2027 CG:SHOP: Weighted set cover and partitions

Lecturer: Mihir Tandon, Scribe: Terry Qiu

Content Table

1 Weighted Set Cover problem **1**

2 Grid and base rectangles **2**

3 Powersets and Optimal Coverings **4**

4 Partitition algorithm, conclusion and the lawn mowing probelm **9**

1 Weighted Set Cover problem

Culberson and Reckhow showed in the 90s that covering a rectilinear polygon P with rectangles is NP complete. More specifically, they showed that the following decision problem is NP complete:

(**Rectangle Cover Decision**). Given a positive integer k and a rectilinear holeless polygon P, does there exists a set of k rectangles that cover P?

Although not necessarily the same, a related problem asks,

(**Unweighted Rectangle Cover**). Given a rectilinear holeless polygon P, can we find a set of rectangles $(S_1, ..., S_n)$ that cover P and minimizes $\sum_{i=1}^{n} c(S_i) = \sum_{i=1}^{n} \alpha$?

This is the problem of minimizing the "cost" of covering a polygon with rectanglest. In the case where the cost is some unit α, this just asks whether we can find a cover of P that is one of the smallest.

This is a specialisation of the following problem considered by Hanauer et al.

(**Weighted Rectangle Cover**). Given a rectilinear polygon P find a set of rectangles $(S_1, ..., S_n)$ such that $\bigcup_{i=1}^{n} S_i = P$ and they minimize the sum $\sum_{j=1}^{n} c_{\alpha,\beta}(S_j)$

Notice that if $\beta = 0$, this becomes the unweighted rectangle cover problem.

There could be multiple different motivations for investigating the linear cost function. The clearest one seems to be that in computer graphics, it may be desirable to investigate a "minimal cost decomposition" of a polygonal shape. We may interpret the cost of initializing the shape and the cost of rendering it as α and $\beta \cdot A(R)$ respectively. The cost of rendering it is ofcourse linear in the area of the rectangle.

tl;dr: we want to find a covering of a rectilinear polgyon with the smallest cost under some give cost function that is linear wtih respect to the size of the rectangle. Some problems are solved (e.g the unweighted set cover problem) by solving the more generalized problem of the cost function.

2 Grid and base rectangles

Definition 1. *(Grid rectangles) Extend each edge of a polygon through infinity.The rays partition the polygon into gird rectangles.*

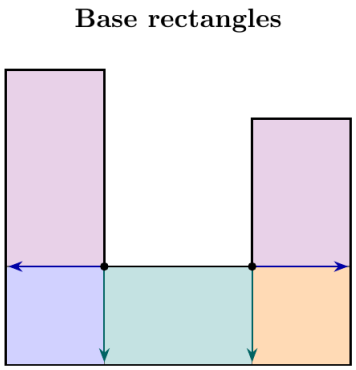


Figure 1

Definition 2. *(Base rectangles) Extend a vertical and horizontal ray inwards from any concave vertex and end them when they touch an edge of the polygon. The rays partition the polygon into base rectangles.*

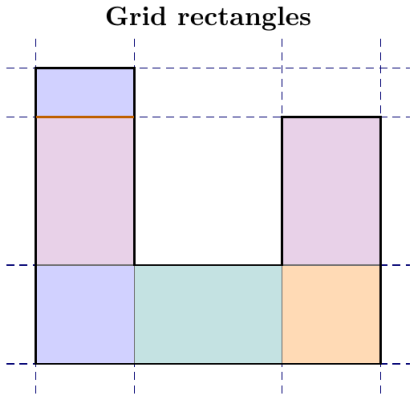


Figure 2

The grid rectangles refine the base rectangles in the sense that every grid rectangle is a subset of a base rectangle.

Lemma 1. *The number of base rectangles is $O(n^2)$ for a polygon P with $2n$ edges.*

Proof. The number of vertices is not more than the number of edges. So there cannot be more than $2n$ horizontal and $2n$ vertical rays in total (a vertical and horizontal ray comes from each $2n$ vertices). There are at most $4n^2$ intersections of the horizontal and vertical rays (each two distinct rays intersect exactly one, and there are at most $2n \cdot 2n$ different intersecting positions).

