}$ of illuminated segments, marked gray. (b, c) Some subarrangements of $\mathcal{A}$ with respect to $\mathcal{I}$. (c) A segment in $\mathcal{L}$ may contribute a number of line segments in a subarrangement; here, segment $s \in \mathcal{L} \setminus \mathcal{I}$ contributes three line segments.

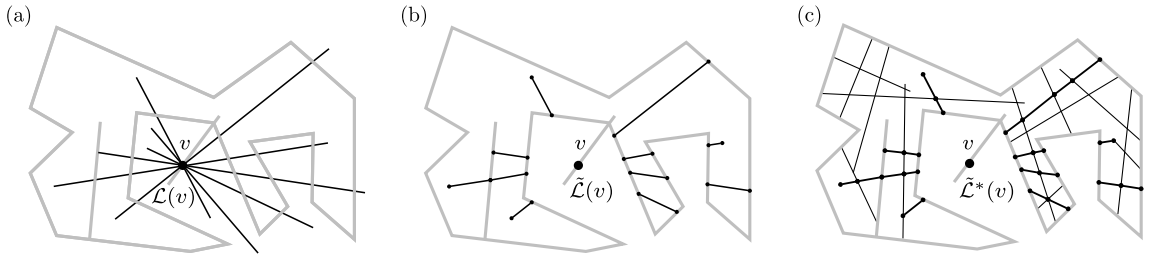

Fig. 18. (a) The set $\mathcal{L}(v)$ of all maximal line-segments that meet at v; here $|\mathcal{L}(v)| = 14$. (b) The set $\tilde{\mathcal{L}}(v) \subset \tilde{\mathcal{L}}$ of all (non-illuminated) segments of a nice subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ that can be illuminated from $v \in V$; here $|\tilde{\mathcal{L}}(v)| = 11$. (c) The set $\tilde{\mathcal{L}}^*(v)$ of all edges contained in the segments in $\tilde{\mathcal{L}}(v)$; here $|\tilde{\mathcal{L}}^*(v)| = 22$.

The next lemma shows that a desired set $\mathcal{I} \subseteq \mathcal{L}$ can be constructed using at most $2g - 1$ line segments in $\mathcal{L}$.

Lemma 3.4. *Let $\mathcal{A} = (\mathcal{L}, V)$ be an (n, g)-arrangement of line segments. There exists a subset $\mathcal{I} \subseteq \mathcal{L}$ of size at most $2g - 1$ that satisfies the three nice properties.*

Proof. Let $G = (V, E)$ be the graph with vertex set V, and in which two vertices v_i and v_j, $i \neq j$, are adjacent if and only if either (i) $\mathcal{L}(v_i) \cap \mathcal{L}(v_j) \neq \emptyset$ (i.e., v_i and v_j lie on a segment L in $\mathcal{L}$), or (ii) some segments $L' \in \mathcal{L}(v_i)$ and $L'' \in \mathcal{L}(v_j)$ intersect. We say that L or, respectively, L' and L'', *correspond(s)* to the edge $\{v_i, v_j\}$ of G; if there are several such pairs (L', L''), we choose (and fix) any of them. Notice that since the union $\bar{\mathcal{L}}$ is connected, G is connected as well. Let T be a spanning tree of G. First, for every edge of T, include in $\mathcal{I}$ either one segment or a pair of segments that corresponds to it. Next, if no segment chosen so far has an endpoint in the unbounded face of the planar subdivision formed by the segments in $\mathcal{L}$, then choose an arbitrary segment in $\mathcal{L}$ that has an endpoint in the unbounded face and include it in $\mathcal{I}$. The resulting set $\mathcal{I}$ consists of at most $2g - 1$ segments all taken from $\mathcal{L}$, and satisfies the three nice properties. □

Taking into account the above lemma, by using at most $2 \cdot (2g - 1) = \Theta(g)$ searchlights to permanently illuminate all segments in $\mathcal{I}$, we may focus only on bounding the number of searchlights that are needed to search any (single) nice subarrangement of $\mathcal{A}$ with respect to the boundary $\mathcal{I}$. Our idea for proving the existence of such a searching strategy that uses 'few' searchlights is built on a nested "balanced" dissection method, discussed in the next sections.

3.2.2. Splitters and induced subarrangements

Let $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ be a nice subarrangement with respect to $\mathcal{I}$ of a given arrangement $\mathcal{A} = (\mathcal{L}, V)$. Since $\mathcal{I}$ induces a partition of $\mathcal{A}$, a line segment $L \in \mathcal{L}$ may contribute several segments in $\tilde{\mathcal{L}}$. Hence now, for $v \in V$, $\tilde{\mathcal{L}}(v)$ denotes the set of all segments in $\tilde{\mathcal{L}}$ that can be illuminated from v (by illuminating the line segments in $\mathcal{L}(v)$), see Fig. 18(a, b); note that $|\tilde{\mathcal{L}}| = O(n^2)$. Next, for a segment $\tilde{L} \in \tilde{\mathcal{L}}(v)$, the line segment in $\mathcal{L}(v)$ that contains $\tilde{L}$ is called the *representative of* $\tilde{L}$ and is denoted by $r(\tilde{L})$. Finally, for a subset $\tilde{\mathcal{S}} \subseteq \tilde{\mathcal{L}}$, the set of representatives of elements in $\tilde{\mathcal{S}}$ is denoted by $r(\tilde{\mathcal{S}})$; note that $|r(\tilde{\mathcal{S}})| \leq |\tilde{\mathcal{S}}|$.

Let us consider the case in which the segments in $\tilde{\mathcal{L}}$ lie within a bounded face F of the planar subdivision formed by the segments in $\mathcal{I}$; the case where F is unbounded is discussed in Subsection 3.2.6. A *splitter* $S \subset F$ for the face F is defined as any line segment, not necessarily in $\tilde{\mathcal{L}}$, whose endpoints are in the union $\bar{\mathcal{I}}$ of segments in $\mathcal{I}$ and that has no other point in common with $\bar{\mathcal{I}}$. Note that S may contain up to two segments in $\tilde{\mathcal{L}}$, and we use $\tilde{\mathcal{L}}(S)$ to denote the set of such segments. On the other hand, S may contain no segment in $\tilde{\mathcal{L}}$. We denote by $\tilde{\mathcal{X}}(S)$ the set of segments in $\tilde{\mathcal{L}}$ that intersect the interior of S but do not intersect any of the elements in $\tilde{\mathcal{L}}(S)$. See Fig. 19(a). Note that $\tilde{\mathcal{X}}(S)$ may be empty.

Suppose now we simultaneously illuminate the segments in $r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S))$ (in addition to those in $\mathcal{I}$) so that $\mathcal{I} \cup r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S))$ is now a boundary for $\mathcal{A}$. Then, by Observation 3.3, $\mathcal{I} \cup r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S))$ satisfies the nice properties (1–3), and the segments in $\tilde{\mathcal{L}}$ that are not illuminated form a number of nice subarrangements of $\tilde{\mathcal{A}}$ *induced by* S; see Fig. 19(b) for

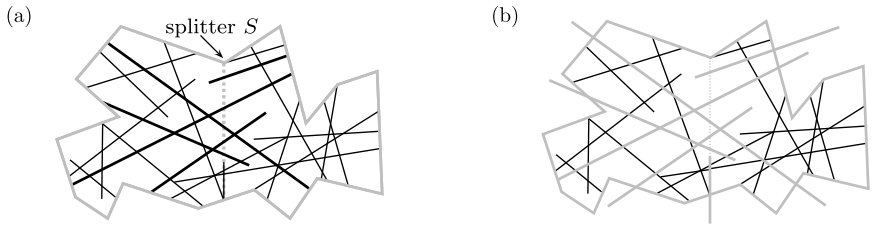

Fig. 19. (a) A subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$: the splitter is marked with the bold dashed gray line, and the elements of $\tilde{\mathcal{X}}(S)$ are marked with bold black lines; notice that $|\tilde{\mathcal{L}}(S)| = 1$. (b) The subarrangements induced by S.

an illustration. (If all segments in $\tilde{\mathcal{L}}$ are illuminated, then we consider the result to be an empty subarrangement $(\emptyset, V, \mathcal{I} \cup r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S)))$.) This method may be used recursively but it has two potential problems that we need to overcome. The first problem is that the set $\tilde{\mathcal{X}}(S)$ may be too large, forcing the searching strategy to use too many searchlights simultaneously. This issue is addressed in Section 3.2.4. The second problem is the depth of recursion — this problem is addressed in the next section.

3.2.3. Balanced splitters

A maximal subsegment L of a segment in $\tilde{\mathcal{L}}$ such that the interior of L has no points in common with other segments in $\tilde{\mathcal{L}} \cup \mathcal{I}$, is called an *edge*. Note that the only points that an edge L may have in common with other segments in $\tilde{\mathcal{L}} \cup \mathcal{I}$ are its endpoints.

Let $\tilde{\mathcal{L}}^*(v)$ denote the set of all edges contained in the segments in $\tilde{\mathcal{L}}(v)$. See Fig. 18(c) for an illustration. A natural divide-and-conquer strategy for searching a nice subarrangement $\tilde{\mathcal{A}}$ is to divide it by using a splitter into a number of "balanced" subarrangements, and the measure we use for balancing subarrangements is related to the size of $\tilde{\mathcal{L}}^*(v)$. Notice that $\sum_{v \in V} |\tilde{\mathcal{L}}^*(v)| = O(n^2)$.

Let $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ be a nice subarrangement and let $v \in V$ be such that $|\tilde{\mathcal{L}}(v)| \geq 1$. We say that a splitter S for any face in $\tilde{\mathcal{A}}$ is *balanced with respect to* v if $|\tilde{\mathcal{L}}_i^*(v)| \leq 2|\tilde{\mathcal{L}}^*(v)|/3$ for all $1 \leq i \leq k$, where $\tilde{\mathcal{A}}_i = (\tilde{\mathcal{L}}_i, V, \mathcal{I} \cup r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S)))$, $1 \leq i \leq k$, are the nice subarrangements of $\tilde{\mathcal{A}}$ induced by S. If v is clear from the context, then we simply say that S is *balanced*.

Lemma 3.5. *Given a nice subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$, for any $v \in V$ such that $|\tilde{\mathcal{L}}(v)| \geq 1$, there exists a balanced splitter with respect to v.*

Proof. (See Fig. 20 for an illustration.) Consider a line segment $L \in \tilde{\mathcal{L}}(v)$. L is contained in a face F of the planar subdivision formed by the segments in $\mathcal{I}$. If L has one of its endpoints in the interior of F, then we (temporarily) extend L within F until it hits the boundary of F. Now, by cutting along all (extended or not) line segments in $\tilde{\mathcal{L}}(v)$, we partition F into several subfaces $F_1, \ldots, F_q$. We refer to these cuts as *essential diagonals*. Next, by adding, as necessary, more internal (*non-essential*) diagonals whose endpoints lie on segments in $\mathcal{I}$ we triangulate arbitrarily each face F_i, $i = 1, \ldots, q$. (Note that the non-essential diagonals, in general, are not segments in $\tilde{\mathcal{L}}(v)$ and contain no segments in $\tilde{\mathcal{L}}(v)$.) Clearly, such a triangulation exists. We obtain a partition of F into triangles whose dual graph is a tree T with maximum degree at most three [19].

