Approximation Algorithms for Some Geometric Packing/Covering/Routing Problems

Joe Mitchell

Stony Brook University

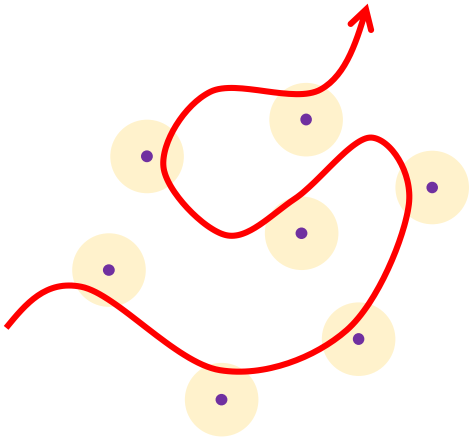
Support from NSF, DARPA, Sandia National Labs, US-Israel Binational Science Foundation

Some NP-Hard Optimization Problems in Geometry

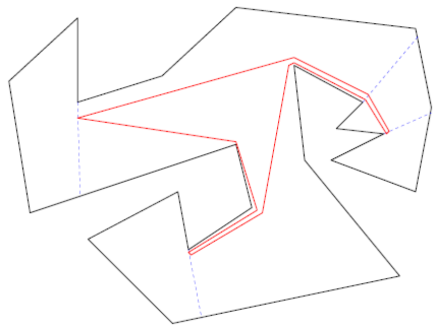
- TSP. vehicle routing variants
- Watchman routes
- Min-Weight Convex Partition
- MACS: Maximum Area Connected Subset
- MIS: Maximum Independent Set
- …

Goal: Exploit geometric structure to get efficient, provable approximation algorithms

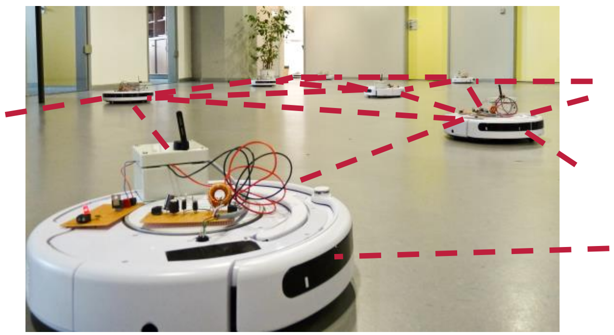
Introduction



- Sampling of optimization problems:
  - Optimal routes/networks to visit regions
  - Optimization of routes for vision/coverage



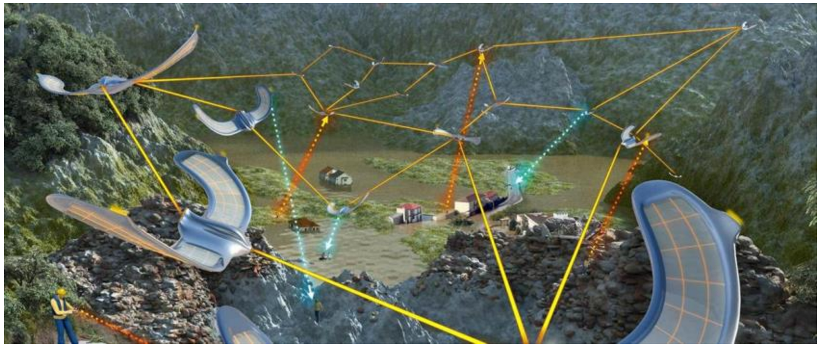
- Aspects of current interest:
  - Uncertainty, robustness of solutions
  - Handling time constraints
- Motivating applications:
  - Robotics
  - Sensor networks
  - Vehicle routing, logistics



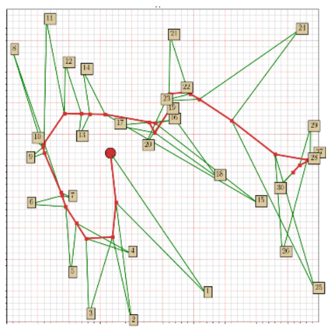
Cooperative Heterogeneous Vehicle Mission Planning

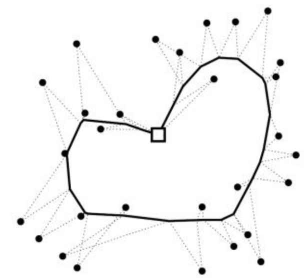
Motivating applications: search and rescue; casualty/disaster response; surveillance; mosaic battlefield

- Vehicles: various classes (ground, air, sea), speeds, capacities, capabilities
- Targets: points, regions; mission task times; precedence constraints
- Constraints: domains of operation; tethers (distance); rendezvous requirements, formations
- Tactical vs strategic; online vs offline

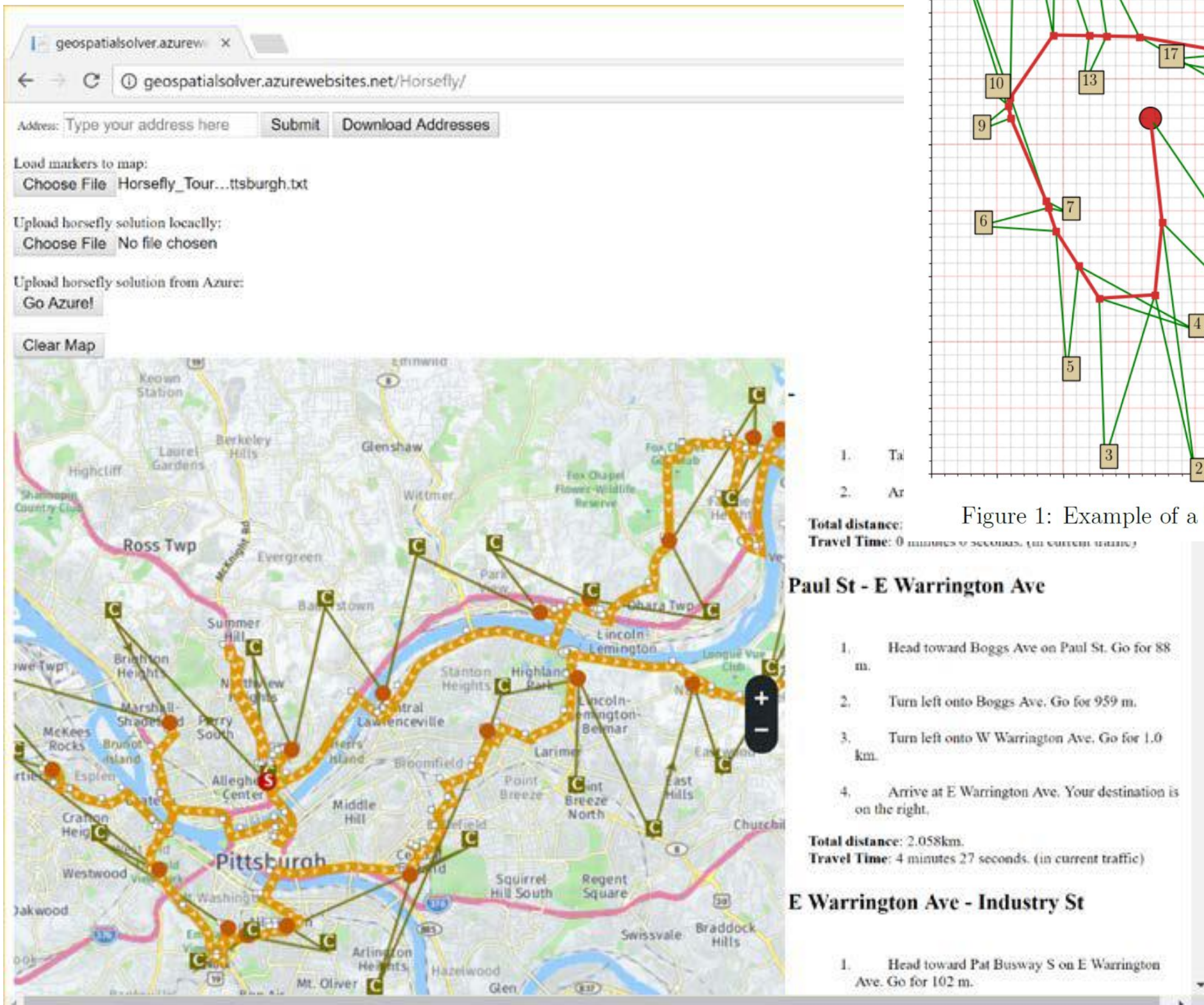


“Horsefly”: Drone-Assisted Mission Planning (DAMP)






- Drone (UAV) picks up a payload from a truck, which continues on its route, and after a successful delivery, the drone returns to the truck to pick up the next payload
- Truck is an “aircraft carrier”; does not stop at targets
- Computing the most efficient routes is challenging because we have to coordinate both vehicles simultaneously
- For a fixed target sequence, the problem is a ***Second Order Cone Program (SOCP)***
- For a fixed truck route, the problem is a challenging geometric scheduling problem
- We studied also generalizations to mutli-trucks, multi-drones



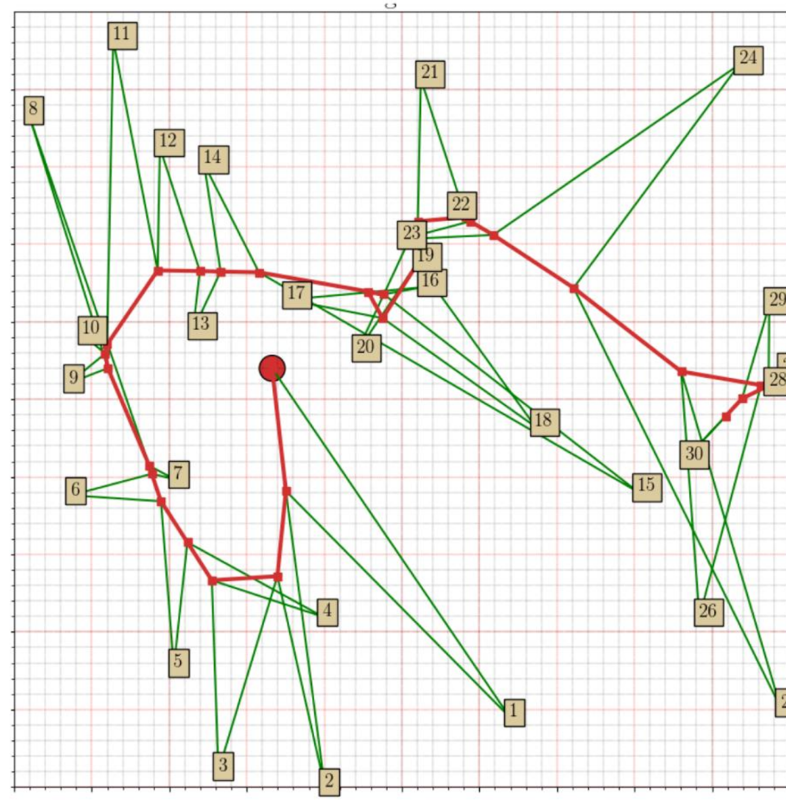


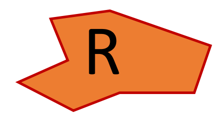
Figure 1: Example of a Horsefly Tour on 30 points. $\varphi = 6.0$

Missions for Agents, UAVs

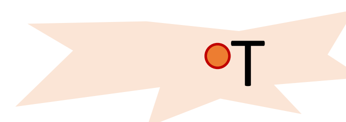
Types of mission tasks:

- Visit target site (point) p
- Visit (any point) of target region R
  - Possible constraint: Mission time (minimum) within R

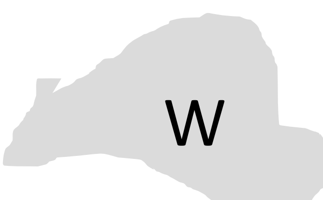
[Jia, Mitchell, 2019: TSPN with time lower bounds. PTAS, dual approximation algorithms]

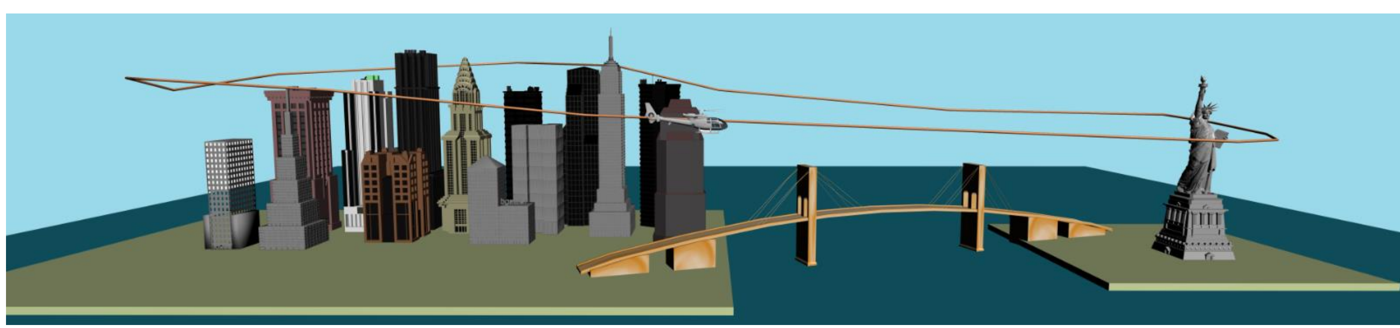


- View a target (point/region) T: visit any point that is visible to T "watchman route problem"



- Sweep a target region (recon, search), W





Approximation Algorithms

For a minimization problem, seek an upper bound on the ratio

α=(worst-case bound given by ALG)/ OPT

α-approximation

Possibly, $\alpha=\alpha(n)$ depends on n, the size of the input

PTAS: For any fixed eps>0, there is a polytime algorithm achieving (1+eps)-approximation

(EPTAS, FPTAS,QPTAS)

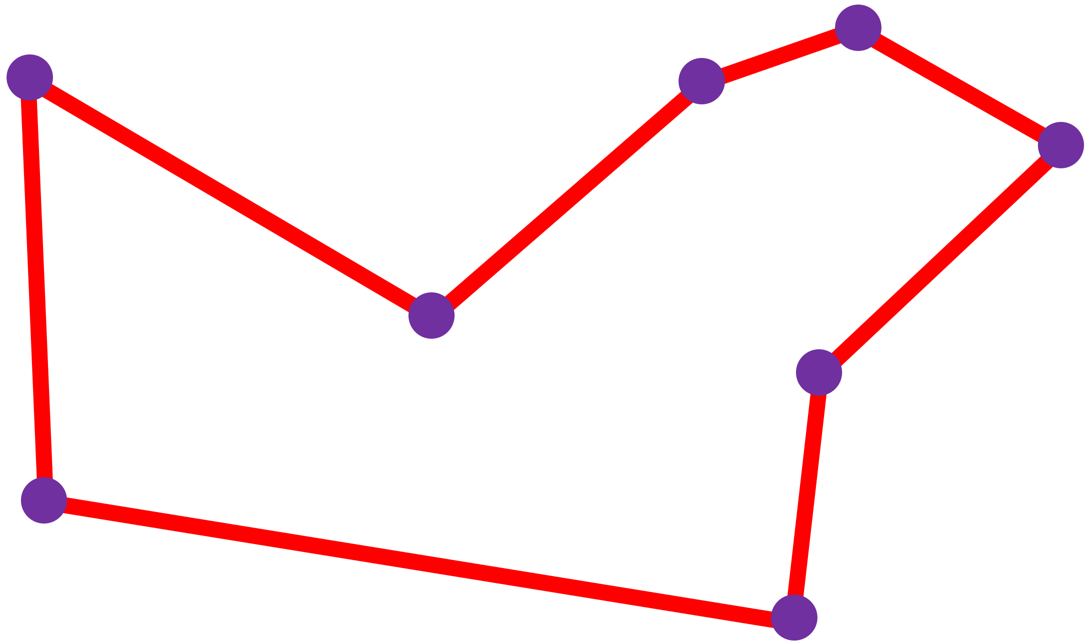
Approximation Techniques

- Solve an easier problem, and use it to solve the hard problem, approximately
- Linear Programming relaxation of an Integer Program; Semi-Definite Programming
- Grid shifting, quadtrees; m-guillotine method
- Approximating subsets; “core sets”
- Local search

Geometric Covering Tours, TSP

Covering Tours

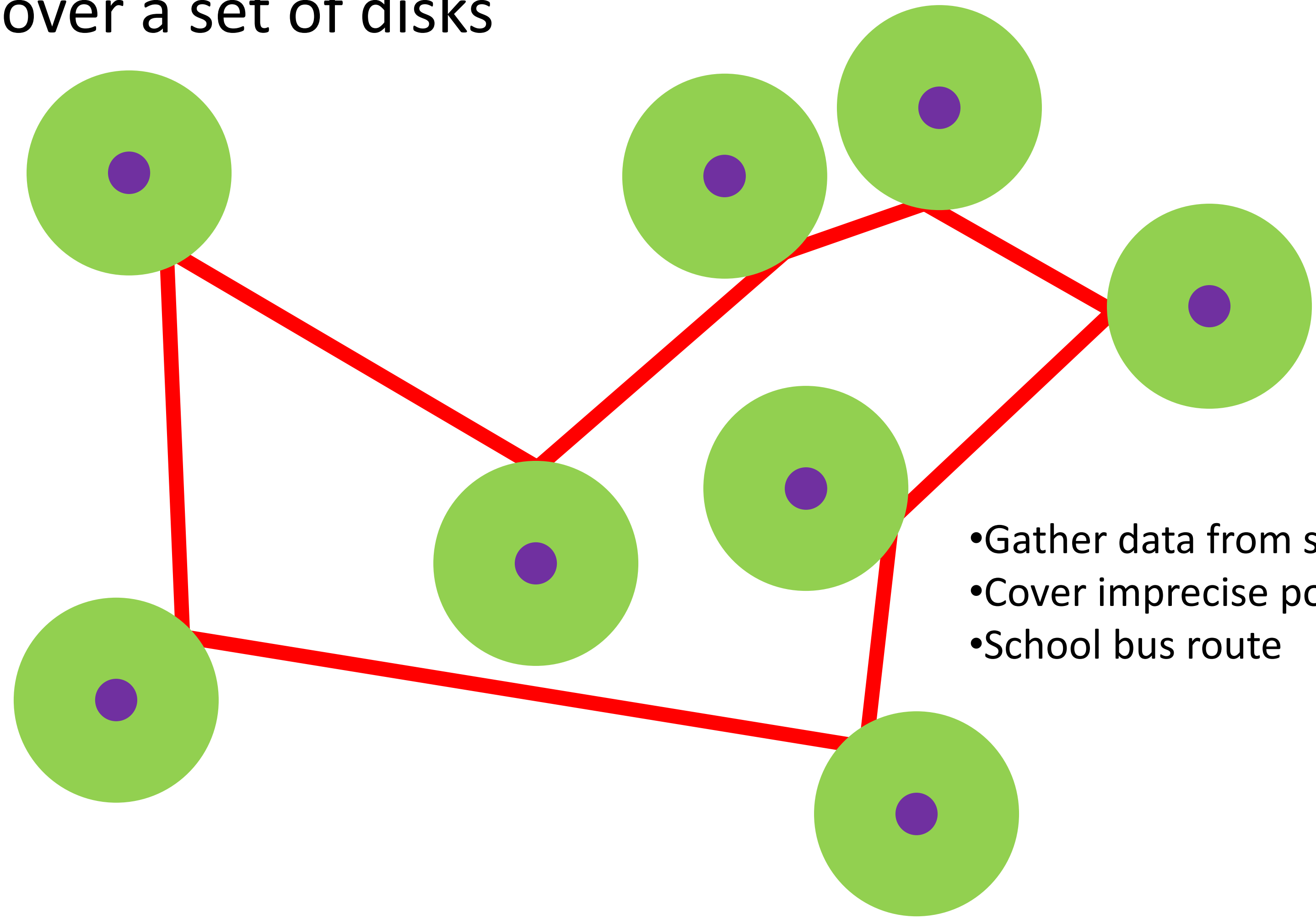
- Cover a point set S



Just geometric TSP

Covering Tours

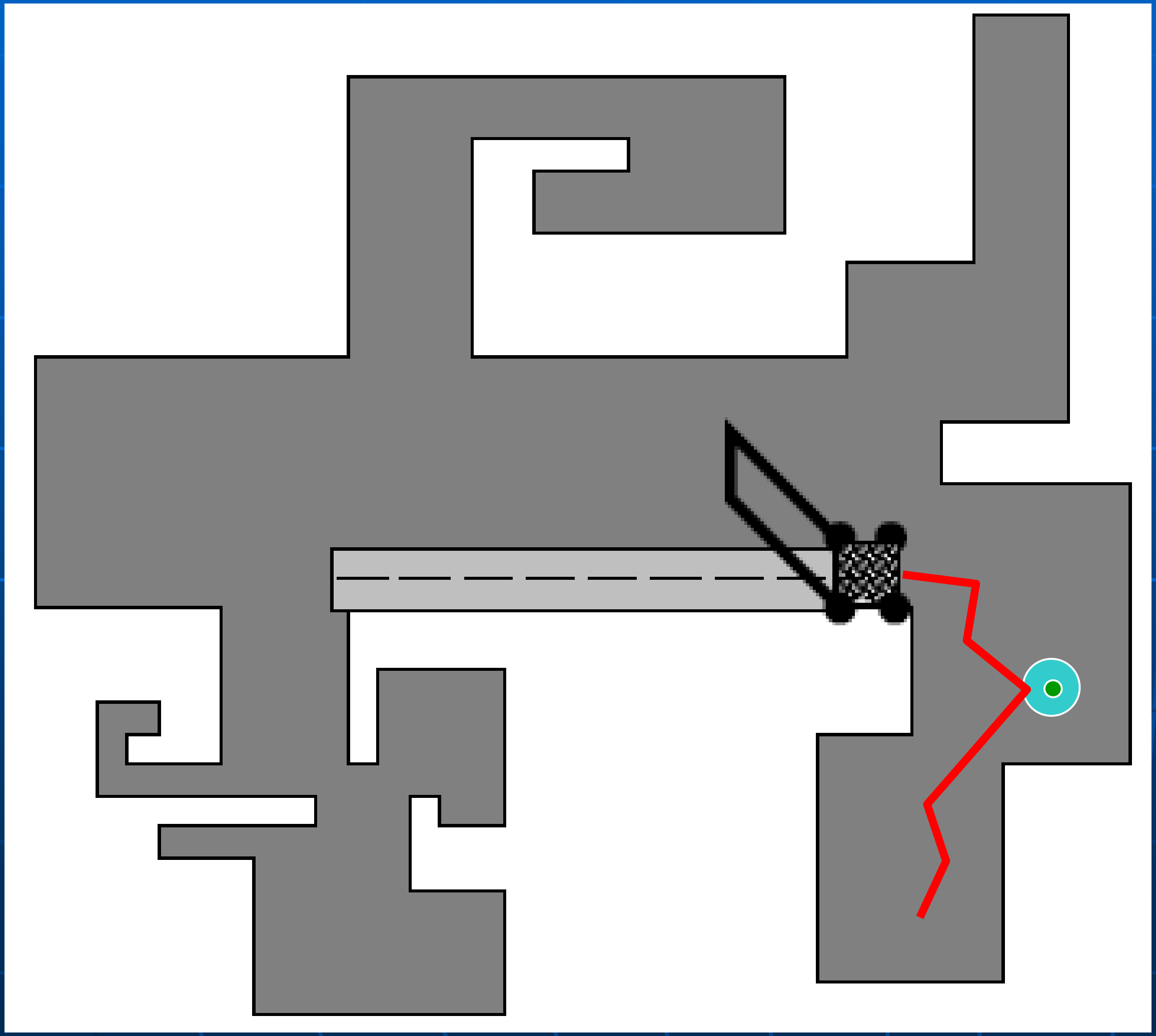
- Cover a set of disks



- Gather data from sensors
- Cover imprecise points
- School bus route

TSP with (circular) neighborhoods

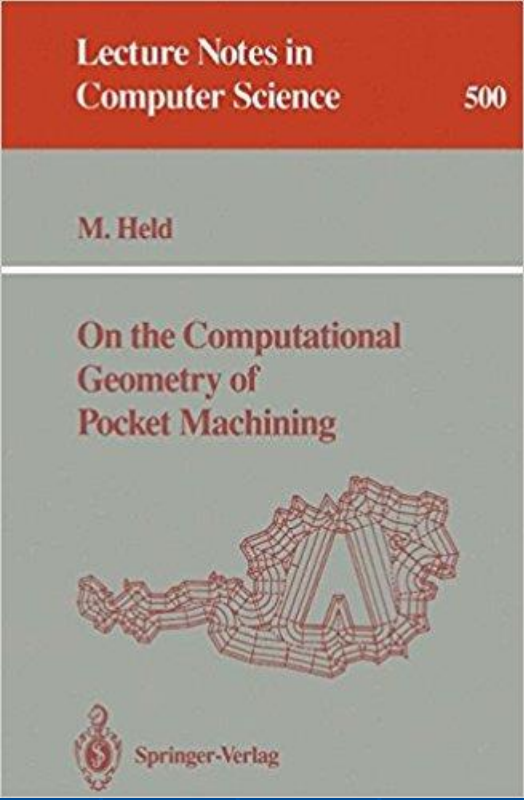
Lawnmower/Milling Problem



[AFM]

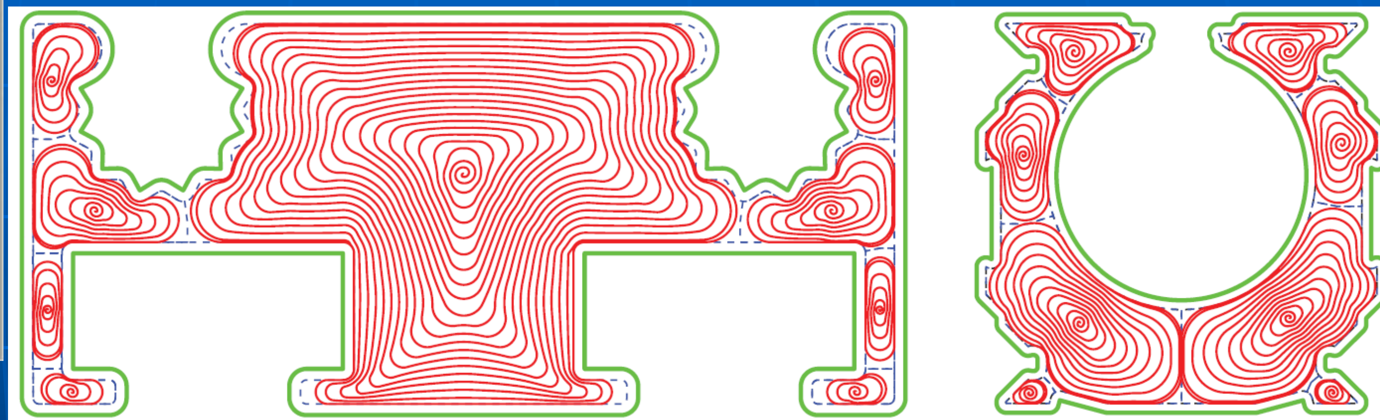
Best method of mowing the lawn?

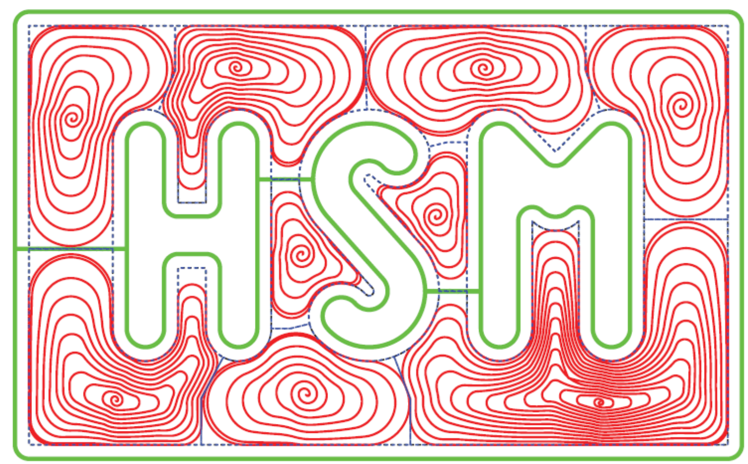
TSPN: Visit the disk centered at each blade of grass



Pocket Machining

[Martin Held]

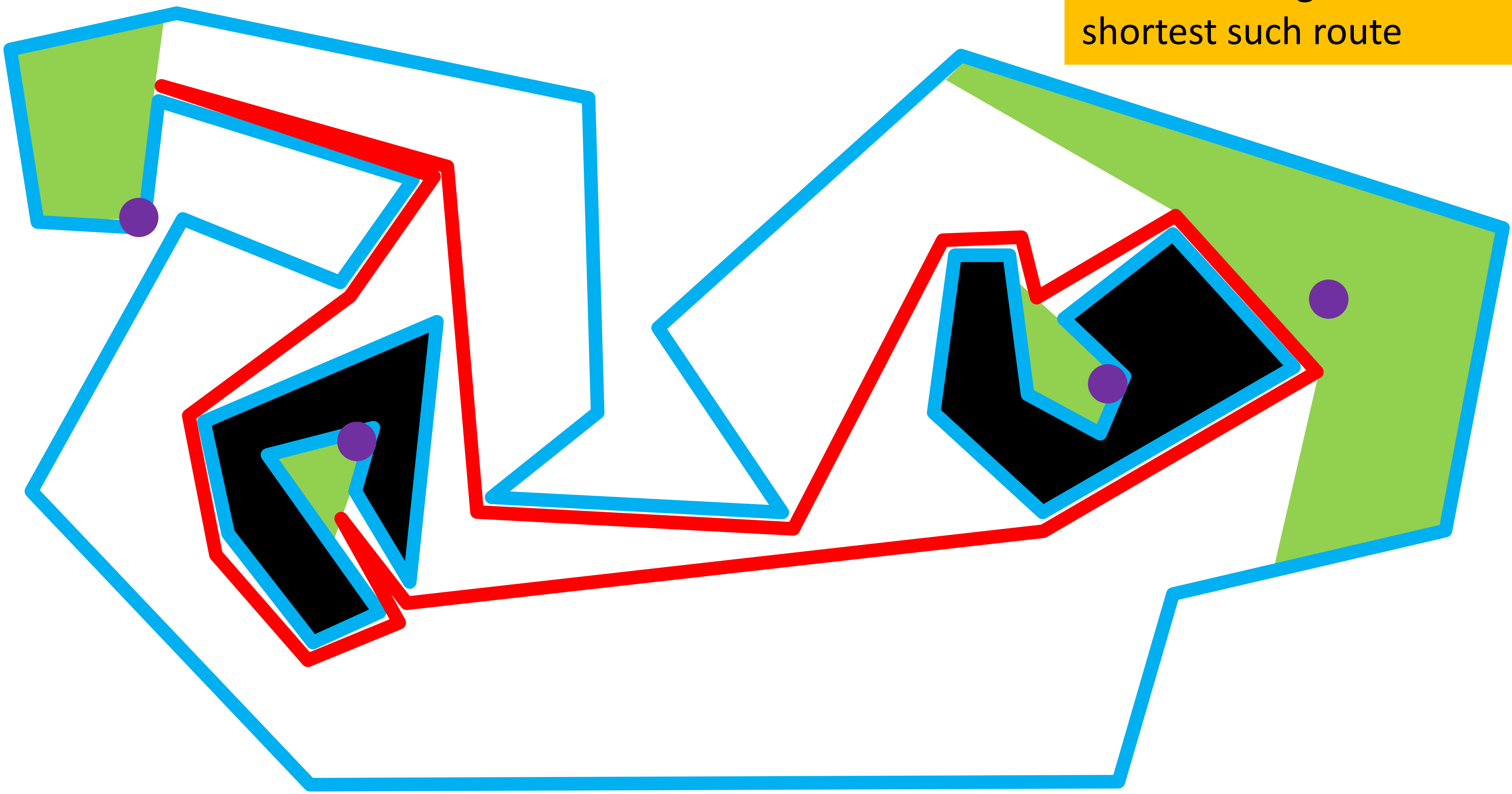




Covering Tours

- Cover a set of visibility polygons

Search for location of an uncertain target: Want shortest such route



Watchman Route Problem

WRP Approximation

- Simple polygons:
  - Sqrt(2)-approx, O(n), for anchored [Tan, DAM 2004]
  - $14(\pi+4)=99.98$-approx, O(n log n), for floating [Carlsson, Jonsson, Nilsson, TR 1997]
  - 2-approx, O(n), for floating [Tan, TCS 2007]
  - 4-approx, $O(n^2)$, for min-link [Alsuwaiyel, Lee, IPL 1995]
- Polygons with holes?

SODA'13: $O(\log^2 n)$, $\Omega(\log n)$

  - O(log n)-approx, rectilinear, rectangle-visibility
- WRP in 3D: No constant-factor, unless P=NP

[Safra, Schwartz 2003]

$\Omega(\log n)$, even for terrains

Variants

- k Agents/Tours
  - Possibly tethered, or otherwise constrained

“tethered TSP”

- Depot(s)
- Offline vs online
- Time windows/constraints

e.g., must see each point for time at least T; min makespan

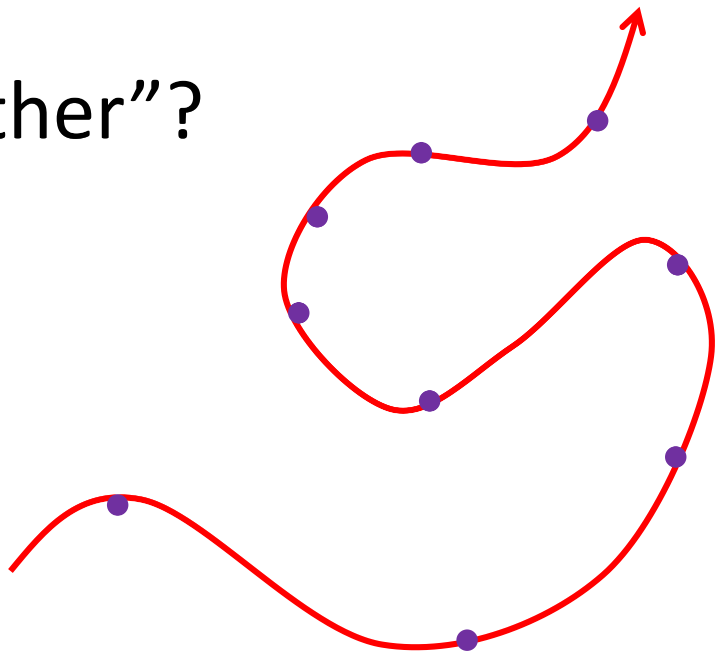
- Sites in motion (kinetic variant)
- Uncertain sites: Stochastic models
- Precedence constraints
- Priorities, “prizes” at sites
- Various objectives, multicriteria optimization

Objective Functions

- Min-length of tour (time to complete search)
  - Euclidean, L_p, weighted lengths
- Optimize edge lengths: min-max (bottleneck), max-min (max-scatter)
- Min turning: # turns, total turn, bounded curvature
- k Tours: min-max, min-sum; min-k for tours of bounded length
- Min-latency
- Stochastic metrics: e.g., max P(tour length < L), or min E(time until a goal is achieved)
- Combined metrics, multicriteria

Application: Mobile Agents to Gather Data from Static Sensors

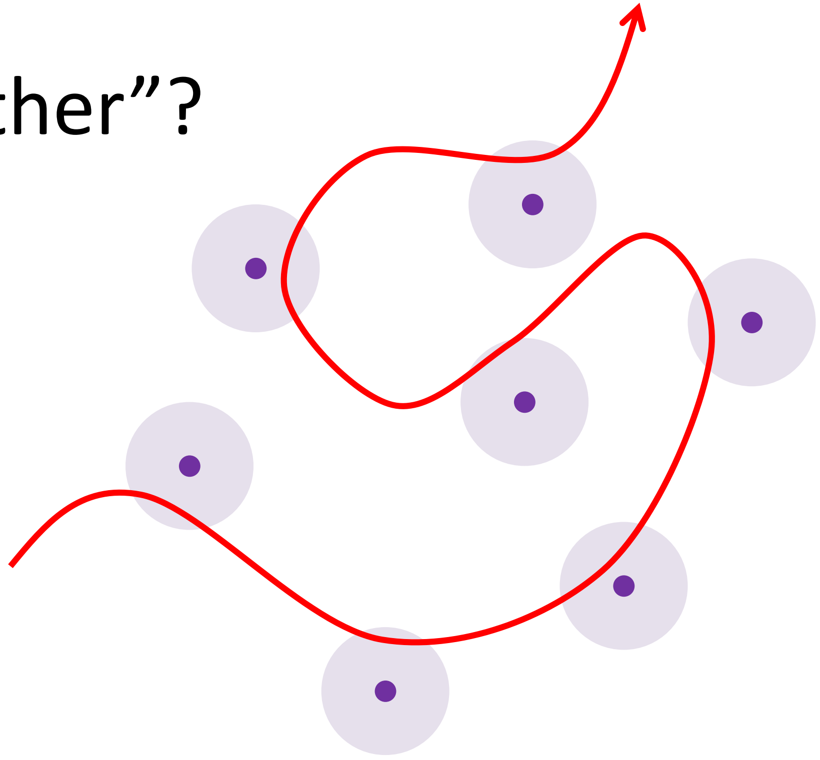
- What does it mean to "gather"?
  - Arrive at a sensor (point)



TSP

Mobile Agents to Gather Data from Static Sensors

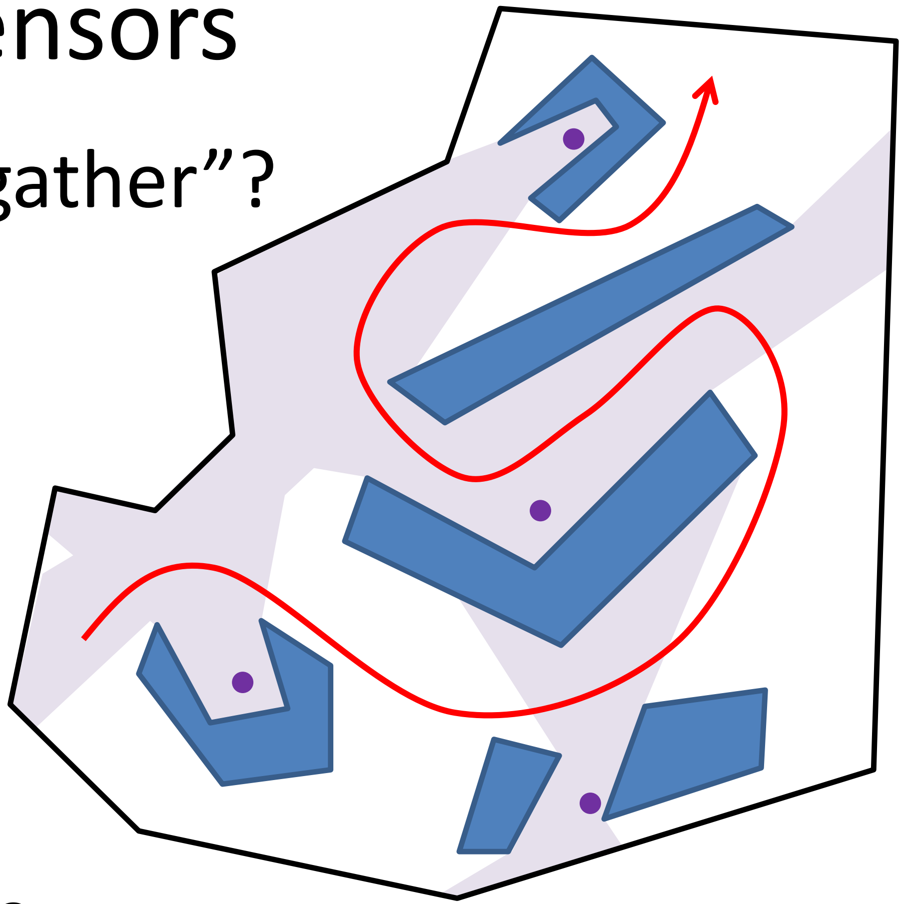
- What does it mean to "gather"?
  - Arrive at a sensor (point)
  - Arrive close to a sensor



