RC25528 (WAT1503-050) March 16, 2015
Computer Science

IBM Research Report

Line Segment Visibility with Sidedness Constraints: Theory and Practice

Jonathan Lenchner
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

Eli Packer
IBM Research Division
Haifa Research Laboratory
Mt. Carmel 31905
Haifa, Israel

Line Segment Visibility with Sidedness Constraints: Theory and Practice

Jonathan Lenchner *   Eli Packer †

Abstract

We study a family of line segment visibility problems, related to classical art gallery problems, which are motivated by monitoring requirements in commercial data centers. Given a collection of non-overlapping line segments in the interior of a polygon and a requirement to monitor the segments from one side or the other we examine the problem of finding a minimal guard set. We consider combinatorial bounds of problem variants where the problem solver gets to decide which side of the segments to guard, the problem poser gets to decide which side to guard, and many others. We show that virtually all variants are NP-hard to solve exactly, but also provide heuristics and experimental results using data from large commercial data centers to give insight into the associated practical problems.

1 Introduction

We study a family of line segment visibility problems, related to classical art gallery problems, which are motivated by monitoring and surveillance requirements in commercial data centers. In traditional art gallery problems (see [15], [21], [23] and [26]) an entire polygonal region must be kept under surveillance. In our case it is a prescribed collection of non-overlapping line segments in the interior of the polygon that must be kept under surveillance, and moreover, typically it is important just to see one side of each segment. Czyzowicz et al. [4] and Toth [24] studied a similar problem to ours, without the presence of a boundary, and where line segment visibility could be from either side – a variant of our problem family that we call the "solver's choice" problem. Toth [24] also studied the problem where lone segments had to be viewed from *both* sides.

We model a data center as a two-dimensional polygonal enclosure where the objects of principal interest from a monitoring perspective are the air intake regions of servers. Servers are typically stored in racks and the air intake regions of the servers are aligned in rows facing in towards the source of cool air which is vented up into a raised floor area through perforated tiles. One technique for monitoring the air intake temperatures of servers, and the technique of interest in the present investigation, is to use statically placed thermal imaging (infrared) cameras. Thermal imaging cameras, however, are relatively expensive and so the challenge is to deploy as few of these cameras as possible while still viewing, in total, all of the air intake regions.

The first data center we studied, a relatively small research data center located in Southbury, Connecticut, is depicted in Figure 1. This data center is especially simple since all air intake regions are vertically aligned. Our visibility problem then reduces to one of finding a minimum set of guards that see all sides of a specified set of segments, where visibility is from a specified side of each segment. Data centers with this property are the easiest to study and have especially simple combinatorial properties. After reviewing related work, we consider cases of this sort, where all line segments in our model are vertically aligned.

A second much larger industrial data center, located in Poughkeepsie, New York, is depicted in Figure 2. This data center has air intakes located in both vertical and horizontal orientations, which is fairly typical. More generally, the air intakes of machines needing monitoring can be oriented arbitrarily. After discussing the case of all vertical segments, for which tight combinatorial bounds are easily found, we consider these latter cases in greater detail. Preliminary results from this work have previously been reported in [5] and [17].

*IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532, lenchner@us.ibm.com

†IBM Haifa Research Lab, Mount Carmel, Haifa, Israel 31905. elip@il.ibm.com.

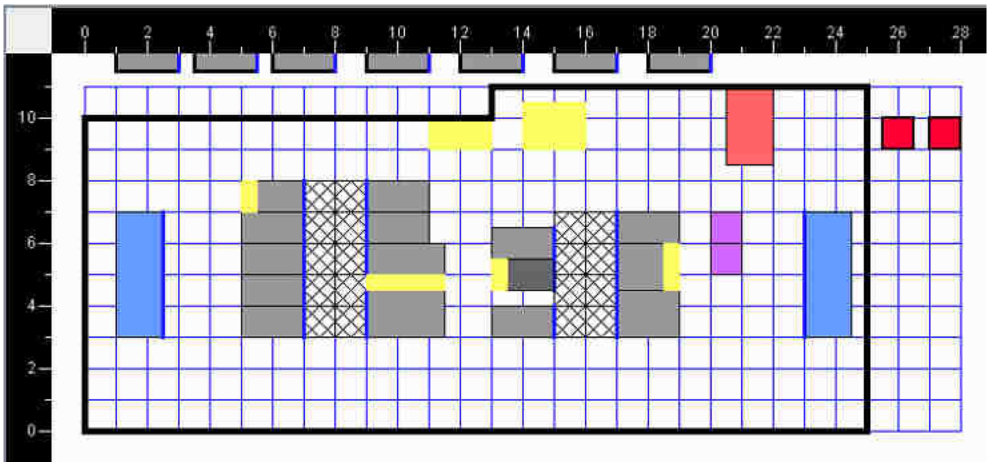

Figure 1. A small research data center in Southbury, Connecticut. The thick black line segments depict the walls (six in number, including a very short vertical segment) of the main room of the data center. Racks of servers are depicted in light grey. The air inlet sides of the servers are indicated with blue line segments. Perforated tiles venting cool air into the raised floor area are depicted using cross-hatched squares. The two air conditioning units, or CRAC (Computer Room Air Conditioning) units in this facility are shown in light blue. The side/direction to which they are venting air is indicated with dark blue segments. Rectangles in the additional colors indicate different sorts of data center equipment, not of direct concern to us in this study.

1.1 Structure of the paper

In the next section we survey related work. In Section 3 we discuss combinaturial bounds for the models defined in this work. In Section 4 we prove that the problems we are studying are all NP-hard. We then present heuristics to solve them in Section 5, and describe experiments we performed along with their implementations in Section 6. We wrap up with conclusions and suggestions for further study in Section 7.

2 Related Work

Our work is closely related to the classical *art gallery* problem – the problems we define and study in this paper can be viewed as its variants.

The art gallery problem was first studied in 1973 when Victor Klee posed the question of how many guards are sufficient to guard the interior of a simple polygon. One of the earliest results was that $\lfloor \frac{n}{3} \rfloor$ guards are always sufficient, and sometimes necessary, to guard a simple n-sided polygon [3, 10]. It is easy to compute a solution with this number of guards by triangulating the polygon, coloring the resulting vertices with three colors, and then placing guards on the vertices with the least used color. However, such a solution is usually far from optimal. In 1986 Lee and Lin proved that finding the minimum number of guards for an arbitrary simple polygon is NP-hard [15]. Over the years numerous variants of the art gallery problem have been presented and studied. Common features that have been altered in different publications are the location of the guards (arbitrary, edge or vertex), their mobility (static or mobile), guarding requirements (guarding the interior of the domain, the boundary of the domain, or specific elements inside the domain), the shape of the polygon (orthogonal, non-orthogonal, with or without holes) and many more. Most variants have been shown to be NP-hard with the exception of some simplified models.

It has become evident over time that even approximating the solution of many of the art gallery variants is a difficult task. Except for the trivial $\lfloor \frac{n}{3} \rfloor$ approximation for the classical problem, researchers have struggled with devising efficient approximations. An interesting result, however, for the classical Klee variant of the problem is the Masters thesis work of Sarma [22], which presented a pseudopolynomial time algorithm that achieves an $O(\log^2 n)$-approximation factor. In subsequent work [7] Sarma and colleagues improved the approximation to $O(\log n)$. As far we know, these algorithms have not been implemented and tested. Approximation algorithms with logarithmic approximation ratios are known for somewhat restricted versions of the problem, e.g., requiring guards to be placed at vertices, or at points of a discrete grid [9, 11, 12]. Constant-factor approximations are known for restricted versions, such as for guarding $1.5D$ terrains [2], and exact methods are known for the special case of rectangle visibility in rectilinear polygons [9]. For further reading on the art gallery problems, we refer the reader to surveys on the art gallery problem [23, 26] and to the book of O'Rourke [21].

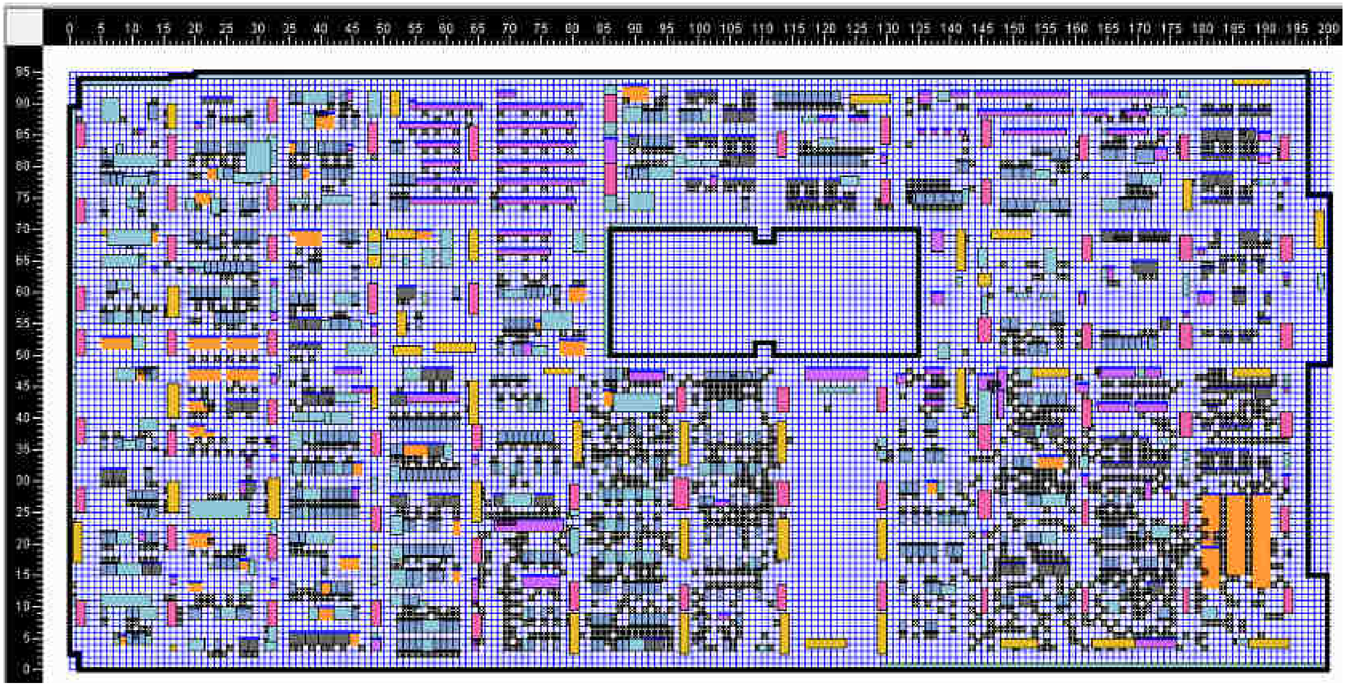

Figure 2. A 72,000 square foot industrial data center in Poughkeepsie, New York. The data center contains a large number of servers, 72 air conditioning units or CRACs, and many other types of equipment. The inset polygon (not part of the data center) is a control room.

