Lecture 1

Instructor: Arindam Khan, Anand Louis *Scribe: Sarvagya Jain, Prashant Gokhale*

1 Introduction

We will study some geometric variants of NP-hard problems such as geometric set-cover, geometric hitting set, independent set/coloring number of geometric intersection graphs, etc.

Definition 1 (Geometric Intersection Graph)**.** *A graph in which vertices are geometric objects (say, disks, cubes, hyperplanes) and two vertices share an edge iff the corresponding objects intersect is called a geometric intersection graph.*

The intersection of graph theory with geometry is a deeply fascinating field. There are many beautiful results in this field. One such result is Koebe's theorem which we state without proof.

Theorem 2 (Koebe's Theorem)**.** *Let G be a planar graph. Then one can assign to each node i a circle C_i in the plane so that their interiors are disjoint, and two nodes are adjacent if and only if the corresponding circles intersect. A graph is planar iff it is a coin graph.*

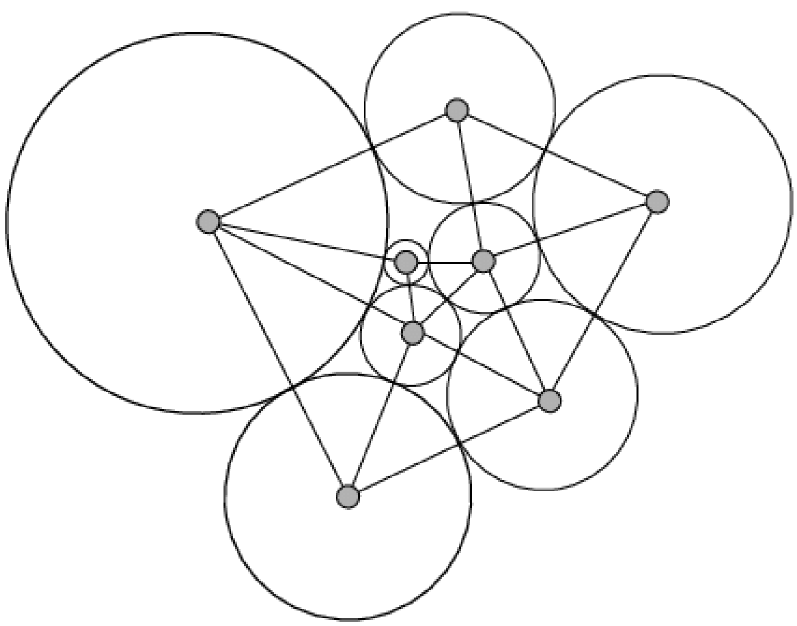


Figure 1: The coin representation of a planar graph

2 Maximum independent set of geometric objects

Given a collection of n objects I, we need to find the maximum cardinality subset $S \subset I$ such that no two objects in S overlap. In the weighted version, objects have weights and we need to find the subset S with maximum weight (along with the non-overlapping condition). In graph theoretic setting, independent set is a set of vertices in a graph such that no two vertices are adjacent. This is a classical NP-Hard problem. It is trivial to get an n approximation (just choose any one vertex). Surprisingly, it is NP-Hard to even get a $n^{1-\varepsilon}$ approximation (Hastad' 99).

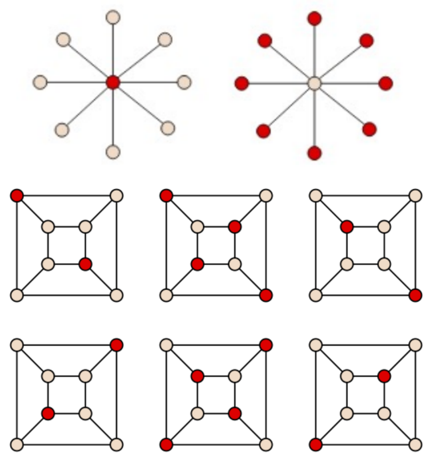


Figure 2: The vertices marked in red represent the independent sets.

In a geometric intersection graph, nodes correspond to geometric objects and there is an edge (u, v) if the objects corresponding to u and v overlap.

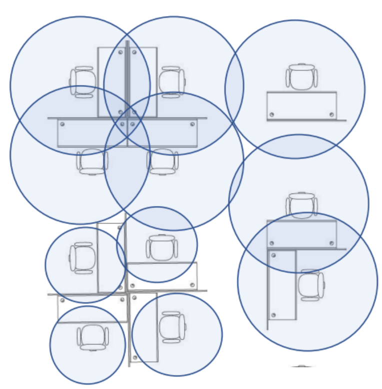


Figure 3: Intersecting Objects

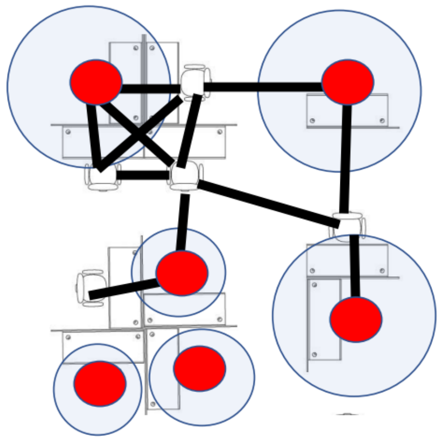


Figure 4: Geometric intersection graph

We first consider the simplest case where objects are intervals. In the unweighted case, we have a well known greedy algorithm which relies on picking the interval that has the smallest right endpoint. In the weighted case, we have a well known dynamic program. Both these strategies give us the required set in polynomial time.

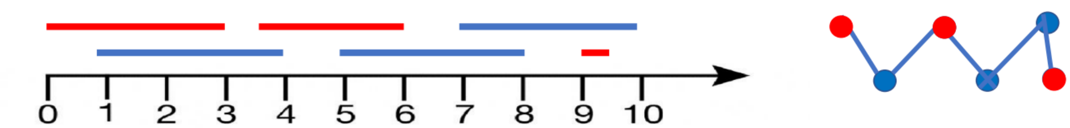


Figure 5: Maximum independent set for a collection of intervals (the intervals marked in red represent the maximum independent set).

We can generalise the greedy algorithm to other object as well.

Algorithm 1: Greedy algorithm for Geometric Independent Set

Data: The collection of objects I

Result: Maximum Independent Set

Initialise S to ϕ (empty set);

while I *is not empty* **do**

- Pick smallest object (say based on area) $i \in I$. Insert i to S;
- Remove i and all objects I_i intersecting i from I;

end

return S;

This is a polynomial time algorithm and clearly returns an independent set.

