

SODA Report

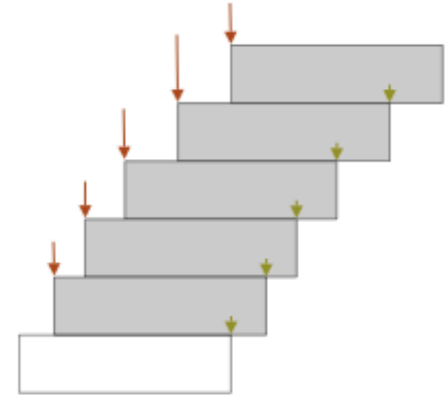
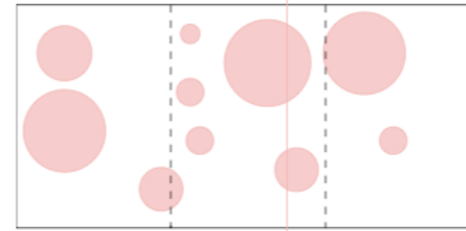
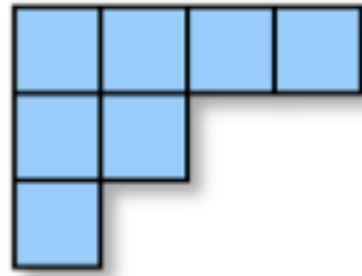
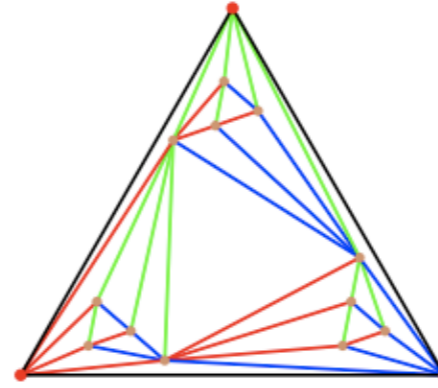


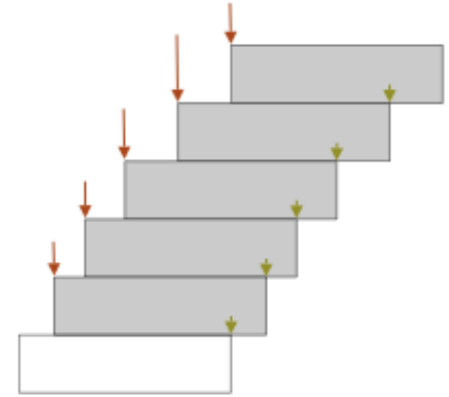
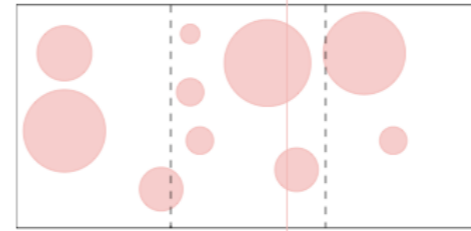
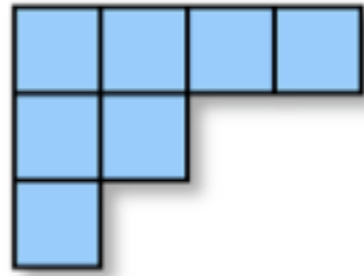
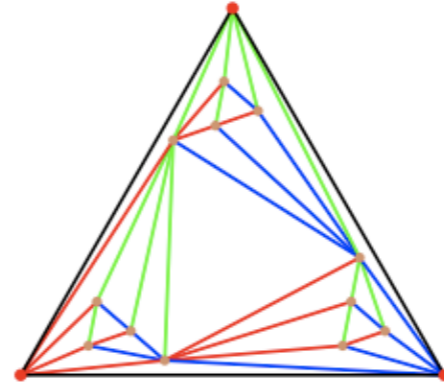
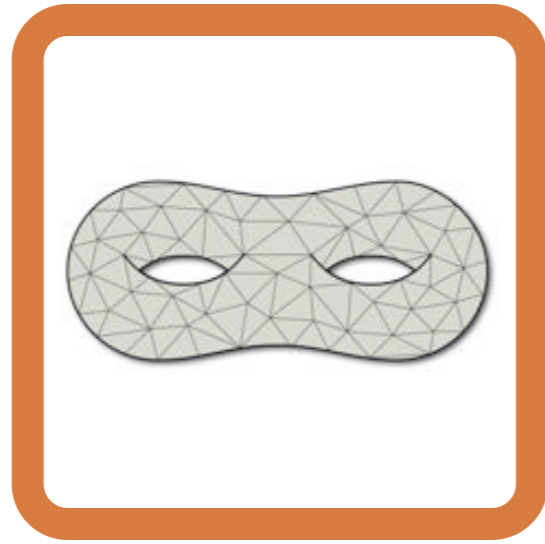
ACM-SIAM Symposium
on Discrete Algorithms

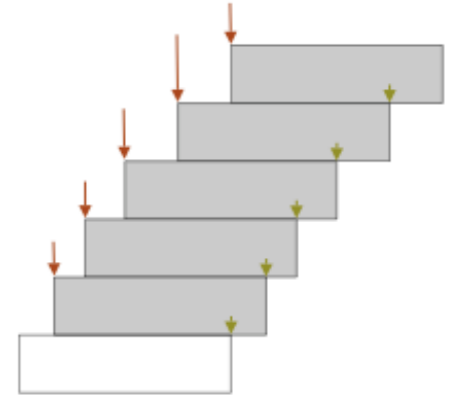
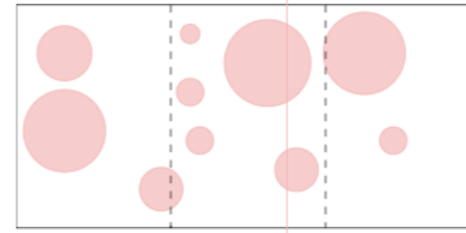
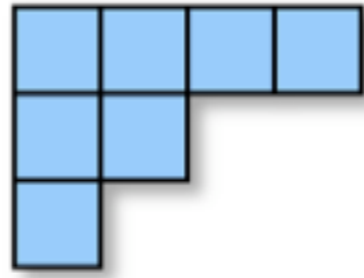
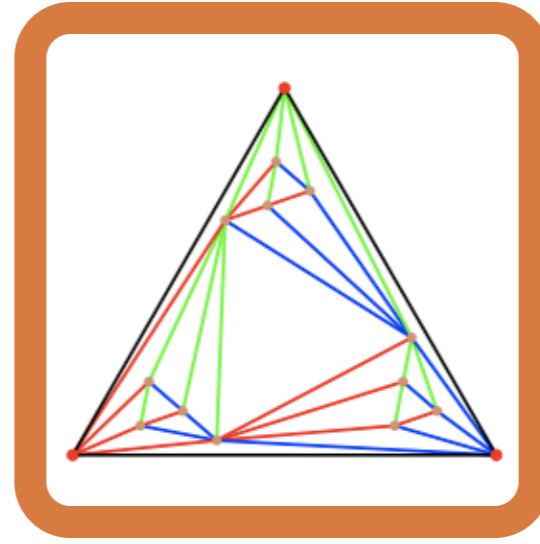
January 20-22, 2008
Holiday Inn, Golden Gateway
San Francisco, California

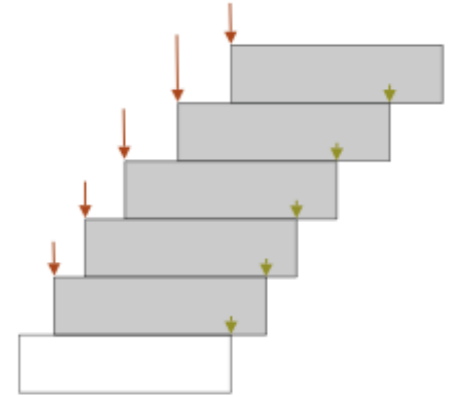
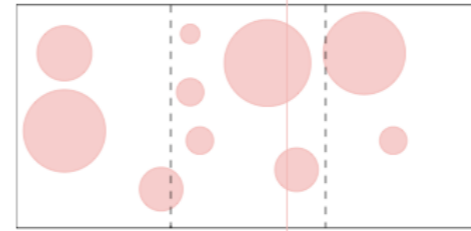
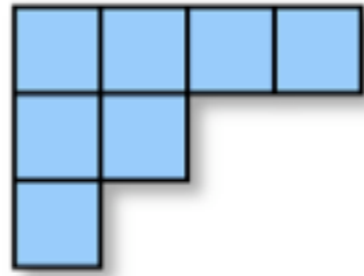
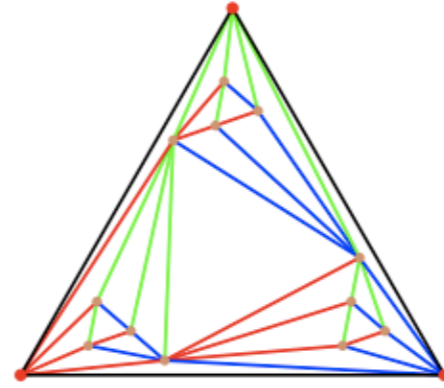
Julián Mestre, AGI

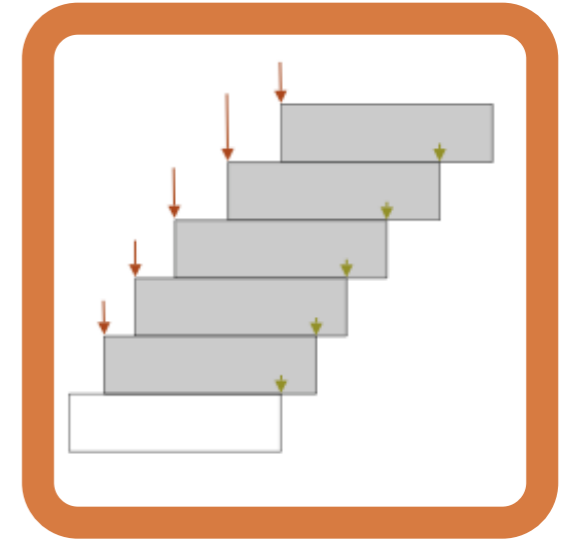
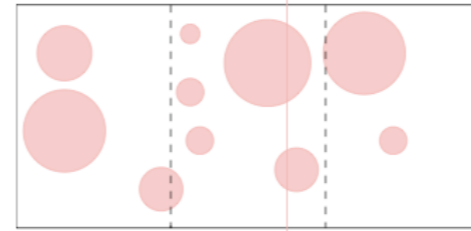
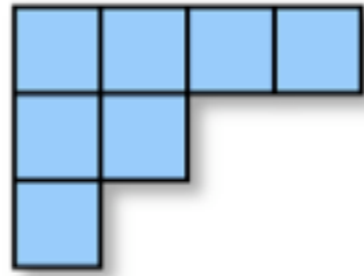
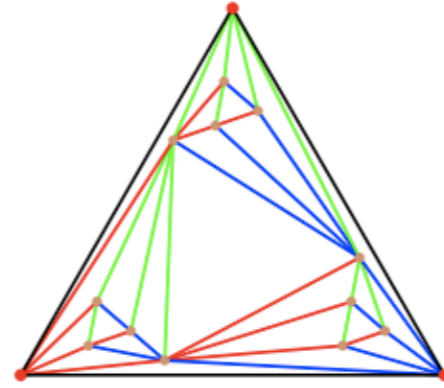


