E0 249: Approximation Algorithms March 11, 2022

Geometric Approximation - Week 3

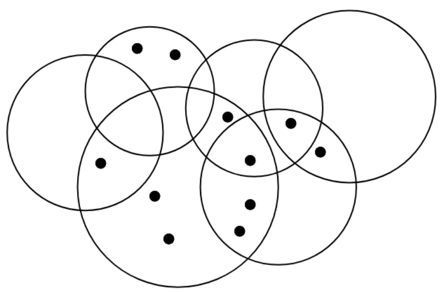
Instructor: Arindam Khan *Scribes: Anish Hebbar, Upamanyu Y*

1 Introduction

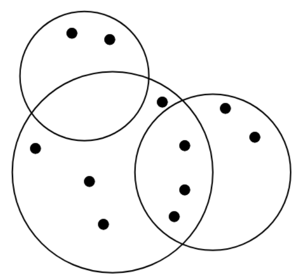
This week, we'll be using techniques like VC dimension and ε-nets, and applying them to prove approximation guarantees. We'll also learn about geometric packing problems, and techniques related to them.
The focus of the first half will be two problems - Geometric Set Cover and Geometric Hitting Set.

1.1 Geometric Set Cover (discrete version)

Problem statement: Given a set of m objects I (which can be weighted or unweighted) and a set of n points P which are contained in these objects, the aim is to find the minimum weight subset $S^* \subseteq I$ which covers all of P.
In the example below, the objects are unweighted circles.



A set cover problem

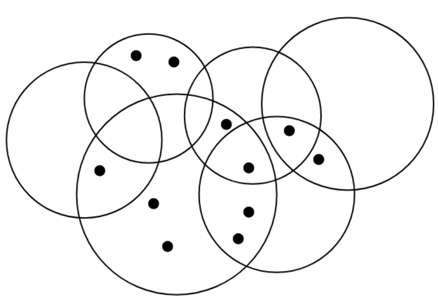


The minimal covering subset

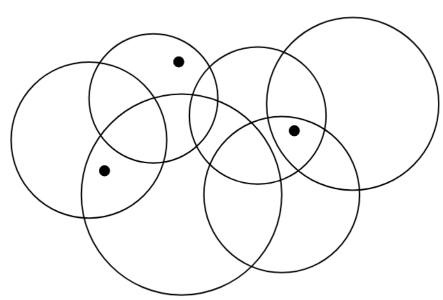
Remark This turns out to be equivalent to the hitting set problem in the dual range space, which we will come across later.

1.2 Hitting Set

Problem statement: Given a set of m objects I and a set of n points P (which can be weighted or unweighted), the aim is to find the minimum subset $S^* \subseteq P$ which 'stabs' all objects in I. That is, for any $A \in I$, we have some $p \in S^*$ for which $p \in A$.



A hitting set problem



A minimal hitting set

We'll see how to use VC dimension and ε-nets to obtain good approximations for these problems. The motivation behind using these techniques is the idea of **sampling** - estimating properties of the whole sample space using a small set of observations. The minimum size sample required is called the **sample complexity**. It turns out that we can capture the structure of a distribution or point set by using a small subset (an ε-net), whose size depends on the complexity of the structure, but is independent of the size of the point set.

2 VC dimension

[Some resources for further reading on this topic are:

- Course notes for E0 234: Introduction to Randomized Algorithms [1]
- Chapter 20 of Geometric Approximation Algorithms by Har-Peled [2]
- Chapter 14 of Probability and Computing by Mitzenmacher and Upfal [3]

Definition 1 (Range space)**.** *A range space is a pair* $(X, \mathcal{R})$*, where*

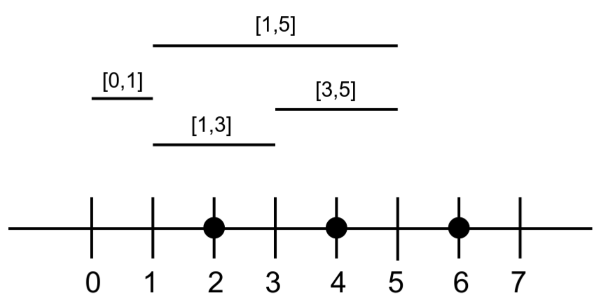
- X *is a finite or infinite set of points, called the ground set*
- $\mathcal{R}$ *is a family of subsets of* X*, called ranges.*

An example is the set of closed intervals on the real line, i.e. $X = \mathbb{R}$ and $\mathcal{R} = \{[a, b] \mid a, b \in \mathbb{R}, a \leq b\}$.

Definition 2 (Projection of a range space)**.** *Given a range space* $(X, \mathcal{R})$ *and a set* $S \subseteq X$*, the projection of* $\mathcal{R}$ *on* S *is defined as*

$$\mathcal{R}|_S = \{R \cap S \mid R \in \mathcal{R}\}.$$

In the image below, $X = \mathbb{R}$ and $\mathcal{R}$ is the four intervals shown. If we take $S = \{2, 4\}$, then we can get every subset of S in the projection:



- For $R = [0, 1]$, $R \cap S = \emptyset$
- For $R = [1, 3]$, $R \cap S = \{2\}$
- For $R = [3, 5]$, $R \cap S = \{4\}$
- For $R = [1, 5]$, $R \cap S = \{2, 4\}$.