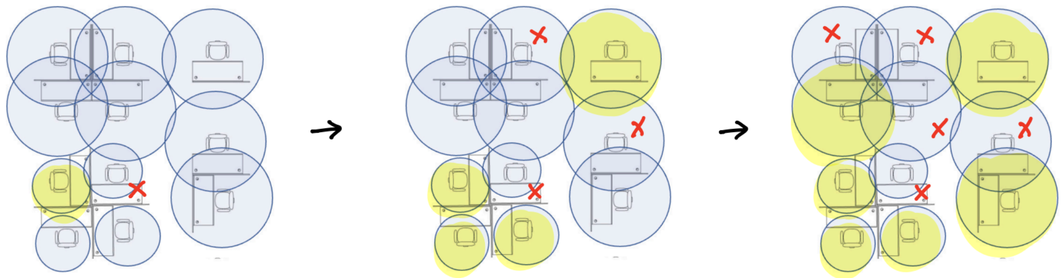


Figure 6: Simulation of the greedy algorithm on an instance.

Theorem 3. *For squares, the greedy algorithm returns a 4-approximation.*

Let S^* be an optimal independent set. Want to show $|S| < 4|S^*|$. What could be a good lower bound on S^*?

Let P^* be minimum piercing set, i.e., minimum cardinality set of points such that each square is stabbed by at least one point in the set.

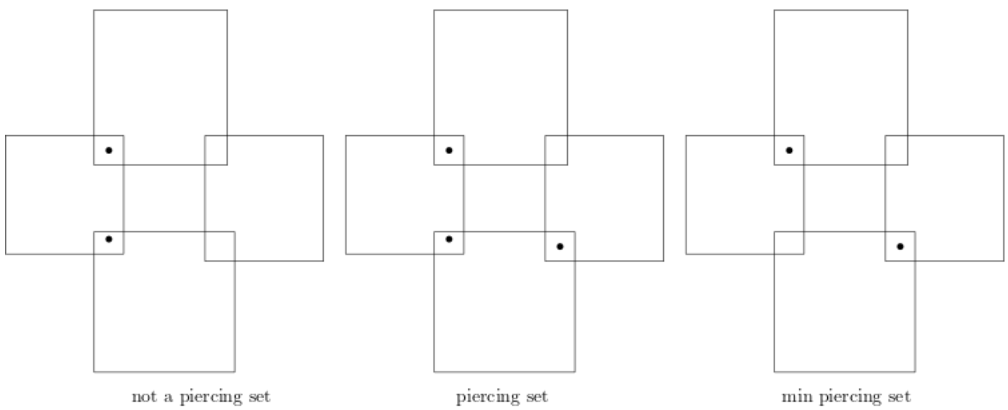


Figure 7: Examples and non-examples of piercing sets.

Observe that independence number is less than equal to piercing number. The squares in S^* are disjoint. So we need at least $|S^*|$ points to stab squares in S^*. Observe that if square i is intersected by a bigger square j, then j is stabbed by at least one corner of i (see Figure 8).

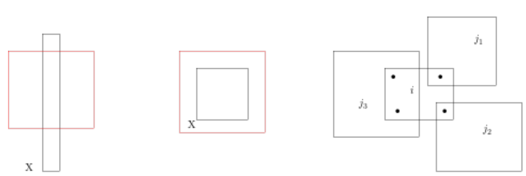


Figure 8: Representation of the key observation

As we took squares in non-decreasing order of size, therefore taking union of all 4 corners of squares in S we get a piercing set. Hence, we have $|P^*| \leq 4|S|$ (P^* is a minimum piercing set).

Combining, we get $|S^*| \leq |P^*| \leq 4|S|$. Hence, greedy algorithm gives us a 4-approximation for squares.

We can extend this algorithm to disks by running the same greedy algorithm. Every larger disk j intersecting a fixed smaller disk i can be stabbed by 16 points (see Figure 9). Thus, the same argument given earlier shows that we have a 16-approximation for disks.

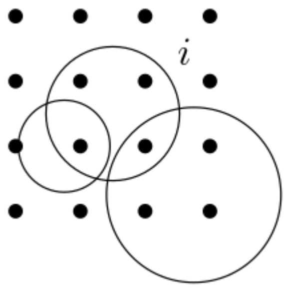


Figure 9: Stabbing by 16 points

Exercise 4. *Extend this to obtain an $O(1)$-approximation for fat objects (that is, convex objects for which the ratio of the radii of min-radius circumscribing disk and max-radius inscribing disk is $O(1)$).*

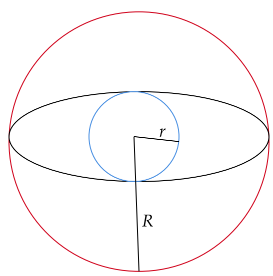


Figure 10: For fat objects, the ratio R/r is bounded above by a constant.

The above method fails to give us an $O(1)$-approximation for:

- weighted squares
- (unweighted/ weighted) rectangles (see Figure 11).

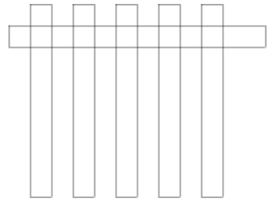


Figure 11: An example in which we require $\Omega(n)$ points to pierce all rectangles.

2.1 Four important "numbers" from combinatorial properties

Let F be a finite family of sets. $G(F)$ is the intersection graph.

- Transversal Number $\tau(F)$: Minimum number of points meeting all sets in F.
- Independence (stability) number $\alpha(F)$: Maximum number of pairwise disjoint sets in F.
- Clique number $\omega(F)$: Maximum number of pairwise intersecting sets in F.
- Coloring number $\chi(F)$: Minimum number of classes in a partition of F into pairwise disjoint sets.

We saw that $\alpha(F) \leq \tau(F)$ and for squares $\tau(F) \leq 4\alpha(F)$. Also for intervals, $\tau(F) = \alpha(F)$. There is a famous conjecture that $\tau(F)$ is $O(\alpha(F))$ for rectangles [Gyarfas, Lehel 1985]. There is an even stronger conjecture stating that $\tau(F) \leq 2\alpha(F) - 1$ for rectangles [Wegner 1965].

Best known bound we have for rectangles is $\lceil \frac{5}{3}\alpha \rceil \leq \tau \leq c\alpha \log \alpha$, where c is some constant [Fon-Der-Flaas and Kostochka '93], [Karolyi '91].