TSP with neighborhoods

Mobile Agents to Gather Data from Static Sensors

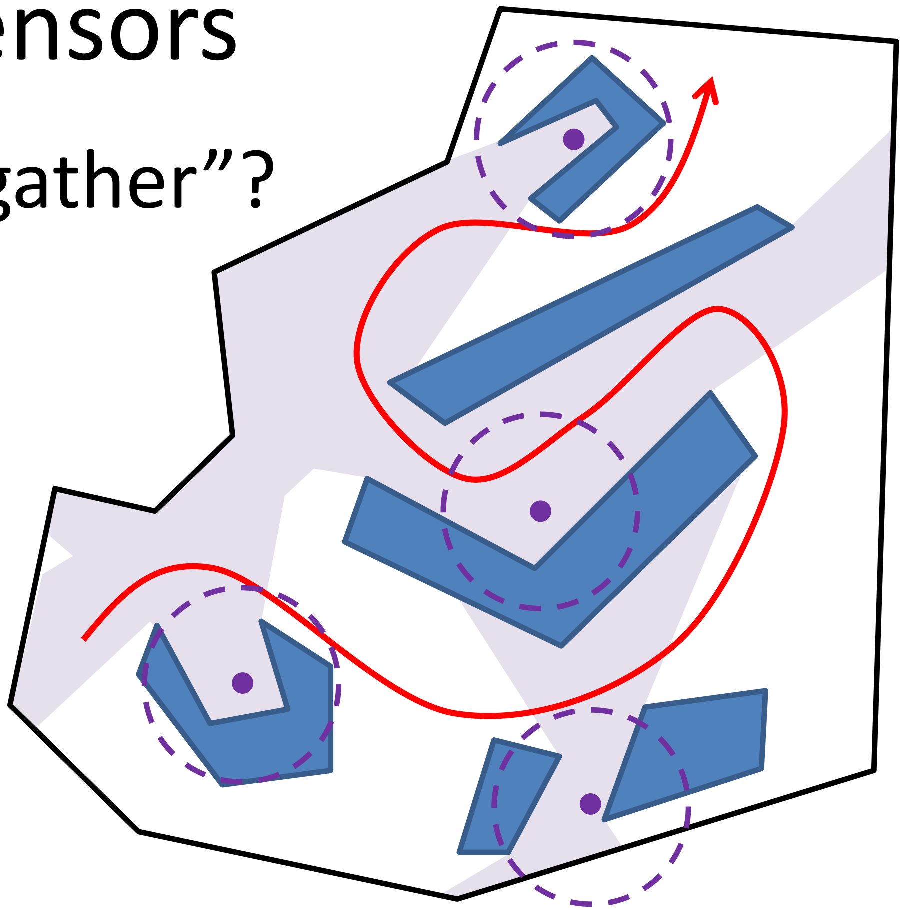
- What does it mean to “gather”?
  - Arrive at a sensor (point)
  - Arrive close to a sensor
  - See a sensor
    - Unlimited sight distance



Watchman route

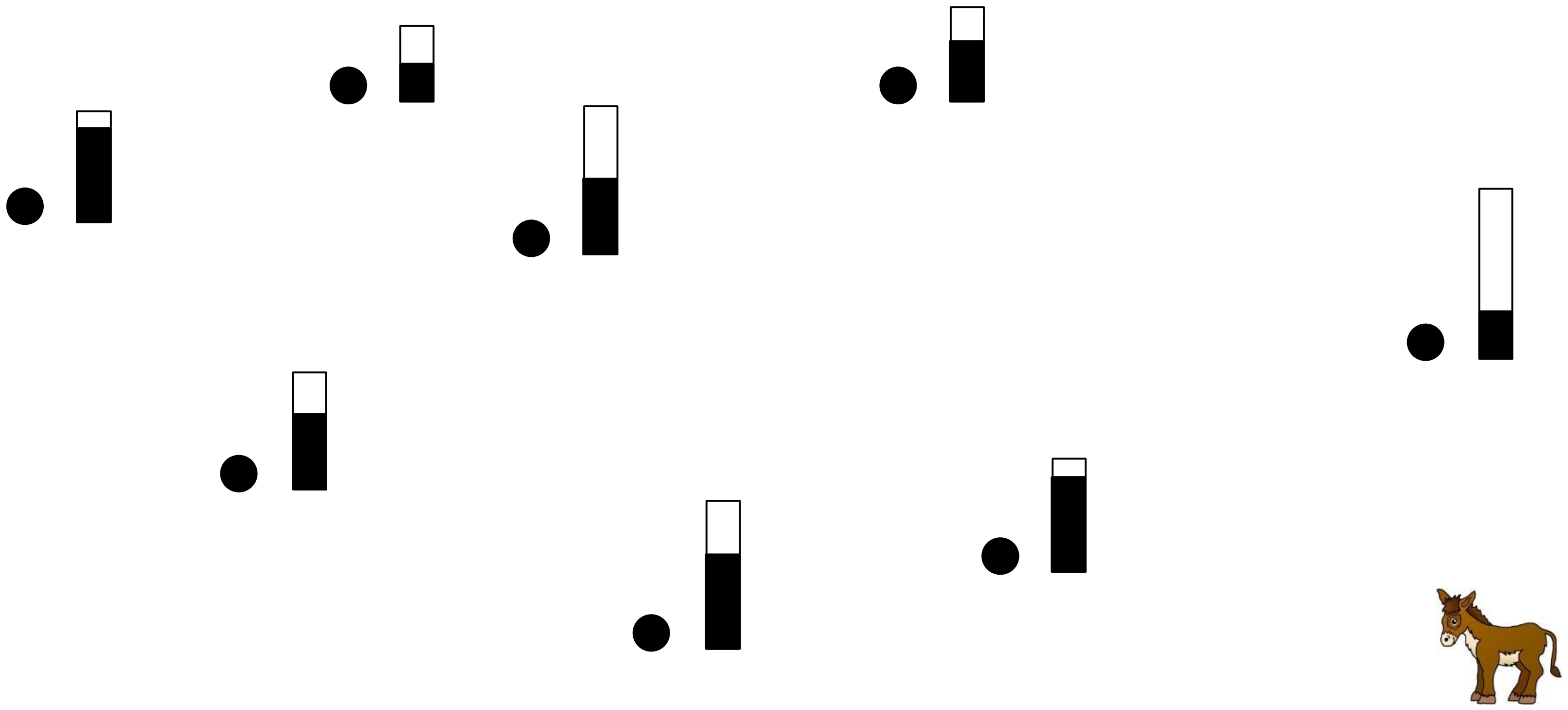
Mobile Agents to Gather Data from Static Sensors

- What does it mean to "gather"?
  - Arrive at a sensor (point)
  - Arrive close to a sensor
  - See a sensor
    - Unlimited sight distance
    - Limited sight distance



Data Gathering Problem

[Citovsky,Gao,M,Zeng, ALGOSENSORS 2015]



Each sensor, at position p_i, has a capacity, c_i
Data arrives at rate r_i
If capacity is reached, additional data is lost

Data Gathering Problem

Given k mules travelling at a constant speed s and n sensors at points $\{p_1, p_2, \ldots, p_n\}$ in a metric space with capacities $\{c_1, c_2, \ldots, c_n\}$ and continuous data accumulation rates $\{r_1, r_2, \ldots, r_n\}$.

Goal: Find a schedule/tours to maximize the data collection rate of all of the mules.

Each sensor, at position p_i, has a capacity, c_i
Data arrives at rate r_i
If capacity is reached, additional data is lost

No Data Loss Problem: Min # of Mules

Given n sensors at points $\{p_1, p_2, \dots, p_n\}$ in a metric space with capacities $\{c_1, c_2, \dots, c_n\}$ and data accumulation rates $\{r_1, r_2, \dots, r_n\}$.

Goal: Find the min # mules and their schedules such that no data is lost.

Our Results

With Sensors	Single Mule	K-mule	No Data Loss
on a Line	exact	$\frac{1}{3}$	exact
on a Tree	exact pseudo-polynomial	$\frac{1}{3}\left(1-1/e^{\frac{1}{2+\varepsilon}}\right)$	12
General Metric Space	$\frac{1}{6}-\varepsilon$		
Euclidean Space	$\frac{1}{3}-\varepsilon$	$\frac{1}{3}\left(1-1/e^{1-\varepsilon}\right)$	
Different Capacities	$O(\frac{1}{m})$		$O(m)$

$m = \log(\frac{c_{max}}{c_{min}})$, c_{max} = largest capacity, c_{min} = smallest capacity

(OPT is periodic if we assume integral capacities, rates, distances)

Related: Variations of TSP

Orienteering Problem: Tour length quota, L

Prize-Collecting TSP: Prize quota (possible costs for skipping sites)

Profitable Tour Problem: max (Prize - Tour length)

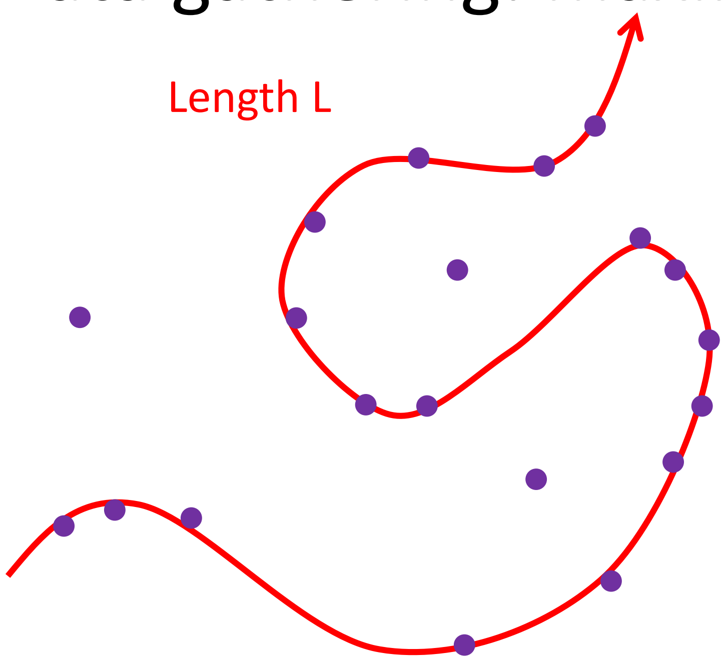
TSP with Time Windows

Orienteering

Given n sites $S=\{p_1, p_2,\ldots,p_n\}$; length bound L.

Goal: Find a path/tour of length ≤L to Max # of sites visited

Data gathering: maximize the *collected data rate*



O(1)-approx [AMN, SoCG'98]

PTAS, for rooted case, based on improved analysis of m-guillotine method for k-TSP [CH, SoCG'06]

Improved PTAS [GKR 2020]: $n^{O(1/\delta)}(\log n)^{(d/\delta)^{O(d)}}$

Q: O(n log n)?

TSP/Orienteering with Time Windows

Given n sites $S=\{p_1, p_2,\ldots,p_n\}$, each with a time window, (r_i,d_i); length bound L.

r_i=release time at p_i
d_i=deadline at p_i

Possible service time, t_i, at p_i;
assume $t_i=0$.

Laxity = $\min_i (d_i-r_i)$

Goal: Find a path/tour to Max # of sites visited during their respective time windows

Time-Window TSP

[WAFR 2016, Jie Gao, Su Jia, M]

- Time Window Prize Collecting (TWPC):

Unit speed robot; must visit each site i during given time window, (r_i, d_i).

(often called "TWTSP")

Goal: max # sites visited (or total "prize")

- Time Window Travelling Salesman (TWTSP):

Robot with speed s; must visit each site i during given time window, (r_i, d_i).

(may not be feasible for small s)

Goal: min *distance* robot travels to visit all sites (in TW)

Time-Window TSP

[WAFR 2016, Jie Gao, Su Jia, M]

Various results, including:

Theorem 2. *Given an instance for 1D TWPC problem with bounded velocity s, let L_{max} be the maximum length of the input segments, and assume the shortest time window has length ≥ 1. Then for any $\epsilon > 0$, in $O((nL_{max})^{O(\frac{\log L_{max}}{\log(1+\varepsilon)})})$ time we can find a path P, such that*

1. *the number of segments that P visits is at least OPT,*
2. *each segment σ_i is visited in $[r_i - \varepsilon L_i, d_i + \varepsilon L_i]$, where $L_i = d_i - r_i$.*

Similar result holds for 1D TWTSP with finite speed.

Dual-approximation algorithms for 2D and metric spaces, yielding approx opt solution, using relaxed speed constraint

J Optim Theory Appl
DOI 10.1007/s10957-015-0767-z

The Orienteering Problem with Time Windows Applied to Robotic Melon Harvesting

Moshe Mann[1] · Boaz Zion[2] ·
Dror Rubinstein[1] · Rafi Linker[1] ·
Itzhak Shmulevich[1]

Received: 19 March 2014 / Accepted: 1 June 2015

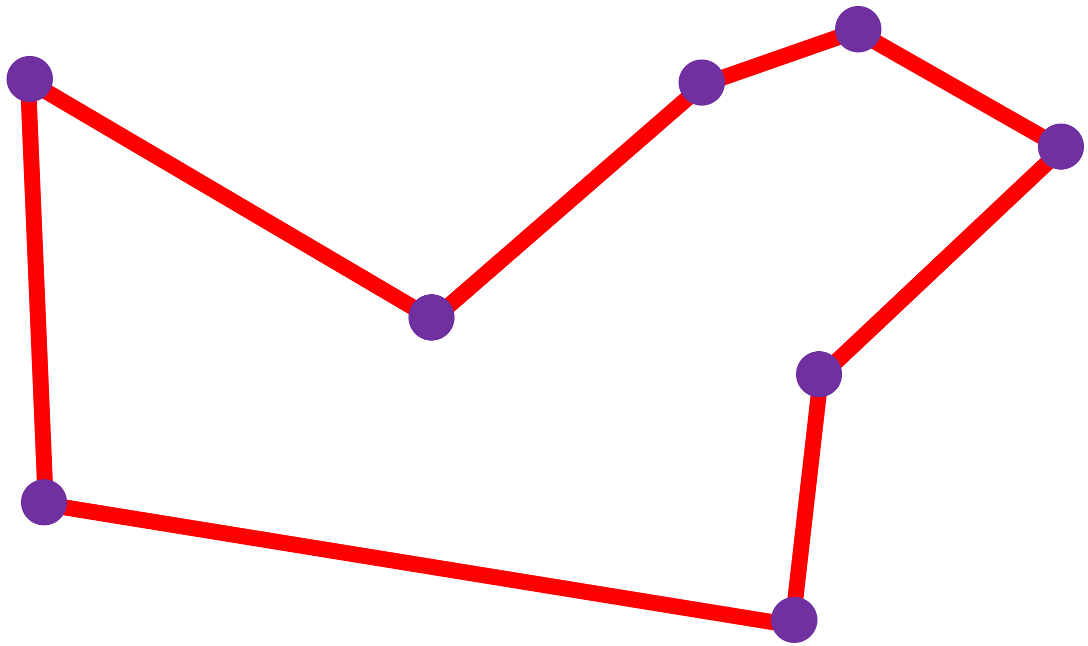
Abstract The goal of a melon harvesting robot is to maximize the number of melons it harvests given a progressive speed. Selecting the sequence of melons that yields this maximum is an example of the orienteering problem with time windows. We present a dynamic programming-based algorithm that yields a strictly optimal solution to this problem. In contrast to similar methods, this algorithm utilizes the unique properties of the robotic harvesting task, such as uniform gain per vertex and time windows, to expand domination criteria and quicken the optimal path selection process. We prove that the complexity of this algorithm is linearithmic in the number of melons and can be implemented online if there is a bound on the density. The results of this algorithm are demonstrated to be significantly better than the standard heuristic solution for a wide range of harvesting robot scenarios.

Keywords Harvesting robot · Orienteering · Time windows · Dynamic programming · Combinatorial optimization

m-Guillotine Method Revisited

Traveling Salesperson Problem: TSP

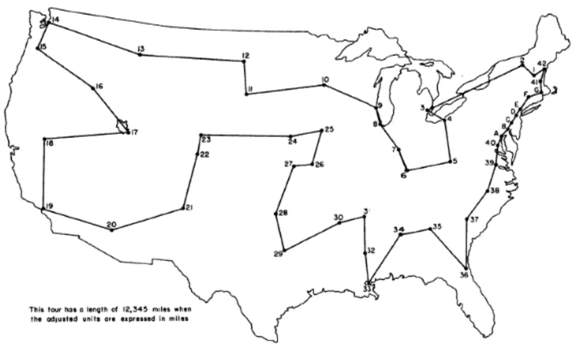
- In Euclidean plane: Find a cycle (polygonal) to visit n points, S, that has the shortest Euclidean length



- Necessarily: it will be a simple polygon with vertex set S (Why? Triangle inequality!)

TSP

- Extremely well studied combinatorial optimization problem

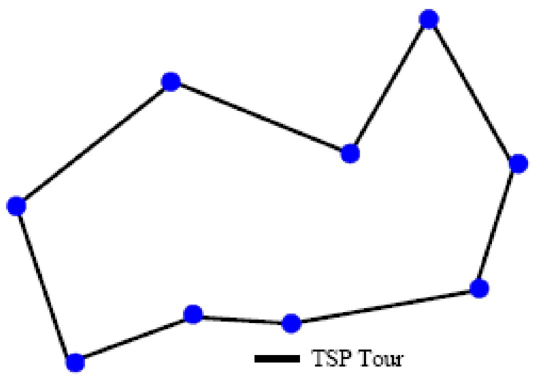


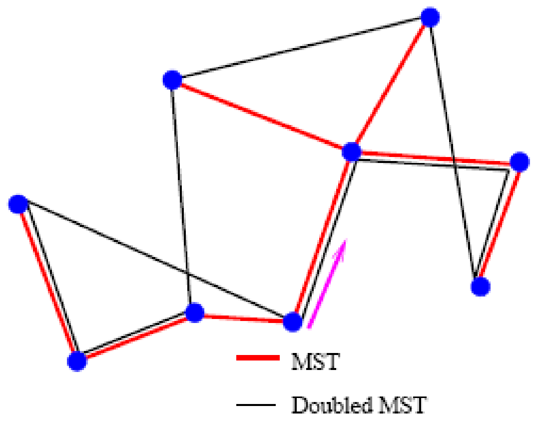
49 cities, 1954

- Many methods to solve to optimality (in worst-case exponential time) or near optimality
- NP-hard, even in Euclidean 2D

Approximating TSP

- Simple 2-approx: double the MST and shortcut (holds in metric spaces)





- Christofides: 1.5-approx

  (use MST $\cup$ min-weight matching on odd-degree nodes of MST)

Recent News: Breaking Below 1.5

A (Slightly) Improved Approximation Algorithm for Metric TSP

Anna R. Karlin[*], Nathan Klein[†], and Shayan Oveis Gharan[‡]

University of Washington

July 6, 2020

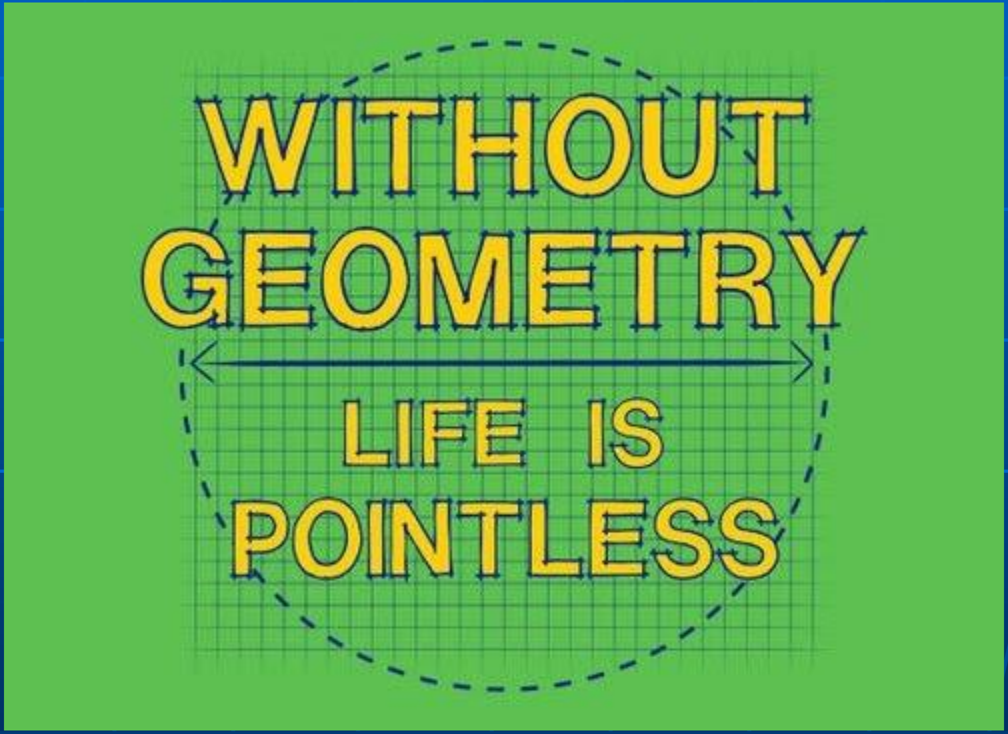
1.499999999999999999999999999999999999-approximation!

Abstract

For some $\epsilon > 10^{-36}$ we give a $3/2 - \epsilon$ approximation algorithm for metric TSP.

https://arxiv.org/pdf/2007.01409.pdf

Can Geometry Help?



PTAS for Geometric TSP

- PTAS in 2D, any fixed dim [Ar'96,M'96,Ar'97,M'97]
- O(n log n): spanners/banyons [Rao-Smith'98]
- NP-hard to get $(1+\varepsilon)$-approx in $R^{O(\log n)}$, for some $\varepsilon>0$ [Tr'97]

Main Idea of PTAS's:

Transform OPT into a near-opt network of special recursive structure that allows efficent optimization by dynamic programming

Recipe for PTAS

OPT → Network with special **recursive** structure

Structure Theorem

increasing length by $\leq (1+\varepsilon)$ factor

Use **dynamic programming** to compute shortest network with the required structure (connectivity, Eulerian subgraph, etc)

Optimal network with special structure → Solution to original problem (tour, tree, etc)

What should the special recursive structure be?

Recipe for PTAS

The Role of Spanners: [Rao-Smith]

OPT → Spanner → Spanner with special **recursive** structure

Structure Theorem

increasing length by $\leq (1+\varepsilon)$ factor

Use **dynamic programming** to compute shortest subgraph of special spanner with the required structure (connectivity, Eulerian subgraph, etc)

Optimal subgraph with special structure → Solution to original problem (tour, tree, etc)

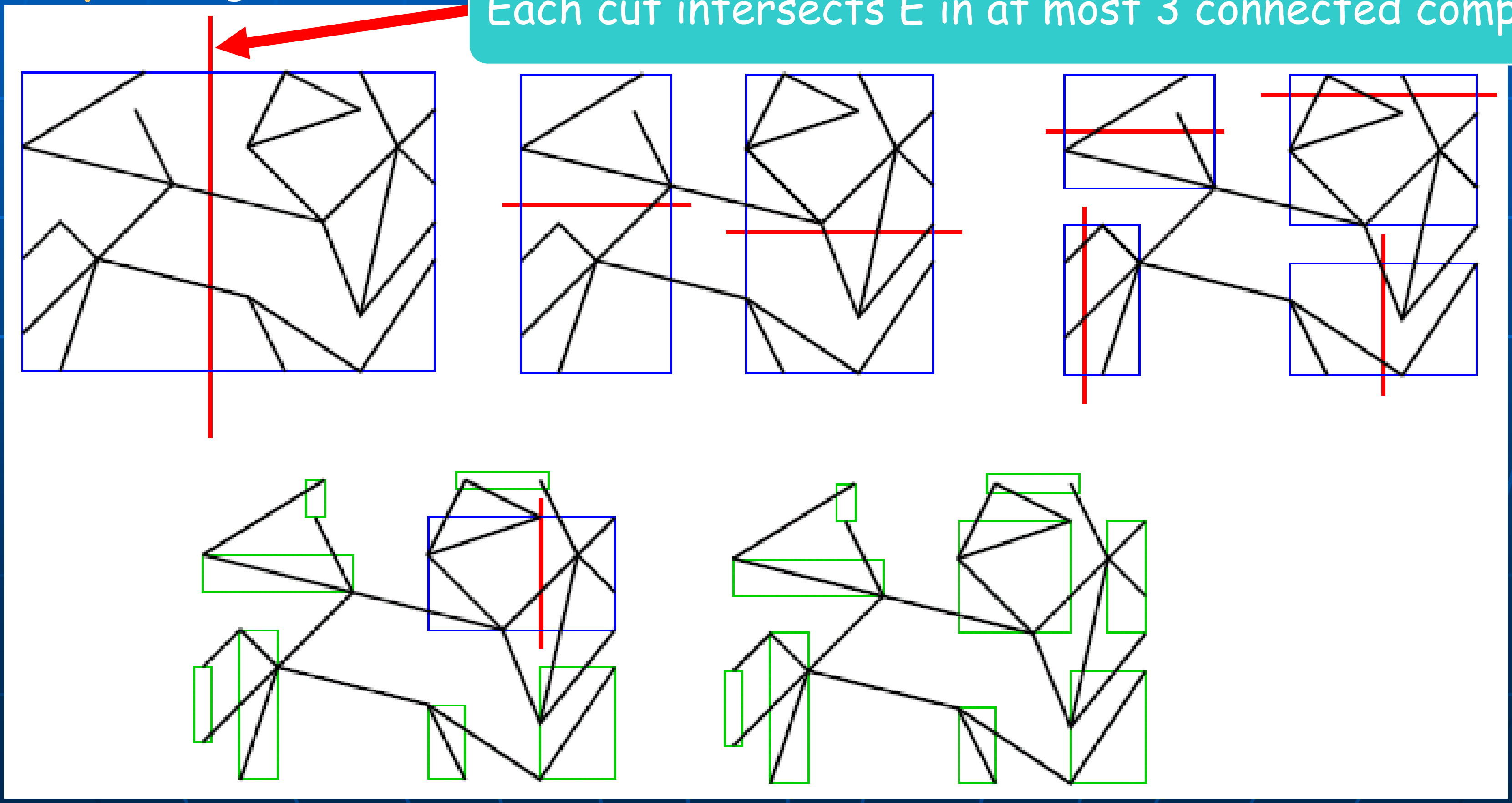
What should the special recursive structure be?

One Possibility:
m-Guillotine Structure

Network edge set E is m-guillotine if it can be recursively partitioned by horiz/vertical cuts, each having small (O(m)) complexity wrt E

Example: 3-guillotine

Each cut intersects E in at most 3 connected components

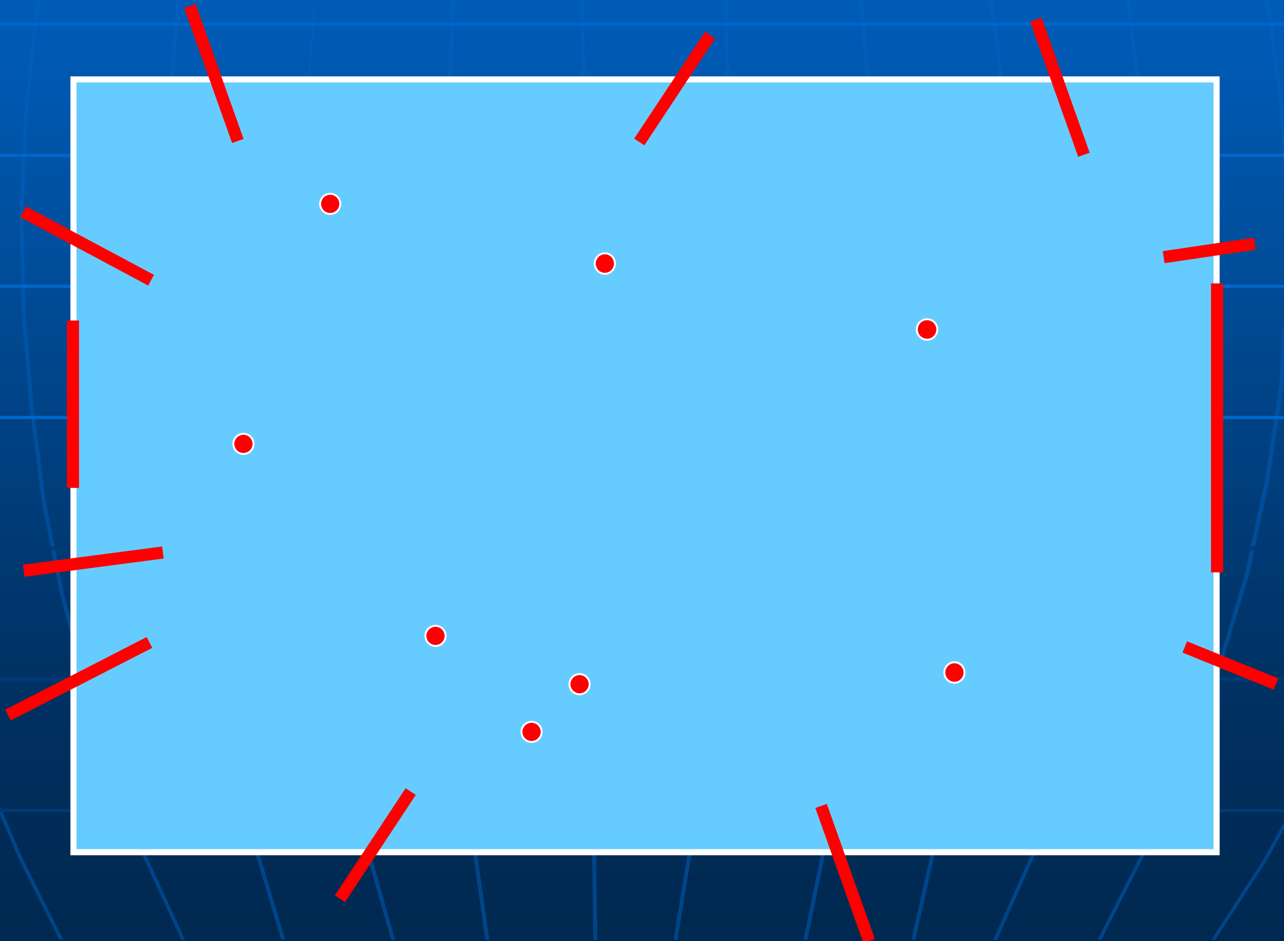


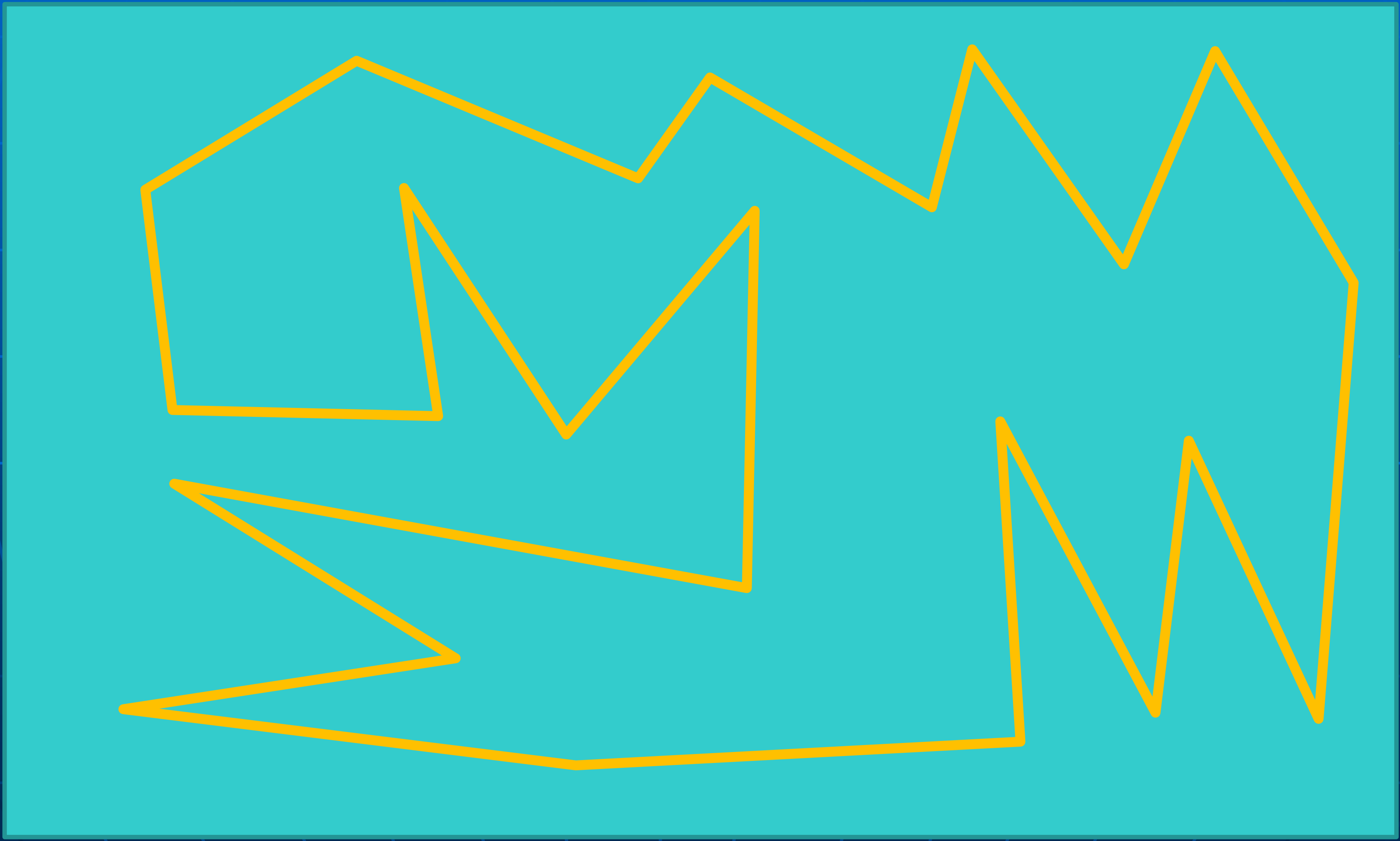
Why m-Guillotine? Desired Recursive Structure

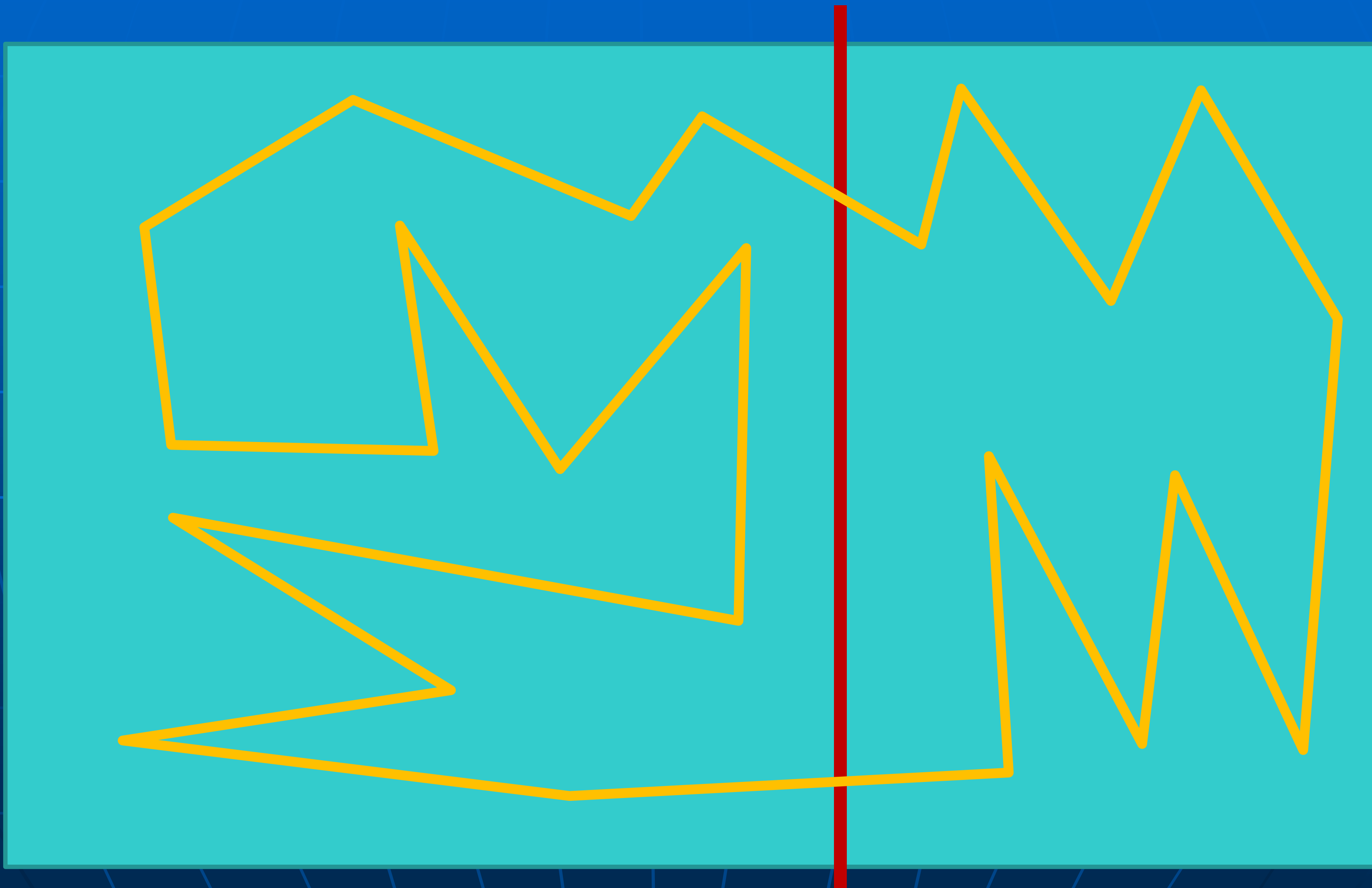
It is "just" what is needed for dynamic programming to work!