For three decades interest in art gallery problems has mainly focused on theoretical aspects. However, since the experimental work of Amit et al. [28] in 2010, a flood of experimental projects have been reported, most of which provide approaches to the classical problem variant. The aim of this recent body of experimental work has been to find close-to-optimal solutions in reasonable time. We refer the reader to a survey of some of these projects [6].

The present work is motivated by practical issues of thermal monitoring of servers in data centers, and the desire to keep equipment safe while keeping cooling and monitoring costs to a minimum. One approach to understanding temperature distributions in data centers is to deploy a large number of static temperature sensors [1]. Often one then feeds the data obtained into a computational fluid dynamics (CFD) model to help understand how changes in the settings of the center's air conditioning units will affect the temperature distribution. The advantage of such an approach is that the cost of sensors is low, however there can be substantial management costs associated with instrumenting all of the necessary equipment with sensors and tying the sensor data into an appropriate backend system. Moreover, the sensors sometimes get in the way of working with the hardware.

Another approach is to outfit a mobile hand-operated cart with sensors and manually move this cart around the data center to gather temperature measurements, as in the work on the Measurement and Management Technology (MMT) system of Hamman et al. [13, 14] The advantage of this method is that it can quickly be adapted to be used in data centers that have not previously been outfitted with sensors (a process that takes a substantial amount of time). It potentially also has lower management costs. However, this method requires a fair amount of manual labor, which can be expensive. A third approach is to outfit a robot with sensors and have the robot autonomously navigate the center and monitor temperatures, either throughout the data center, or at specific locations [16, 20]. A final approach is the one discussed here, that of deploying fixed thermal imaging cameras. Such an approach is discussed in depth in [18], though not from the perspective of placing a minimum number of cameras, as in the current study.

3 Combinatorial Bounds

We shall consider the cases where the segments to be monitored are either all vertical, all axis-aligned, or alternatively, all arbitrarily aligned. Segments are assumed to be non-overlapping/non-intersecting. Within these cases we identify several variants of the basic visibility problem. Namely, if visibility must be from a given side, but that side is specified by the problem poser, we say the problem is an instance of the **Poser's Choice** problem. If, on the other hand, the solver has the choice of which side to monitor the segment from, we say that it is an instance of the **Solver's Choice** problem. We consider two variants of the Solver's Choice problem, a variant where the solver must monitor the entire segment, but may monitor some points from one side and some points from the other side (the problem studied by Czyzowicz et al. [4] and Toth [24]), and a variant where the solver must monitor the entire segment from one side. A final variant is where the solver must monitor the entire segment from both sides (a variant also considered by Toth [24]).

An additional variation that slices through all these variants is to consider the problem where we are satisfied with guarding all segments except for a length of δ along each segment (or, what turns out to be equivalent, an omission of length δ when summed over all segments) for a fixed arbitrarily small $\delta > 0$. In all of these cases we ask what is the worst-case number of guards needed to see a set of N segments. In what follows we use the terms "guards" and "cameras" interchangeably.

3.1 All Vertical Segments

In this section we obtain tight combinatorial bounds for the case where all segments are vertically aligned. In this and subsequent sections we consider the problem where all segments are contained in a bounding rectangle. As noted earlier, the choice of bounding region has only an $O(1)$ impact on the combinatorial bounds.

Theorem 1 *Given n vertical segments in the plane and a bounding rectangle,*

(a) $\lceil \frac{n}{2} \rceil$ cameras are sufficient, and for each n sometimes necessary, to entirely see all n segments from the left (i.e. from the same prescribed side).

(b) $\lceil \frac{2n}{3} \rceil + 1$ cameras are sufficient, and for each n sometimes necessary, to entirely see all segments from both sides.

(c) $\lceil \frac{n}{3} \rceil$ cameras are sufficient, and for each n sometimes necessary, to entirely see all segments from one side or the other (Solver's choice).

(d) $\lfloor \frac{n}{2} \rfloor + 1$ cameras are sufficient, and for each n sometimes necessary, to entirely see all segments from one side or the other (Poser's choice).

Proof. In all problem instances let ϵ be the closest any segment endpoint comes to the boundary rectangle and in an effort to establish the upper bounds asserted in the theorem (sufficiency conditions), let us extend all segments so each top and bottom endpoint is exactly distance ϵ from the rectangle boundary. Cameras which see the extended segments entirely will surely see the shorter segments entirely (from whatever prescribed sides).

In case (a) we number the segments in left to right order $\ell_1, ..., \ell_k$ (ignoring all segments but the top-most ones if there are segments lying directly above one another) and then position cameras c_j for $j = 1, ..., \lceil \frac{k}{2} \rceil$ above and to the left of each ℓ_i for i odd so that c_j can see ℓ_{2j-1} and ℓ_{2j} (if there is an ℓ_{2j}), and any segments lying below these segments from the left. This establishes that $\lceil \frac{n}{2} \rceil$ cameras are sufficient. To see that $\lceil \frac{n}{2} \rceil$ cameras may also be necessary, take ϵ to be very small (much smaller than the height h and width w of the bounding rectangle) and now group segments, all of height $h - 2\epsilon$ at the same distance from one another, such that any camera which sees two consecutive segments from the left sees only a tiny fraction of the segment next over to the right (if there is such a segment). In such an arrangement, $\lceil \frac{n}{2} \rceil$ cameras are clearly necessary.

In case (b) we again use the left to right segment numbering used in case (a), ignoring all segments but the top-most ones. We shall use a single camera to see the left side of the left-most segment and another camera to see the right side of the right-most camera. Neglecting these segment sides, let us start with the first group of 4 segments (deferring consideration of the case where we have fewer than 4 segments until later). We then place a camera just above and to the left of the second segment so that the camera can entirely see the right side of the first segment, the left side of the second segment and the left side of the third segment, and place a second camera just above and to the right of the third segment so that it entirely sees the left side of the second segment, the right side of the third segment and the left side of the fourth segment. Next place a camera just above and to the left of the fifth segment and another camera above and to the right of the sixth segment. Analogously, place cameras just above and to the left of the $(3i - 1)$st segment and just above and to the right of the $3i$th segment. Figure 3

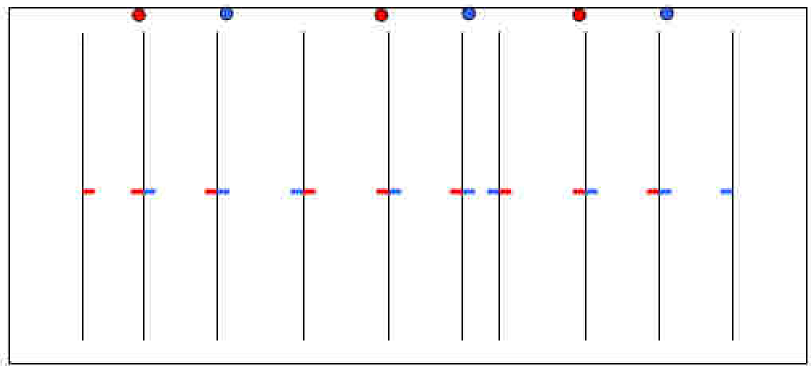

Figure 3. Positioning of cameras to show that $\lceil \frac{2n}{3} \rceil + 1$ cameras can always see both sides of all of n vertical segments. The red-marked cameras see the red-marked sides of segments and the blue-marked cameras seen the blue-marked sides of segments. red cameras are placed just above and to the left of the $(3i-1)$st segment and blue cameras are placed just above and to the right of the $3i$th segment.

illustrates these camera positions and the associated segment-wise coverage by the various cameras. For the case $n \geq 4,\ n \equiv 0 \bmod 3$, since each of $\frac{n}{3}$ red and $\frac{n}{3}$ blue cameras sees 3 independent segment sides, with the exception of the right-most blue camera, which sees just two sides, and all segment sides are thus-wise covered by one camera with the exception of the left-most segment side, which can be covered by placing a camera at the very left of the rectangle we see that we can, in this case, use $\frac{2n}{3} + 1$ cameras to see all segment sides. If $n \equiv 1 \bmod 3$ then $\lceil \frac{2n}{3} \rceil + 1$ affords one more camera then for $n-1 \ \equiv 0 \bmod 3$. If we consider the segments to be added in left to right order then the right-most blue camera already sees the left side of the nth segment so a single additional camera can be added to see the right side of this nth and right-most segment. If $n \equiv 2 \bmod 3$, again $\lceil \frac{2n}{3} \rceil + 1$ affords one more camera then for $n-1 \ \equiv 1 \bmod 3$. Again considering the segments to be placed in left to right order, if we place the second to last camera between the $n-1$st and nth segments, then with our additional camera we just have to guard the rightmost side of the right most camera which is easily handled by placing the camera anywhere to the right of the segment. Hence the cases $n \geq 4$ of arbitrary modularity are handled. A glance at Figure 3 for the cases of $n = 1, 2$ and 3 segments shows that $\lceil \frac{2n}{3} \rceil + 1$ suffices in these cases as well.

