Electronic Colloquium on Computational Complexity, Report No. 19 (2011)

On the Approximability of a Geometric Set Cover Problem

Valentin E. Brimkov[1], Andrew Leach[2], Jimmy Wu[2], and Michael Mastroianni[2]

[1] Mathematics Department, SUNY Buffalo State College, Buffalo, NY 14222, USA
[2] Mathematics Department, University at Buffalo, Buffalo, NY 1426-2900, USA

Abstract. Given a finite set of straight line segments S in $\mathbb{R}^2$ and some $k \in \mathbb{N}$, is there a subset V of points on segments in S with $|V| \leq k$ such that each segment of S contains at least one point in V? This is a special case of the set covering problem where the family of subsets given can be taken as a set of intersections of the straight line segments in S. Requiring that the given subsets can be interpreted geometrically this way is a major restriction on the input, yet we have shown that the problem is still strongly NP-complete. In light of this result, we studied the performance of two polynomial-time approximation algorithms which return segment coverings. We obtain certain theoretical results, and in particular we show that the performance ratio for each of these algorithms is unbounded, in general.

Keyword: guarding set of segments, set cover, vertex cover, approximation algorithm, worst case performance

1 Introduction

Given a finite set of straight line segments, we wish to find a minimum number of points so that every segment contains at least one chosen point. Consider any physical structure that can be modeled by a finite set of straight line segments. Some examples could be a network of streets in a city, tunnels in a mine, corridors in a building or pipes in a factory. We want to find a minimum number of locations where to place “guards” in a way that any point of the structure can be “seen” by at least one guard. An equivalent problem is to find a minimum number of locations to place “terminals” so that any point of the network has a direct access to at least one “terminal” at all times. For brevity, we call this problem *Guarding a Set of Segments* (GSS).

GSS is germane to the set cover (SC), vertex cover (VC) and edge cover (EC) problems. These are fundamental combinatorial problems that play an important role in complexity theory. It should be noted that we can find applications of GSS anywhere that we find applications of VC where a planar embedding of the graph is relevant or the vertices of the graph represent objects with geometric locations.

GSS can be formulated as a special case of the set cover problem (see Section 2), under certain conditions as a vertex cover problem and under other conditions as an edge cover problem. However, in general, GSS and VC are different, as Figure 1 demonstrates. It is well-known that both SC and VC are NP-complete while EC is solvable in polynomial-time [3], [4]. In the special case where at most two segments intersect at any single point, GSS is solvable in polynomial-time. This can be done by solving the edge cover problem after reducing the problem to the graph where vertices are segments and edges are intersections (see Section 4). In a recent work [2] we proved that GSS is NP-complete as well. For this reason, if we allow more than two segments to intersect at a single point, it is unrealistic to expect any efficient algorithm for finding the optimal solution.

Let us also mention that the proof of the GSS’ NP-completeness from [2] as a matter of fact demonstrates that VC on planar cubic graphs (3PVC), which is a strongly NP-complete problem, is a special case of GSS. Thus GSS appears to be “sandwiched” between 3PVC and SC—two NP-complete problems with quite different approximability (constant and unbounded, respectively). This makes the question about GSS approximability definitely interesting.

We remark that GSS also belongs to the class of the *art gallery problems.* A great variety of such problems have been studied for at least four decades. The reader is referred to the monograph of Joseph O’Rourke [7] and the more recent one of Jorge Urrutia [9]. See also [1], [5] and the bibliography therein for a couple of examples of art-gallery problems defined on sets of segments, and [10], [11] for possible

ISSN 1433-8092

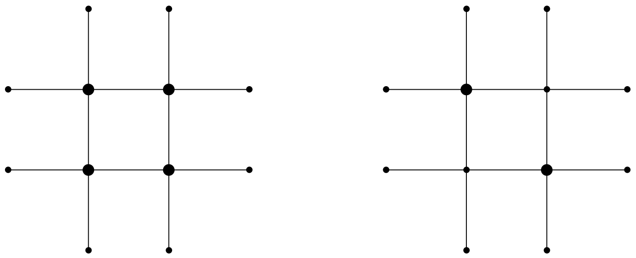

Fig. 1. *Left:* Any minimum vertex cover of the given plane graph requires four vertices. One such cover is marked by thick dots. *Right:* Two vertices can guard the same set of segments. One optimal solution is exhibited.

applications of related studies to efficient wireless communication. We consider two basic greedy-type algorithms for finding approximate solutions of GSS. We show that for each of these, theoretically the ratio of the approximate to the optimal solution can increase without bound with the increase of the number of segments. We also obtain other results about these algorirhms' performance.

The paper is organized as follows. The next section includes certain preliminaries (including notation) useful for understanding the rest of the paper. In Section 3 we present two approximation algorithms and study their theoretical performance. We find instances of GSS where each algorithm can perform as poorly as we choose. Section 4 details identities of a visibility function introduced in the next section as well as some theoretical results which were obtained using these identities. These results reveal how the number of segments intersecting at any given point affects the performance of the greedy algorithms. We conclude with some open roblems and final remarks in Section 5.

2 Preliminaries

2.1 Vertex Cover, Set Cover, and GSS

Given a universe set U and an arbitrary family of subsets $F \subseteq \mathcal{P}(U)$, the optimization *set cover* problem looks for a minimum cover $C \subseteq F$ so that $\bigcup_{s \in C} s = U$.