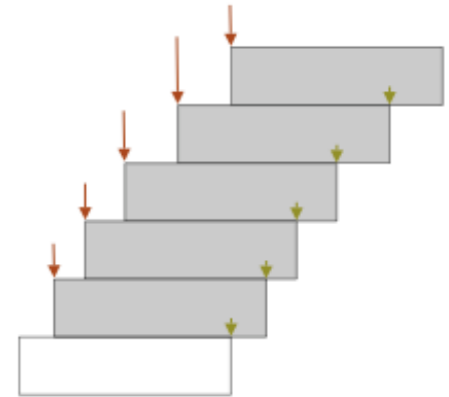
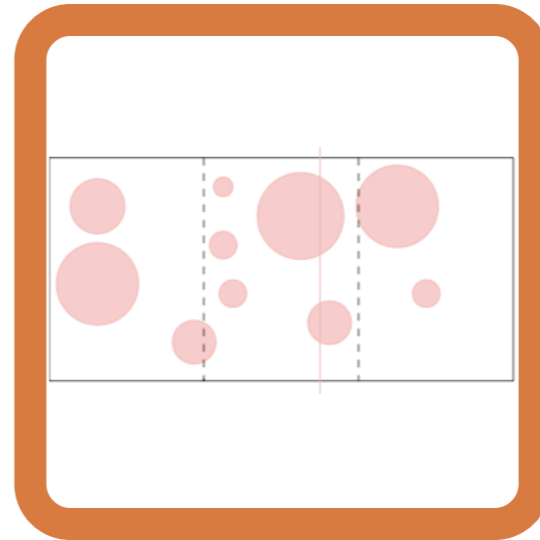
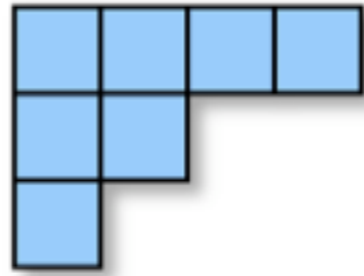
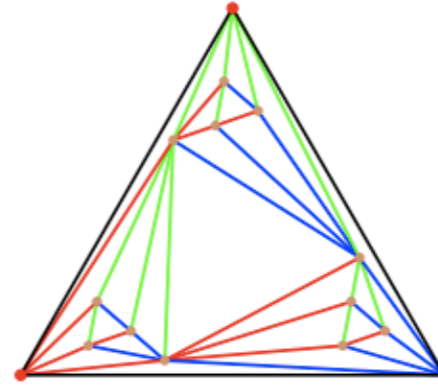
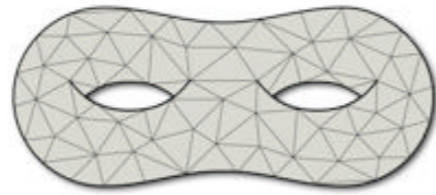


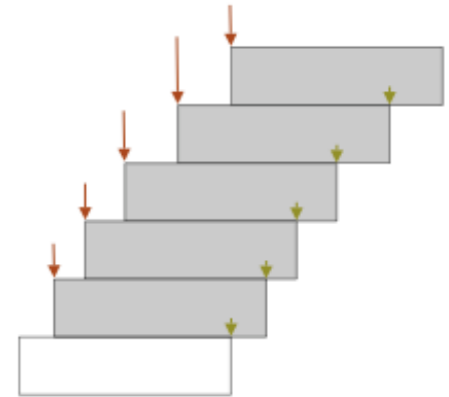
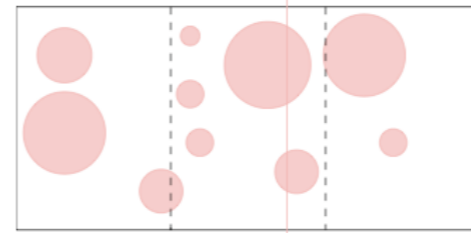
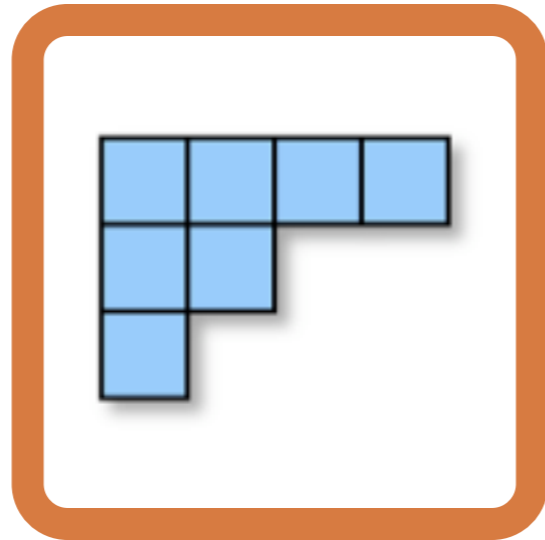
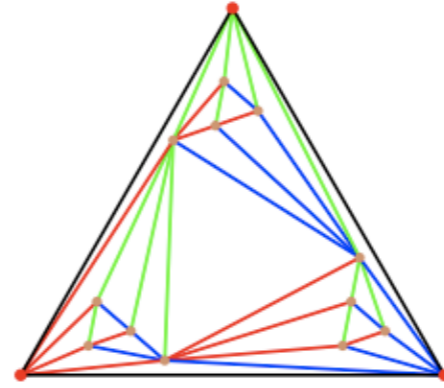


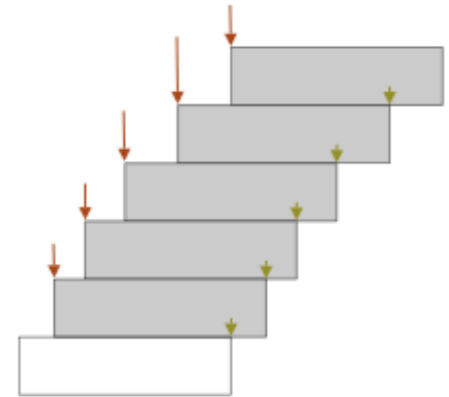
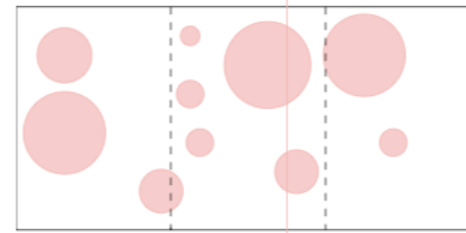
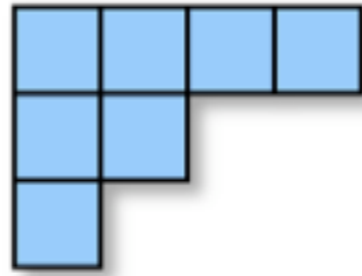
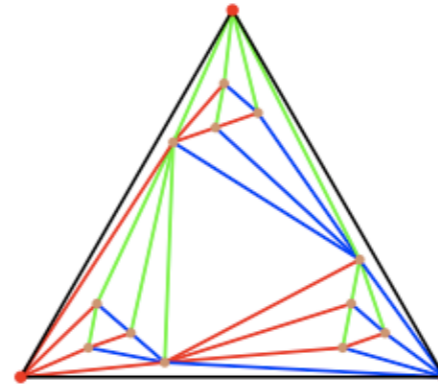


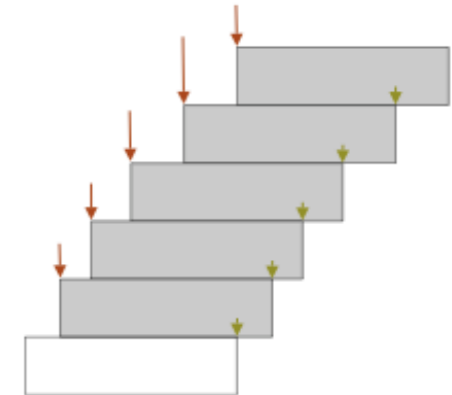
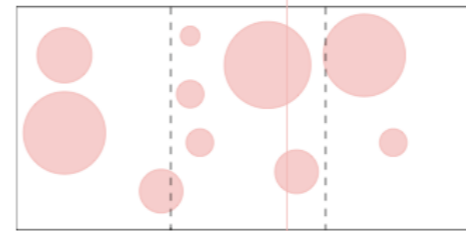
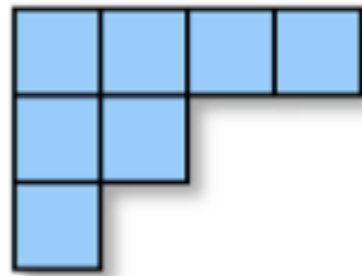
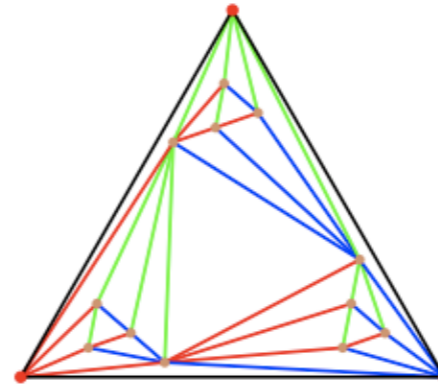