To complete case (b), it remains to show that $\lceil \frac{2n}{3} \rceil + 1$ cameras are sometimes necessary for arbitrarily large n. Let us assume our rectangle is axis-aligned and let x_k denote the x-coordinate of a kth segment ℓ_k. Then, for any $\epsilon > 0$ we can place n equally spaced vertical segments (equally spaced in the sense that $x_{i+1} - x_i = x_{j+1} - x_j$ for any $1 \leq i, j \leq n-1$), each extending sufficiently close to the boundary rectangle such that a camera c placed such that $x_i \leq x(c) < x_{i+1}$ can see at most ϵ of segments x_k for $k \leq i-2$ and $k \geq i+2$. Moreover, c can see at most 3 segment sides. Further, note that any camera that sees the left side of the left-most segment or the right side of the right-most segment will see at most one additional segment plus at most ϵ of any others. Since we *must* see the left side of the left-most segment and the right side of the right-most segment, to see all $2n$ sides of all segments requires $k+2$ cameras where

$$3k + 4 \geq 2n \tag{1}$$

so that

$$k + 2 \geq \frac{2n}{3} + \frac{2}{3} \tag{2}$$

which is equivalent to

$$k + 2 = \left\lceil \frac{2n}{3} \right\rceil + 1. \tag{3}$$

Case (c) is rather easy. To see that $\lceil \frac{n}{3} \rceil$ cameras are sufficient, in the worst case, again assume that no two segments are positioned one on top of the other since viewing segments entirely will be easier if they *are* positioned one on top of another. As usual, extend all segments so the top-most points are the same distance from the top of the rectangle and the bottom-most points are the same distance from the bottom of the rectangle. Assuming $n \equiv 0 \bmod 3$, i.e. $n = 3k$ for some positive integer k, position the ith camera just above and to the

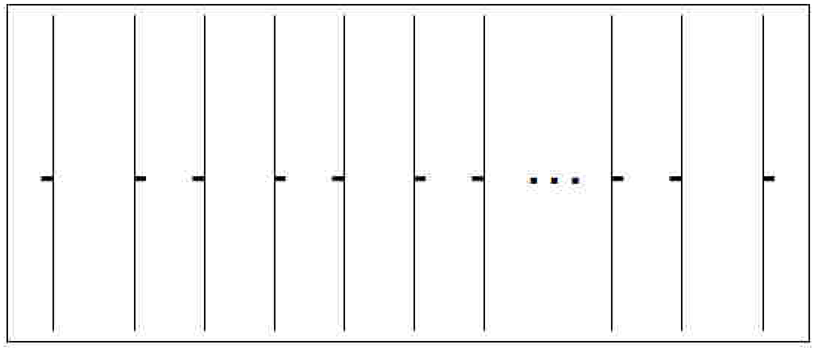

Figure 4. n segments (for $n \equiv 0 \bmod 2$) placed in such a way that with a poser's choice of sides as shown (by the "ticky" marks), a solution seeing all sides requires $\lfloor \frac{n}{2} \rfloor + 1$ cameras. The cameras must be placed between ℓ_{2i} and ℓ_{2i+1} for $1 \leq i \leq \frac{n-2}{2}$. If $n \equiv 1 \bmod 2$, just remove the rightmost segment to get an analogous case, still requiring $\lfloor \frac{n}{2} \rfloor + 1$ cameras.

right of the $(3i-1)$st segment so that it can entirely see the right side of the $(3i-2)$nd and $(3i-1)$st segments as well as the left side of the $3i$th segment. If $n \equiv 1$ or $n \equiv 2 \bmod 3$, and so $n = 3k+1$ or $n = 3k+2$, then do the same for the first $3k$ segments and position a final camera anywhere between the final two segments to either see the $(3k+1)$st segment from the right if $n = 3k+1$ or the $(3k+1)$st segment from the right and the $(3k+2)$nd segment from the left if $n = 3k+2$. The argument that $\lceil \frac{n}{3} \rceil$ cameras are sometimes necessary is essentially the same as the necessity proof in case (b).

For case (d), first let us consider the case $n \equiv 0 \bmod 2$ and consider the segments $\ell_1, ..., \ell_n$ numbered left to right. Let us say the a segment ℓ_k is "left-pointing" if we are required to see it from the left side, and "right-pointing" otherwise. If ℓ_1 is not left-pointing then we can use one camera to see the left two segments, surround the right-most $n-2$ segments with a smaller rectangle than our original bounding rectangle, with left edge to the right of ℓ_2 and be done by induction. If ℓ_1 is pointing leftward then we can assume ℓ_2 is pointing rightward since otherwise we again could see ℓ_1 and ℓ_2 with a single camera and complete the argument by induction. But since ℓ_2 is pointing rightward we can see ℓ_2 and ℓ_3 with a single camera, and in fact would also be able to see ℓ_4 with this camera if it were pointing leftward. If in fact ℓ_4 were pointing leftward then we could see the left-*two*-most segments with two cameras and so be able to finish off the argument by induction. Hence we may assume ℓ_4 is pointing rightward. We continue the argument in this way , seeing ℓ_1 with a single camera and all pairs (ℓ_{2i}, ℓ_{2i+1}) with a single camera for $1 \leq i \leq \frac{n-2}{2}$ and then at the end, see ℓ_n with a single camera. In all we see the n segments with $\frac{n-2}{2} + 2 = \lfloor \frac{n}{2} \rfloor + 1$ cameras. The same argument now applies for the case $n \equiv 1 \bmod 2$ but now we see the first segment with a single camera and the remaining $n-1$ segments with $\frac{n-1}{2}$ cameras for a total of $\frac{n-1}{2} + 1 = \frac{n+1}{2} = \lfloor \frac{n}{2} \rfloor + 1$ cameras once again. The examples described in Figure 5 also show that

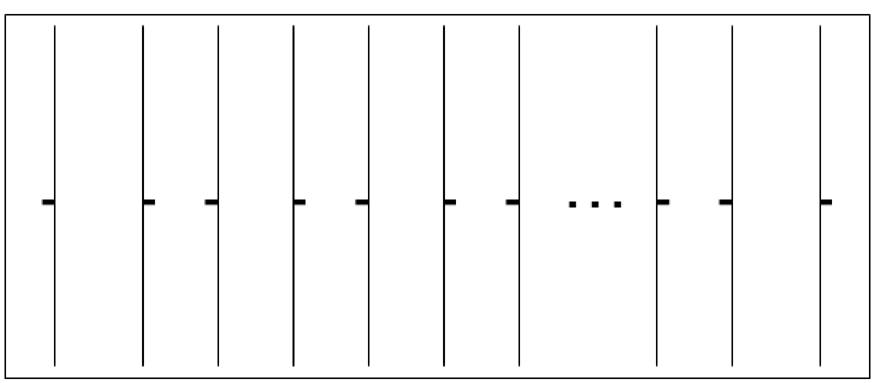

Figure 5. n segments (for $n \equiv 0 \bmod 2$) placed in such a way that with a poser's choice of sides as shown (by the "ticky" marks), a solution seeing all sides requires $\lfloor \frac{n}{2} \rfloor + 1$ cameras. The cameras must be placed between ℓ_{2i} and ℓ_{2i+1} for $1 \leq i \leq \frac{n-2}{2}$. If $n \equiv 1 \bmod 2$, just remove the rightmost segment to get an analogous case, still requiring $\lfloor \frac{n}{2} \rfloor + 1$ cameras.

$\lfloor \frac{n}{2} \rfloor + 1$ cameras are sometimes necessary. $\square$

Remark 2 *Note that stipulating just that all but some δ of each segment be guarded still requires the same number of cameras in each of the cases of Theorem 1.*

3.2 Axis-Aligned Segments

We now proceed to the considerably harder case of arbitrary axis aligned segments. We consider first the Solver's Choice problem. There are two natural variants of this problem, one (a) where the solver is able to choose to see some points on a given segment from one side and some points from the other side, and another variant (b) where, though the solver gets a choice of sides, all points on each segment must be viewable entirely from one side or the other. Solver's Choice variant (a) for axis-aligned segments was first considered by Czyzowicz et al. [4], where they showed that $\lceil \frac{n+1}{2} \rceil$ cameras always suffice. A little bit of an addition to their argument shows that $\lceil \frac{n+1}{2} \rceil$ cameras also suffice for Solver's Choice variant (b). We present this extended argument below.

Both the Czyzowicz et al. argument and our addition utilize the following classic theorem of Tutte's [25][1]:

Theorem 3 ***(Tutte)*** *A graph G has a perfect matching iff every subset of vertices S is such that the number of connected components of $G \setminus S$ of odd order is less than or equal to the number of vertices in S.*

Theorem 4 *Given n axis-aligned segments contained in a bounding rectangle, it is always possible to see the solver's choice of sides, where the solver must choose to see each segment entirely from one side or the other, using at most $\lceil \frac{n+1}{2} \rceil$ cameras.*

Proof. Czyzowicz et al. [4] actually considered the problem in the plane, where the bound is exactly the same as it is in a rectangle – indeed, the first step in the Czyzowicz et al. proof is to place the segments in a bounding rectangle – a rectangle which in our case is just given to us. Next, extend all segments so they come within some very small ϵ of another segment or one of the rectangular "walls." After extending all n line segments in this fashion, it is easy to see by induction that we end up with $n+1$ "rooms." See Figure 6 which is taken

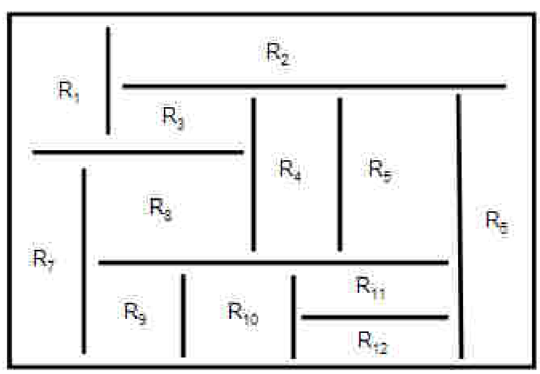

Figure 6. $n = 11$ segments extended so they come within some very small ϵ of one another or one of the rectangular walls, leaving $n+1 = 12$ "rooms" $R_1, ..., R_{12}$.

from Czyzowicz et al. [4] – as is this entire illustration and figures including the two sub-figures of Figure 7. Consider the graph whose vertices are the rooms and such that there is an edge between vertices iff there is, in Czyzowicz et al.'s terminology, a "door" between rooms, in other words there is a path between the rooms that does not pass through any other room (this is *not* equivalent to the rooms sharing a common corner; see Figure 9 and, for example, rooms R_2 and R_4 in Figure 6.). We claim that this graph has a near-perfect matching (a perfect matching if n is even and a perfect matching on $n-1$ of the vertices if n is odd. Assume the graph has an even number of vertices (i.e. by throwing in an extra axis-aligned segment if necessary) and try to apply Tutte's Theorem. Remove a set of vertices S and let k equal the number of connected components after removing vertices in S. See Figure 7. We consider the remaining connected components to be orthogonal polygons, given by the union of the associated closed rooms, as shown in the right-hand drawing in Figure 7. Each of the k orthogonal polygons contains at least 4 corner points, or at least $4k$ corner points in total. Now consider adding

[1] To gain insight into this theorem consider the tree of height two with three leaves and then look at variations on the same theme. For extra insight into Tutte's Theorem we recommend the wonderful proof due to Lovász in [19].