The optimization vertex cover problem has as an input a graph $G = (V, E)$, and one looks for a minimum set C of vertices of G, such that every edge of G is incident to at least one vertex of C. The set C is called a minimum *vertex cover* of G.

Now let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of segments in the plane. Denote by $\bar{S}$ the set of all points of segments in S and by V the set of all intersection points of segments of S. The elements of V will be called *vertices* of $\bar{S}$. If $A \subseteq V$, let S_A denote the set of segments which contain the vertices in A and $\bar{S}_A$ denote the set of points on segments in S_A. We consider as degenerate the case of intersecting collinear segments since it is trivial to discover such segments and merge them into one. In terms of the above notations, our problem can be formulated as follows.

Guarding a Set of Segments (GSS)
Find a minimum subset of vertices $\Gamma \subseteq V$ such that $\bar{S}_\Gamma = \bar{S}$.

In other words, one has to locate a minimum number of guards at the vertices of $\bar{S}$ so that every point of $\bar{S}$ *is seen* by at least one guard.

We will assume throughout that the set $\bar{S}$ is connected (otherwise the problem would just be a set of problems which do have connected segments). It is easy to see that the requirement to locate guards at vertices is not a restriction of the generality either: every non-vertex point on a segment s can see the points of s only, while each of the vertices on s can see s and other segments. Lastly, in the case where the input is a single segment assign one of the endpoints as a vertex.

In view of the above remarks, GSS admits formulation in terms of a set cover problem, as follows.

Set Cover Formulation of GSS
Let S be a set of segments and V the set of their intersections, called vertices *of S. Find a minimum sized subset of vertices $\Gamma \subseteq V$ such that $S_\Gamma = S$.*

2.2 Other Notations

For a universe set U and a subset S, by $|S|$ and S^c we denote the cardinality of S and its complement.

A *matching* in a graph is any set of edges without common vertices. A matching is *maximal* if it is not a proper subset of any other matching in the graph, and *maximum* if it has the largest possible number of edges. *Minimum maximal matching* is a maximal matching of minimum size.

Now let Π be a minimization problem with a set of instances D_Π. Given an instance $I \in D_\Pi$, let $Opt(I)$ be the value of an optimal solution for I. Given an approximation algorithm A for Π, let $A(I)$ be the value of the approximation solution to I found by A.

Define an *approximation ratio* of A on I as $R_A(I) = \frac{A(I)}{Opt(I)}$. Then $\frac{A(I)-Opt(I)}{Opt(I)} = R_A(I) - 1$ is the (*relative*) *approximation error* of A on I.

Now define the (*worst case*) *performance ratio* of A as $R_A = \inf\{r \geq 1 \ : \ R_A(I) \leq r \ for \ all \ I \in D_\Pi\}$.

Clearly, $R_A = 1$ for any algorithm A that always finds an optimal solution for an arbitrary instance $I \in D_\Pi$. Otherwise, the larger R_A, the poorer A's worst-case performance.

For more details the reader is referred to [3, 8].

3 Two Approximation Algorithms for GSS and their Worst-Case Performance

The approximation algorithms we consider can all be classified as greedy algorithms adapted from the related set cover and vertex cover problems.

G1 **Set Cover Greedy**
On each iteration, this algorithm simply chooses a vertex of greatest degree ($deg(v) = |S_{\{v\}}|$), removes it and all incident segments from the set of segments, and repeats until there are no longer any segments to cover.

G2 **Vertex Cover Matching Greedy**
The approximation algorithm for the vertex cover problem finds a maximal matching for the graph and selects both endpoints of each edge of the matching. Because in GSS there can be multiple intersections along a segment, we needed to make some changes before this algorithm can be correctly applied to GSS. For this algorithm, "both endpoints of each edge" was translated to "all useful intersections along each segment."
On each iteration, G2 chooses a segment with fewest intersections on it, adds intersections on the segment to the cover in order of greatest degree first, ignoring intersections that do not contribute any new segments to the cover. It then removes all incident segments from the GSS and repeats the previous steps of the algorithm until there are no longer any segments to cover. The segments chosen on each iteration form a *matching for the set of segments* in the sense that no two of them intersect.

In this sections we study the worst-case performance of G1 and G2. We have the following theorem.

Theorem 1. *The set cover greedy algorithm (G1) on GSS instances has a performance ratio that is* $\Omega(\log |S|)$*, where* $|S|$ *is the number of segments.*

Proof To show that the approximation error of algorithm G1 can be arbitrarily large with respect to the optimal solution, we provide a nontrivial extension to an approach to a known estimation of a greedy solution to the set cover problem (see, e.g., [8]).

We define the placement of line segments in the plane as follows. Fix an $m \in \mathbb{N}$ with $m > 1$, $a \in \mathbb{Q}$ with $a > 0$, and place the m points $a_1 = (1, a), (2, a), \ldots, (m, a) = a_m$. Let $n = \sum_{i=2}^{m} \left\lfloor \frac{m}{i} \right\rfloor$. Place the n points $b_{2,0} = (1, 0), (2, 0), \ldots, (n, 0) = b_{m,0}$ partitioned into groups of size $\left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{m}{3} \right\rfloor, \ldots, \left\lfloor \frac{m}{m-1} \right\rfloor, \left\lfloor \frac{m}{m} \right\rfloor$ where each partition is appended to the previous partition in the positive direction along the line $y = 0$. We index the elements $b_{i,j}$ with two subscripts, i and j, where i is the index of the partitions of size $\left\lfloor \frac{m}{i} \right\rfloor$, $(i = 2 \ldots m)$, and j is the index (starting from 0) within each partition $(j = 0 \ldots \left\lfloor \frac{m}{i} \right\rfloor - 1)$. The points $b_{i,j}$ are spaced