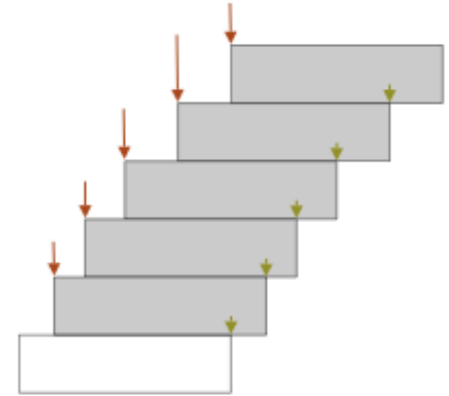
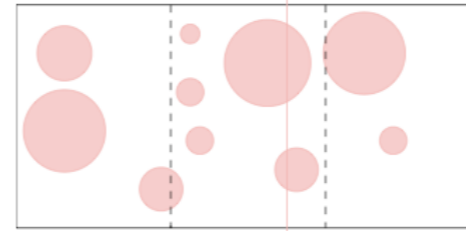
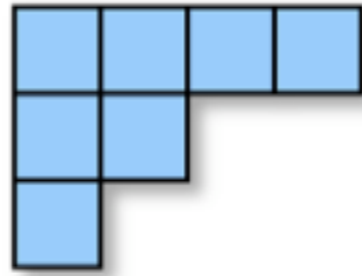
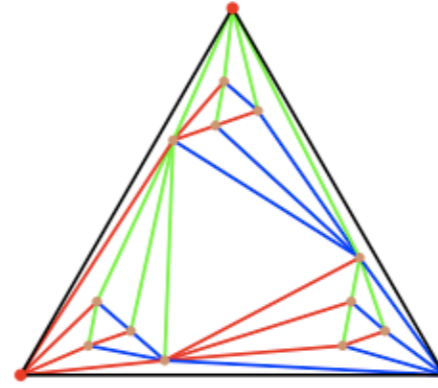
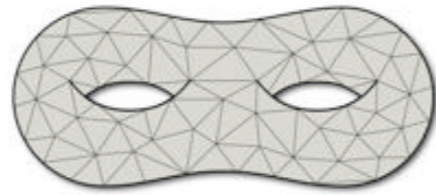


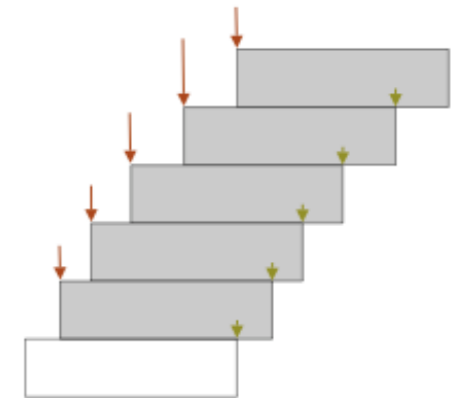
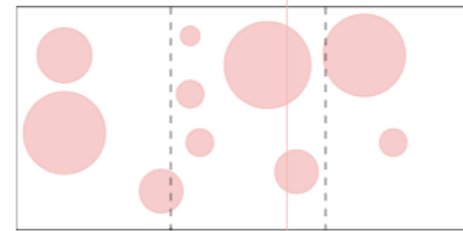
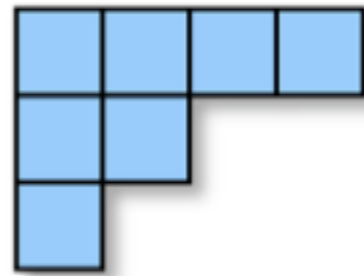
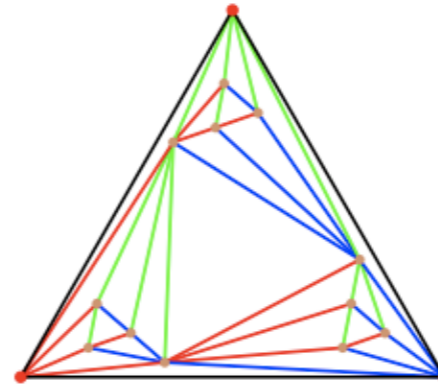
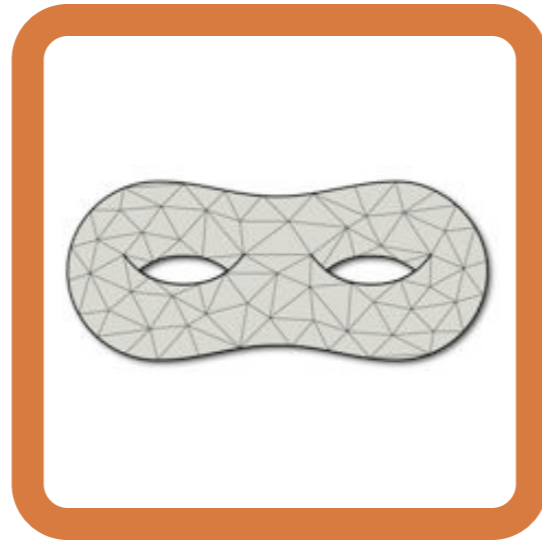
Business Meeting

Business Meeting



- ◎ \approx 450 submissions, 135 accepted
- ◎ Electronic Proceedings?
- ◎ Next SODA:
 - Manhattan, NYC
 - Claire Mathieu, PC Chair
 - Two submissions deadlines: Abstract + Paper
 - Must include all your proofs





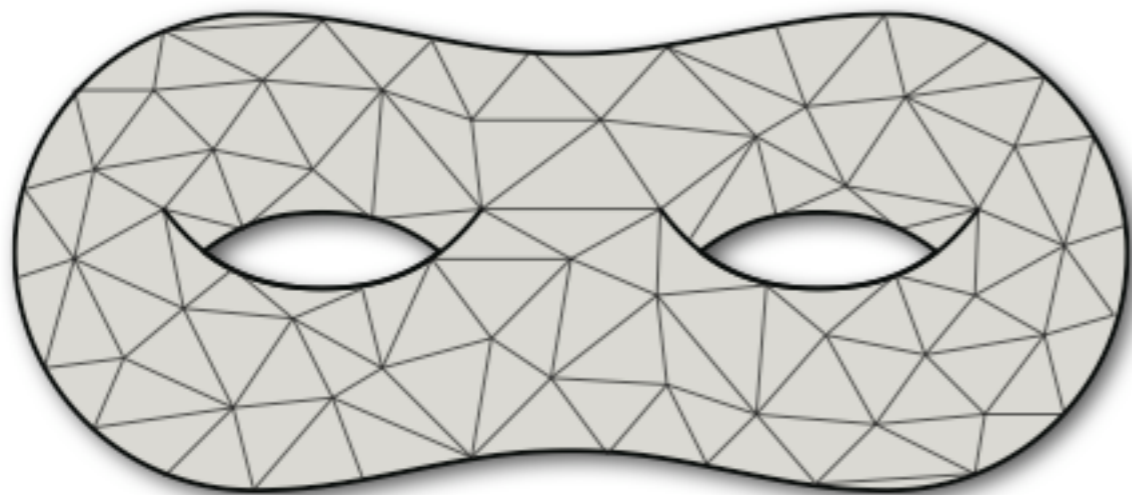
Finding one Tight Cycle

by S. Cabello, M. DeVos, J. Erickson, and B. Mohar

Finding one Tight Cycle



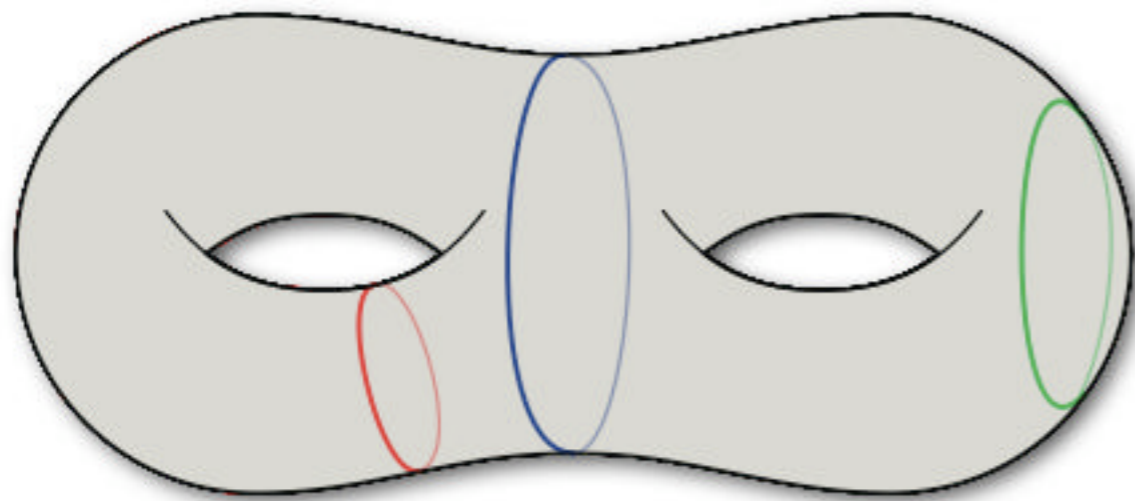
- Combinatorial surface (G, Σ) where G is weighted graph embedded on a 2-manifold Σ
- Tight cycle: Shortest in its homeotopy class



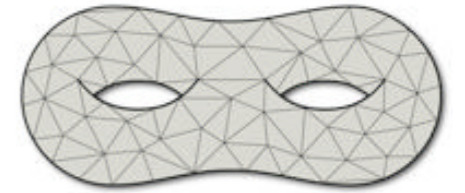
Finding one Tight Cycle



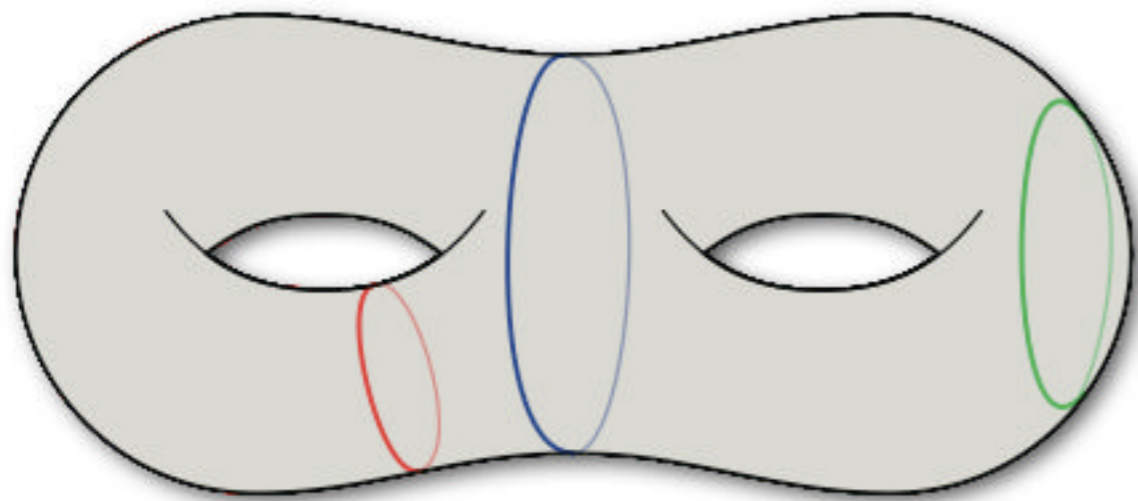
- Combinatorial surface (G, Σ) where G is weighted graph embedded on a 2-manifold Σ
- Tight cycle: Shortest in its homeotopy class