S is said to be **shattered** by $\mathcal{R}$ if $\mathcal{R}|_S = 2^S$ (as in the above example).
If we take $S = \{2, 4, 6\}$, then it is not shattered - we cannot get the subset $\{2, 6\}$, as any interval that contains 2 and 6 must also contain 4. In general, no three points on the real line can be shattered by intervals.

Definition 3 (VC dimension)**.** *The Vapnik-Chervonenkis (VC) dimension of a range space* $(X, \mathcal{R})$ *is the maximum cardinality of a set* $S \subseteq X$ *shattered by* $\mathcal{R}$*. If arbitrarily large sets are shattered by* $\mathcal{R}$*, then the VC dimension is infinite.*

So in the example of $X = \mathbb{R}$ and $\mathcal{R} =$ the set of closed intervals on $\mathbb{R}$, the VC dimension is 2 (as no 3 points can be shattered).

Remark For the VC dimension of $\mathcal{R}$ to be $\geq d$, we only need *some* set S with cardinality d which is shattered by $\mathcal{R}$. On the other hand, to show that the VC dimension is $\leq d$, we need to show that *all* sets of cardinality $> d$ are not shattered by $\mathcal{R}$. (In fact, it is sufficient to show that all sets of cardinality $d + 1$ are not shattered, as the subset of a shattered set is shattered).

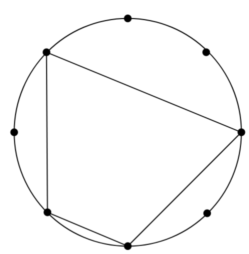
Theorem 4 (Sauer-Shelah Theorem)**.** *For a range space* $(X, \mathcal{R})$ *with* $|X| = n$ *and VC dimension* d*, we have* $|\mathcal{R}| \leq (n^d)$*.*

So for a finite ground set, a low VC dimension implies that the cardinality of $\mathcal{R}$ is low.
For example, let X be a set of n points in $\mathbb{R}$, and elements of $\mathcal{R}$ be intervals. Recall that the VC dimension is 2 as no three points can be separated. An interval is defined by a pair of points, and there are $n + \binom{n}{2}$ pairs we can choose, so we have $|\mathcal{R}| < n^2$.

Some examples:

Convex sets

If we take $X = \mathbb{R}^2$ and $\mathcal{R} =$ all closed convex subsets of $\mathbb{R}^2$, then the VC dimension is infinite. To see this, for any $n \in \mathbb{N}$, take S_n to be a set of n points on the circumference of a circle.

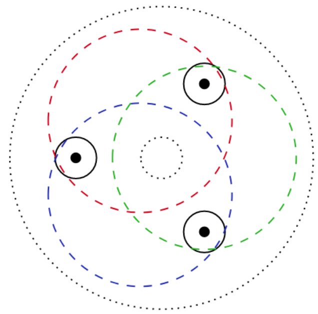


A subset of 4 points for S_8

Any nonempty $Y \subseteq S_n$ defines a convex polygon (whose vertices are the elements of Y) which contains no elements of $S_n \setminus Y$. The empty set can also be obtained as a projection, so all subsets of S_n can be obtained. That is, S_n can be shattered for any n.

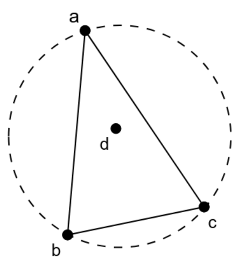
Discs

For $X = \mathbb{R}^2$ and $\mathcal{R} =$ all discs in $\mathbb{R}^2$, we show that the VC dimension is 3.

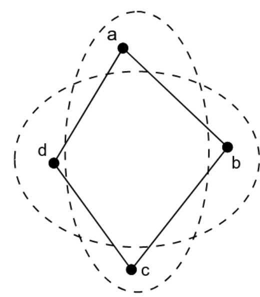


3 points being shattered by 8 discs

Any 3 points in general position can be shattered (an example is shown above). So the VC dimension is ≥ 3. To show that it is ≤ 3, we need to show that no set of cardinality 4 is shattered.



Case 1

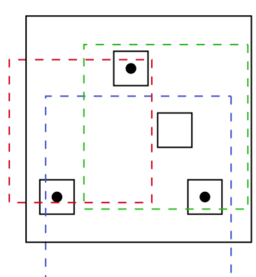


Case 2 - this is not possible for discs.

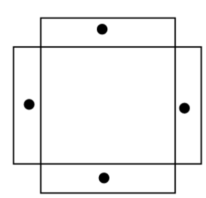
Let the set be $P = \{a, b, c, d\}$.
Case 1: If the convex hull of P has only 3 of the points on its boundary, say a, b, c, then d lies in the interior. So any convex set that contains a, b and c must also contain d. So we cannot obtain the set $\{a, b, c\}$.
Case 2: Suppose the convex hull has all four points on the boundary, a, b, c, d in clockwise order. If we have discs whose projections are $\{a, c\}$ and $\{b, d\}$, then they must intersect at 4 points, which is not possible (atmost two points of intersection are possible).