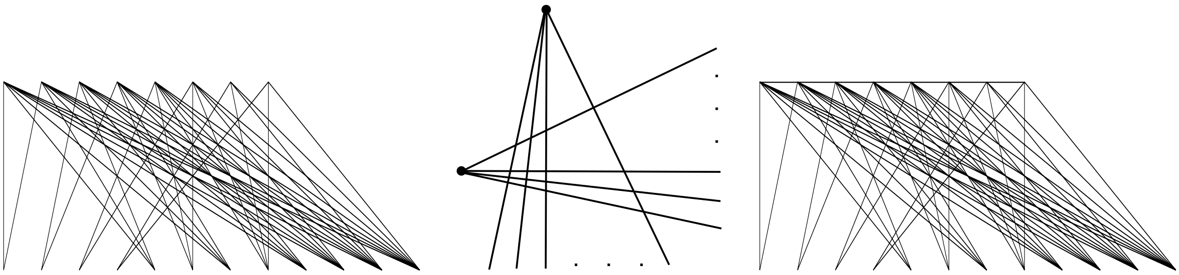

Fig. 2. *Left:* Illustration to the construction of Theorem 1 for $m = 8$. *Middle:* If each of the two sheaves of segments contains $n/2$ segments (for any even $n \geq 4$), then algorithm G2 finds an approximate solution of $n/2 + 1$ segments, while the optimal solution comprises the two vertices marked by thick dots. Note that on such sort of instances G1 finds the optimal solution. *Right:* Illustration to the proof of Proposition 1.

exactly 1 unit apart along the positive x-axis with $b_{2,0}$ having an x coordinate of 1.[1] We have one row of points on $y = a$ which is placed on integer coordinates in the following order:

$$a_1, a_2, \ldots, a_m$$

and one row on $y = 0$ which is placed on integer coordinates in the following order:

$$b_{2,0}, b_{2,1}, \ldots, b_{2,\lfloor \frac{m}{2} \rfloor - 1}, b_{3,0}, b_{3,1}, \ldots, b_{3,\lfloor \frac{m}{3} \rfloor - 1}, \ldots, b_{m-1,\lfloor \frac{m}{m-1} \rfloor - 1}, b_{m,0}$$

We will connect segments from points on $y = 0$ to points on $y = a$ in such a way that a solution is $A = \{a_1, \ldots, a_m\}$ while the set cover greedy algorithm always finds the solution $B = \{b_{2,0}, b_{2,2}, \ldots, b_{2,\lfloor \frac{m}{2} \rfloor - 1}, \ldots, b_{m-1,\lfloor \frac{m}{m-1} \rfloor - 1}, b_{m,0}\}$. Denoting the above constructed instance I, this will make the approximation ratio

$$R_{G1}(I) \geq \frac{n}{m} = \frac{\sum\limits_{i=2}^{m} \lfloor \frac{m}{i} \rfloor}{m} \approx \frac{m \log m}{m} = \log m = \Theta(\log |S|)$$

The final equality follows from the obvious fact that $m \log m \leq |S| \leq m^2 \log m$. To place the segments, connect $b_{i,j}$ to all points in the set $\{a_{ji+1}, a_{ji+2}, \ldots, a_{ji+i}\}$ forming i segments for each $b_{i,j}$. In this way, $b_{i,j}$ lies on exactly i segments. Figure 2 (left) illustrates the construction for $m = 8$. Since all segments in this construction have an a_i as an end point for some $1 \ 6 \ i \ 6 \ m$, $\{a_1, \ldots, a_m\}$ is a solution to GSS. It is not necessary to show that this is an optimal solution, as this shows that m is an upperbound for the number of guards in an optimal solution.

We will construct a set of disconnected segments. Adopt the convention of naming a segment by its end point with smaller y value first and end point with larger y value second. Then for each partition of B, order all segments connecting to points in this partition in lexicographic order with respect to x coordinate (i.e., $\overline{(x_1, y_1)(x_1', y_1')} < \overline{(x_2, y_2)(x_2', y_2')}$ iff $x_1 < x_2$ or $x_1 = x_2$, $x_1' < x_2'$); $\overline{pq}$ denotes the segment with endpoints $p, q \in \mathbb{R}^2$). Add to the set: the first segment in the first partition, second segment in the second partition, third segment in the third partition, and so on. We end up with the set of $m - 1$ segments $\{\overline{b_{2,0}a_1}, \overline{b_{3,0}a_2}, \overline{b_{4,0}a_3}, \ldots, \overline{b_{m,0}a_{m-1}}\}$. Since each segment is longer than the preceding one with the imposed ordering and none of the segments shares an end point with another, none of the segments intersect. It follows that at least $m - 1$ guards are required for a solution because there must be a guard on each of these segments to cover the entire construction. This is the best lower bound possible since an optimal

[1] While not essential to our proof, the reader may be interested to know how to compute the x coordinate from the subscripts i, j. The elements $b_{i,j}$ are sorted lexicographically by i first, then j. We assign to each $b_{i,j}$ a single index k and note that for every b_k, k is its x coordinate (where $k = 1 \ldots n$). For any $i > 2$, there are $\sum_{r=2}^{i-1} \lfloor \frac{m}{r} \rfloor$ points in the partitions to the left of partition i. When $i = 2$, this number is 0, so to generalize the previous summation for $i \geq 2$, we write it as $\sum_{r=1}^{i-1} \lfloor \frac{m}{r} \rfloor - m$. As previously stated, j is the 0-indexed position of $b_{i,j}$ within partition i, so the expression for k becomes $\sum_{r=1}^{i-1} \lfloor \frac{m}{r} \rfloor - m + j + 1$.

solution has $m-1$ guards for $m = 2$ and $m = 3$ (where the optimal solution is to place guards on all points in B rather than A).