Since each intersection defines a base rectangle, the number of base rectangles grows at $O(n^2)$. □

The proof of the following lemma in the paper states that $\mathcal{V}(P) \leq 2n$. Why is it not the case that $\mathcal{V}(P) = 2n$ always? Remember that there are holes in the polygon and we can have holes that share vertices. If it is hole free then I think we always have the number of vertices equals the number of edges.

The following maybe useful for practical purposes. Recall that the set of all base rectangles partition our rectangle. So we can divide the polygon into regions which we know how to solve.

Lemma 2 (Computing base rectangles)**.** *We can compute all the base rectangles in time complexity $O(n \log n + n^2)$ in the worst case.*

tl;dr: We defined base and grid rectangles of a polygon. We showed that the base rectangles grows only quadratically with the number of edges and we are aware that there is a way to compute base rectangles in $O(n \log n + n^2)$.

3 Powersets and Optimal Coverings

Definition 3. *(Rectanglur powerset) The power set of a set of rectangle $\mathcal{R}$ is the set of all rectangles that a strict subset of rectangles in $\mathcal{R}$ can cover. We write $\Gamma(\mathcal{R})$ for the powerset of $\mathcal{R}$. See Figure 3.*

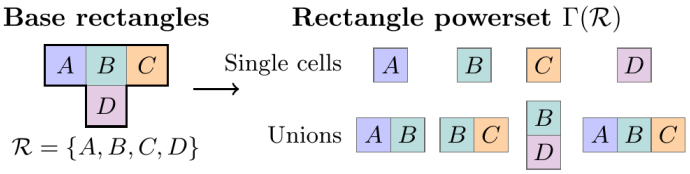


Figure 3

Lemma 3. *For a set of rectangles $\mathcal{R}$ of size r, the size of its power set $|\Gamma(\mathcal{R})|$ is $O(r^2)$.*

Proof. Union all the vertices of all the rectangles in the set. There are at most $4r$ vertices. The top left and top right vertices identify uniquely a rectangle. There are thus $\binom{4r}{2}$ distinct rectangles. Algebraically, this resolves to $k \cdot r^2 \in O(r^2)$ for some constant k. See Figure 4. □

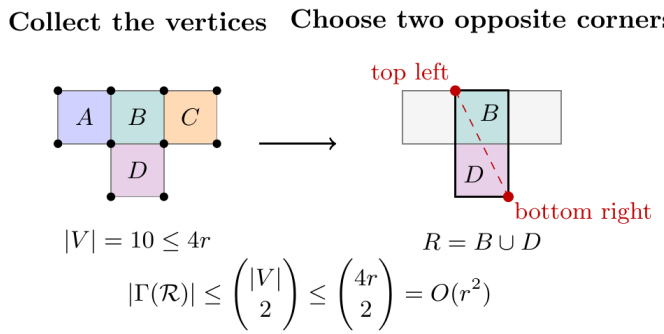


Figure 4

Lemma 4. *Let P be a polygon with $2n$ edges. Let $\mathcal{B}(P)$ be its base rectangles. $|\Gamma(\mathcal{B}(p))|$ is $O(n^4)$.*

Proof. Use Lemma 3. □

Theorem 1. *For every polygon P, and α, β non negative. There exists an optimal weight cover that is formed from rectangles in the power set of the grid rectangles. That is, there exists an optimal weighted cover C such that $C \subseteq \Gamma(\mathcal{D}(P))$.*

Proof. The core idea is to show that given any cover C'', we can find a cover C' that is not larger than C' and not costlier than C''. Most importantly, we can show that $C' \subseteq \Gamma(D(P))$.

The authors do this by running sweep lines in both the horizontal and vertical direction. The sweep line is a glorified term for scaning the polygon left to

This is a very important theorem. It says that to find a cover, it suffices to look towards the rectangles covered by the grid rectangles. This tells us where to look for.

right or top to down.

Suppose C'' is any cover of P. Let $C' = C''$ to begin with. Suppose also we scan from the left to the right, because top down is similar. We call a x coordinate that is not an x coordinate of any vertex of the original polygon P an x event. This is a bit of a mouth full and slightly confusing. But essentially it is where one rectangle ends and another begins but where there is not polygonal vertices with the same x value. See the figure.