Finding one Tight Cycle



- Combinatorial surface (G, Σ) where G is weighted graph embedded on a 2-manifold Σ
- Tight cycle: Shortest in its homeotopy class



Find a tight
nontrivial cycle in
 $O(n \log n)$ time

Finding one Tight Cycle

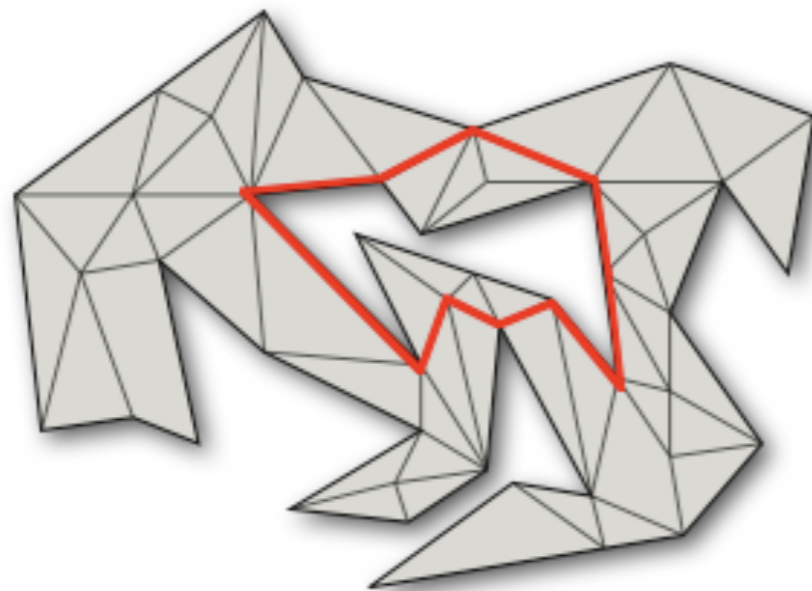


- genus = 0 & boundries < 2 (only contractible cycles)
- genus = 0 & boundries = 2 (exactly one)
- otherwise do “topological surgery”
- runtime dominated by min-cut computation [F]

Finding one Tight Cycle



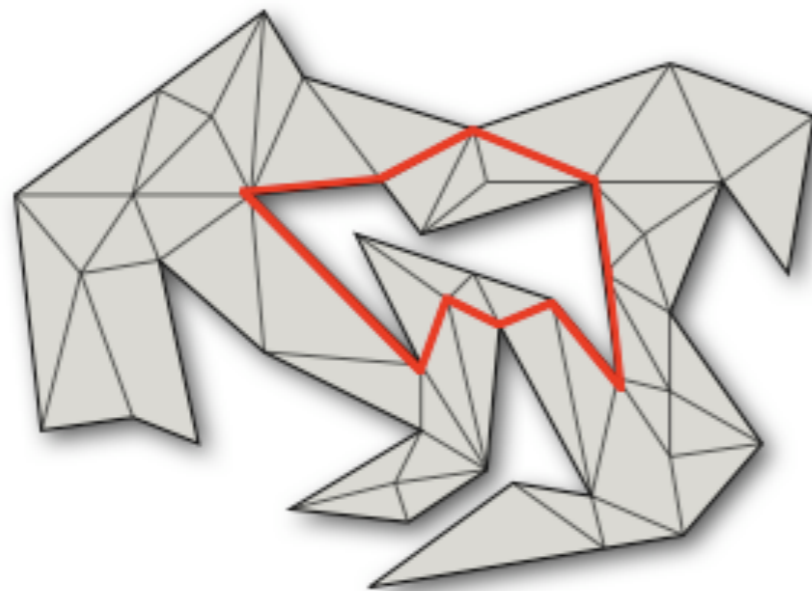
- genus = 0 & boundries < 2 (only contractible cycles)
- genus = 0 & boundries = 2 (exactly one)
- otherwise do “topological surgery”
- runtime dominated by min-cut computation [F]



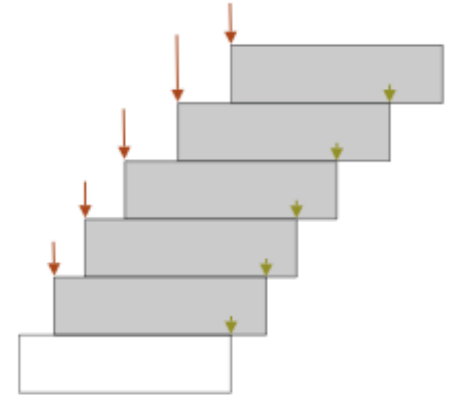
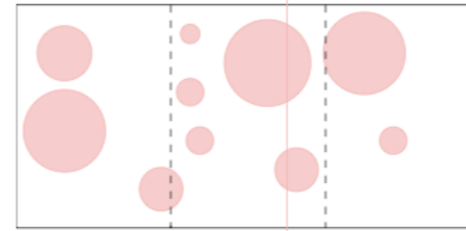
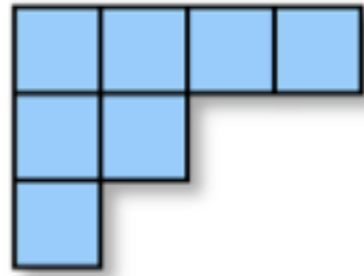
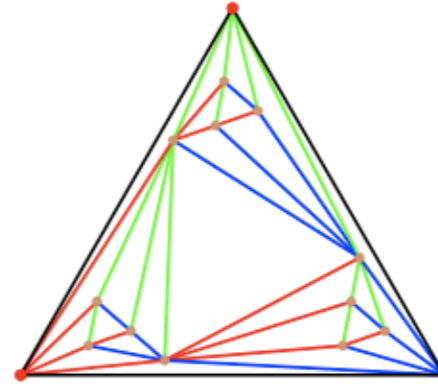
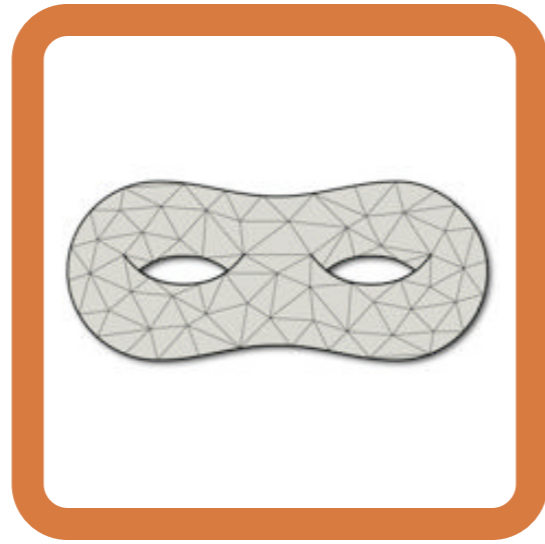
Finding one Tight Cycle

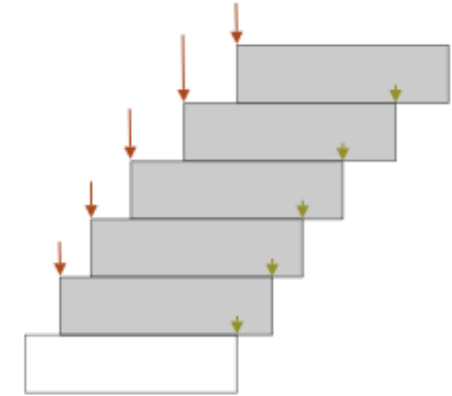
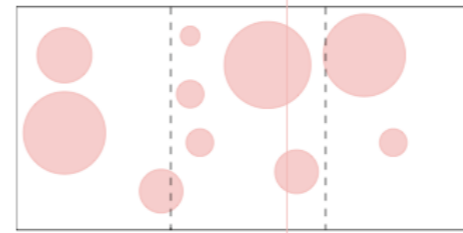
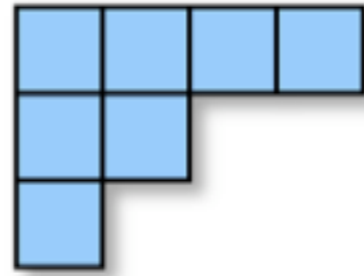
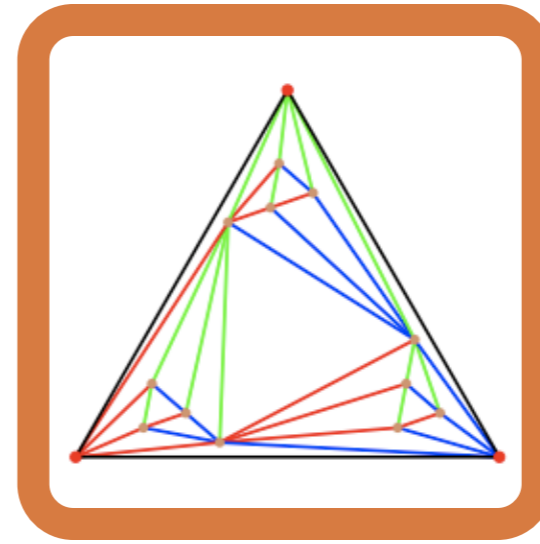


- genus = 0 & boundries < 2 (only contractible cycles)
- genus = 0 & boundries = 2 (exactly one)
- otherwise do “topological surgery”
- runtime dominated by min-cut computation [F]