Now, to each edge e of T we assign a weight $w(e)$ that equals the number of edges in $\tilde{\mathcal{L}}^*(v)$ contained in the diagonal to which e corresponds. Note that $w(e) \leq n - 3$ for each diagonal e, since e can intersect with at most $n - 4$ segments within F, excluding L and the three or more segments forming ∂F. Finally, by directing each edge e of T toward the side having a larger sum of edge weights (breaking ties arbitrarily), we obtain a subcubic directed weighted tree with the following property: there is at least one edge e incident to a sink vertex of T such that removal of e splits T into two subtrees, each having a total edge weight sum of no greater than 2/3 of the total edge weight sum of T.[13] The diagonal to which e corresponds is a desired balanced splitter with respect to v. □

We search a nice subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ using the following recursive divide-and-conquer strategy: Choose a guard $v \in V$ arbitrarily, and find a balanced splitter S with respect to v based on the above lemma, illuminate the segments in $r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S))$ (in addition to those in $\mathcal{I}$), and then, using the same strategy with respect to this v, search each of the nice subarrangements induced by S. Every branch of recursion continues until we arrive at a nice subarrangement $\tilde{\mathcal{A}}_v = (\tilde{\mathcal{L}}_v, V, \mathcal{I}_v)$ such that $\tilde{\mathcal{L}}_v^*(v) = \emptyset$. (We say $\tilde{\mathcal{A}}_v$ is *empty with respect to* v.) We then select another guard $u \in V$ and apply the same recursive strategy to $\tilde{\mathcal{A}}_v$ with respect to u, until we obtain a nice subarrangement that is empty with respect to both v and u. Continuing the above strategy over all guards in an arbitrary order, we eventually arrive at a

[13] The fact of existence of a balanced 1-edge separator in a weighted tree of bounded degree seems to be a folklore result and has already appeared in the literature. However, we failed to affiliate its authorship, and the proof we present is a modification of the proof for 1-node separators in trees, presented by Satish Rao in his lecture notes on foundations of parallel and distributed systems.

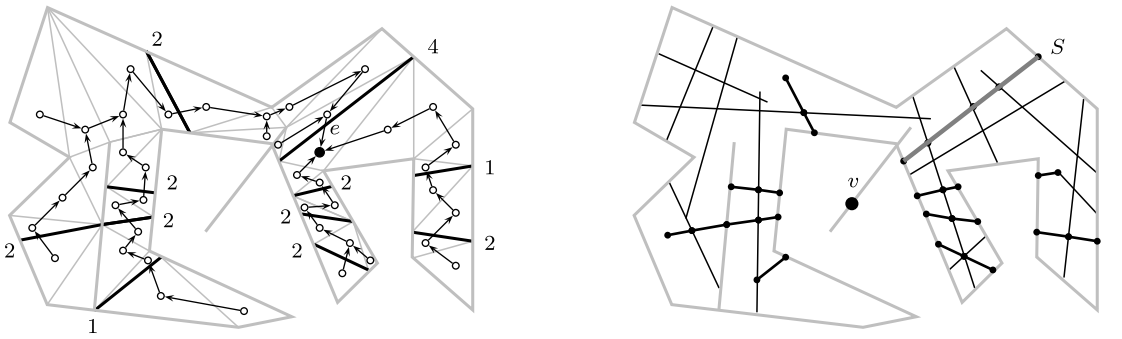

Fig. 20. Illustration for the proof of Lemma 3.5 (continuing with the example in Fig. 18). Essential diagonals are marked with bold black lines and only non-zero weights in the directed dual tree T of the triangulation are shown. Edge e incident to the sink vertex (marked with the solid dot in the left figure) corresponds to a desired balanced splitter S with respect to v in the right figure (note that here, $S \in \tilde{\mathcal{L}}(v)$): there are $9 \leq \frac{2}{3} \cdot |\tilde{\mathcal{L}}(v)| = \frac{2}{3} \cdot 22 = 14\frac{2}{3}$ edges contributed by segments in $\mathcal{L}(v)$ on both sides of S.

subarrangement $\tilde{\mathcal{A}}_V = (\emptyset, V, \mathcal{I}_V)$ that is empty with respect to all guards in V and hence is completely illuminated. By Lemma 3.5 and by the fact that $|\tilde{\mathcal{L}}^*(v)| = O(n^2)$ for each $v \in V$, the depth of the recursion leading to $\tilde{\mathcal{A}}_V$ is $O(g \cdot \log n)$.[14]

What is the total number of searchlights that are sufficient? Assume for now that for every balanced splitter S that we encounter in the above process, the number $x(S)$ of points that the segments in $r(\tilde{\mathcal{X}}(S))$ have in common with S is bounded by some constant $c \geq 0$. Then dividing an arrangement into the subarrangements induced by S requires at most c searchlights, and consequently, no more than $O(g \log n)$ searchlights will be used simultaneously (by all guards, and hence by any single guard) each time a subarrangement that is empty with respect to all guards is reached at a deepest level of recursion. Therefore, the above recursive procedure can be executed successfully if we place $O(g \log n)$ searchlights at every guard, and hence the total number of searchlights used by all guards is $O(g^2 \log n)$.

However, if no such c exists, in particular, if $x(S) = \Theta(n)$ holds, then unless we modify the strategy presented above in which all segments in $r(\tilde{\mathcal{L}}(S) \cup \tilde{\mathcal{X}}(S))$ are illuminated simultaneously, the total number of searchlights needed may exceed $O(g^2 \log n)$. In the next section we overcome this difficulty (informally speaking) by using a sequence of "separators" that separate the sides of S "piece by piece" and isolate the subarrangements induced by S one by one. Each separator consists of $O(1)$ segments, and consequently $O(g^2 \log n)$ remains an upper bound on the total number of required searchlights.

3.2.4. Small separators through splitters

Let $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ be a nice subarrangement, $v \in V$ be any guard and let, as before, F be a bounded face of the planar subdivision formed by the segments in $\mathcal{I}$. Following the previous notation, let S be a balanced splitter of $\tilde{\mathcal{A}}$ with respect to v whose existence is guaranteed by Lemma 3.5. Without loss of generality assume that S is vertical and F forms a simple polygon.[15] Refer to the two induced (sub)faces F_i of F, $i = 1, 2$, as the *left* and *right* faces, respectively. Partition the boundary of F into three parts: the endpoints of S, the (open) left half ∂F_1, and the (open) right half ∂F_2. We say that a line segment $L \in \tilde{\mathcal{L}} \setminus \tilde{\mathcal{L}}(S)$ is *left with respect to* S if either (i) it has an endpoint in ∂F_1, or (ii) it shares an endpoint with S and lies within F_1. Analogously, $L \in \tilde{\mathcal{L}} \setminus \tilde{\mathcal{L}}(S)$ is *right with respect to* S if either (i) it has an endpoint in ∂F_2, or (ii) it shares an endpoint with S and lies within F_2. See Fig. 21(a), which illustrates how to identify left and right segments in $\tilde{\mathcal{L}}$. Note that a segment may be both left and right, and by Observation 3.3, every segment in $\tilde{\mathcal{L}} \setminus \tilde{\mathcal{L}}(S)$ is left or right.

Our algorithm for searching subarrangement $\mathcal{A}$ is based upon the concept of a separator of a curve, and an (S, x)-curve, where $x \in S$ belongs to a segment in $\tilde{\mathcal{X}}(S)$. Before we formally introduce them, let us sketch the role they play in our final algorithm (see Fig. 22 for an illustration). For searching a nice subarrangement, we determine all points x_i that S has in common with $\tilde{\mathcal{X}}(S)$. Having those points increasingly sorted according to their y-coordinates, we move along S starting with x_1 and visiting all consecutive points x_i one by one. Once we are done processing some point x_i, it is guaranteed that some face $F_i \subset F$ that contains the segment $\overline{bx_i}$, where b is the bottom endpoint of S, is clear. The border of F_i is formed by segments of $\mathcal{I}$ (this is the part that coincides with the boundary of the nice subarrangement) and an (S, x_i)-curve $\mathcal{C}_i$ (this is the part of the boundary of F_i that lies within F and intersects some segments of the subarrangment; it may coincide with some segments in $\mathcal{I}$). Clearly, such an (S, x_i)-curve is an artificial object from the perspective of a search strategy because many points of such a curve do not belong to any segment in $\tilde{\mathcal{L}} \cup \mathcal{I}$. However, the points of $\mathcal{C}_i$ that do not belong to segments in $\tilde{\mathcal{L}} \cup \mathcal{I}$ cannot be used by a fugitive and hence in order to guarantee that the fugitive cannot cross $\mathcal{C}_i$, it is enough to illuminate all its points that belong to some segments in $\tilde{\mathcal{L}}$, i.e., to the segments that form a *separator* of a curve. In our strategy, provided that all segments within F_i are clear and all segments of a separator of $\mathcal{C}_i$ are illuminated

[14] Observe that, for a segment $\tilde{L} \in \tilde{\mathcal{L}}(w)$, $w \in V$, by illuminating its representative $r(\tilde{L}) \in \mathcal{L}(w)$, it may happen that we illuminate several other segments in $\tilde{\mathcal{L}}(w)$. Thus, our recursion might have stopped earlier.

[15] If F forms a weakly simple polygon then, in order to define left and right segments properly (for the definition of left and right segments, see below), we have to define ∂F_1 and ∂F_2 more carefully, since they may overlap.

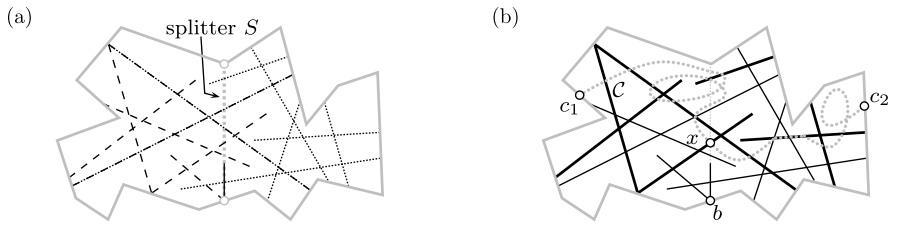

Fig. 21. (a) A subarrangement $\tilde{A}$ and its left (dashed) and right (dotted) segments with respect to a splitter S. (b) A splitter S, a (dotted gray) curve $\mathcal{C}$ and its (S, x)-separator (the black bold lines).

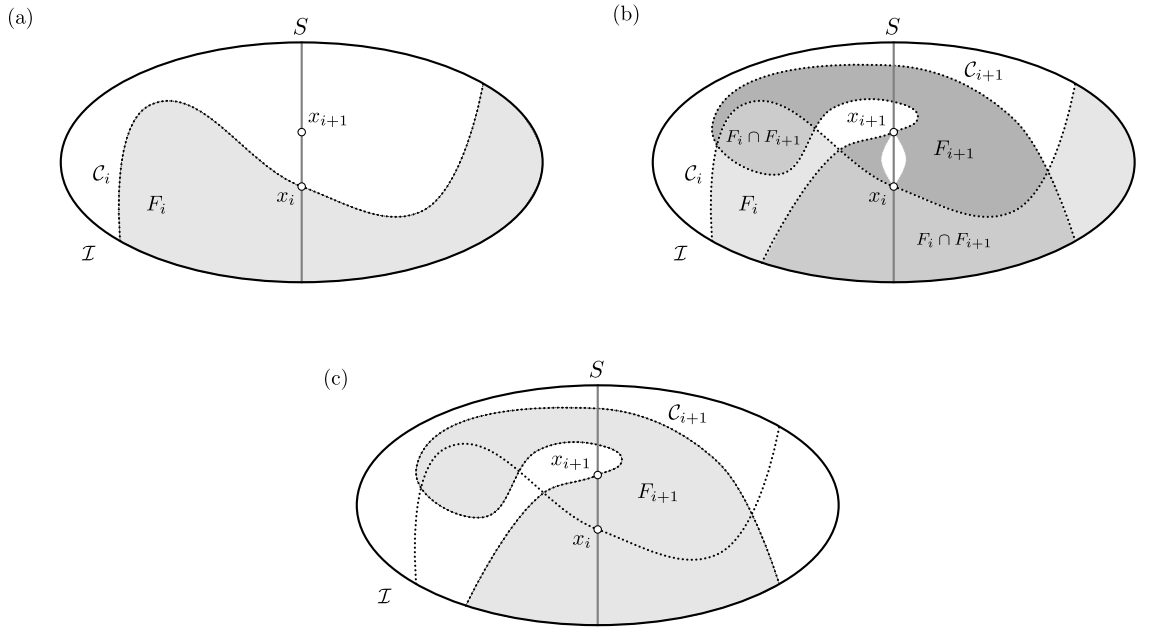

Fig. 22. (a) An illustration of the idea of the final algorithm. (a) (Separator of) $\mathcal{C}_i$ is illuminated and all segments within face F_i are recursively searched. (b) (Separator of) $\mathcal{C}_i$ remains illuminated, (separator of) $\mathcal{C}_{i+1}$ starts being illuminated, and we recursively search all segments within $F_{i+1} \setminus F_i$. Observe that no segment in $\mathcal{L}$ intersects the open segment $\overline{x_i x_{i+1}}$. (c) (Separator of) $\mathcal{C}_i$ stops being illuminated, (separator of) $\mathcal{C}_{i+1}$ remains illuminated, and all segments within F_{i+1} are recursively searched.

along with the ones in $\mathcal{I}$, the algorithm starts clearing segments within the next face $F_{i+1} \subset F$. To that end, the segments in a separator of the (S, x_{i+1})-curve $\mathcal{C}_{i+1}$ are first illuminated. Then, the nice subarrangements between $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ are recursively cleared (i.e., the ones within $F_{i+1} \setminus F_i$). The processing of x_{i+1} finishes by keeping illuminated only segments in $\mathcal{I}$ and the separator of $\mathcal{C}_{i+1}$. The formal description of our algorithm is provided in Section 3.2.5, and this section is devoted to a proof that one can find for each point x_i the corresponding (S, x_i)-curve having a separator of fixed size.