Squares and rectangles

For $X = \mathbb{R}^2$ and $\mathcal{R} =$ all axis-parallel squares in $\mathbb{R}^2$, the VC dimension is 3. The proof is the exact same as that for discs; squares cannot 'cross', i.e. cannot intersect at more than 2 points.

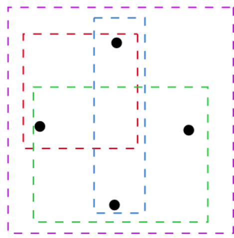


3 points being shattered by 8 squares

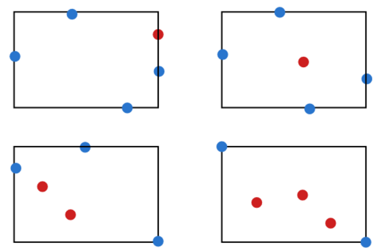


Not possible for squares

However, if we let $\mathcal{R}$ be the set of all axis-parallel rectangles in $\mathbb{R}^2$ instead, then some configuration of 4 points can be shattered, (as rectangles can cross each other). For 5 points, we consider the minimal rectangle enclosing these points.



4 points being shattered
(not all rectangles shown)



In each case, the set of blue
points cannot be shattered.

- If all 5 points are on the boundary, then there is a set of 4 (one point on each side) that cannot be obtained.
- If there is atleast one point on the interior, then the set of points on the boundary cannot be obtained, for convexity reasons.

So in this case the VC dimension is 4.

In general, most simple geometric ranges have low VC dimension.

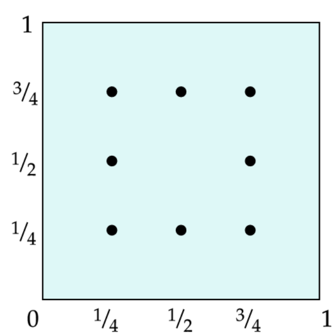
3 ε-nets

An ε-net is a combinatorial object that intersects with every range of sufficient size.

Definition 5 (combinatorial definition)**.** *Let $(X, \mathcal{R})$ be a range space, and let $A \subseteq X$ be a finite subset. A subset $N \subseteq A$ is a combinatorial ε-net for A if N has a nonempty intersection with every set $R \in \mathcal{R}$ such that $|R \cap A| \geq \varepsilon |A|$.*

Definition 6 (probabilistic definition)**.** *Let $(X, \mathcal{R})$ be a range space, and $\mathcal{D}$ be a probability distribution on X. A set $N \subseteq X$ is an ε-net for X with respect to $\mathcal{D}$ if for any $R \in \mathcal{R}$ such that $\Pr_{\mathcal{D}}(R) \geq \varepsilon$, the set R contains at least one point from N. That is:*

$$\forall\, R \in \mathcal{R},\;\; \Pr_{\mathcal{D}}(R) \geq \varepsilon \implies R \cap N \neq \emptyset\,.$$



For example, let A be the unit square and let $\mathcal{R}$ be all closed rectangles. The set of points N is a combinatorial 1/4-net. Note that this corresponds to the probabilistic setting where $\mathcal{D}$ is the uniform distribution on A.

Theorem 7 (ε-net theorem)**.** *Let $(X, \mathcal{R})$ be a range space with VC dimension d, and let $\mathcal{D}$ be a probability distribution on X. For any $0 < \delta, \varepsilon \leq \frac{1}{2}$, there is an $m = O(\frac{d}{\varepsilon} \ln \frac{d}{\varepsilon} + \frac{1}{\varepsilon} \ln \frac{1}{\delta})$ such that a random sample from $\mathcal{D}$ of size m is an ε-net for X with probability at least $1 - \delta$.*

Notes: m does not depend on $|X|$. Also, the $\frac{1}{\varepsilon} \ln \frac{1}{\delta}$ term is small, and can be treated as constant.

Approximation for hitting set

We use the above theorem to get an $O(d\ln(d\,\mathrm{OPT}))$-approximation for hitting set on a range space with VC dimension d.

Problem setup: X has n elements $e_1, \dots e_n$, and $\mathcal{R}$ has m sets $S_1, \dots S_m$.

Algorithm:

1. Guess OPT by binary search. Take $\varepsilon = \frac{1}{2\,\mathrm{OPT}}$.
2. Initialise weights $w(e_i) := 1$ for all $i \in [n]$.
3. Loop:
   - Find an ε-net N_ε of size $O(\frac{d}{\varepsilon}\ln\frac{d}{\varepsilon})$, using the above theorem.
   - If all sets are hit, return N_ε and stop.
   - Else, there is some S_j for which $S_j \cap N_\varepsilon = \emptyset$. Double the weights of each element of S_j, i.e. assign $w(e_i) := 2w(e_i)\;\forall\, e_i \in S_j$.
   - Goto loop.

