Optimization and approximation on systems of geometric objects

van Leeuwen, E.J.

Chapter 9

Geometric Set Cover and Unit Squares

Geometric Set Cover can be approximated better than general Minimum Set Cover, but for many object types the approximability has not been settled yet. We give a ptas for Geometric Set Cover on unit squares, improving on the earlier 2-approximation algorithm [209]. This is the one of the first approximation schemes for Geometric Set Cover on two-dimensional objects (together with the recently appeared [217]) and the first that extends to the weighted case. The scheme in fact extends to the more general budgeted maximum coverage problem. Moreover, we prove that the scheme essentially has optimal running time (up to constants), unless the exponential time hypothesis is false.

Besides these positive algorithmic results, we also give several negative results. We show that on convex polygons, translated copies of a single polygon, rotated copies of a single convex polygon, and α-fat objects, Geometric Set Cover is as hard as Minimum Set Cover. These hardness results all carry over (mutatis mutandis) to the budgeted case. If the polygons have constant description complexity, Geometric Set Cover is still APX-hard on convex polygons. We also obtain APX-hardness results for Geometric Set Cover on axis-parallel rectangles and ellipses.

9.1 A ptas on Unit Squares

We consider Geometric Set Cover on unit squares and show that it has a ptas by applying the shifting technique.

So let $\mathcal{P}$ be a set of points and $\mathcal{S}$ a set of axis-aligned unit squares. For sake of notation, when referring to the (x, y)-coordinates of a square, we mean the coordinates of the bottom left corner of that square. For a square s, the x-coordinate of (the bottom left corner of) s is denoted by $x(s)$, while the y-coordinate is denoted by $y(s)$. By scaling and translating (as in Chapter 4), we can assume that no horizontal (vertical) boundary of a square is on the same line as the horizontal (vertical) boundary of another square. Furthermore, we can assume that all points are fully contained in the squares they are in, i.e. no point lies on the boundary of a square. Finally, we assume that none of the square or point coordinates are integers.

Consider the horizontal lines $y = h$ ($h \in \mathbb{Z}$). They partition the plane into horizontal slabs of height 1. Any point is contained in a slab and every square intersects precisely one line. Let $k \geq 1$ be some integer we determine later. For any k consecutive slabs, the points in these slabs must be covered by a subset of the squares intersecting the $k + 1$ horizontal lines defining those k slabs. Using the shifting technique, it suffices to prove that we can optimally solve Geometric Set Cover on unit squares if we restrict to k consecutive slabs and the $k + 1$ lines defining them.

Theorem 9.1.1 *For any instance of Geometric Set Cover on a set of unit squares $\mathcal{S}$ where all points of $\mathcal{P}$ are inside $k \geq 1$ consecutive height 1 horizontal slabs, one can find an optimal solution in $\mathrm{O}((3|\mathcal{S}|)^{4k+4}\,|\mathcal{P}|)$ time.*

The next few pages are devoted to proving this theorem.

The idea will be to apply a sweep-line algorithm. This requires that we somehow bound the number of squares of an optimal solution that intersect the sweep-line at a given sweep-line position. To this end, consider the subset of squares of an optimum solution intersecting a horizontal line $y = h$ for some $h \in \mathbb{Z}$. Any such square must appear on the lower or upper envelope of this subset, or all points it covers would be covered by other squares. This is because the union of a set of connected axis-aligned squares intersecting a common axis-parallel line cannot have any holes. Following this observation, for each position of the sweep-line and for each of the $k + 1$ integer horizontal lines, we should consider at most two squares intersecting the sweep-line: one that will appear on the upper envelope and one that will appear on the lower envelope of the final solution. Obviously, we could also have a single square both on the upper and lower envelope, or, in fact, have no square at all.

Although at first glance this approach seems feasible, there is a problem with the dynamic programming. A square might appear on the lower envelope for some position of the sweep-line and on the upper envelope for a later position. In fact, several other squares might appear on the lower or upper envelope before this square appears on the upper envelope. This makes it difficult to avoid counting certain squares twice. To circumvent this, we split the sweep-line into k parts, one part per slab. We move these parts at different speeds, but always in such a way that if a square appears both on the lower and the upper envelope, then the split sweep-line is positioned such that it intersects the square both at the point where the square appears on the lower and on the upper envelope.

Though intuitively it seems like this would work, the split sweep-line trick requires a rigorous proof. We do this by formalizing the sweep-line process.

Just as in any sweep-line algorithm, we maintain a data structure (the *front*) containing the squares that are 'active' at a given position of the sweep-lines. The difficulty in this sweep-line algorithm arises in maintaining the front and consequently in finding squares that can be validly inserted into the front. Therefore we start by defining the front that we use and the (four types of) insertions that we are allowed to perform.

Let $\mathcal{S}^l$ and $\mathcal{S}^r$ be two dummy sets of $k+1$ squares each, such that the squares in $\mathcal{S}^l$ ($\mathcal{S}^r$) are to the left (right) of all squares in $\mathcal{S}$ and each integer horizontal line intersects precisely one square of $\mathcal{S}^l$ and one square of $\mathcal{S}^r$. Let $\overline{\mathcal{S}} = \mathcal{S} \cup \mathcal{S}^l \cup \mathcal{S}^r$. Given some set $S \subseteq \overline{\mathcal{S}}$, let S_i denote the set of squares in S intersecting line i. Let $R_i \subseteq S_i$ be the set containing precisely:

- the rightmost square of S_i (denote it by s_i),
- those squares s that overlap part of the left boundary of s_i and whose right boundary is not fully covered by squares of S_i.

We now define a front. For a better understanding of the definition, imagine that the squares are being inserted in order of increasing x-coordinate and that we want to keep track of the upper and lower envelope of each line i.