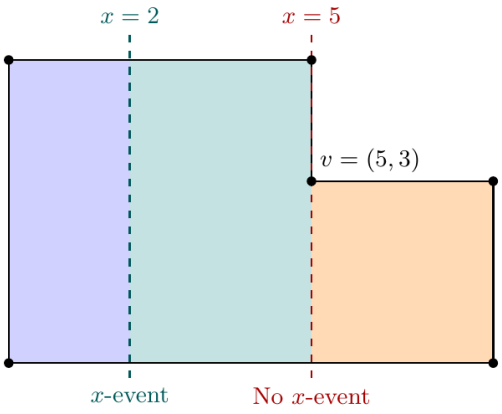


Figure 5

As we go from left to right, when we encounter an x-event, we do the following:

Step 1. Notice that an x-event occurs at a vertical edge where two sides are rectangles of different height. Notice that we use the plural because there could be more than one rectangle stack on top of each other on both sides. See

Step 2. We can shift the vertical edge where the x-event occurs by some δ. If

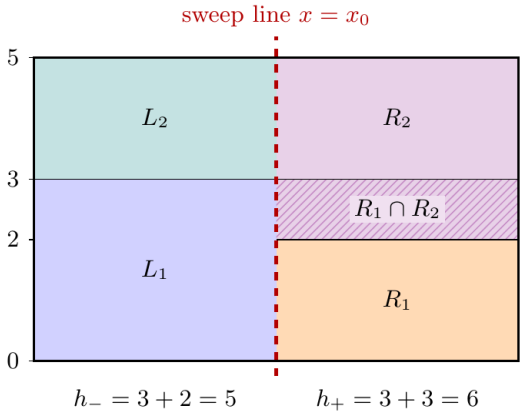


Figure 6

we shift right we have $\delta \geq 0$, if we shift left we have $\delta < 0$. As can be readily shown the difference in area after some shift is

$$\beta \cdot \delta(h^- - h^+)$$

The algebra needed to show this is true is left to an exercise to the reader

where h^- and h^+ are the sum of the heights of the rectangles on each side. They may not be equal because there might be overlap on either side. That is, the rectangles may not be entirely disjoint.

In the case where they are equal, we choose the $\delta = x^+ - x$ such that the right hand side "disappears". The area is the same in this case.

When they are not equal, move the vertical line to the side that makes the term positive. That is, the area decreases in this case. See figure. Doing this across all $x-$events still gives us a feasible solution. That is, the resulting rectangle still covers the polygon. The authors gives a proof to handle degenerate cases. Here we won't repeat it because the core idea has already been illustrated. □

Intuitively, this reduces "waste", by always going in the direction where the rectangles are not disjoint

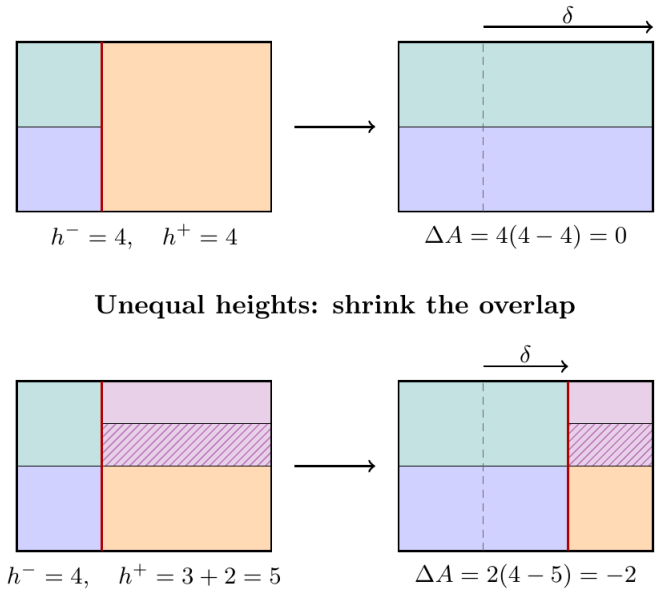


Figure 7

Applying theorem 1 to a cover of an L-shaped polygon has the following effect.