If the algorithm terminates, then from our choice of ε we get an $O(d\ln(d\,\mathrm{OPT}))$ approximation. This is a variant of a Multiplicative Weight Update (MWU) algorithm. Intuitively, the total weight of points is increasing by a rate of $(1+\varepsilon)$, and OPT is increasing at a faster rate of $(1+\frac{1}{\mathrm{OPT}}) = (1+2\varepsilon)$. So it should terminate quickly with a good guarantee. We prove this with the following theorem:

Theorem 8. *If $\exists$ a hitting set of size OPT, the doubling process can happen at most $O(\mathrm{OPT}\cdot\log\frac{n}{\mathrm{OPT}})$ times, and the total weight is at most n^4/OPT^3.*

Proof. Let H be an optimal set.
For input X, say the set S_j is returned by some iteration. Then $w(S_j) \le \varepsilon w(X)$.
Thus in each iteration, $w(X)$ becomes at most $w(X) + w(S_j) \le (1+\varepsilon)w(X)$.
So the total weight of X after k iterations is

$$w(X) \le n(1+\varepsilon)^k \le ne^{\varepsilon k} \qquad [\text{as } 1+\varepsilon \le e^\varepsilon \text{ for } \varepsilon > 0]$$

As H is a hitting set, $H \cap S_j \neq \emptyset$. So atleast one element of H is doubled in every iteration. Say $h \in H$ is doubled z_h times.
We use Jensen's inequality: given $p_i \ge 0$ and $\sum_i p_i = 1$, then for a convex function $\phi(x)$ we have $\sum_i p_i\phi(x_i) \ge \phi(\sum_i p_i x_i)$. Take $\phi(x) = 2^x$ and $p_i = 2\varepsilon$ for $i \in [|H|]$ (note that $|H| = 1/2\varepsilon$). This gives us

$$w(H) = \sum_{h\in H} 2^{z_h} = \frac{1}{2\varepsilon}\sum_{h\in H} 2\varepsilon\cdot 2^{z_h} \ge \frac{1}{2\varepsilon}2^{\sum 2\varepsilon z_h} \ge 2^{2\varepsilon k}/2\varepsilon\,.$$

The last inequality holds because $\sum_h z_h \ge k$.
As $w(H) \le w(X)$, we have

$$\begin{aligned}
&(2^{2\varepsilon k})/2\varepsilon \le ne^{\varepsilon k} \le n\cdot 2^{\frac{3}{2}\varepsilon k} \qquad [\text{as } e \le 2^{\frac{3}{2}}]\\
\implies& 2^{2\varepsilon k - 3\varepsilon k/2} \le 2n\varepsilon\\
\implies& 2^{2\varepsilon k} \le 2n\varepsilon \implies \varepsilon k/2 \le \log 2n\varepsilon\\
\implies& k \le \frac{2}{\varepsilon}\log(2n\varepsilon) = O(\mathrm{OPT}\log\frac{n}{\mathrm{OPT}})
\end{aligned}$$

Thus $w(X) \leq ne^{\varepsilon k} \leq ne^{2\log(2n\varepsilon)} \leq O(n^3/\operatorname{OPT}^2)$. $\square$

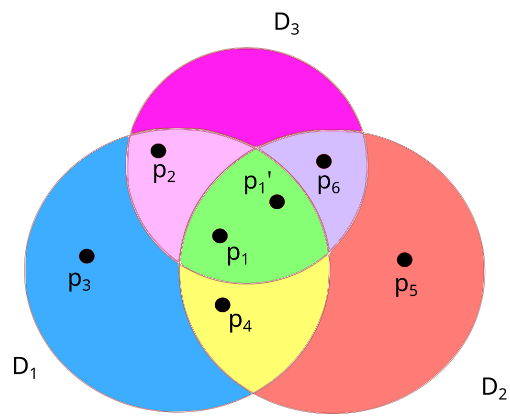
Further results

If $\exists$ ε-nets of size $O(\frac{d}{\varepsilon})$, then we can take $\frac{1}{\varepsilon} = \Theta(\operatorname{OPT})$ and use the same argument to get an $O(d)$-approximation.

- ε-nets of size $O(\frac{1}{\varepsilon})$ exist for discs in $\mathbb{R}^2$, giving an $O(d)$-approximation [4].
- For rectangles, ε-nets of size $O(\frac{1}{\varepsilon}\log\log\frac{1}{\varepsilon})$ exist [5]. Taking $\varepsilon = \Theta(\frac{1}{\operatorname{OPT}})$ gives an $O(\operatorname{OPT}\log\log\operatorname{OPT})$-approximation, which is currently the best known.
- Getting an $O(1)$ approximation for geometric hitting set/set cover for rectangles is an open problem. Note that set cover is just the hitting set problem in the dual range space.