Definition 9.1.2 *Let S be the union of $\mathcal{S}^l$ and some subset of $\overline{\mathcal{S}}$. Then a* front $F = \{u_1, \ldots, u_{k+1}, l_1, \ldots, l_{k+1}, b_1, \ldots, b_{k+1}, x_1, \ldots, x_k\}$ *for S has the following properties:*

- $u_i, l_i \in R_i$ *with* $u_i = s_i$ *or* $l_i = s_i$*,*
- $y(s) \leq y(u_i)$ *for any* $s \in S_i$ *to the right of* u_i *(i.e. with* $x(s) > x(u_i)$*),*
- $y(s) \geq y(l_i)$ *for any* $s \in S_i$ *to the right of* l_i *(i.e. with* $x(s) > x(l_i)$*),*
- b_i *is equal to:*
  - *the lowest square of* S_i *to the right of* l_i *if* $x(u_i) > x(l_i)$*,*
  - *the highest square of* S_i *to the right of* u_i *if* $x(l_i) > x(u_i)$*,*
  - s_i *if* $x(u_i) = x(l_i)$ *(i.e. if* $u_i = l_i$*),*
- x_i *is equal to the larger of the x-coordinate from which* l_{i+1} *starts appearing on the lower envelope of* S_{i+1} *and the x-coordinate from which* u_i *starts appearing on the upper envelope of* S_i*.*

An example is depicted in Figure 9.1.

Fronts are the representative of the current state of the sweep-line algorithm. The squares u_i and l_i track the current square on respectively the upper and the lower envelope of line i. The value of x_i is the x-coordinate of the part of the sweep-line between lines i and $i+1$. The square b_i is used in checking if a certain square may be inserted or not.

We can make two observations about a front. First, $y(u_i) \geq y(l_i)$ and, since $u_i, l_i \in R_i$, $|x(u_i) - x(l_i)| < 1$ for any $i = 1, \ldots, k+1$. Secondly, if $x(u_i) \geq x(l_i)$, then $y(u_i) \leq y(b_i) \leq y(l_i)$. If $x(u_i) \leq x(l_i)$, then $y(l_i) \leq y(b_i) \leq y(u_i)$.

For a given front, we distinguish four types of insertions that are possible. An upper-insertion for squares that will appear only on the upper envelope for some line, a lower-insertion for squares appearing only on the lower envelope,

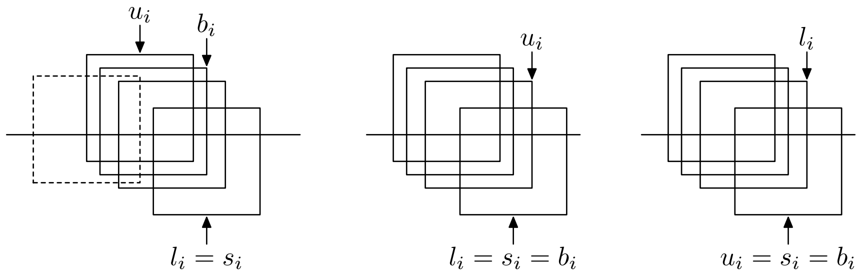

Figure 9.1: The left figure shows a set S_i. The dashed square is not in R_i and thus not in a front for S_i. The four solid squares are in R_i. From Definition 9.1.2, the labeling in the figure is correct. The middle figure shows the same set R_i, but with a different (and still correct) labeling. The labeling in the right figure however is incorrect.

and a middle-insertion or a skip-insertion for squares appearing on both envelopes. We define these four insertions, describe when they may be applied, and prove that any geometric set cover can be obtained using these insertions.

From now on, S will denote the union of $\mathcal{S}^l$ and some subset of $\overline{\mathcal{S}}$.

Definition 9.1.3 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k\}$. We say that s is* upper-insertable *into F if all of the following hold:*

1. *$y(s) > y(l_i)$ and if $x(l_i) > x(u_i)$, then $y(s) > y(b_i)$,*
2. *$x(s) \in (x(l_i), x(l_i) + 1]$ and $x(s) \in (x(u_i), x(u_i) + 1]$,*
3. *$x'_i > x_i$,*
4. *any point of $\mathcal{P}$ in $[x_i, x'_i] \times [i, i + 1]$ is covered by u_i or l_{i+1},*

where x'_i is the x-coordinate from which s is on the upper envelope of $(S \cup \{s\})_i$.

Condition 1 ensures that s lies above l_i and all squares between u_i and l_i (represented by b_i), Condition 2 ensures that s appears on the upper envelope of $(S \cup \{s\})_i$, Condition 3 ensures that this appearance happens after u_i appears on the upper envelope, and Condition 4 ensures that we cover all points between two consecutive sweep-line positions. An example of upper-insertable squares and squares that are not upper-insertable is given in Figure 9.2.

Proposition 9.1.4 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k\}$. If Condition 2 of Definition 9.1.3 holds for F and s, then s appears on the upper envelope of $(S \cup \{s\})_i$ to the right of u_i.*

Proof: By Condition 2, $x(s) > \max\{x(u_i), x(l_i)\} = x(s_i)$, and thus s appears on the upper envelope of $(S \cup \{s\})_i$ to the right of u_i. □

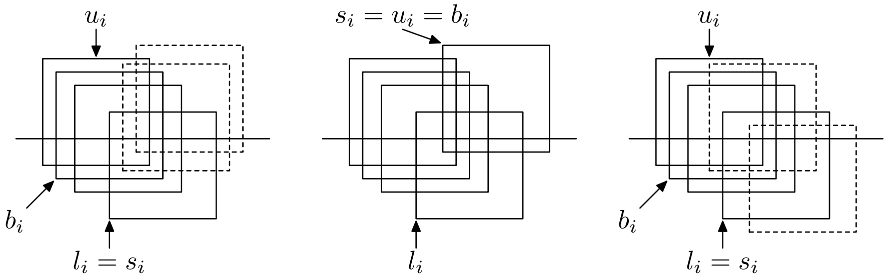

Figure 9.2: The left figure shows two (dashed) squares that are upper-insertable into the front of Figure 9.1. The middle figure shows the resulting front after upper-inserting the rightmost of these squares. The right figure shows two (dashed) squares that are not upper-insertable.