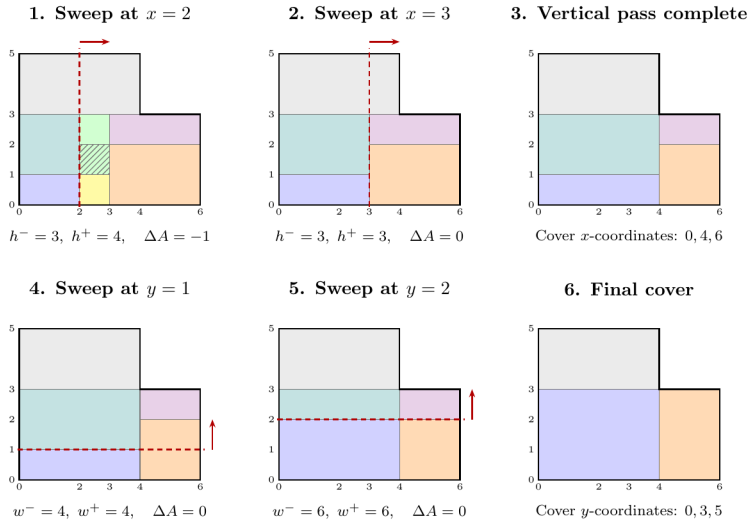


Figure 8

Theorem 2. *For every polygon P, and α, β non negative. There exists an optimal weight cover that is formed from rectangles in the power set of the grid rectangles. That is, there exists an optimal weighted cover C such that $C \subseteq \Gamma(\mathcal{B}(P))$.*

Proof. Essentially, the same as theorem 1. Except now, we start an optimal solution given by theorem 1 and then modify using similar sweep line reasonings to get a rectangular cover aligned with the base rectangles. □

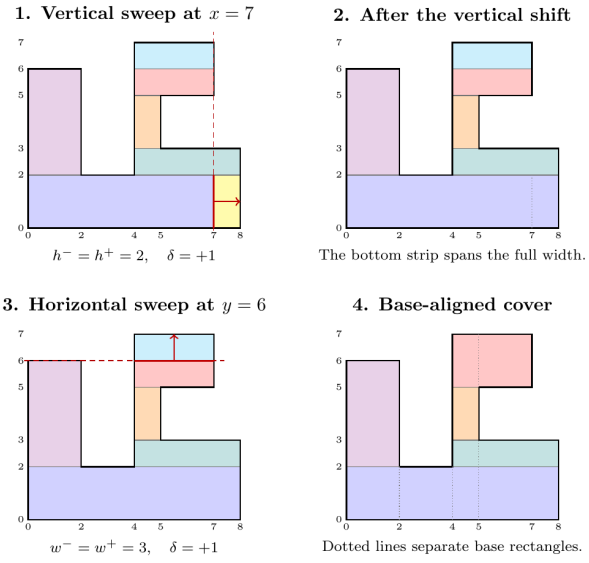


Figure 9

This diagram is not to say every cover given by theorem 2 is a partition. Recall that a cover returned by theorem 2 is really just a cover formed by

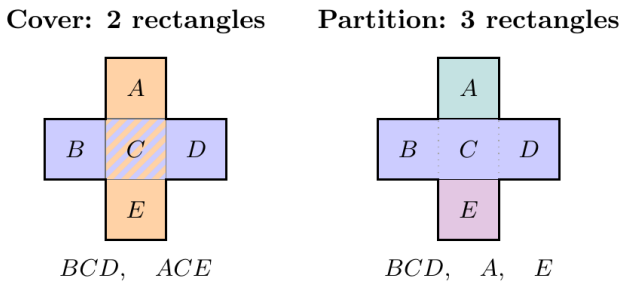


Figure 10

rectangles in the power set of the base rectangles. So the case on the left could in principle be a cover returned by theorem 2.

Theorem 3. *Let A_{min} be the minimum area of any base rectangle of a given polygon P. Then, a partition is an optimal cover of P with cost function α, β for $\alpha, \beta \geq 0$ if $\beta \cdot A_{min} \geq \alpha$.*

Proof. Start with a cover from theorem 2. If it is a partition then we are done. Otherwise there is an overlap somewhere. For each overlap, remove the overlapping region from one rectangle that contains it. This new rectangle is necessarily a base rectangle. So the new cost of the region is $2\alpha+\beta(R_1+A+R_2)$. Previously, we had $2\alpha + \beta(R_1 + A + R_2 + A)$. Subtracting the new cost from the original we have

$$\delta = \beta A - \alpha$$

If the assumption holds then, we have $\beta\alpha \geq \beta A_{min} \geq \alpha$. This implies that the difference is non negative. So the new cost must be not greater than the old cost. Do this for every overlap and we will arrive at a partition with cost not greater than an optimal cover. □

The overlap could be multiple base rectangles but the inequality would still hold. That is, A might be the sum of the area of multiple base rectangles.