Suppose we have two distinct segments, one containing b_{i,j_1} and the other containing b_{i,j_2}. It is easy to see that these two segments do not intersect unless $j_1 = j_2$ (in which case they intersect at b_{i,j_1}). In other words, no two segments extending from the same partition intersect each other at a point different from an end point they could share. As mentioned, in the case that $j_1 = j_2$, the two segments both contain b_{i,j_1} and this is their intersection. Without loss of generality, assume $j_1 < j_2$. We are interested in intersections with y coordinate $0 < y < a$. Then b_{i,j_1} is to the left of b_{i,j_2} on the line $y = 0$. All segments containing b_{i,j_1} connect to an endpoint to the left of or on $a_{(j_1+1)i}$ (on $y = a$) while all segments containing b_{i,j_2} connect to an end point to the right of $a_{j_2 i}$ which must be to the right of or equal to $a_{(j_1+1)i}$. It follows that no two distinct segments containing b_{i,j_1} and b_{i,j_2}, respectively, contain any intersections with y coordinate $y > 0$. Since there are a total of $m - 1$ partitions, it is clear that all unlabeled intersection points (i.e. intersection points with y coordinate $0 < y < a$) have degree at most $m - 1$.

For any $1 \leq l \leq m$ and any $2 \leq i \leq m$, $\{b_{i,0}, \ldots, b_{i,\lfloor \frac{m}{i} \rfloor - 1}\}$ contains at most one point which connects to a_l. In other words, each partition of the points on $y = 0$ contains at most one point which connects to a_l. So each a_l has degree at most the number of partitions $= m - 1$.

Now we can say that by construction, the only point of degree m is $b_{m,0}$. This is the point of highest degree, so the set cover greedy algorithm would place a guard here and remove all segments containing $b_{m,0}$. This deletes an entire partition.

Now there are only $m - 2$ partitions left and the unlabeled intersections (intersections which are not end points) as well as points on $y = a$ now take on this as the maximum degree because of the remarks above. The next partition of the points on $y = 0$ from the right are the only points of degree $m - 1$ so guards are placed at each of these and another partition is removed.

Continue in this way and all points labeled b will be chosen by the set cover greedy algorithm. □

It turns out that at worst, the performance of G2 can even be poorer than the one of G1. Figure 2 (middle) illustrates that a solution for a GSS problem on n segments found by G2 can be $(n/2 + 1)/2$ times the optimal, which is found by G1. The opposite can also be true: a solution found by G1 can be an unbounded number of times worse than the optimal one found by G2, if we modify G2 to select matching segments randomly instead of by least intersections. Note that this cannot be the case with the vertex cover problem, for which the performance of the matching greedy algorithm is always bounded by 2. Henceforth we refer to this modified version of G2 as RG2. More precisely, we have the following:

Proposition 1. *There is a class of GSS on $O(m^2 \log m)$ segments for which RG2 finds an optimal solution of m guards while G1 finds an approximate solution of $\Omega(m \log m)$ guards.*

Proof Consider the construction of Theorem 1 with the only difference that an additional segment connects all upper a-vertices (see Figure 2, right). This segment forms a maximal matching in S and assume that this is the one used by RG2. Then obviously RG2 takes as a solution the m points on that segment, which is the optimal solution.

By adding this segment, the degree of every a-vertex is increased by one, so both a- and b-vertices contain ones of maximal degree. Assume that G1 starts from the lower-rightmost vertex. Removing this vertex together with all adjacent segments, the degree of all maximal-degree a-vertices decrease by one. So, before the next iteration of G1, both upper and lower vertices contain vertices of maximal degree. Continuing the process, this pattern remains the same until the algorithm termination. Thus, G1's solution will be composed of all $m \log m$ lower points, and its performance ratio will be proportional to $\log m$. Since the construction and its analysis is transparent, details are omitted. □

4 Other Approximability Results

In this section we obtain some results that help to characterize the bounds of approximation for families of line segment sets. To this end, we first develop some mathematical tools that provide formalized set representation of line segments and their intersection points.

Suppose S is a set of segments in the plane (not necessarily connected) and V the set of intersections of segments in S (if a segment exists which intersects no other segments, then add one of its end points to V). If $A \subseteq V$, then $S_A = \{s \in S | a \text{ lies on } s \text{ for some } a \in A\}$.

Fact 1 $S_{A\cup B} = S_A \cup S_B$

Proof: $s \in S_{A\cup B} \iff s \in S_A$ or $s \in S_B \iff s \in S_A \cup S_B$.

Fact 2 $A \subseteq B \Rightarrow S_A \subseteq S_B$

Fact 3 $S_{A\cap B} \subseteq S_A \cap S_B$

Proof: $A \cap B \subseteq A$ and $A \cap B \subseteq B \Rightarrow S_{A\cap B} \subseteq S_A$ and $S_{A\cap B} \subseteq S_B \Rightarrow S_{A\cap B} \subseteq S_A \cap S_B$.