As a consequence of this proposition, the x-coordinate x'_i of Definition 9.1.3 does indeed exist (if Condition 2 holds).

Lemma 9.1.5 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k\}$ that is upper-insertable into F. Then between the appearance of u_i and the appearance of s on the upper envelope of $(S \cup \{s\})_i$ no other squares appear on the upper envelope of $(S \cup \{s\})_i$.*

Proof: If $u_i = s_i$, this follows from $x(s) > x(u_i) = x(s_i)$ and $x'_i > x_i$. So assume that $u_i \neq s_i$. Then $l_i = s_i$ and $x(l_i) > x(u_i)$. Recall the definition of a front and observe that b_i is the highest square of S_i to the right of u_i. As $x(l_i) - x(u_i) < 1$ and $y(b_i) < y(u_i)$, it suffices for s to lie above b_i (i.e. $y(s) > y(b_i)$) and for s to cover the x-range $[x(u_i)+1, x(l_i)+1]$ (i.e. $x(l_i) < x(s) < x(u_i)+1$). This holds from the definition of upper-insertable. □

Lemma 9.1.6 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k\}$ that is upper-insertable into F. Then $S \cup \{s\}$ has a front F' equal to F, except u_i is replaced by s, x_i is set to x'_i, where x'_i is equal to $x(s)$ if $y(s) > y(u_i)$ and to $x(u_i) + 1$ otherwise, and if $x(u_i) \leq x(l_i)$ or $y(s) \leq y(b_i)$, b_i is set to s.*

Proof: Since $x(s) > \max\{x(u_i), x(l_i)\} = x(s_i)$ by Condition 2 of Definition 9.1.3, we can replace u_i by s. Note that l_i can remain the same by Condition 1 and 2. By Lemma 9.1.5, x'_i is indeed the x-coordinate from which s appears on the upper envelope of $(S \cup \{s\})_i$. From Condition 3, x_i should be set to x'_i. If $x(u_i) \leq x(l_i)$, then as $x(s) > x(l_i)$, b_i should be set to s. If $x(u_i) > x(l_i)$, then b_i should only be changed if s lies below b_i, i.e. if $y(s) \leq y(b_i)$. Then the front F' is indeed a front for $S \cup \{s\}$. □

Constructing the front F' from F as prescribed in the lemma statement is called the *upper-insertion* of s into F.

Definition 9.1.7 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{2, \ldots, k+1\}$. We say that s is* lower-insertable *into F if all of the following hold:*

1. *$y(s) < y(u_i)$ and if $x(u_i) > x(l_i)$, then $y(s) < y(b_i)$,*
2. *$x(s) \in (x(l_i), x(l_i)+1]$ and $x(s) \in (x(u_i), x(u_i)+1]$,*
3. *$x'_{i-1} > x_{i-1}$,*
4. *any point of $\mathcal{P}$ in $[x_{i-1}, x'_{i-1}] \times [i-1, i]$ is covered by u_{i-1} or l_i.*

Here x'_{i-1} is the x-coordinate by which s is on the lower envelope of $(S \cup \{s\})_i$.

Lemma 9.1.8 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{2, \ldots, k+1\}$ that is lower-insertable into F. Then $S \cup \{s\}$ has a front F' equal to F, except l_i is replaced by s, x_{i-1} is set to x'_{i-1}, where x'_{i-1} is equal to $x(s)$ if $y(s) < y(l_i)$ and to $x(l_i)+1$ otherwise, and if $x(l_i) \leq x(u_i)$ or $y(s) \geq y(b_i)$, b_i is set to s. Furthermore, between the appearance of l_i and the appearance of s on the lower envelope of $(S \cup \{s\})_i$ no other squares appear on the lower envelope of $(S \cup \{s\})_i$.*

Constructing the front F' from F as prescribed in the lemma statement is called the *lower-insertion* of s into F.

We define middle-insertable, which combines upper- and lower-insertable, except that we drop the constraints that $y(s) > y(l_i)$ and $y(s) < y(u_i)$.

Definition 9.1.9 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k+1\}$. We say that s is* middle-insertable *into F if all of the following hold:*

1. *if $x(l_i) > x(u_i)$, then $y(s) > y(b_i)$, and if $x(u_i) > x(l_i)$, then $y(s) < y(b_i)$,*
2. *$x(s) \in (x(l_i), x(l_i)+1]$ and $x(s) \in (x(u_i), x(u_i)+1]$,*
3. *$x'_i > x_i$ (if $i \neq k+1$) and $x'_{i-1} > x_{i-1}$ (if $i \neq 1$),*
4. *any point of $\mathcal{P}$ in $[x_i, x'_i] \times [i, i+1]$ (if $i \neq k+1$) or $[x_{i-1}, x'_{i-1}] \times [i-1, i]$ (if $i \neq 1$) is covered by u_{i-1}, l_i, u_i, or l_{i+1},*

where x'_i (x'_{i-1}) is the x-coordinate from which s appears on the upper (lower) envelope of $(S \cup \{s\})_i$.