Exercise 5. *Let $F = \{R_1, R_2, ...R_n\}$ be a family of rectangles in the plane, each rectangle R_i has width w_i and height h_i and $w_i \geq h_i$. Let K be the maximum aspect ratio, then $\tau(F) \leq 2(K+1)\alpha(F)$.*

Exercise 6. *If F is a family of unit squares in $\mathbb{R}^2$, then $\tau(F) \leq 2\alpha(F)$.*

Note that $\omega(F) \leq \chi(F)$ for rectangles as each rectangle in a clique needs a different colours.

Conjecture: $\chi(F) = O(\omega(F))$ for rectangles. The conjecture if true will imply an $O(1)$-approximation algorithm for maximum weight independent set of rectangles (MISR).

Best known bounds: $3\omega \leq \chi \leq O(\omega \log \omega)$ by [Hendler '98] and [Chalermsook-Walczak '21] respectively.

The graphs which admit $\chi(F) = O(\omega(F))$ are called χ-bounded [survery, Scott-Seymour '20].

3 Shifted Grids

This technique was first introduced by Hochbaum-Maass in 1985. We will give a PTAS to find the maximum independent set of unit disks/squares using this technique. The technique can also be extended to fat objects with similar size constraints.

3.1 Algorithm

The basic idea is similar to a divide and conquer algorithm.

The algorithm for unit disks goes as follows:

- Take a large number $b = \Theta(1/\varepsilon)$.
- Take a random shifting parameter $l \in \{0, 1, \ldots, b-1\}$, uniformly at random.
- Form a grid of side length b, such that the grid contains the lines $x = l$ and $y = l$. In other words, consider the grid formed by the lines $\{x = l + mb : m \in \mathbb{Z}\} \cup \{y = l + nb : n \in \mathbb{Z}\}$.

- Remove all the objects that intersect the gridlines.
- Solve sub-problems inside each grid-cell by brute force.

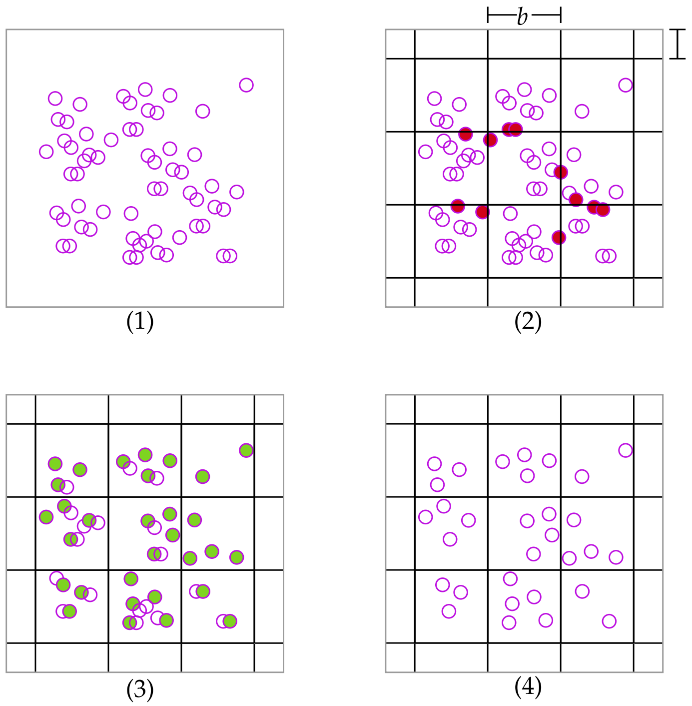


Figure 12: Run of the shifted grid algorithm on an instance. Solid red circles in (2) intersect the grid lines, hence are removed. Solid green circles in (3) represent the independent sets.

Arguing using area, in a $b \times b$ square, the maximum independent set has at most $b^2/(\pi/4)$, i.e., $O(b^2)$ disks. So one can solve the sub-problem in a $b \times b$ square in $O(n_i^{O(b^2)})$ time by going over all the subsets of the set of n_i disks in the i^{th} grid-cell having at most $O(b^2)$ disks $\left(\sum_{j=0}^{O(b^2)} \binom{n_i}{j} = O\left(n_i^{O(b^2)}\right)\right)$.

Thus, using $\sum_i a_i^k \leq (\sum_i a_i)^k$ for $a_i \geq 0$, the total run-time is $O\left(\sum_i n_i^{O(b^2)}\right) = O\left(n^{O(b^2)}\right)$.

3.2 Analysis

Let T^* be the optimal solution for an instance I. Fix a disk $t \in T^*$. The probability t crosses a horizontal gridline is $\frac{1}{b}$. This is because a unit disk can cross only one of $y = 0, y = 1, \ldots, y = b - 1$ (see Figure 13). Similarly one gets the result for a vertical gridline. Using union bound,

$$\mathbf{P}[t \text{ crosses grid boundary}] \leq \mathbf{P}[t \text{ crosses horizontal grid boundary}] + \mathbf{P}[t \text{ crosses vertical grid boundary}]$$

$$\implies \mathbf{P}[t \text{ crosses grid boundary}] \leq \frac{1}{b} + \frac{1}{b} = \frac{2}{b}.$$

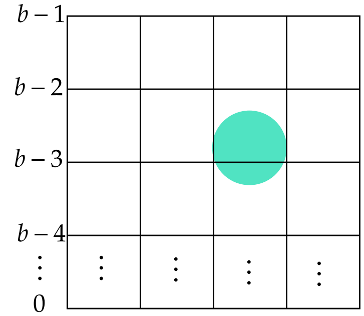


Figure 13: A unit disk can intersect the horizontal grid for at most 1 out of the b different values of l.

Let T be the solution returned by the algorithm. Let S be the disks in I that crossed the gridlines. Then,

$$\mathbb{E}[|T|] = \mathbb{E}[OPT(I \setminus S)]$$

As $OPT(I) \setminus S$ is a feasible solution of $OPT(I \setminus S)$, therefore

$$\mathbb{E}[OPT(I \setminus S)] \geq \mathbb{E}[OPT(I) \setminus S] = \mathbb{E}[T^* \setminus S]$$

Now note that $T^* \setminus S$ is the set of objects in T^* that do not cross any gridlines. Hence using linearity of expectation and the bound on probability of an object crossing a gridline, we get

$$\mathbb{E}[|T|] \geq \mathbb{E}[T^* \setminus S] \geq \left(1 - \frac{2}{b}\right) |T^*|.$$

As $b = \Theta\left(\frac{1}{\varepsilon}\right)$, we get an $(1+\varepsilon)$-approximation in $O\left(n^{O(1/\varepsilon^2)}\right)$ time (a PTAS).