Find a min-cut in
planar graphs in
 $O(n)$ time

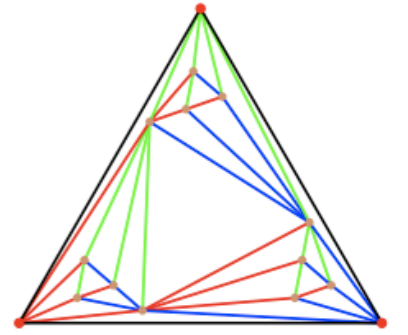




Greedy Drawings of Triangulations

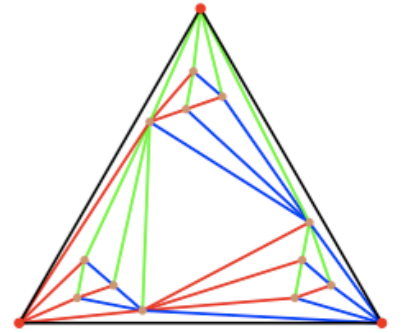
by R. Dhandapani

Greedy Drawings



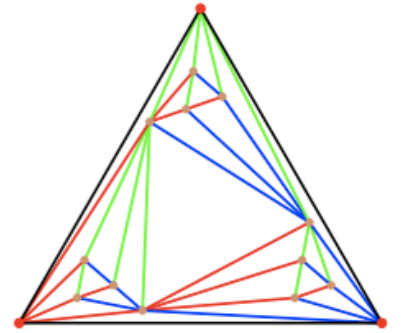
- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does

Greedy Drawings



- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does

Greedy Drawings

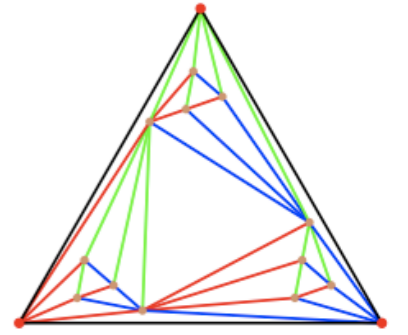


- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does

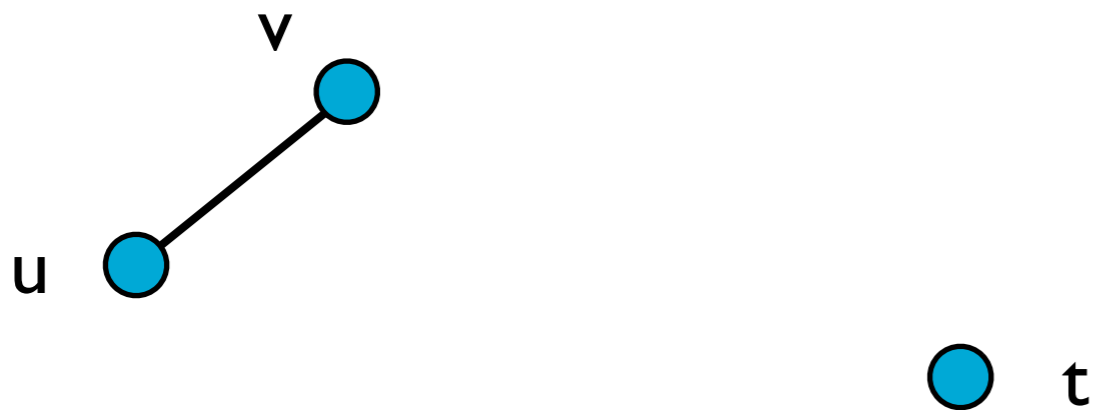
u ●

● t

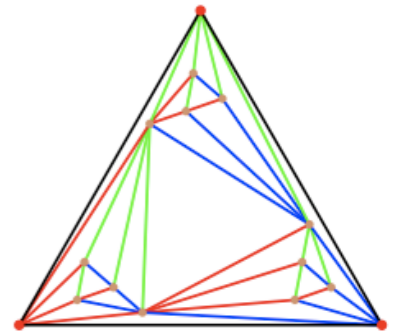
Greedy Drawings



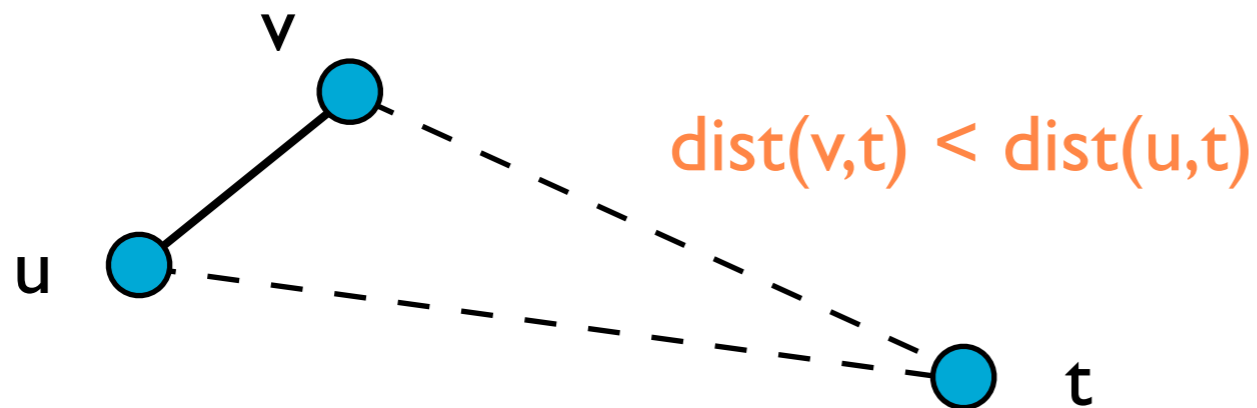
- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does



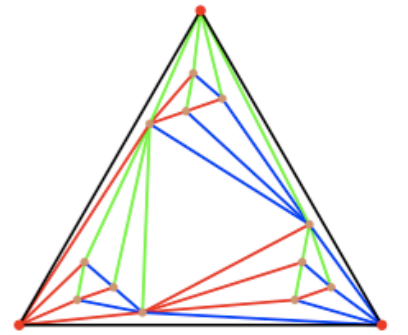
Greedy Drawings



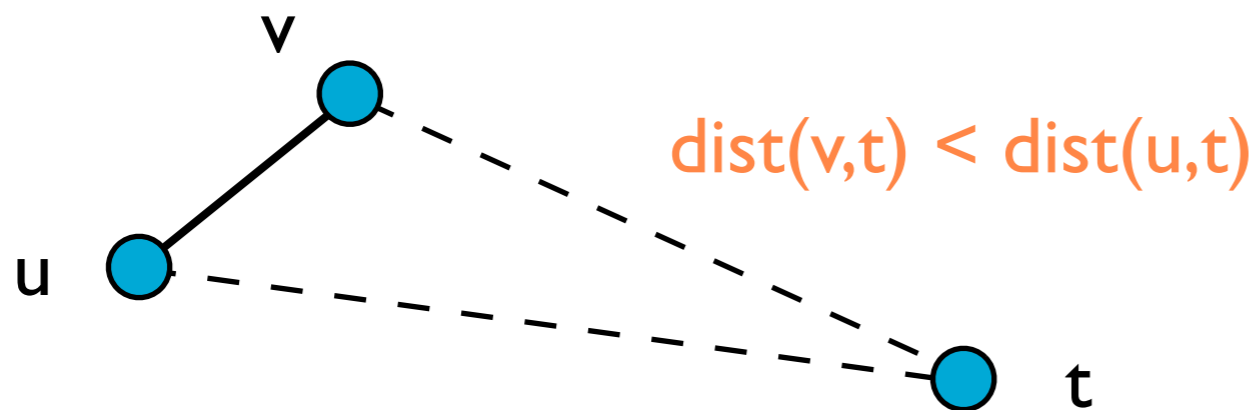
- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does



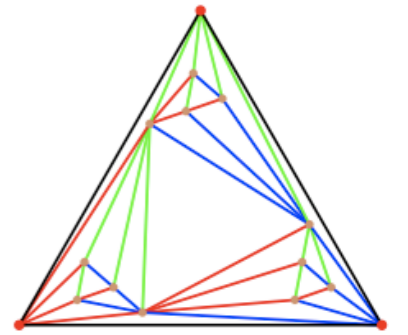
Greedy Drawings



- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does

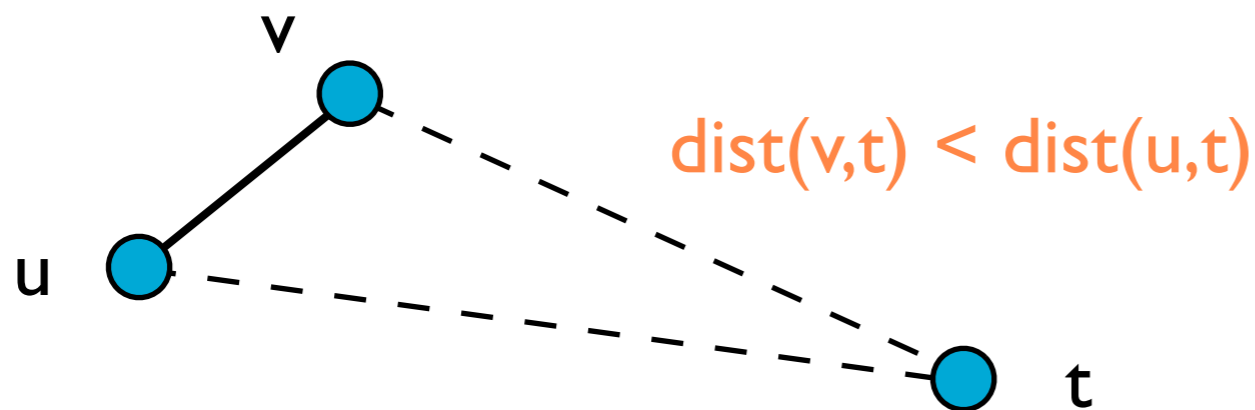


Greedy Drawings

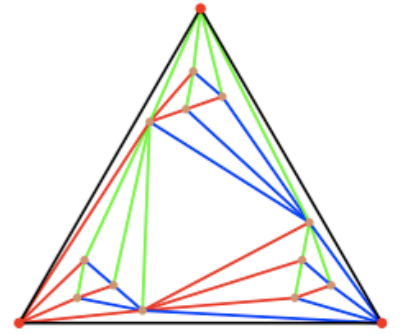


- Ah-hoc networks: Geographical routing
- Two pitfalls : GPS is expensive & “lakes”
- Solution: Virtual coordinates & greedy drawings [RPSS]
- Which graphs have greedy drawings? [PR]
 - $K_{1,6}$ does not
 - Conjecture: Every planar 3-connected graph does