Lemma 9.1.10 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k+1\}$ that is middle-insertable into F. Then $S \cup \{s\}$ has a front F' equal to F, except u_i, l_i, and b_i are replaced by s, x_i is set to x'_i (if $i \neq k+1$), where x'_i is equal to $x(s)$ if $y(s) > y(u_i)$ and to $x(u_i)+1$ otherwise, and x_{i-1} is set to x'_{i-1} (if $i \neq 1$), where x'_{i-1} is equal to $x(s)$ if $y(s) < y(l_i)$ and to $x(l_i)+1$ otherwise. Furthermore, between the appearance of u_i (l_i) and the appearance of s on the upper (lower) envelope of $(S \cup \{s\})_i$ no other squares appear on the upper (lower) envelope of $(S \cup \{s\})_i$.*

Constructing the front F' from F as prescribed in the lemma statement is called the *middle-insertion* of s into F.

Definition 9.1.11 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k+1\}$. We say that s is* skip-insertable *into F if all of the following hold:*

1. $u_i = l_i$,
2. $x(s) > 1 + \max\{x(u_i), x(l_i)\}$,
3. *(if $i \neq k+1$) $x(s) > x_i$ and (if $i \neq 1$) $x(s) > x_{i-1}$*
4. *any point of $\mathcal{P}$ in $[x_i, x(s)] \times [i, i+1]$ (if $i \neq k+1$) or in $[x_{i-1}, x(s)] \times [i-1, i]$ (if $i \neq 1$) is covered by u_{i-1}, l_i, u_i, or l_{i+1}.*

Lemma 9.1.12 *Let F be a front for some S and let $s \notin S$ be a square intersecting line $i \in \{1, \ldots, k+1\}$ that is skip-insertable into F. Then $S \cup \{s\}$ has a front F' equal to F, except u_i, l_i, and b_i are replaced by s, x_i is set to $x(s)$ (if $i \neq k+1$), and x_{i-1} is set to $x(s)$ (if $i \neq 1$). Furthermore, between the appearance of u_i (l_i) and the appearance of s on the upper (lower) envelope of $(S \cup \{s\})_i$ no other squares appear on the upper (lower) envelope of $(S \cup \{s\})_i$.*

Constructing the front F' from F as prescribed in the lemma statement is called the *skip-insertion* of s into F.

In general, we call an upper-/lower-/middle-/skip-insertion an *insertion* and we say s is *insertable* if it is upper-/lower-/middle-/skip-insertable. A *valid insertion* is the upper- (respectively lower-/middle-/skip-) insertion of a square that is upper- (respectively lower-/middle-/skip-) insertable.

We now prove that any set cover can be obtained using a sequence of valid insertions. Denote by F^l and F^r the fronts for $\mathcal{S}^l$ and $\overline{\mathcal{S}}$.

Lemma 9.1.13 *Assume $\mathcal{P} = \emptyset$. Let S be some set such that $S = \mathcal{S}^l \cup S_i \cup \mathcal{S}^r$ for some $i \in \{1, \ldots, k+1\}$ and any square in S_i appears on the lower or the upper envelope of S_i. Then there is a sequence of $|S_i| + k - 1$ valid insertions starting from F^l, leading to fronts $F^l = F_0, F_1, \ldots, F_{|S_i|+k-1} = F^r$ such that for any square $s \in S_i$, there is a front F_j containing s.*

Proof: We assume that if $i = 1$, then no squares of S_i appear only on the lower envelope of S_i. Similarly, if $i = k+1$, assume that no squares of S_i appear only on the upper envelope of S_i. Order the squares in $S_i \backslash \mathcal{S}^l$ by increasing x-coordinate, i.e. $s_1, \ldots, s_{|S_i|-1}$. Note that the squares appearing on the upper envelope form an increasing subsequence of S_i. Similarly, the squares appearing on the lower envelope form an increasing subsequence. We claim that one can obtain the requested sequence of valid insertions by inserting s_j into F_{j-1} for all $j = 1, \ldots, |S_i| - 1$ as follows: if s_j appears

- only on the upper envelope of S_i, then s_j is upper-insertable and will be upper-inserted;
- only on the lower envelope of S_i, then s_j is lower-insertable and will be lower-inserted;
- on the upper and lower envelope of S_i and a square of S_i covers part of its left boundary, then s_j is middle-insertable and will be middle-inserted;
- on the upper and lower envelope of S_i and no square of S_i covers part of its left boundary, then s_j is skip-insertable and will be skip-inserted.

We prove this by induction on the number j of inserted squares.

Suppose that $j = 0$. Since s_1 is the leftmost square of $S_i \backslash \mathcal{S}^l$, it appears on both envelopes of S_i and no square of S_i covers part of its left boundary. By the definition of $F^l = F_0$, s_1 is skip-insertable into F_0 and can be skip-inserted.

Assume that $j > 0$ and consider the current front F_j. If s_{j+1} appears on both envelopes of S_i and no square of S_i covers part of its left boundary, then $x(s_{j+1}) > 1 + x(s_{j'})$ for any $j' < j + 1$. As squares are inserted in order of increasing x-coordinate, in F_j, $x(s_{j+1}) > 1 + \max\{x(u_i), x(l_i)\}$. Since $S = \mathcal{S}^l \cup S_i \cup \mathcal{S}^r$, this implies that (if $i \neq k+1$) $x(s_{j+1}) > x_i$ and (if $i \neq 1$) $x(s_{j+1}) > x_{i-1}$. Finally, observe that s_j must appear on both envelopes of S_i and thus must have been middle- or skip-inserted. But then $u_i = l_i = s_j$. Hence s_{j+1} is skip-insertable into F_j.