Definition 9. *The* ***dual range space*** *to a range space $S = (X, \mathcal{R})$ is the space $S^* = (\mathcal{R}, X^*)$, where $X^* = \{\mathcal{R}_p : p \in X\}$. $\mathcal{R}_p$ is the set of ranges $r \in R$ that contain the point $p \in X$.*

Consider the below example with disks, with the range space denoted by an incidence matrix between disks and points. The dual range space is essentially obtained by taking the transpose of the incidence matrix. One can also see that $S^{**} = S$ (dual of dual range space).



	p_1	p_1'	p_2	p_3	p_4	p_5	p_6
D_1	1	1	1	1	1	0	0
D_2	1	1	0	0	1	1	1
D_3	1	1	1	0	0	0	1

	D_1	D_2	D_3
p_1	1	1	1
p_1'	1	1	1
p_2	1	0	1
p_3	1	0	0
p_4	1	1	0
p_5	0	1	0
p_6	0	1	1

Lemma 10. *Consider a range space $(X, \mathcal{R})$ with VC-dimension d. Then the dual range space $S^* = (\mathcal{R}, X^*)$ has VC-dimension $\leq 2^{d+1}$. (See [2] or [1] for details)*

The above gives us a bound on the VC-dimension of the dual range space. In particular, it shows that if the VC-dimension of the range space is $O(1)$ (which is true in most geometric settings), so is the VC-dimension of the dual range space. Thus, to solve Geometric Set Cover with Rectangles, we instead solve Geometric hitting set with dual range space of rectangles, which has VC-dimension $O(1)$.

LP-based approach for hitting set [6] :

Natural LP: (LP1)

$$\min \sum_{u\in X} x_u = J$$

$$\sum_{u\in S} x_u \geq 1,\ \forall S \in \mathcal{R}$$

$$x_u \geq 0,\ \forall u \in X$$

Equivalent LP: (LP2)

$$\max \quad \varepsilon$$

$$\sum_{i\in S} \mu_i \geq \varepsilon,\ \forall S \in \mathcal{R}$$

$$\sum_{u\in X} \mu_u = 1$$

$$\varepsilon, \mu_u \geq 0,\ \forall u \in X$$

LP1 clearly corresponds to the hitting set problem, with J being the size of the optimal hitting set. We can transform it into the equivalent LP (LP2), using the transformation

$$\varepsilon = \frac{1}{\sum_{u\in X} x_u},\ \ \mu_u = \varepsilon \cdot x_u\, \forall u \in X$$

Indeed, if J^* and ε^* are the optimums respectively for the two LPs, then the above equivalence transform shows that $J^* = \frac{1}{\varepsilon^*}$.

Algorithm:

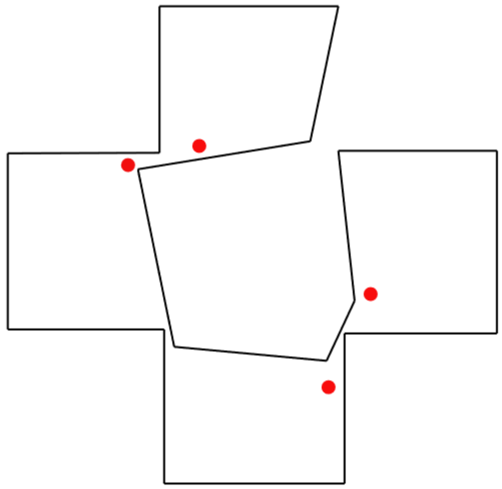
- Solve LP2 to obtain μ^*, ε^*.
- Find ε^*-net H with $weight(u) = \mu_u, \forall u \in X$. As $\sum_{i\in S} \mu_i \geq \varepsilon, \forall S \in \mathcal{R}$, H is a hitting set.

Note that the above LP based approach can be naturally generalized to the weighted case, where we want to minimize

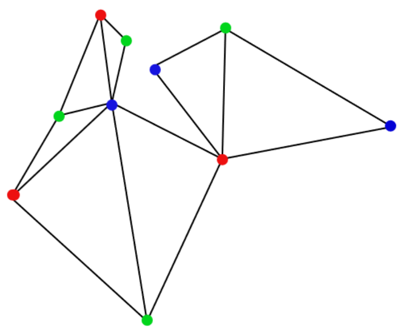
$$J = \sum_{u\in X} w_u \cdot x_u$$

Application: Art Gallery Theorem
The art gallery problem is a classical visibility problem in computational geometry, where one has to guard an art gallery (polygon) with the minimum number of guards (points that together "see" the whole polygon). A set S of points is said to guard a polygon if for every point p in the polygon, there is some $q \in S$ such that the line segment between p and q does not leave the polygon.

For example, 4 points are needed to cover the below polygon.

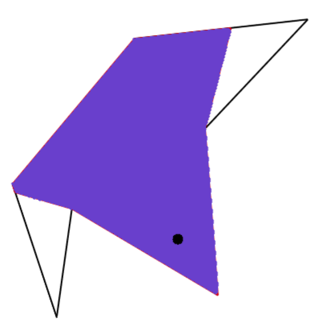


In general, one can show that $n/3$ points always suffice to cover any arbitrary polygon via a colouring argument, as triangulations of planar polygons are always 3 colourable. As seen below, simply pick all points of a given colour to obtain a valid solution. However, the polygon can easily be seen to be covered by 2 points, so this does not necessarily achieve the optimum.



- Consider the range space $S = (P, \mathcal{R})$, where R is the set of visibility polygons $V_p(P)$, $p \in P$.

$$V_p(P) = \{q : q \in P, pq \subseteq P\}$$



Theorem 11. *VC-dimension(S)* $= O(1)$ *(See [2], Section 6.4 for details)*

- We want to cover the entire polygon using the minimum number of visibility polygons. But this is just geometric set cover for the range space S.
- Using the previous algorithms discussed in this lecture, we obtain an $O(\log \mathrm{OPT})$ approximation algorithm.