We conclude this section with the following exercise:

Exercise 7. *Extend the above algorithm to the weighted setting. Also, derandomize the above algorithm.*

3.3 Some related problems

- Hitting set/'Piercing' (continuous version): Given a set of n objects I, find the minimum cardinality set of points S^* that stabs all objects in I.
- Hitting set (discrete version): Given a set of n objects and a set of m points P (can be weighted), find the minimum weight subset S^* of P that stabs all objects in I.
- Geometric set cover (discrete version): Given a set of m objects I (can be weighted) and a set of n points P , find minimum weight subset S^* of I that covers all points in P.

- Geometric set cover (continuous version): Given a set of n points P, find the minimum number of objects from a given class of objects (for example, unit disks) that cover all points in P.

Remark Shifting gives us PTAS for piercing and geometric set cover problems for unit squares. Additionally, it generalizes to higher dimensions.

We have the following theorem due to Hochbaum and Maass which we state without proof:

Theorem 8. *There exists a PTAS for the geometric set cover problem in which we cover the points using the class of axis aligned d-dimensional hyper-cuboids of fixed dimensions, i.e., for every integer $l \geq 1$ we get a $(1+1/l)^d$ approximation to the problem in $O\left(l^d \cdot n^{2l^d+1}\right)$ time.*

4 Quadtrees

Shifted Quadtrees were introduced by [Erlebach-Jansen-Siedel '01/Chan '01]. They are a generalisation of shifted grids that can combat the problem of squares/ disks having different sizes.

Shifted hierarchial subdivision [EJS]:

- Scale disks so that the largest disks have diameter 1.
- Partition disks into $l = \left\lfloor \log_{k+1} \frac{1}{d_{min}} \right\rfloor$ levels: Disk i belongs to level j if $(k+1)^{-j} \geq d_i > (k+1)^{-j-1}$, where d_i denotes the diameter of the i^{th} disk.
- Hierarchial Grid: Choose a random shift parameter (r,s), $0 \leq r, s \leq k$. For each level $j \in \{0, 1, \ldots, l\}$ define horizontal gridlines- $r(k+1)^{-j}$, $(k+1)^{-j+1} + r(k+1)^{-j+1}$, $\ldots$ and so on. Define vertical gridlines- $s(k+1)^{-j}$, $(k+1)^{-j+1} + s(k+1)^{-j+1}$, $\ldots$ and so on.

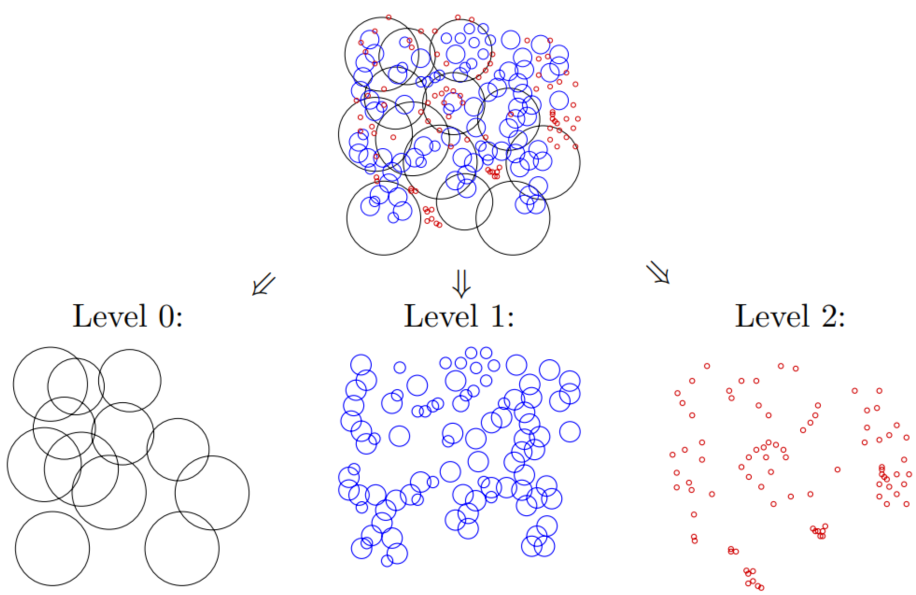


Figure 14: Partitioning the disks into levels ($k=2$).

Observe that every disk can hit at most 1 horizontal and at most 1 vertical gridline on it's level.

Delete all disks that hit the gridlines.

Lemma 9. *$\mathbf{P}$[a disk survives] $= \left(1 - \frac{1}{k}\right)^2$.*

Proof. Since the probability of the disk being deleted due to horizontal and vertical lines each is $\frac{1}{k}$, we can write $\mathbf{P}$[a disk survives] $= \left(1 - \frac{1}{k}\right)^2$ using independence. □

So, we can use these grids to obtain simpler subproblems by incurring only a $1 - \left(1 - \frac{1}{k}\right)^2$ factor loss. These subproblems can be solved using dynamic program in $n^{O(\frac{1}{\varepsilon^4})}$ time.

Remark Chan used Quadtree to give $n^{O(\frac{1}{\varepsilon})}$ time. In fact, in d-dimensions it takes $n^{O(\frac{1}{\varepsilon^{d-1}})}$ time.

Definition 10 (r-grid interval)**.** *An interval of the form $[ri, r(i+1)]$, where $i \in \mathbb{Z}$.*

Definition 11 (r-grid cell)**.** *A cell that can be written as cartesian product of d r-grid intervals, where d is the number of dimensions.*

Definition 12 (Quadtree Cell)**.** *It is the union of r-grid cells for $r = 2^{-l}$, where $l \in \mathbb{N} \cup \{0\}$.*

Definition 13 (k-aligned)**.** *We say that a disk of diameter r is k-aligned if it is inside a quadtree cell of size $\leq kr$.*

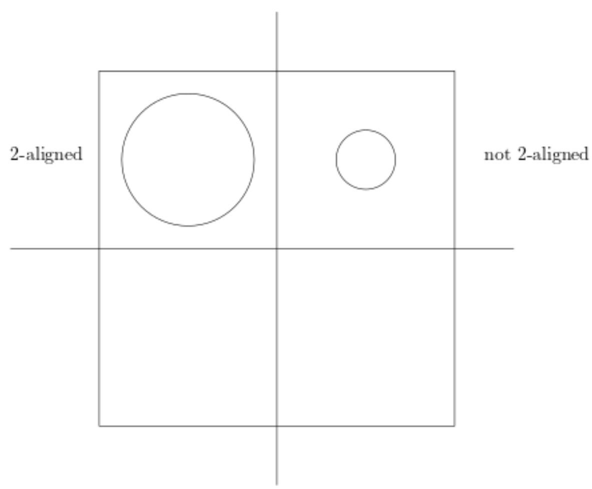


Figure 15: Example and non-examples of 2-aligned disks.