If s_{j+1} appears only on the upper envelope of S_i, then there must be squares appearing on the lower envelope of S_i covering the bottom left corner of s_{j+1}. By induction, the rightmost such square must be l_i. Hence $y(s_{j+1}) > y(l_i)$ and $x(s_{j+1}) \in (x(l_i), 1 + x(l_i)]$. But then there are squares covering (part of) the left boundary of s_{j+1} appearing on the upper envelope of S_i. By induction, the right-most such square must be u_i and thus $x(s_{j+1}) \in (x(u_i), 1 + x(u_i)]$. As squares are inserted in order of increasing x-coordinate, s_{j+1} appears on the upper envelope of S_i after u_i. Then $x'_i > x_i$. Finally, suppose that $x(l_i) > x(u_i)$. By induction, any square $s \in S_i$ with $x(u_i) < x(s) \leq x(l_i)$, and in particular b_i, does not appear on the upper envelope of S_i. Therefore $y(s_{j+1}) > y(b_i)$. Hence s_{j+1} is upper-insertable into F_j.

The cases when s_{j+1} appears only on the lower envelope of S_i or when s_{j+1} appears on both envelopes of S_i and (part of) its left boundary is covered are similar. Finally, apply skip-insertions to insert the squares of $\mathcal{S}^r$. □

Lemma 9.1.14 *Assume $\mathcal{P} = \emptyset$. Let S be some subset of $\overline{\mathcal{S}}$ containing $\mathcal{S}^l \cup \mathcal{S}^r$, such that for the set S_i of squares in S intersecting line i for $i \in \{1, \ldots, k+1\}$, any square in S_i appears on the upper or lower envelope of S_i. Then there is a sequence of $|S| - k - 1$ valid insertions starting from $F_0 = F^l$, leading to $F_1, \ldots, F_{|S|-k-1} = F^r$ such that for any square $s \in S$, there is a front F_j containing s.*

Proof: Following the proof of the previous lemma, we can insert the squares intersecting each horizontal line in order of increasing x-coordinate. However, we should interleave the sequences of the different lines. For any $i = 1, \ldots, k$, consider the squares appearing on the upper envelope of S_i and the lower envelope of S_{i+1}. Order these squares according to the x-coordinate from which they appear on the upper envelope of S_i or on the lower envelope of S_{i+1} respectively. Combining these two orders, we can extend this to an order by which to insert the squares of S. We claim that the j-th square s_j according to this order is insertable into F_{j-1} and that after inserting s_j, all squares $s_{j'}$ with $j' > j$ are still insertable.

We prove this by induction on the number j of inserted squares. Trivially, all squares are insertable into F_0. Now consider any $j \geq 0$. By induction, s_{j+1} is insertable into F_j. Let F_{j+1} be the front arising from the insertion of s_{j+1} into F_j. Suppose that $s_{j+1} \in S_i$ for some i. As squares of S_i are inserted in order of increasing x-coordinate, it follows from Lemma 9.1.13 that all $s_{j'} \in S_i$ with $j' > j + 1$ are still insertable into F_{j+1}.

To see that remaining squares in $S_{i'}$ for $i' \neq i$ are still insertable, it suffices to see that from the perspective of such a square s', only a change to $x_{i'}$ (if $i' \neq k + 1$) or $x_{i'-1}$ (if $i' \neq 1$) can affect its insertability. We may thus assume that s' appears on the lower envelope of S_{i+1} (if $i \neq k + 1$) or the upper envelope of S_{i-1} (if $i \neq 1$). Assume without loss of generality that $i \neq k + 1$, s' appears on the lower envelope of S_{i+1}, and s_{j+1} appears on the upper envelope of S_i. Keeping in mind the definition of x_i (Definition 9.1.2), as all noninserted squares on the lower envelope of S_{i+1} start appearing on this envelope at a larger x-coordinate than the x-coordinate from which s_{j+1} appears on the upper envelope of S_i, the condition that $x'_i > x_i$ will still hold for s' in F_{j+1}.

The other case, when $i \neq 1$, s' appears on the lower envelope of S_{i-1}, and s_{j+1} appears on the lower envelope of S_i, is similar. We thus found a sequence of valid insertions as in the lemma statement. □

Lemma 9.1.15 *Let S be any smallest subset of $\overline{\mathcal{S}}$ containing $\mathcal{S}^l \cup \mathcal{S}^r$ and covering all points in $\mathcal{P}$. Then there is a sequence of $|S| - k - 1$ valid insertions starting from F^l, leading to $F_1, \ldots, F_{|S|-k-1} = F^r$ such that for any square $s \in S$, there is a front F_j containing s.*

Proof: This follows from the preceding lemma. Note that in the definitions of insertable, the coverage constraints are satisfied by S. □

The converse of this lemma is also true.

Lemma 9.1.16 *Let $l \geq 0$. Then any sequence of $l + k + 1$ valid insertions starting from F^l and resulting in F^r corresponds to a set $S \subseteq \mathcal{S}$ of cardinality l covering all points in $\mathcal{P}$.*

Proof: Take S to be the set of inserted squares, except those in $\mathcal{S}^r$. Because we only performed valid insertions, the set S covers all points in $\mathcal{P}$. □

Proof of Theorem 9.1.1: Construct a directed graph G with $V(G)$ equal to the set of all fronts and a directed edge from front F to F' if F' can be obtained from F by a single valid insertion. From the definition of a front, $|V(G)| = O(|\overline{\mathcal{S}}|^{4k+3})$. As each front allows for at most $4|\overline{\mathcal{S}}|$ valid insertions, $|E(G)| = O(|\overline{\mathcal{S}}|^{4k+4})$. Because the validity of an insertion can be checked in $O(|\mathcal{P}|)$ time, G can be constructed in $O(|\overline{\mathcal{S}}|^{4k+4}|\mathcal{P}|)$ time.