Every triangulated planar graph has a Greedy Drawing

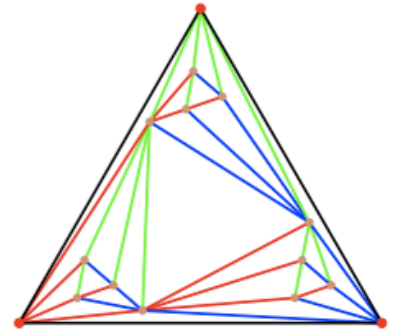


Greedy Drawings

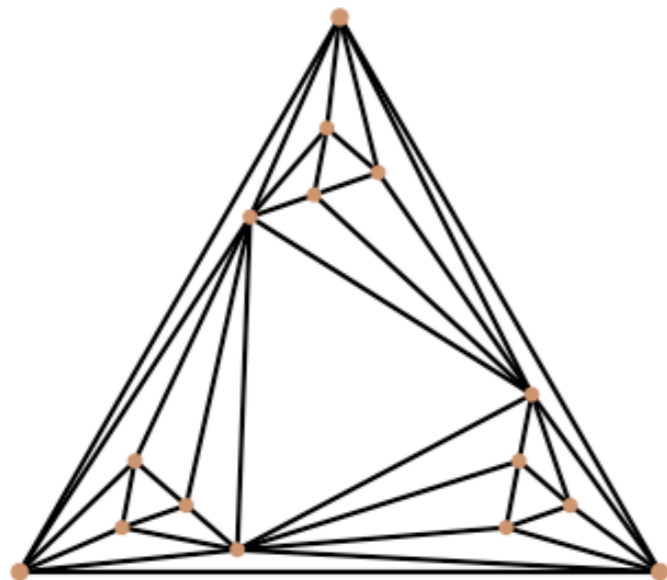


- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist

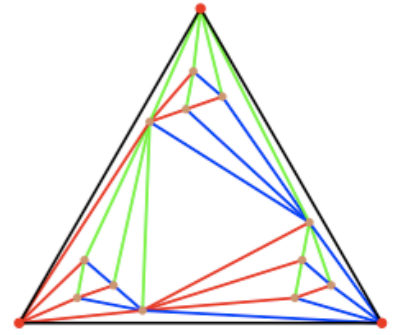
Greedy Drawings



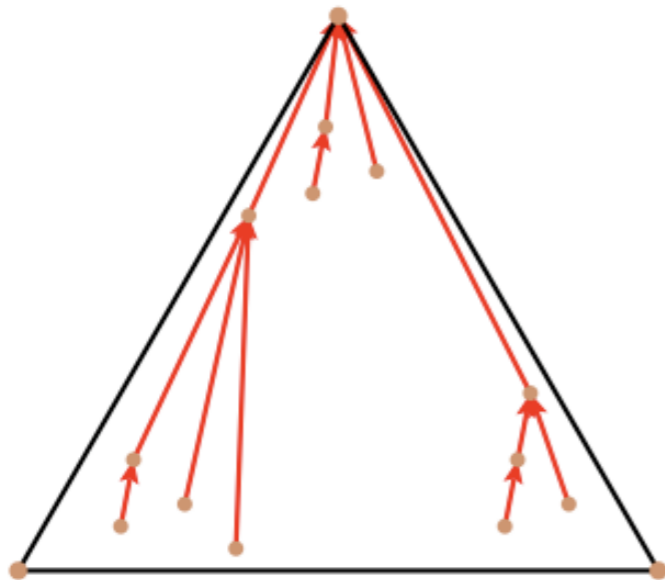
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



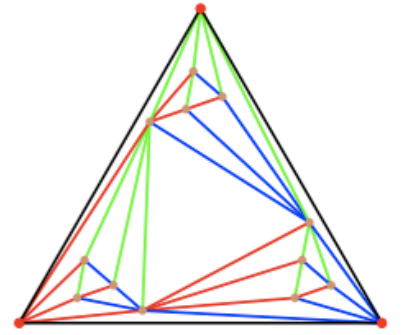
Greedy Drawings



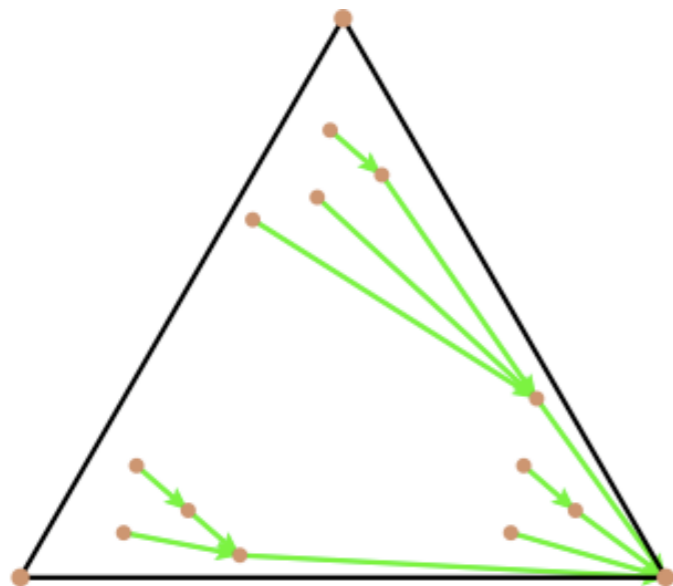
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



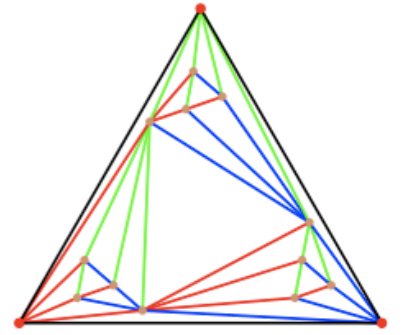
Greedy Drawings



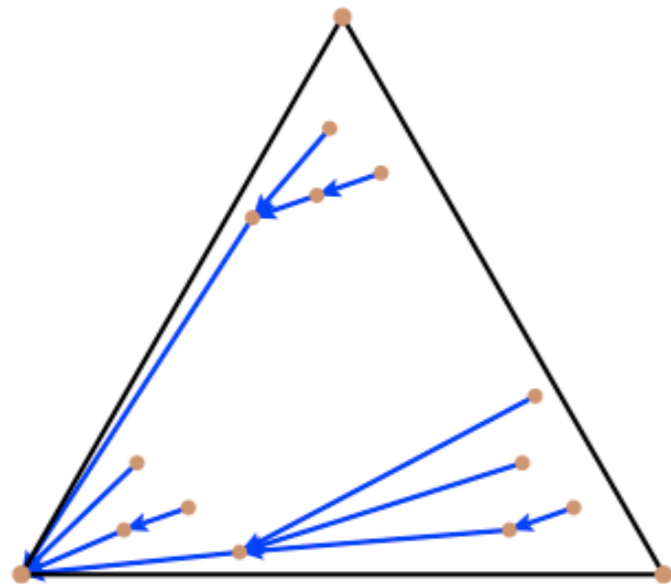
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



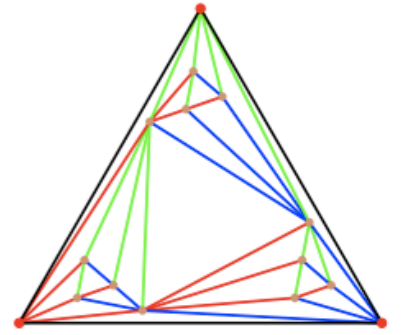
Greedy Drawings



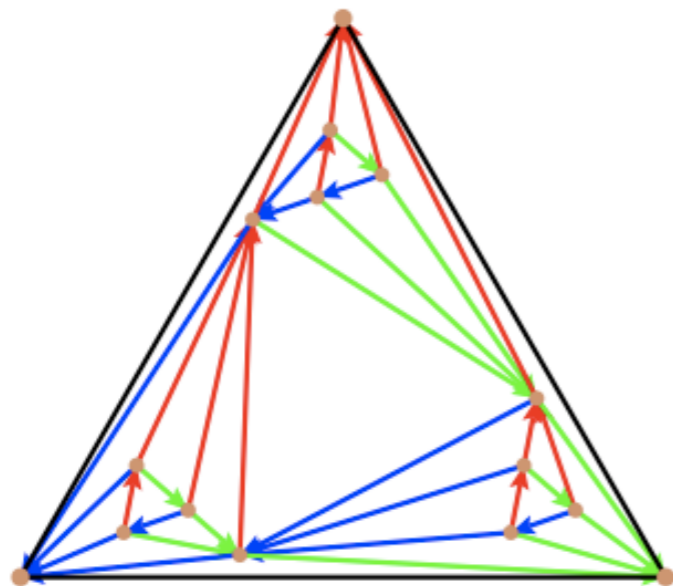
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



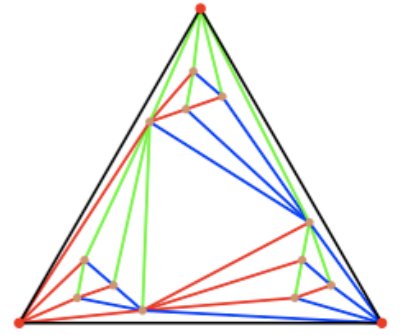
Greedy Drawings



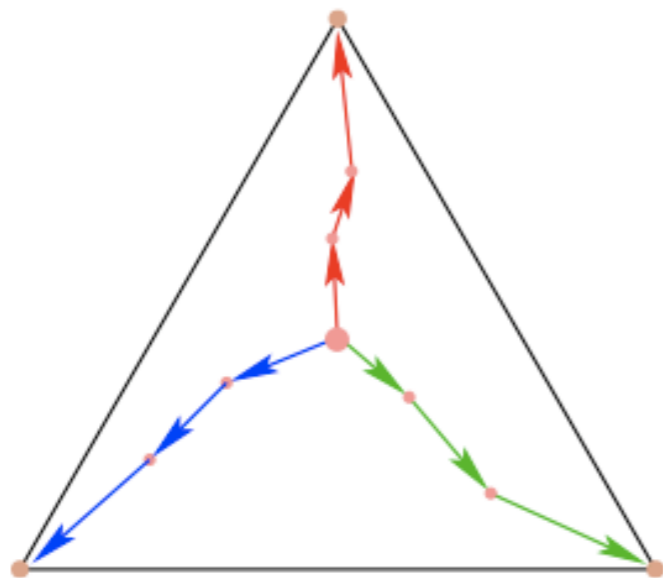
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