Intuitively, the disks are large compared to the cell. If all items are k-aligned, it can be solved in polynomial time by divide and conquer with DP.

Lemma 14. *If all disks are k-aligned, then maximum independent set can be solved exactly in $n^{O(k)}$ time and space.*

Remark This lemma can be extended to fat objects and d-dimensions.

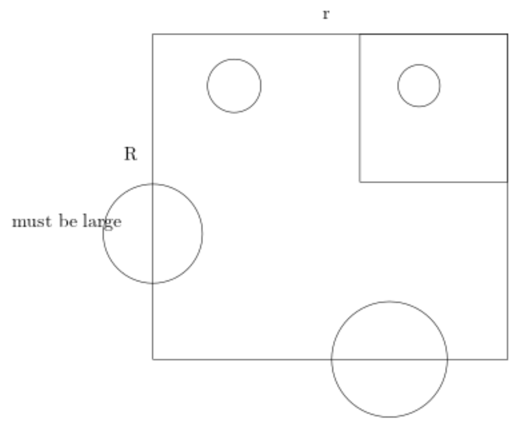


Figure 16: Representation of the proof

Proof. Let R be a quadtree cell of size r (as in the figure). Say disk S intersects R. From definition of k-alignedness, if S has size $< \frac{r}{k}$, it must be inside a cell of size r. That is, S cannot intersect boundary of R. Hence, disks intersecting the boundary have size $\geq \frac{r}{k}$. So, the boundary can be intersected by at most $O(k)$ disks of size $\geq \frac{r}{k}$.

So, OPT can have $O(k)$ disks intersecting the boundary. So, we guess these $O(k)$ disks and then apply divide and conquer inside. □

Definition 15 (Subroutine *pack*$[R, B]$)**.** *Given some fixed set of disks B that intersects boundary of R, pack$[R, B]$ is the maximum cardinality subset of non-overlapping disks lying completely inside R such that they are disjoint from B.*

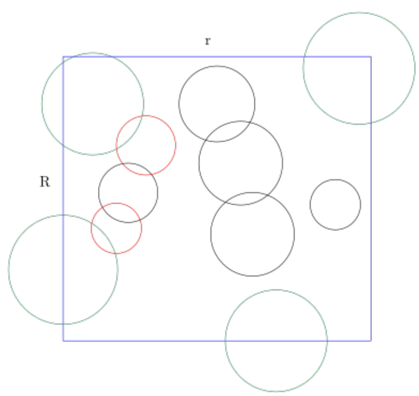


Figure 17: *pack*$[R, B]$.

Recursive formula (for DP): Let $\hat{R}$ be the smallest quad tree cell that contains centers of all disks in R. Split $\hat{R}$ into 4 (2^d for d dimensional case) quadtree subcells $\{R_i\}_{i=1}^4$.

B are fixed disks (drawn in black). B' is some set of disks disjoint from B in which the disks intersect $\cup_i \partial R_i$ but not ∂R (drawn in green).

Trying out all interface combinations, $pack[R, B] = \max_{B'} \left(\sum_{i=1}^4 \text{pack} \left[R_i, \, (B' \cup B)|_{\partial R_i} \right] + |B'| \right)$.

We have already seen that B has $O(k)$ disks.

Similarly. B' can have at most $4 \times O(k) = O(k)$ disks. Hence, number of possible choices for B and B' is $n^{O(k)}$. Since, the number of centers in R_i is less than centers in $\hat{R}$ (as $\hat{R}$ was smallest quadtree cell, here we have used property of quadtree). Number of quadtree cells generated by the recursion is $O(n)$. Hence, total time for bottom up computation of $DP = n^{O(k)}$ time.

We make all/many disks k-aligned by shifting.

Lemma 16. *Fix an odd number $k > 2$. Take j uniformly at random from $\{0, 1, \ldots, k-1\}$ & define $v^{(j)} = (j/k, j/k)$ Then for any object S inside $[0,1)^2$, the shifted object $S + v^{(j)}$ is $2k$-aligner with probability $(1 - 2/k)$.*

Proof. In this lemma, we are shifting the object and fixing the grid. Say s has center $p = (p_1, p_2)$ and diameter r.

Case A (large object): $r > 1/k$, then $S + v(j) \subseteq [0,2)^2 \forall j$. Then $s + v^{(j)}$ is $2k$-aligned as it is contained in quadtree cell of size $< 2kr$.

Case B (small objects): $r \leq 1/k$. Say, $2kr \in \left(2^{-l}, 2^{-l+1}\right] \forall l \in \mathbb{N}$. If $s + v^{(j)}$ is not $2k$-aligned, the disk with center $p + v^{(j)}$ and size r cannot be inside 2^{-l} size grid cell.

So for some $i \in \{1,2\}$. the interval with center $P_i + j/k$ and length r is not inside any 2^{-l} - grid interval.

Equivalently, we can think of a fixed disk of diameter r & random shifted gridline with shift j/k.

We have that $\mathbf{P}$[Disk is cut by a gridline] $\leq (r/(1/k))/k \leq 1/k$. The last inequality follows as $r \leq 1/k$.

$\mathbf{P}$[Disk is cut by vertical or horizontal gridline] $\leq 2/k$. Hence, it is $2k$-aligned with probability at least $1 - 2/k$. □

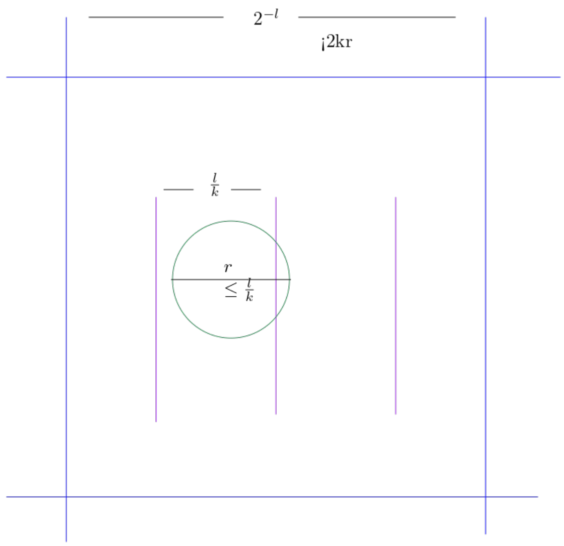


Figure 18: Representation of proof

Theorem 17. $(1+\varepsilon)$ *approximation in $n^{0(1/\varepsilon)}$ time and space.*

Proof. Use random shift for objects, then use $k = 2/\varepsilon$. Delete objects that are not $2k$-aligned in $n^{o(1)}$ time (by loosing $2/k \approx \varepsilon$ fraction of profit.) Then, in $n^{o(k)}$ time solve $2k$-aligned case. □

Remark The above algorithm:

- can be derandomized.
- works for weighted, arbitrary fat objects.
- does not work for piercing.