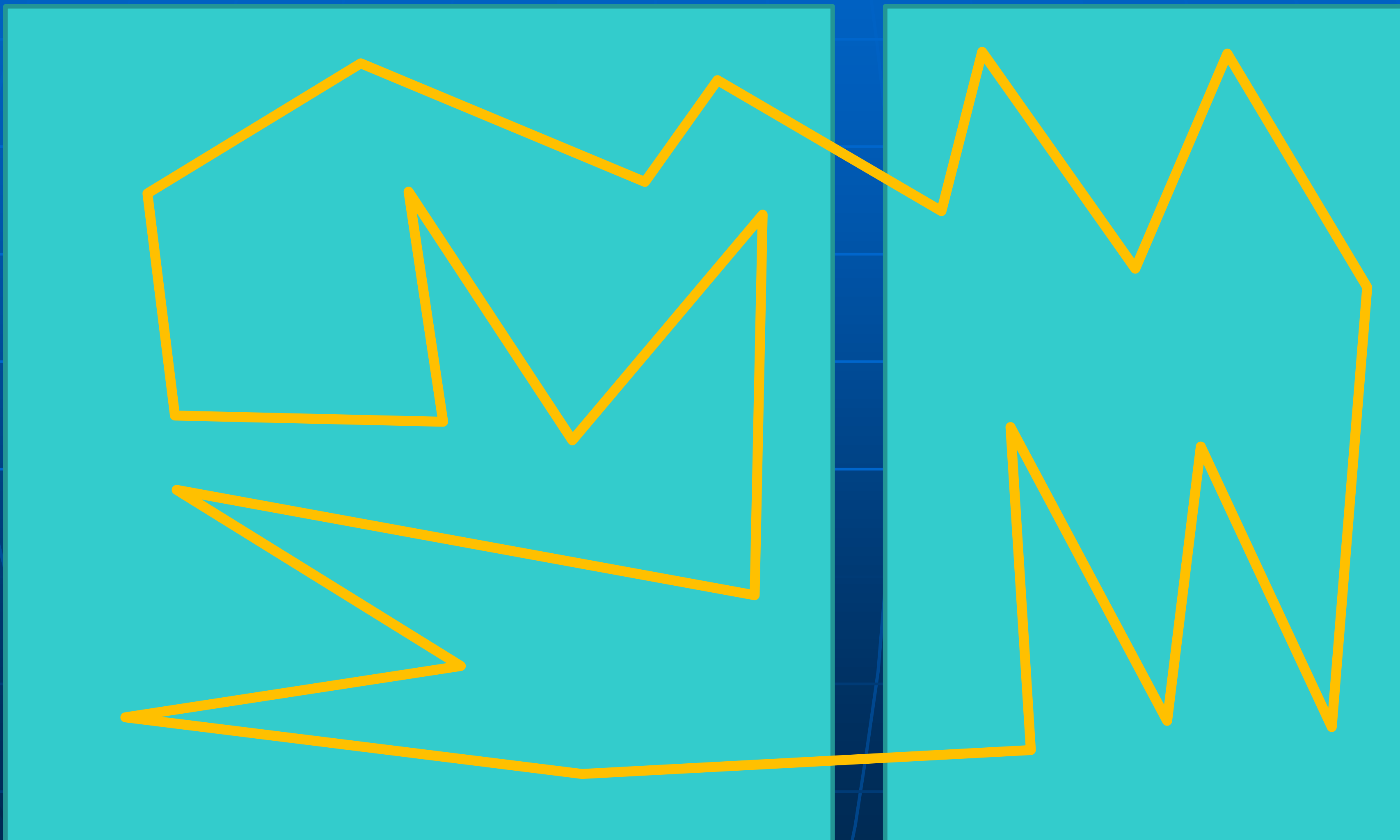
Rectangular subproblem in dynamic program (recursion)

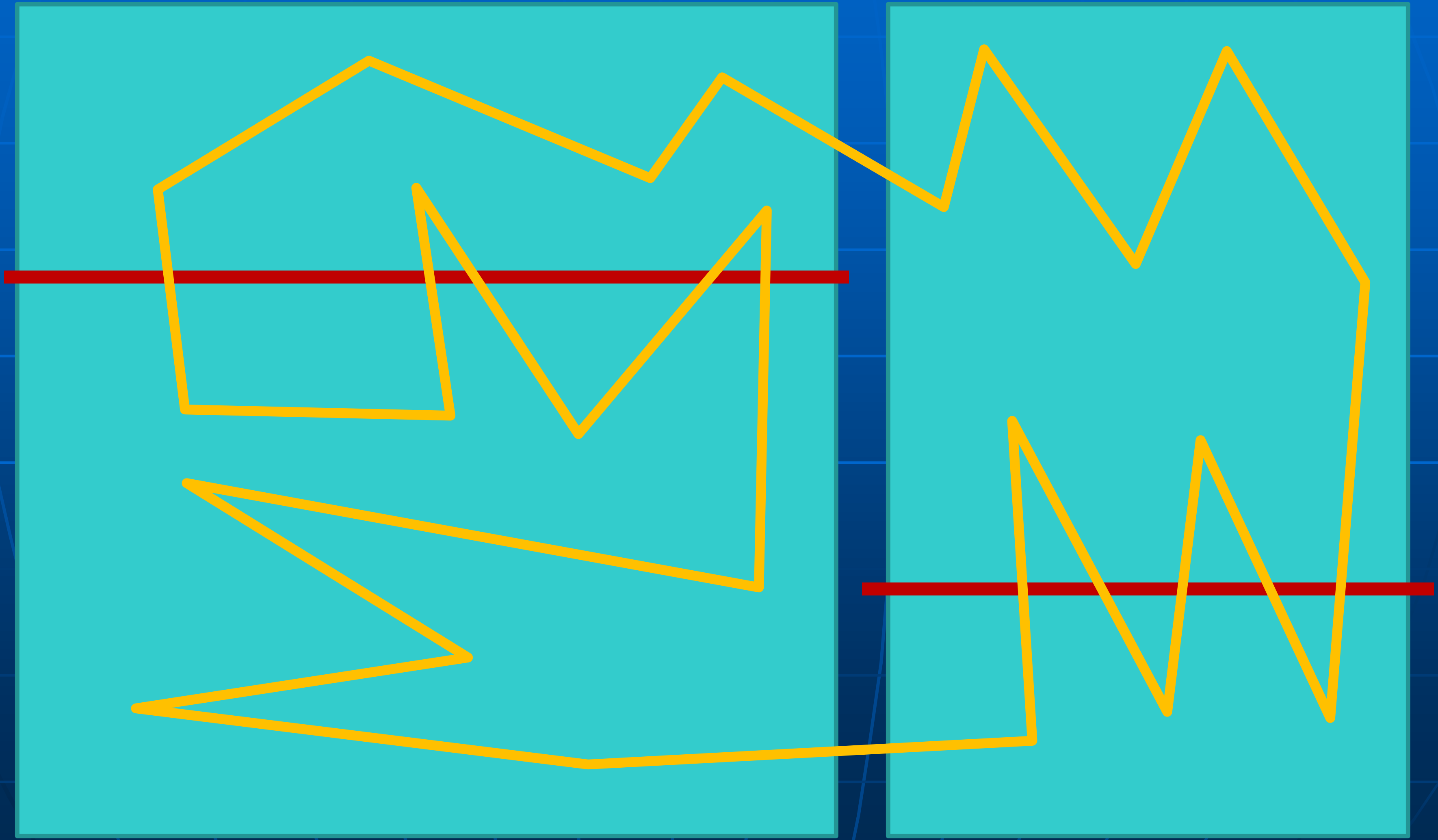
Constant (O(m)) information flow across boundary

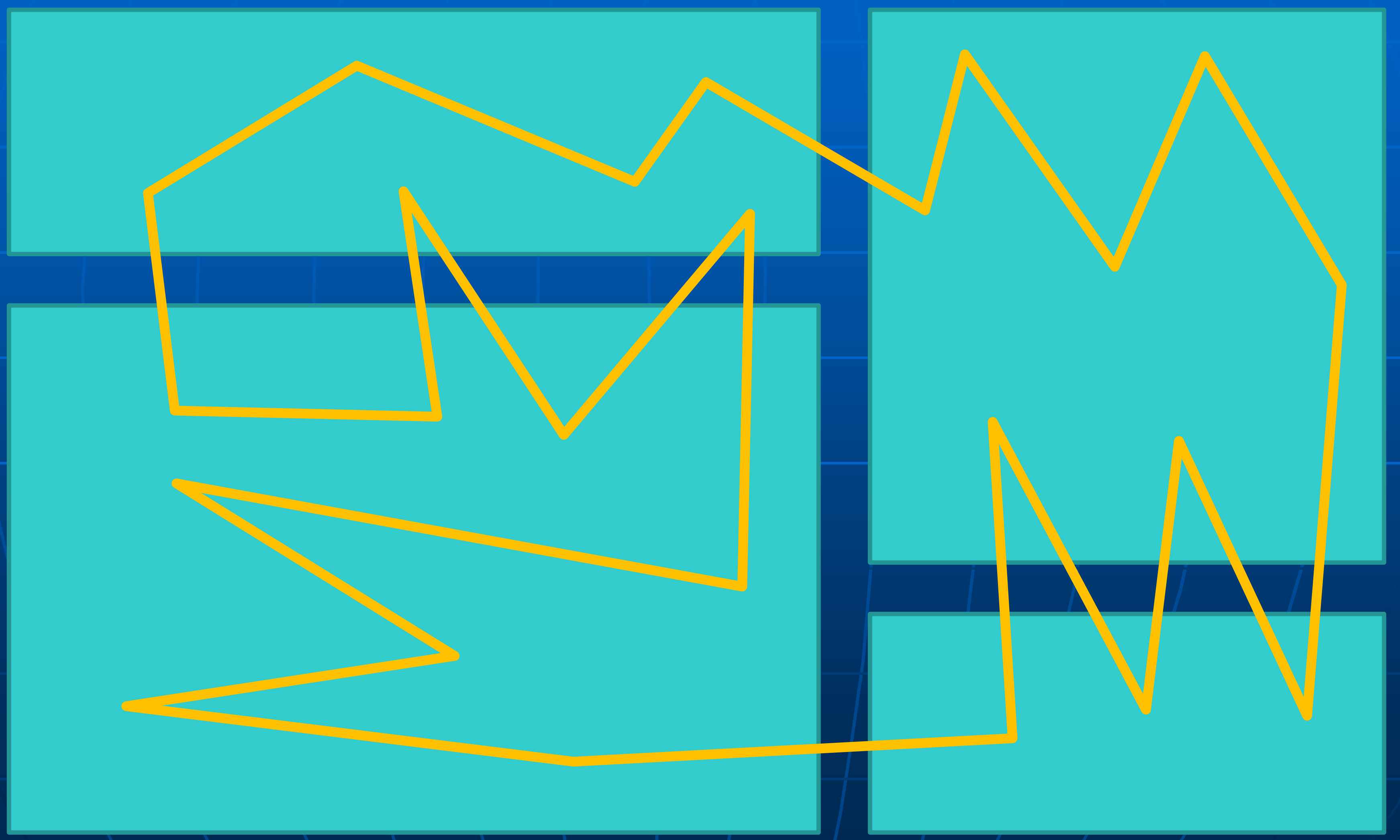


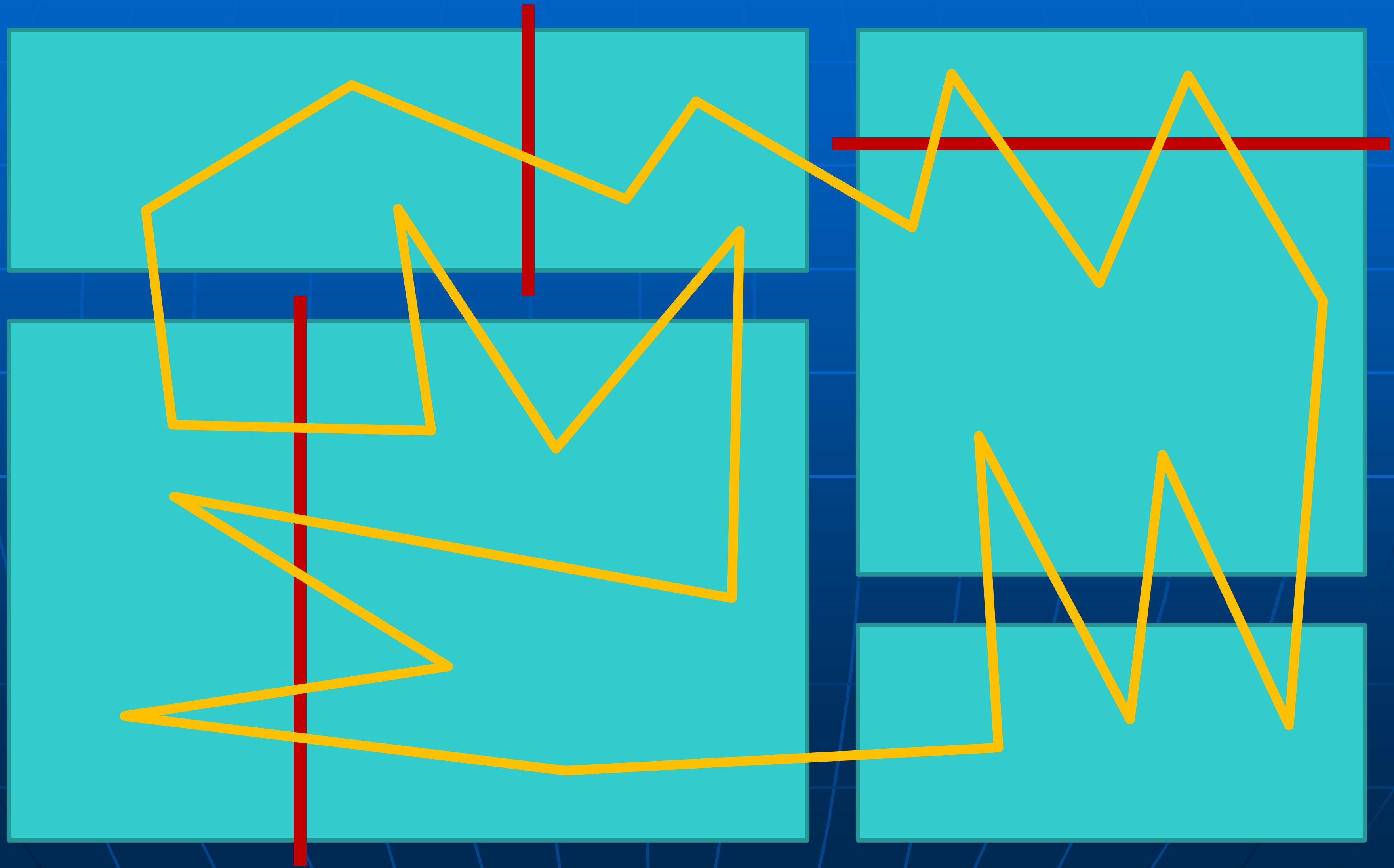


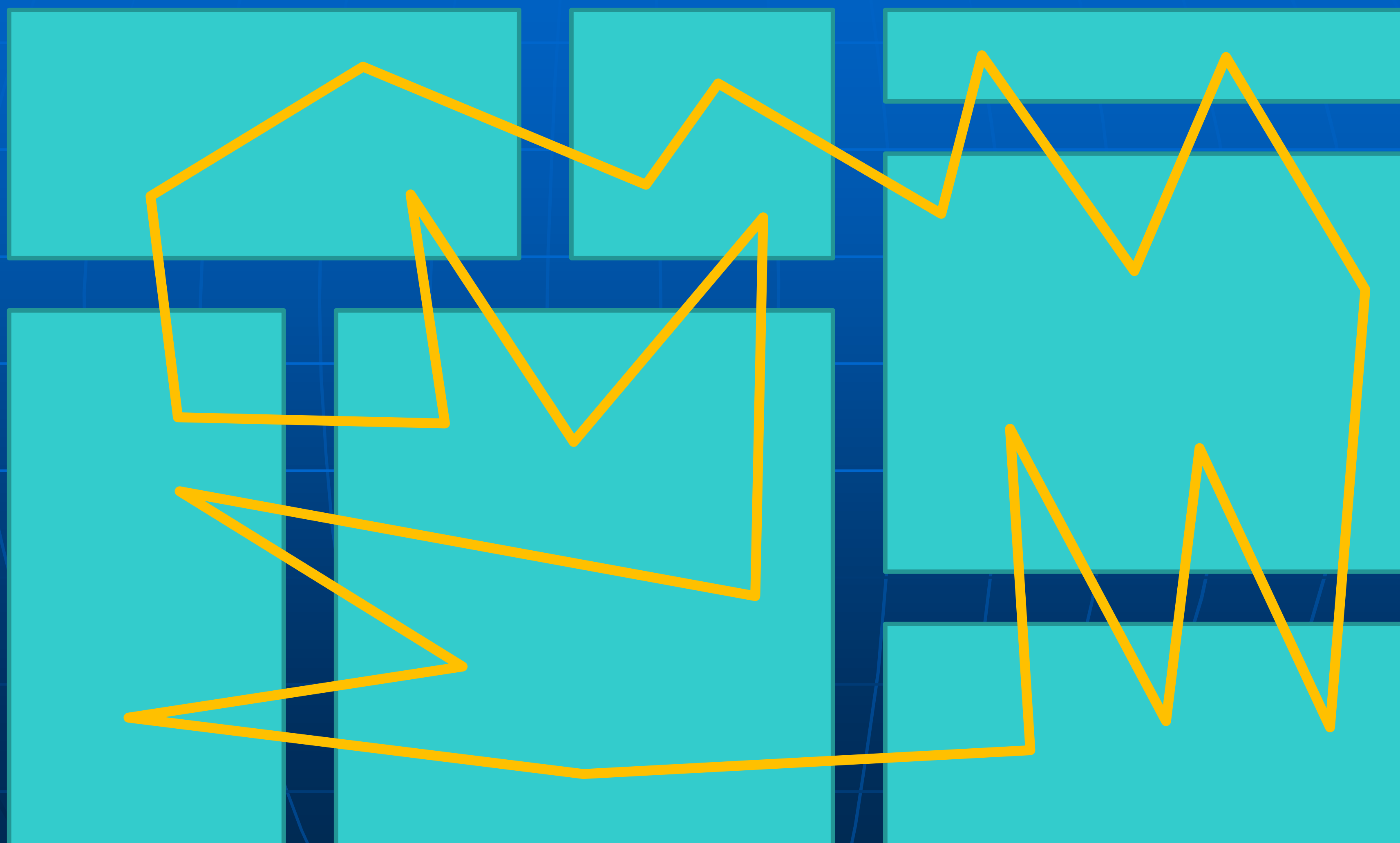








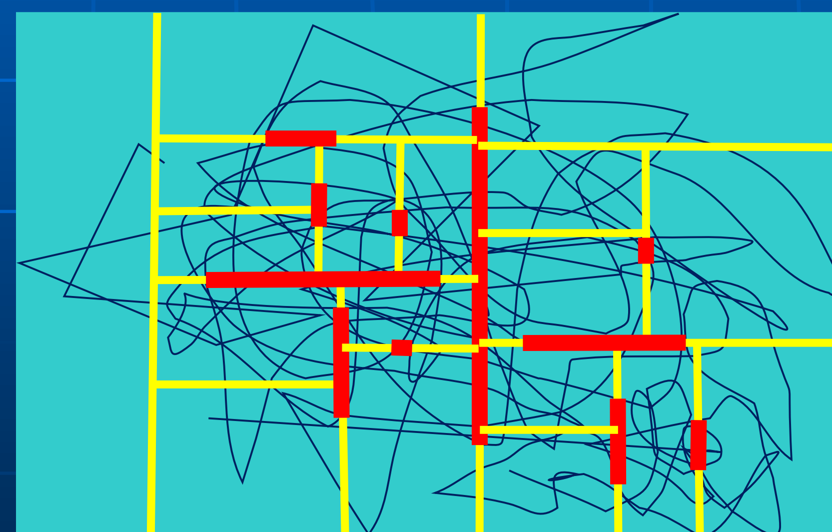




m-Guillotine Structure Theorem

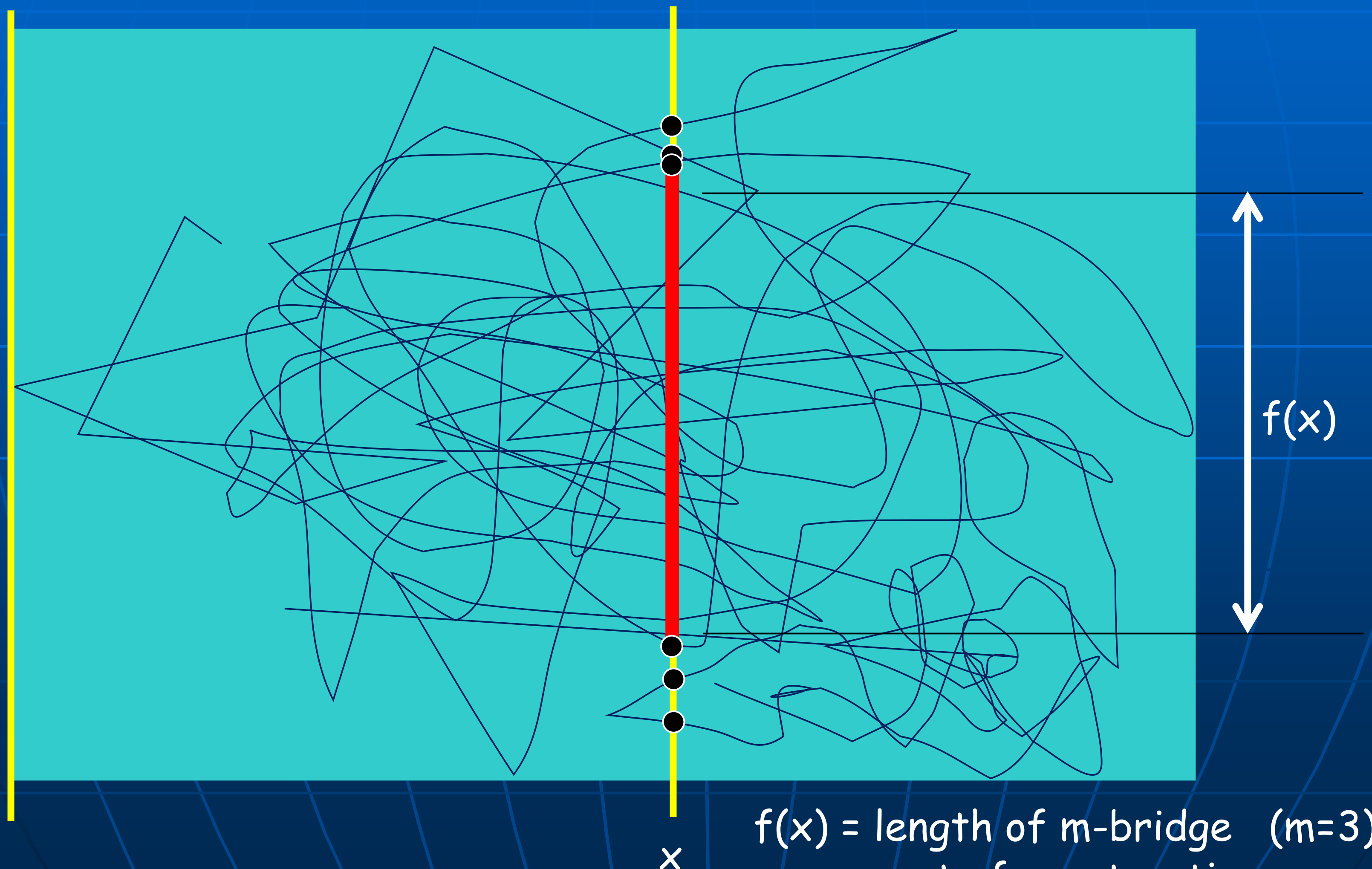
Any set E of edges of length L can be made to be m-guillotine by adding length O(L/m) to E, for any positive integer m.

Proof: Based on a charging scheme.



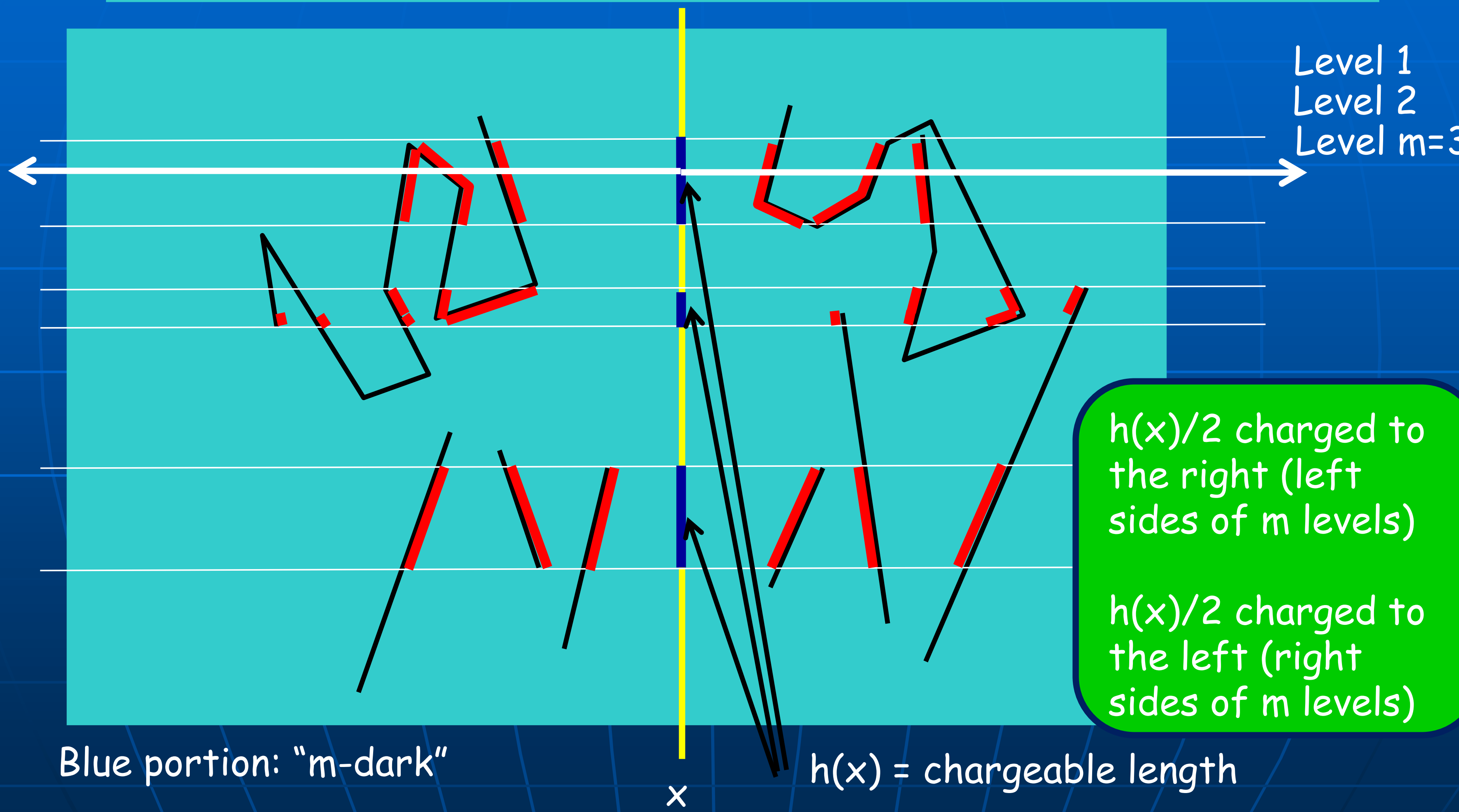
While this "scribble" may not be m-guillotine, it is "close" in that it can be made m-guillotine by adding only (1/m)th of its length

Possible Vertical Cuts



f(x) = length of m-bridge (m=3)
= cost of construction

Paying for the Bridge Construction: The Chargeable Length



h(x)/2 charged to the right (left sides of m levels)

h(x)/2 charged to the left (right sides of m levels)

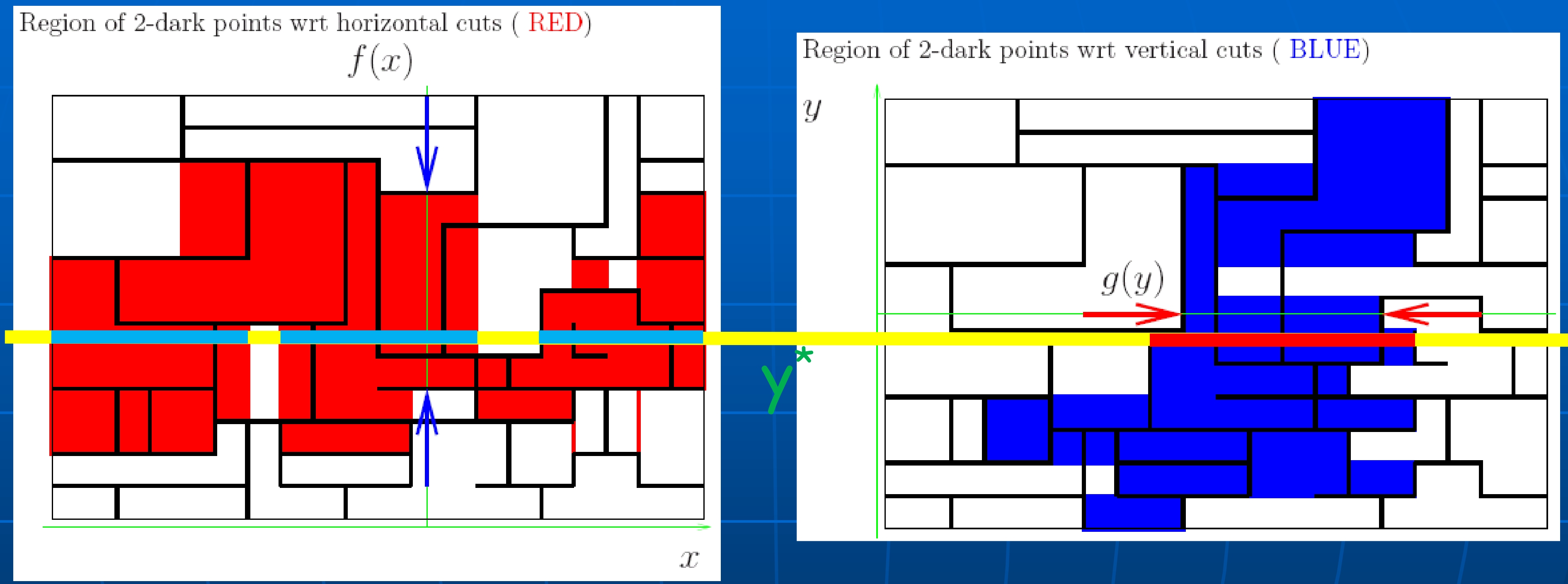
NOTE: After cut, charged lengths never charged again

Key Lemma

- There exists a cut whose chargeable length is ≥ its cost (length of m-span)

There exists a "favorable cut"

Key Lemma: A cut exists with chargeable length ≥ cost (length of m-span)



WLOG: RED area ≥ BLUE area

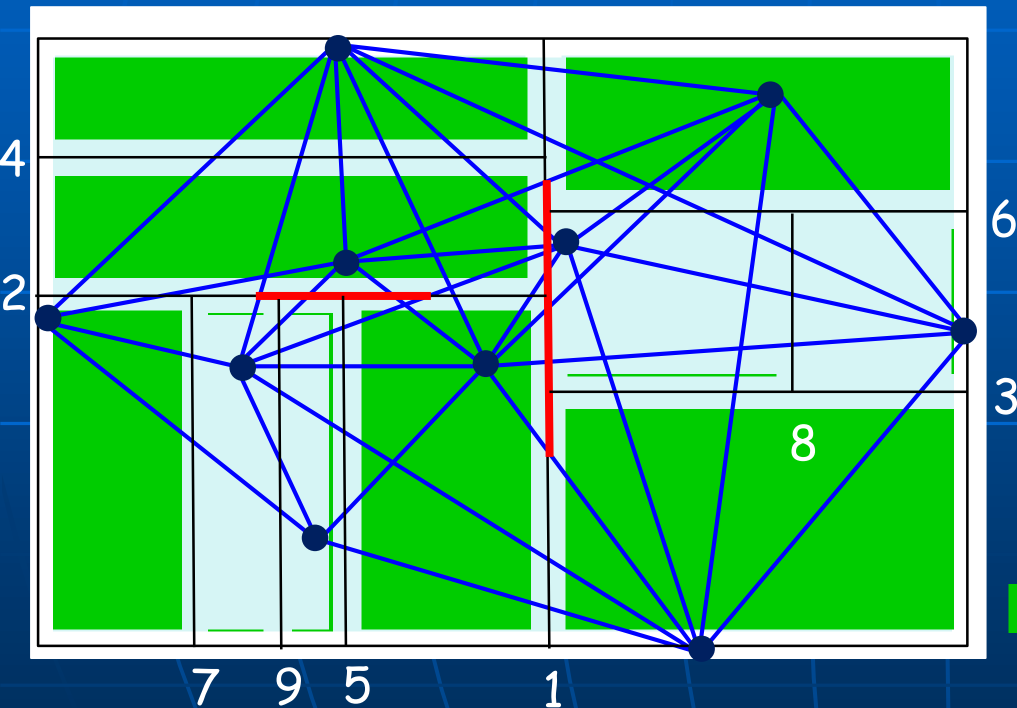
Thus, there exists a y^* such that
m-dark portion of cut $y=y^*$ ≥ $g(y)$ = cost of cut = m-span

Chargeable Length ≥ Cost of Cut

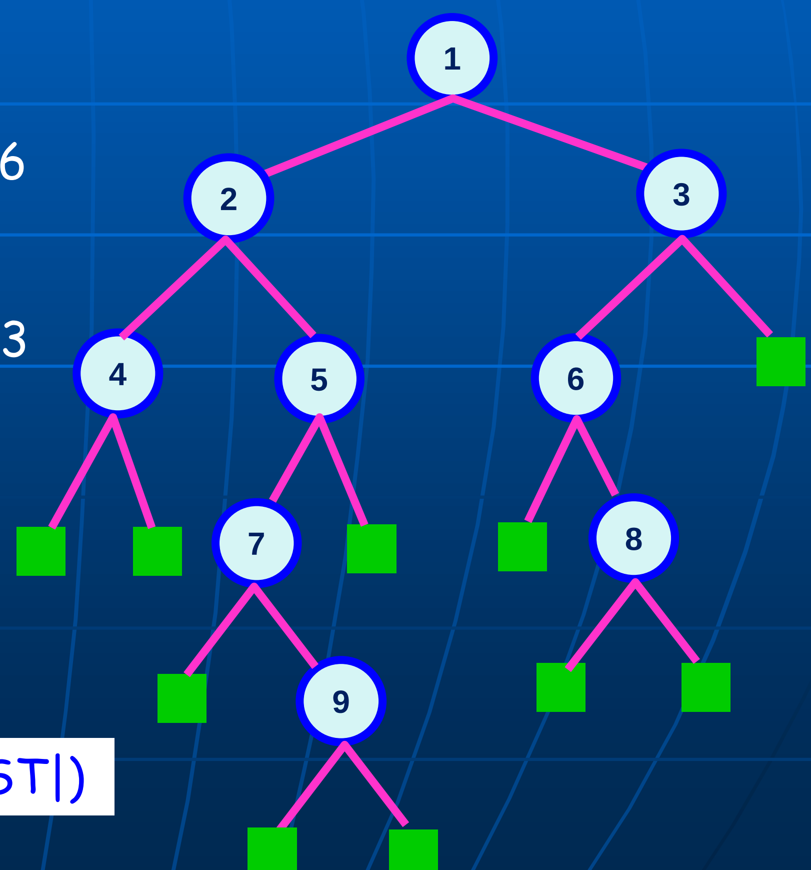
A Simple O(n log n) PTAS

- Build (1+ε)-spanner, G, of n points O(n log n) [Rao-Smith]
- Compute m-guillotine version, G_m O(n log n)
  - Associated (balanced) tree, $T(G_m)$

Points S



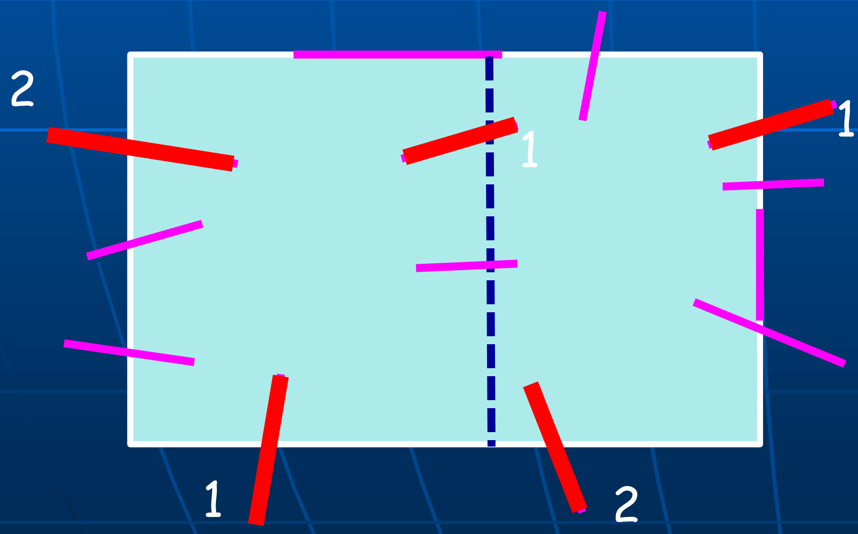
Tree $T(G_m)$



(1+ε)-spanner, G, of weight O(|MST|)

A Simple O(n log n) PTAS

- Build $(1+\varepsilon)$-spanner, G, of n points O(n log n)
- Compute m-guillotine version, G_m O(n log n)
  - Associated (balanced) tree, $T(G_m)$
- DP: Min-weight spanning subgraph of G_m
  - O(n) Subproblems: nodes of $T(G_m)$, plus O(1) info ($2^{O(m)}$)



Specify which subset of the O(m) boundary segs of G_m OPT should use and how many times (1 or 2) it uses it

OPT can be moved onto spanner G (factor $(1+\varepsilon)$), which is subgraph of G_m

– $|OPT_G| \leq (1+\varepsilon)|OPT|$

Sum of bridge lengths added:

$|bridges| = O(\varepsilon |G|) = O(\varepsilon |MST|) = O(\varepsilon |OPT|)$

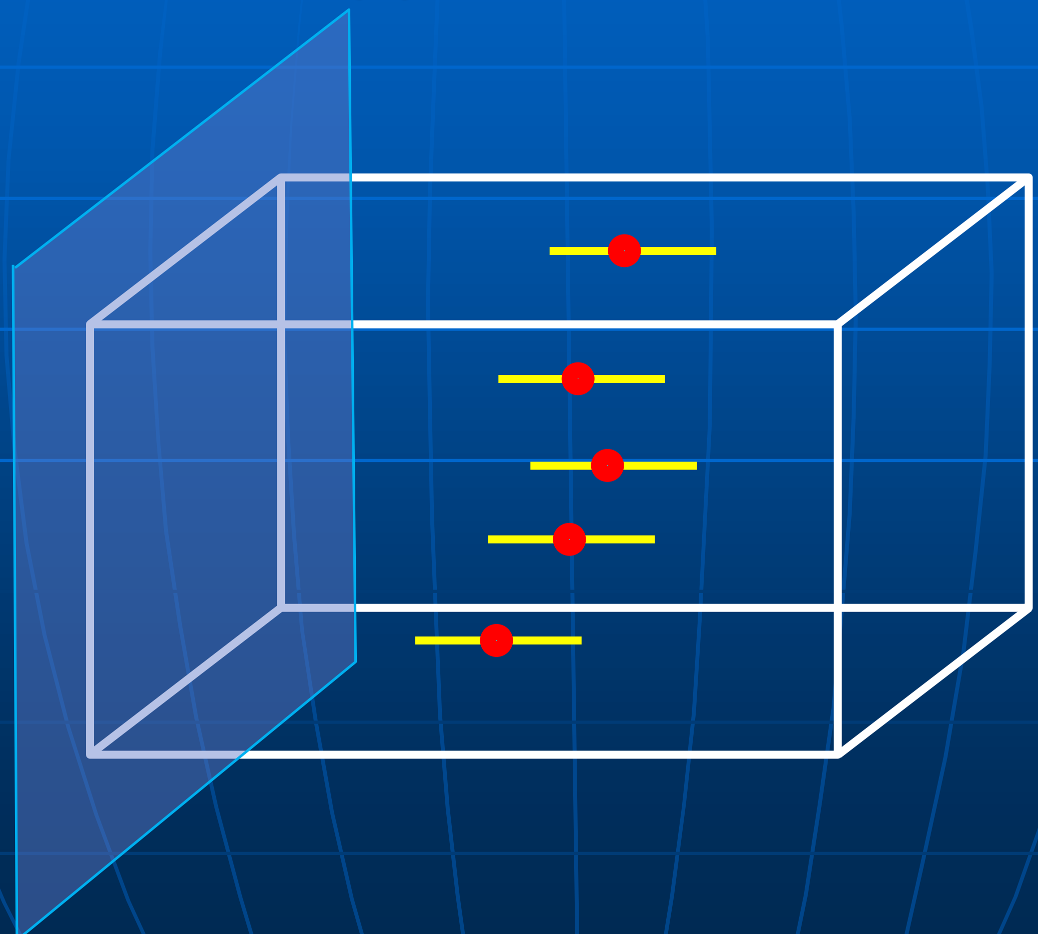
Algorithm computes OPT_{G_m}

$$|OPT_{G_m}| \leq |OPT_G| + |bridges|$$

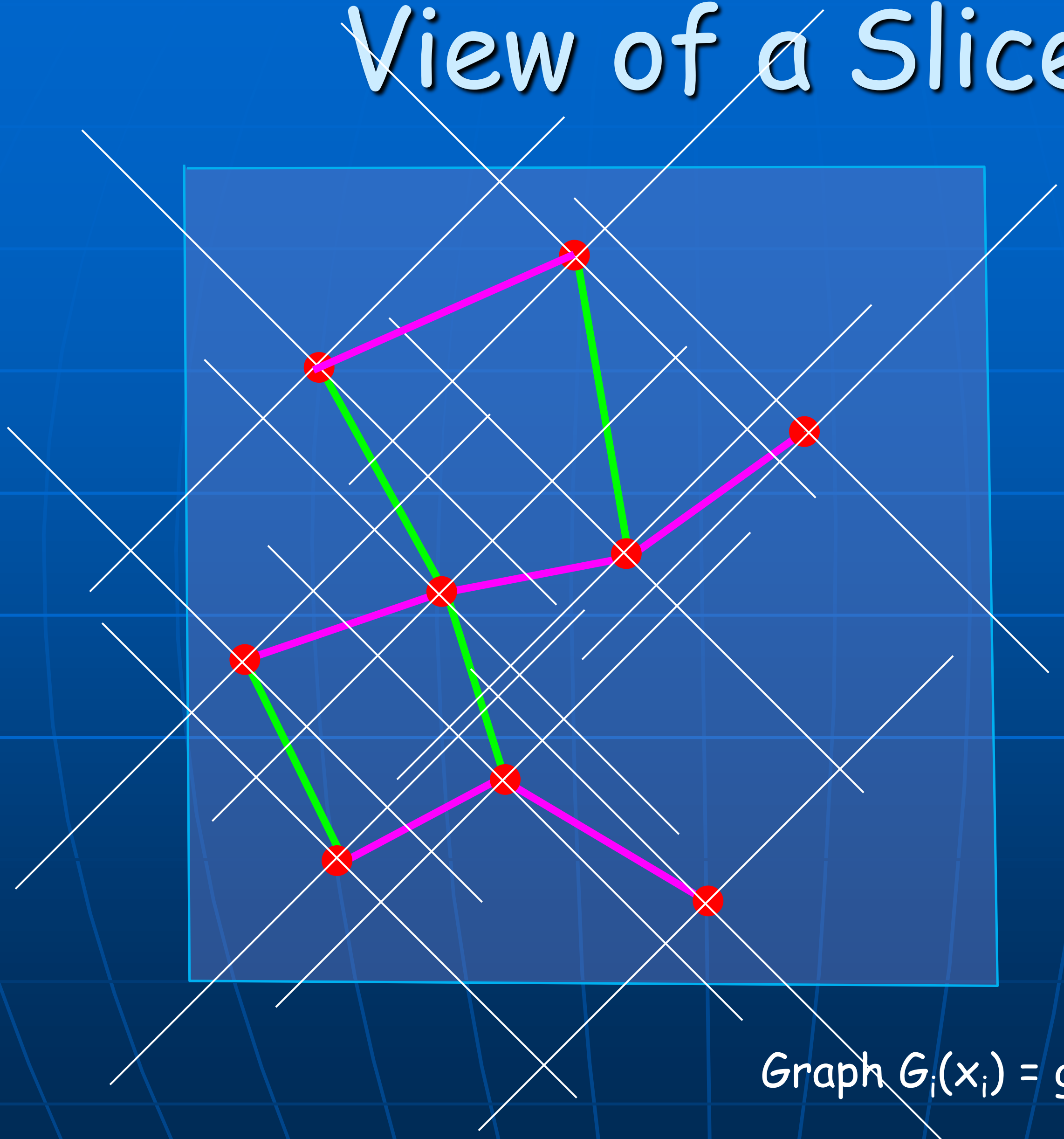
$$\leq (1+\varepsilon)|OPT| + O(\varepsilon |OPT|)$$