4 Rectangle Packing Problems

I think packing problems are appealing to mathematicians and computer scientists because they seem very simple – just place these items into the container. Yet they tend to be extremely complicated to actually solve.

- Erik Demaine, Professor at the Massachusetts Institute of Technology.

Bin packing Variant: In this setting we are given n rectangles, and we wish to pack them into minimum number of bins (non overlapping axis - parallel packing).

- For 2D bin packing, where rectangles can be translated both vertically and horizontally, the best known result is a $1.405 = 1 + \ln(1.5)$ asymptotic approximation , and it is known that it has no APTAS. [7].
- In the d - dimensional case, 1.69^{d-1} approximation is possible [8].
- For the Guillotine variant, where there is an extra constraint that the packing must be guillotine separable, an APTAS is known [9]. There is a conjecture that any $2D$ bin packing can be converted into a guillotine separable packing, using at most $4/3$ times the original number of bins, plus a constant.
- In Uniform Round-SAP (Storage allocation Problem), where rectangles can only be moved in one direction (vertically), a $(2 + \varepsilon)$ approximation is known, and it has been shown that there is no APTAS. For arbitrary profiles (not just rectangles), there is an $O(\log \log n)$ approximation [10].
- In Rectangle Colouring (Corresponds to rectangles being fixed), we have to find the chromatic number for the geometric intersection graph defined by the rectangles, and an $O(\log \mathrm{OPT})$ approximation is known for this case [12].

Knapsack Variant: In this setting we are given n rectangles with associated profit, and we wish to pack maximum profit subset of rectangles into a single knapsack (non overlapping axis - parallel packing).

- For 2D knapsack, where rectangles can be translated both vertically and horizontally, the best known result is a 1.89 approximation [13] and it remains open if there exists a PTAS.
- In the $d-$ dimensional case, $(1+\varepsilon)3^d$ approximation is possible [15], and it is open if there is a $poly(d)$ approximation, or even $f(d)$ hardness.
- For the Guillotine variant, where there is an extra constraint that the packing must be guillotine separable, a PPTAS [14] is known, and it remains open if a PTAS is possible.
- In Uniform SAP (Storage allocation Problem), where rectangles can only be moved in one direction (vertically), a 1.969 approximation is known [11].
- In MWISR (Maximum weight independent set of rectangles) (Corresponds to rectangles being fixed), we have to find the maximum weight independent set in the geometric intersection graph defined by the rectangles, and an $O(\log \log n)$ approximation is known[12] .

We will now describe a PTAS for packing squares into a 1×1 unit knapsack, in order to maximise the packed area (Profit = area case).

Problem Statement:

- Given: n squares $I = \{S_1, S_2, \cdots S_n\}$. Square S_i has sidelength s_i.
- Goal: Find axis-parallel nonoverlapping packing of maximum profit (area) subset of squares.

We will use the following high-level strategy.

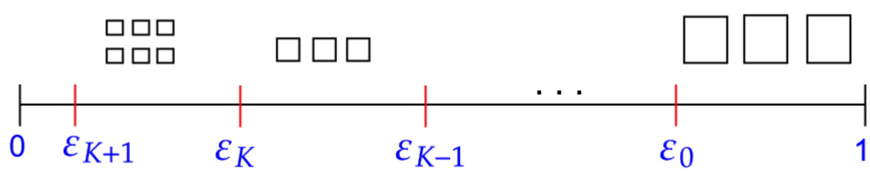
1. Start with an optimal packing P_0.
2. Modify P_0 to obtain a structured packing P_1 such that $area(P_0) \approx area(P_1)$
3. Find packing P_2 in polynomial time such that $area(P_1) \approx area(P_2)$

Item classification and shifting:
For some 2 constants ε_{large} and ε_{small}, define a square S_i to be

- Small if $s_i \leq \varepsilon_{small}$
- Large if $s_i \geq \varepsilon_{large}$
- Medium if $s_i \in (\varepsilon_{small}, \varepsilon_{large})$

Lemma 12. *For any given $\varepsilon > 0$ and positive increasing function f such that $f(x) < x$, there exists ε_{large} such that $\varepsilon > \varepsilon_{large} > f(\varepsilon_{large}) = \varepsilon_{small} = \Omega_\varepsilon(1)$, such that total area of squares with side length in $(\varepsilon_{small}, \varepsilon_{large})$ is at most ε.*

Proof. Take $K = \frac{1}{\varepsilon}$, $\varepsilon_0 = \varepsilon$. Define $\varepsilon_{i+1} = f(\varepsilon_i)$, $0 \leq i \leq K$.