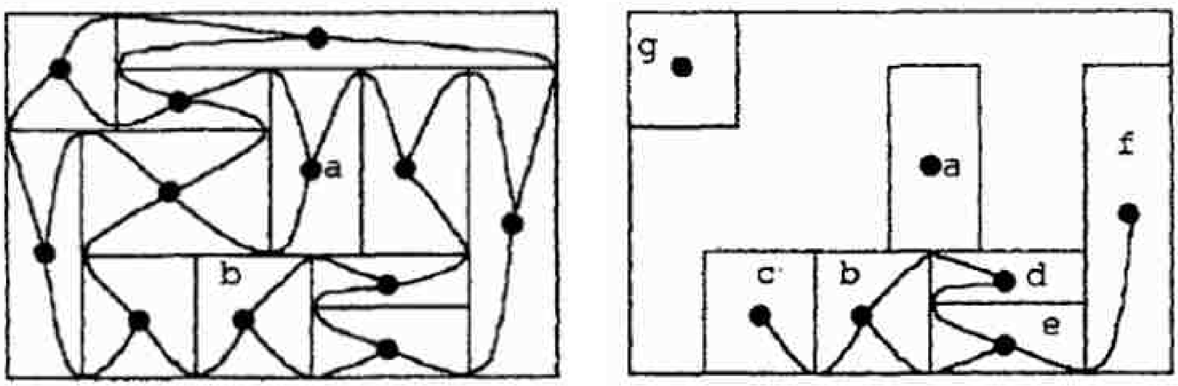

Figure 7. On the left, the room connectivity graph and on the right, the graph after removing the set of vertices S, leaving $k = 2$ connected components.

vertices back to the graph, in other words, adding rooms back. Adding a room back, removes at most 4 corner points from the cumulative set of rooms, at most one corner point for each of its corners. So we go from at least $4k$ corner points down to 4 corner points by the time we add all $|S|$ vertices back. Hence $|S| \geq k - 1$ or $k \leq |S| + 1$. Tutte's Theorem says that G has a perfect matching if the number of connected components of odd order is no bigger than $|S|$. The only way then that we could not have a perfect matching is if $k = |S| + 1$ and all connected components are of odd order. But then, remember that the total number of vertices, $|V|$, was even and $|V| = |S| + \sum_{i=1}^{k} |CC_i|$, where $|CC_i|$ denotes the number of vertices in the ith connected component. Thus if (1) $|S|$ is odd, then k is even and since each $|CC_i|$ is odd, $\sum_{i=1}^{k} |CC_i|$ must be even and $|V| = |S| + \sum_{i=1}^{k} |CC_i|$ is the sum of an odd and an even and so odd, a contradiction. On the other hand, if (2) $|S|$ is even, then k is odd and $\sum_{i=1}^{k} |CC_i|$ is odd , so again $|V| = |S| + \sum_{i=1}^{k} |CC_i|$, being the sum of an even and an odd, is odd, which is again a contradiction. Thus it must be that $k \leq |S|$ so that Tutte's Theorem applies and we have a near perfect matching.

To finish up we need to show how to use a camera at the junction ("doorway") between every pair of rooms in the near perfect matching and a single camera in the interior of the possibly one remaining room so as to see all segment sides and associated room walls, from the solver's choice of sides. Regardless of choice of segments we shall show how to guard the *right* sides of the vertical segments and the *top* sides of all horizontal segments. To see that such camera placements are always possible suppose we have connected rooms A and B separated by a horizontal wall. Figure 8 gives a generic depiction of all such examples. There is one additional way in which rooms A and B can be separated by a horizontal wall, as in Figure 9, but note that in this case rooms A and B do not share a "door" and hence are not connected. If A and B share the same left vertical wall like they do in cases (a), (b) and (d) in the figure, we place the camera to the upper left of the separating horizontal wall (denoted by h in the figure). If A and B have distinct left vertical walls, as in cases (c) and (e) in the figure, then we place the camera just above and to the right of the right edge of h, so the camera sees the entire left walls of both A and B in case (c) and the entire left wall of A and all of the left wall of B except for a small "blind spot" in case (e). In all cases the camera also sees the upward pointing side of the bottom wall of B. By an analogous and symmetrical argument camera placements are possible in the case where A and B are separated by a vertical wall (rotate each of (a) through (e), along with the associated camera placements by 90 degrees in the counter-clockwise direction) which also see the right and top sides of all room walls, modulo a small blind spot on the bottom right in rotated version of case (e). Let us then further investigate case (e), noting that the argument for the rotated version is completely analogous.

We note that there is necessarily an additional room on the other side of the "doorway" at the upper left corner of room B. In the near perfect matching this room is either a singleton or connected to another room. If the room is a singleton the single camera used to see the right side of the left wall and the top side of the bottom wall can clearly also cover the previously left blindspot. If not a singleton, then either the connected rooms are separated by a vertical or horizontal wall. If separated by a horizontal wall then it is easy to survey the cases (a) - (e) to see that next camera, which will "hug" a vertical segment above h will, in coordination with a suitably placed first camera, will be able to see the previously left blind spot, since the second camera can be positioned to have a sharper angle on the previously left blind spot and the blind spot can be made arbitrarily small and arbitrarily close to the original segment h. If the rooms are instead separated by a vertical

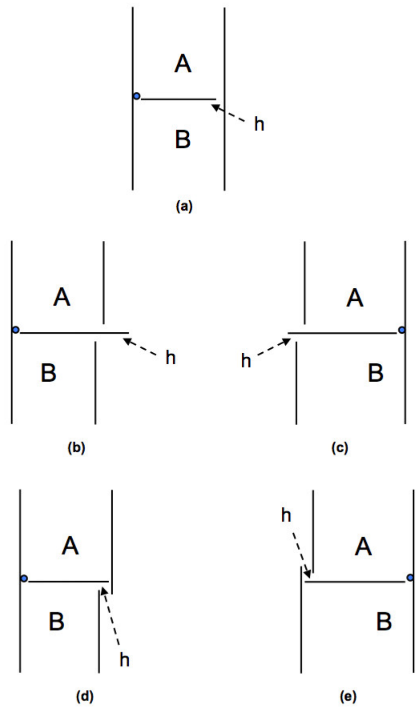

Figure 8. The different ways in which two connected rooms A and B can be separated by a horizontal wall and the associated camera placements used to see the tops of all horizontal walls and the right side of all vertical walls. In cases (a) through (d) these walls are seen completely. In case (e) there is a small "blind spot that does not allow the left wall of room B to be seen completely.

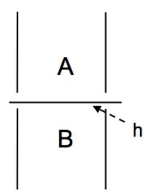

Figure 9. Two adjacent rooms separated by a horizontal wall that are not connected since they do not have a "door" between them.

wall then a similar analysis results in the same conclusion. Since there are only finitely many rooms, this process of seemingly pushing the possible blindspot forward, terminates after only finitely many steps. The theorem is thus proved. $\square$

We make the additional observation that our choice to place cameras to see all vertical segments from the right and all horizontal segments from the top was completely arbitrary. Therefore we have the following:

Corollary 5 *Given n axis-aligned segments contained in a bounding rectangle, it is always possible to see the poser's choice of sides, if the poser is constrained so specify the same side (right or left) for all vertical segments, and the same side (top or bottom) for all horizontal segments, using at most $\lceil \frac{n+1}{2} \rceil$ cameras.*

The above result is comparable to Theorem 1 (a) - which gives essentially the same bound in the all vertical case. By virtue of the "sometimes necessary" part of Theorem 1 (a) we know that $\lceil \frac{n}{2} \rceil$ cameras are sometimes necessary if all segments are required to be viewed from the poser's choice of sides, if the poser is constrained to specify all segments of a given type be viewed from the same side. With the addition of horizontal segments we do not know of an example that actually requires $\lceil \frac{n+1}{2} \rceil$ when $\lceil \frac{n+1}{2} \rceil > \lceil \frac{n}{2} \rceil$, in other words when n is even. In the case of Solver's Choice for axis aligned segments where the solver must see all points on a given segment from the same side (Theorem 4) there is a more significant gap. Theorem 1 (c) showed that $\lceil \frac{n}{3} \rceil$ cameras are always sufficient to guard n vertical segments from the solver's choice of sides. However, Figure 10 gives an example that requires $\lceil \frac{3n}{8} \rceil$ cameras to view a set of n axis aligned segments from the solver's choice of sides,

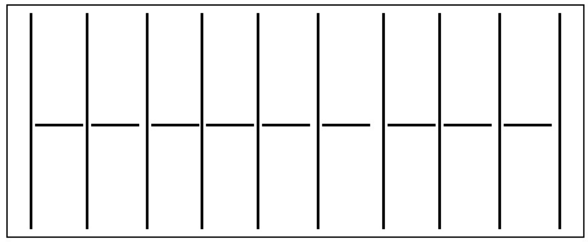

Figure 10. A set of n axis aligned segments requiring $\frac{3n}{8}$ cameras to entirely see the solver's choice of sides.

while Theorem 4 established the sufficiency of $\lceil \frac{n+1}{2} \rceil$ cameras. In Figure 10 we have a sequence of alternating vertical and horizontal segments. The pictured sequence has 10 vertical segments and 9 horizontal segments but an arbitrary alternating sequence of such segments works just the same, as long as the sequence starts and ends with a vertical segment. The left three segments in this figure require two guards to see all segments entirely from a single side. Any placement of a single guard, for example, between the two vertical segments, will leave at least a tiny "blindspot." These two guards then will see most if they are positioned with one just above and to the right of the second vertical segment from the left, and the other just below and to the right of the second vertical segment. In this case the two guards together see all of the first five segments. There must now be a guard that sees the third horizontal segment and it is easy to see that there is no benefit in seeing only part of this third horizontal segment. Thus to see most, the guard responsible for seeing the third horizontal segment should be placed either above and just to the right of the fourth vertical segment, or below and just to the right of the fourth vertical segment. Either way the three guards see the first eight segments entirely as well as just over half of the ninth segment, which is also the fifth vertical segment. Unfortunately one can easily see that there is no way to leverage the half seen vertical guards when placing further guards.