5 Guillotine Cuts

5.1 Definitions

Definition 18 (Guillotine cut)**.** *An end-to-end cut along a straight line to divide a (rectangular) piece into two smaller pieces.*

Definition 19 (Guillotine Cutting Sequence)**.** *A series of guillotine cuts, each cut separating a sub-piece into two new sub-pieces (see Figure 19).*

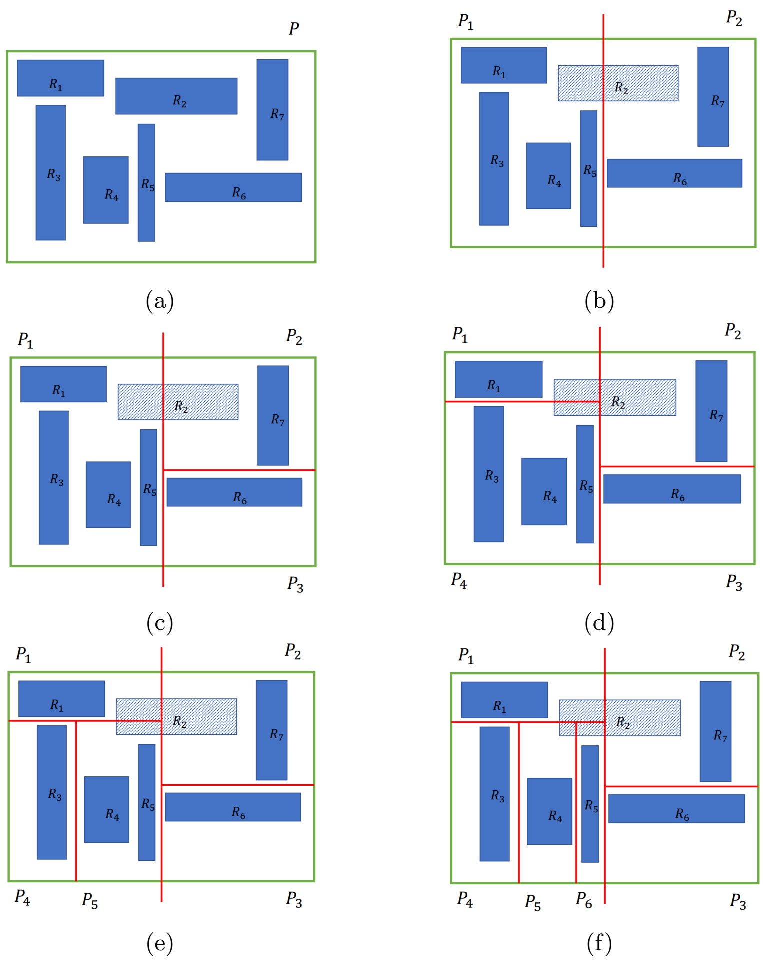


Figure 19: Example of a Guillotine Cutting Sequence.

A cutting sequence can be naturally imagined as a binary tree (see Figure 20):

- Each node corresponds to a rectangular region.
- Each non-leaf node (say the one corresponding to region P) contains two children (corresponding to P_1, P_2 obtained by a guillotine cut in P).
- Each leaf node contains exactly one item.

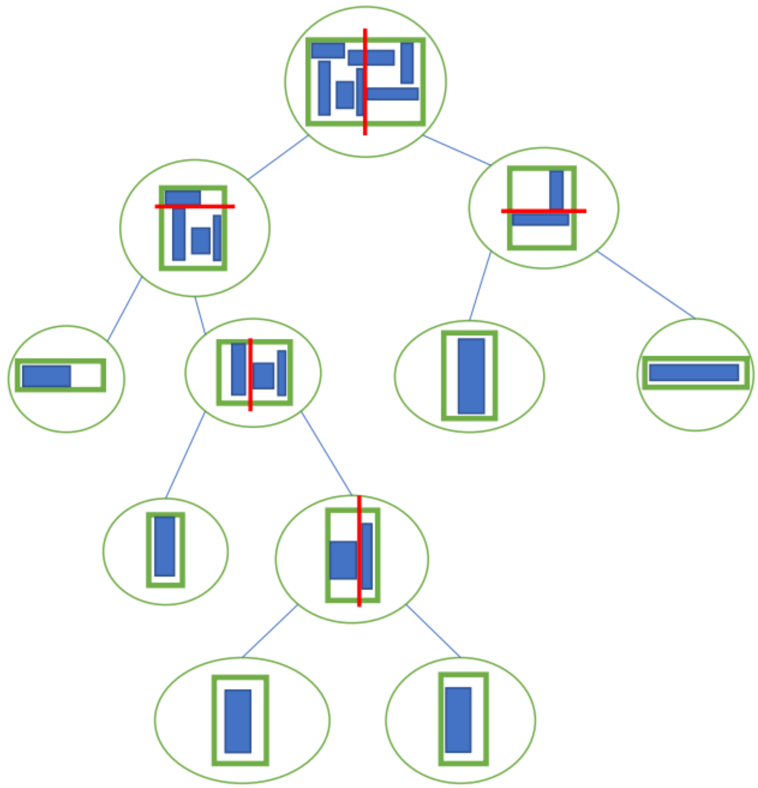


Figure 20: Viewing Guillotine Cutting Sequence as a binary tree.

Definition 20 (Killed Rectangle)**.** *We say that a rectangle is killed if a guillotine cut in the guillotine cutting sequence passes through that rectangle.*

Definition 21 (Extracted Rectangle)**.** *A rectangle is extracted if it is not killed and is the only rectangle in its sub-piece.*

Definition 22 (Guillotine Separable)**.** *A given configuration is guillotine separable if all rectangles can be extracted using some cutting sequence.*

It is not always possible to separate out all rectangles using a sequence of guillotine cuts (see Figure 21).

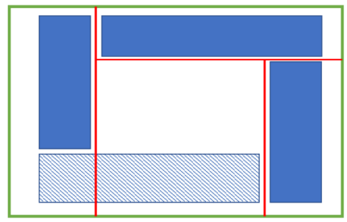


Figure 21: A configuration which is not Guillotine Separable. Observe that it suffices to consider the cuts passing along the edges of rectangles.