Higher Dimensions

- m-guillotine applies:

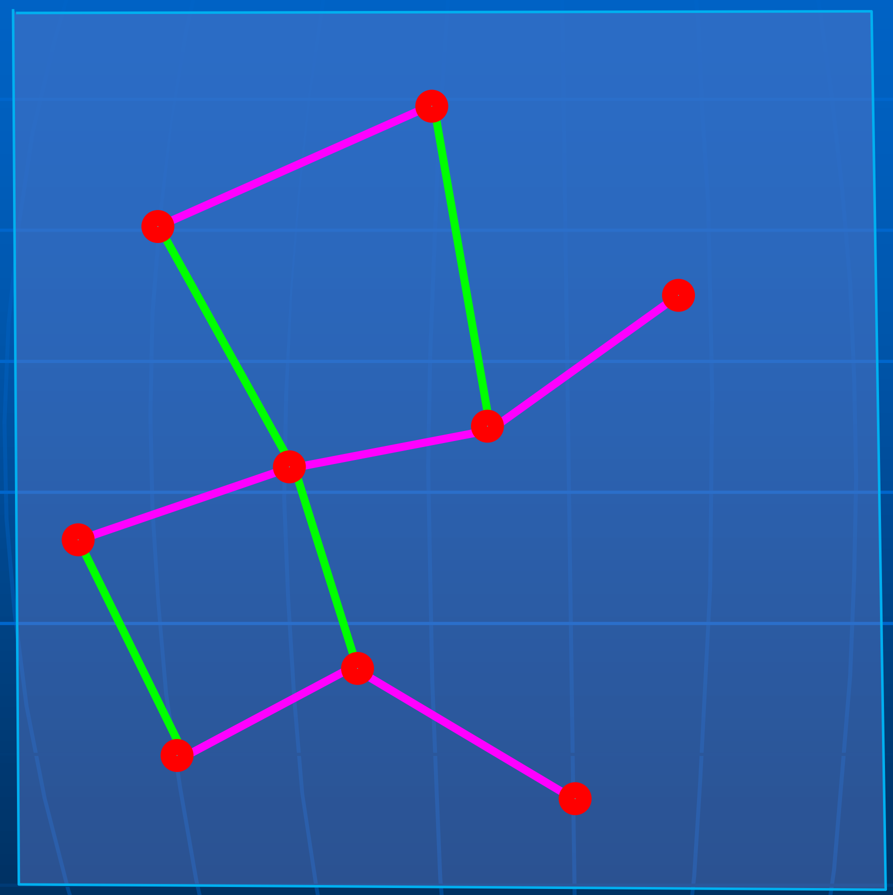


View of a Slice



Cross section orthogonal to x_i

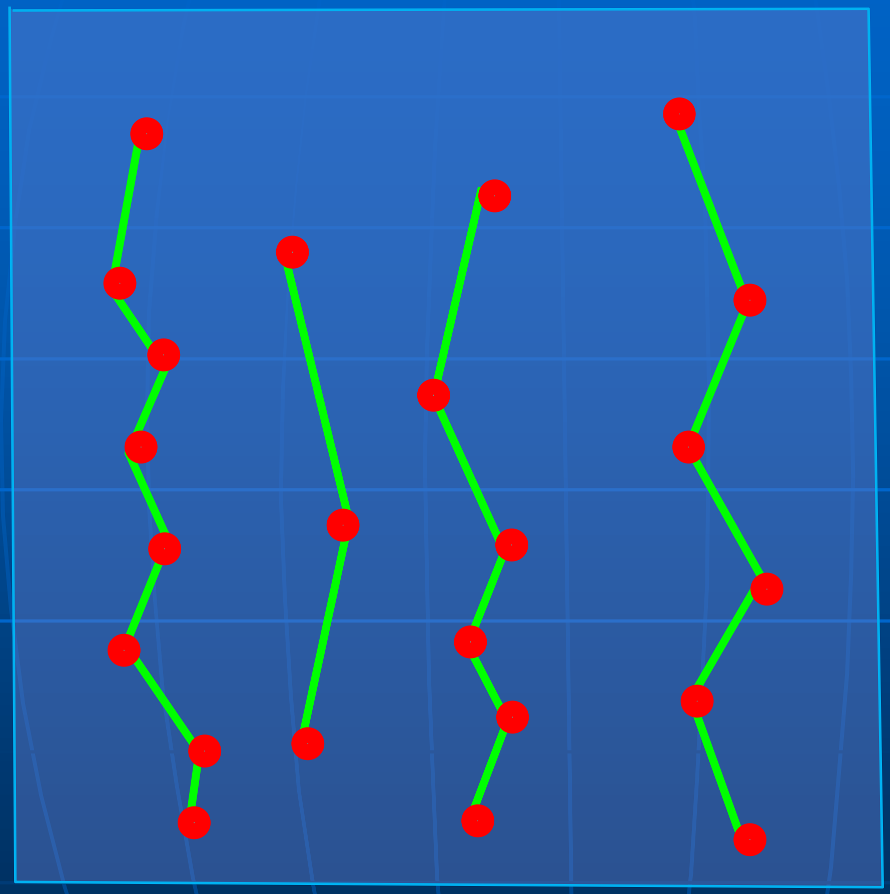
View of a Slice



Graph $G_i(x_i)$ = green $\cup$ magenta edges
= union of disjoint green/magenta paths

Cross section orthogonal to x_i

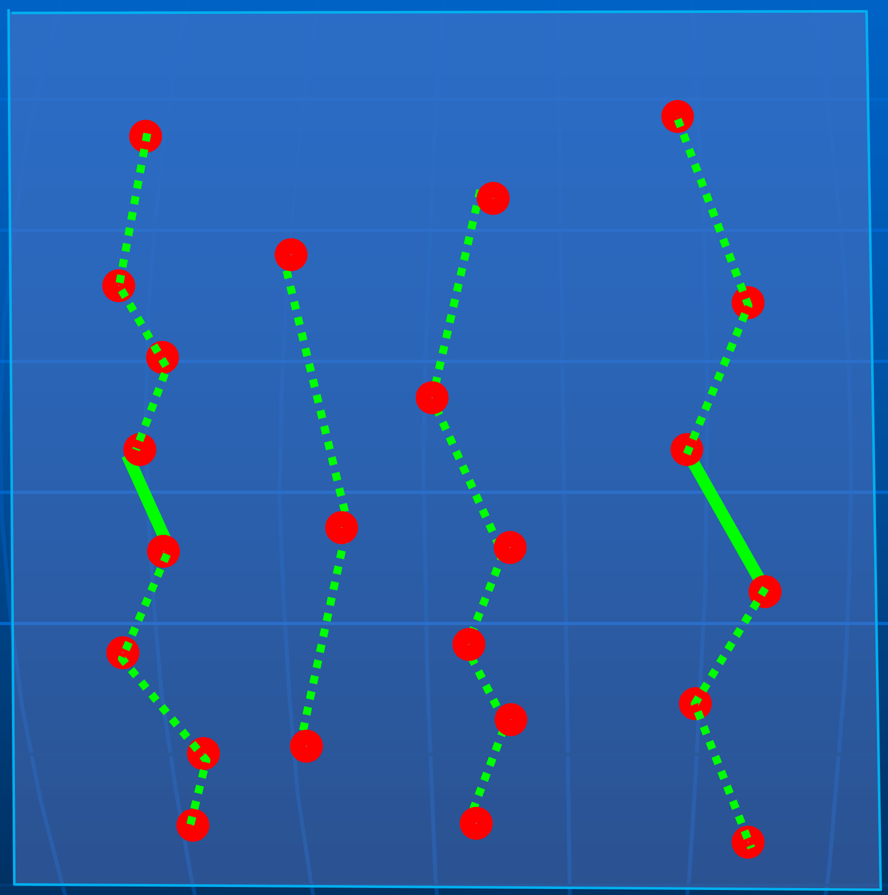
View of a Slice



m-deep green edges

Cross section orthogonal to x_i

View of a Slice



Similarly define m-deep magenta edges.

$G_i^{(m)}(x_i)$ = all m-deep edges

Claim: The m-deep graph, $G_i^{(m)}(x_i)$, has $O(m^{d-1})$ connected components

m-deep (m=3) green edges

Cross section orthogonal to x_i

m-Guillotine in R^d

- "Cost" of cut orthogonal at x_i = length of the m-deep edge set = $f_i(x_i)=|G_i^{(m)}(x_i)|$
  - Integrate wrt x_i = $Area_i$ (of 2-manifold M_i)
- "Chargeable" length of cut = $\sum_{j\neq i} h_j(x_i)$, where $h_j(x_i)$ = length of the x_i-cross-section of M_j
- Lemma: If $Area_i = \min_j(Area_j)$, then there exists x_i^* with $f_i(x_i^*) \leq 1/(d-1) \sum_{j\neq i} h_j(x_i^*)$

i.e., there exists a favorable cut

m-Guillotine in R^d

- Each unit of edge length parallel to x_j gets charged at most (d-1) times, each time with 1/m(d-1) of its length (and will not be charged again after the cut)

Other Metrics

- General metric TSP: APX-hard
- Bounded intrinsic (doubling) dimension:

Doubling dimension k=ddim(S) of finite metric space S: Every ball of radius r can be covered by 2^k balls of radius r/2

- Talwar'04: QPTAS $2^{(\mathrm{ddim}(S)/\varepsilon \cdot \log n)^{O(\mathrm{ddim}(S))}}$
- Bartal-Gottlieb-Krauthgamer'12: PTAS

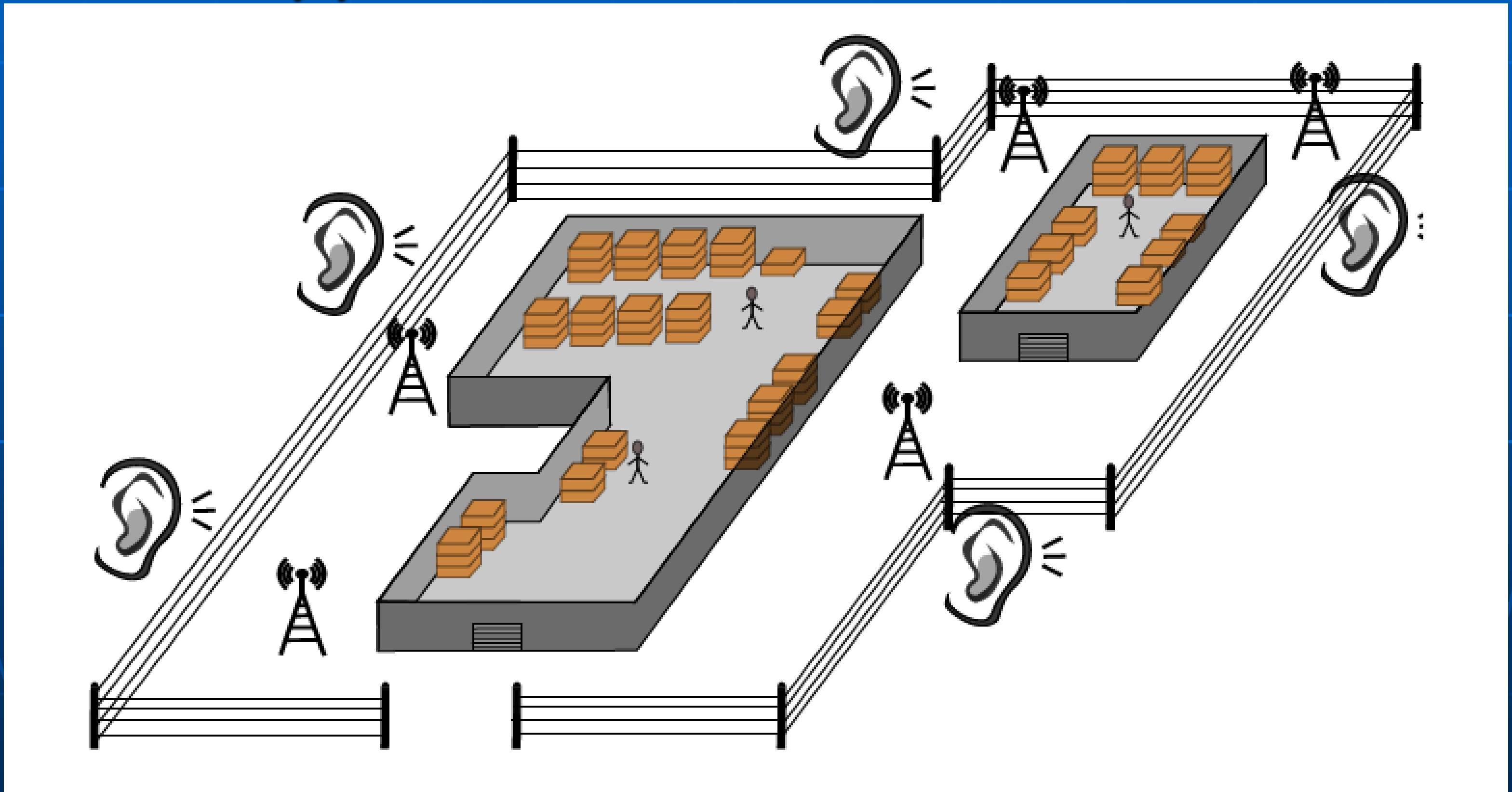
Theorem 1.3. *A $(1+\varepsilon)$-approximation to the optimal tour of a metric TSP instance S on $n = |S|$ points can be computed by a randomized algorithm in time* $n^{2^{O(\mathrm{ddim}(S))}} \cdot 2^{(2^{\mathrm{ddim}(S)}/\varepsilon)^{O(\mathrm{ddim}(S))}\sqrt{\log n}}$.

Many Applications

- Optimal paths, tours, trees
- Min weight networks
- Optimal partitioning
- Packing problems, MIS
- Covering problems (disk cover, guarding, jamming)
- Facility location, relay placement
- Optimal separation
- Robot localization

Protective Jamming

"Guards" provide jamming: protect from eavesdroppers



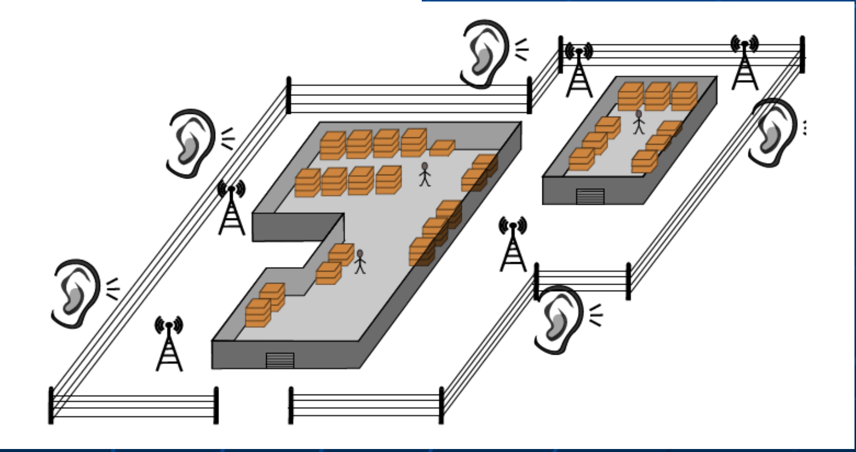
[Sankararaman,Abu-Affash,Efrat,Eriksson-Bique,Polishchuk,Ramasubramanian,Segal MobiHoc 2012]

Protective Jamming

THEOREM 4. *Given storage region(s) $\mathcal{S}$, fence $\mathcal{F}$, thresholds $\delta_{\mathsf{s}}, \delta_{\mathsf{e}}$ and jammer power $\hat{P}$, under the* NJ *interference model, we can compute a set of locations $J \subset \mathcal{F} \setminus \mathcal{S}$ in time $O((T/\varepsilon^{O(1)})^{O(1/\varepsilon^2)})$ where $T = \min\{\mathcal{L}_{\mathcal{F}}^2, \mathcal{L}_{\mathcal{S}}^2, n^2\mathsf{OPT}^2\}$ such that $|J| \le (1+\varepsilon)\mathsf{OPT}$ and if jammers of power $\hat{P}$ are placed at J,*

(i) For any point $p_e \in \mathcal{E}$, $\mathsf{SIR}(J, p_e) < (1+\varepsilon)\delta_{\mathsf{e}}$.

(ii) For any point $p_s \in \mathcal{S}$, $\mathsf{SIR}(J, p_s) > \delta_{\mathsf{s}}$.



[S. Sankararaman, E. Arkin, Y. Cassuto, A. Efrat, J. Mitchell, M. Segal, 2014]

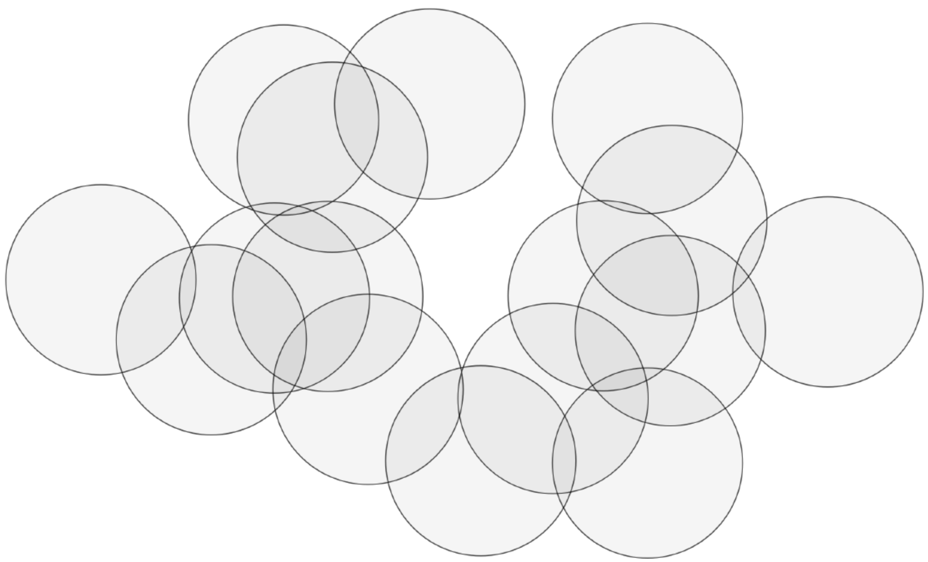
MACS: Maximum Area Connected Subset

Definition: Connected Unit-disk k-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer k

Out: A connected subset S of size k

Goal: Maximize the area covered by S



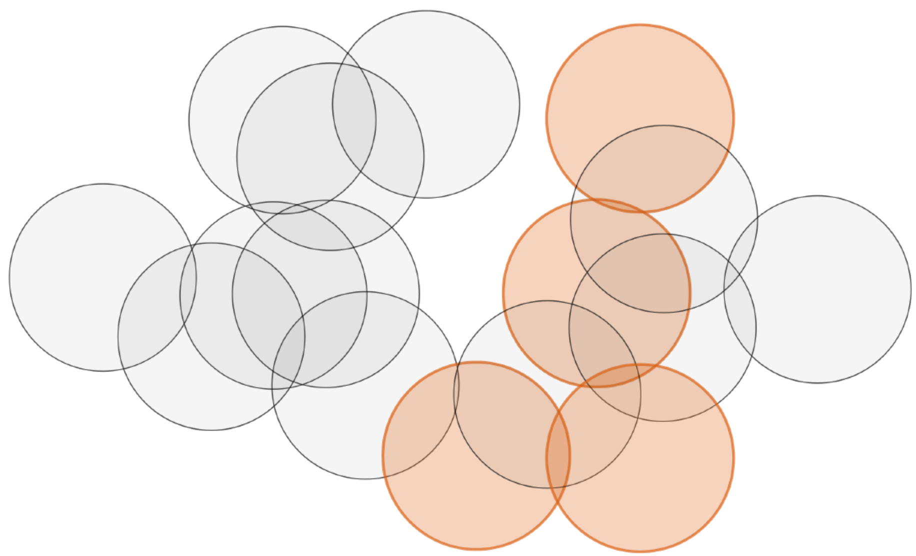
[Chien-Chung Huang, Mathieu Mari, Claire Mathieu, Joseph S. B. Mitchell, and Nabil Mustafa, APPROX 2019]

Definition: Connected Unit-disk *k*-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer k

Out: A connected subset S of size k

Goal: Maximize the area covered by S



$k = 4$

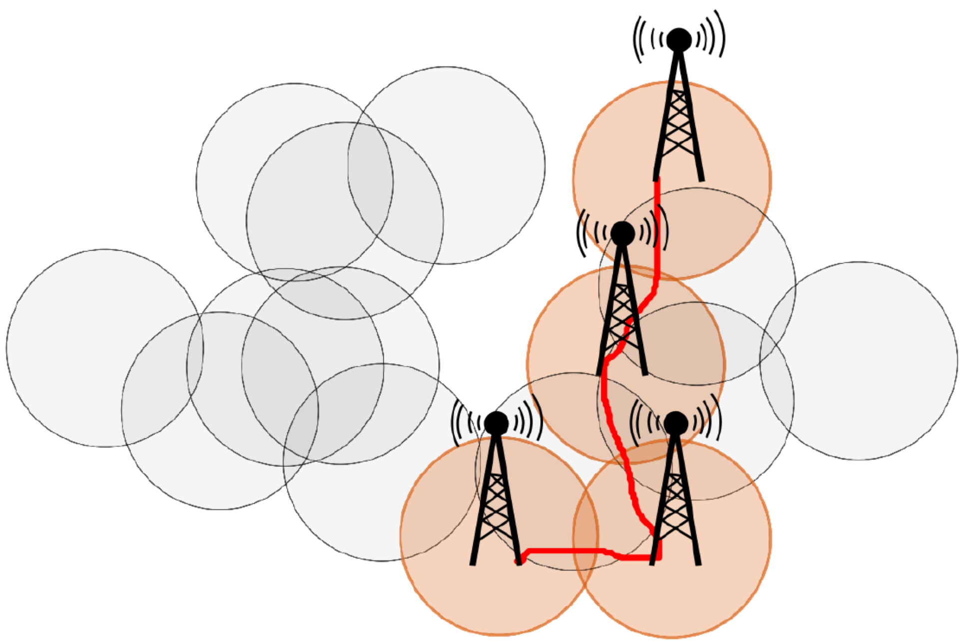
[Chien-Chung Huang, Mathieu Mari, Claire Mathieu, Joseph S. B. Mitchell, and Nabil Mustafa, APPROX 2019]

Definition: Connected Unit-disk k-coverage

In: A (connected) set of unit-area-disks in the Euclidean plane and an integer k

Out: A connected subset S of size k

Goal: Maximize the area covered by S



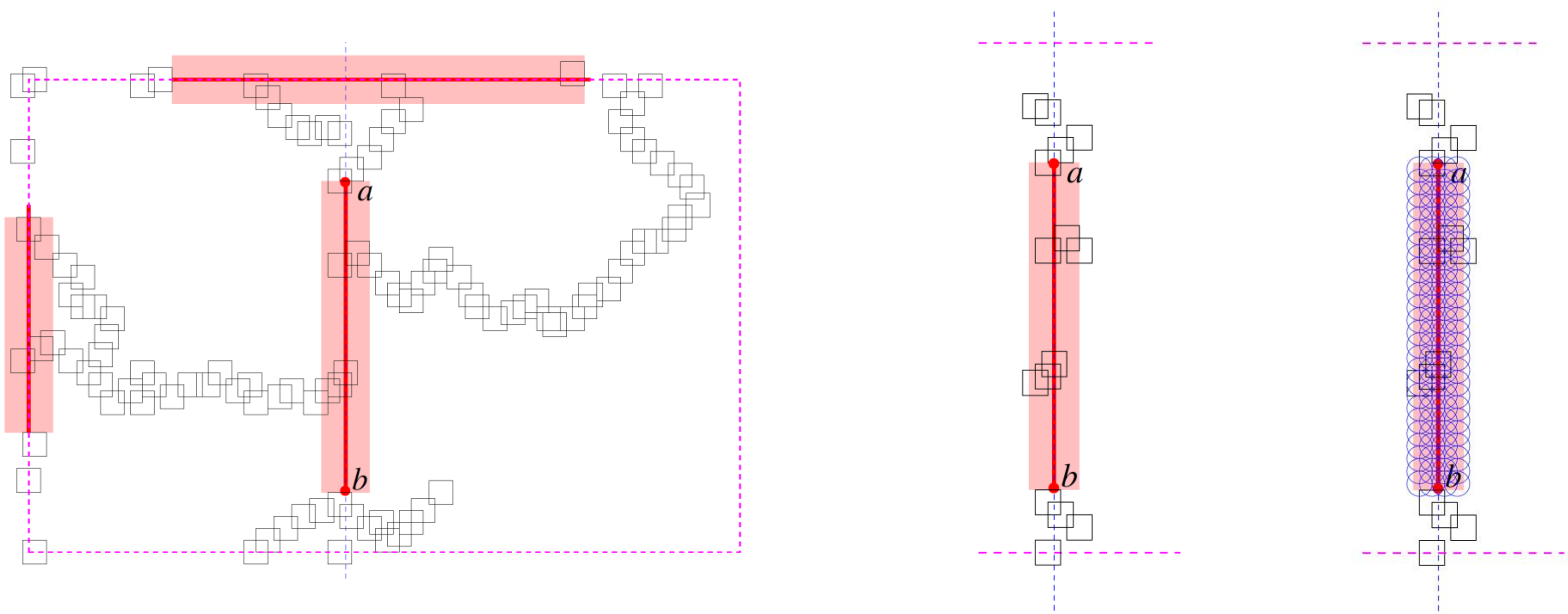
[Chien-Chung Huang, Mathieu Mari, Claire Mathieu, Joseph S. B. Mitchell, and Nabil Mustafa, APPROX 2019]

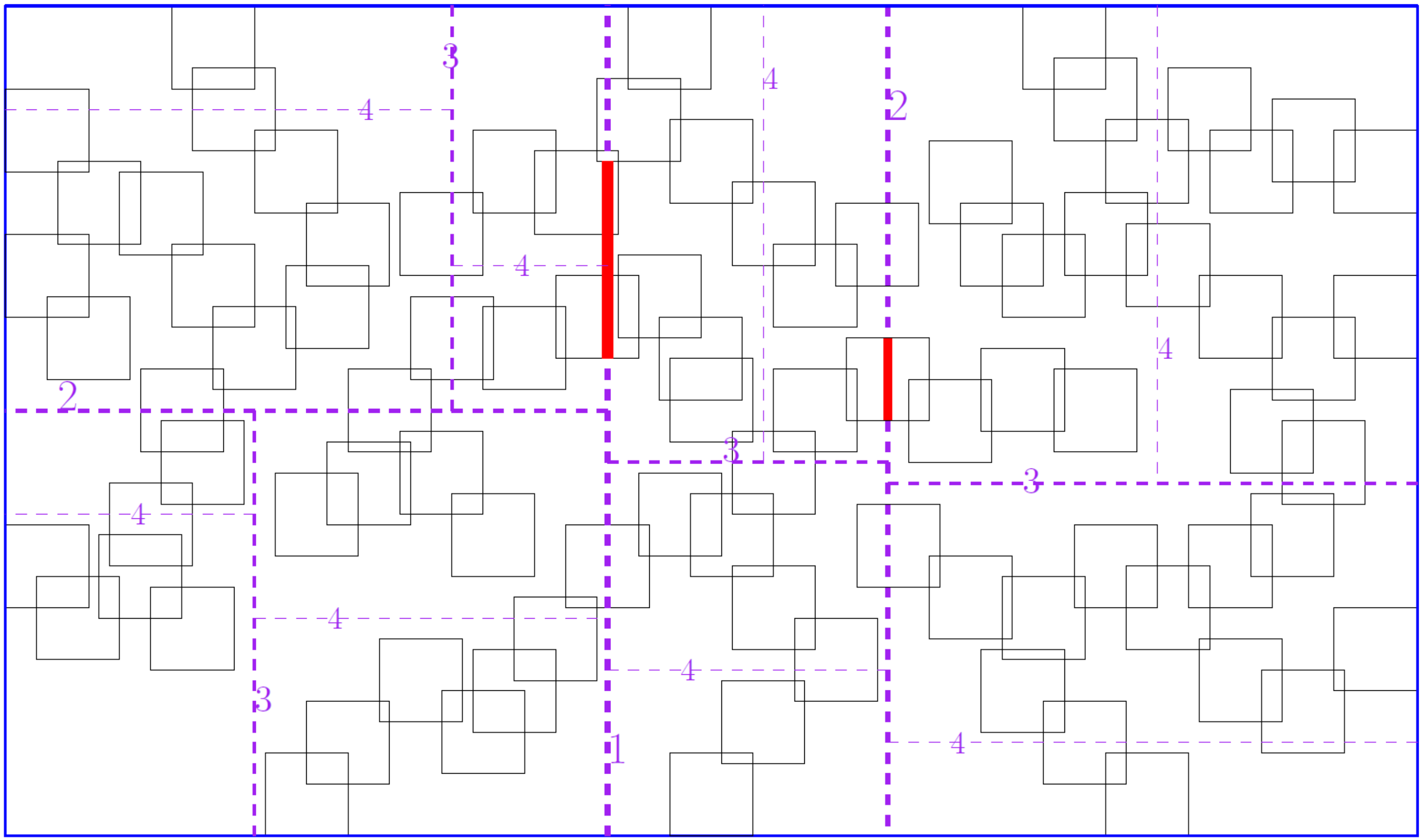
Theorem 1 (Hardness)**.** MACS *is NP-hard.*

Theorem 2 (Approximation)**.** MACS *has a* $(1/2)$*-approximation that can be computed in polynomial time (Algorithm 1).*

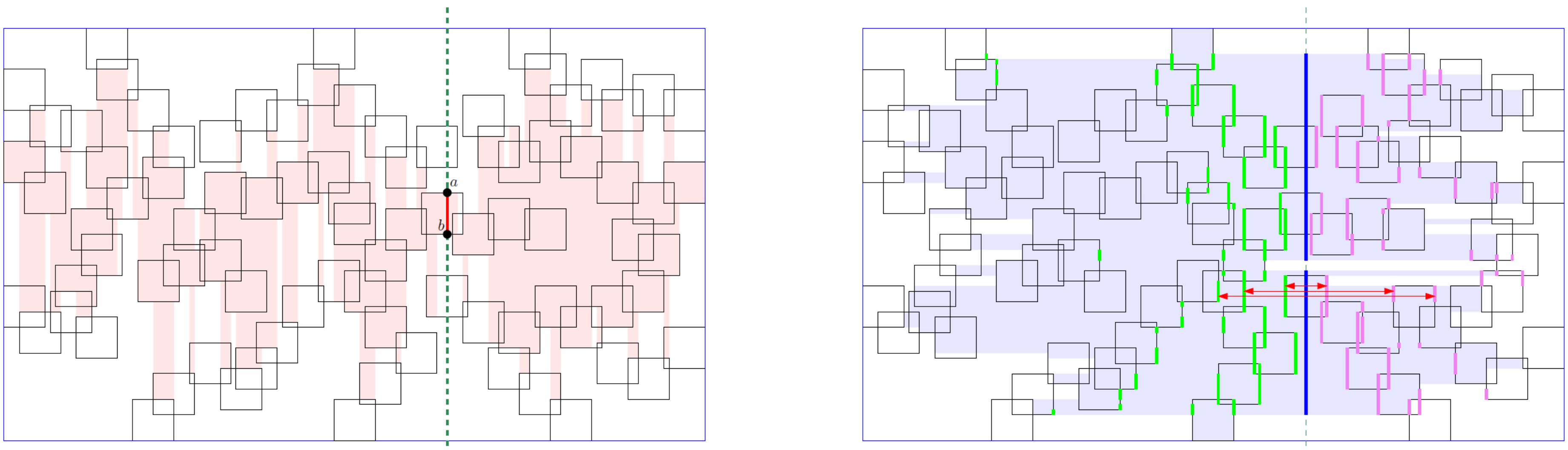
With resource augmentation, we obtain a $(1-\varepsilon)$-approximation.

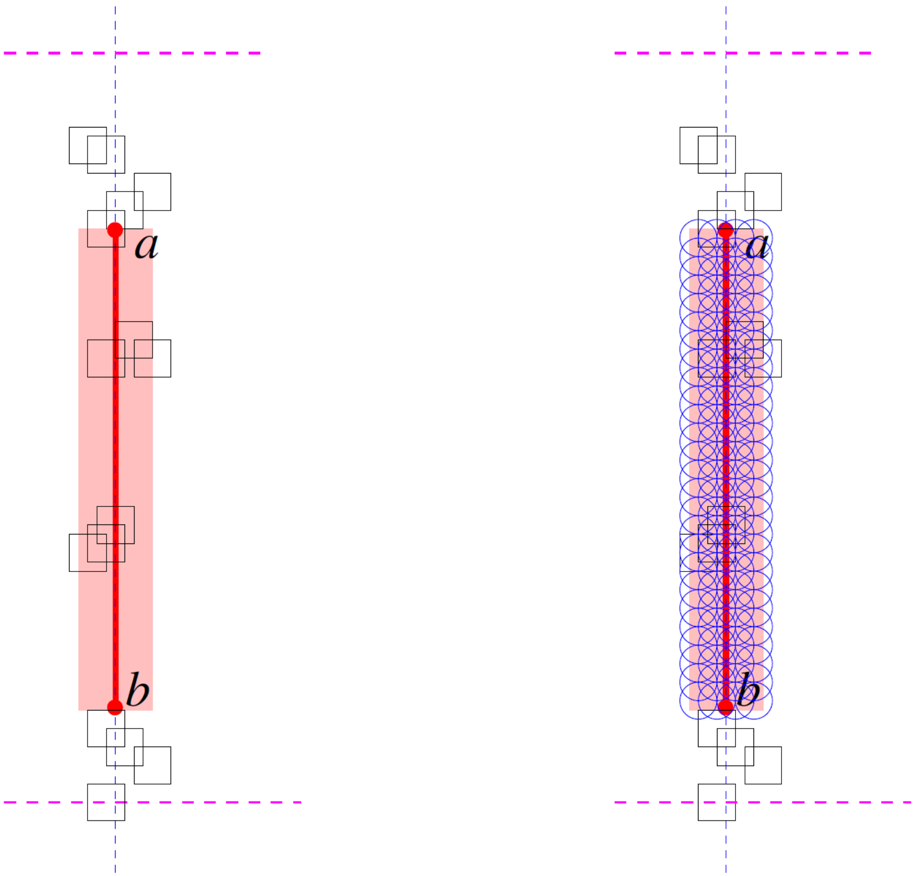
Theorem 3 (Resource augmentation)**.** *Let* $\varepsilon > 0$ *be fixed. Given a set* $X \subseteq \mathbb{R}^2$ *of points and a positive integer* k*, there is a deterministic algorithm that computes, in time* $n^{O(\varepsilon^{-1})}$*, a subset* $S \subseteq X$ *of size at most* k *and a set* $S_{add} \subseteq \mathbb{R}^2$ *of at most* εk *points, such that* $UDG(S \cup S_{add})$ *is connected, and the area covered by the unit disks centered at* S *is at least* $(1-\varepsilon)\mathbf{OPT}(X,k)$*.*





3
4
4
2
4
4
2
4
3
3
4
4
4
3
1
4

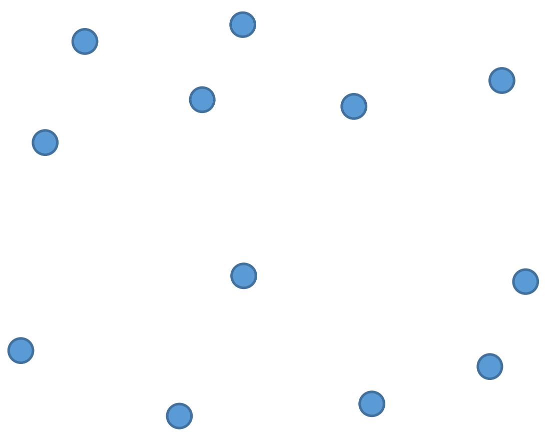

a
b



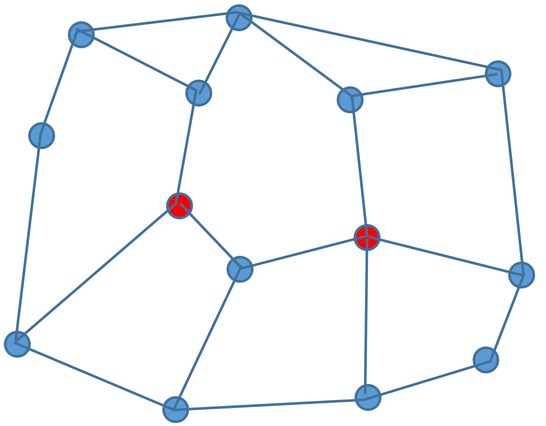
a
b
a
b

Application: Min-Weight Convex Subdivision

- Given a set S of n points in 2D
- Goal: An embedded planar graph $G = (V, E)$ with straight edges (E), convex faces, and $S \subseteq V$, having minimum total Euclidean length of edge set E.