Thus we are left with the same guarding situation that we started with, but with 8 fewer segments and 3 guards already used. Hence we conclude that this example and a generic axis aligned set of segments may require as many as $\lceil \frac{3n}{8} \rceil$ cameras to view all segments from the solver's choice of sides when all points on a segment must be viewed from the same side.

Note, however, that if it were sufficient to see all but some arbitrarily small length δ of each segment, then we could actually get away with one camera for every three segments in the example of Figure 10. The segments may be thought of as a sequence of triplets of segments, starting with a vertical, then a horizontal and vertical, or VHV for short, and following with a horizontal, then a vertical and then a horizontal, or HVH, and continuing in this fashion. One can guard all but δ of each VHV by placing a guard just above and to the right of the horizontal segment, and one can guard *all* of each HVH by placing a guard just above and to the right of the vertical segment. In this all-but δ variant of the solver's choice problem we know of no worse case then the one that required $\lceil \frac{n}{3} \rceil$ cameras for all vertical segments.

If the problem poser has full choice of which sides of segments he can specify to be viewed, then more than $\frac{n}{2} + O(1)$ cameras may be necessary, as the example in Figure 11 shows. The segments in each "H" (or, in the

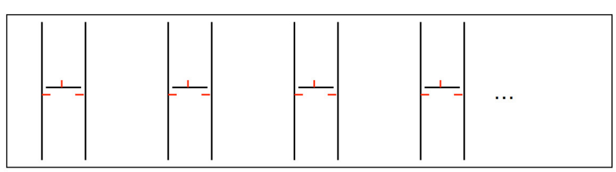

Figure 11. A set of n segments (the segments in black) for the full Poser's Choice problem, requiring $\frac{2n}{3}$ cameras. The tiny red segments are not part of the problem, but rather indicate which side of the respective segments must be seen. The segments in each of the "H" 's require 2 segments for all of the specified segment sides to be seen entirely.

nomenclature of the previous paragraph, each VHV) require two guards for all of the specified segment sides to be seen entirely. We can add a vertical segment to the left of all segments pointing to the left and a vertical segment to the right of all segments pointing to the right and the resulting collection will require $\frac{2n}{3} + 2$ cameras. The best upper bound we have been able to establish for the full Poser's Choice problem is the following:

Theorem 6 *Given n axis-aligned segments contained in a bounding rectangle, it is always possible to see the poser's choice of sides using at most $\lceil \frac{3n}{4} \rceil$ cameras.*

Proof. As in the proof of Theorem 4 extend all segments so they come within some very small ϵ of another segment or of the bounding rectangle. Once gain use the near-perfect matching guaranteed by Tutte's Theorem to pair up adjacent rooms. There are only two cases in which we have adjacent rooms, and a poser's choice of segment sides, such that a single camera cannot see all required sides of segments within the rooms. These cases are shown in Figure 12. Call matches of the form shown in Figure 12 "bad matches" since a single camera cannot

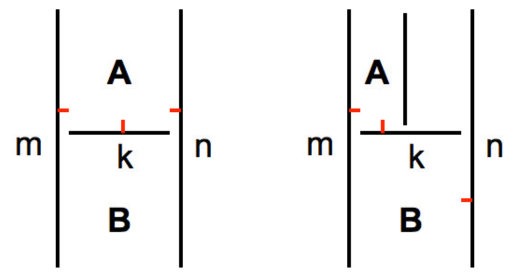

Figure 12. Two examples where a single camera cannot see all points on the required segment sides because of the inevitable blindspot.

entirely see all the needed segment sides. Note that in each of these two examples, one can use two cameras to entirely see the needed segment sides in the two rooms, marked respectively A and B in each example. We will actually use two cameras to see the three problematic segments entirely in each bad match. If we do this one bad match at a time, we can avoid having too many bad matches, since once a segment has been seen entirely by one camera it doesn't need to be seen again. Thus, under the assumption that none of the problematic edges of one bad match appear in another bad match, there can be at most $\frac{n}{3}$ bad matches.

We now break the analysis into two cases: (i) There are $b \leq \frac{n}{4}$ bad matches, or (2) there are $\frac{n}{4} < b \leq \frac{n}{3}$ bad matches. In case (i) we use 2 cameras in each bad match to see the three problematic segment sides entirely along with any other required segment sides required to be seen within the two rooms, and 1 camera in each good (i.e. non-bad) match to see required segment sides within those rooms, thus using at most $2(\frac{n}{4}) + \lceil \frac{n}{4} \rceil = \lceil \frac{3n}{4} \rceil$ guards. In case (2) suppose there are $\frac{n}{3} - h$ bad matches for $0 \leq h < \frac{n}{12}$. In this case use 2 cameras to see each of the 3 problematic defining line segments (i.e. the analogs of m, n, k in Figure 12) and 1 camera to see each remaining line segment. Again a simple computation shows that $\lceil \frac{3n}{4} \rceil$ guards suffice. □

Despite this result and the $\frac{2n}{3} + O(1)$ lower bound from Figure 11, we have the following somewhat practical caveat:

Theorem 7 *Given any $\delta > 0$ and n axis-aligned segments contained in a bounding rectangle, it is always possible to see the poser's choice of sides, using at most $\lceil \frac{n+1}{2} \rceil$ cameras if we are required to see all the requested segments except, cumulatively, at most length δ along these segments.*

Proof. Looking back at the proof of Theorem 6, and in particular at Figure 12, we see that the only cases in which a guard or camera fails to see the requisite side of all requested segments in a given room occurs when it is required to leave an arbitrarily small blindspot. Since only a finite number of cameras are used they can be coordinated to leave blindspots of cumulative length at most δ. □

It is important to note that we cannot hope to see both sides of all segments using this method; in particular we only ever see one side of the segment k in Figure 12, which separates room A from room B. Indeed, the best conceivable combinatorial bound for seeing both sides of a set of axis aligned segments cannot be any better than the $\lceil \frac{2n}{3} \rceil + 1$ bound we obtained for the case of all vertical segments back in Theorem 1. Recall that this bound was tight even when we allowed for omission of up to length δ along all segments.

3.3 Arbitrarily Aligned Segments

Although we have not studied the line segment visibility problem in any detail for non axis-aligned segments it is worth stating what is known of these results. In 2003 Csaba Toth [24] extended the result of Theorem 4 to the case of the Solver's Choice Problem for segments of arbitrary orientation where the segments can be viewed partially from one side and partially from another. The result is still that $\lceil \frac{n+1}{2} \rceil$ guards suffice to see all segments. Urrutia [27] has given a construction showing that $\lfloor \frac{2n-3}{5} \rfloor$ guards are sometimes necessary in the

Solver's Choice Problem for arbitrarily aligned segments in either the case where points of a segment can be seen from either side or just one side.

One of the two tight bounds that are known in 2D (the case of all vertical segments being considered 1D) is the case of arbitrarily aligned segments where the segments have to be seen from *both* sides. In this case Toth [24] has given a $\lfloor \frac{4n+1}{5} \rfloor$ bound as well as provided a concrete example showing that this bound is tight.

The following tables show the state of our knowledge regarding both upper and lower bounds for each of the problems we have examined, up to constant factors:

		Solver's Choice 	Poser's Choice (All segments from same specified side)	Poser's Choice	Both Sides
Vertical	U n/3 L n/3	U n/3 L n/3	U n/2 L n/2	U n/2 L n/2	U 2n/3 L 2n/3
Orthogonal	U n/2 L n/3	U n/2 L 3n/8	U n/2 L n/2	U 3n/4 L 2n/3	U 4n/5 L 2n/3
Arbitrary	U n/2 L 2n/5	U 3n/4 L 2n/5	U 3n/4 L n/2	U 3n/4 L n/2	U 4n/5 L 4n/5

Figure 13. A table summarizing what we know for the various problem variants. All stated results are modulo constant factors.

		Solver's Choice 	Poser's Choice (All segments from same specified side)	Poser's Choice	Both Sides
Vertical	U n/3 L n/3	U n/3 L n/3	U n/2 L n/2	U n/2 L n/2	U 2n/3 L 2n/3
Orthogonal	U n/2 L n/3	U n/2 L n/3	U n/2 L n/2	U n/2 L n/2	U 4n/5 L 2n/3
Arbitrary	U n/2 L 2n/5	U 3n/4 L 2n/5	U 3n/4 L n/2	U 3n/4 L n/2	U 4n/5 L 4n/5

Figure 14. A table summarizing what we know for the various problem variants, where we allow segments to be viewed from the indicated sides but also allow that up to some fixed δ be left unguarded, for arbitrarily small δ. All stated results are modulo constant factors.

As one moves from the top-left to the bottom-right of each table the problems become consistently harder. Thus the number of cameras required to solve cell (i, j) is less than or equal to the number of cameras needed to solve either cell $(i+1, j)$ or $(i, j+1)$, for all i, j for the same table. Moreover, the same is true for any established upper and lower bounds. If we ever understand everything about this family of problems the upper and lower bounds in each cell will be equal. For now, most have gaps. In many cases, we have not established the bounds for a given cell independently but the bound is "inherited" from a neighboring cell. For example in the first table, in the cell showing upper and lower bounds for the problem of viewing both sides in the axis-aligned problem variant, the upper bound of $\frac{4n}{5}$ comes from the same problem for segments with arbitrary alignment. The upper bound in the axis aligned problem variant can be no higher than this value. Similarly, in this same cell, the lower bound of $\frac{2n}{3}$ comes from the same problem for segments that are all vertical. The lower bound for the axis aligned problem variant can be no lower than this value.

4 Hardness Results

In this section we prove that many variants of the problems we discuss in this paper are NP-hard. We begin with the simplest version.

Theorem 8 *Guarding vertical segments from the left is NP-hard.*

Proof. We show a reduction from 3-SAT. Each variable gadget consists of a batch of six segments (of the same length) and each clause gadget consists of a batch of five segments. Figure 15 illustrates the reduction.