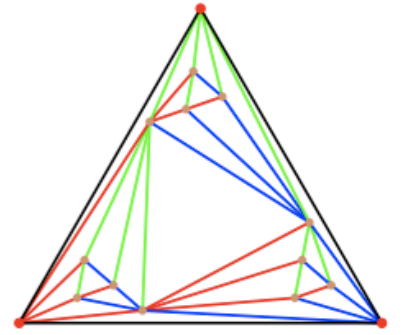
Greedy Drawings



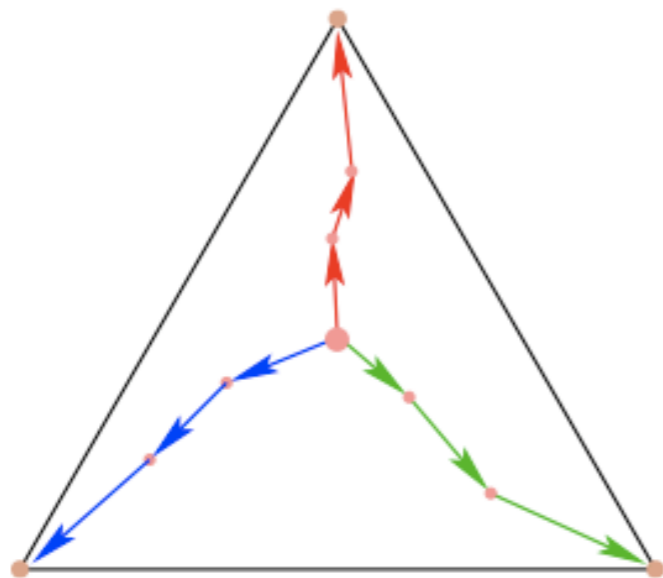
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



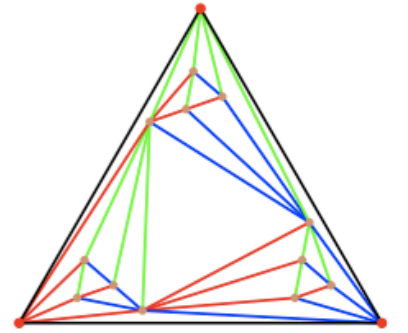
Greedy Drawings



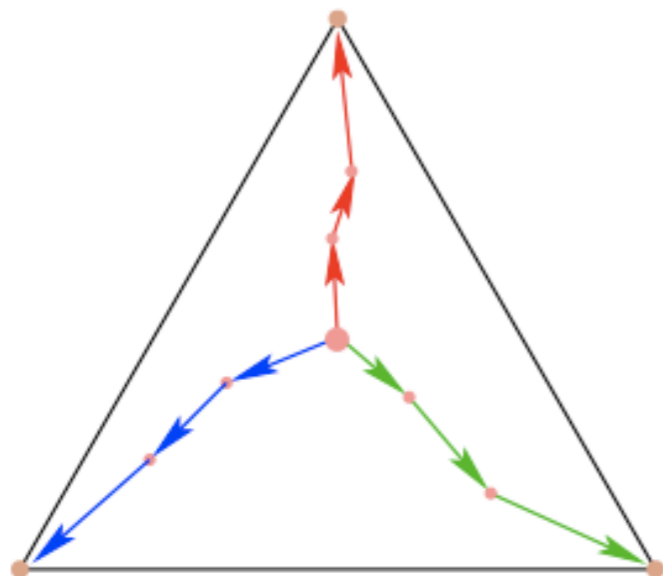
- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



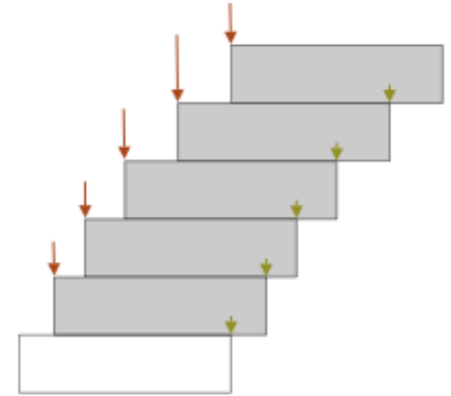
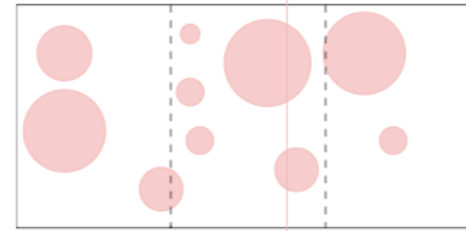
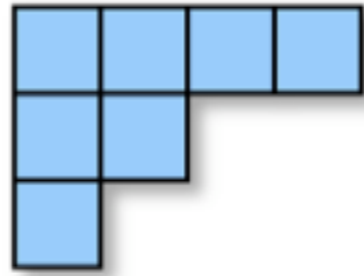
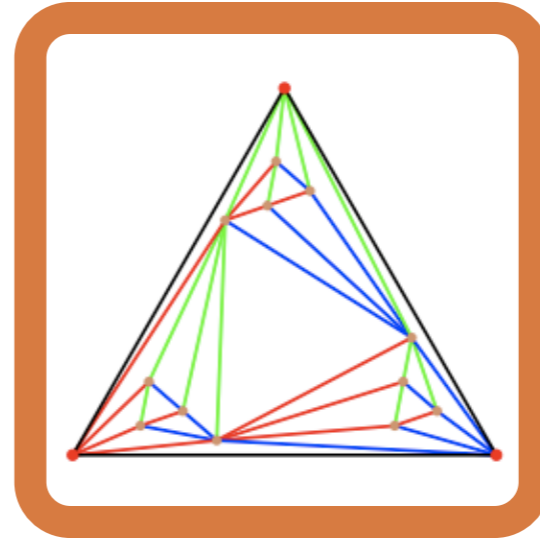
Greedy Drawings

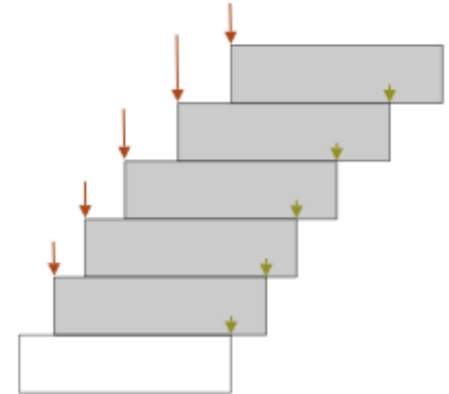
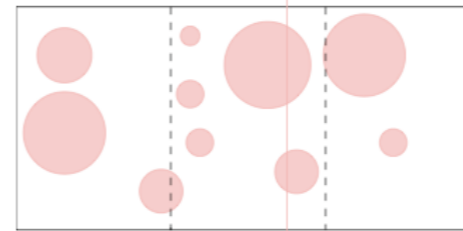
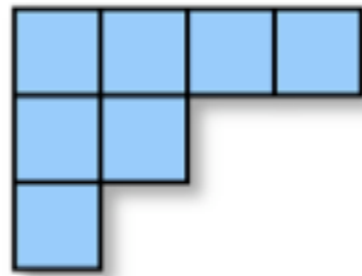
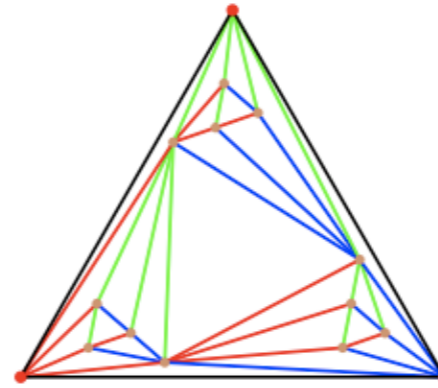


- Toolbox for drawing planar graphs:
 - Rubber band embedding [T]
 - Canonical ordering [dFPP]
 - Schnyder realizer [S]
- “Good” weights always exist