Let $\mathcal{C} \subset F$ be a curve that may have points in common with the boundary of F. A subset $\mathcal{S} \subseteq \tilde{\mathcal{L}}$ of segments forms a *separator of* $\mathcal{C}$ if every point that $\mathcal{C}$ has in common with a segment in $\tilde{\mathcal{L}}$ belongs to some segment in $\mathcal{S} \cup \mathcal{I}$. Clearly, any curve $\mathcal{C} \subset F$ has a separator, and a *minimum* separator of $\mathcal{C}$ is one having the minimum cardinality.

Consider now a non-illuminated point $x \in S$ that belongs to a segment in $\tilde{\mathcal{X}}(S)$. Let $\mathcal{C} \subset F$ be a curve such that (see Fig. 21(b)):

a) $x \in \mathcal{C} \cap S$;
b) $\mathcal{C}$ does not intersect the open line segment $\overline{bx}$, where b is the bottom endpoint of S;
c) both endpoints c_1 and c_2 of $\mathcal{C}$ belong to the boundary $\mathcal{I}$;
d) $c_1 \in \partial F_1$ and $c_2 \in \partial F_2$; in particular, if $\mathcal{C}$ is closed, then $\mathcal{C}$ touches $\mathcal{I}$ at one of the endpoints of S (and then both c_1 and c_2 correspond to this endpoint).

Any curve that satisfies the above condition is called an (S, x)-*curve*. We emphasize that $\mathcal{C}$ may touch and/or move along some segments in $\mathcal{I}$ several times.

Clearly, there exists a curve $\mathcal{C}$ satisfying (a–d), and we call any separator of such $\mathcal{C}$ an (S, x)-*separator*. The *weight* of $\mathcal{C}$ is then defined as $|\mathcal{S}|$, where $\mathcal{S}$ is a minimal (S, x)-separator of $\mathcal{C}$. For two points $x, y \in \mathcal{C}$, let $\mathcal{C}(x, y)$ denote the (closed) part of $\mathcal{C}$ between points x and y.

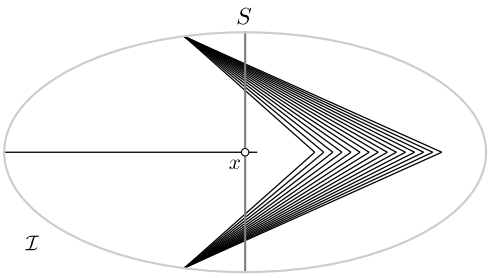

Fig. 23. There is no (S,x)-curve of weight less than 16 that crosses the splitter S exactly once.

We now prove a crucial lemma in our analysis. In particular, in this lemma we prove two claims. The first claim is about the existence of a small (of fixed size) (S,x)-separator for each non-illuminated point $x \in S$ that belongs to a segment in $\tilde{\mathcal{X}}(S)$. The second claim is more subtle and, informally speaking, it is related to the way how the separators are used in our final algorithm. To give a sketch of this part of our method, in our algorithm for two consecutive (distinct) points x'_1 and x'_2 on S that belong to two segments in $\tilde{\mathcal{X}}(S)$, the relevant (S,x'_1)- and (S,x'_2)-separators $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, are illuminated and we recursively search all subarrangements 'between' those separators. To bound the depth of recursion by $O(\log n)$, we need to guarantee that separators $\mathcal{S}_1$ and $\mathcal{S}_2$ are 'safe' (see the definition below). Note that the separator size only influences the number of guards used in each recursion level, but not the depth of recursion. The crucial condition that allows us to bound the depth of recursion is related to the number of edges in $\tilde{\mathcal{L}}^*(v)$, which can be informally explained as follows. Recall that the splitter S is selected carefully to be balanced, i.e., on each of its sides there is a constant fraction of all edges in $\tilde{\mathcal{L}}^*(v)$. On one side of S we have edges in $\tilde{\mathcal{L}}^*(v)$ contained in left segments from $\tilde{\mathcal{L}}(v)$, and on the other side of S – the edges in $\tilde{\mathcal{L}}^*(v)$ contained in right segments from $\tilde{\mathcal{L}}(v)$. Since the (S,x)-curves are by definition allowed to cross the line segment $\overline{xt}$, where t is the top endpoint of S, it could potentially happen that an edge in $\tilde{\mathcal{L}}^*(v)$ contained in a left segment and an edge in $\tilde{\mathcal{L}}^*(v)$ contained in a right segment could end up in the same nice subarrangement for which a recursive call is performed. We want to exclude this possibility to guarantee the recursion depth of $O(\log n)$, and the selection of 'safe' separators defined below will provide us this desired property.

Let $\mathcal{C}$ be an (S,x)-curve. Recall that S is a balanced splitter with respect to $v \in V$, and $x \in S$ is a non-illuminated point that belongs to a segment in $\tilde{\mathcal{X}}(S)$. If $\mathcal{C}$ crosses S only at x, then any separator of $\mathcal{C}$ is called *safe*. Suppose now that $\mathcal{C}$ crosses S three times at points x, x''_1 and x''_2, where x is closer to b than x''_1, and x''_1 is closer to b than x''_2 (see Fig. 31 for an illustration). Let $\mathcal{S}$ be a separator of $\mathcal{C}$. For any set of segments $\mathcal{S}' \subseteq \tilde{\mathcal{L}}$, $\mathcal{S} \cup \mathcal{S}'$ is clearly an (S,x)-separator, which is not minimal if $\mathcal{S}' \setminus \mathcal{S} \neq \emptyset$. Now, let F_x be the subface of F bounded by $\mathcal{C}$ and $\overline{x''_1 x''_2}$ (again, see Fig. 31). We say that the (S,x)-separator $\mathcal{S} \cup \mathcal{S}'$ is *safe* for $\mathcal{C}$ if no subarrangement $\tilde{\mathcal{A}}_x = (\tilde{\mathcal{L}}_x, V, \mathcal{I} \cup r(\mathcal{S} \cup \mathcal{S}'))$ that has a non-illuminated segment $L \in \tilde{\mathcal{L}}_x$ such that $L \cap F_x \neq \emptyset$, has *both* a subsegment of a right segment *and* a subsegment of a left segment in $\tilde{\mathcal{L}}(v)$ with respect to S.

Note that in the above definition we focus on curves $\mathcal{C}$ that cross S either once or three times. This is a consequence of the proof of the following lemma. In particular, one could allow arbitrary number of crossings as long as it is guaranteed that the crucial requirement is met, that is: No subarrangement $\tilde{\mathcal{A}}_x = (\tilde{\mathcal{L}}_x, V, \mathcal{I} \cup r(\mathcal{S} \cup \mathcal{S}'))$ that has a non-illuminated segment $L \in \tilde{\mathcal{L}}_x$ such that $L \cap F_x \neq \emptyset$, has *both* a subsegment of a right segment *and* a subsegment of a left segment in $\tilde{\mathcal{L}}(v)$ with respect to S. However, in our following proof we are able to guarantee that we can always find $\mathcal{C}$ with either one or three crossing, and for this reason we keep the above definition simpler by excluding arbitrary number of crossings.

Lemma 3.6. *Let S be a splitter for $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ with respect to $v \in V$. For each $x \in S$ that belongs to a segment in $\tilde{\mathcal{X}}(S)$, there exists an (S,x)-curve $\mathcal{C}$ with a safe separator of size at most 15.*

Briefly, the idea of the proof is as follows. We take an (S,x)-curve that crosses the splitter S exactly once and has minimum weight. If the weight is bounded by 15, then we have found a desired safe (S,x)-separator. (Fig. 23 provides an example where any (S,x)-curve crossing the splitter S exactly once is of weight at least 16.) Otherwise, we modify this curve so that the new (S,x)-curve will cross S three times and have a safe separator. Moreover, the latter is of size at most 15 as required.

Proof. Of all the simple (S,x)-curves that have one point in common with the internal part of S, choose one, $\mathcal{C}$ with endpoints c_1 and c_2, of minimum weight. Assume that $\mathcal{C}$ is not closed, i.e., $c_1 \in \partial F_1$ and $c_2 \in \partial F_2$; we omit a similar argument for the case in which $\mathcal{C}$ is closed, i.e., $c_1 = c_2$.

Consider now a minimum separator $\mathcal{S}$ of $\mathcal{C}$ and let $L_1 \in \mathcal{S}$ be a line segment such that $x \in L_1$. Suppose that L_1 is left. Now, if $\mathcal{C}$ does not go along L_1 from c_1 to x within F_1, that is, $\mathcal{C}(c_1,x) \cap F_1 \neq L_1 \cap F_1$, we modify $\mathcal{C}$ by replacing $\mathcal{C}(c_1,x)$ with the line segment $\overline{c'_1 x}$, where $c'_1 \in \partial F_1$ is an endpoint of L_1. The new $\mathcal{C}$ is a simple (S,x)-curve, is of the same weight, and still has one point in common with the internal part of S. Hence in the following, we assume that $\mathcal{C}(c_1,x) \cap F_1 = L_1 \cap F_1$.

Now, since $\mathcal{C}$ is simple, $\mathcal{C}(c_1,x)\cap F_1 = L_1\cap F_1$, and $\mathcal{C}$ has one point in common with the internal part of S, without loss of generality we may additionally assume that for any $L\in\tilde{\mathcal{L}}$, $L\cap\mathcal{C}\cap F_2$ has at most one connected component, i.e., it is an empty set, a point, or a segment. Indeed, if this is not the case, then there exist two distinct points $y,z\in L\cap\mathcal{C}\cap F_2$, where y appears before z when $\mathcal{C}$ is traversed from c_1 to c_2. Then, $\mathcal{C}$ can be replaced with $\mathcal{C}(c_1,y)\cup\overline{yz}\cup\mathcal{C}(z,c_2)$, with the same separator $\mathcal{S}$ of $\mathcal{C}$.

The above assumption, in particular, the fact that no three segments in $\mathcal{S}$ may now have a point in $\mathcal{C}\cap F_2$ in common, allows us to order the segments in $\mathcal{S}=\{L_1,\ldots,L_k\}$ according to their first appearance along $\mathcal{C}\cap F_2$ when $\mathcal{C}$ is traversed from c_1 to c_2.

Recall that L_1 is left. Now we prove that all but possibly L_k are also left segments. If there is no right segment in $\mathcal{S}$ or $k=1$, then the claim follows; so let $j\in\{2,\ldots,k\}$ be the smallest index such that L_j is right. We argue that there exists $y\in F_2\cap\mathcal{C}\cap L_j$. Suppose for a contradiction that $\mathcal{C}$ and L_j have no point in common in F_2. But then, $\mathcal{C}(c_1,x)\cap F_1 = L_1\cap F_1$ and the fact that $\mathcal{C}$ has one point in common with S imply that $\mathcal{S}\setminus\{L_j\}$ is an (S,x)-separator of $\mathcal{C}$, which gives the required contradiction with the choice of $\mathcal{C}$. So let y be the first point in F_2 that $\mathcal{C}$ and L_j have in common when $\mathcal{C}$ is traversed from c_1 to c_2.