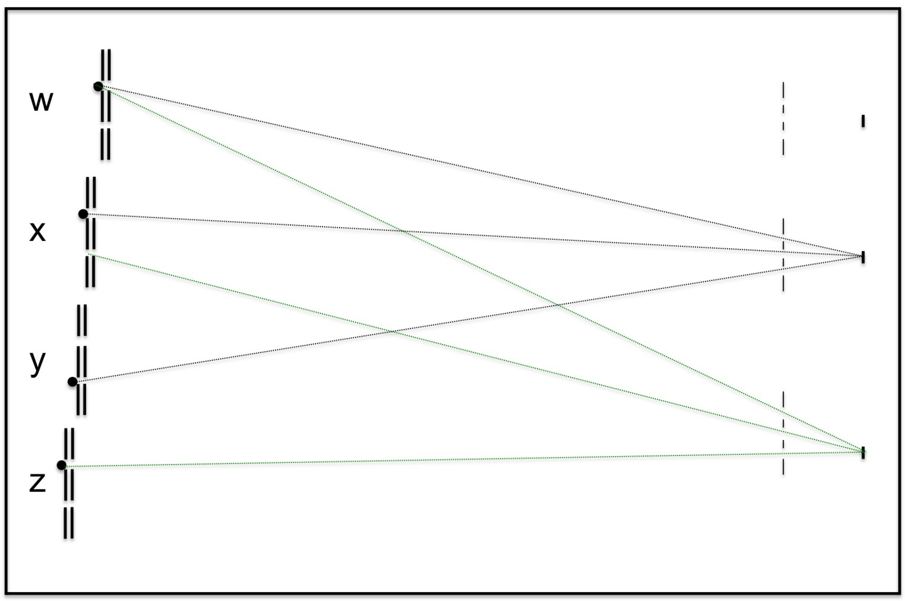

Figure 15. Demonstration of the reduction from 3-SAT in the proof of Theorem 8. On the left are the gadgets for the variables w, x, y and z. On the right are the gadgets for the clauses $(w \cup x \cup \bar{y})$ and $(w \cup \bar{x} \cup z)$. The (point) guards associated with the respective literals are indicated as circles to the left of the respective truth or false gaps. The line of site of a guard to the critical rightmost segment of each clause, c_r is indicated with a dotted line. Lines of sight for guards in the same clause are each of the same color - either black or green above.

The variable gadgets are located on the left. Each gadget consists of two very close columns of three vertical segments. Note that each is shifted to the left with respect to the one above it. In each variable gadget we refer to the gap between the top two segments and the middle two segments as the *truth* gap, and the gap between the middle two segments and the bottom two segments as the *false* gap. We refer to either gap as a *literal gap*. The clause gadgets are located on the right in one column. Each clause consists of one short segment on the right and four segments of the same x-coordinate to its left. For any clause gadget c, we denote by c_r the right segment and by c_L the four segments to its left. We define the guarding instance so that all segments have to be guarded from the left. For any clause gadget c, consider the segment c_r. The segments c_L will block c_r from seeing almost the entire bounding box. The three gaps in between c_L constitute the only way that c_r can *see* far. They are fine-tuned so that c_r *sees* the gaps in the variable gadgets that corresponds to the gadget literals.

For example, consider Figure 15 which corresponds to a formula with three clauses (two of which are given in detail) and four variables (w, x, y and z). Consider the middle clause gadget c' that corresponds to $(w \cup x \cup \bar{y})$: c'_r *sees* the three literal gaps that correspond to w, x and $\bar{y}$ (the lines of sight from the guards to the critical segment c_r' are drawn with dashed black lines). The third clause, $(w \cup \bar{x} \cup z)$, is similarly depicted, this time with green dashed lines indicating the lines of site. The clause gadgets are easy to construct. The segments c_r are equally spaced and in a vertical line. One then connects each of the guards associated with the clause with the top and bottom of c_r. These determine the gaps between the blockers in the associated batch of four segments c_L. To make sure that blocker groups do not overlap with one another, just move all blocker groups sufficiently far to the right.

Next we prove that a 3-SAT formula with C variables is satisfied if and only if the segments are guarded by C guards.

$\Rightarrow$ **Direction:** Suppose the 3-SAT formula is satisfiable. We place one guard near each corresponding literal gap. It is positioned infinitesimally to the left of the corresponding gap so that it can *see* through the entire gadget so that it would see the vast majority of the bounding box if the clause gadgets were not present. We argue that all segments are guarded as follows:

- Since the guard is located infinitesimally to the left of a gap, the segments of its gadget are guarded by it.
- The four left segments of the clause gadgets are guarded by all guards since the pairs of vertical segments associated with a variable gadget are within ϵ of one another and the guard is just to the left and centered vertically in the gap, and, moreover, if h is the height of the gap (distance between vertically aligned segments), then $w \gg \epsilon$.
- Since the 3-SAT formula is satisfied, the right segment of any clause gadget is guarded by at least one guard – this follows by the geometry of the construction.

$\Leftarrow$ **Direction:** Suppose all segments are guarded by C guards. In order to guard the right segments of each variable gadget from the left we are forced to position one guard for each such gadget to the left of the right segments. Note that it is easy to guard the four segments of all clause gadgets: placing a single guard next to either any truth or false gap will take care of that. However, in order for one of the C guards to see the entire right segment of a particular clause gadget it must be positioned at the exact location in one of the literal gaps, just to the left of the left segments in the variable gadget, from which the blocking segments were determined (a location that henceforth we call a "reference guard location"), or to the right of such a location, but within a very small distance of the visibility line going from the reference guard location to the midpoint of the right segment in the clause gadget, so that the guard can still see the entire right segment of the clause gadget. To see the right segments of the variable gadgets we must use one guard per variable gadget, and to see each of the right-most segments of the clause gadgets, we must use one of the three guard regions that can be interpreted as a requisite truth assignment associated with a particular variable in each respective clause. Thus the fact that the C guards see the left-hand side of all segments implies that there is a satisfying assignment of the associated 3-SAT formula. $\square$

Theorem 9 *The following variants and any combination of them are NP-hard:* **(a)** *Guarding vertical segments from the poser's choice of side.* **(b)** *Guarding vertical segments from both sides.* **(c)** *Guarding segments with any orientation from the poser's choice of side.* **(d)** *Guarding with restricted angle.* **(e)** *Guarding at least one point in each segment from the left.* **(f)** *Guarding vertical segments from the solver's choice of side.*

Proof. **(a)** Generalization from Theorem 8. **(b)** We add two guards, one to the left of the structure and one to its right (denoted by g_l and g_r, respectively). We tune the structure such that g_r sees the right side of all segments except the ones in the left columns of the variable gadgets. So that g_r sees the right side of the right-most segments of the top-most variable gadget we can extend the bounding rectangle down as far as necessary, while sliding g_r down but keeping the rest of the structure fixed. g_l will see the left side of segments in the left columns of the variable gadgets. The decision problem measure will be set to $C + 2$. Note that to see the left side of the left segments in the variable gadgets we need to position a guard to their left and to see the right side of the right segment of the clause gadgets we must position a guard to their collective right. Additionally, to see the right side of the left segments of the variable gadgets we need to place an additional C guards, one per variable gadget. Since neither of the first two guards see the left side of the right-most segments of the clause gadgets, these are the responsibility off the additional C guards. The rest of the details of the hardness proof are identical to Theorem 8. See Figure 16 for an illustration. **(c)** Generalization of Theorem 8. **(d)** We simply compress the model horizontally so that the visibility angle of the guards suffice to view everything necessary for Theorem 8 to hold. **(e)** It is easy to fine-tune the model (if necessary) to ensure that this variant holds. The important segments are (as in Theorem 8) the right segments of the gadget clauses – we make sure that no gap other than those from the corresponding literal gaps can see any point of these segments. **(f)** We modify the proof of (b) as follows. We duplicate all segments and place the replacement twin segments infinitesimally close to each other, centered about the same point as the original. For any original segment s, let s_r and s_l be the two segments that replace it in this segments (s_r being the right one, and s_l the left one). In order to guard all segments with $C + 2$ guards, the solver is forced to guard s_r from the right and s_l from the left, for each pair of such segments, mimiking the similar cover shown in (b) above, where each pair of duplicate segment act as a one segment that needs to be covered from both sides.

It is easy to demonstrate that any meaningful combination of the above variants is also NP-hard. We omit the details. □

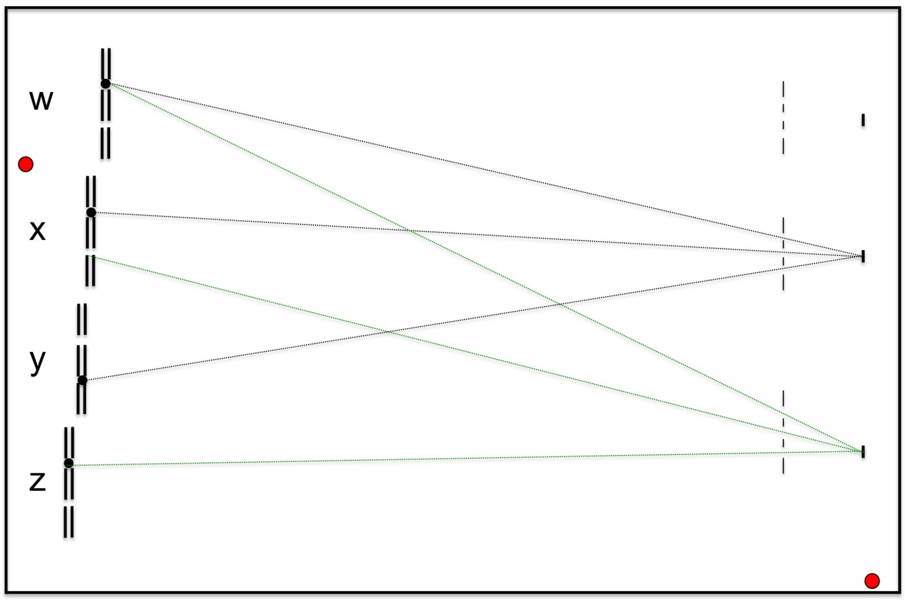

Figure 16. Demonstration of the reduction Theorem 9(b). Note the changes from Figure 15: the guards are now in between the segments of the variable gadgets and the two new special guards (in red), g_l to the left, and g_r to the right.