So we ask for the next best thing, i.e., is it possible to separate out a constant fraction of rectangles using a sequence of guillotine cuts? Pach-Tardos [SoCG'00] conjectured it to be true.

Some results pertaining to the conjecture:

- Given n rectangles, there are instance where we can not separate out $> n/2$ rectangles using a sequence of guillotine cuts [Abed et al., APPROX'15].

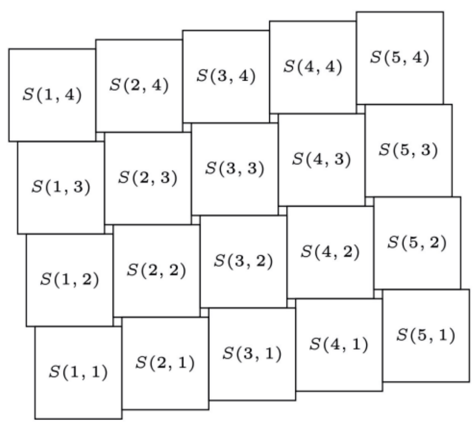


Figure 22: An example in which we cannot separate out $> n/2$ rectangles.

- There is an algorithm to separate out $>$ n/$(\log n + 1)$ rectangles using a sequence of guillotine cuts [K., Reddy, APPROX'20].

5.2 Algorithm

Now we will try to get an $n/(\log n + 1)$-approximation algorithm for the maximum independent set of rectangles (MISR) problem using guillotine cuts.

Though the rectangles in OPT of MISR are non-overlapping, they may not be guillotine separable. Suppose we have that for any embedding of n non-overlapping rectangles, there are n/α fraction of rectangles separable by guillotine cuts. Then, there is a set of cardinality $\frac{|OPT|}{\alpha}$ rectangles that are guillotine separable. We will show a Dynamic Program (DP) that returns optimal guillotine separable set for a MISR instance in $O\left(n^5\right)$

time. DP returns guillotine separable set R' and by optimality, $|R'| \geq \frac{|OPT|}{\alpha}$. Guillotine separable rectangles are obviously independent and hence this gives us a α-approximation for MISR.

Remark If we assume Pach-Tardos Conjecture, this procedure results in a 2-approximation.

Given an instance of MISR, it is easy to convert it into an instance in which all rectangle corners have integral coordinates in $[0, 2n-1] \times [0, 2n-1]$ without changing anything related to guillotine separability (see Figure 23). Furthermore, this can be done in $O(n^5)$ time.

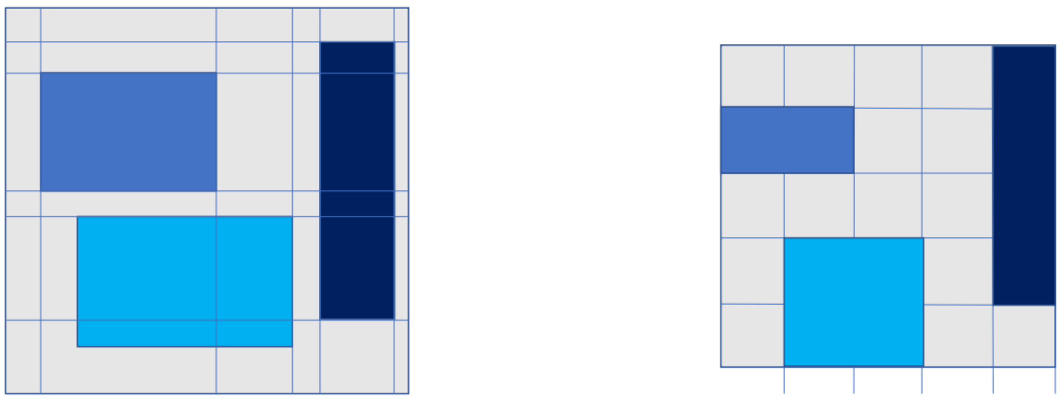


Figure 23: Processing before the DP.

Having done this processing, we now use the following DP:

- $DP[C]$ stores optimal guillotine separable set for rectangular region C.
- Base cases (see Figure 24):
  - If C coincide with a rectangle R, give $\{R\}$ as solution.
  - If C contains no rectangle, give ϕ as solution.

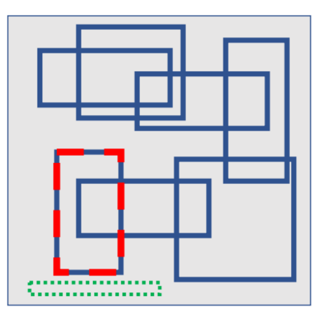


Figure 24: Illustration of the base cases of DP.

- Recurrence (see Figure 25): $DP[C] = DP[C_1] \cup DP[C_2]$, where $|DP[C_1]| + |DP[C_2]|$ is maximum among all partitions C_1, C_2 of C by some guillotine cut passing across C.

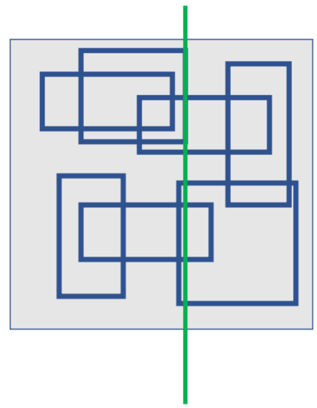


Figure 25: The green line is traversed across and the quantity $|DP[C_1]| + |DP[C_2]|$ is calculated for each partition. Similarly one does this along the y-direction with a horizontal line. We choose the C_1 and C_2 for which $|DP[C_1]| + |DP[C_2]|$ is maximised.

For a given DP cell C, there are $O(n)$ possible cuts. To bound the number of DP cells, we bound the number of rectangular regions in a $2n \times 2n$ grid. For forming a rectangle we need to choose two diagonally opposite points and for each point we have at most $(2n)^2$ possibilities (at most $2n$ choices for each coordinate). Thus, there are $O\left(n^4\right)$ DP cells and hence we get an $O\left(n^5\right)$ run time.

5.3 Analysis

First we make the following observations:

- Observation 1: If all rectangles intersect a straight line then they are guillotine separable (see Figure 26).