We now construct a curve $\mathcal{C}'$ from $\mathcal{C}$ by replacing $\mathcal{C}(y,c_2)$ by the line segment $\overline{yc_2'}$, where $c_2'\in\partial F_2$ is an endpoint of L_j. Again, since $\mathcal{C}'$ is a simple (S,x)-curve, we obtain $j=k$ by the minimality of $\mathcal{C}$. This proves that $L_1,\ldots,L_{k-1}$ are left segments, as required.

Recall that we assumed L_1 to be a left segment, and in such a case $\mathcal{S}$ is said to be *left* (see Fig. 24(a)). On the other hand, if L_1 is right, by similar arguments, in particular, by ordering the segments in $\mathcal{S}=\{L_1,\ldots,L_k\}$ according to their first appearance along $\mathcal{C}\cap F_1$ when $\mathcal{C}$ is traversed from c_2 to c_1, we may conclude that segments $L_1,\ldots,L_{k-1}$ are right segments; in such a case $\mathcal{S}$ is said to be *right*. (Notice that $\mathcal{S}$ may be both left and right).

Assume in the following that $\mathcal{S}$ is left. (The argument for the other case is symmetric and is omitted.) Clearly, if $|\mathcal{S}| = k\le 15$, then there is nothing to prove, so assume that $k>15$.

The remaining part of the proof is divided into two parts. In the first part we obtain a new (S,x)-curve (that crosses S three times) and has weight at most 15. In the second part, we construct a safe (S,x)-separator for this curve, also of size at most 15, as stated in the lemma.

We start the first part of the proof by introducing some notation. Let $\tilde{\mathcal{L}}^+\subseteq\tilde{\mathcal{X}}(S)\subseteq\tilde{\mathcal{L}}$ be the set of left segments that intersect the vertical open line segment $\overline{xt}$, where t is the top endpoint of S, and analogously, let $\tilde{\mathcal{L}}^-\subseteq\tilde{\mathcal{X}}(S)\subseteq\tilde{\mathcal{L}}$ be the set of left segments that intersect the vertical line segment $\overline{bx}$, where b is the bottom endpoint of S; see Fig. 24(b) for an illustration and notice that $L_1\in\tilde{\mathcal{L}}^-$. Next, let $\tilde{\mathcal{R}}\subset\tilde{\mathcal{L}}$ denote the set of all right segments. We say that two segments $L',L''\in\tilde{\mathcal{L}}^-\cup\tilde{\mathcal{L}}^+\cup\tilde{\mathcal{R}}$ form a *good pair* (L',L'') if and only if a point of L' can be connected to a point of L'' with a "free" curve, where a simple curve is called *free* if it is contained in F_2 and has no point in common with a segment in $\tilde{\mathcal{L}}$, except for its endpoints. In particular, if L' and L'' intersect, then they form a good pair (in this case, the free curve degenerates to the point of their intersection).

For $i\in\{1,\ldots,k\}$, let $\mathcal{X}(i)$ denote the set of all segments in $\tilde{\mathcal{L}}$ having a point in common with $L_i\cap\mathcal{C}\cap F_2$. (See Fig. 24(c) for an illustration; we emphasize that $\mathcal{X}(i)$, $\mathcal{X}(i+1)$ and $\mathcal{X}(i+2)$ may have an element in common.) Next, for $i,j\in\{1,\ldots,k\}$, $i\le j$, let $\mathcal{X}_i^j=\bigcup_{t=i}^j\mathcal{X}(t)$. Observe that every segment that intersects $\mathcal{C}\cap F_2$ belongs to $\mathcal{X}_1^k$.

Before continuing, let us provide some intuition regarding the above definitions. Good pairs play an important role in our analysis because some types of good pairs are proved to be non-existent while we argue that some other good pairs have to exist — and we utilize the latter ones to construct our new (S,x)-curve. The fact that certain good pairs are not allowed follows from observations that otherwise one could use them to construct a 'bypass' that provides a new (S,x)-curve having a separator of smaller size than $\mathcal{C}$, contradicting the minimality of $\mathcal{C}$. For example, we argue (Claim 3.8) that there is no good pair in $\mathcal{X}_1^{k-3}\times\tilde{\mathcal{R}}$. Intuitively, such a good pair in $\mathcal{X}(i)\times\tilde{\mathcal{R}}$ for some $i\in\{1,\ldots,k-3\}$ gives us a 'shortcut' that leads from L_i directly to ∂F_2 bypassing $L_{i+1},\ldots,L_k$ and thus providing a smaller separator. Similarly, a good pair in $\mathcal{X}_1^m\times\mathcal{X}_{m+4}^k$ for some $m\in\{1,\ldots,k-4\}$ can be used to bypass the segments $L_{m+1},\ldots,L_{m+3}$ (see Claim 3.9). Having excluded those good pairs, we prove in Claims 3.10 and 3.11 that two certain good pairs need to exist. Those pairs will be used to construct our new curve by taking $\mathcal{C}$ and creating 'bypass' that starts at one of the segments L_2,L_3,L_4 and ends at one of the segments L_{k-6},L_{k-5},L_{k-4}. This 'bypass' introduces the part of the new curve that crosses S twice, which (including the intersection of S and $\mathcal{C}$ at x) implies that the new curve will cross S three times. (We point out that the new curve may have points in common with the boundary $\mathcal{I}$ of F.)

Claim 3.7. *No segment in $\mathcal{X}_1^{k-2}$ can be connected with a free curve to ∂F_2.*

Proof of Claim 3.7. (See Fig. 25.) Contrary to our claim, suppose there exists a free curve $\mathcal{C}'$ connecting a segment $L'\in\mathcal{X}(i)$ with ∂F_2, for some $i\in\{1,\ldots,k-2\}$; notice that $\mathcal{C}'$ may intersect $\mathcal{C}$ at several points. Let p_1 be the endpoint of $\mathcal{C}'$ such that $p_1\in L'$ and let $p_0\in L'\cap\mathcal{C}\cap F_2$ be the closest point to p_1 (along L'); note that $p_0=p_1$ may hold. The concatenation of $\mathcal{C}(c_1,p_0)$, $\overline{p_0p_1}$ and $\mathcal{C}'$ includes a simple (S,x)-curve that intersects S exactly once, with $\{L_1,\ldots,L_i,L'\}$ as its (S,x)-separator. (Notice that this concatenation may have a loop.) Since $i+1<k$, we obtain a contradiction with the minimality of $\mathcal{C}$. □

Claim 3.8. *There is no good pair in $\mathcal{X}_1^{k-3}\times\tilde{\mathcal{R}}$.*

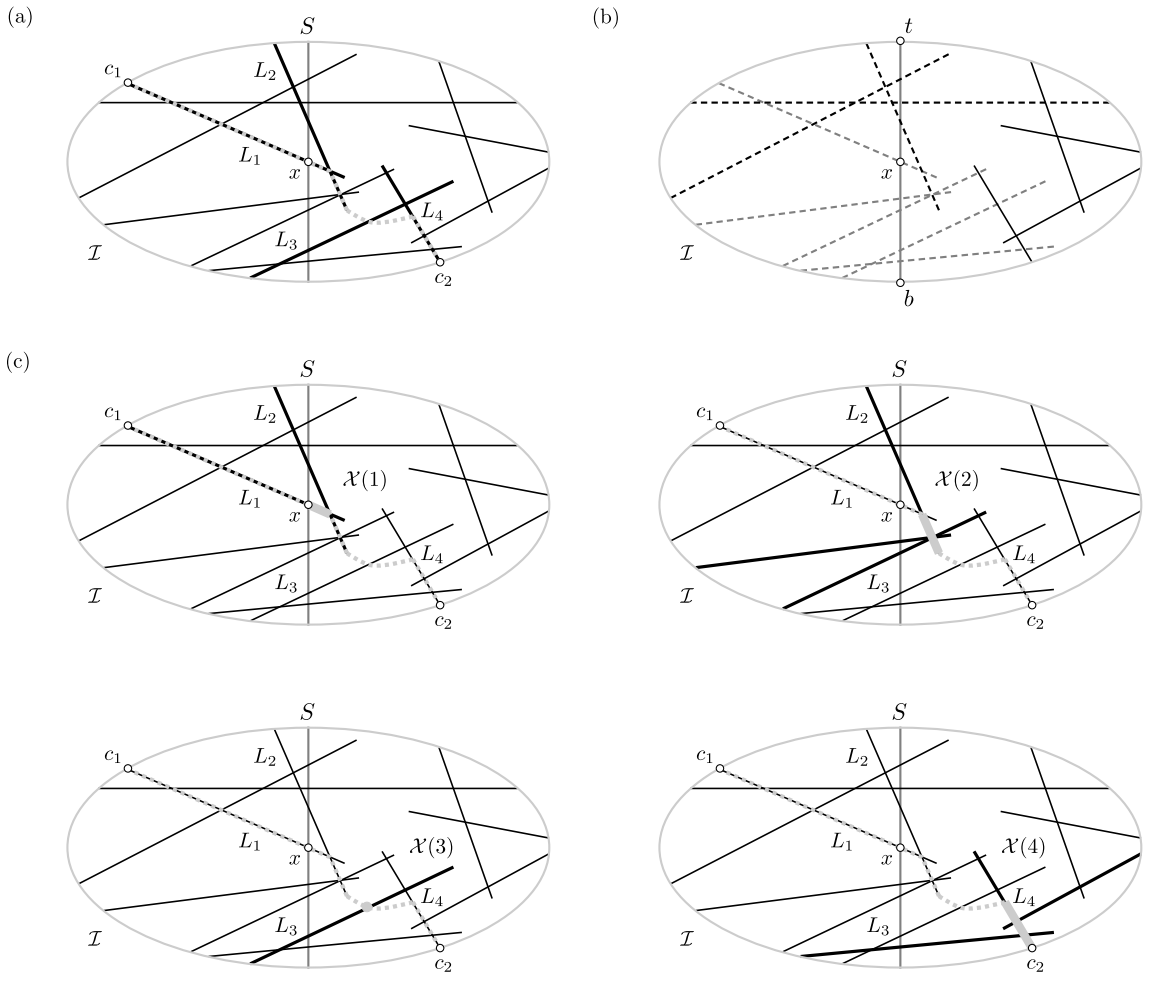

Fig. 24. (a) $\mathcal{S} = \{L_1, L_2, L_3, L_4\}$ is a left separator: L_1, L_2, L_3 are left segments and L_4 is right. (b) Sets $\tilde{\mathcal{L}}^+$ and $\tilde{\mathcal{L}}^-$ (black and gray dashed lines, respectively). (c) Sets $\mathcal{X}(1)$, $\mathcal{X}(2)$, $\mathcal{X}(3)$ and $\mathcal{X}(4)$ (marked with bold lines).

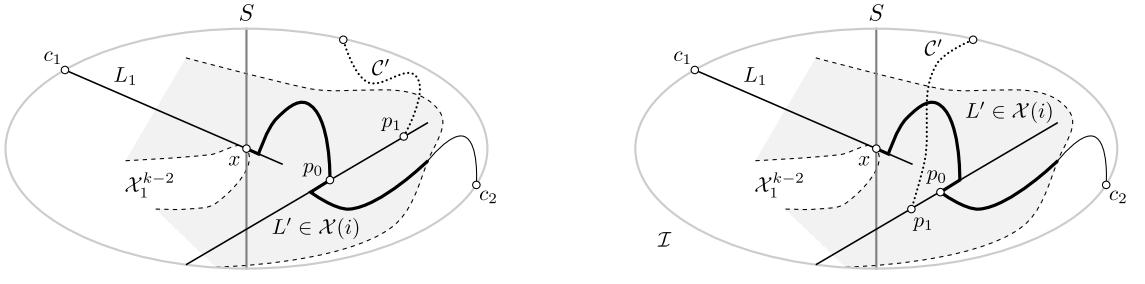

Fig. 25. An illustration of the proof of Claim 3.7. Here, $L' \in \tilde{\mathcal{L}}^-$. The gray region indicates the possible range of left segments in $\mathcal{X}_1^{k-2}$.