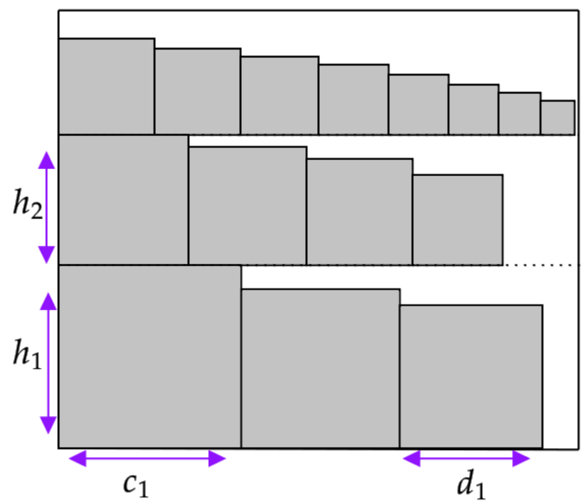
We obtain $K+1$ disjoint ranges $(\varepsilon_i, \varepsilon_{i-1}]$, $\forall i \in [K+1]$. So, there exists i such that the total area of all squares in OPT with side length in $(\varepsilon_i, \varepsilon_{i-1})$ is $\leq \frac{1}{K} = \varepsilon$. So set $\varepsilon_{small} = \varepsilon_i, \varepsilon_{large} = \varepsilon_{i-1}$ ☐

So, from now on we will ignore medium squares, as they contribute at most ε area to OPT.
Packing of large squares:

- By pushing all the large squares to the bottom left, we can see that the number of large squares in OPT is at most $\frac{1}{\varepsilon^2}$, as they don't overlap and have sidelength atleast ε.
- The number of positions for left bottom corners of these squares is $O_{\varepsilon_{large}}(1) \leq (\frac{1}{\varepsilon_{large}}!) \cdot \frac{1}{\varepsilon_{large}})^2$. The bound comes looking at the maximum number of large items in a chain (vertical and horizontal), and all possible permutations of it.
- We can now brute force to find the packing in $n^{O_{\varepsilon_{large}}(1)}$ time by trying all possible packings of large squares, so if all items are large we solve the problem exactly in polynomial time.

So, from now on we have "guessed" all large squares in OPT (note that their packing in our solution might be different from how they were packed in OPT).

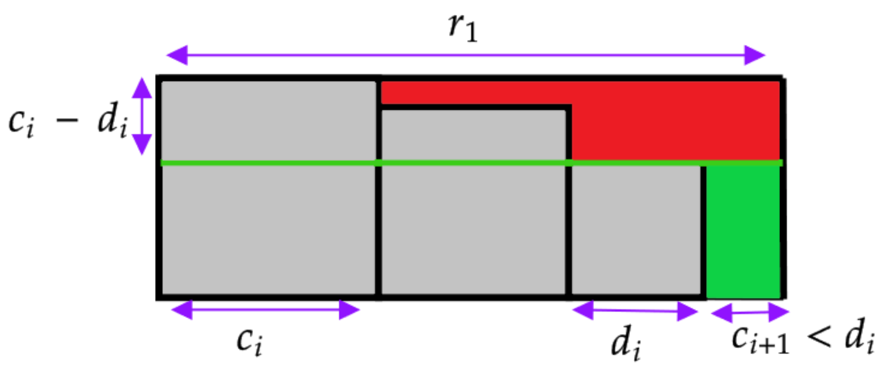
Packing of small squares:
We will need to use NFD (Next-Fit-Decreasing) as a subroutine. The procedure is as follows.



- Sort squares by height.
- Squares are packed left-justified on a level until there is insufficient space at the right to accommodate the next rectangle
- Then we start a new level and proceed.

Lemma 13. *Consider a set of squares S with sidelengths $\leq \delta$. If NFD cannot place any square in a rectangle $R := r_1 \times r_2$, where $r_1, r_2 \geq \delta$, then total wasted space is $\leq \delta(r_1 + r_2)$*

Proof. Let l be the number of shelves in the packing, and the length of the smallest and largest square in level i be c_i, d_i respectively.



- Non increasing order in NFD $\implies c_{i+1} \leq d_i$
- Total wasted part (red)

$$\leq \sum_{i=1}^{l} (c_i - d_i) \cdot r_1 \leq \left(\sum_{i=1}^{l-1} (c_i - c_{i+1}) + c_l - d_l \right) \cdot r_1$$
$$\leq (c_1 - d_l) \cdot r_1 \leq \delta \cdot r_1$$

- Total wasted part (green)

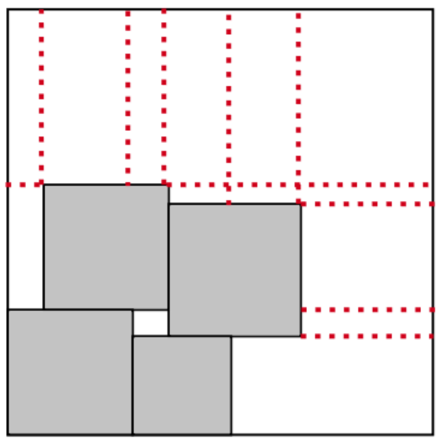
$$\leq \sum_{i=1}^{l} d_i^2 \leq \delta \cdot \sum_{i=1}^{l} d_i \leq \delta \cdot r_2$$

Corollary 14. *If all squares are small, NFD will pack a total area of at least* $\min\{$*total area of squares*$, 1 - 2 \cdot \varepsilon_{small}\}$

Using the above corollary, we immediately get a $2 + \varepsilon$ approximation by packing only large or small squares, and then taking the best packing. ☐

PTAS using Grid - decomposition

- Pack large squares in OPT using brute force.
- Extend their edges to create a grid. Each square creates 2 gridlines vertically and horizontally. So, we get $\leq (2 \cdot N_{large} + 2)^2 = \beta = O_{\varepsilon_{large}}(1)$ many grid cells, where N_{large} is the number of large items.