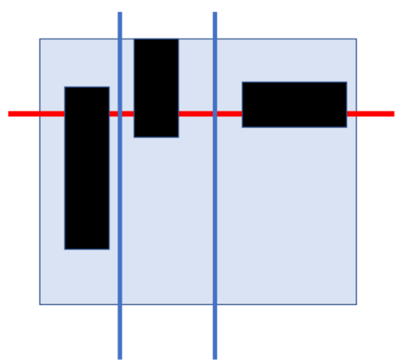
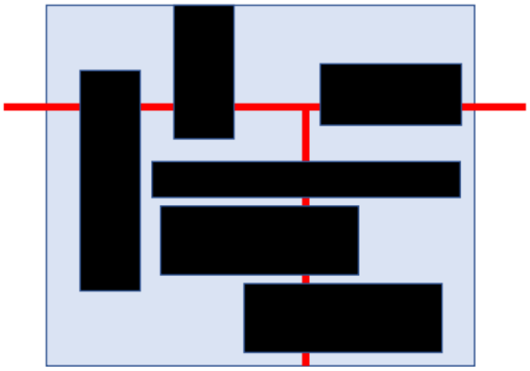
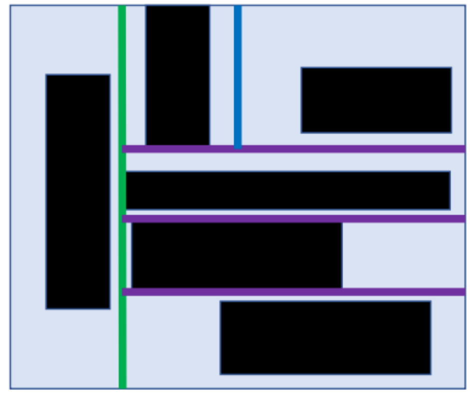


Figure 26: All rectangles intersect the red line. The blue guillotine cuts separate these rectangles.

- Observation 2: If all rectangles intersect a ”T” then they are guillotine separable (see Figure 27).



(a)



(b)

Figure 27: All rectangles intersect the red "T" in (a). In (b), we can see the cuts separating them.

Suppose we have rectangles embedded in $[0, 2n] \times [0, 2n]$ grid with integral corners. The idea is to partition the rectangles into $\log n + 1$ guillotine separable sets. The following procedure gives us such a partition:

- For level i, introduce vertical pole lines at $(2j + 1)n/2^{i-1}$ for all j satisfying $0 \leq j \leq 2^{i-1} - 1$.

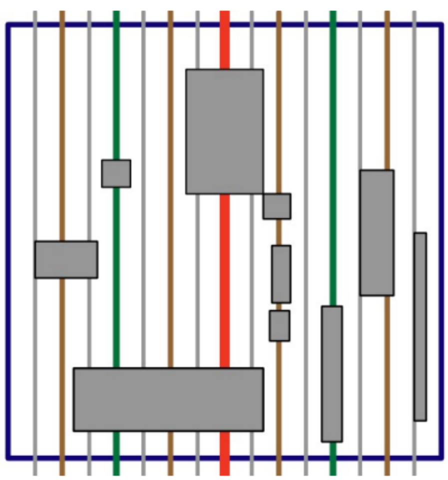


Figure 28: In this example, red line represents the pole line for level 1, green lines represent the pole lines for level 2, brown lines represent the pole lines for level 3 and silver lines represent the pole lines for level 4.

- We introduce pole lines for levels 1 to $\log n + 1$. This ensures that for every rectangle, there is a pole line among the union of pole lines intersecting it.
- The union of all pole lines from level 1 to i divides the plane into 2^i equal partitions.
- Let level of a rectangle be the smallest level i such that some level-i pole line intersects the rectangle.
- By partitioning the rectangles based on their levels, we get a partition into $(\log n + 1)$ color classes.

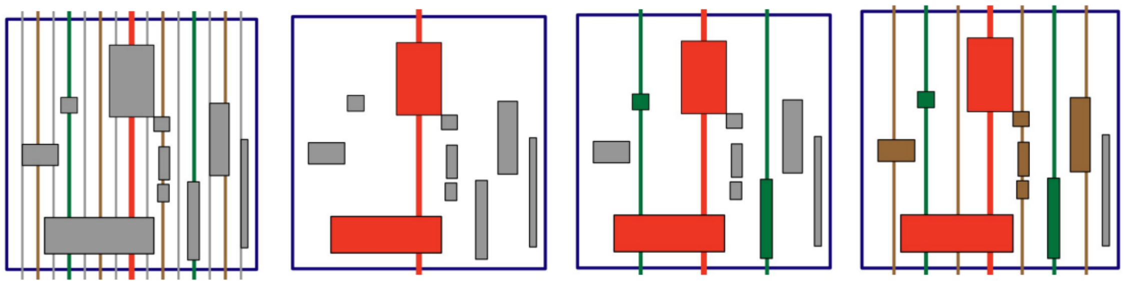


Figure 29: Classifying rectangles into different colour classes based on their levels.

- It is easily seen that each color class is guillotine separable (see Figure 30).

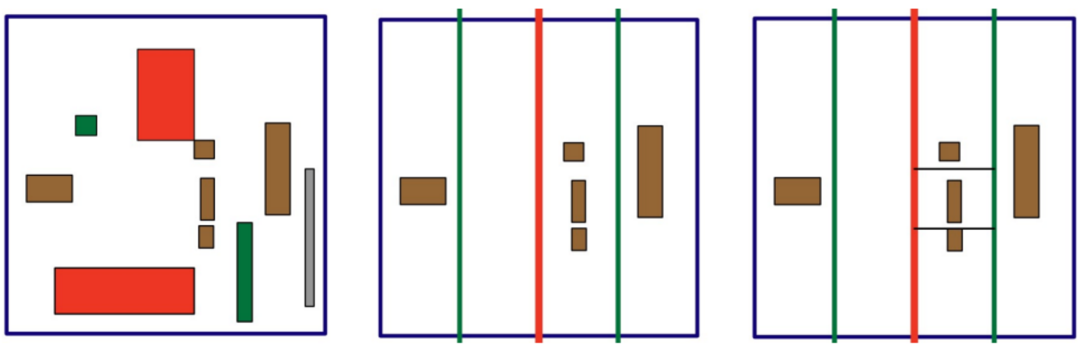


Figure 30: The rectangles in brown colour class are guillotine separable. Similarly the other color classes are also guillotine separable.

By taking the color class of maximum cardinality we get a guillotine separable set with $\geq \frac{n}{1+\log n}$ rectangles (we have $\leq \log n + 1$ colour classes and n rectangles, so by pigeon hole principle one of them must have cardinality $\geq \frac{n}{1+\log n}$). This shows that $\alpha \geq \frac{n}{1+\log n}$. Hence we are done.

Remark By allowing more generalised guillotine cuts (instead of rectangular region allow orthogonal polygons with t sides) Galvez et al. gave a $(2 + \varepsilon)$-approximation for MISR.