Proof of Claim 3.8. (See Fig. 26.) Contrary to our claim, suppose there exists a free curve $\mathcal{C}'$ connecting a segment $L' \in \mathcal{X}(i)$ with a segment $L'' \in \tilde{\mathcal{R}}$, for some $i \in \{1, \ldots, k-3\}$; again note that $\mathcal{C}'$ may intersect $\mathcal{C}$ at several points. Let p_1 and p_2 be the endpoints of $\mathcal{C}'$ such that $p_1 \in L'$ and let $p_0 \in L' \cap \mathcal{C} \cap F_2$ be the closest point to p_1 (along L'); note that $p_0 = p_1$ may hold. Then the concatenation of $\mathcal{C}(c_1, p_0)$, $\overline{p_0 p_1}$, $\mathcal{C}'$ and $\overline{p_2 z}$, where $z \in \partial F_2$ is the endpoint of L'', includes a simple (S, x)-curve that intersects S exactly once, with $\{L_1, \ldots, L_i, L', L''\}$ as its (S, x)-separator. (Again notice that the aforementioned concatenation may have a loop.) Since $i + 2 < k$, we obtain a contradiction with the minimality of $\mathcal{C}$. □

Claim 3.9. *There is no good pair in $\mathcal{X}_1^m \times \mathcal{X}_{m+4}^k$ for each $m \in \{1, \ldots, k-4\}$.*

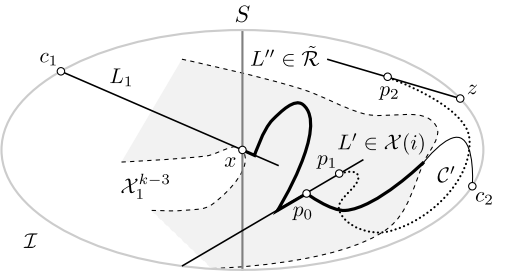

Fig. 26. An illustration of the proof of Claim 3.8. Here, $L' \in \tilde{\mathcal{L}}^-$. The gray region indicates the possible range of left segments in $\mathcal{X}_1^{k-3}$.

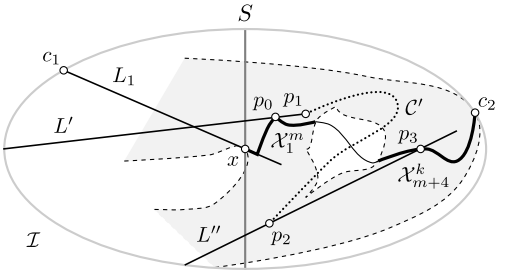

Fig. 27. An illustration of the proof of Claim 3.9. Here, $L' \in \tilde{\mathcal{L}}^+$ and $L'' \in \tilde{\mathcal{L}}^-$. The gray region indicates the possible range of left segments in $\mathcal{X}_1^m \cup \mathcal{X}_{m+4}^k$.

Proof of Claim 3.9. (See Fig. 27.) Contrary to our claim, suppose there exists a good pair $(L', L'') \in \mathcal{X}(i) \times \mathcal{X}(j)$, for some $i \in \{1, \ldots, m\}$ and $j \in \{m+4, \ldots, k\}$. Let $\mathcal{C}'$ be a free curve with endpoints p_1 and p_2 that connects L' and L'', where $p_1 \in L'$ and $p_2 \in L''$; again notice that $\mathcal{C}'$ may intersect $\mathcal{C}$ in several points. Let $p_0 \in L' \cap \mathcal{C} \cap F_2$ be the closest point to p_1 (along L'), and let $p_3 \in L'' \cap \mathcal{C} \cap F_2$ be the closest point to p_2 (along L''). The concatenation of $\mathcal{C}(c_1, p_0)$, $\overline{p_0 p_1}$, $\mathcal{C}'$, $\overline{p_2 p_3}$ and $\mathcal{C}(p_3, c_2)$ includes a simple (S, x)-curve that intersects S exactly once, with $\{L_1, \ldots, L_i\} \cup \{L', L''\} \cup \{L_j, \ldots, L_k\}$ as its (S, x)-separator. Since the size of the latter is equal to $i + 2 + (k - j + 1)$, which is strictly smaller than $k = |\mathcal{S}|$, we obtain a contradiction with the minimality of $\mathcal{C}$. □

Claim 3.10. *There exists a good pair $(L_1', L_1'') \in \tilde{\mathcal{L}}^- \times \tilde{\mathcal{L}}^+$ such that $\{L_1', L_1''\} \cap \mathcal{X}_2^4 \neq \emptyset$.*

Proof of Claim 3.10. Note that $\mathcal{X}_2^4 \neq \emptyset$ because $\{L_2, L_3, L_4\} \subseteq \mathcal{X}_2^4$. Also, all segments in $\mathcal{X}_2^4$ are left because $\mathcal{S}$ is left. If $\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^- \neq \emptyset$ and $\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^+ \neq \emptyset$, then the claim follows. Indeed, consider a pair $(L_1', L_1'') \in (\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^-) \times (\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^+)$ that minimizes the length of $\mathcal{C}(p', p'')$, where $p' \in L_1' \cap \mathcal{C}$ and $p'' \in L_1'' \cap \mathcal{C}$, respectively. Then, (L_1', L_1'') is a required good pair.

Assume without loss of generality that $\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^+ = \emptyset$ (the case where $\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^- = \emptyset$ is symmetric). Hence, $\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^- \neq \emptyset$ because $(\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^-) \cup (\mathcal{X}_2^4 \cap \tilde{\mathcal{L}}^+) = \mathcal{X}_2^4$ by Claim 3.8. Assign now colors to line segments as follows:

- gray is assigned to line segments in $\mathcal{X}_1^1$;
- red is assigned to line segments in $\mathcal{X}_2^4$;
- black is assigned to line segments in $\mathcal{X}_5^k$;
- blue is assigned to line segments in $\tilde{\mathcal{L}}^+$;
- green is assigned to line segments $\mathcal{I} \cup \tilde{\mathcal{R}}$.

Notice that not all segments in $\tilde{\mathcal{L}}^-$ have a color assigned, i.e., segments that do not intersect $\mathcal{C}$ in F_2, or in other words, those in $\tilde{\mathcal{L}}^- \setminus \mathcal{X}_1^k$. On the other hand, some segments may have more than one color assigned. Suppose for a contradiction that the claim does not hold, i.e., there is no good pair $(L_1', L_1'') \in \tilde{\mathcal{L}}^- \times \tilde{\mathcal{L}}^+$ such that $\{L_1', L_1''\} \cap \mathcal{X}_2^4 \neq \emptyset$.

Let F' be the subface of F bounded by $\mathcal{C}$ and the boundary of F such that $\overline{xt} \subset F'$. By Claims 3.7 and 3.8, there is no good pair (L', L'') of segments such that L' has been assigned color red and L'' has been assigned color green. Consequently, there exists in $F_2 \cap F'$ a simple polyline $\mathcal{P}$ with both endpoints on $\mathcal{C}$ separating red segments from green segments such that each point of $\mathcal{P}$ either belongs to no segment or belongs to a gray or a black segment, and $\mathcal{P}$ contains a point of a gray segment and contains a point of a black segment. (See Fig. 28.) This is due to the following observations: first, segments (in $\mathcal{L}^-$) with no color have no point in common with $F_2 \cap F'$ since none of these segments intersects either $\mathcal{C}$ in F_2 or $\overline{xt}$; second, there is no good pair formed by a red and blue segments. In such $\mathcal{P}$, there exist two points p' and p'' that belong to gray and black segments L' and L'', respectively, and the open curve $\mathcal{P}(p', p'')$ has no point in common with any segment. This implies that (L', L'') is a good pair, which then contradicts Claim 3.9. □

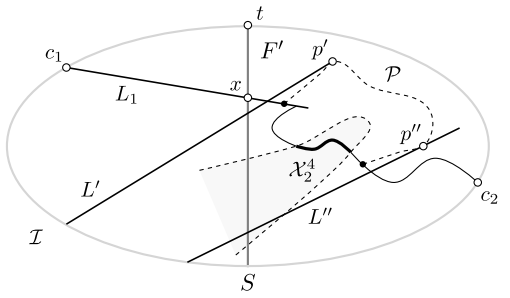

Fig. 28. An illustration of the proof of Claim 3.10. The gray region indicates the possible range of left segments in $\mathcal{X}_2^4$.

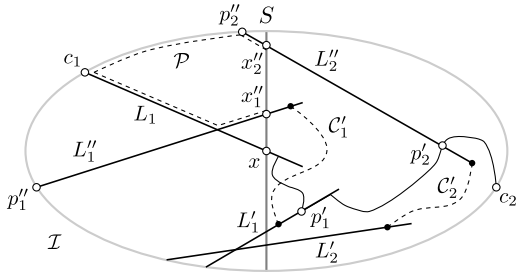

Fig. 29. An illustration of the construction of an (S, x)-curve $\tilde{\mathcal{C}}$ that intersects S more than once but at most three times and has a separator $\tilde{\mathcal{S}}$ of size at most 15.

By the same arguments as above, we may derive the following claim.

Claim 3.11. *There exists a good pair* $(L_2', L_2'') \in \tilde{\mathcal{L}}^- \times \tilde{\mathcal{L}}^+$ *such that* $\{L_2', L_2''\} \cap \mathcal{X}_{k-6}^{k-4} \neq \emptyset$.

Now, having proved the above claims, we continue by constructing the desired (not necessarily simple) (S, x)-curve $\tilde{\mathcal{C}}$ of weight at most 15 (see Fig. 29 for an illustration). Let $(L_1', L_1'') \in \tilde{\mathcal{L}}^- \times \tilde{\mathcal{L}}^+$ be a good pair from Claim 3.10 and let $\mathcal{C}_1'$ be a free curve that connects L_1' and L_1''. Similarly, let $(L_2', L_2'') \in \tilde{\mathcal{L}}^- \times \tilde{\mathcal{L}}^+$ be a good pair from Claim 3.11 and let $\mathcal{C}_2'$ be a free curve that connects L_2' and L_2''. Next, let $\mathcal{C}_1 \subseteq L_1' \cup L_1'' \cup \mathcal{C}_1'$ be the simple curve whose one endpoint, denoted by p_1'', is the endpoint of L_1'' in ∂F_1 and whose other endpoint, denoted by p_1', is the closest point to p_1'' (along $\mathcal{C}_1$) in $\mathcal{C}$. Similarly, let $\mathcal{C}_2 \subseteq L_2' \cup L_2'' \cup \mathcal{C}_2'$ be the simple curve whose one endpoint, denoted by p_2'', is the endpoint of L_2'' in ∂F_1 and whose other endpoint, denoted by p_2', is the closest point to p_2'' (along $\mathcal{C}_1$) in $\mathcal{C}$. Finally, let $\mathcal{P}$ be the shortest (simple) curve contained in $L_1 \cup L_1'' \cup L_2'' \cup \partial F_1$ that connects points x_1'' and x_2'', where $x_1'' = S \cap \mathcal{C}_1$ and $x_2'' = S \cap \mathcal{C}_2$.

Now, we observe that the concatenation of $\mathcal{C}(c_1, p_1')$, $\mathcal{C}_1(p_1', x_1'')$, $\mathcal{P}$, $\mathcal{C}_2(x_2'', p_2')$ and $\mathcal{C}(p_2', c_2)$ includes an (S, x)-curve $\tilde{\mathcal{C}}$ with a possible overlapping only along $L_1 \cap F_1$. (Recall that c_1 and c_2 are the endpoints of $\mathcal{C}$, and $c_1 \in L_1$.) And $\tilde{\mathcal{C}}$ has a separator $\tilde{\mathcal{S}}$ such that $\tilde{\mathcal{S}} \subseteq \{L_1, \ldots, L_4, L_1', L_1'', L_2', L_2'', L_{k-6}, \ldots, L_{k-1}, L_k\}$. Since $|\tilde{\mathcal{S}}| \leq 15$, the proof is completed.