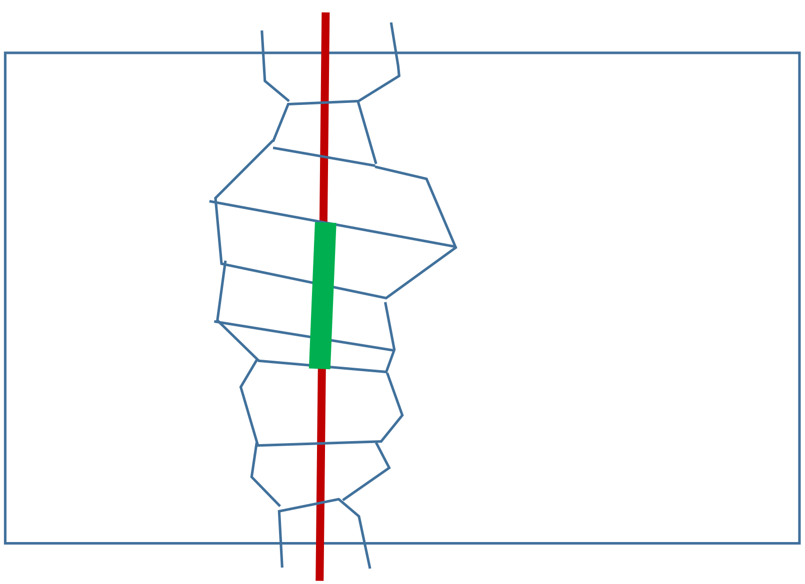
Input



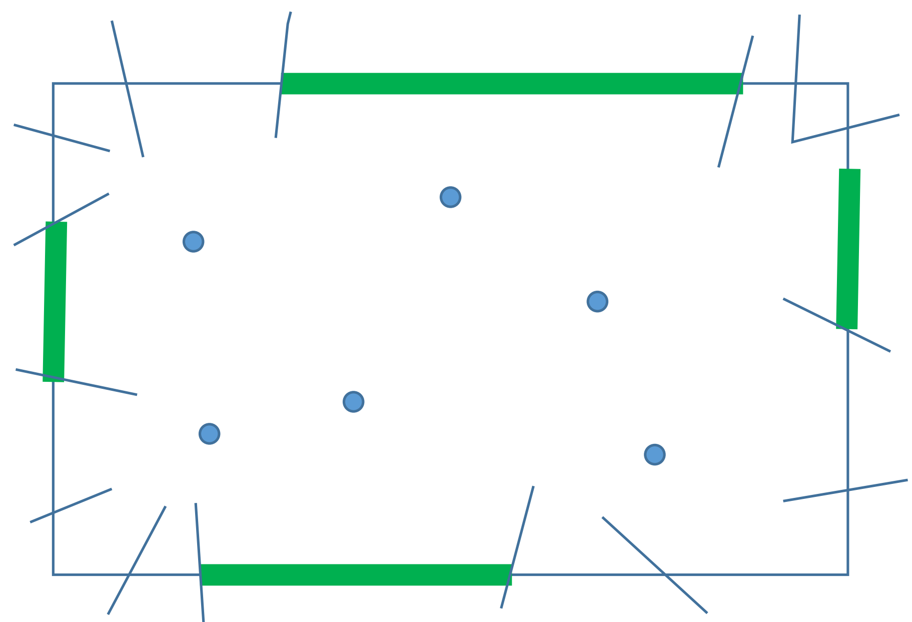
Output

PTAS

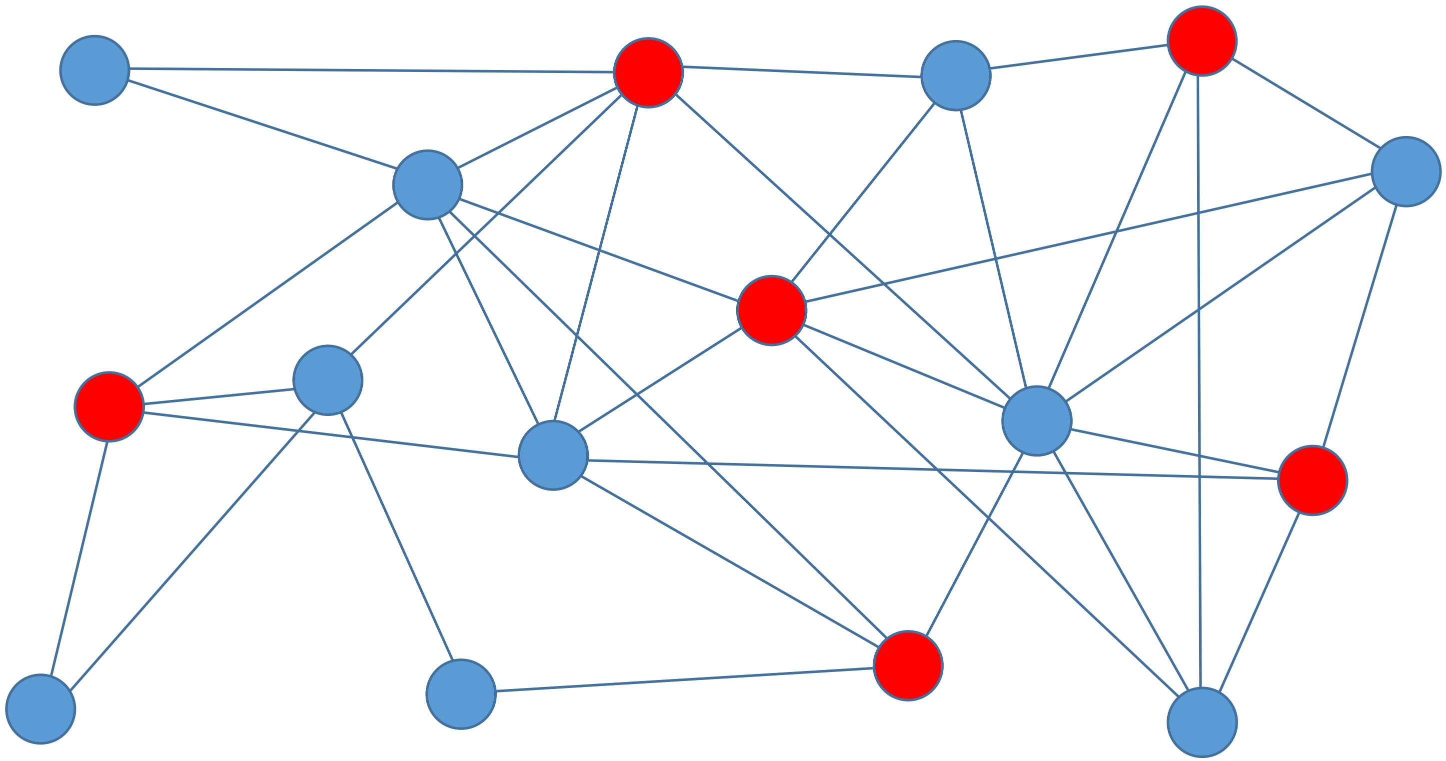
- Structure Thm implies OPT can be transformed to be m-guillotine, increasing length by factor $(1+\epsilon)$, adding Steiner points.
- Algorithm: DP, searching for min-weight m-guillotine convex subdivision



Subproblem:



Maximum Independent Set (MIS)



Best known polytime approx factor: $O(n/\log^2 n)$ [Boppana-Halldórsson]

No polytime algorithm with approx $n^{1-\delta}$ for $\delta>0$, unless P=NP [Zuckerman]

PTAS in planar graphs

Geometric Instances of MIS

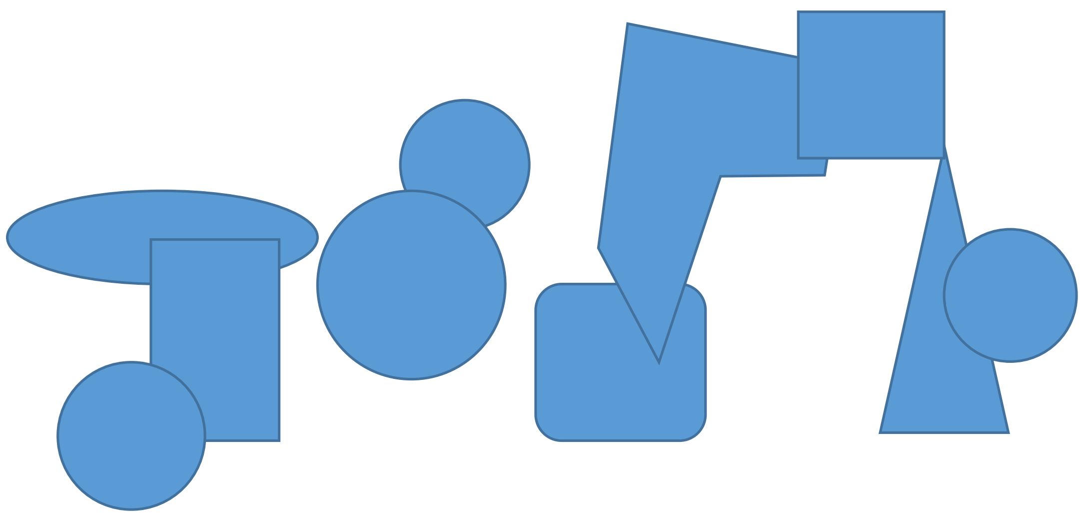
- Graph G given by intersection graph of objects
  - Disks, squares, fat regions
  - Pseudodisks
  - Rectangles (MISR)
  - Polygons
  - General connected sets, with various assumptions
  - Outerstring graphs
  - etc

A Basic Geometry Problem

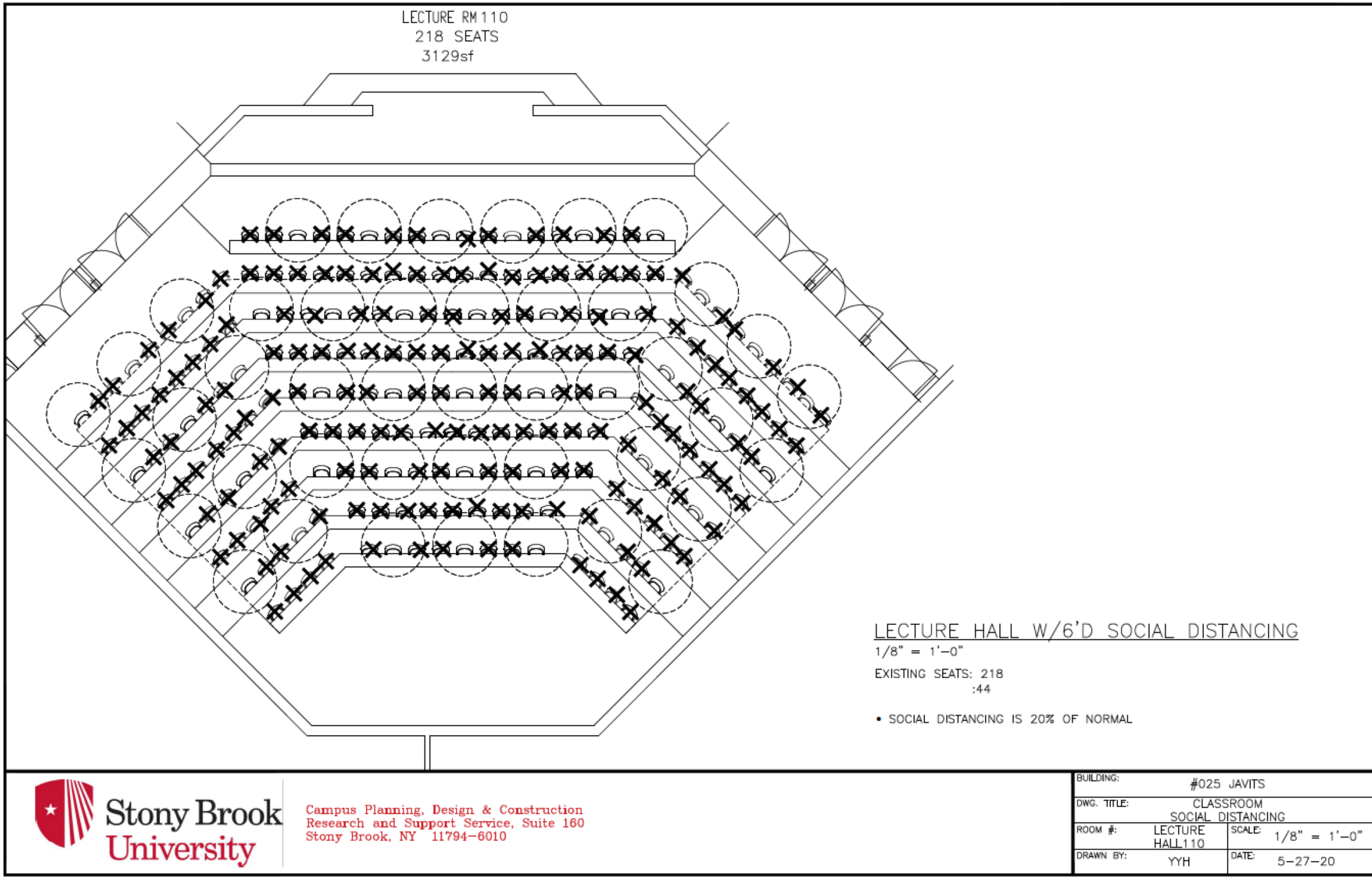
Maximum Independent Set (MIS):

Given a set S of bodies in the plane.

Find a max-cardinality subset, S^*, that is pairwise-disjoint.

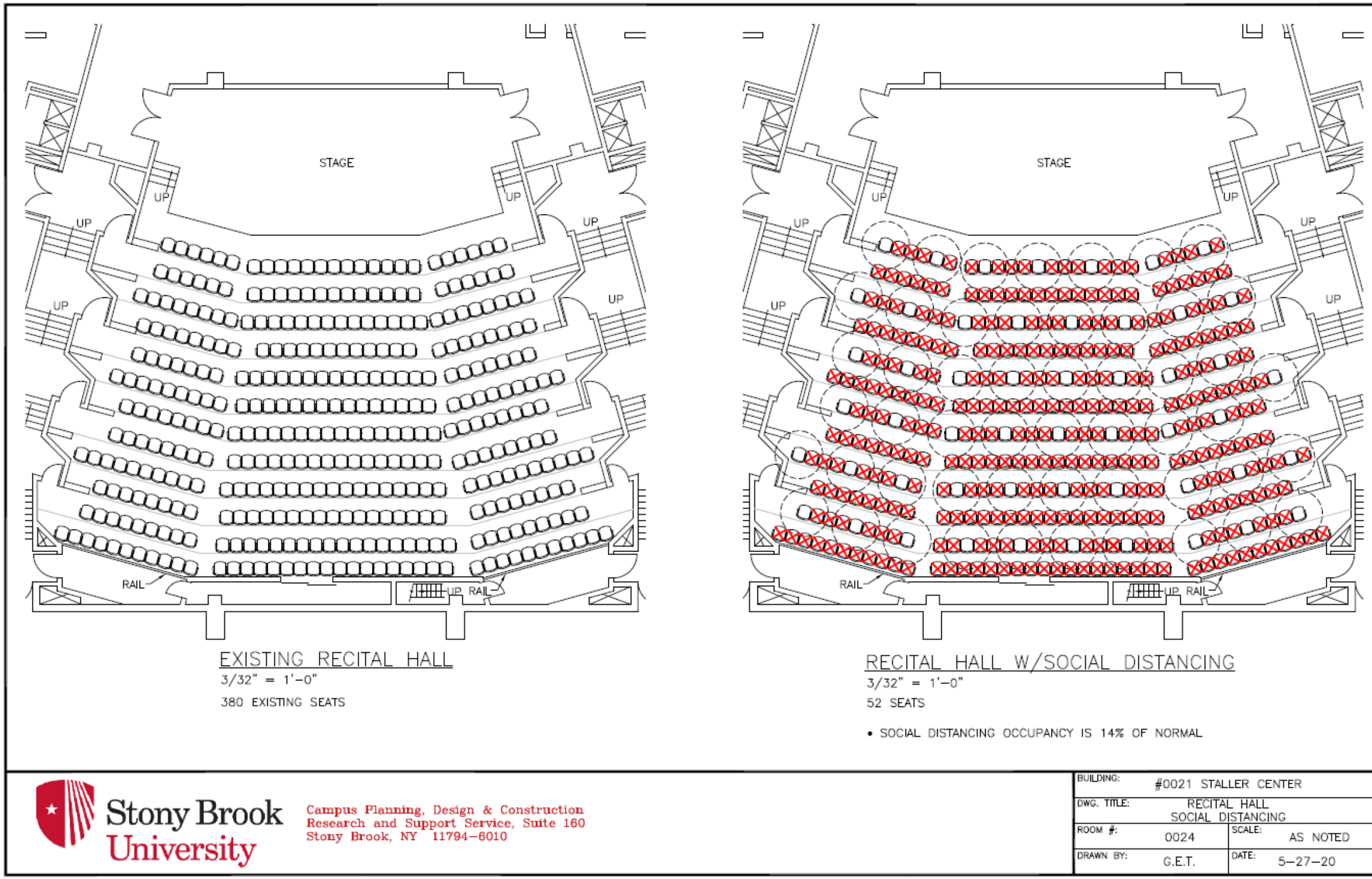


MIS=Most Efficient Social Distancing



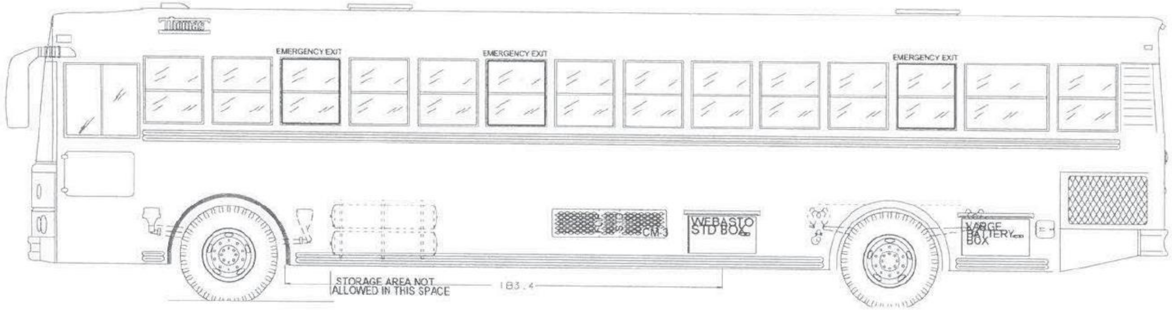
MIS=Most Efficient Social Distancing

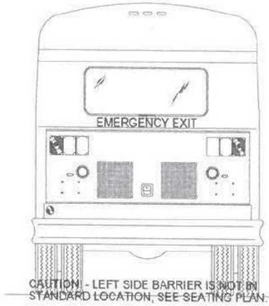
Figure 5 – Lecture Hall Social Distancing Mock-Up



CAD DRAWING - DO NOT MAKE MANUAL CHANGES

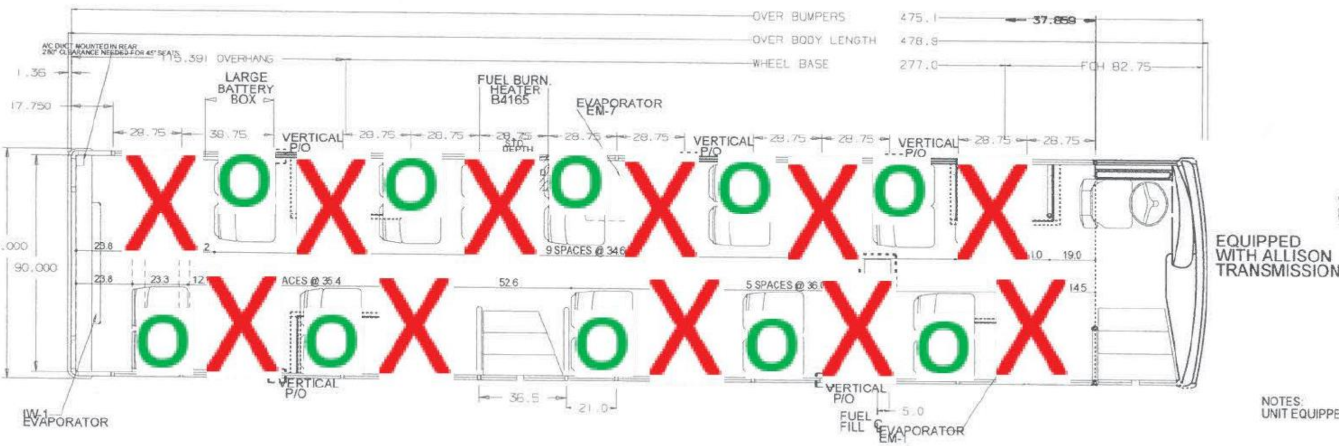
SEATING CAPACITY: 42 + DRIVER

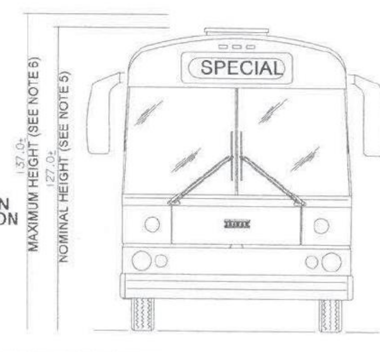




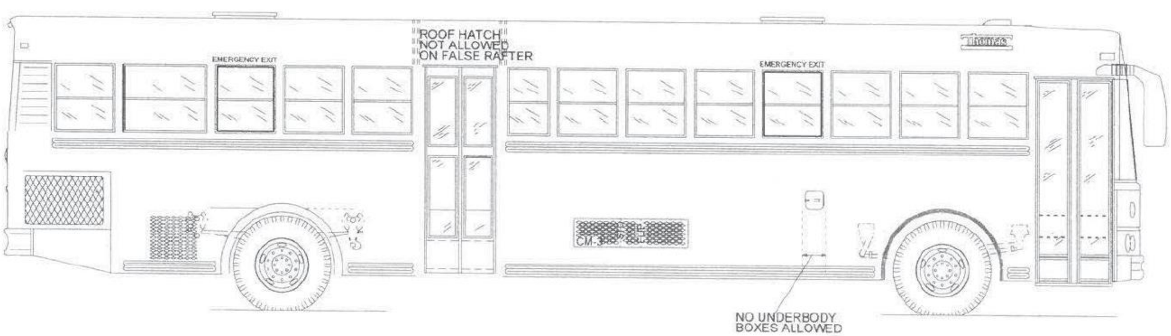
CAUTION! - LEFT SIDE BARRIER IS NOT IN STANDARD LOCATION, SEE SEATING PLAN.

CAUTION! - RIGHT SIDE BARRIER IS NOT IN STANDARD LOCATION, SEE SEATING PLAN.





NOTES:
UNIT EQUIPPED WITH PLYWOOD FLOOR



Model: Saf-T-Liner HDX
Quote Number: 345931
Locality: NY

ALL DIMENSIONS ARE FOR REFERENCE ONLY

GENERAL NOTES

1. SOME ITEMS, SUCH AS CV MIRRORS, ROOF LUGGAGE RACKS, AND OTHER ITEMS, ARE SHOWN IN ONE VIEW ONLY FOR CLARITY.
2. THIS DRAWING IS A REPRESENTATION ONLY AND MAY NOT HAVE ALL ITEMS REQUIRED. PLEASE REFER TO ORDER!
3. THE CLEARANCE BETWEEN BOTTOM OF BUS OR BOTTOM OF UNDERBODY COMPARTMENTS AND GROUND WILL VARY ACCORDING TO TIRE SIZE, BUS LOAD, AND SUSPENSION TYPE.
4. THE MAXIMUM WIDTH AT BELT LINE OVER GUARD RAILS IS 96"
5. THE NOMINAL BUS HEIGHT IS BASED ON A STANDARD BODY AND LARGEST SIZE TIRES.
6. THE MAXIMUM BUS HEIGHT IS BASED ON BODY WITH OPTIONS SUCH AS A/C UNITS, STROBE LIGHT, LARGEST SUSPENSION ETC.
7. "OVER BUMPER" DIMENSION DOES NOT TAKE INTO ACCOUNT OPTIONS THAT MAY ADD TO THE TOTAL LENGTH OF THE BUS, SUCH AS MIRRORS, LIGHTS AND CROSSING ARMS.

SIDE	ROW	SPACING	KNEE ROOM
LEFT		34.6	29.3
RIGHT		36.0	30.7
RIGHT		35.4	30.1

THOMAS BUILT BUSES, INC
HIGH POINT, NC

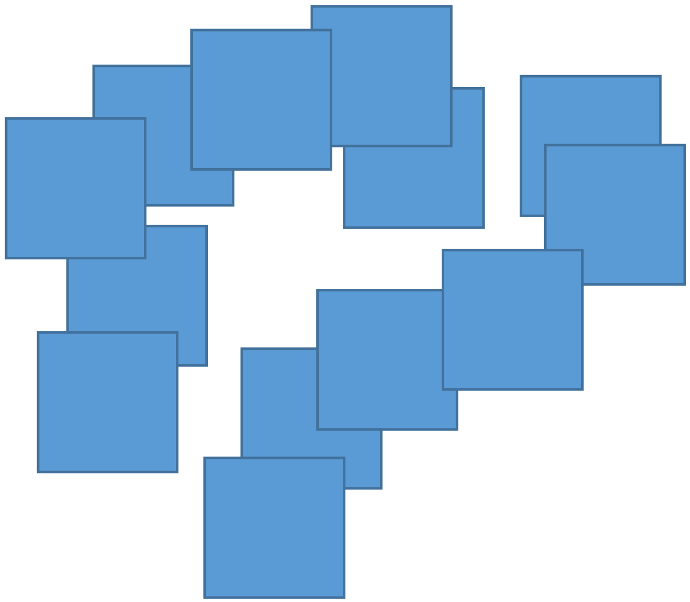
TITLE PLAN AND ELEVATION BODY 141YN

DRN: 06-07-18 BY: T. Dean
SCALE 3/8"=12"
SIZE S
DWG. NO. 742205

CAD DRAWING - DO NOT MAKE MANUAL CHANGES
CAD DRAWING - DO NOT MAKE MANUAL CHANGES
CAD DRAWING - DO NOT MAKE MANUAL CHANGES

Program Version: 18.7.26.1

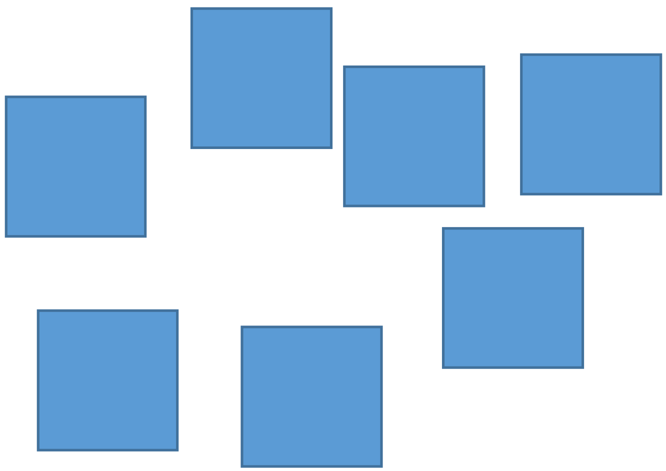
Special Case: S = { unit squares }



|S| = 13

PTASs known: [Chan'03, EJS'05]

Special Case: S = { unit squares }



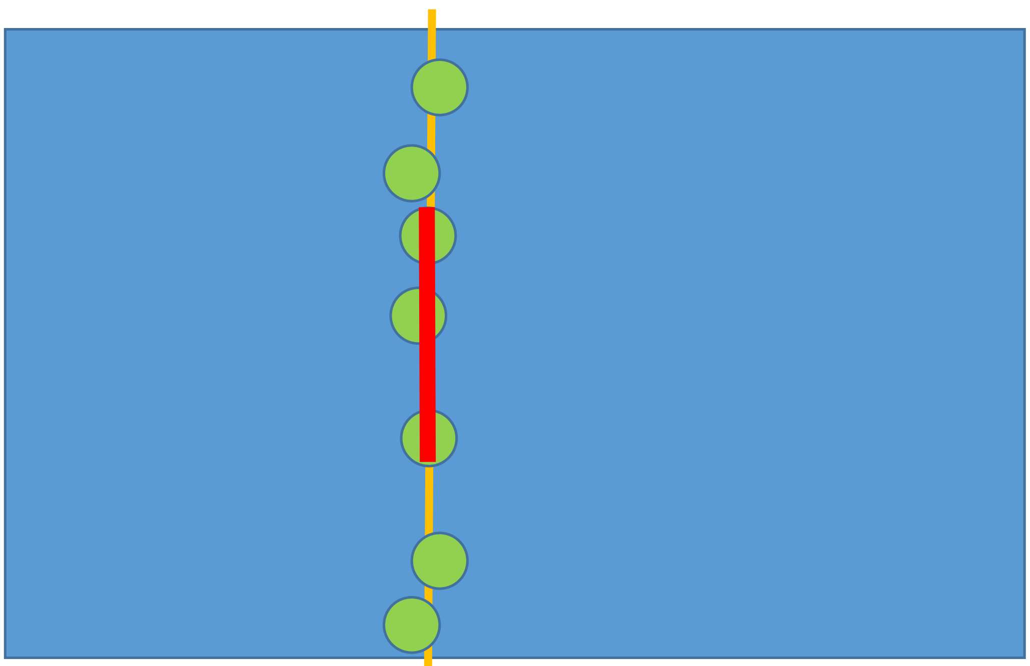
$|S^*| = 7$

PTASs known: [Chan'03, EJS'05]

Maximum Independent Set: Unit Disks

- Consider an OPT set of (disjoint) unit disks
- The boundaries of these disks give a network (1-net)
- m-guillotine structure Thm: can make network m-guillotine, adding (red) length O(1/m)*|OPT|

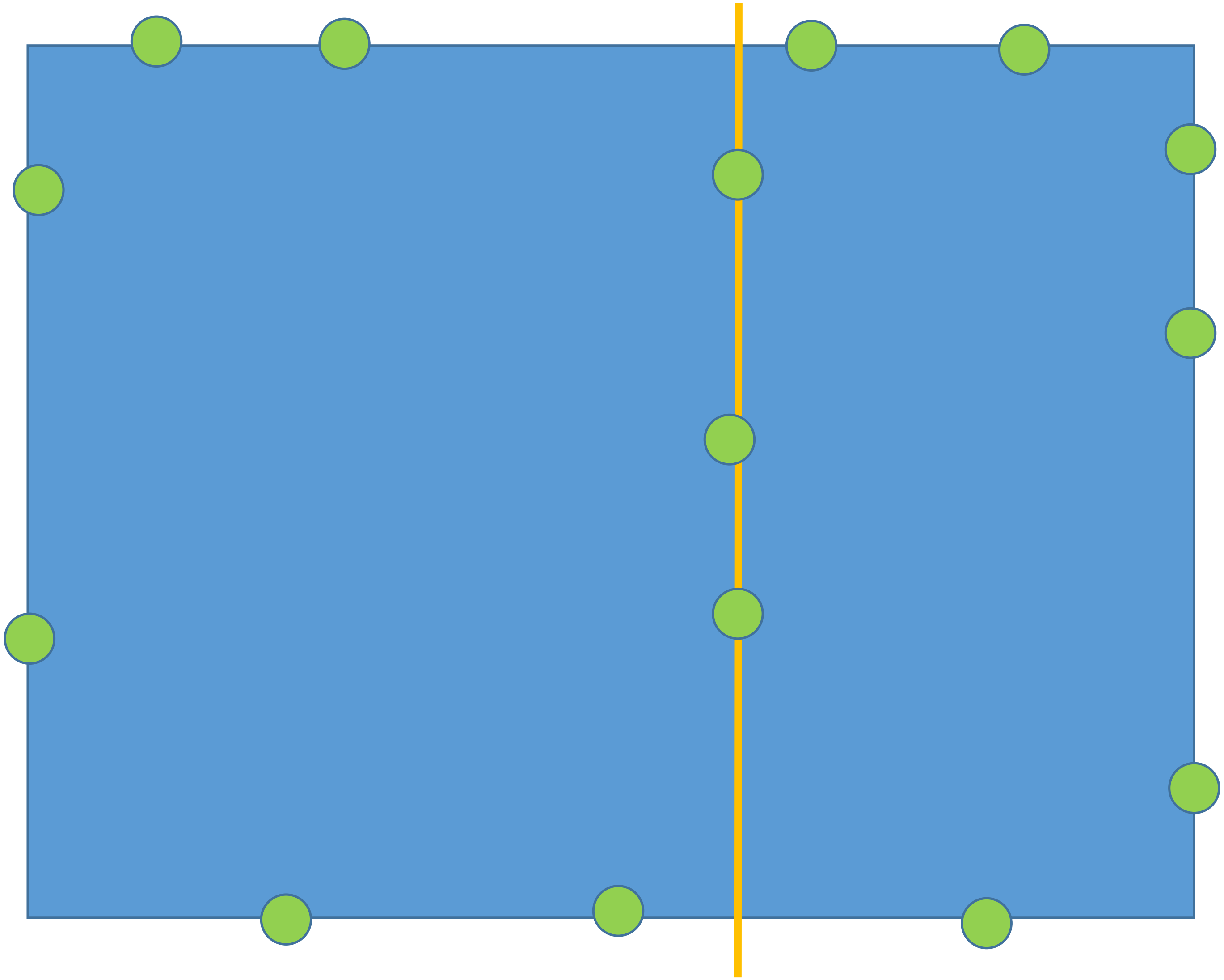
Give up the disks stabbed by the red m-spans: At most O(1/m) of disks are given up.

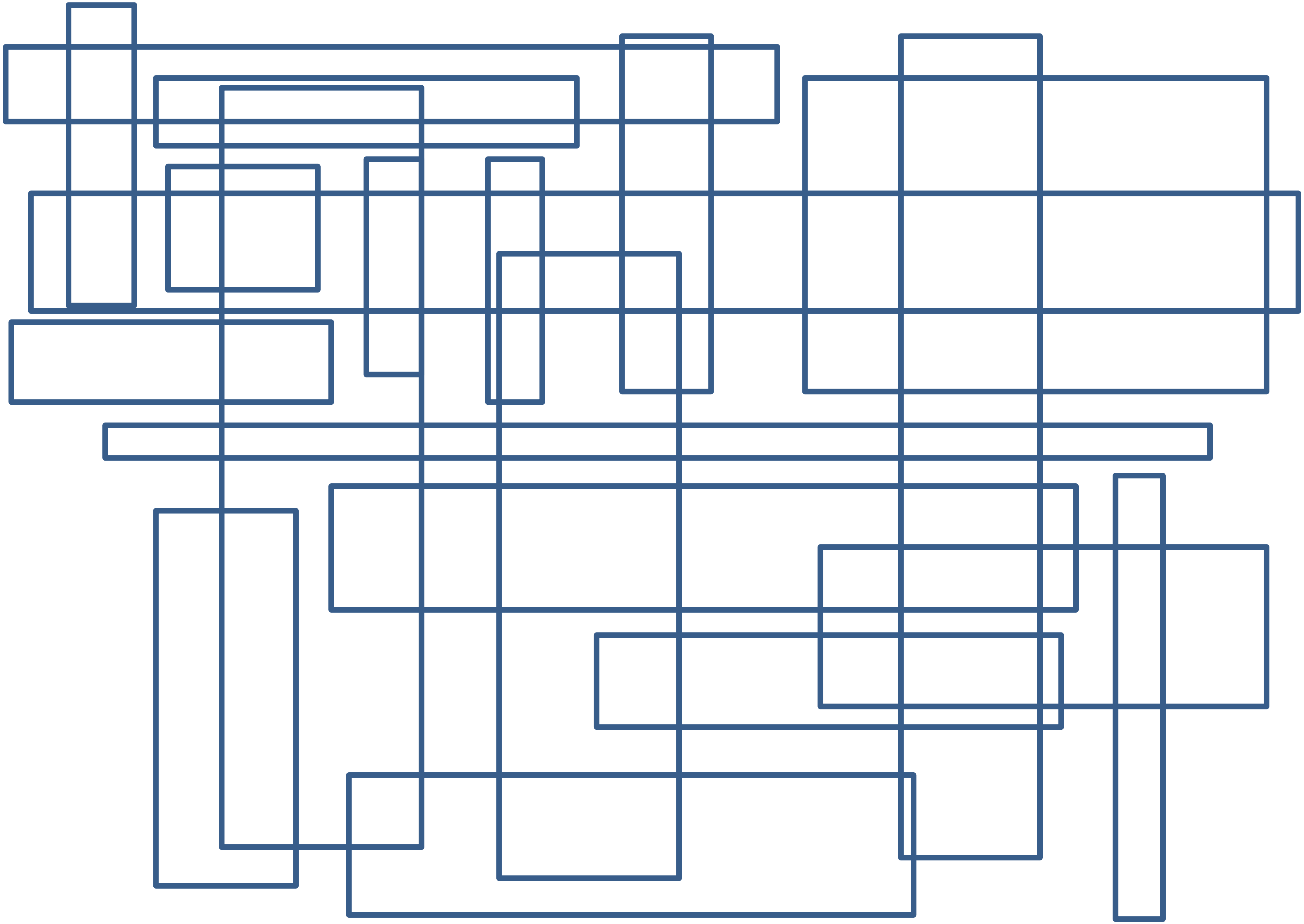


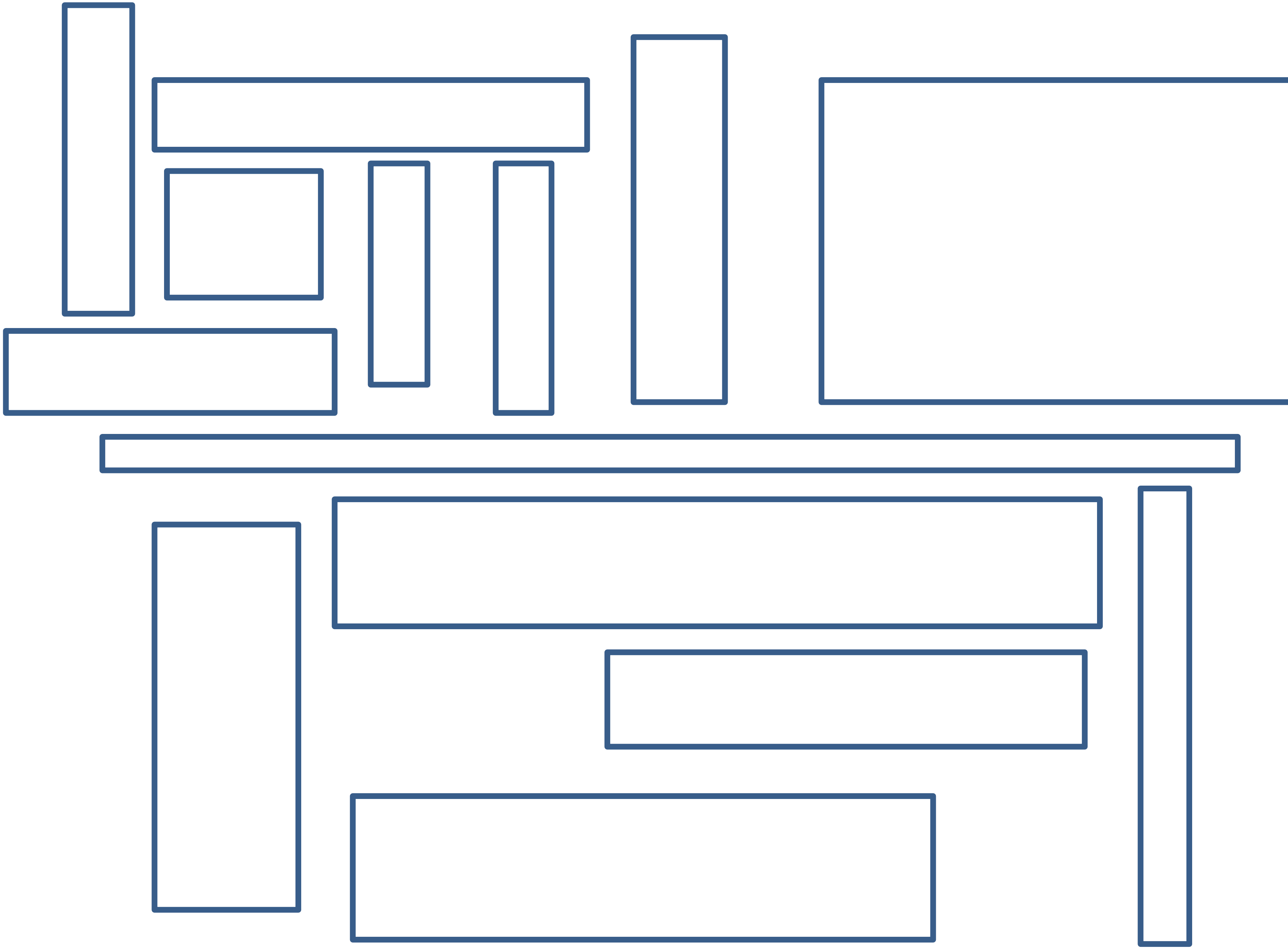
Subproblem

Only O(m) disks specified crossing boundary of rectangle R.
Optimize over cuts, choices of O(m) disks crossing cut

Time $n^{O(m)}$

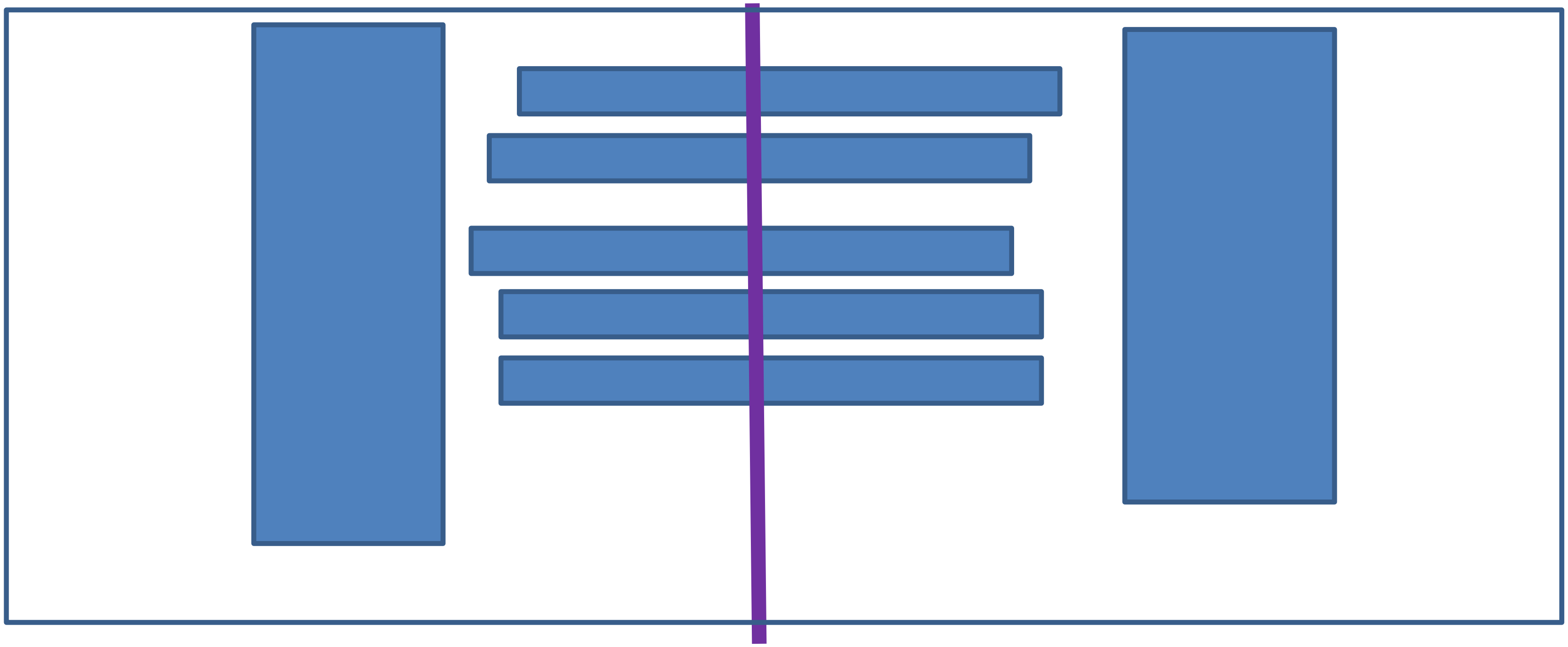






Key Idea

- Rectangles that are cut should be "paid for"
- But, we cannot assume there is a newly exposed rectangle/vertex when we cross a rectangle:



Approximations

- Disks, fat regions: PTAS   (1-ε)-approx, for any ε>0, in polytime

  (1- ε)-approx in $n^{O(1/\varepsilon^{d-1})}$ [Chan]