From Lemma 9.1.15 and 9.1.16, a shortest path in G from F^l to F^r corresponds to a minimum subset of $\mathcal{S}$ covering all points in $\mathcal{P}$. Using breadth-first search, a shortest path can be found in $O(|E(G)|) = O(|\overline{\mathcal{S}}|^{4k+4})$ time. Observe that $|\overline{\mathcal{S}}| = |\mathcal{S}| + |\mathcal{S}^l| + |\mathcal{S}^r| \leq 3|\mathcal{S}|$, because if no square intersects a certain line, we may ignore this line. Then the running time of the algorithm is $O((3|\mathcal{S}|)^{4k+4}|\mathcal{P}|)$. □

Combining Theorem 9.1.1 with the shifting technique, we obtain a ptas for Geometric Set Cover on unit squares. For each integer $0 \leq a \leq k-1$, let L_a denote the set of squares intersecting a line $y \equiv a \pmod{k}$. Moreover, for each $b \in \mathbb{Z}$, let $\mathcal{P}_a^b$ denote the set of points between lines $y = bk + a$ and $y = (b+1)k + a$. Apply the algorithm of Theorem 9.1.1 to each such set $\mathcal{P}_a^b$ and denote the returned set of squares by C_a^b. Then let $C_a = \bigcup_{b\in\mathbb{Z}} C_a^b$ and let $C_{\min}$ denote a smallest such set. Trivially, each C_a is a geometric set cover for $\mathcal{P}$, as $\bigcup_{b\in\mathbb{Z}} \mathcal{P}_a^b = \mathcal{P}$ for any value of a.

Lemma 9.1.17 *$|C_{\min}| \leq (1 + 1/k) \cdot |OPT|$, where OPT is a minimum geometric set cover.*

Proof: Let $\mathcal{S}_a^b$ denote the set of all squares in $\mathcal{S}$ covering at least one point in $\mathcal{P}_a^b$ for some a, b. We may assume that $C_a^b \subseteq \mathcal{S}_a^b$. Now observe that $OPT \cap \mathcal{S}_a^b$ is a cover for $\mathcal{P}_a^b$. Hence $|C_a^b| \leq |OPT \cap \mathcal{S}_a^b|$ and

$$|C_a| \leq \sum_{b\in\mathbb{Z}} |OPT \cap \mathcal{S}_a^b| \leq |OPT| + |OPT \cap L_a|$$

for any $0 \leq a \leq k-1$. A square is in L_a for precisely one value of a. Then

$$k \cdot |C_{\min}| \leq \sum_{a=0}^{k-1} |C_a| \leq \sum_{a=0}^{k-1} \Big(|OPT| + |OPT \cap L_a|\Big) = (k+1) \cdot |OPT|.$$

Therefore $|C_{\min}| \leq (1 + 1/k) \cdot |OPT|$. □

Theorem 9.1.18 *There is a ptas for Geometric Set Cover on unit squares.*

Proof: Consider some $\epsilon > 0$ and let $k = \max\{1, \lceil 1/\epsilon \rceil\}$. Following Theorem 9.1.1, we can compute $C_{\min}$ in $O(k|\mathcal{P}| \cdot (3|\mathcal{S}|)^{4k+4}|\mathcal{P}|)$ time. From the choice of k and Lemma 9.1.17, this is a $(1+\epsilon)$-approximation. The theorem follows immediately. □

9.1.1 Geometric Budgeted Maximum Coverage

The above ptas easily extends to the weighted case of Geometric Set Cover, by weighting the graph constructed in the proof of Theorem 9.1.1. We can however extend to the more general budgeted case as well.

Let $\mathcal{S}$ be a set of unit squares, $\mathcal{P}$ a set of points, c a cost function over $\mathcal{S}$, p a nonnegative profit function over $\mathcal{P}$, and B a budget. Let $p_{\max}$ denote the maximum profit of any single point. We define the function $cov(s)$ as the set of points in $\mathcal{P}$ covered by a square $s \in \mathcal{S}$. This notation extends to $cov(S)$ for a set $S \subseteq \mathcal{S}$. Abusing notation, we will use $p(S)$ to denote $p(cov(S))$.

Let $k \geq 2$ be an integer we determine later. Use slabs as before.

Theorem 9.1.19 *For any instance of Geometric Budgeted Maximum Coverage on a set of unit squares $\mathcal{S}$ where all points are inside $k-1$ consecutive height 1 horizontal slabs and all profits are positive integers, one can find for all $0 \leq r \leq |\mathcal{P}| \cdot p_{\max}$ a cheapest set of profit at least r (if one exists) in $O((3|\mathcal{S}|)^{4k}\,(|\mathcal{P}| \cdot p_{\max})^2)$ time.*

Proof: We modify the algorithm described above. Assume the cost of squares in $\mathcal{S}^l \cup \mathcal{S}^r$ to be zero. Remove the coverage constraints from the four definitions of insertable. Then, as in the proof of Theorem 9.1.1, we construct a directed graph G with $V(G)$ equal to the set of all fronts and an edge from F to F' if F' can be obtained from F by a single valid insertion.

Alter this graph G as follows. For any edge in $E(G)$ from some front F to a front F', we replace the edge by a path. The number of edges of the path is equal to the total profit of the points covered by the insertion. For example, for an upper insertion of a square s intersecting line i, this is the total profit of the points covered by u_i or l_{i+1} in $[x_i, x'_i] \times [i, i+1]$. The cost of inserting s is modeled by assigning a weight of $c(s)$ to the first edge of the path and assigning weight 0 to all other edges.

Now the number of edges on a F^l–F^r path minus $k+1$ is equal to the profit of the solution corresponding to this path. Its cost is equal to the weight of the path. Hence we aim to find for any $0 \leq r \leq |\mathcal{P}| \cdot p_{\max}$ a lightest path of length at least r. A straightforward dynamic programming algorithm for this problem takes $O(|E(G)| \cdot |\mathcal{P}| \cdot p_{\max}) = O((3|\mathcal{S}|)^{4k}(|\mathcal{P}| \cdot p_{\max})^2)$ time. □

By slightly changing the dynamic programming algorithm of Theorem 9.1.19, we can also deal with points of zero profit.