In the remaining part of the proof, we construct an (S, x)-separator that is safe for $\tilde{\mathcal{C}}$. This separator consists of the segments in $\tilde{S}$ and the two good pairs L_1', L_1'' and L_2', L_2'', respectively. We emphasize that the separator $\tilde{S}$ itself may not be safe. Informally, the reason for this is the existence of a nice subarrangement that may contain segments that lie on both sides of S because $\tilde{\mathcal{C}}$ crosses S more than once (we refer here to the subarrangement from the definition of a 'safe' separator). This subarrangement may simultaneously contain subsegments of left and right segments in $\tilde{\mathcal{L}}(v)$. Thus, our efforts will focus on proving that the addition of L_1', L_1'', L_2', L_2'' to $\tilde{S}$ will further subdivide this subarrangement into nice subarrangements with required properties. More precisely, we need to focus our analysis on a nice subarrangement $\tilde{\mathcal{A}}_x$ created by this subdivision and consisting, again, segments on both sides of S. The $\tilde{\mathcal{A}}_x$ may contain a subsegment of a left segment in $\tilde{\mathcal{L}}(v)$ but we argue that no subsegment of a right segment from $\tilde{\mathcal{L}}(v)$ is present in $\tilde{\mathcal{A}}_x$.

We start with the following technical lemma.

Claim 3.12. *There is no good pair* $(L', L'') \in \{L_1', L_1''\} \times \{L_2', L_2''\}$.

Sketch of proof. The proof follows by arguments similar to those in the proof of Claim 3.9. Namely, if such a pair existed, we could construct a "bypass" (curve) of a minimum separator of size at most four (contributed by segments in $\{L_1', L_1'', L_2', L_2''\}$), which together with segments in $\mathcal{X}_1^4 \cup \mathcal{X}_{k-6}^k$ would imply the existence of a simple (S, x)-curve of weight at most $4+4+7 = 15$ (since $k \geq 16$) that intersects S exactly once. This gives a contradiction with the minimality of $\mathcal{C}$. We omit the details. □

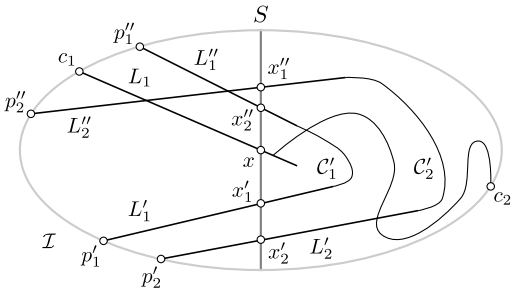

Fig. 30. $\mathcal{C}_1 \subseteq L_1' \cup L_1'' \cup \mathcal{C}_1'$ and $\mathcal{C}_2 \subseteq L_2' \cup L_2'' \cup \mathcal{C}_2'$.

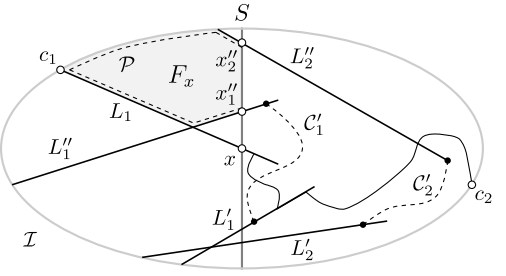

Fig. 31. In F_x, no non-illuminated subsegment of a left segment in $\tilde{\mathcal{L}}(v)$ exists that can be connected along a non-illuminated path to a non-illuminated subsegment of a right segment in $\tilde{\mathcal{L}}(v)$.

We now introduce some notation (see Fig. 30 for an illustration). For $i \in \{1, 2\}$, let $\mathcal{C}_i'$ be a free curve in F_2 that connects L_i' and L_i'' and let $\mathcal{C}_i \subseteq L_i' \cup \mathcal{C}_i' \cup L_i''$ be the simple curve with the endpoints p_i' and p_i'', where p_i' (resp. p_i'') is the endpoint of L_i' (reps. L_i'') on ∂F_1. Note that p_1', p_1'', p_2' and p_2'' may not be distinct. Then, no point in $\mathcal{C}_1 \cap \mathcal{C}_2$ lies in F_2 (since otherwise, we obtain a contradiction with Claim 3.12). In particular, the segments in $\{L_1', L_1'', L_2', L_2''\}$ are distinct, and if either L_1' and L_2' or L_1'' and L_2'' intersect, then their intersections are not in F_2 (but can lie on the splitter S).

Thus, keeping in mind also the minimality of $\mathcal{C}$, we observe:

- The y-coordinate of the intersection $x_1' = L_1' \cap S$ is not smaller than the y-coordinate of the intersection $x_1'' = L_2' \cap S$.
- The y-coordinate of the intersection $x_1'' = L_1'' \cap S$ is not greater than the y-coordinate of the intersection $x_2'' = L_2'' \cap S$.

Next, we have the following claim.

Claim 3.13. *There is no right segment in $\tilde{\mathcal{L}}(v)$ intersecting $\mathcal{C}_2$.*

Sketch of proof. The proof follows by arguments similar to those in the proof of Claim 3.8. Namely, if such a segment R existed, we could construct a "bypass" (curve) with a minimum separator of size at most three (contributed by segments in $\{L_2', L_2'', R\}$), which together with segments in $\mathcal{X}_1^{k-4}$ would imply the existence of a simple (S, x)-curve of weight at most $k-1$ that intersects S exactly once. This contradicts the minimality of $\mathcal{C}$. We omit details. □

For the (S, x)-separator $\tilde{\mathcal{S}}$ of $\tilde{\mathcal{C}}$ constructed above, define its (not necessarily minimal) separator

$$\mathcal{S}^\circ = \tilde{\mathcal{S}} \cup \{L_1', L_1'', L_2', L_2''\}.$$

Observe that $|\mathcal{S}^\circ| \leq 15$. Let F_x be the subface of F bounded by $\mathcal{P}$ and $\overline{x_1''x_2''}$ (Fig. 31). Consider now any subarrangement $\tilde{\mathcal{A}}_x = (\tilde{\mathcal{L}}_x, V, \mathcal{I} \cup r(\mathcal{S}^\circ))$ that has a non-illuminated segment $L \in \tilde{\mathcal{L}}_x$ such that $L \cap F_x \neq \emptyset$. We have the following claim.

Claim 3.14. *No segment in $\tilde{\mathcal{L}}_x$ is contained in a right segment in $\tilde{\mathcal{L}}(v)$.*

Proof. Suppose on the contrary that there is in $\tilde{\mathcal{L}}_x$ a non-illuminated subsegment R of a right segment in $\tilde{\mathcal{L}}(v)$. Then, R intersects in F_2 either L_2' or $\mathcal{C}_2'$. Due to Claims 3.8 and 3.13, none of those situations is possible, which gives a required contradiction. □

Note that Claim 3.14 implies that $\mathcal{S}^\circ$ is safe for $\tilde{\mathcal{C}}$, which completes the proof of Lemma 3.6. □

3.2.5. Searching strategy for nice subarrangements

Let $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ be a nice subarrangement of an arrangement $\mathcal{A} = (\mathcal{L}, V)$, let $v \in V$ be selected arbitrarily and let S be a balanced splitter in $\tilde{\mathcal{A}}$ with respect to v, whose existence is guaranteed by Lemma 3.5. Recall that for a set of line segments $X \subseteq \tilde{\mathcal{L}}$, $r(X)$ is the (minimum) set of line segments in $\mathcal{L}$ whose illumination ensures that all line segments in X are illuminated. Without loss of generality assume that S is vertical, with the bottom endpoint denoted by b. Let $X = \{x_1, \ldots, x_m\}$ be the points that S has in common with segments in $\mathcal{X}(S)$; the points in X are ordered with respect to the increasing y-coordinate. Next, let $\mathcal{C}_i$, $i \in \{1, \ldots, m\}$, be an (S, x_i)-curve having a safe (S, x_i)-separator $\mathcal{S}_i^\circ = \{L_1^i, \ldots, L_{k_i}^i\}$ of size $O(1)$; such $\mathcal{C}_i$ exists by Lemma 3.6. Let F_i, $i \in \{1, \ldots, m\}$, denote the (bounded) face of the planar subdivision formed by segments in $\mathcal{I}$ and the curve $\mathcal{C}_i$ such that $\overline{bx_i} \subset F_i$ (condition (b) for $\mathcal{C}_i$). Denote by x_{m+1} and F_{m+1} the top endpoint of S and the face of the planar subdivision formed by $\mathcal{I}$ such that $\overline{bx_{m+1}} \subset F_{m+1}$, respectively.

Informally speaking, we construct a searching strategy that illuminates separators $\mathcal{S}_i^\circ$ one by one, keeping always the current one and the previous one illuminated, and making recursive calls to search subarrangements 'between' the two separators. We now describe the first few steps of this searching process formally, and then we provide a pseudocode for constructing our searching strategy, together with its correctness proof.

Consider any subarrangement $\tilde{\mathcal{A}}_1 = (\tilde{\mathcal{L}}_1, V, \mathcal{I} \cup r(\tilde{\mathcal{L}}(S)) \cup r(\mathcal{S}_1^\circ))$ whose non-illuminated segments in $\tilde{\mathcal{L}}_1$ lie within the face F_1; we emphasize that together with line segments in $\mathcal{I}$, we illuminate at most two elements in $r(\tilde{\mathcal{L}}(S))$ and all $O(1)$ line segments in $r(\mathcal{S}_1^\circ)$. The boundary $\mathcal{I} \cup \tilde{\mathcal{L}}(S) \cup r(\mathcal{S}_1^\circ)$ satisfies the nice properties (1–3). We apply a recursive call to $\tilde{\mathcal{A}}_1$, which results in clearing $\tilde{\mathcal{A}}_1$. Then, we keep $r(\mathcal{S}_1^\circ)$ illuminated (but not $r(\tilde{\mathcal{L}}(S))$).

Next, consider any subarrangement $\tilde{\mathcal{A}}_2 = (\tilde{\mathcal{L}}_2, V, \mathcal{I} \cup r(\tilde{\mathcal{L}}(S)) \cup r(\mathcal{S}_1^\circ) \cup r(\mathcal{S}_2^\circ))$ whose non-illuminated segments in $\tilde{\mathcal{L}}_2$ lie within F_2. Together with the segments in $\mathcal{I} \cup r(\tilde{\mathcal{L}}(S)) \cup r(\mathcal{S}_1^\circ)$, we now illuminate the segments in $r(\mathcal{S}_2^\circ)$, and $\mathcal{I} \cup r(\mathcal{S}_1^\circ) \cup r(\mathcal{S}_2^\circ)$ maintains the nice properties (1-3). Since $\mathcal{S}_1^\circ$ and $\mathcal{S}_2^\circ$ are the two separators whose curves $\mathcal{C}_1$ and $\mathcal{C}_2$ pass through the two consecutive points x_1 and x_2 along our splitter S, respectively:

- If the elements in $\tilde{\mathcal{L}}_2$ lie within $F_1 \cap F_2$, $\tilde{\mathcal{A}}_2$ is already clear because $\mathcal{I} \cup r(\mathcal{S}_1^\circ)$ is kept illuminated.
- Otherwise, that is if the elements in $\tilde{\mathcal{L}}_2$ lie within $F_2 \setminus F_1$, we apply a recursive call to $\tilde{\mathcal{A}}_2$.

Consequently, $\tilde{\mathcal{A}}_2$ becomes clear, and we turn off the searchlights illuminating $r(\mathcal{S}_1^\circ)$, keeping only the segments in $r(\tilde{\mathcal{L}}(S)) \cup r(\mathcal{S}_2^\circ)$ illuminated (together with $\mathcal{I}$), to keep $\tilde{\mathcal{A}}_2$ (and any other subarrangement within F_2) clear for the next step.