5 Heuristics for Guarding Data Centers

In this section we describe practical heuristics for computing guarding sets for data centers. A typical data center can be modeled as a polygon with many rectangular holes, each hole representing a cluster of computers or a visibility obstacle (e.g. a rack, cluster of racks, or computer room air conditioning unit (CRAC)). Single edges of some of the holes need to be guarded. These represent the fronts of the racks or clusters, where the computers are accessible and have their air intakes. This setting is analogous to the algorithmic model where segments need to be guarded from problem poser's choice of sides, though we have only thus far investigated theoretical problems where the segments are free-standing, all segments need to be guarded, and there are no holes. Note that guards cannot be positioned inside holes, so the segments that need to be guarded will necessarily be guarded from *outside* each hole.

Our heuristics proceed as follows. Similar to the approach in [28], we extend the edges of the polygon (only in the non-convex case where the extension goes through the interior of the polygon) and the holes in both directions until they hit other holes or the boundary. We then consider the induced arrangement A and place candidate guards in the center of each face F that does not lie inside a hole. See Figure 17 for a simple example, where the polygon is a rectangle and, analogous to the case of a data center, the holes are all rectangles. It is

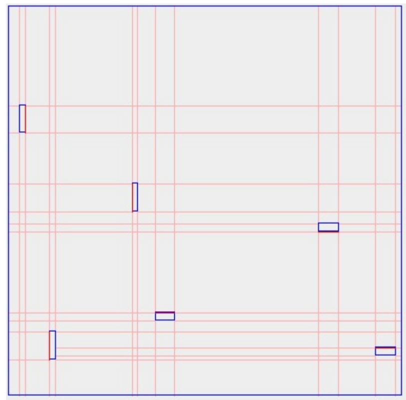

Figure 17. A sample room with six holes and the induced arrangement A after extending the edges of the holes in both directions until they hit other holes or the boundary. Since the polygon in this case is just a rectangle (and hence convex), the edges of the polygon need not be extended.

easy to verify that this set of candidate guards cover the necessary segments (the ones that need to be guarded).

We differentiate between two visibility models:

- *Complete* guarding: Some guard must fully see each segment that needs to be guarded.
- *Shared* guarding: A segment can be covered by multiple guards. Each guard can cover different parts of a given segment This variant potentially decreases the guarding set size, however, it is more costly to process.

Given a set of candidates, we employ two methods for selecting the guarding set: (a) Define the problem as a set cover problem and formulate an Integer Programming (IP) problem to solve it optimally. (b) Use a greedy method, similar to [28]. The idea, in this greedy case, is to iteratively select a guard that sees the as-yet largest number of uncovered segments that need to be guarded, and continues, until all segments are guarded.

5.1 Integer Programming Formulation

5.1.1 Complete Guarding

Let S be the collection of n segments and C the collection of m candidate guards.

Let $M_{n\times m}$ be a matrix whose entry $m_{i,j}$ is 1 if and only if candidate c_j sees segment s_i completely. Consequently, our problem can be formulated as an IP instance as follows:

Min $\Sigma(g \in \vec{G})$ **subject to**

$$\mathbf{M} \cdot \vec{G} \geq \vec{1}$$
$$\vec{G} \in \{0,1\}X \cdots X\{0,1\}$$

where G is a vector of m elements that represents the status of the candidates (1 if chosen, 0 otherwise).

5.1.2 Shared Guarding

For each candidate c, we compute its visibility polygon, from which we derive for each segment s the intervals of s seen by c. We then partition s into a connected set of intervals, each one seen by a fixed set of candidates. Having partitioning the segments, the formulation is similar to the formulation in Section 5.1.1. Here, instead of complete segments, we use the set of intervals comprising all the segments.

6 Experiments

We conducted experiments on both real-world and randomly generated data centers. We ran the heuristics described in Section 5 on each such data center. The layouts of the various data centers we tested, showing only features of relevance for monitoring, are given in Figures 18 and 19.

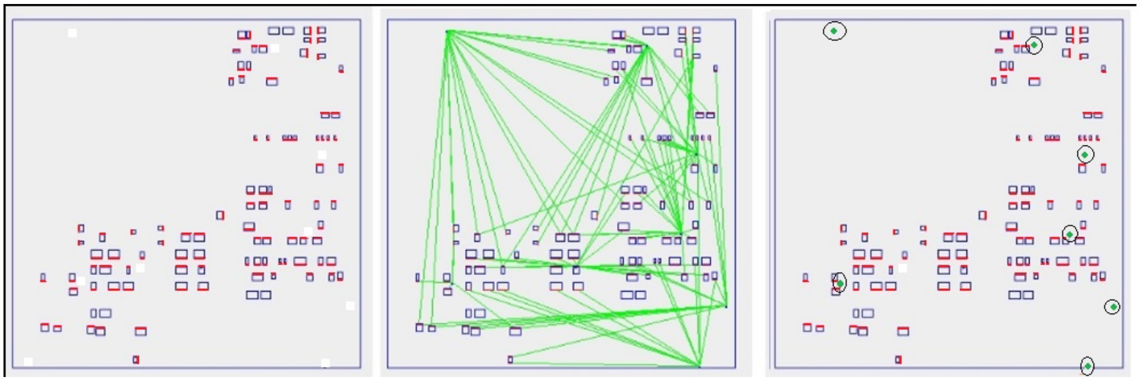

Figure 18. Data center *DC1*. Segments that need to be guarded are drawn in red. On the left is the data center, on the right we have added the guards found by our IP heuristic (marked and circled), and in the middle we show connecting lines between the guards and the segments they see.

The rectangles in the random data centers are obviously not as neatly aligned as their counterparts in the real-world data centers, and the edges of rectangles that need to be guarded in the randomly generated data centers do not as systematically face in a given direction (typically the direction of the cool air source, as described in the Introduction). The random data centers are therefore somewhat more difficult to guard. The only convention adopted when generating the rectangles in the random data centers, was that the rectangles approximate the size of real racks or real clusters of racks. Otherwise their locations were picked randomly but such that they do not overlap.

Before presenting the results, we point out the following important observation. The shared guarding version of the problem, where multiple guards could collectively be used to see segments, was found to be practically implementable only for very small data centers. Moreover, the results obtained with it were very similar to the results with the complete guarding model. Hence, we do not consider this model in our experimental results.

Data cluster	# clusters	# segments to guard	# candidates	Greedy solution	Greedy time (sec.)	IP solution	IP time (sec.)
DC1	97	91	4505	8	27	7	147
DC2	296	239	8318	76	66	X	X
Random	19	19	854	4	4	4	1
Random	35	35	2070	6	5	6	11
Random	52	52	3689	10	8	8	49
Random	66	66	4215	12	11	11	78
Random	78	78	5136	16	21	14	158
Random	92	92	5715	20	35	18	284
Random	116	116	6628	27	52	23	406
Random	127	127	6932	30	61	X	X

Table 1. Results obtained from our experimental implementations. X refers to cases where the IP implementation did not complete.

The results of our experiments are shown in Table 1 and illustrated in Figure 20. While *DC1* was small enough to be applicable for IP, *DC2* was too big for the IP to complete. Our tests with the random data were helpful in evaluating the performance of the heuristics.

We observed a tradeoff between quality of results and performance in choosing between the greedy and IP heuristics. The latter usually produces better results, but took more time to complete. This tradeoff is evident in both Table 1 and Figure 20. In Figure 20 we can also observe the exponential and super linear time performance of the IP heuristic and the greedy heuristic, respectively. Note that in large instances, the IP instance takes an unreasonable amount of time due to its exponential behavior, and thus beyond a certain point we are limited to using the greedy heuristic. We encountered this situation in several instances (note the X's in the table).

Considering the above, we recommend first trying the IP implementation and if the IP implementation takes too long to complete, resorting to the greedy heuristic.

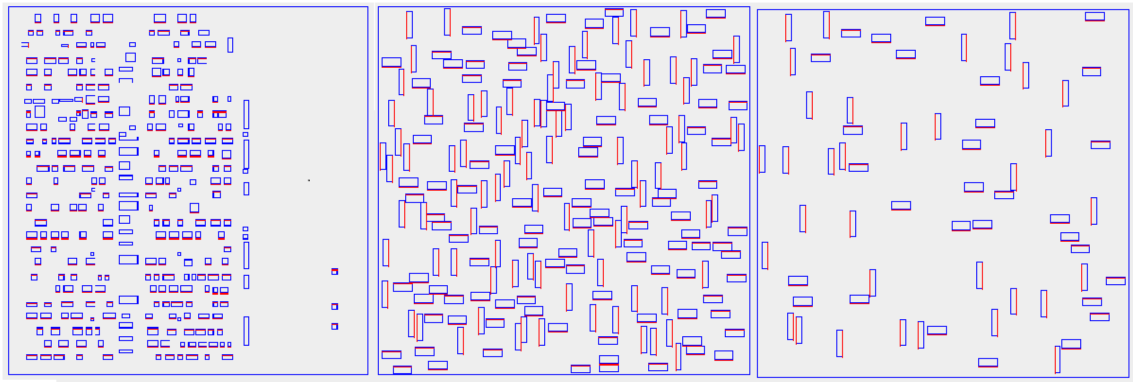

Figure 19. Other data centers. From left to right: data center $DC2$ and random data centers with two different equipment densities.

7 Conclusions and Future Work

In this paper we have examined a family of visibility problems motivated by the desire to efficiently monitor critical locations in a computer data center. We have considered several models of these data centers and their contents. In the simplest case, we modeled the data center as a polygonal enclosure containing a family of one-sided line segments that need to be guarded. We formulated many variations on the guarding theme, in some cases the problem solver got to choose which side of the segments to guard and in others it was the problem poser. Sometimes both sides needed to be guarded, and sometimes everything but some small length δ. We considered combinatorial bounds for this family of problems and found a number of interesting upper and lower bounds as summarized in the tables of Figures 13 and 14. Our interest in the subtly different variations is so that some day we can hopefully characterize precisely where the requirement for more cameras comes from as we move from the easier to harder problems.

After considering the combinatorial bounds, we showed that finding exact solutions to virtually all of these problems is NP Hard. From there we moved to a more realistic models of data center where the items to be monitored, the racks and rack clusters, were rectangles with distinguished edges that needed guarding. We described two heuristics for coming up with guarding sets and tested these heuristics against real world data centers as well as randomly generated ones.

Although we have stated and solved the various data center visibility problems as 2D problems, these are just simplified models of what are really 3D problems. Rather than one sided line segments in a 2D world we really have 2D rectangular slabs that we must monitor from one side or the other in a 3D world. It would be natural to next tackle this more realistic model.