We now apply the shifting technique and scaling to obtain a ptas. Start by assuming integer profits. For each integer $0 \leq a \leq k-1$, let N_a denote the set of points between lines $y = bk + a$ and $y = bk + a + 1$ for any $b \in \mathbb{Z}$. Moreover, for any $b \in \mathbb{Z}$, let $\mathcal{P}_a^b$ be the set of points between lines $y = bk + a + 1$ and $y = (b+1)k + a$.

For any $0 \leq r \leq |\mathcal{P}| \cdot p_{\max}$, let $C_a^b(r)$ denote the set returned by the algorithm of Theorem 9.1.19, applied on $\mathcal{S}$ and $\mathcal{P}_a^b$, attaining profit at least r. We assume that $c(C_a^b(r)) = \infty$ if $C_a^b(r)$ profit at least r cannot be attained.

Let the nonempty sets $\mathcal{P}_a^b$ are numbered arbitrarily $\mathcal{P}_a^0, \ldots, \mathcal{P}_a^{l_a}$, and let $C_a^0, \ldots, C_a^{l_a}$ be the corresponding solutions. Define

$$\begin{array}{rcl} \mathsf{s}_a(0,r) & = & c(C_a^0(r)) \\ \mathsf{s}_a(b,r) & = & \min_{0\le r'\le r} \{c(C_a^b(r')) + s_a(b-1, r-r')\} \end{array}$$

for $1 \le b \le l_a$ and $0 \le r \le |\mathcal{P}| \cdot p_{\max}$. Observe that computing s_a takes $\mathrm{O}(|\mathcal{P}| \cdot (|\mathcal{P}| \cdot p_{\max})^2)$ time.

Let C_a denote a set attaining $\max_{0\le r\le |\mathcal{P}|\cdot p_{\max}}\{r \mid \mathsf{s}_a(l_a, r) \le B\}$ and let $C_{\max}$ denote a most profitable such set. By definition, $c(C_{\max}) \le B$.

Lemma 9.1.20 $p(C_{\max}) \ge (1 - 1/k) \cdot p(OPT)$, *where OPT is an optimal solution.*

Proof: Let $\mathcal{S}_a^b$ denote the set of squares in $\mathcal{S}$ covering at least one point in $\mathcal{P}_a^b$. Then it can be easily seen that

$$c\left(C_a^b\left(p\left(cov\left(OPT \cap \mathcal{S}_a^b\right) \cap \mathcal{P}_a^b\right)\right)\right) \le c(OPT \cap \mathcal{S}_a^b)$$

for any $0 \le a \le k-1$ and $0 \le b \le l_a$. Because for fixed a the sets $\mathcal{S}_a^b$ are pairwise disjoint, $\sum_{b=0}^{l_a} c(OPT \cap \mathcal{S}_a^b) \le B$. Then it follows from the definition of s and by induction that

$$p(C_a) \ge \sum_{b=0}^{l_a} p(cov(OPT \cap \mathcal{S}_a^b) \cap \mathcal{P}_a^b).$$

Since

$$\sum_{b=0}^{l_a} p(cov(OPT \cap \mathcal{S}_a^b) \cap \mathcal{P}_a^b) = p(OPT) - p(cov(OPT) \cap N_a)$$

and any point is in N_a for precisely one value of a,

$$\begin{array}{rcl} k \cdot p(C_{\max}) & \ge & \sum_{a=0}^{k-1} p(C_a) \\ & \ge & \sum_{a=0}^{k-1} \Big(p(OPT) - p(cov(OPT) \cap N_a)\Big) \\ & = & (k-1) \cdot p(OPT). \end{array}$$

Hence $p(C_{\max}) \ge (1 - 1/k) \cdot p(OPT)$. □

Theorem 9.1.21 *There is a ptas for Geometric Budgeted Maximum Coverage on unit squares.*

Proof: Consider some $\epsilon > 0$ and let $k = \max\{2, \lceil 1/\epsilon \rceil\}$. To deal with non-integer profits and to achieve polynomial running time, we first scale the profits. Define the integer profit function p' by $p'(u) = \left\lfloor \frac{|\mathcal{P}| \cdot p(u)}{\epsilon \cdot p_{\max}} \right\rfloor$ for any $u \in \mathcal{P}$. Now apply the above algorithm with p' and compute $C_{\max}$. Following Lemma 9.1.20, $p'(C_{\max}) \geq (1 - 1/k) \cdot p'(OPT)$, where OPT is an optimal solution with profit function p. Hence, as $p(OPT) \geq p_{\max}$,

$$\begin{aligned} p(C_{\max}) &\geq \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot p'(C_{\max}) \\ &\geq (1 - 1/k) \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \cdot p'(OPT) \\ &\geq (1 - 1/k) \cdot \left(p(OPT) - |\mathcal{P}| \cdot \frac{\epsilon \cdot p_{\max}}{|\mathcal{P}|} \right) \\ &\geq (1 - 1/k) \cdot (1 - \epsilon) \cdot p(OPT) \\ &\geq (1 - \epsilon)^2 \cdot p(OPT). \end{aligned}$$

The running time is

$$\mathrm{O}(k|\mathcal{P}| \cdot (3|\mathcal{S}|)^{4k} \, |\mathcal{P}|^4/\epsilon^2 + k|\mathcal{P}| \cdot |\mathcal{P}|^4/\epsilon^2) = \mathrm{O}(k|\mathcal{P}| \cdot (3|\mathcal{S}|)^{4k} \, |\mathcal{P}|^4/\epsilon^2),$$

because $p'_{\max} \leq |\mathcal{P}|/\epsilon$. This gives the ptas. □

9.1.2 Optimality and Relation to Domination

Geometric Set Cover and the geometric version of Minimum Dominating Set are closely related. We exploit this relation here to give a ptas for Minimum-Weight Dominating Set on unit square graphs and to show that the algorithms for (the budgeted version of) Geometric Set Cover developed above are essentially optimal.