We apply the same argument to any subarrangement $\tilde{\mathcal{A}}_i = (\tilde{\mathcal{L}}_i, V, \mathcal{I} \cup r(\tilde{\mathcal{L}}(S)) \cup r(\mathcal{S}_{i-1}^\circ) \cup r(\mathcal{S}_i^\circ))$, whose non-illuminated segments in $\tilde{\mathcal{L}}_i$ lie within F_i, $i = 3, \ldots, m+1$. Note that the segments within $F \setminus F_m$ become clear when $i = m+1$, which eventually results in clearing the whole $\tilde{\mathcal{A}}$.

The pseudocode of algorithm SNS (*Search Nice Subarrangement*) is given below. The algorithm takes a nice subarrangement $\hat{\mathcal{A}} = (\hat{\mathcal{L}}, V, \hat{\mathcal{I}})$ and $v \in V$ as input. In the first step, it is checked if $\hat{\mathcal{A}}$ is already cleared, and if this is the case, then the computation stops. Otherwise, it may be the case that $\hat{\mathcal{A}}$ is not cleared but all line segments of the subarrangement $\hat{\mathcal{A}}$ contained in $\hat{\mathcal{L}}(v)$ are already cleared: then another guard in V is selected as v. Step 2 computes S, the points $x_1, \ldots, x_{m+1}$ with the corresponding safe separators.[16] The main part of SNS, i.e., Step 3, performs the clearing as described above.

Algorithm SNS($\hat{\mathcal{A}}, v$)

Step 1: If there are no contaminated line segments in $\hat{\mathcal{L}}$, then **exit** (the subarrangement $\hat{\mathcal{A}}$ is clear). If there are no contaminated line segments in $\hat{\mathcal{L}}(v)$, then let (a new) $v \in V$ be any guard with at least one contaminated line segment in $\hat{\mathcal{L}}(v)$.

Step 2: Let S be a balanced splitter of $\hat{\mathcal{A}}$ with respect to v. Determine the points $x_1, \ldots, x_{m+1}$, a safe (S, x_i)-separator $\mathcal{S}_i^\circ$ and the face F_i for each $i \in \{1, \ldots, m+1\}$, where $\mathcal{S}_{m+1}^\circ = \emptyset$. Let $F_0 = \emptyset$ and $\mathcal{S}_0^\circ = \emptyset$.

Step 3: For each $i := 1, \ldots, m+1$ (in this order) do:

Step 3a: Stop illuminating any line segments except for those in $\hat{\mathcal{I}} \cup r(\hat{\mathcal{L}}(S)) \cup r(\mathcal{S}_{i-1}^\circ)$.

Step 3b: Illuminate additionally the line segments in $r(\mathcal{S}_i^\circ)$ so that the segments in $\hat{\mathcal{I}} \cup r(\hat{\mathcal{L}}(S)) \cup r(\mathcal{S}_{i-1}^\circ) \cup r(\mathcal{S}_i^\circ)$ are now illuminated.

Step 3c: For each nice subarrangement $\hat{\mathcal{A}}_i$ of $(\hat{\mathcal{L}}, V, \hat{\mathcal{I}} \cup r(\hat{\mathcal{L}}(S)) \cup r(\mathcal{S}_{i-1}^\circ) \cup r(\mathcal{S}_i^\circ))$ within the face $F_i \setminus F_{i-1}$, call SNS($\hat{\mathcal{A}}_i, v$).

Theorem 3.15. *Any nice (n, g)-subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ can be searched with $O(g^2 \cdot \log n)$ searchlights.*

[16] Note that we use this algorithm to prove our upper bound on the number of searchlight needed for clearing a nice subarrangement and therefore we skip the details of the efficiency of computing S and the separators. We only remark that there are $O(n^2)$ possible splitters S, for each splitter it can be verified in polynomial time whether it is balanced or not and minimal separators can be obtained in polynomial time by a modification of a shortest path algorithm.

Fig. 32. Adding a virtual simple closed polyline $\mathcal{P}$ allows us to treat the unbounded face as a bounded one.

Proof. Given any nice subarrangement $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$, the call to SNS$(\tilde{\mathcal{A}}, v)$, where $v \in V$ is selected arbitrarily, results in clearing $\tilde{\mathcal{A}}$. This follows from a simple induction on the total number of representatives of contaminated line segments in $\tilde{\mathcal{L}}$, as we sketch below. The way the separators are illuminated in Steps 3a and 3b of Algorithm SNS gives two observations. First, if a face F_{i-1}, $i \in \{2, \ldots, m+1\}$, contains no contaminated subsegment at the beginning of the i-th iteration of the main loop in Step 3, then F_{i-1} has no contaminated subsegment at the end of Step 3b in that iteration. Second, at the end of Step 3c of this iteration, there is no contaminated subsegment in the face $F_i \setminus F_{i-1}$ (due to the induction hypothesis). Thus, at the end of the $(m+1)$-st iteration, F_{m+1} has no contaminated subsegment, which implies that the input nice subarrangement is clear. (Observe that each recursive call SNS$(\tilde{\mathcal{A}}_i, v)$ results in decreasing the number of non-illuminated representatives since S is balanced; see the discussion below.)

We now bound the number of searchlight that SNS uses for $\tilde{\mathcal{A}}$. The analysis is divided into two parts: first we bound the recursion depth, and then we bound the number of searchlights used by each instance of SNS.

Note that for a given choice of $v \in V$ several (possibly zero) subsequent recursive calls to SNS take v as part of the input. Then, if $\tilde{\mathcal{L}}(v)$ has no contaminated line segments, then another guard in V is selected and searching of the current nice subarrangement continues. Thus, for bounding the depth of the recursion, it is enough to argue that for any choice of v, after $O(\log n)$ recursive calls the input nice subarrangement has no contaminated line segments in $\tilde{\mathcal{L}}(v)$. This, however, is due to the two following facts. First, in each call to SNS a balanced splitter S is selected: such a splitter exists by Lemma 3.5. Second, any subarrangement $\tilde{\mathcal{A}}_i$, $i \in \{1, \ldots, m+1\}$, in Step 3c can have either a non-illuminated subsegment of a left segment in $\tilde{\mathcal{L}}(v)$ or a non-illuminated subsegment of a right segment in $\tilde{\mathcal{L}}(v)$, but not both. This is due to the fact that each separator S_i° is safe; such separator exists by Lemma 3.6.

By Lemma 3.6, illuminating the separator $\mathcal{S}_i^\circ$, $i \in \{1, \ldots, m\}$, requires $O(1)$ searchlights, and the procedure illuminates at most two separators at a time. The maximum depth of recursion of SNS is $O(\log n)$ per guard and hence its total recursion depth is $O(g \log n)$. Moreover, $O(g \log n)$ searchlights will be used simultaneously (by all guards, and hence by any single guard) at a deepest level of recursion. Therefore, placing $O(g \log n)$ searchlights at each guard is sufficient and hence the total number of searchlights used by all guards in SNS is $O(g^2 \log n)$. Finally, by Lemma 3.4, $|\mathcal{I}| \leq 2g - 1$. □

3.2.6. Subarrangements within the unbounded face

We have assumed that non-illuminated segments of $\tilde{\mathcal{A}} = (\tilde{\mathcal{L}}, V, \mathcal{I})$ are within a bounded face of the planar subdivision formed by line segments in $\mathcal{I}$. If non-illuminated segments are within the unbounded face F of this subdivision, then all we need is to add a virtual simple closed polyline $\mathcal{P} \subset F$ to $\mathcal{I}$ such that $\mathcal{P}$ encloses $\mathcal{I}$ and shares a vertex only with one line segment endpoint or one intersection of segments in $\mathcal{I}$. See Fig. 32. Then essentially the same argument as above applies to the section of the unbounded face enclosed by $\mathcal{P}$. (The details are omitted.) Consequently, we obtain the following theorem.

Theorem 3.16. *Any (n, g)-arrangement of line segments can be searched with $O(g^2 \cdot \log n)$ searchlights.*

Proof. Let $\mathcal{A} = (\mathcal{L}, V)$ be an (n, g)-arrangement of line segments. By permanently illuminating a set $\mathcal{I} \subseteq \mathcal{L}$ of segments using at most $O(g)$ searchlights (Lemma 3.4), we first divide $\mathcal{A}$ into a number of nice subarrangements. Next, by Theorem 3.15, we search each of the nice subarrangements, one by one, using $O(g^2 \log n)$ searchlights. □

Corollary 3.17. *For arrangements of line segments, we have $s(n, g) = O(g^2 \cdot \log n)$.*

3.3. Upper bounds in terms of guard degrees

The argument leading to the upper bound on $s(\mathcal{A})$ given in Theorem 2.4 for (n, g)-arrangements $\mathcal{A} = (\mathcal{L}, V)$ of *lines* carries over to the case of (n, g)-arrangements of *line segments* without any change. Therefore we have the following corollaries, where $\Gamma(\mathcal{A})$ is the maximum number of guards in V that lie on a set of parallel line segments in $\mathcal{L}$, $\deg_{\mathcal{A}}(v)$ is the number of maximal subsegments of line segments in $\mathcal{L}$ ending at v, and $\Delta(\mathcal{A}) = \max_{v \in V} \deg_{\mathcal{A}}(v)$.

Corollary 3.18. *For any (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ of line segments, we have*

$$s(\mathcal{A}) \leq \min\left\{\frac{1}{2}\sum_{v \in V} \deg_{\mathcal{A}}(v) - \Gamma(\mathcal{A}), n - 1\right\}.$$

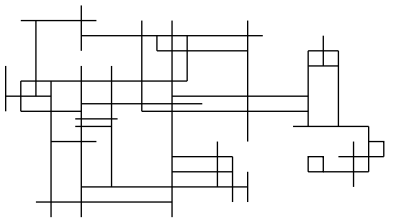

Fig. 33. An example of a grid.

Corollary 3.19. *Let $\mathcal{A}$ be an (n, g)-arrangement of line segments with $\Delta(\mathcal{A}) \leq 4$. Then*

$$s(\mathcal{A}) \leq \min\{2g, n\} - 1.$$

Additionally, if all guards belong to some lines that are all parallel, then $s(\mathcal{A}) \leq \min\{g, n-1\}$.

In particular, the above corollary can be applied for the case of grids. Recall — a *grid* $\mathcal{G}$ is a connected arrangement of axis-aligned line segments, see Fig. 33 for an example. The problem of guarding a grid was introduced by Ntafos in 1986, who proved that the minimum guard set for a given n-segment grid can be found in $O(n^{2.5})$ time [17]. For any grid $\mathcal{G} = (\mathcal{L}, V)$, we have $\Delta(\mathcal{G}) \leq 4$ and $|\Gamma(\mathcal{G})| = g$, which results in the following corollary.

Corollary 3.20. *For grids, we have $s(n, g) \leq \min\{g, n-1\}$.*

In particular, for a given (n, g)-grid $\mathcal{G} = (\mathcal{L}, V)$, if V is minimal, then each guard in V requires at least one searchlight to search $\mathcal{G}$, which yields $s(n, g) = g$ for the case of grids (see Fig. 7; recall that for grids we have $\lfloor \frac{n}{2} \rfloor \leq g$).

4. Concluding remarks

We conclude with a few open problems.

Problem 1. Provide better estimates on $s(n, g)$ in the case of (n, g)-arrangements of lines. In particular, prove or disprove that $s(n, g) \leq 2g$.

Problem 2. Provide better estimates on $s(n, g)$ in the general case of (n, g)-arrangements of line segments. Without any strong evidence, we conjecture that the upper bound on $s(n, g)$ can be improved up to $O(g \log \Delta)$, which will then match the lower bound proved in Subsection 3.1; in other words, we expect that the worst-case total number of searchlights necessary to successfully search any (n, g)-arrangement strictly depends on the degree $\Delta(\mathcal{A})$ of an arrangement $\mathcal{A} = (\mathcal{L}, V)$ to be searched.