- For cell $Q := r_1 \times r_2$, if $r_1, r_2 \geq \frac{1}{\varepsilon} \cdot \varepsilon_{small}$, start packing small rectangles in Q by NFD. Else ignore cell Q.
- By the lemma on wasted space in NFD, either we pack all the small items of the total wasted space is

$$\leq \beta \cdot \frac{1}{\varepsilon} \cdot \varepsilon_{small} + \beta \cdot 2\varepsilon_{small} = \beta(\frac{1}{\varepsilon} + 2)\varepsilon_{small}$$

  As an ignored cell can have maximum area $\leq \frac{1}{\varepsilon} \cdot \varepsilon_{small}$, and at most $2 \cdot \varepsilon_{small}$ area can be wasted in a packed cell. We can now choose f such that

$$\beta(\frac{1}{\varepsilon} + 2)\varepsilon_{small} \leq \varepsilon \iff \varepsilon_{small} \leq \frac{\varepsilon^2}{1 + 2\varepsilon} O_{\varepsilon_{large}}(1)$$

- Then, we have only wasted at most ε area, giving us a PTAS.

Variants of square packing:

- For General square packing in $2D$ knapsack, there is a PTAS [16], and an EPTAS [17]
- For $d-$dimensional cube packing into $d-$dimensional knapsack, $d > 2$, a PTAS is known [14].

- An open problem of Meir and Moser asks if the squares of sidelength $1/n$ for $n \geq 1$ can be packed perfectly into a square of are $\sum_{i=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$. It is known that this can be done for all squares of sidelength n^{-t}, $\frac{1}{2} < t < 1$, $n \geq n_0$, for any n_0 sufficiently large depending on t. (In this case we need to pack perfectly into a square of area $\sum_{i=n_0}^{\infty} \frac{1}{n^{2t}}$) [18].

References

[1] "Course notes for E0 234: Introduction to Randomized Algorithms." [Online]. Available: https://www.csa.iisc.ac.in/~arindamkhan/courses/RandAlgo21/RandAlgo.html#lectures

[2] S. Har-peled, "Geometric Approximation Algorithms," 2008. [Online]. Available: https://sarielhp.org/book/

[3] M. Mitzenmacher and E. Upfal, "Probability and Computing," in *Probability and Computing*, 2005.

[4] J. Matousek, E. Welzl, and R. Seidel, "How to net a lot with little: small ε-nets for disks and halfspaces," pp. 16–22, 1990.

[5] B. Aronov, E. Ezra, and M. Sharir, "Small-size ε-nets for axis-parallel rectangles and boxes," *Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 639–648, 2009.

[6] G. Even, D. Rawitz, and S. Shahar, "Hitting sets when the VC-dimension is small," *Information Processing Letters*, vol. 95, no. 2, pp. 358–362, 2005.

[7] N. Bansal and A. Khan, "Improved approximation algorithm for two-dimensional bin packing," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 13–25, 2014.

[8] A. Caprara, "Packing 2-dimensional bins in harmony," *Annual Symposium on Foundations of Computer Science - Proceedings*, pp. 490–499, 2002.

[9] N. Bansal, A. Lodi, and M. Sviridenko, "A tale of two dimensional bin packing," *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, vol. 2005, pp. 657–666, 2005.

[10] A. W. Debajyoti Kar, Arindam Khan, "Approximation Algorithms for Round-UFP and Round-SAP."

[11] T. Mömke and A. Wiese, "Breaking the barrier of 2 for the storage allocation problem," *Leibniz International Proceedings in Informatics, LIPIcs*, vol. 168, 2020.

[12] P. Chalermsook and B. Walczak, "Coloring and maximum weight independent set of rectangles," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 860–868, 2021.

[13] W. Galvez, F. Grandoni, S. Heydrich, S. Ingala, A. Khan, and A. Wiese, "Approximating geometric knapsack via l-packings," *Annual Symposium on Foundations of Computer Science - Proceedings*, vol. 2017-October, pp. 260–271, 2017.

[14] K. V. N. S. Klaus Jansen, Arindam Khan, Marvin Lira, "A PTAS for Packing Hypercubes into a Knapsack." [Online]. Available: https://arxiv.org/abs/2202.11902

[15] A. Khan, A. Maiti, A. Sharma, and A. Wiese, "On guillotine separable packings for the two-dimensional geometric knapsack problem," *Leibniz International Proceedings in Informatics, LIPIcs*, vol. 189, 2021.

[16] K. Jansen and R. Solis-Oba, "A polynomial time approximation scheme for the square packing problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5035 LNCS, pp. 184–198, 2008.

[17] S. Heydrich and A. Wiese, "Faster approximation schemes for the two-dimensional knapsack problem," *ACM Transactions on Algorithms*, vol. 15, no. 4, p. , 2019.

[18] T. Tao, “Perfectly packing a square by squares of nearly harmonic sidelength.” [Online]. Available: https://arxiv.org/abs/2202.03594