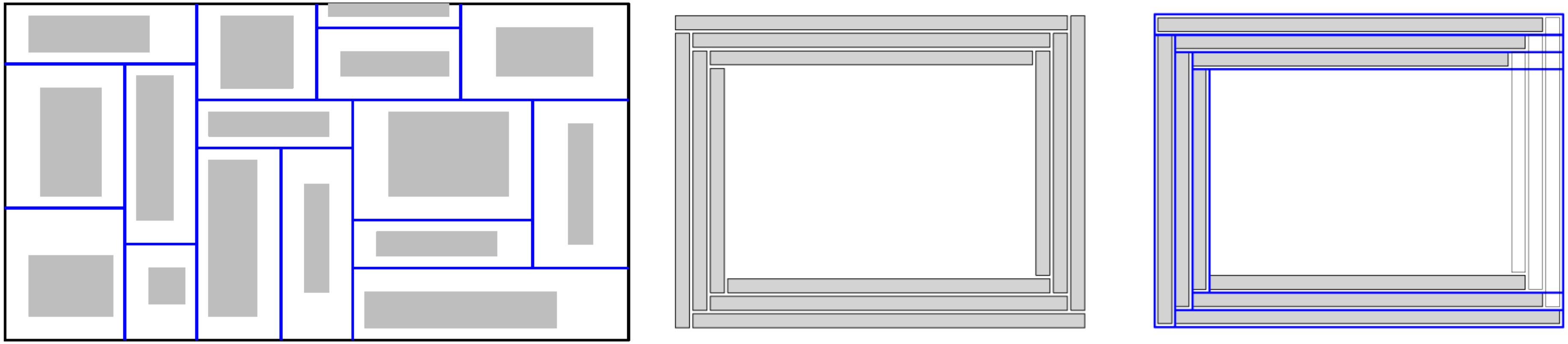
  Also: PTAS for pseudodisks [Chan, Har-Peled]

- Rectangles: MISR   Rectangles are neither fat nor pseudodisks!
  - QPTAS
    - $n^{poly((\log n)/\varepsilon)}$ [Adamaszek, Har-Peled, and Wiese]
    - $n^{O(((\log\log n)/\varepsilon)^4)}$ [Chuzhoy and Ene]
  - PTAS for "long" rectangles [Adamaszek, Har-Peled, and Wiese]
  - Polytime: O(loglog n)-approx [Chalermsook, Chuzhoy]
  - Parameterized Approximation Scheme: [Grandoni,Kratsch,Wiese,2019]

    For any k, ε, in time $f(k, \varepsilon)n^{g(\varepsilon)}$ either gives indep subset of ≥k/(1+ ε), or declares OPT<k
  - **Here: O(1)-Approx in polytime**

MISR: One Approach

- Show that any set of disjoint rectangles (e.g., the rectangles of OPT) has a constant fraction subset that has a perfect BSP

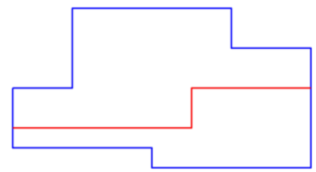


[Pach-Tardos Conjecture]

Conjecture 1. *For any set of n interior-disjoint axis-aligned rectangles in the plane, there exists a subset of size $\Omega(n)$ that has a perfect orthogonal BSP.*

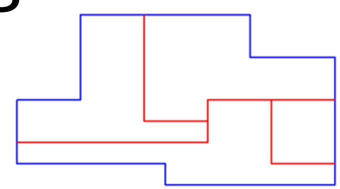
Main Ideas

- Use more general cuts to get O(1) complexity pieces – “CCRs”



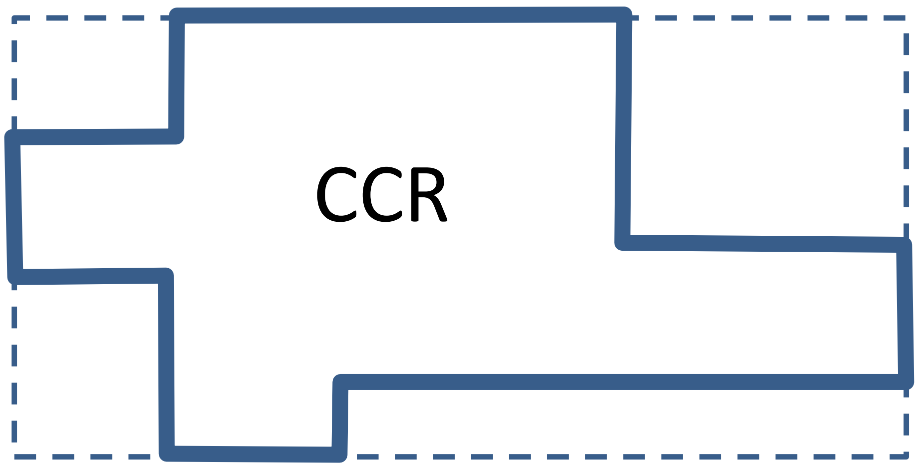
- Use K-ary cutting instead of just binary

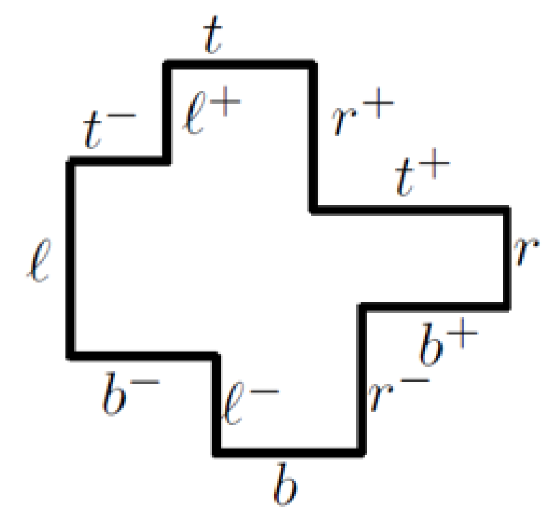
K≤5

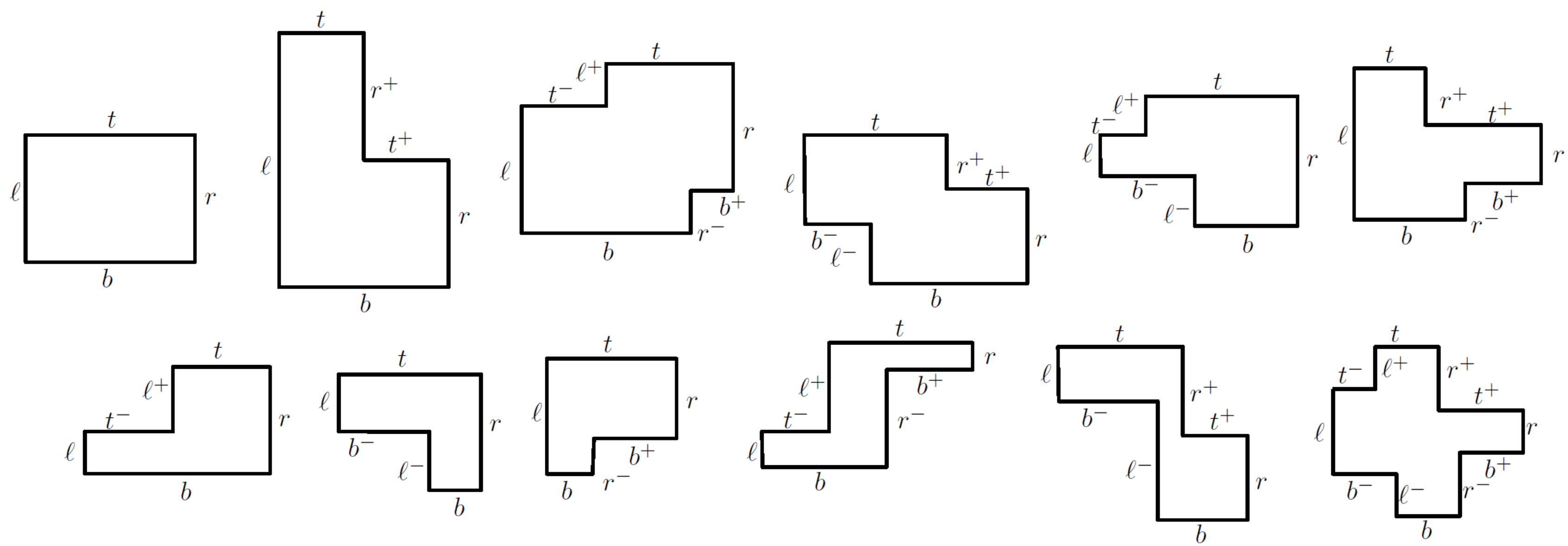


- Charging scheme to prove a structural theorem: Can afford to discard a constant fraction of input rectangles, to enable a “nearly perfect CCR-partition”
- DP to optimize

Corner Clipped Rectangles

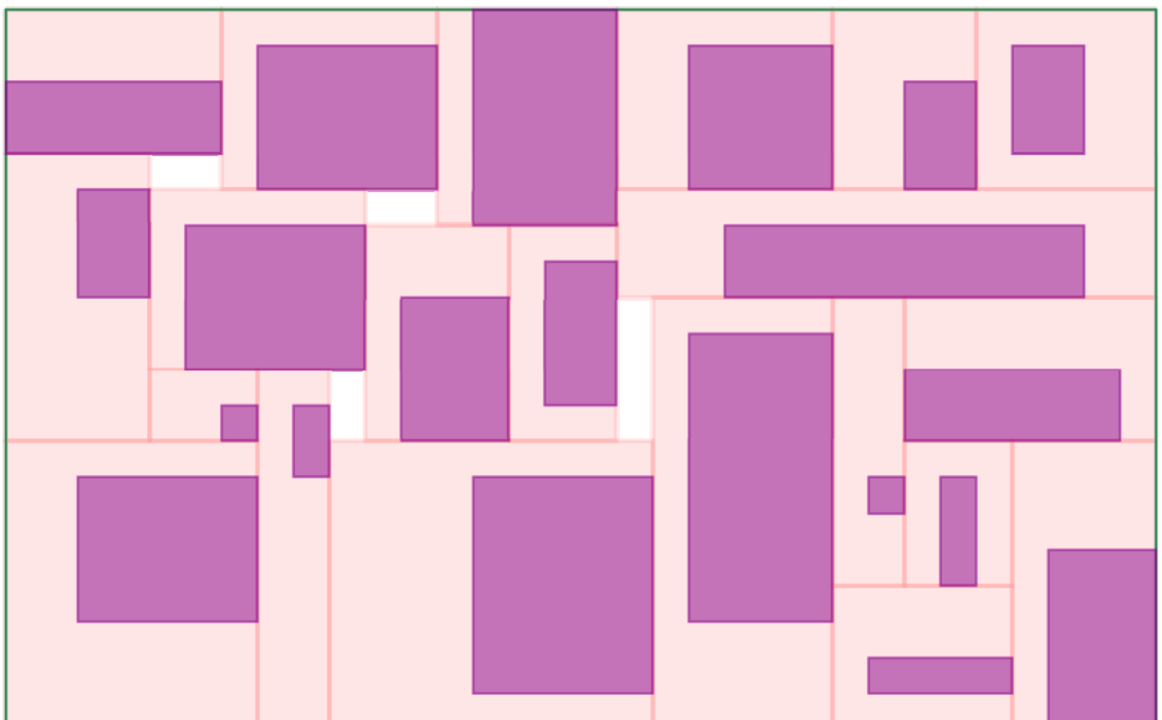






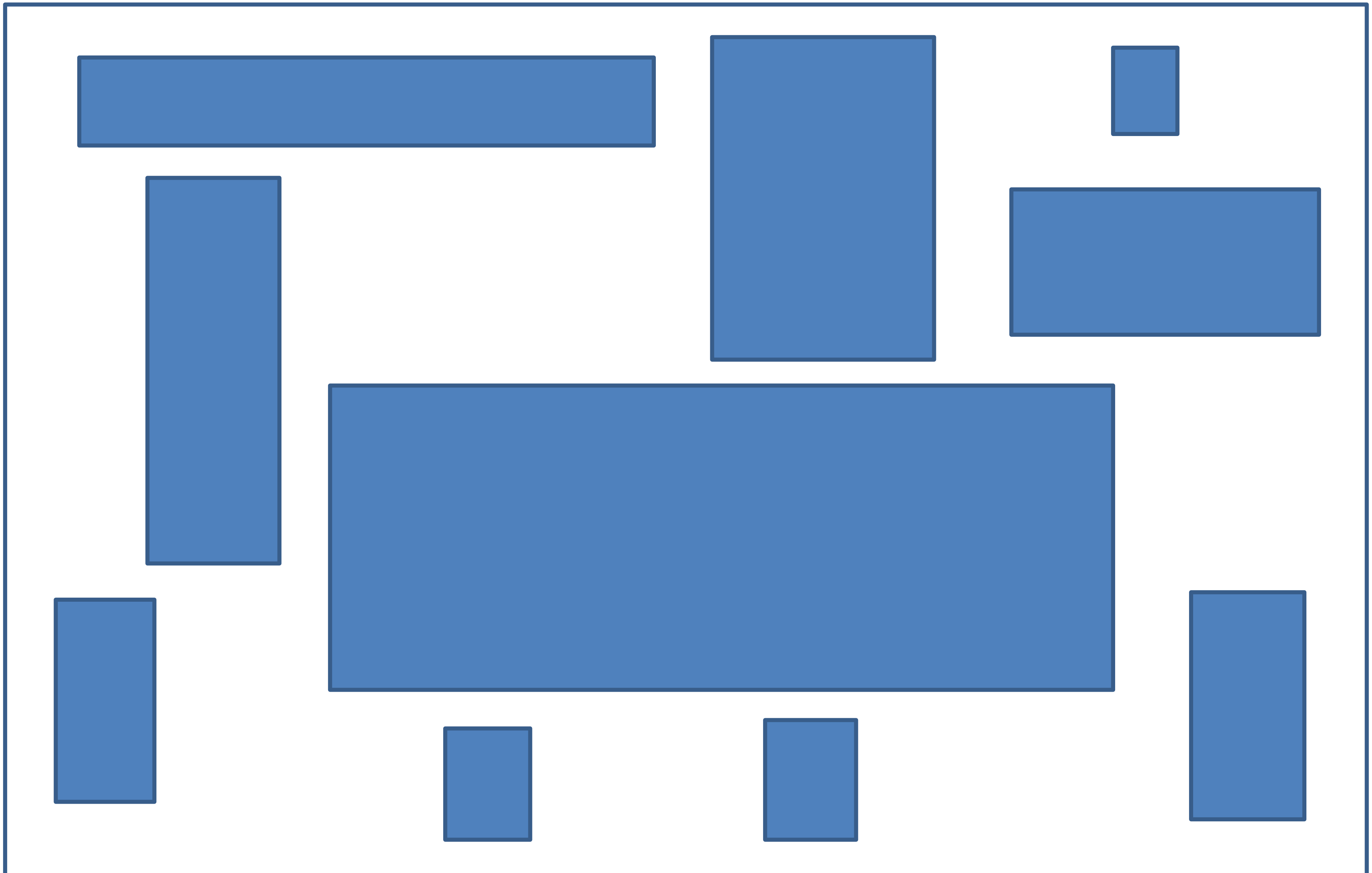
Maximal Rectangles

- Transform any set I of k disjoint rectangles into a set I' of *maximal* disjoint rectangles

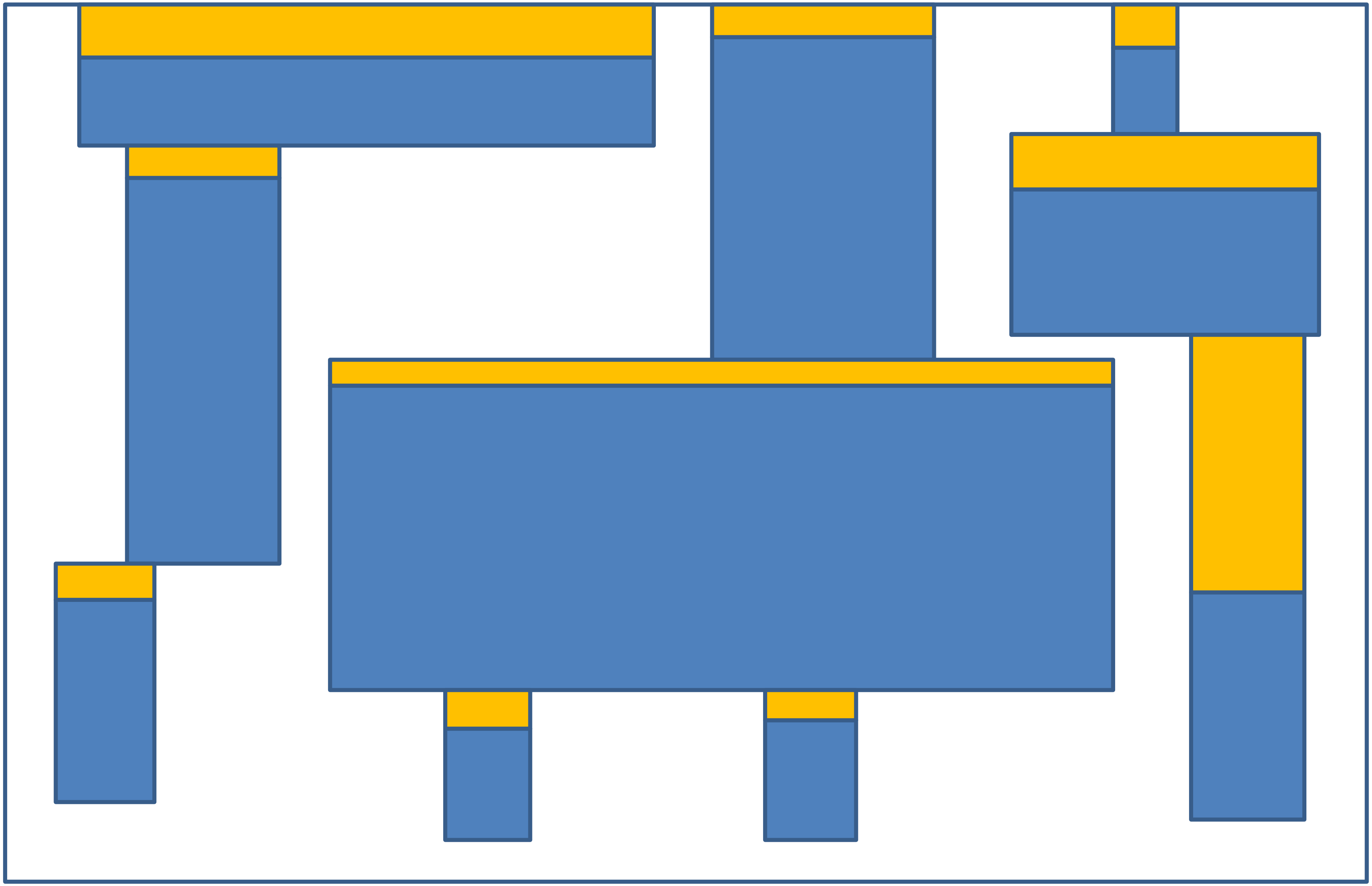


Will show that I' has a constant-fraction subset for which there is a "nearly perfect CCR-partition" wrt the subset

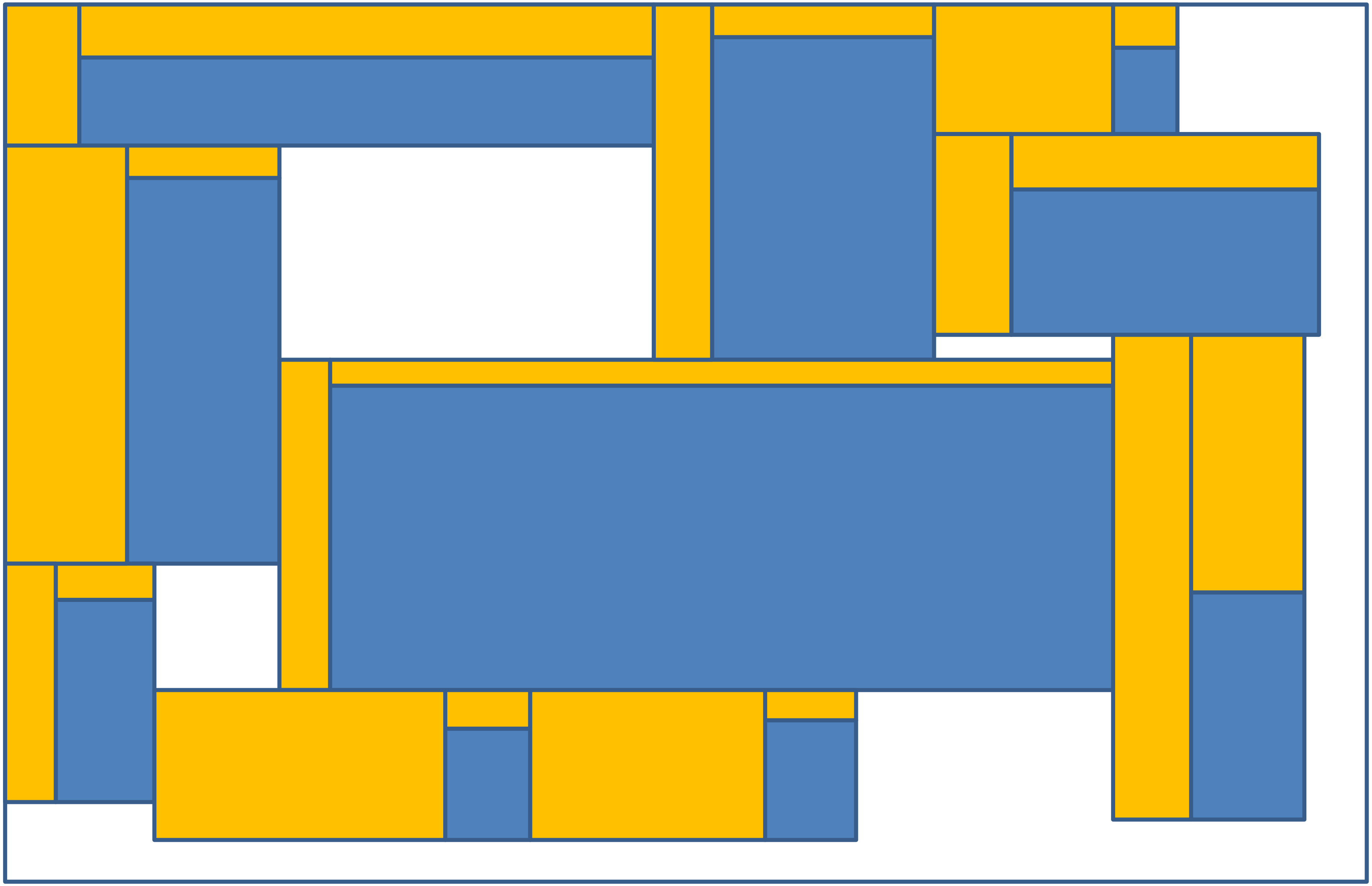
Rectangle Maximal Expansions



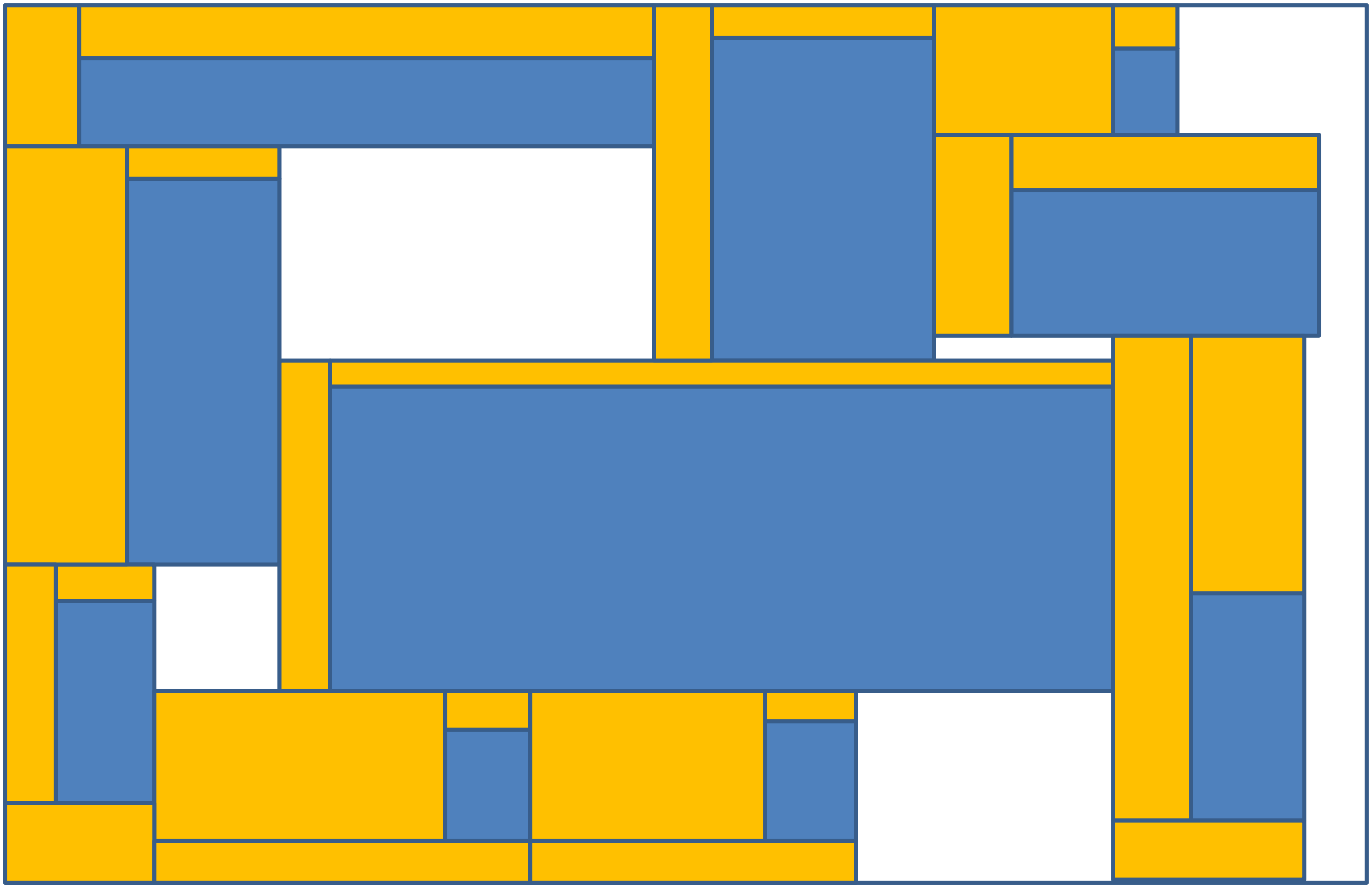
Rectangle Maximal Expansions



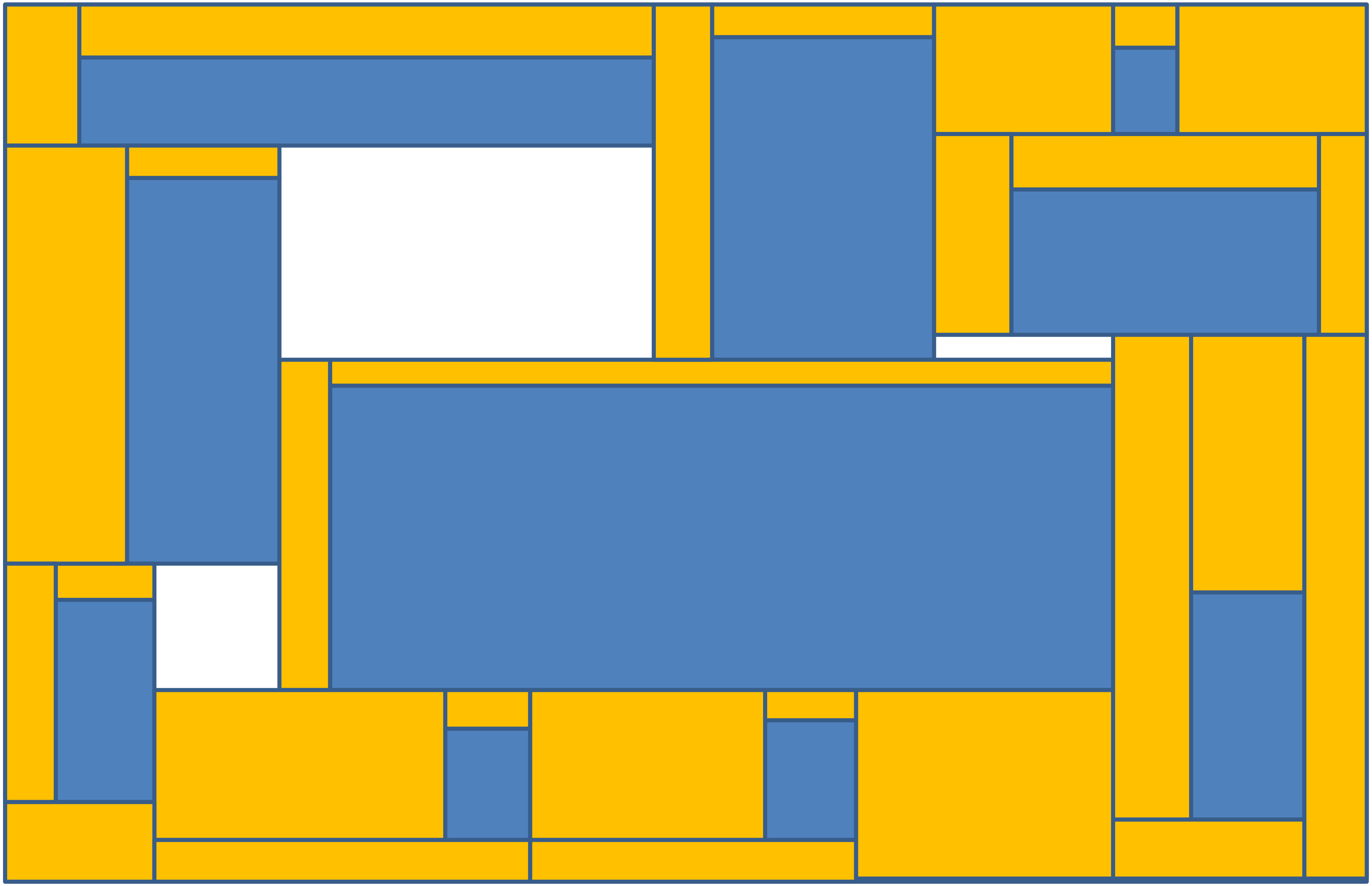
Rectangle Maximal Expansions



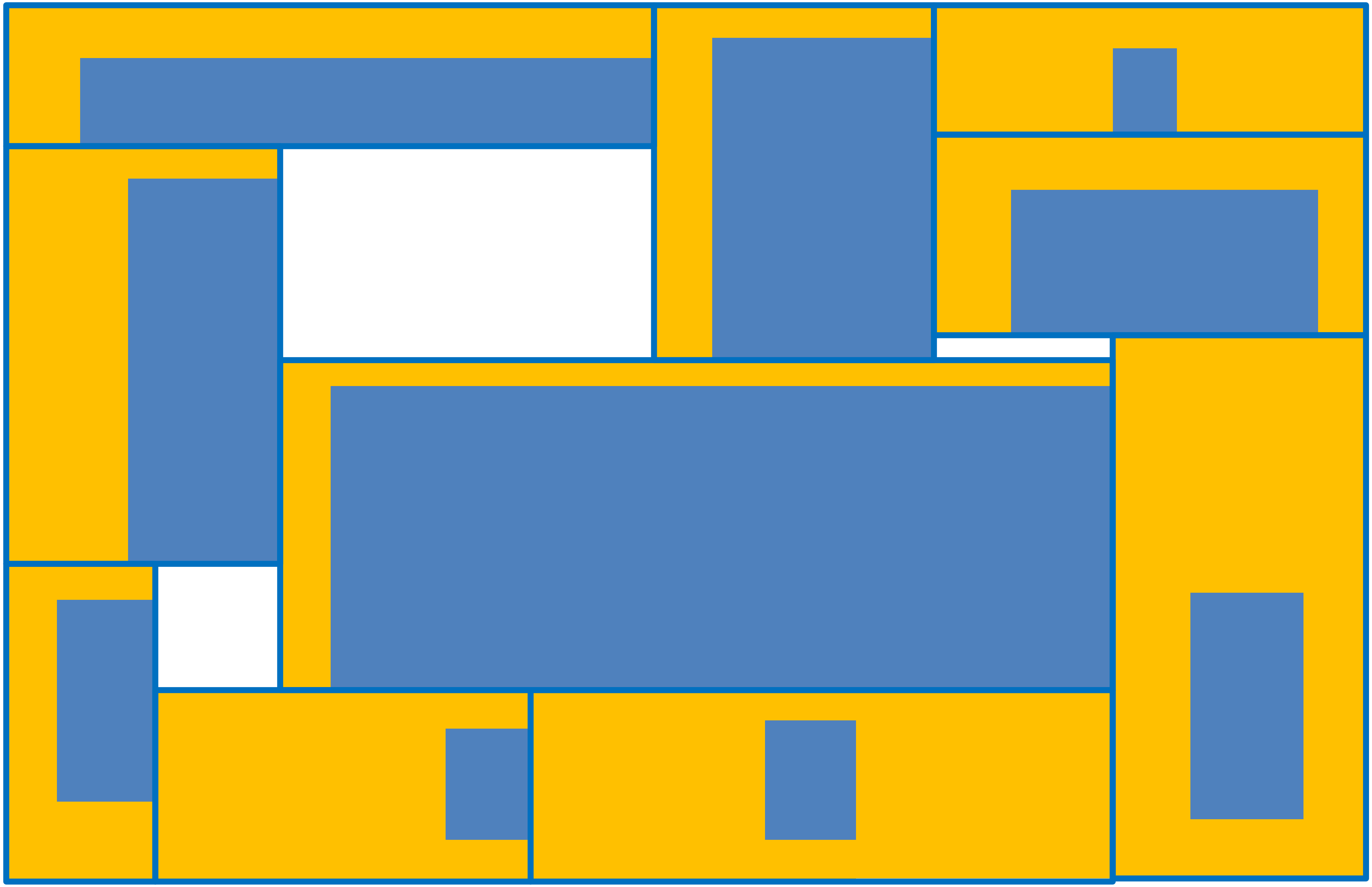
Rectangle Maximal Expansions



Rectangle Maximal Expansions

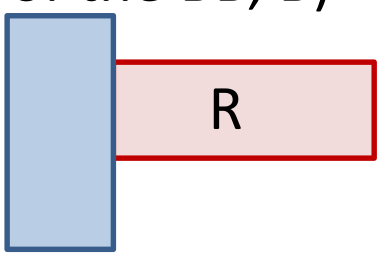


Rectangle Maximal Expansions



Nesting Among Maximal Rectangles

Def: A rectangle R is *nesting* to its left/right/top/bottom if its corresponding side is contained in the interior of an abutting rectangle's side (or the side of the BB, B)

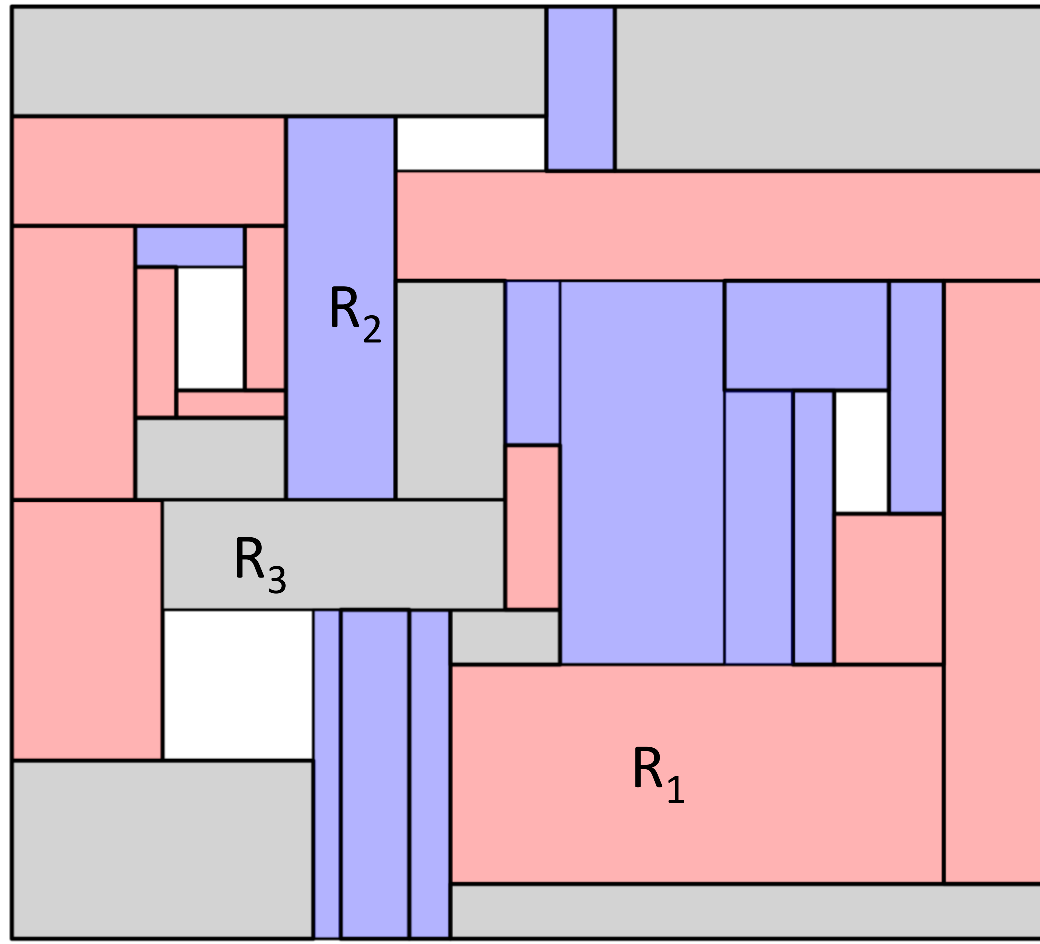


Example:

R_1 is horiz nested (red)

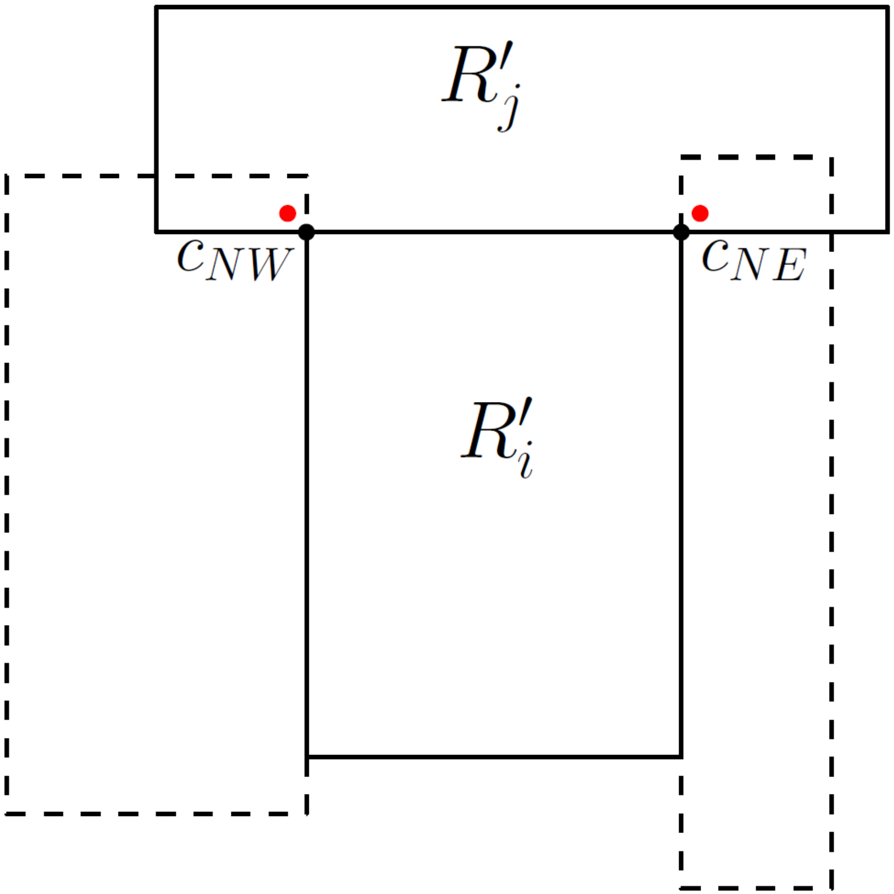
R_2 is vert nested (blue)