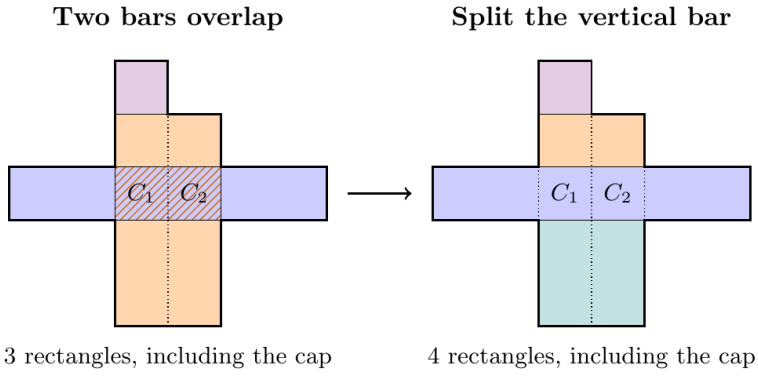


Figure 11

tl;dr There are algorithms that can help us compute optimal covers of a polygon and partitions of it based on some cost function.

4 Partitition algorithm, conclusion and the lawn mowing probelm

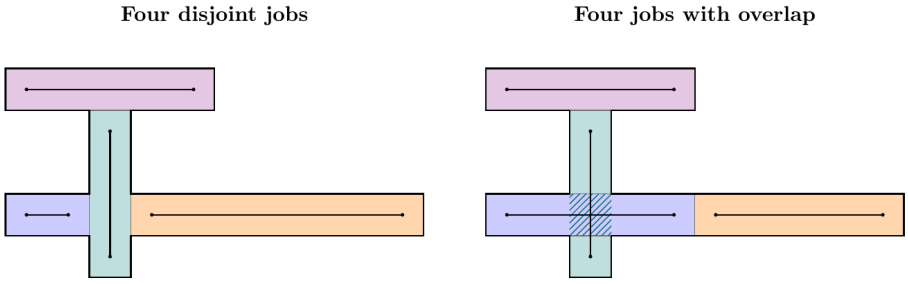
Technically, we can already compute partitions of a lawn using theorem 3. But recall that theorem 3 is a conditional theorem that says we can have a optimal cost partition given some condition holds. Ohtsuki's algorithm below gives a way to compute a partition with the minimum number of rectangles without respect to some cost function.

Lemma 5. *(Ohtsuki's) We can compute the minimum sized partition of a polygon in $O(n^{1.5} \log n)$ time.*

Transitioning back to our lawn mowing problem, notice that Ohtsuki's algorithm could be a starting point for our partitioning subroutine. First compute a partition using Ohsuki's algorithm, then make random guesses to modify the partition using roughly the ideas embodied in the processing procedure embodied in the paper. It's not obvious the best divide and conquer strategy has to be one stricly where the rectangles do not overlap.

For example, suppose Ohtsuki's algorithm says that the best partition has 14 rectangles, if we relax the condition and allow overlaps between rectangles, can we reduce the number of rectangles neeeded to cover the region? Is this necessarily better or worse or neither? In what cases could it be worse or better? The figure below gives a toy example. The example is slightly silly but its hard to prove that similar cases might not arise with larger graphs.

Given the same load balancing algorithm, while the total area of the rectangles in a partition is always less than that of a cover, there could be scenarios where having some overlap could actually improve performance



One 2×2 cutter per rectangle. Each black segment is traversed out and back.

Figure 12

We can also try computing the optimal cover of different $\alpha, \beta > 0$ and seeing whether some values produce better rectangular covers.

tl;dr The partitions and covering algorithms we learnt help us with dividing our lawns into simple pieces. Making practical adjustments to them and knowing when to do what adjustment could be investigated.