Fact 4 $S_A \backslash S_B \subseteq S_{A\backslash B}$

Proof: $s \in S_A \backslash S_B \Rightarrow s \in S_A, s \notin S_B \Rightarrow \exists a \in A, s \in S_{\{a\}}$ but $\forall b \in B, s \notin S_{\{b\}} \Rightarrow a \notin B \Rightarrow a \in A\backslash B \Rightarrow s \in S_{A\backslash B}$.

Fact 5 $(S_A)^c \subseteq S_{A^c}$

Fact 6 $S_{\{a\}} \subseteq S_{\{b\}} \Rightarrow a = b$

Proof: $S_{\{a\}} \subseteq S_{\{b\}} \Rightarrow$ at least 2 segments that a sees are seen by $b \Rightarrow$ they are both the intersection of a common pair of segments $\Rightarrow a = b$.

Fact 7 $S_A = S_{\{b\}} \Rightarrow A = \{b\}$

Proof: $A \neq \{b\} \Rightarrow \exists a \in A, a \neq b$ so that $S_{\{a\}} \subseteq S_{\{b\}} \Rightarrow a = b$.

Fact 8 $|A| \leq |S_A| \leq k|A|$ *where* $k = \sup\{|S_{\{a\}}| : a \in A\}$

Proof: $|A| \leq |S_A| \leq \sum\limits_{a\in A} |S_{\{a\}}| \leq \sum\limits_{a\in A} k \leq k|A|$.
The first inequality holds because of the $n-$gon.

Next we present three properties characteristic of many greedy-type approximation algorithms, show the relationships between them, and present some approximability results that can be derived for algorithms with such properties. We use the notation A to refer to a general algorithm for solving GSS, and σ^A for the subset of V that A outputs as a solution.

Property 1. An algorithm A selects vertices one at a time and will only add an intersection point to its solution set if it guards at least one segment not yet covered.

Property 2. For any subset $B \subseteq \sigma^A$, $|B| \leq |S_B|$.

Property 3. $\left|\sigma^A\right| \leq |S|$.

Proposition 2. *Property 1 implies Property 2. In other words, any algorithm A that exhibits Property 1 must also exhibits Property 2.*

Proof Let $\sigma^A = \{b_1, b_2, \ldots, b_m\}$, where $m = |\sigma^A|$ and the vertices b_i are listed in the order that A chose them (b_1 being the first point chosen). Let p_i be the number of new segments covered by b_i at the time it was chosen (i.e. the number of segments covered by b_i that are not covered by $\{b_1, \ldots, b_{i-1}\}$). By Property 1, $p_i \geq 1$. Let $B \subseteq \sigma^A = \{b_{i_1}, \ldots, b_{i_m}\}$, where the points b_{i_j} are listed by order of appearance in σ^A (i.e. the order that A selected them). Now, for each b_{i_j}, observe that since it covers p_{i_j} segments not covered by the set $\{b_{i_1}, \ldots, b_{i_j-1}\}$, it must cover at least p_{i_j} segments not covered by the set $\{b_{i_1}, \ldots, b_{i_{j-1}}\}$. Since each $p_{i_j} \geq 1$, $|B| = k \leq \sum_{j\in\{1,\ldots,k\}} p_{i_j} \leq |S_B|$, the total number of segments covered by B. □

Remark 1. Property 2 implies Property 3. This is trivial to show: simply let $B = \sigma^A$. We can chain these results together to show that Property 1 also implies Property 3. Thus, we have a hierarchy of properties, where Property 1 is the strongest since it guarantees the other two and Property 3 is the weakest and most general. Finally, we note that since the greedy algorithm G1 as previously described exhibits Property 1 it therefore has the other two as well.

The following result shows that for any approximation algorithm exhibiting Property 3, the performance ratio can be no greater than the maximum degree of any intersection point. For example, any greedy algorithm with Property 3 will have a performance ratio no worse than 2 on sets of line segments with only intersection points of degree 2 or less.

Proposition 3. *For an instance I of GSS let S be the set of segments, σ^A the solution found by an algorithm A, and σ^* an optimal solution. If the algorithm satifies Property 3, and $k = \sup\{\left|S_{\{v\}}\right| : v \in V\}$, then $R_A \leq k$.*

Proof Since placing a guard at a point removes at most k segments from the available segments to place guards on, we have $|\sigma^*| \geq \left\lceil \frac{|S|}{k} \right\rceil$. Then

$$|\sigma^A| \leq |S| \leq k \left\lceil \frac{|S|}{k} \right\rceil \leq k|\sigma^*|.$$

Hence $R_A = \frac{|\sigma^A|}{|\sigma^*|} \leq k$. This becomes a strict inequality if the set of segments is connected. □

The following theorem may be useful to provide an upper bound on the approximation ratio of G1.

Theorem 2. *For any instance I of GSS with segments S and intersections V, let $k = \sup\{|S_{\{v\}}| : v \in V\}$ be the maximum degree of any point in V and $l = |V_k|$ be the number of points with degree k.*

$$\text{If } l \leq k, \text{ then } R_{G1} \leq k - \frac{kl(2k-l+1)}{2\,|S|}$$

$$\text{If } l > k, \text{ then } R_{G1} \leq k - \frac{k^3-k^2}{2\,|S|}$$

Proof The beginning of the proof is identical for both cases. We use the same notation introduced in Proposition 2. As before, let $\sigma^{G1} = \{b_1, b_2, \ldots, b_m\}$, where $m = |\sigma^{G1}|$ and the vertices b_i are listed in the order that G1 chose them (b_1 being the first point chosen). Let p_i be the number of new segments covered by b_i at the time it was chosen (i.e. the number of segments covered by b_i that are not covered by $\{b_1, \ldots, b_{i-1}\}$). For brevity, in this proof the *degree* of a vertex will always refer to its degree in the original segment set S, and the *effective degree* will mean the updated degree (after segments have been covered and removed by G1) of the vertex at that particular point in the algorithm's operation. At each step, G1 always chooses a vertex with the largest effective degree in that iteration.