An interesting variant of the problems we have stated, that we have not explored at all, is the watchman's route variant. Given a set of one sided segment and a segment guarding requirement (or rectangles with distinguished edges), find the shortest route such that a mobile guard can see all required segment sides (distinguished edges) at some point during the route. Dumitrescu et al. [8] studied a problem of this ilk for lines and line segments but where all the lines or line segments are connected and the watchman's route is constrained to lie within the union of the lines or line segments. Interestingly, the 2D variants of the Dumitrescu et al. problems are polynomially tractable, though the 3D variants are not. Our problem is considerably different in that our segments are not connected and have sidedness constraints. The watchman's route problem in our case is especially relevant to the case where the guard is a robot, as in the data center robot of [16, 20], or to a team of robots. The robot described in [16, 20] travels at unit speed, but makes just L1 moves and has turn costs, so if such a robot is considered one should consider not the shortest route but the most efficient travel time. How far from optimal can it be to build a tour out of a minimal guard set?

Acknowledgements

We acknowledge Csaba Tóth, who showed us the example establishing the $\frac{2n}{3} + O(1)$ lower bound for the case of Poser's Choice and axis-aligned segments. We also thank Matya Katz, Joe Mitchell and Gila Morgenstern for helpful conversations. Finally we thank Hendrik Hamann, Fernando Marianno, and the IBM MMT team who

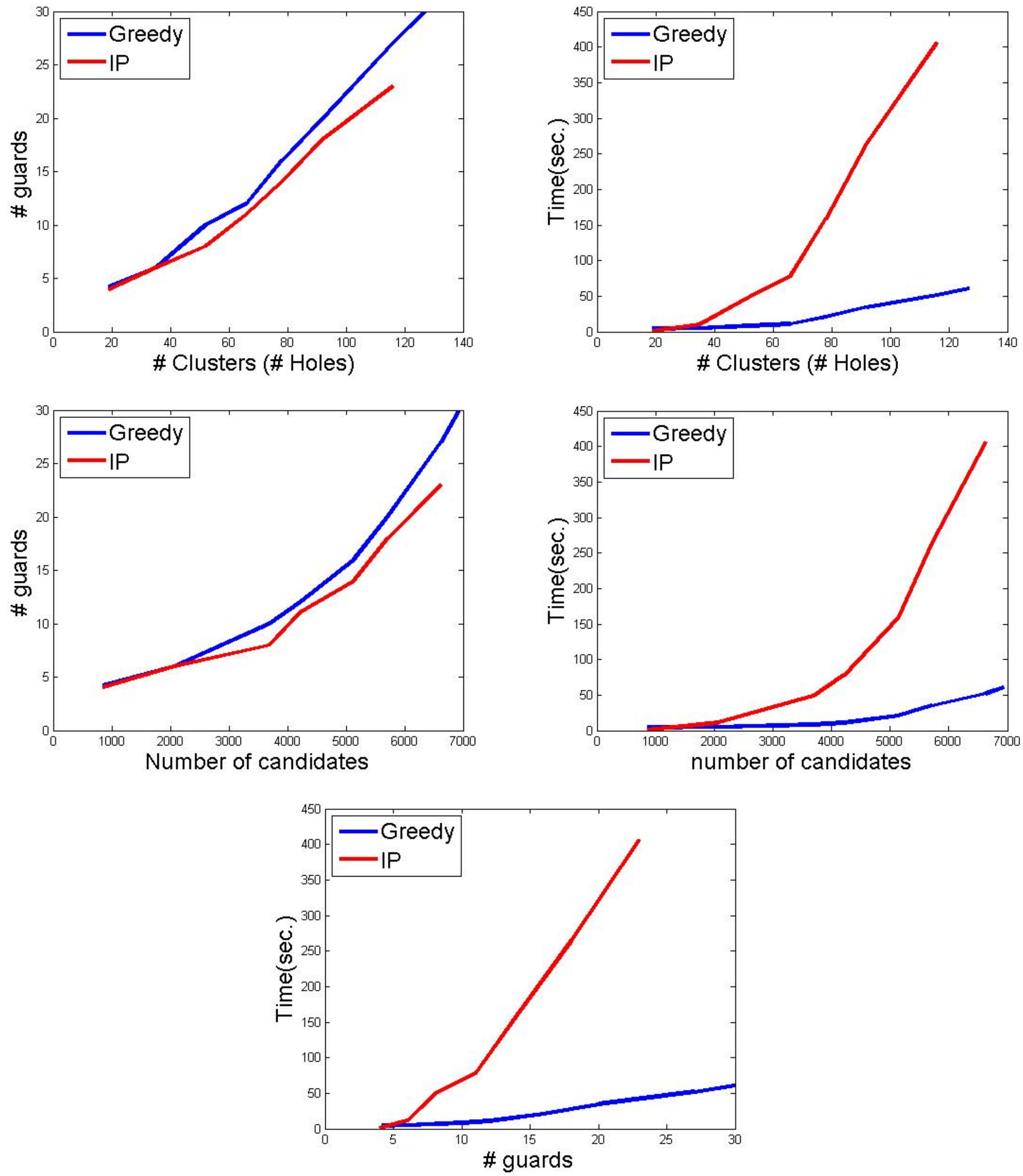

Figure 20. Comparing features of our experiment for the two real world data centers and for random data centers having different numbers of segments that must be seen.

gave us access to some of the data files used in the experimental part of this work. Figures 1 and 2 were obtained from this team as well. Hendrik and his team are the data center energy efficiency pioneers at IBM.

References

[1] C. Bash, C. Patel, and R. Sharma. Dynamic thermal management of air cooled data centers. *Proceedings of the 10th Conference on Thermal and Thermomechanical Phenomena in Electronics Systems (ITHERM 06)*, pages 445–452, 2006.

[2] B. Ben-Moshe, M. J. Katz, and J. S. B. Mitchell. A constant-factor approximation algorithm for optimal 1.5d terrain guarding. *SIAM J. Comput.*, 36(6):1631–1647, 2007.

[3] V. Chvatal. A combinaturial theorem in plane geometry. *Journal of Combinaturial Theory Ser B18*, pages 39–41, 1975.

[4] J. Czyzowicz, E. Rivera-Campo, J. Urrutia, and J. Zaks. On illuminating line segments in the plane. *Discrete Mathematics*, 137:147–153, 1995.

[5] R. Das, J. Kephart, J. Lenchner, and E. Packer. Internal surveillance problems in data centers. *Fall Workshop on Computational Geometry*, 2008.

[6] P. J. de Rezende, C. C. de Souza, S. Friedrichs, M. Hemmer, A. Kroller, and D. C. Tozoni. Engineering art galleries. *CoRR*, abs/1410.8720, 2014.

[7] A. Deshpande, T. Kim, E. D. Demaine, and S. E. Sarma. A pseudopolynomial time log(N)-approximation algorithm for art gallery problems. In *WADS*, pages 163–174, 2007.

[8] A. Dumitrescu, J. S. B. Mitchell, and P. Zylinski. Watchman routes for lines and line segments. *Computational Geometry*, 2014.

[9] A. Efrat and S. Har-Peled. Guarding galleries and terrains. In *IFIP TCS*, pages 181–192, 2002.

[10] S. Fisk. A short proof of Chvatal's watchman theorem. *J. Combin. Theory Ser. B 24*, page 374, 1978.

[11] S. K. Ghosh. Approximation algorithms for art gallery problems. *Proc. of the Canadian Information Processing Society Congress*, pages 429–434, 1987.

[12] H. Gonzalez-Banos. A randomized art-gallery algorithm for sensor placement. In *SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry*, pages 232–240, New York, NY, USA, 2001. ACM Press.

[13] H. F. Hamann, J. Lacey, M. OBoyle, R. R. Schmidt, and M. Iyengar. Rapid three dimensional thermal characterization of large-scale computing facilities. *IEEE Transactions on Components and Packaging Technologies*, 38(2):444–448, 2008.

[14] H. F. Hamann, M. Schappert, M. Iyengar, T. van Kessel, and A. Claassen. Methods and techniques for measuring and improving data center best practices. *Proceedings of 11th Intersociety Conference on Thermomechanical Phenomena in Electronic Systems*, pages 1146–1152, 2008.

[15] D. T. Lee and A. K. Lin. Computational complexity of art gallery problems. *IEEE Trans. Info. Th IT-32*, pages 276–282, 1986.

[16] J. Lenchner, C. Isci, J. Kephart, C. Mansley, J. Connell, and S. McIntosh. Towards data center self-diagnosis using a mobile robot. *Proceedings of the 8th International Conference on Autonomic Computing (ICAC)*, pages 81–90, 2011.

[17] J. Lenchner and E. Packer. Visibility problems concerning one-sided segments. *Fall Workshop on Computational Geometry*, 2012.

[18] H. Liu, E. K. Lee, D. Pompili, and X. Kong. Thermal camera networks for large datacenters using real-time thermal monitoring mechanism. *The Journal of Supercomputing*, 2013.

[19] L. Lovasz and M. D. Plummer. *Matching Theory*. North-Holland Press, Amsterdam, 1986.

[20] C. Mansley, J. Connell, C. Isci, J. Lenchner, J. O. Kephart, S. McIntosh, and M. Schappert. Robotic mapping and monitoring of data centers. *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[21] J. O'Rourke. *Art gallery theorems and algorithms*. Oxford University Press, Oxford, 1987.

[22] S. Sarma. A pseudo-polynomial time $O(log^2 n)$-approximation algorithm for art gallery problems. M.S. thesis, Massachusetts Institute of Technology, 2006.

[23] T. Shermer. Recent results in art galleries. *Proc. of the IEEE*, 90(9):1384–1399, 1992.

[24] C. Tóth. Illuminating disjoint line segments in the plane. *Discrete and Computational Geometry*, 30:489–505, 2003.

[25] W. T. Tutte. The factorization of linear graphs. *J. London Math. Soc.*, 22:107–111, 1947.

[26] J. Urrutia. Art gallery and illumination problems. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. Elsevier Science Publishers, Amsterdam, 2000.

[27] J. Urrutia. Art gallery and illumination problems. In J. Sac and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. Elsevier Science Publishers, Amsterdam, 2000.

[28] Y.Amit, J. Mitchell, and E.Packer. Locating guards for visibility coverage of polygons. *Int. J. Comput. Geometry Appl.*, 20:601–630, 2010.