R_3 is not nested in any direction

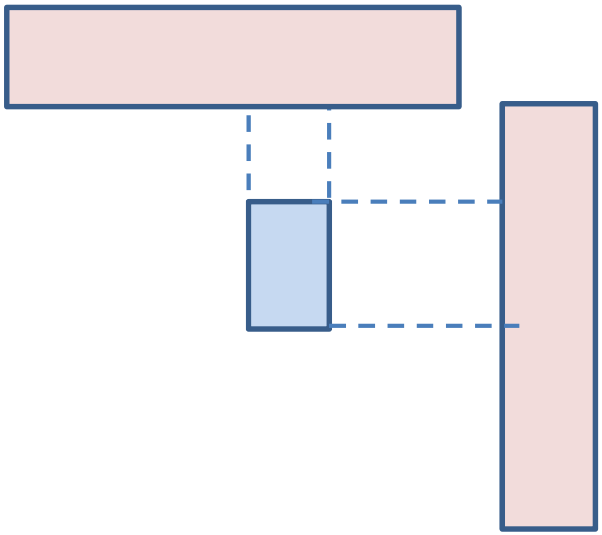


Why Maximality Is Useful

Observation 1. *For a set I' of independent rectangles that are maximal within $BB(\mathcal{R})$, a rectangle $R'_i \in I'$ cannot be nested both vertically and horizontally.*

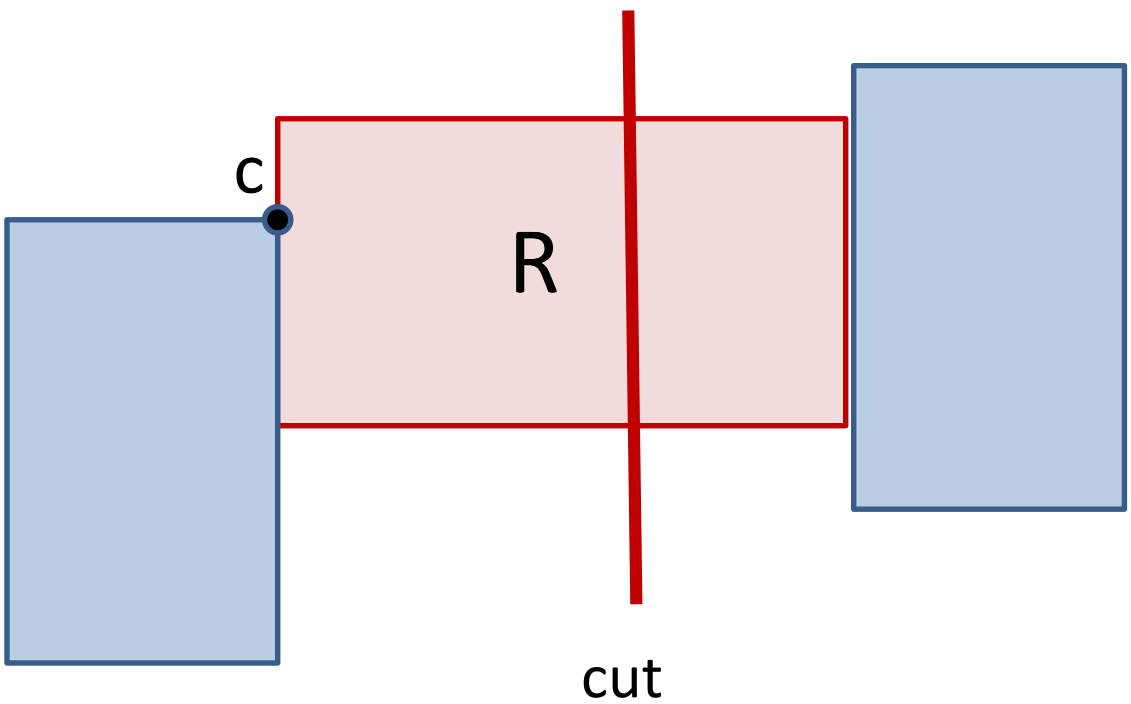


Note that the claim is not true without maximality:



Why Nesting Concept Is Useful

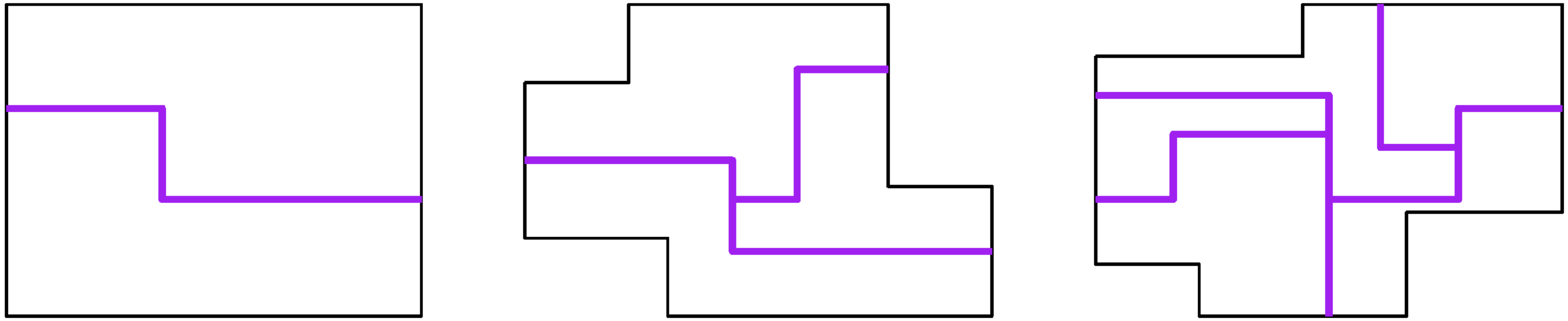
If R is *not* nested on at least one side, there is hope to be able to “charge” R to a corner, c, when a cut segment crosses R



CCR-Partitions

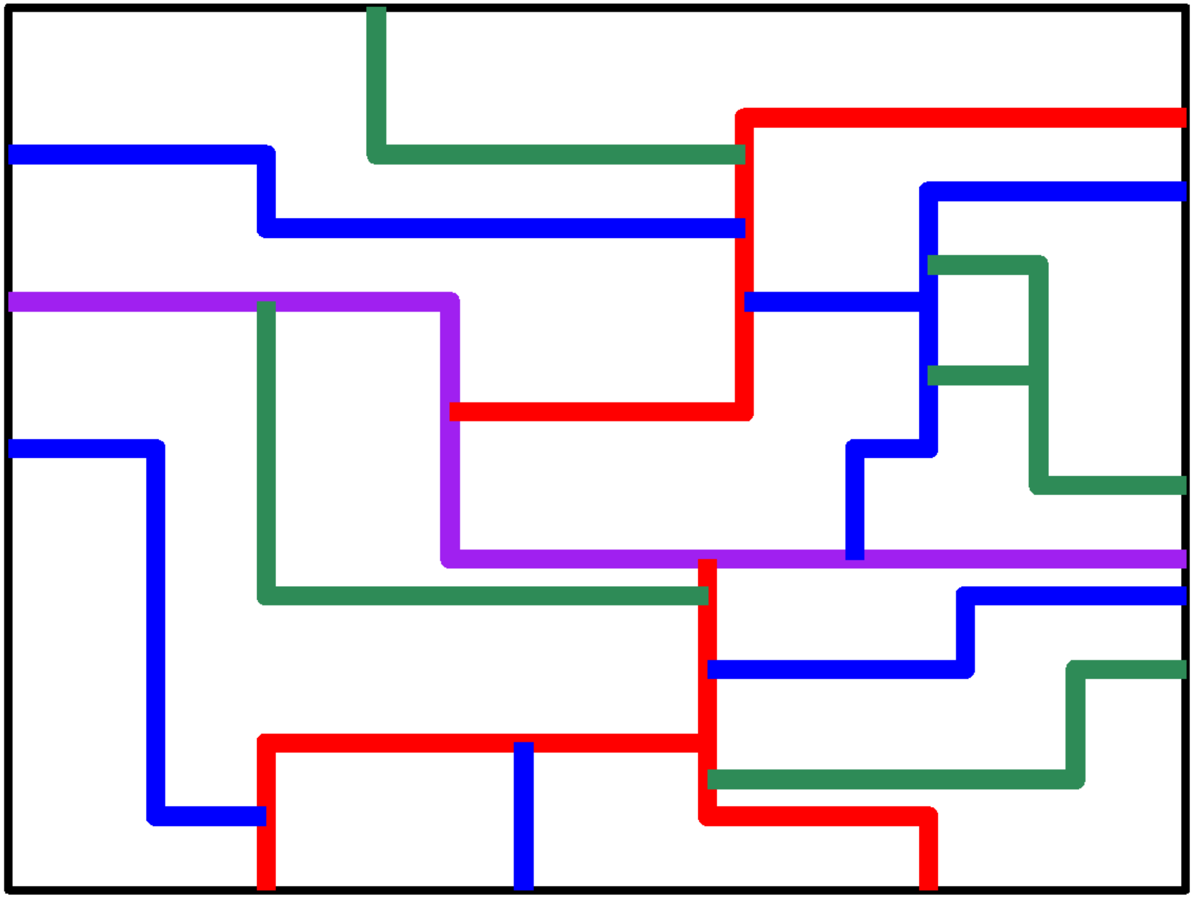
- Recursive partitioning of the BB, B, of input
- Each face Q is a CCR
- A *cut*, consisting of O(1) hor/vert segments partitions Q into at most 5 subfaces (CCRs)
- A CCR-partition is *perfect* wrt input rectangles if no rectangle is penetrated by a cut segment, each leaf face has exactly 1 input rectangle
- *Nearly perfect* CCR-partition: each cut segment penetrates at most 2 input rectangles, each leaf face has ≤1 input rectangle

K-ary Cuts

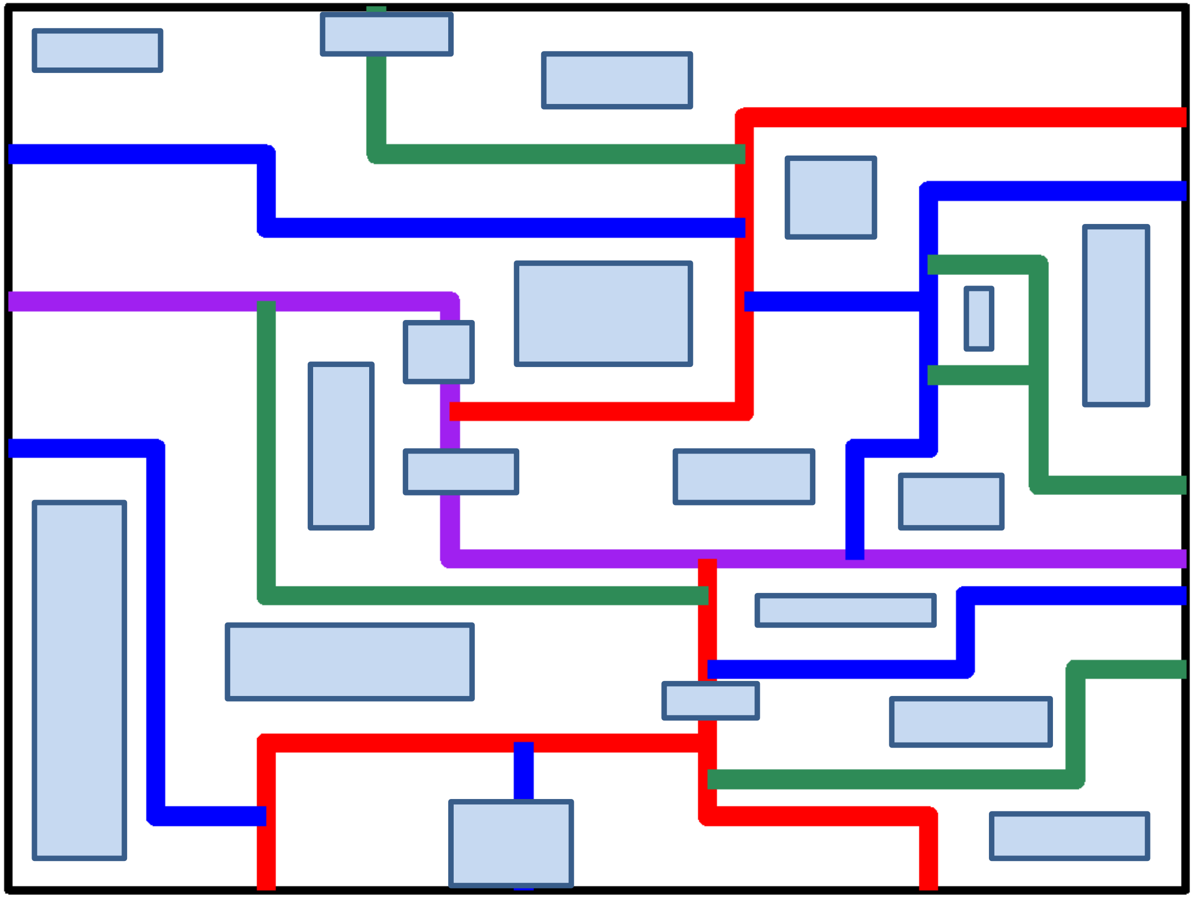


K=2 K=3 K=5

CCR Partition

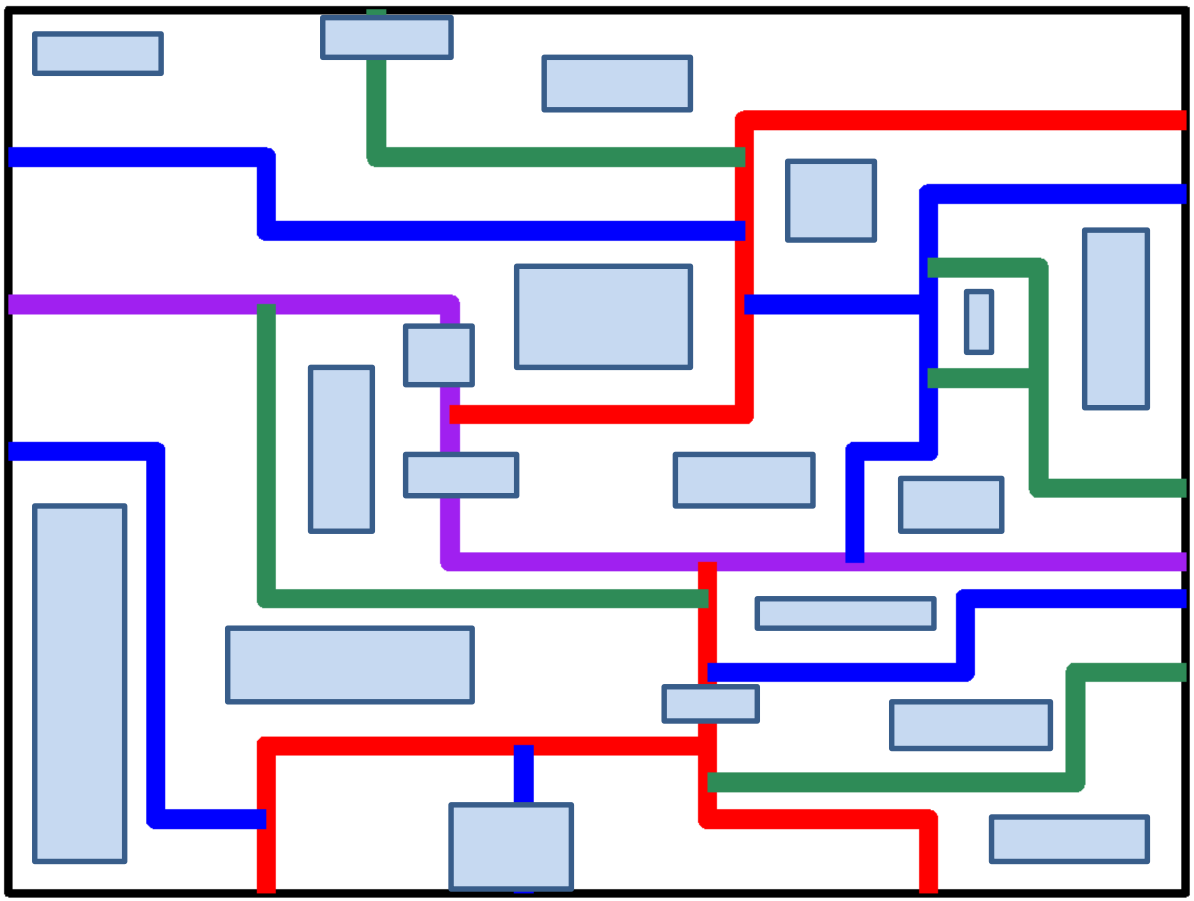


Nearly Perfect CCR Partition



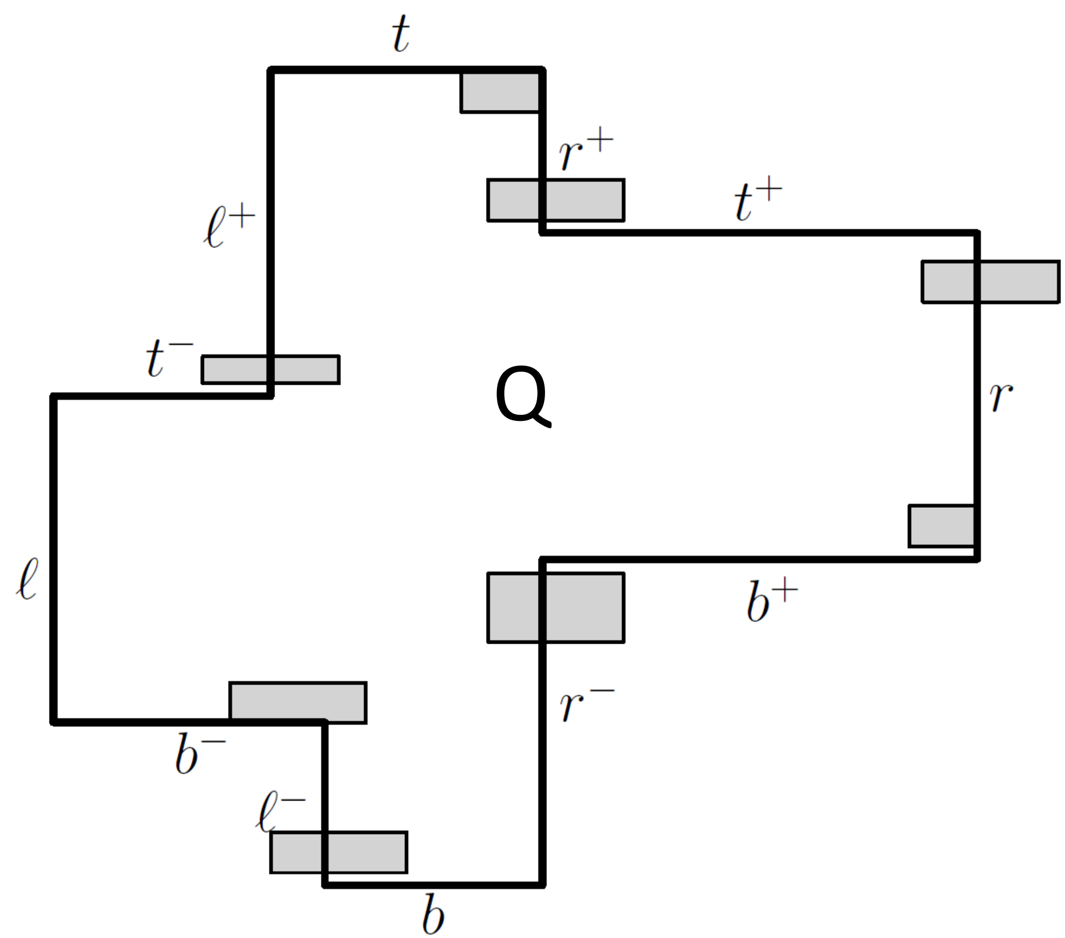
The Structure Theorem

Theorem 3.1. *For any set $I = \{R_1, \ldots, R_k\}$ of k interior disjoint (axis-aligned) rectangles in the plane within a bounding box B, there exists a K-ary CCR-partition of the bounding box B, with $K \leq 5$, recursively cutting B into corner-clipped rectangles (CCRs), such that the CCR-partition is nearly perfect with respect to a subset of I of size $\Omega(k)$ (at least $k/10$).* More carefully: at least k/3



The Algorithm: DP Subproblem

Subproblem S=(Q,I_S), where I_S is a set of “special” (specified) rectangles, at most 2 per vertical side of the CCR face Q.



Dynamic Program

- Optimize over K-ary cuts (K≤5) for a CCR subproblem, S, to compute f(S), the max cardinality of an indep subset of input rectangles for which there is a nearly perfect CCR-partition

$$f(\mathcal{S}) = \begin{cases} 0 & \text{if } \mathcal{R}(\mathcal{S}) = \emptyset, \\ \max_{\chi \in \gamma(\mathcal{S}), I_\chi} (f(\mathcal{S}_1) + \cdots + f(\mathcal{S}_K) + |I_\chi|) & \text{otherwise,} \end{cases}$$

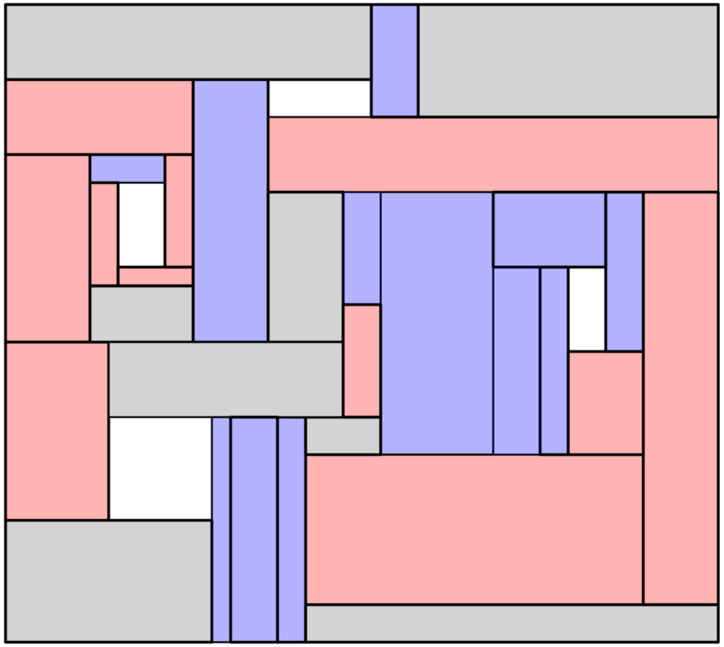
Here, I_χ is the set of rectangles (at most 2 per vertical segment of χ) that are penetrated by vertical cut segments and become special rectangles specified for the new subproblems, and $\gamma(\mathcal{S})$ is the set of all eligible K-ary CCR-cuts

Theorem 4.1. *There is a polynomial-time* (1/10)*-approximation algorithm for maximum independent set for a set of axis-aligned rectangles in the plane.*

Crudely counted: time is $O(n^{34})$

Proof of the Structure Theorem

- Let $\mathrm{I} = \{R_1, R_2, \ldots, R_k\}$ be an OPT set
- Let I' = maximal expansions of I
- $\mathrm{I}' = \mathrm{I}_h \cup \mathrm{I}_v \cup \mathrm{I}_0$ (partition)
  - I_h = red (nested horiz)
  - I_v = blue (nested vert)
  - I_0 = gray (not nested)
- WLOG: $|\mathrm{I}_h| \leq k/2$
- # Non-red rectangles $\geq k/2$



Proof of the Structure Theorem

- Goal: Keep a subset of $\Omega(k)$ rectangles of I', for which there is a nearly perfect CCR-partition

Claim 5.1. *If a CCR-partition is nearly perfect with respect to a set $A' \subset I'$ of maximal rectangles, then it is nearly perfect with respect to the corresponding set $A \subseteq I$ of input rectangles.*

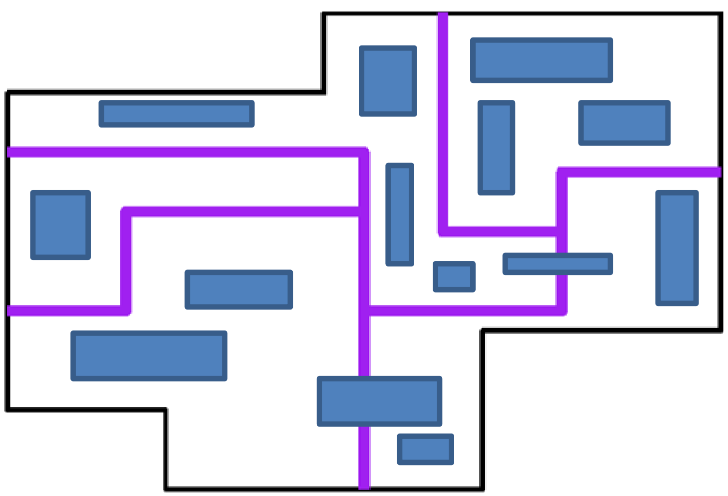
- Process of selecting a subset of I' (k rectangles):
  - Initially, all rectangles of I' are *active*
  - During process, some rectangles are *discarded*

    Removed from active status
  - Charging scheme argument: $\leq (9/10)k$ discarded

More carefully: At most (2/3)k discarded

Process of Cutting; CCR-Partition

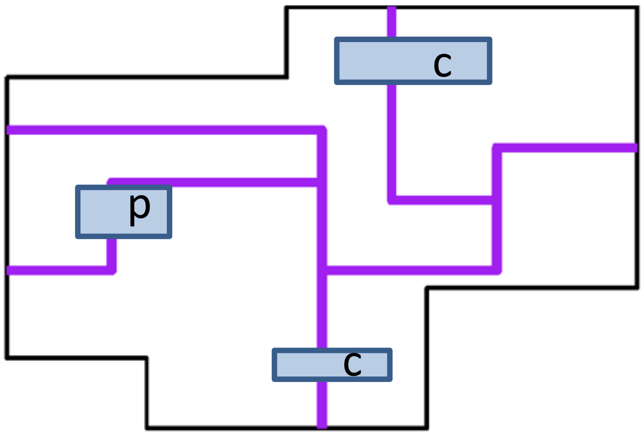
- Starting with BB(R), we recursively partition faces of a CCR-partitioning during the process
- Face Q: If Q has >1 rectangle within it, we partition it with a cut χ into at most 5 subfaces

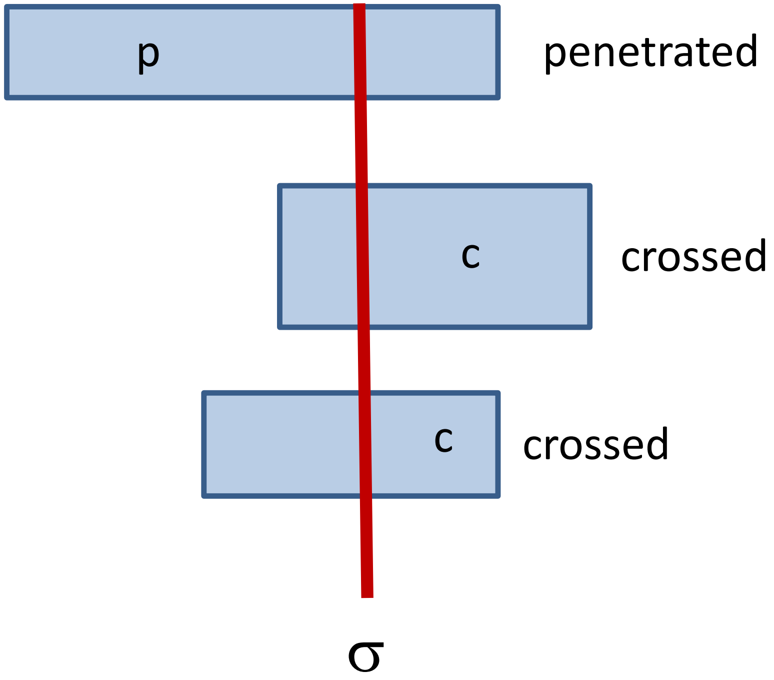


Properties of a Cut

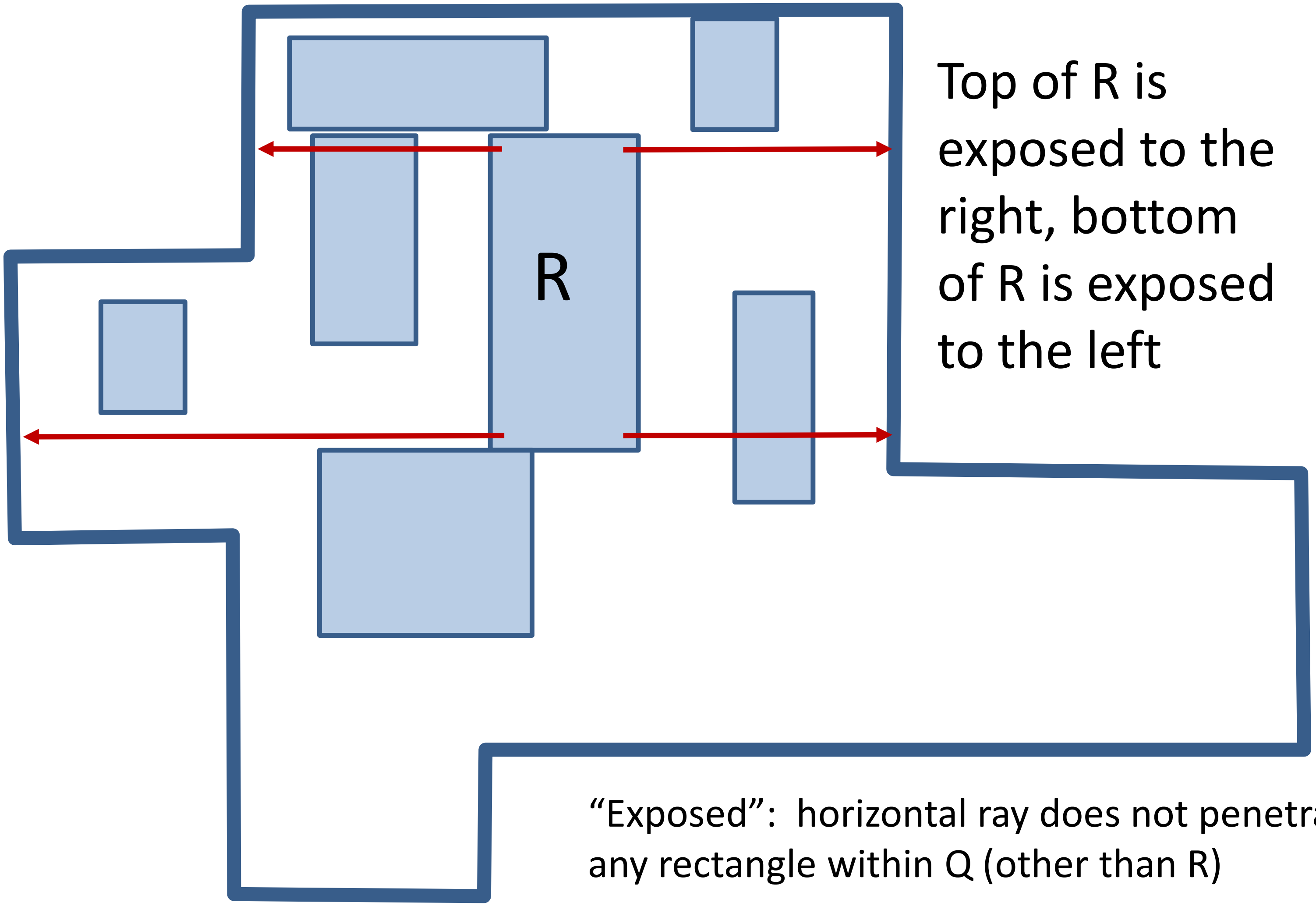
- cut χ : consists of horizontal/vertical portions
  - Horizontal does not penetrate any rectangle
  - Vertical portion σ:

(will always be subsets of "fences")





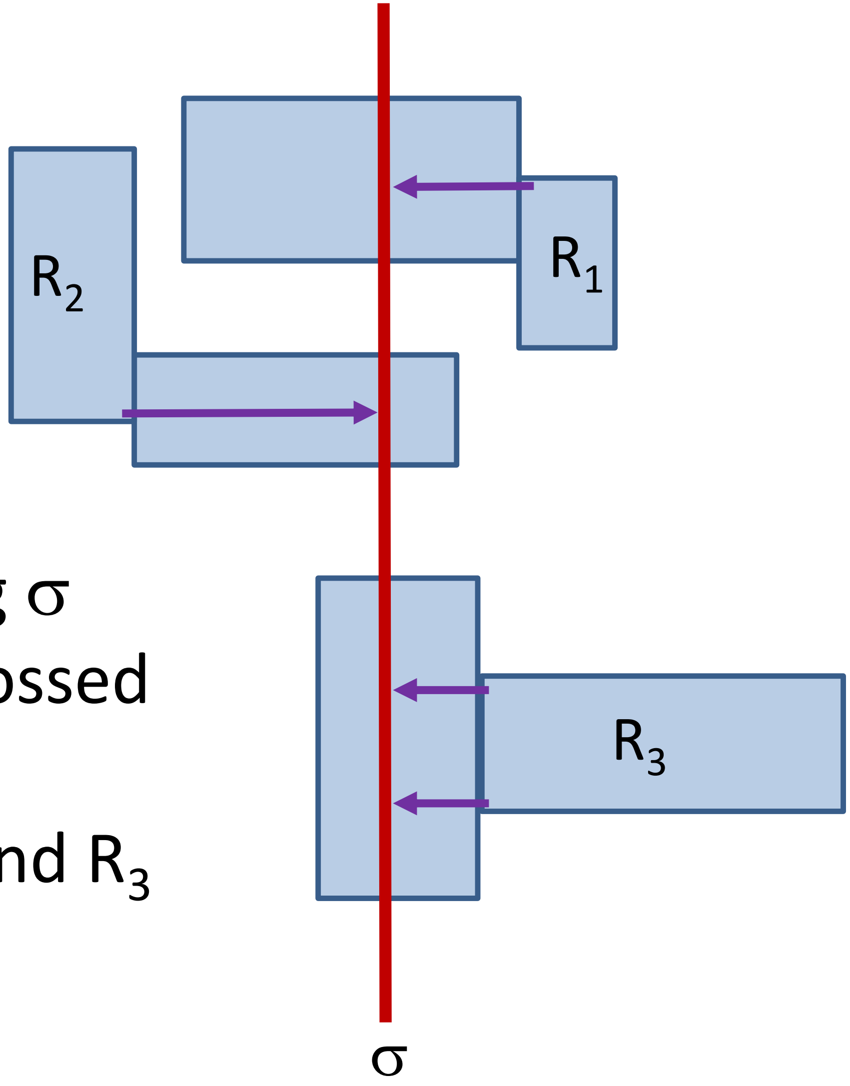
Notion of Being "Exposed"



Top of R is exposed to the right, bottom of R is exposed to the left

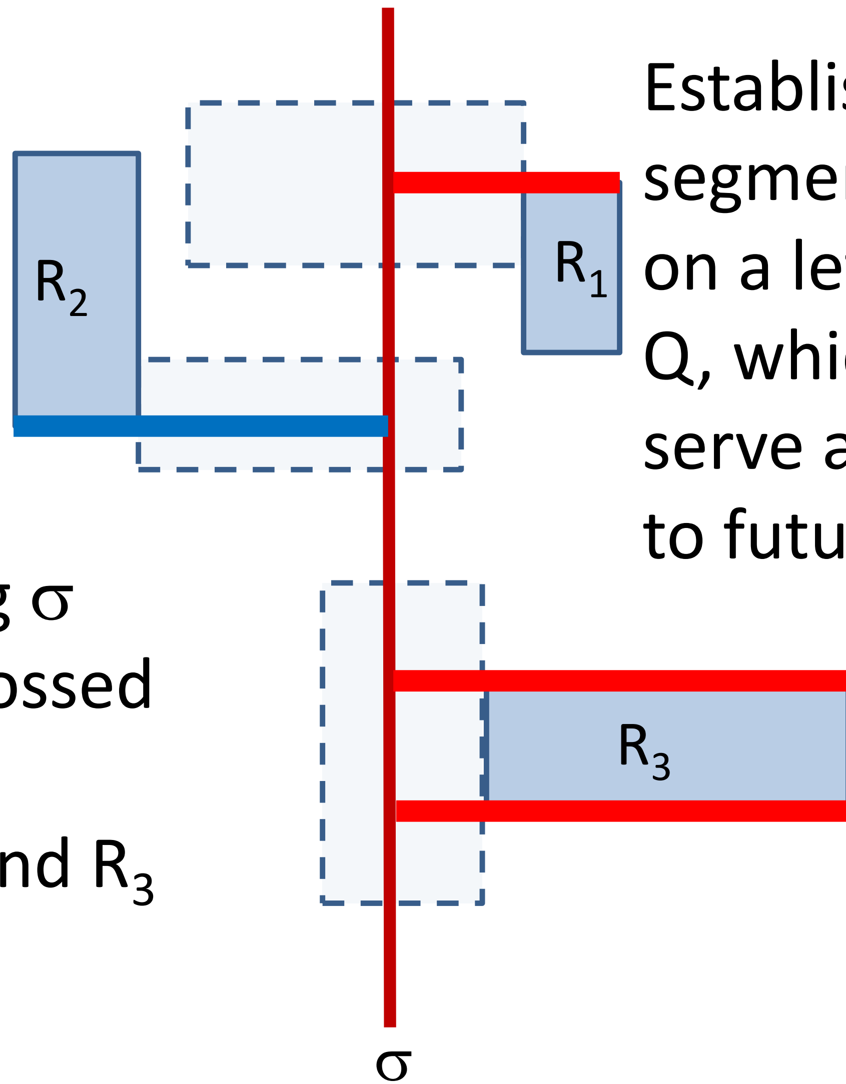
"Exposed": horizontal ray does not penetrate any rectangle within Q (other than R)

A Cut Exposes Some Rectangles



When we cut along σ and remove the crossed rectangles, the rectangles R_1, R_2, and R_3 become exposed

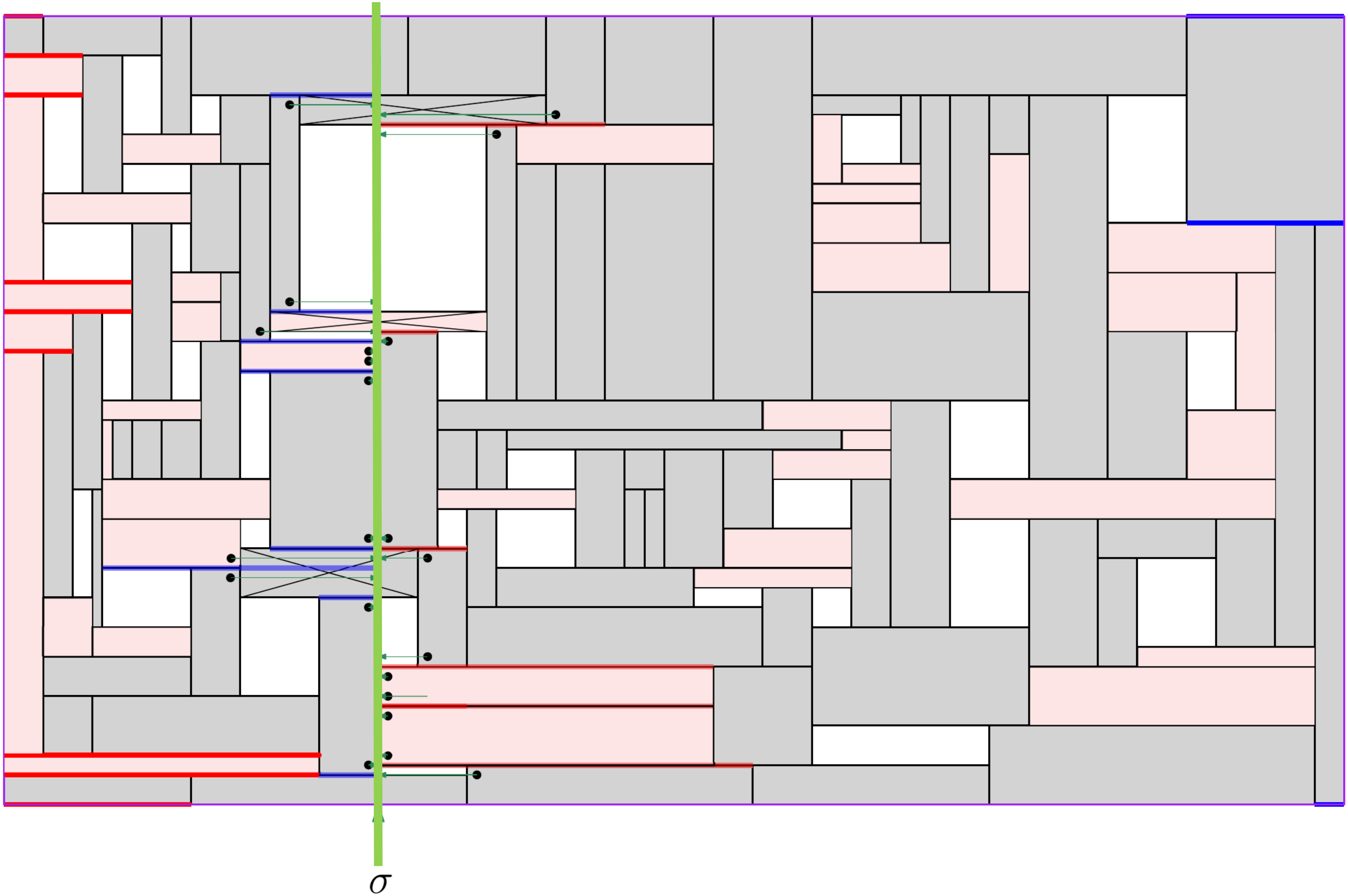
A Cut Exposes Some Rectangles



Establish *fences*, horiz segments, anchored on a left/right side of Q, which then will serve as an “obstacle” to future cuts