We know that $p_1 = k$ by the definition of G1, since k is the largest degree available. Assuming $2 \leq l$, next we claim that $p_2 \geq k-1$. Choosing the first point b_1 removes k segments from S, reducing the effective degree of any other vertices by at most 1 (because any two line segments can only intersect at one point, and that intersection is at b_1). The remaining degree-k vertices must therefore have effective degree of no less than $k-1$. Continuing this argument, if $i \leq l$ then we have $p_i \geq k-(i-1)$. In other words, when it is time to select a b_i, exactly $i-1$ points have been chosen before it. Therefore, all remaining vertices, and in particular the remaining degree-k vertices (since $i-1 < l$ we are sure at least one remains), have had their effective degrees reduced by at most $i-1$. Thus we can guarantee the existence of a vertex with effective degree at least $k-(i-1)$.

So, we obtain the following progression: $p_1 = k, p_2 \geq k-1, p_3 \geq k-2, \ldots, p_i \geq k-i+1$. If we sum the terms to obtain $k + (k-1) + \cdots + (k-i+1) = q \leq \sum_{j\in\{1,\ldots,i\}} p_j$, we can assert that $\left|S_{\{b_1,\ldots,b_i\}}\right| = \sum_{j\in\{1,\ldots,i\}} p_j \geq q$. Thus G1 uses i points to cover at least q segments. Since, as previously stated, $p_j \geq 1$ for all $b_j \in \sigma^{G1}$, to cover the remaining uncovered segments (i.e. $S \setminus S_{\{b_1,\ldots,b_i\}}$) we know that G1 chooses at most $\left|S \setminus S_{\{b_1,\ldots,b_i\}}\right| \leq |S| - q$ points . Therefore, $\left|\sigma^{G1}\right| \leq i + |S| - q$.

The details of where i ends and the subsequent value of q is where the argument diverges into two cases depending on whether l is greater than k or not. If $l \leq k$ then $i = l$. Then, the final term in the progression is $p_l \geq k-l+1$, so $q = k + (k-1) + \cdots + (k-l+1) = (k + (k-l+1))l/2$. Substituting values, and simplifying, $\left|\sigma^{G1}\right| \leq l + |S| - \frac{l(2k-l+1)}{2}$. Using the fact from the proof of Proposition 3 that $|\sigma^*| \geq \left\lceil \frac{|S|}{k} \right\rceil \geq \frac{|S|}{k}$, we obtain $R_{G1} = \frac{|\sigma^{G1}|}{|\sigma^*|} \leq |S|\frac{k}{|S|} - \frac{l(2k-l+1)}{2} \cdot \frac{k}{|S|} = k - \frac{kl(2k-l+1)}{2|S|}$.

If $l > k$, then the final term in the progression becomes $p_k \geq 1 = k - (k-1)$, where i ends at k. In this case, $q = k + (k-1) + \cdots + 1 = (k+1)k/2 = (k^2+k)/2$. After substituting and simplifying, we have $\left|\sigma^{G1}\right| \leq k + |S| - (k^2+k)/2 = |S| - \frac{k^2-k}{2}$. Finally, using the same argument for the first case, $R_{G1} = \frac{|\sigma^{G1}|}{|\sigma^*|} \leq |S|\frac{k}{|S|} - \frac{k^2-k}{2} \cdot \frac{k}{|S|} = k - \frac{k^3-k^2}{2|S|}$. □

Let W be a set of intersections and/or end points and $r \in \mathbb{N}$ be fixed. Denote

$$\begin{aligned} W_{r-} &= \{v \in W : |S_{\{v\}}| \leq r\}, \\ W_r &= \{v \in W : |S_{\{v\}}| = r\}, \\ W_{r+} &= \{v \in W : |S_{\{v\}}| > r\} \end{aligned}$$

The following theorem provides an a posteriori bound for the performance of G1.

Theorem 3. *Let V be the set of intersections in some connected set of segments S, g^1 the solution given by the set cover greedy algorithm G1, and σ^* an optimal solution for the GSS on S. If $k = \sup\{|S_{\{v\}}| : v \in V\}$, then*

$$R_{g^1} \leq \inf_{2 \leq r \leq k} \{r + \frac{|V_{r+}|}{|\sigma^*|}\}$$

Proof In each iteration, G1 chooses a point with the highest effective degree, and removes the segments it covers. Consider the iteration at which all points remaining have effective degree less than or equal to some integer r. Denote the set of points chosen by the greedy algorithm thus far as σ^{G1}_{r+}.