Observe that two squares of side length 1 centered on points p and p' intersect if and only if p is contained in the square of side length 2 centered on p' and p' is contained in the square of side length 2 centered on p. Hence given a collection of unit squares $\mathcal{S}$, the minimum dominating set problem on $G[\mathcal{S}]$ is equivalent to the Geometric Set Cover problem on $\mathcal{S}'$ and the centers of $\mathcal{S}'$ as point set, where $\mathcal{S}'$ is obtained from $\mathcal{S}$ using the preceding observation (see also Mihalák [209]). Then the following result is immediate from Theorem 9.1.21.

Theorem 9.1.22 *There is a ptas for Minimum-Weight Dominating Set on unit square graphs.*

Recall from the discussion of Section 6.3.5 that the techniques developed earlier were not sufficient to give a ptas for the weighted case of Minimum Dominating Set. Theorem 9.1.22 therefore is the first ptas for Minimum-Weight Dominating Set on intersection graphs of two-dimensional objects.

Another consequence of the above reduction from Minimum Dominating Set on unit square graphs to Geometric Set Cover on unit squares is the following. Recall from Section 6.4 that the exponential time hypothesis states that n-variable 3SAT cannot be decided in $2^{o(n)}$ time.

Theorem 9.1.23 *If there exist constants $\delta \geq 1$, $0 < \beta < 1$ such that Geometric Set Cover or Geometric Budgeted Maximum Coverage on unit squares of density d have a ptas with running time $2^{O(1/\epsilon)^{\delta}} d^{O(1/\epsilon)^{1-\beta}} n^{O(1)}$, then the exponential time hypothesis is false.*

This is immediate from Theorem 6.4.4. Note that the algorithms of Theorem 9.1.18 and 9.1.21 are optimal for dense instances, but might still be slightly improved for nondense instances. It seems though that the analysis of the above algorithms does not improve when assuming bounded density. We believe however that some small changes to the algorithm are sufficient to make it optimal in this sense.

Similarly, we can show from Theorem 6.4.9 that Geometric Set Cover and Geometric Budgeted Maximum Coverage on unit squares have no eptas.

Theorem 9.1.24 *Geometric Set Cover and Geometric Budgeted Maximum Coverage on n unit squares of density $d = d(n) = \Omega(n^{\alpha})$ for some constant $0 < \alpha \leq 1$ cannot have an eptas, unless FPT=W[1].*

This again gives an indication that there is little chance to improve on the ptas of Theorem 9.1.22.

Finally, we note that Theorem 9.1.23 and Theorem 9.1.24 also hold (mutatis mutandis) on unit disks.

9.2 Hardness of Approximation

Not much is clear yet about the approximability of Geometric Set Cover. The approximation scheme of the previous section all but settled its approximability on unit squares. This gives hope for the existence of a ptas on unit disks. For more general objects however, we know almost nothing. In this section, we give several hardness results, showing that Geometric Set Cover is as hard as Minimum Set Cover in some cases and APX-hard in others.

Theorem 9.2.1 *Geometric Set Cover is not approximable within $(1 - \epsilon) \ln n$ for any $\epsilon > 0$, unless NP $\subset$ DTIME($n^{O(\log \log n)}$), on the following objects:*

- *convex polygons,*
- *translated copies of a single polygon,*
- *rotated copies of a single convex polygon,*
- *α-fat objects for any $\alpha > 1$.*

Geometric Set Cover is APX-hard on the following objects:

- *convex polygons with r corners, where $r \geq 4$,*
- *α-fat objects of constant description complexity for any $\alpha > 1$,*
- *rectangles,*
- *ellipses.*

Proof: The reductions are essentially the same as those in Section 8.5. The gadgets proposed there need only be slightly modified. In short, each object that we used to model an element $u \in \mathbb{U}$ for the universe $\mathbb{U}$ of the minimum set cover instance that we are reducing from, will be a point instead. For rectangles and ellipses, we use the gadget of the proof of Theorem 8.5.8, but any small plate or $S_{i,j}$ will be a point instead of a rectangle.

The $\ln n$-hardness on rotated copies of a single polygon follows by applying the same ideas as in Theorem 8.5.3, but on slices of a single large disk. Rotations of the disks then allow for the same kind of construction. □

The $\ln n$-hardness on translated copies of a single polygon seems somewhat at odds with Laue's [188] constant-factor approximation algorithm on translated copies of a fixed three-dimensional polytope. However, the complexity of the polygon used in the hardness result depends on the number of polygons (i.e. on the number of sets in the minimum set cover instance). Hence the polygon may not been assumed to be fixed. When looking closely at Laue's result, one can observe that the approximation ratio of his algorithm actually depends (linearly) on the complexity of the polytope if it is nonconvex.

Using an idea by Khuller, Moss, and Naor [165] to reduce from Minimum Set Cover to Budgeted Maximum Coverage, we can also prove hardness results for Geometric Budgeted Maximum Coverage.

Theorem 9.2.2 *Geometric Budgeted Maximum Coverage is not approximable within ratio better than $(1 - 1/e)$, unless NP $\subset$ DTIME($n^{O(\log\log n)}$), on the following objects:*

- *convex polygons,*
- *translated copies of a single polygon,*
- *rotated copies of a single convex polygon,*
- *α-fat objects for any $\alpha > 1$.*

Note that the APX-hardness results of Theorem 9.2.1 cannot be transferred to Geometric Budgeted Geometric Coverage using this trick. The underlying reductions are not from Minimum Set Cover, but from Minimum k-Set Cover and Minimum Vertex Cover, for which the idea of Khuller, Moss, and Naor does not appear to work.