When we cut along σ and remove the crossed rectangles, the rectangles R_1, R_2, and R_3 become exposed

Example: Cut σ Exposes Tops/Bottoms; Fences



Fence Invariant

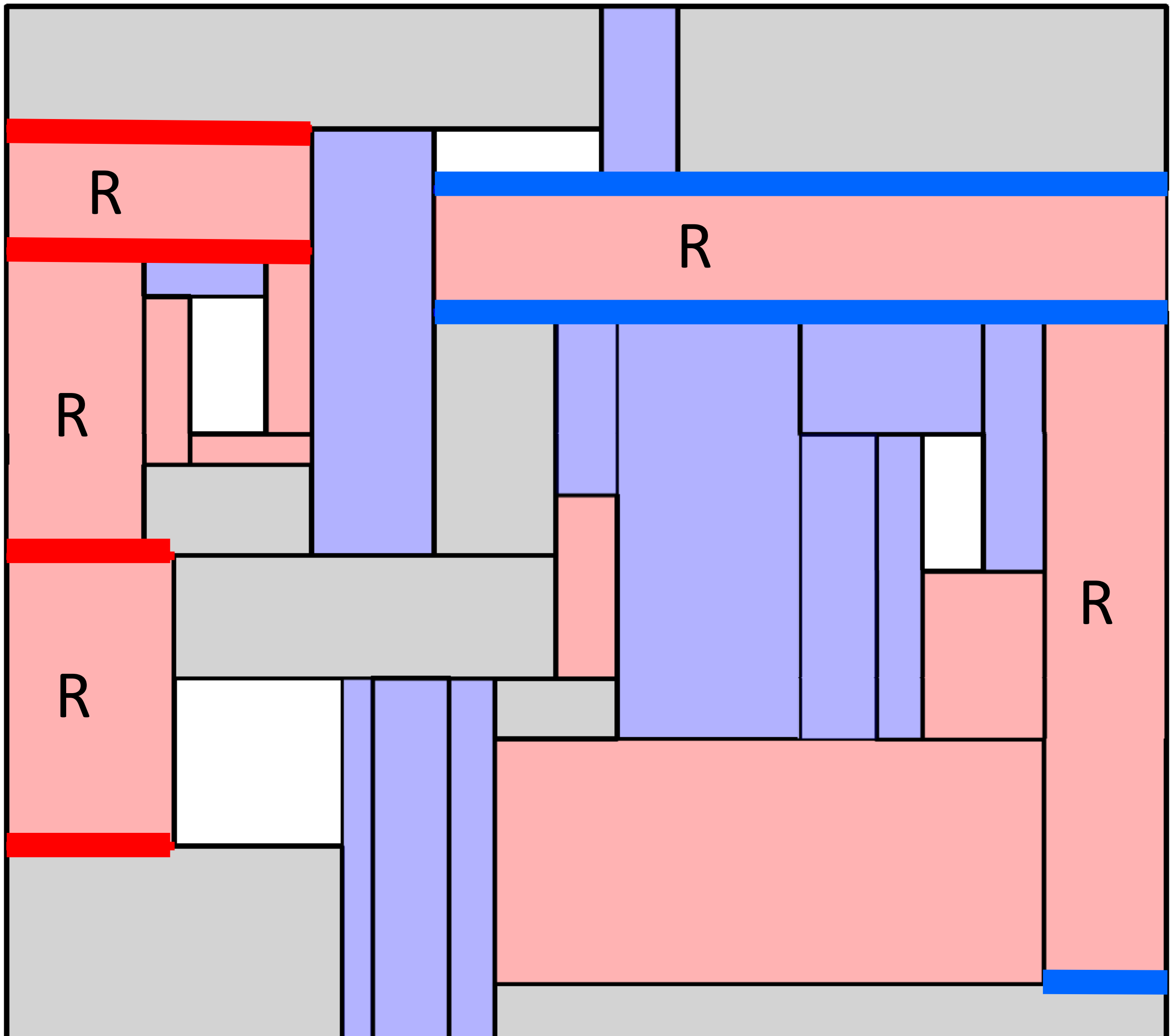
We establish fence (obstacle) segments to maintain the following invariant:

Fence Invariant

For any face Q and rectangle R within Q, if R is exposed to left/right (on its top or bottom), there is a fence (horizontal segment obstacle) established that anchors R to the left/right

Initial Fences

Anchored rectangles R



Key Technical Lemma

Lemma 5.2. *Let I' be a set of maximal rectangles associated with an independent set I of rectangles. Let Q be a CCR whose edges lie on the grid lines of $\mathcal{G}$, defined by the coordinates of the rectangles I' (and thus of I). Let $\{\alpha_1, \dots, \alpha_{k_\alpha}\}$ be a set of "red" horizontal anchored (grid) segments within Q, that are anchored with left endpoints on the left sides (ℓ, ℓ^+, or ℓ^-) of Q, and let $\{\beta_1, \dots, \beta_{k_\beta}\}$ be a set of "blue" horizontal anchored (grid) segments within Q, that are anchored with right endpoints on the right sides (r, r^+, or r^-) of Q. Then, assuming that Q contains at least two grid cells (faces of $\mathcal{G}$), there exists a CCR-cut χ with the following properties:*

(i) *χ partitions Q into $O(1)$ (at most 5) CCR faces;*

(ii) *χ is comprised of $O(1)$ horizontal/vertical segments on the grid $\mathcal{G}$, with endpoints on the grid;*

(iii) *horizontal cut segments of χ are a subset of the given red/blue anchored segments;*

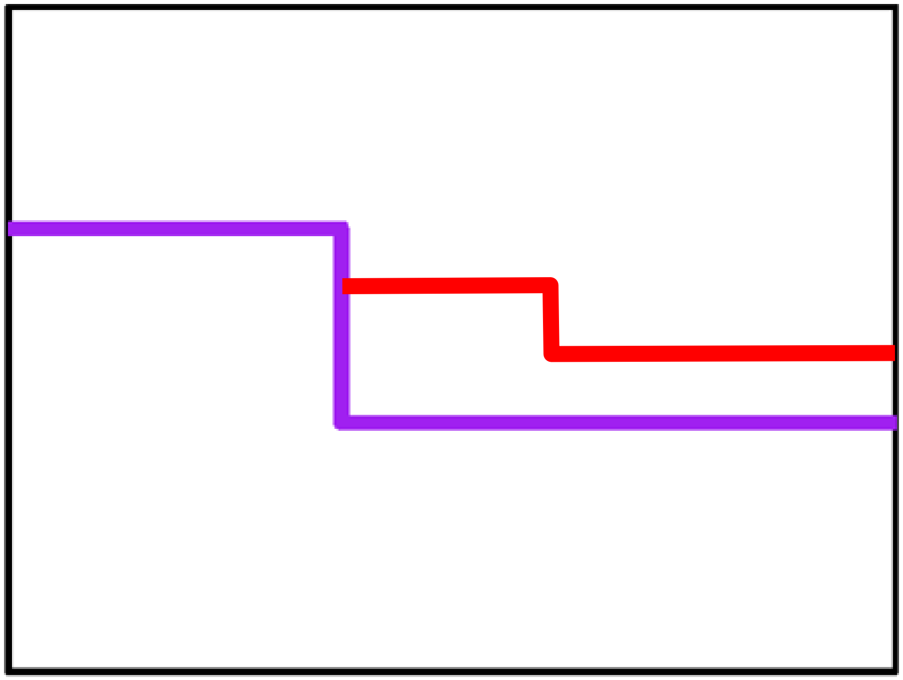
(iv) *vertical cut segments of χ do not cross any of the given red/blue anchored segments;*

(v) *there are at most 2 vertical cut segments of χ.*

For any set of horiz segments (fences) anchored on the left/right of a CCR face Q, there exists a cut, with O(1) horiz/vert segments partitioning Q into at most 5 CCR subfaces, with horiz cut segments contained within the fences (and thus not penetrating any rectangle), and at most 2 vertical cut segments, not crossing any fences.

Care Is Needed

Not enough just to use straight, “L”, and “Z” cuts, since we must create CCR faces with the cuts



Proof: Case Analysis

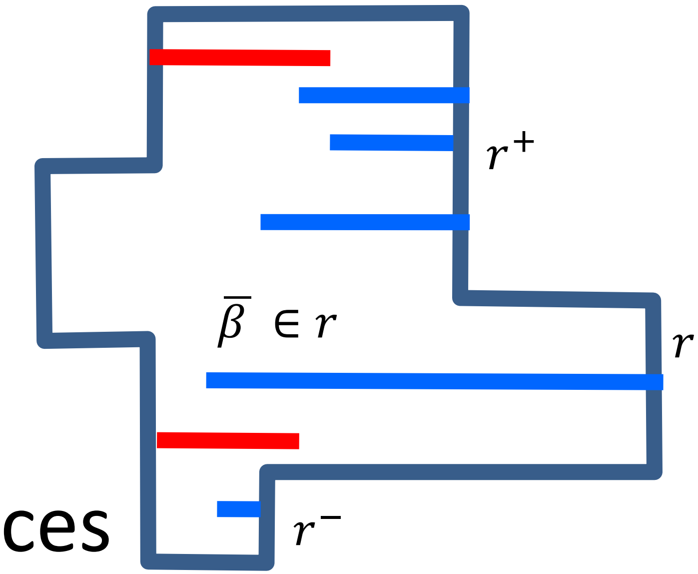
$\overline{\beta}$ = right anchored fence with leftmost left endpoint

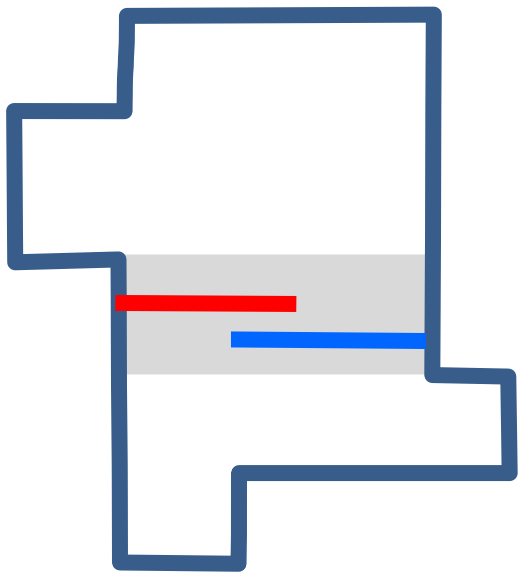
(1) $\overline{\beta} \in r$

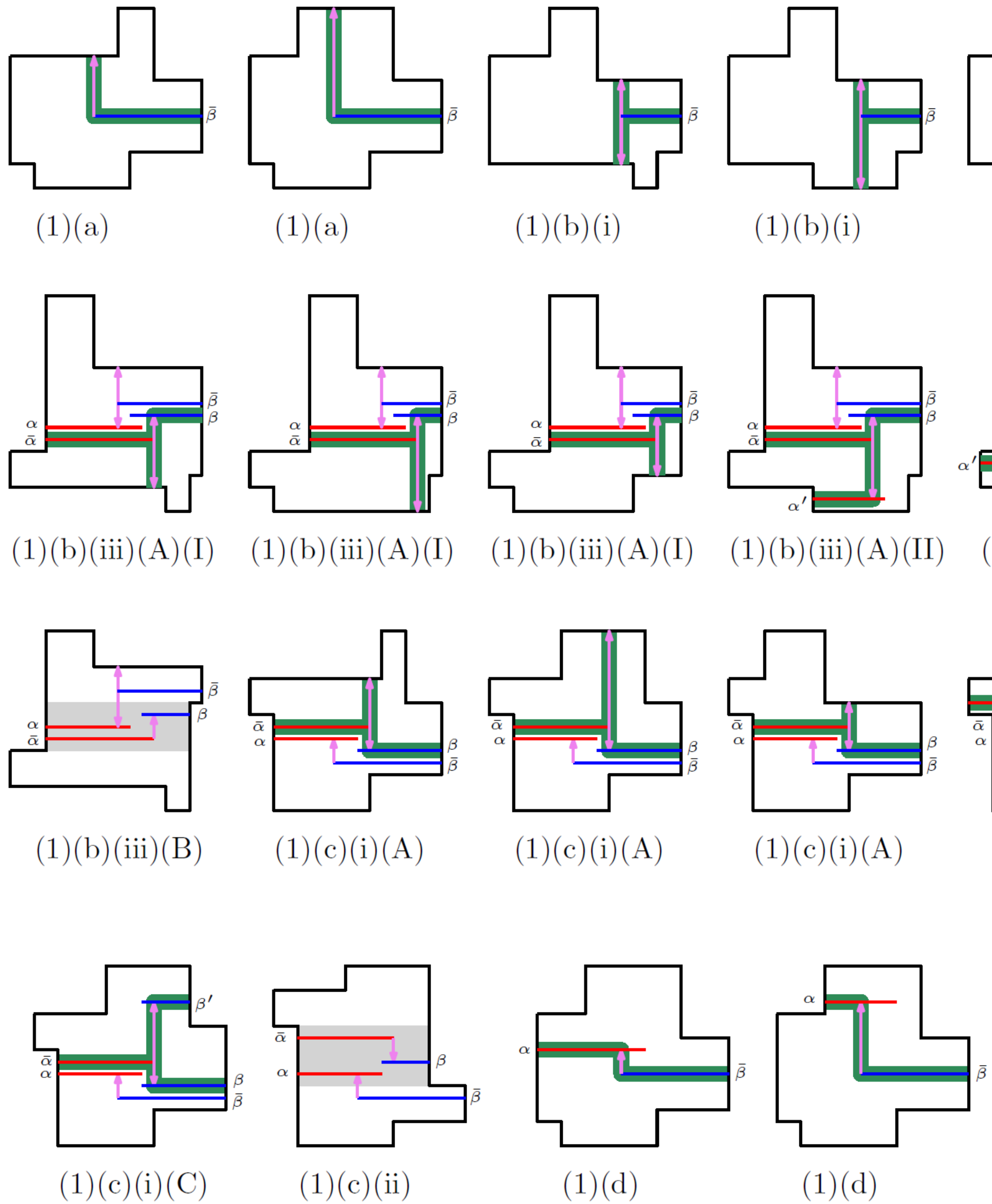
(2) $\overline{\beta} \in r^{+}$ (symmetric: $\overline{\beta} \in r^{-}$)

(3) Mid (gray) has no vertical

Separation between left/right fences

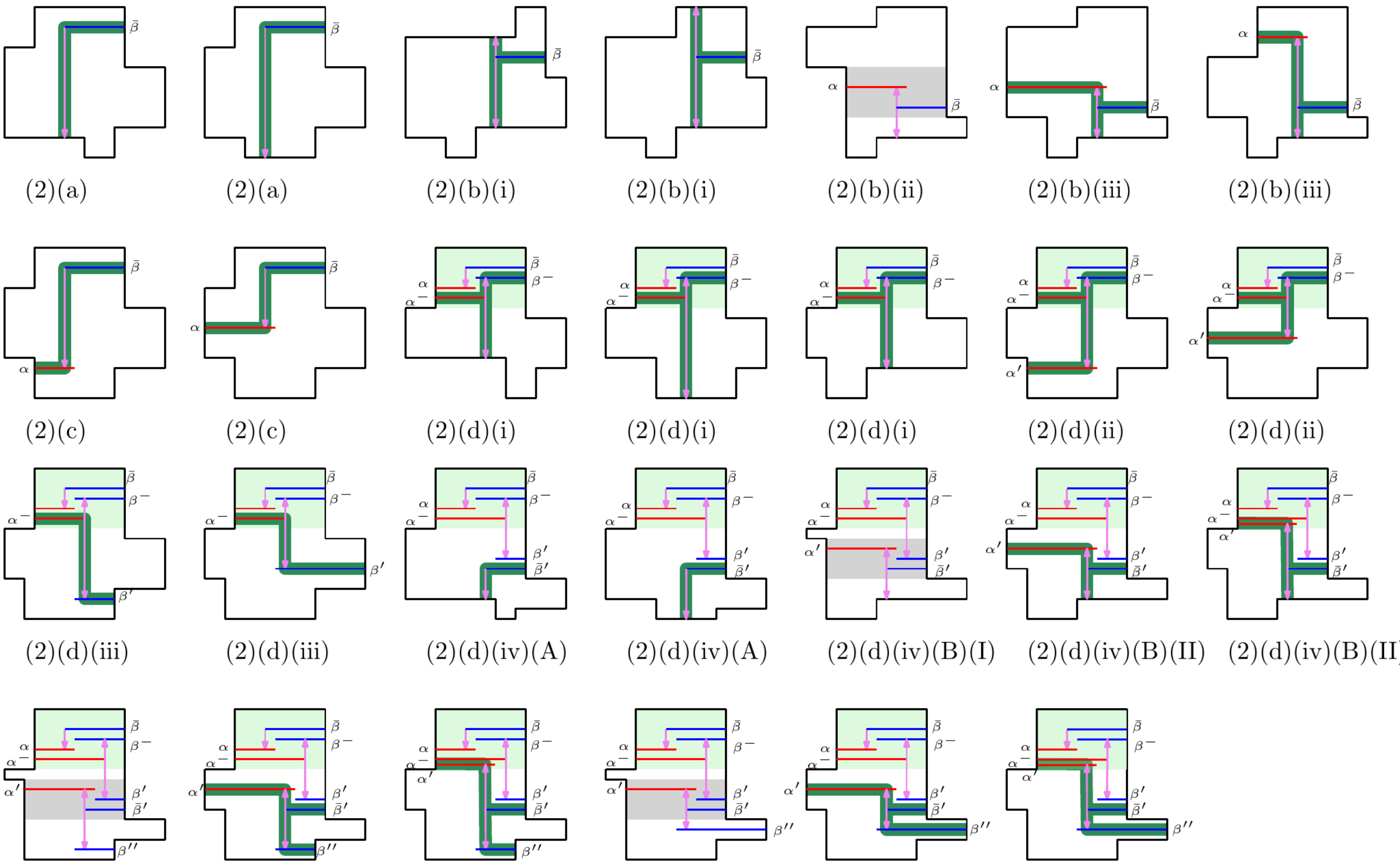






Technical Lemma

Case (1) $\bar{\beta} \in r$

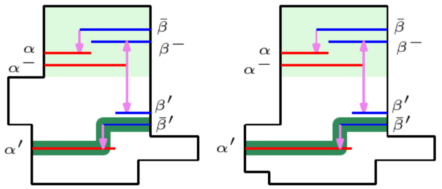


(2)(a) (2)(a) (2)(b)(i) (2)(b)(i) (2)(b)(ii) (2)(b)(iii) (2)(b)(iii)

(2)(c) (2)(c) (2)(d)(i) (2)(d)(i) (2)(d)(i) (2)(d)(ii) (2)(d)(ii)

(2)(d)(iii) (2)(d)(iii) (2)(d)(iv)(A) (2)(d)(iv)(A) (2)(d)(iv)(B)(I) (2)(d)(iv)(B)(II) (2)(d)(iv)(B)(II)

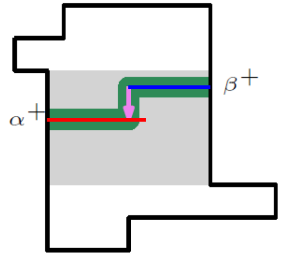
(2)(d)(iv)(C)(I) (2)(d)(iv)(C)(II) (2)(d)(iv)(C)(II) (2)(d)(iv)(D)(I) (2)(d)(iv)(D)(II) (2)(d)(iv)(D)(II)



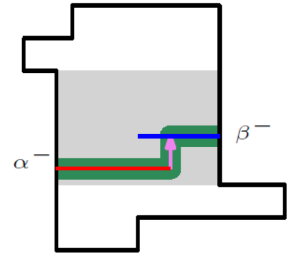
(2)(d)(iv)(E) (2)(d)(iv)(E)

Technical Lemma

Case (2) $\bar{\beta} \in r^{+}$

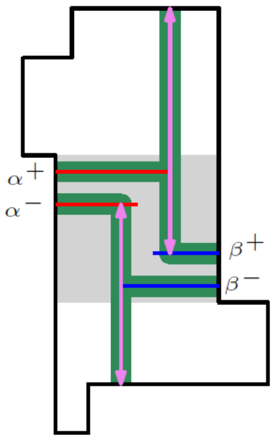


(3)(a)

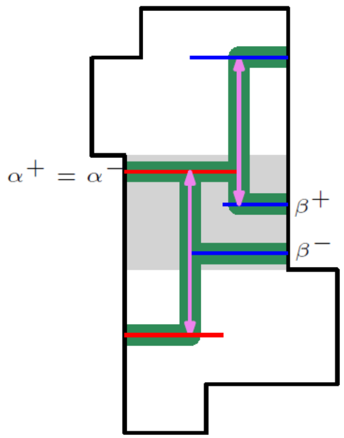


(3)(b)

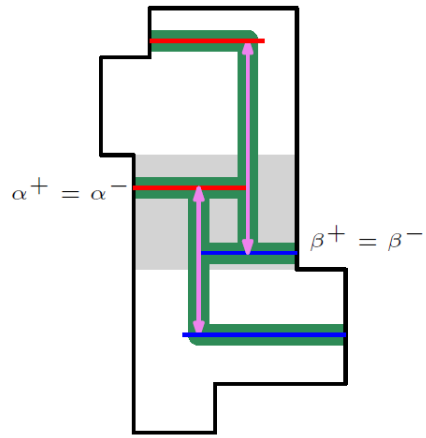
Result in K=5 pieces



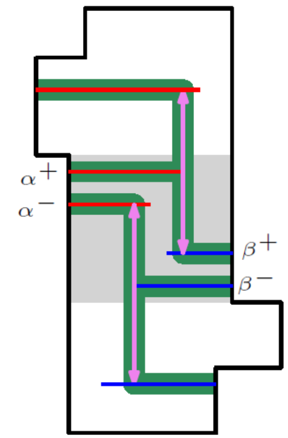
(3)(c)



(3)(c)



(3)(c)

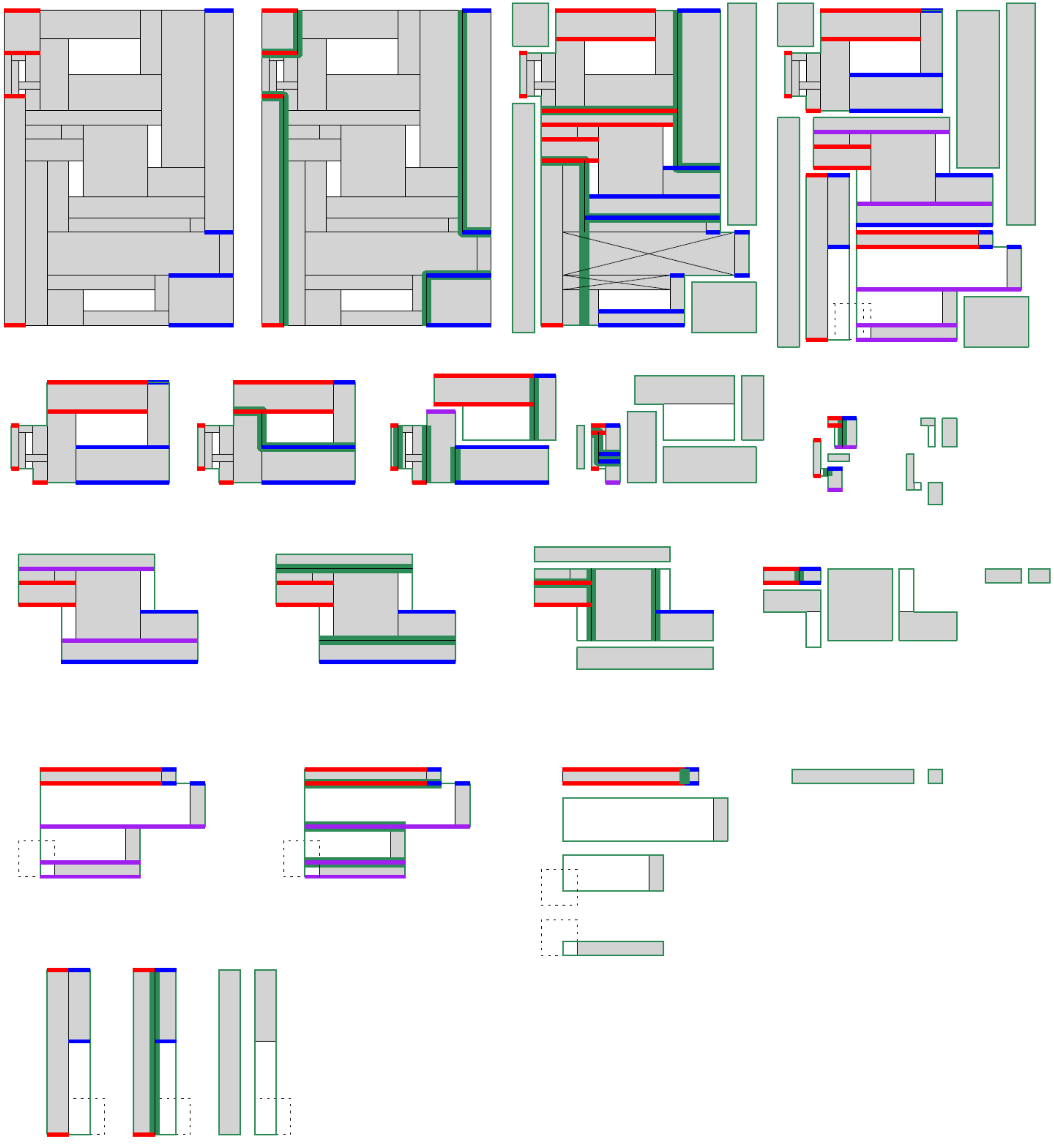


(3)(c)

Technical Lemma

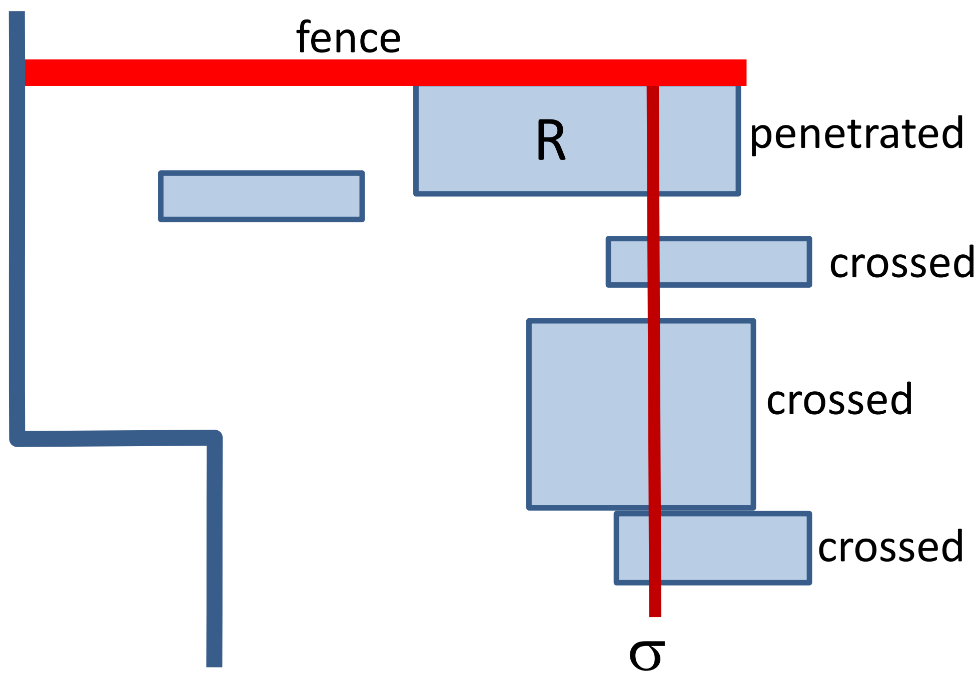
Case (3) Mid (gray) region has no vertical cut separating left/right fences

Example



Fences, Anchored Rectangles Are Not Cut

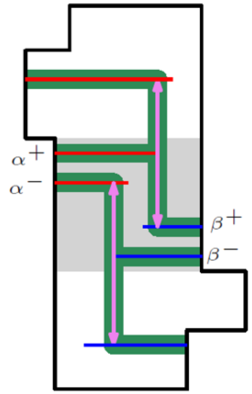
- As a result of the Technical Lemma and the Fence Invariant, no anchored rectangle R is ever *crossed* by a (vertical) cut segment
  (it may be penetrated)



Vertical Cut Segments

- Since at most 2 vertical cut segments in any cut provided by the Technical Lemma case analysis, for any vertical cut segment σ, to ≥1 of its sides (left or right) there is no other vertical cut segment of the cut

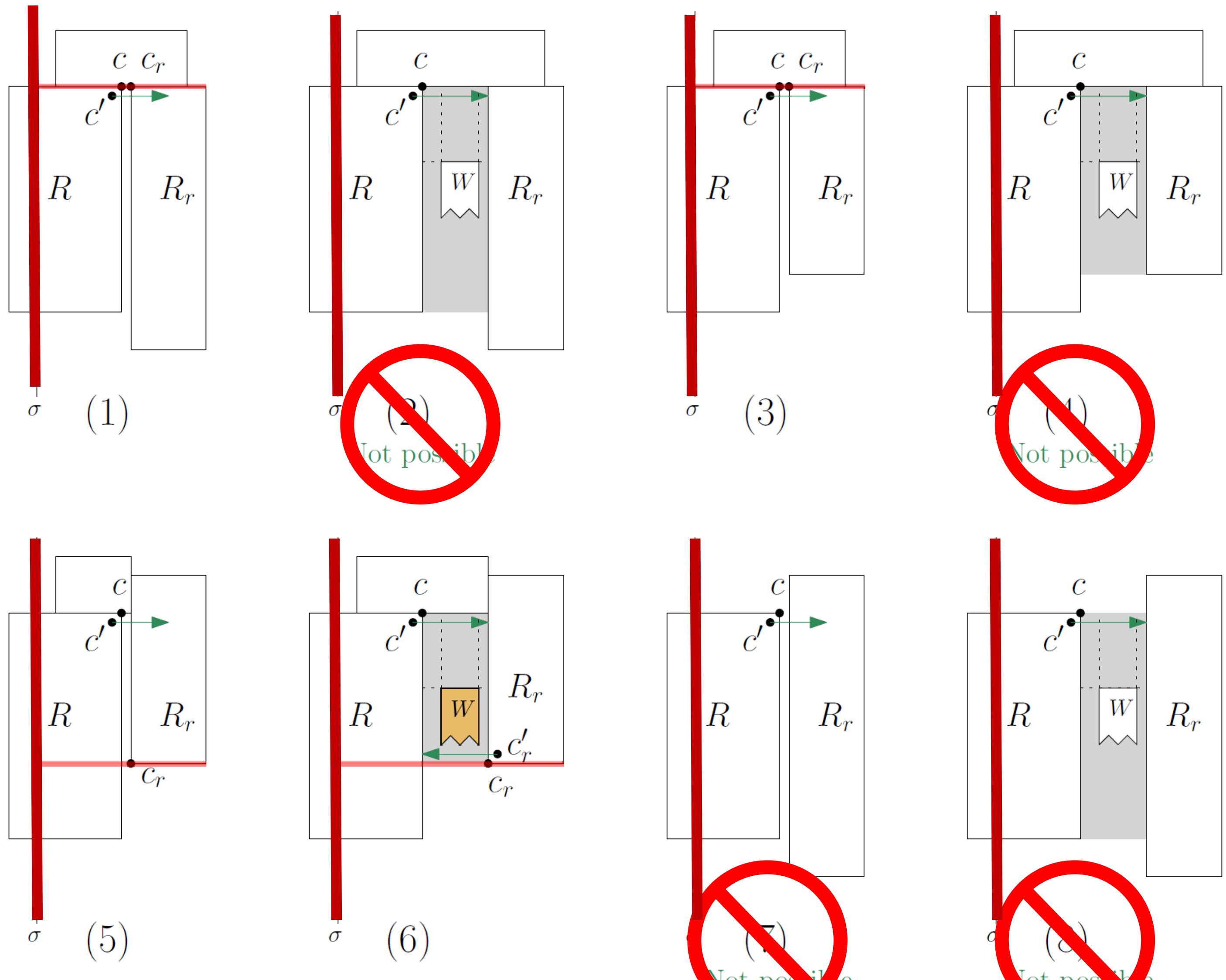
  WLOG: No vert cut segment to the right of σ



- Goal: Charge off non-red rectangles that are crossed by vertical cut segment σ

Charging Off a Non-Red Crossed Rectangle, R

WLOG: No vert cut segment to the right of σ



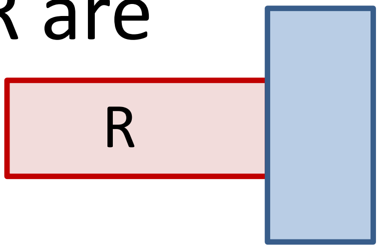
Note: R_r is **not** nested on its left

Charging Properties

- No corner is ever charged more than once
- If we charge a corner, c, of R_r , then R_r has not previously been crossed (and discarded)

  R would have become exposed; fence

- If we charge a corner, c, of R_r , then R_r will not subsequently be crossed (since a fence is established)
- At most 2 corners of a red rectangle R are charged (left ones or right ones)



In cases (1),(3),(5),(6), the charged rectangle R_r is **not** nested on its left

Accounting for Rectangles that are Cut/Crossed

- Red rectangles: h_0 uncut; h_χ are cut (discarded)
- Non-red rect: m_0 uncut; m_χ are cut (discarded)
- Goal: Show that $h_0+m_0 \geq k/10$

- Charge of "1" for each cut non-red rectangle
- Total charge = $m_\chi \leq \; 2h_0 + 4m_0$

Only uncut rectangles are assigned charge

≤2 corners of red rectangle charged

≤4 corners of non-red rectangle charged

Accounting for Rectangles that are Cut/Crossed

- Total charge = $m_\chi \leq 2h_0 + 4m_0$

Thus,

Recall: $m_0 + m_\chi \geq k/2$ (WLOG)

$$4(h_0 + m_0) \geq m_\chi + 2h_0 = (m_0 + m_\chi) + 2h_0 - m_0 \geq (k/2) + 2h_0 - m_0$$

$$2h_0 + 5m_0 \geq k/2,$$

$$5(h_0 + m_0) - 3h_0 \geq k/2.$$

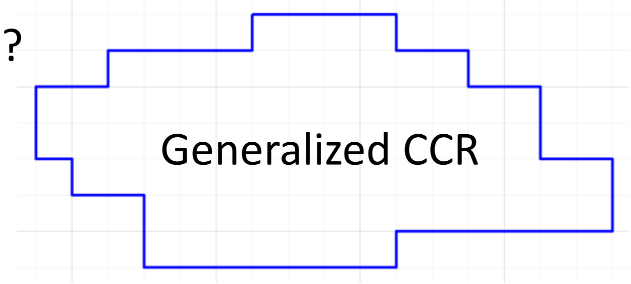
$$5(h_0 + m_0) \geq k/2,$$

Thus, $h_0 + m_0 \geq k/10$ QED

Conclusion

- Improve the approx factor and/or running time
- PTAS?

Better than factor 3? 2?



- Weights
  - O(log n/loglog n)-approx [Chan, Har-Peled]
  - O(loglog n)-approx [Chalermsook, Walczak, SODA21]
- Higher dimensions?
- Pach-Tardos conjecture about perfect BSP's

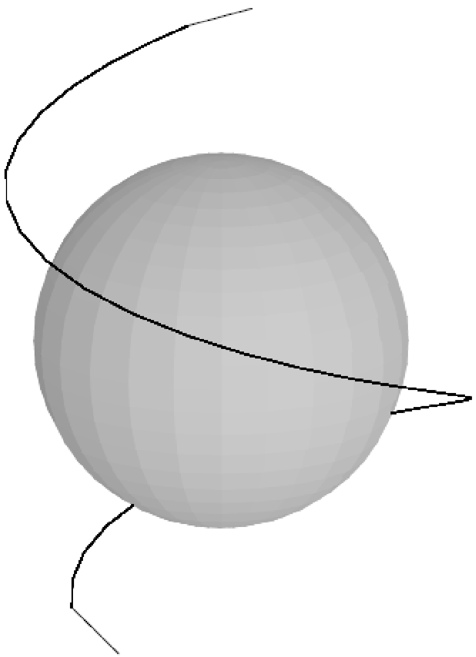
Conjecture 1. *For any set of n interior-disjoint axis-aligned rectangles in the plane, there exists a subset of size $\Omega(n)$ that has a perfect orthogonal BSP.*

Problem Discussed

- Added 3 slides about the problem mentioned: Find a shortest path/cycle in outer space in order to do a visibility coverage of planet earth

External Watchman Path for a Sphere

- Short **Path**
  Length 11.08



Two segments and a spiral:

$$\{(\underline{(1-at^2)}\sin(b\pi t),\ \underline{(1-at^2)}\cos(b\pi t),\ ct) \mid -1 \le t \le 1\}$$

Fatten spiral near middle

$$a = 0.4,\ b = 1.18,\ c = 1.12,\ x_0 = -0.37,\ y_0 = -0.199,\ z_0 = 1.24$$

By computer search

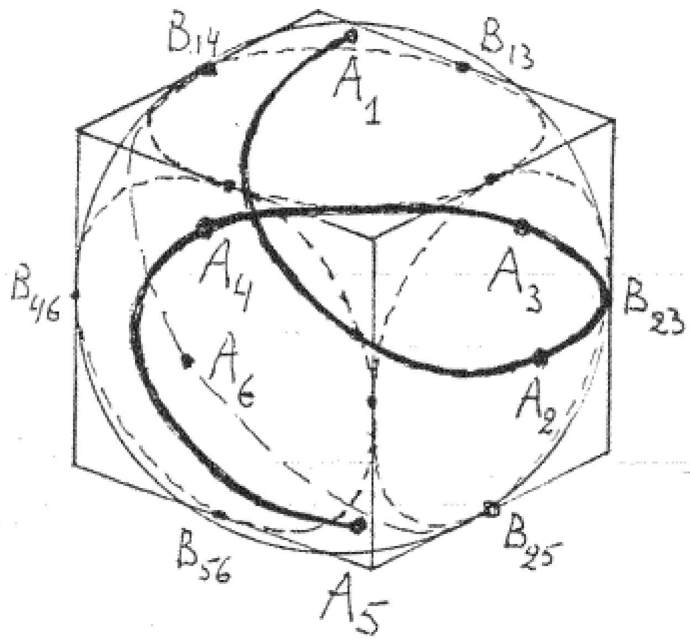
The Asteroid Surveying Problem and Other Puzzles

Timothy M. Chan   Alexander Golynski   Alejandro López-Ortiz   Claude-Guy Quimper

[SoCG'03 video]

External Watchman Path for a Sphere

- Short **Path**
  Length 10.726



a "rather short" inspection curve that lies at the constant altitude of $\sqrt{2} - 1$

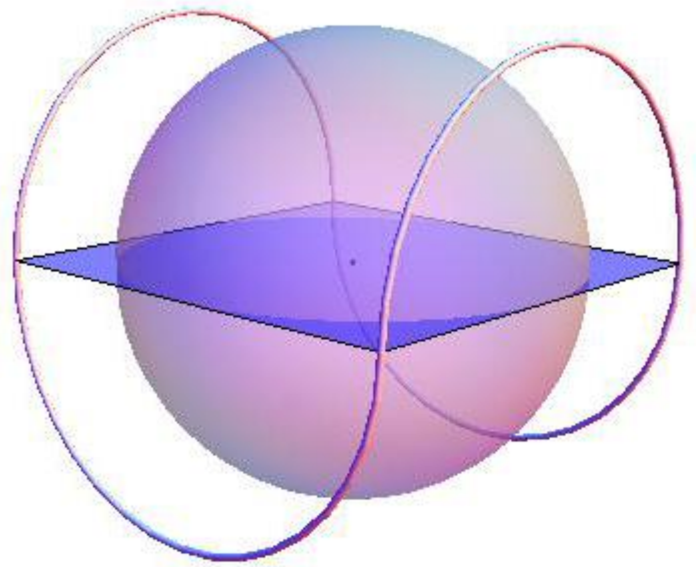
$$L = \pi(2 + \sqrt{2}) \approx 10.726$$

SHORTEST INSPECTION CURVES FOR THE SPHERE

V. A. Zalgaller* *Journal of Mathematical Sciences, Vol. 131, No. 1, 2005*

External Watchman Cycle for a Sphere

Shortest **Cycle** ?



"Shortest Inspection Curves for the Sphere"
V. A. Zalgaller

"baseball stitch curve"

[discussions: Jin-ichi Itoh, Joe O'Rourke, Anton Petrunin, Y. Tanoue, Costin Vilcu]

108 double stitches