Problem 3. The function $s(n, g)$ describes the total number of searchlights required by g guards. It is natural then to ask what is the maximum number $\dot{s}(n, g)$ of searchlights ever needed per some guard to search any (n, g)-arrangement of line segments. Notice that the proofs of Theorems 3.2 and 3.16 imply that there are infinitely many arbitrarily large arrangements $\mathcal{A}$ that require $\Omega(\log \Delta(\mathcal{A}))$ searchlights per a guard and $\dot{s}(n, g) = O(g \log n)$, respectively. Keeping in mind the above conjecture for Problem 2, we expect that to search an (n, g)-arrangement $\mathcal{A} = (\mathcal{L}, V)$ of line segments, $O(\log \Delta(\mathcal{A}))$ searchlights are needed per a guard in V.

Define the *switch number* as the minimum number of steps needed to search an arrangement using a given placement of searchlights. The discussions leading to the proof of Theorem 2.4 and Corollary 3.18 show that an (n, g)-grid $\mathcal{G} = (\mathcal{L}, V)$ of lines or line segments can be searched in at most $3g + 1$ steps using one searchlight at every guard (all searchlights are aimed toward the left in step 1, and thereafter the searchlights, one by one, change directions at most three times). Focusing only on the case of searching a grid of *line segments* using exactly *one* searchlight at every guard, we denote by $sn(\mathcal{G})$ the search number of grid $\mathcal{G} = (\mathcal{L}, V)$, and by $sn(n, g)$ the maximum of $sn(\mathcal{G})$ over all (n, g)-grids. We then propose:

Problem 4. Given an (n, g)-grid $\mathcal{G} = (\mathcal{L}, V)$ of line segments, determine $sn(\mathcal{G})$. We expect this problem to be NP-hard, and pose the problem of identifying non-trivial subclasses of grids for which the problem is solvable in polynomial time.

Problem 5. Obtain tight bounds on $sn(n, g)$.

In the rest of this section we present the following initial result on Problem 5.

Theorem 4.1.

1. *For any $1 \leq g \leq n-1$, $sn(n, g) \leq 3g + 1$.*

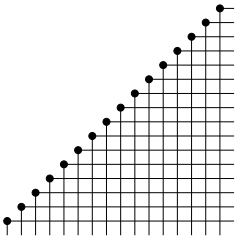

Fig. 34. Grid $\mathcal{T}_g$ for the case $k = 5$, $n = 32$ and $g = 16$, where the black dots show the guard positions. $\mathcal{T}_g$ has a switch number $sn(\mathcal{T}_g) \geq \log_2 g$.

2. *For every* $k = 1, 2, \ldots$, $n = 2^k$ *and* $g = n/2$, $sn(n, g) \geq \log_2 g$.

Proof. The first claim holds because, as discussed above, $sn(\mathcal{G}) \leq 3g + 1$ for any (n, g)-grid of line segments. To prove the second claim, we show by induction that for any $k = 2, 3, \ldots$ the triangular (n, g)-grid $\mathcal{T}_g$ consisting of $n = 2^k$ line segments and $g = n/2$ guards depicted in Fig. 34 satisfies $sn(\mathcal{T}_g) \geq \log_2 g$. (The case $k = 1$, $n = 2$ and $g = 1$ is trivial.) Since every intersection in $\mathcal{T}_g$ has to be illuminated by two searchlights (one aimed down and one aimed right) simultaneously at least once during a search (we say an intersection is *checked* when this happens), it suffices to show that it takes at least $\log_2 g$ steps to check all intersections of $\mathcal{T}_g$. We now show $\mathrm{OPT}(g) \geq \log_2 g$, $g = 2^{k-1}$, by induction on k, where $\mathrm{OPT}(g)$ denotes the minimum number of steps sufficient to check all intersections in $\mathcal{T}_g$.

Obviously $\mathrm{OPT}(g) \geq \log_2 g$ holds for $k = 2$ ($n = 4$ and $g = 2$). Now, consider $\mathcal{T}_g$ for some $k \geq 3$, and assume $\mathrm{OPT}(g/2) \geq \log_2(g/2) = \log_2 g - 1$. Without loss of generality we may assume that all g searchlights are on at every step while all intersections in $\mathcal{T}_g$ are checked in optimal $\mathrm{OPT}(g)$ steps. Thus at every step at least $g/2$ searchlights are either aimed right or aimed down. Fix a step t, and let $\mathcal{T}_{g/2}$ be any sub-grid of $\mathcal{T}_g$ consisting only of $g/2$ guards whose searchlights are aimed right (or aimed down) together with the g line segments incident on them. Since (i) none of the intersections in $\mathcal{T}_{g/2}$ are checked in step t, and (ii) all intersections in $\mathcal{T}_{g/2}$ must be checked in $\mathrm{OPT}(g)$ steps (when all intersections in $\mathcal{T}_g$ are checked in $\mathrm{OPT}(g)$ steps), we have $\mathrm{OPT}(g) \geq \mathrm{OPT}(g/2) + 1 \geq \log_2 g$, where the second inequality is by the inductive hypothesis. □

Acknowledgement

We thank Giovanni Viglietta for providing constructive comments and helpful suggestions in improving the contents of this paper. Paweł Żyliński wishes to express his gratitude to ks. Waldemar Waluk for discussion, mostly not on the topic.

References

[1] P. Bose, J. Cardinal, S. Collette, F. Hurtado, M. Korman, S. Langerman, P. Taslakian, Coloring and guarding arrangements, Discrete Math. Theor. Comput. Sci. 15 (3) (2013) 139–154.
[2] V.E. Brimkov, A. Leach, M. Mastroianni, J. Wu, Guarding a set of line segments in the plane, Theoret. Comput. Sci. 412 (15) (2011) 1313–1324.
[3] B. Brodén, M. Hammar, B.J. Nilsson, Guarding lines and 2-link polygons is APX-hard, in: Proceedings of the 13th Canadian Conference on Computational Geometry, CCCG, 2001, pp. 45–48.
[4] R.W. Dawes, Some pursuit–evasion problems on grids, Inform. Process. Lett. 43 (5) (1992) 241–247.
[5] A. Dumitrescu, H. Kok, I. Suzuki, P. Żyliński, Vision-based pursuit–evasion in a grid, SIAM J. Discrete Math. 24 (3) (2010) 1177–1204.
[6] A. Dumitrescu, J.S.B. Mitchell, P. Żyliński, Watchman routes for lines and segments, Comput. Geom. 47 (4) (2014) 527–538.
[7] A. Dumitrescu, J.S.B. Mitchell, P. Żyliński, The minimum guarding tree problem, Discrete Math. Algorithms Appl. 6 (1) (2014) #1450011.
[8] F.V. Fomin, D.M. Thilikos, An annotated bibliography on guaranteed graph searching, Theoret. Comput. Sci. 399 (3) (2008) 236–245.
[9] L.P. Gewali, S. Ntafos, Covering grids and orthogonal polygons with periscope guards, Comput. Geom. 2 (6) (1993) 309–334.
[10] T. Kameda, I. Suzuki, Z.J. Zhang, Finding the minimum-distance schedule for a boundary searcher with a flashlight, in: Proceedings of the 9th Latin American Symposium on Theoretical Informatics, LATIN 2010, in: Lecture Notes in Computer Science, vol. 6034, 2010, pp. 84–95.
[11] T. Kameda, M. Yamashita, I. Suzuki, On-line polygon search by a seven-state boundary 1-searcher, IEEE Trans. Robot. 22 (3) (2006) 446–460.
[12] T. Kameda, Z. Zhang, M. Yamashita, Simple characterization of polygons searchable by 1-searcher, in: Proceedings of the 18th Canadian Conference on Computational Geometry, CCCG, 2006, pp. 113–116.
[13] L.M. Kirousis, C.H. Papadimitriou, Searching and pebbling, Theoret. Comput. Sci. 47 (3) (1986) 205–218.
[14] A. Kosowski, M. Małafiejski, P. Żyliński, Cooperative mobile guards in grids, Comput. Geom. 37 (2) (2007) 59–71.
[15] N. Megiddo, S.L. Hakimi, M.R. Garey, D.S. Johnson, C.H. Papadimitriou, The complexity of searching a graph, J. ACM 35 (1) (1988) 18–44.
[16] S.W. Neufeld, A pursuit–evasion problem on a grid, Inform. Process. Lett. 58 (1) (1996) 5–9.
[17] S. Ntafos, On gallery watchman in grids, Inform. Process. Lett. 23 (2) (1986) 99–102.
[18] K.J. Obermeyer, A. Ganguli, F. Bullo, A complete algorithm for searchlight scheduling, Internat. J. Comput. Geom. Appl. 21 (1) (2011) 101–130.
[19] J. O'Rourke, Art Gallery Theorems and Algorithms, Oxford University Press, New York, 1987.
[20] T.D. Parsons, Pursuit–evasion in a graph, in: Theory and Applications of Graphs, in: Lecture Notes in Mathematics, vol. 642, 1978, pp. 426–441.
[21] T. Shermer, Recent results in art galleries, Proc. I.E.E.E. 80 (9) (1992) 1384–1399.
[22] B.H. Simov, G. Slutzki, S.M. Lavalle, Clearing a polygon with two 1-searchers, Internat. J. Comput. Geom. Appl. 19 (1) (2009) 59–92.
[23] K. Sugihara, I. Suzuki, On a pursuit–evasion problem related to motion coordination of mobile robots, in: Proceedings of the 21st Annual Hawaii International Conference on System Sciences, 1988, pp. 218–226.
[24] K. Sugihara, I. Suzuki, Optimal algorithms for a pursuit–evasion problem in grids, SIAM J. Discrete Math. 2 (1) (1989) 126–143.

[25] K. Sugihara, I. Suzuki, M. Yamashita, The searchlight scheduling problem, SIAM J. Comput. 19 (6) (1990) 1024–1040.
[26] I. Suzuki, M. Yamashita, Searching for a mobile intruder in a polygonal region, SIAM J. Comput. 21 (5) (1992) 863–888.
[27] I. Suzuki, M. Yamashita, H. Umemoto, T. Kameda, Bushiness and a tight worst-case upper bound on the search number of a simple polygon, Inform. Process. Lett. 66 (1) (1998) 49–52.
[28] I. Suzuki, P. Żyliński, Capturing an evader in a building — randomized and deterministic algorithms for mobile robots, IEEE Robotics & Automation Magazine 15 (2) (2008) 16–26.
[29] C.D. Tóth, Illumination in the presence of opaque line segments in the plane, Comput. Geom. 21 (3) (2002) 193–204.
[30] C.D. Tóth, Illuminating disjoint line segments in the plane, Discrete Comput. Geom. 30 (3) (2003) 489–505.
[31] J. Urrutia, Art gallery and illumination problems, in: Handbook of Computational Geometry, Elsevier, 2000, pp. 973–1027.
[32] G. Viglietta, Searching polyhedra by rotating planes, Internat. J. Comput. Geom. Appl. 22 (3) (2012) 243–275.
[33] G. Viglietta, Partial searchlight scheduling is strongly PSPACE-complete, in: Proceedings of the 25th Canadian Conference on Computational Geometry, CCCG, 2013, pp. 55–60.
[34] G. Viglietta, M. Monge, The 3-dimensional searchlight scheduling problem, in: Proceedings of the 22nd Canadian Conference on Computational Geometry, CCCG, 2010, pp. 9–12.
[35] D.B. West, Introduction to Graph Theory, Prentice Hall, 2001.
[36] N. Xu, P. Brass, On the complexity of guarding problems on orthogonal arrangements, in: Proceedings of the 20th Annual Fall Workshop on Computational Geometry, FWCG, 2010, #33.
[37] M. Yamashita, I. Suzuki, T. Kameda, Searching a polygonal region by a group of stationary k-searchers, Inform. Process. Lett. 92 (1) (2004) 1–8.
[38] W.C.K. Yen, C.Y. Tang, An optimal algorithm for solving the searchlight guarding problem on weighted trees, Inform. Sci. 87 (1–3) (1995) 79–105.
[39] W.C.K. Yen, C.Y. Tang, An optimal algorithm for solving the searchlight guarding problem on weighted interval graphs, Inform. Sci. 100 (1–4) (1997) 1–25.
[40] W.C.K. Yen, C.Y. Tang, An optimal algorithm for solving the searchlight guarding problem on weighted two-terminal series-parallel graphs, Acta Inform. 36 (2) (1999) 143–172.