The line segments that G1 still needs to cover are $S \setminus S_{\sigma^{G1}_{r+}}$ but $S = S_{\sigma^*}$, so we can rewrite this as $S_{\sigma^*} \setminus S_{\sigma^{G1}_{r+}}$. By Fact 3 we have that $S_{\sigma^*} \setminus S_{\sigma^{G1}_{r+}} \subseteq S_{\sigma^* \setminus \sigma^{G1}_{r+}}$. Note that at the iteration where we interrupted G1 to obtain σ^{G1}_{r+}, no point in $\sigma^* \setminus \sigma^{G1}_{r+}$ can have an effective degree greater than r, by construction. It follows that each point in $\sigma^* \setminus \sigma^{G1}_{r+}$ can contribute no more than r line segments to the total count of uncovered segments at that iteration. This gives $|S_{\sigma^*} \setminus S_{\sigma^{G1}_{r+}}| \leq |S_{\sigma^* \setminus \sigma^{G1}_{r+}}| \leq r|\sigma^* \setminus \sigma^{G1}_{r+}| \leq r|\sigma^*|$. Since G1 has Property 3 and treats the remaining segments, $S_{\sigma *} \setminus S_{\sigma^{G1}_{r+}}$, as essentially a new GSS instance, we know that G1 will choose no more than $|S_{\sigma^*} \setminus S_{\sigma^{G1}_{r+}}|$ line segments. From this and the previous inequalities we have that the total number of points chosen by G1 after having chosen σ^{G1}_{r+} must be less than or equal to $r|\sigma^*|$.

Note that σ^{G1}_{r+} is clearly a subset of V_{r+} since the effective degree of a vertex at any point must always be less than or equal to its actual degree. So, we have $|\sigma^{G1}| \leq \sigma^{G1}_{r+} + r|\sigma^*| \leq |V_{r+}| + r|\sigma^*|$. Finally, $R_{G1} \leq \frac{|v_{r+}| + r|\sigma^*|}{|\sigma^*|} = r + \frac{|V_{r+}|}{|\sigma^*|}$. □

Remark 2. Consider, for example, a set of line segments such that two intersections have degree 5 and the rest have degree 2. Proposition 3 would give that the approximation ratio can be no worse than 5, which is true, but we can do better. Theorem 3 gives that a choice of r = 2 would leave 2 intersections points of degree greater than r, yielding an approximation ratio no worse than $2 + \frac{2}{|\sigma^*|}$. Since $|\sigma^*|$ is greater than 2, we have tightened the worst approximation ratio bound to 3.

Now let S be a set of segments such that no more than two segments intersect at any point. In other words, the maximum degree is 2. This property allows S to be tranformed into a graph $G(V, E)$ in the following manner. Let each segment in S become a vertex in V and every intersection of a pair of segments become an edge in E connecting the pair of vertices that correspond to the intersecting segments. Let $n = |V| = |S|$. Note the one-to-one correspondence between segments of S and the vertices in V of G, as well as the one-to-one correspondence between intersection points of S and edges in E of G. Further, a set of intersections is a minimum cover of S if and only if the corresponding set of edges is a minimum edge cover on G (an *edge cover* in a graph is a set of edges that are incident to all vertices). Note that since minimum edge cover is solvable in polynomial time, we can actually use this equivalence to solve this special case of GSS in polynomial time as well.

Denote by $\mathrm{M}_{\max}$ and $\mathrm{M}_{\min}$ a maximum matching and a minimum maximal matching of G, respectively. Then with the denotations of Theorem 3 we can state the following lemma.

Lemma 1. (a) $2\left|M_{\max}\right| \leq n$ (b) $\left|M_{\max}\right| \leq 2\left|M_{\min}\right|$

Proof For (a), note that each edge of $|M_{\max}|$ is incident to two vertices, so the inequality is immediately apparent.

For (b), first note that any edge in E must be incident to an edge in $M_{\max}$ by the definition of maximal matching. For every $e_{\max} \in M_{\max}$, let $E_{e_{\max}} := \{e_{\min} \in M_{\min} | e_{min}$ is incident to $e_{\max}\}$. Then $\left|E_{e_{\max}}\right| \leq 2$ since no more than one edge from $M_{\min}$ may be incident with each vertex of $e_{\max}$. This implies that $\left|M_{\max}\right| \leq 2\left|M_{\min}\right|$, which is the desired result.

□

Now we can obtain the following result.

Theorem 4. *Greedy algorithm G1 has performance ratio of no more than $\frac{3}{2}$ for the special case of GSS instances in which the highest degree intersection is 2.*

Proof Since the largest intersection degree is 2, the greedy algorithm G1 randomly selects degree 2 intersections and removes the covered segments on each iteration. This continues until no degree 2 intersections are left. Each chosen intersection corresponds to an edge in E, and we claim this set of edges is a maximal matching on G. Note that at this point in the algorithm's execution, each segment contains at most one chosen intersection, which for the graph G means that each vertex is incident to at most one chosen edge, i.e. that no two chosen edges are adjacent (share a vertex). This is precisely the definition of a matching. To show that it is maximal, observe that since no degree two intersections remain in the set of segments that have yet to be uncovered, the graph contains no edge connecting two vertices that are still uncovered (not incident to any chosen edge). Thus, there are no edges that can be added to the matching without destroying the matching property.

Let M be such a maximal matching in G constructed by G1 operating on corresponding GSS instance. There are $n-2|M|$ remaining uncovered line segments in S, and one point is needed to cover each of them since they do not intersect. Restating the previous sentence in terms of the graph G, there are $n-2|M|$ remaining uncovered vertices in V, and one edge is needed to cover each of them since no pair of them is adjacent. So, the solution size is $|M|+n-2|M| = n-|M|$. Clearly, $\left|g^1\right|$ is maximized when $|M|$ is minimized. This is the case when M is a minimum maximal matching, i.e., $|M| = |M_{\min}|$.

Thus $\left|g^1\right| \leq n - |M_{\min}|$.

It is well known that on any graph G there exists a minimum edge cover that consists of a maximum maximal matching plus one additional edge for each uncovered vertex (see, e.g., [6], [8]). As stated previously, a minimum edge cover on G is an optimum solution for the corresponding GSS. Thus, we obtain the following result regarding the size of the optimal solution: $|\sigma^*| = |M_{\max}| + n - 2\left|M_{\max}\right| = n - |M_{\max}|$.

Lemma 1 (a) and (b) state that $2\left|M_{\max}\right| \leq n$ and $\left|M_{\max}\right| \leq 2\left|M_{\min}\right|$. Combining these inequalities with some simple algebraic transformations, we consecutively obtain

$$\begin{aligned}
2\left|M_{\max}\right| + \left|M_{\max}\right| &\leq n + 2\left|M_{\min}\right| \\
-2\left|M_{\min}\right| &\leq n - 3\left|M_{\max}\right| \\
2n - 2\left|M_{\min}\right| &\leq 3n - 3\left|M_{\max}\right| \\
2(n - |M_{\min}|) &\leq 3(n - |M_{\max}|) \\
2\left|g^1\right| &\leq 3\left|\sigma^*\right| \\
\frac{\left|g^1\right|}{\left|\sigma^*\right|} &\leq \frac{3}{2}
\end{aligned}$$

Since this is true for any GSS instance in which the maximum intersection degree is 2, we have shown that $\frac{3}{2}$ is an upper bound on the performance ratio of G1 on this special case of GSS. □

As a final note, we remark that $\frac{3}{2}$ is a supremum as well as an upper bound on the performance ratio of G1 in this special case. It is easy to find examples for which G1 attains an approximation ratio of $\frac{3}{2}$.

5 Open Problems and Concluding Remarks

In this section we propose a few open problems whose resolution would help to characterize the relation between GSS and other covering problems.

We have previously shown that planar vertex cover for graphs of degree at most 3 can be reduced to GSS in polynomial time. In this sense, 3PVC is a subset of GSS and our belief is that the same can be said for planar vertex cover in general.

Problem 1: *Give a polynomial time reduction from planar vertex cover to GSS.*

As previously mentioned, GSS appears to be "sandwiched" in between 3PVC and SC. Since 3PVC is constant approximable but SC is only logarithmically approximable it is of interest to determine which GSS is. Doing so would help to clarify why 3PVC can be approximated well or why SC can only be approximated poorly. We have shown that some greedy type approximation algorithms do not perform with constant approximability, yet they still did well in practice. For this reason it is difficult to conjecture whether or not GSS admits a constant approximation.

Problem 2: *Either prove that it is "hard" to obtain a better than logarithmic approximation factor for GSS OR provide a polynomial time algorithm with a better approximation factor. For the latter case, find the best possible approximation factor achievable by a polynomial time algorithm (and provide such an algorithm).*[2]

The above list of open problems is by no means comprehensive and there is certainly opportunity for additional theoretical exploration. The study of GSS is an interesting combination of combinatorial optimization and computational geometry that offers potential to illuminate problems in both.

Acknowledgements

This work was supported by NSF grants No 0802964 and No 0802994.

References

1. Bose, P., D. Kirkpatrick, Z. Li, Worst-case-optimal algorithm for guarding planar graphs and polyhedral surfaces, *Computational Geometry: Theory and Applications* **26**(3) (2003) 209–219
2. Brimkov, V.E., A. Leach, M. Mastroianni, J. Wu, Guarding a set of line segments in the plane, *Theoretical Computer Science*, in press (DOI 10.1016/j.tcs.2010.08.014)
3. Garey, M. and Johnson, D., *Computers and Intractability*, W.H. Freeman & Company, San Francisco, 1979
4. Karp, R., Reducibility among combinatorial problems, in R.E. Miller and J.W. Thatcher (eds.), *Complexity of Computer Computation*, Plenum Press, New York, 85–103, 1972
5. Ǩaucic, B., B. Žalik, A new approach for vertex guarding of planar graphs, *J. of Computing and Information Technology* **10**(3) (2002) 189–194
6. Norman, R.Z., M.O. Rubin, An algorithm for minimum cover of a graph, *Proc. American Math Society* **10** (1959) 315–319
7. O'Rourke, J., *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987
8. Papadimitriou, Ch., K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, New Jersey, 1982
9. Urrutia, J., Art Gallery and Illumination Problems, Ch. 22 in J.-R. Sack, J. Urrutia (eds.), *Handbook of Computational Geometry*, North Holland, Amsterdam, 2000
10. Fabila-Monroy, R., A. Ruis Vargas, J. Urrutia, On Modem Illumination Problems, XIII Encuentros de Geometria Computacional, Zaragoza, Spain, 2009, $http\ : //www.matem.unam.mx/\ urrutia/online_papers/Modems.pdf$
11. Aichholzer, O., R. Fabila-Monroy, D. Flores-Pealoza, T. Hackl, C. Huemer, J. Urrutia, B. Vogtenhuber, Modem Illumination of Monotone Polygons, In Proc. 25th European Workshop on Computational Geometry EuroCG '09, Brussels, Belgium, 2009, pp. 167–170

[2] The authors have considered some approaches such as those well-known of Lund & Yannakakis and Feige for proving that it is hard to approximate within a logarithmic factor, but to no avail. Simply extending results for Set Cover is not likely to be effective as the geometric construction of GSS places a challenging restriction on the families of subsets that can be created from the universe set.