- 1) Algorithmic version
- 2) Full Conjecture

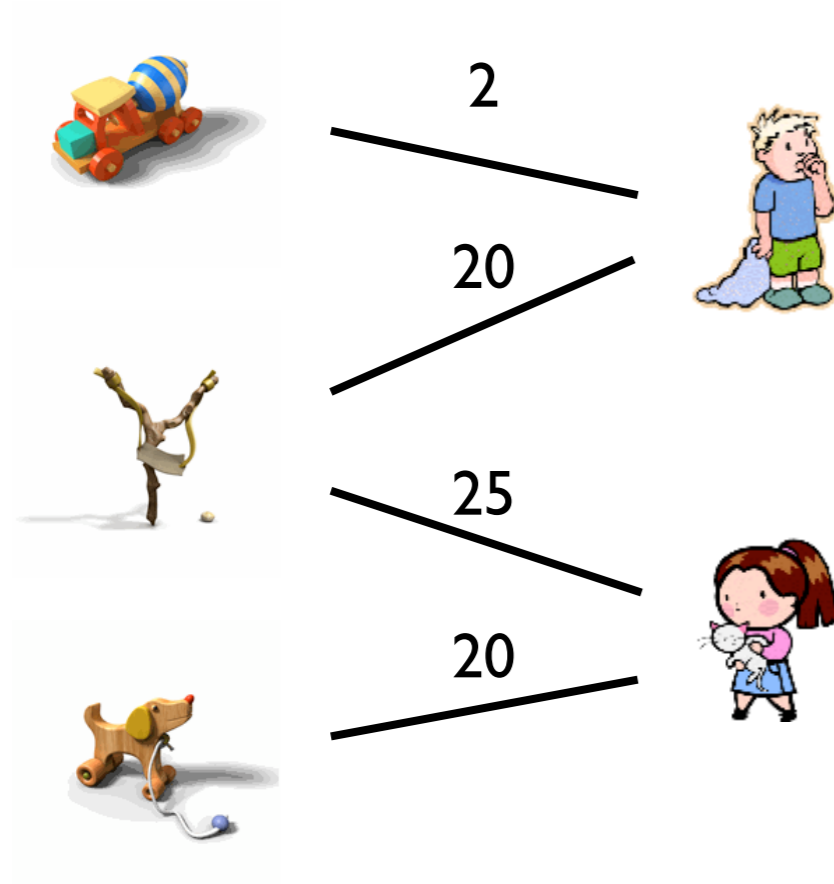




On Allocations that Maximize Fairness

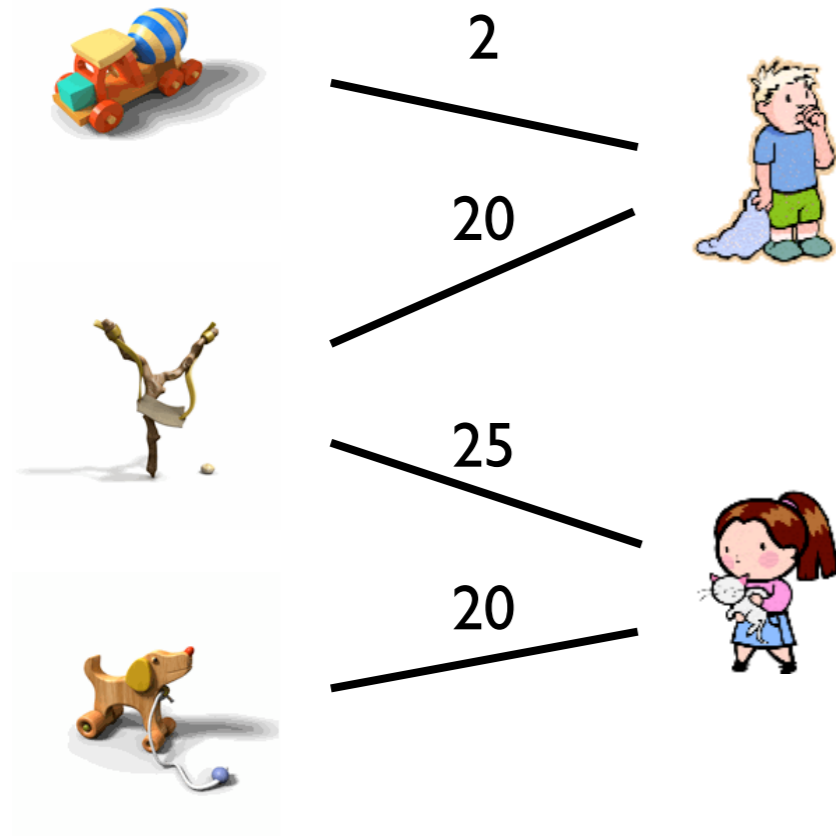
by U. Feige

The Santa Clause Problem



Maximize happiness of
least happy kid

The Santa Clause Problem

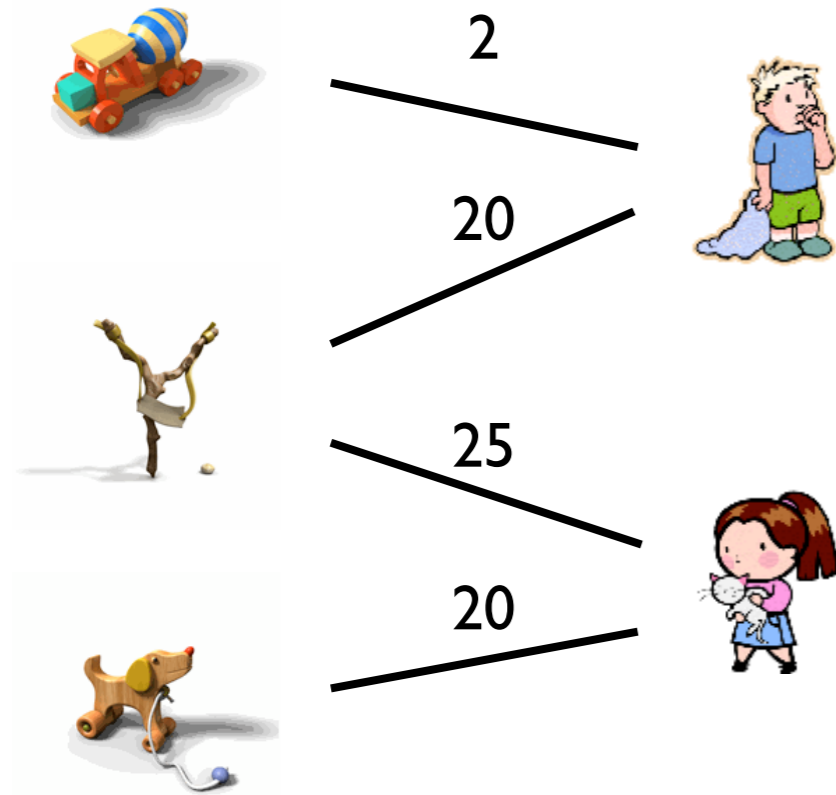


maximize T

$$\sum_j x_{ij} = 1 \quad \text{for all } i$$

$$\sum_i \min(p_{ij}, T) x_{ij} > T \quad \text{for all } j$$

The Santa Clause Problem



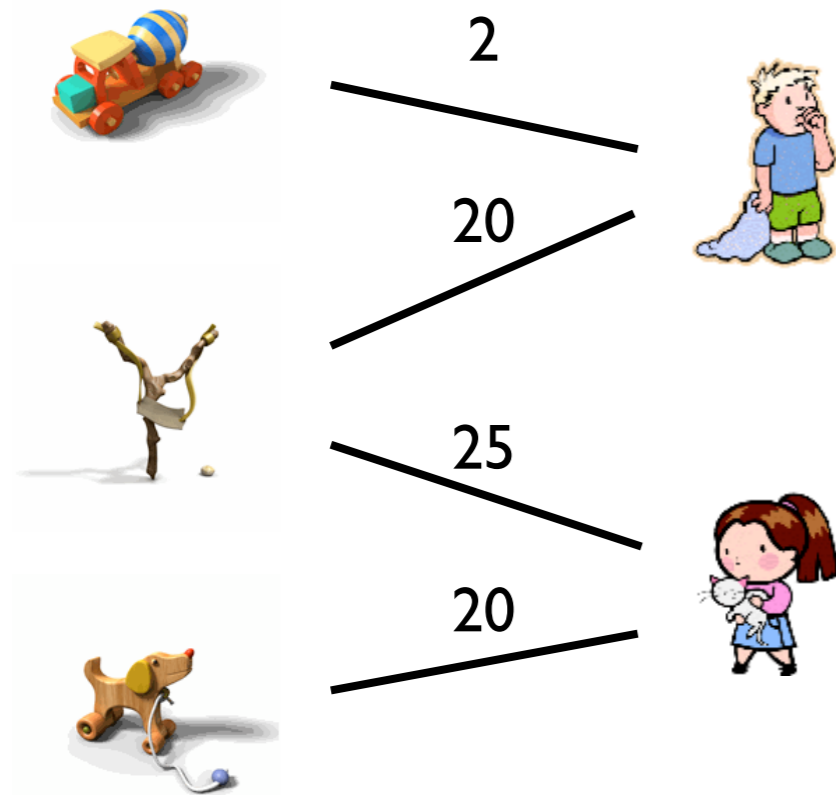
maximize T

$$\sum_j x_{ij} = 1 \quad \text{for all } i$$

$$\sum_i \min(p_{ij}, T) x_{ij} > T \quad \text{for all } j$$

- Stronger LP [BS]
- General case
- Restricted case p_{ij} in $\{0, p_i\}$

The Santa Clause Problem



maximize T

$$\sum_j x_{ij} = 1 \quad \text{for all } i$$

$$\sum_i \min(p_{ij}, T) x_{ij} > T \quad \text{for all } j$$

- Stronger LP [BS]
- General case
- Restricted case p_{ij} in $\{0, p_i\}$

$O(I)$ integrality gap for restricted case

The Santa Clause Problem



maximize T

$$x_{j,C} = 0 \text{ if } p(j,C) < T$$

$$\sum_C x_{j,C} = 1 \text{ for all } j$$

$$\sum_{j,C \mid i \text{ in } C} x_{j,C} = 1 \text{ for all } i$$

The Santa Clause Problem



- Configuration LP

maximize T

$$x_{j,C} = 0 \text{ if } p(j,C) < T$$

$$\sum_C x_{j,C} = 1 \text{ for all } j$$

$$\sum_{j,C \mid i \text{ in } C} x_{j,C} = 1 \text{ for all } i$$

The Santa Clause Problem



- Configuration LP
- Hard instances of [BS]

maximize T

$$x_{j,C} = 0 \text{ if } p(j,C) < T$$

$$\sum_C x_{j,C} = 1 \text{ for all } j$$

$$\sum_{j,C \mid i \text{ in } C} x_{j,C} = 1 \text{ for all } i$$

The Santa Clause Problem



- Configuration LP
- Hard instances of [BS]
- IG is at most 5, but still not constructive [AFS]

maximize T

$$x_{j,C} = 0 \text{ if } p(j,C) < T$$

$$\sum_C x_{j,C} = 1 \text{ for all } j$$

$$\sum_{j,C \mid i \text{ in } C} x_{j,C} = 1 \text{ for all } i$$

The Santa Clause Problem



- Configuration LP
- Hard instances of [BS]
- IG is at most 5, but still not constructive [AFS]

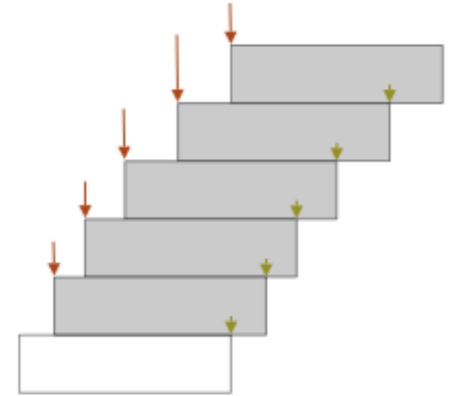
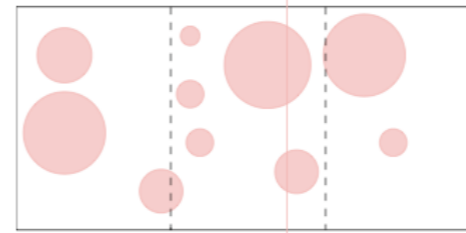
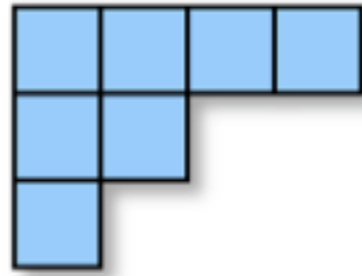
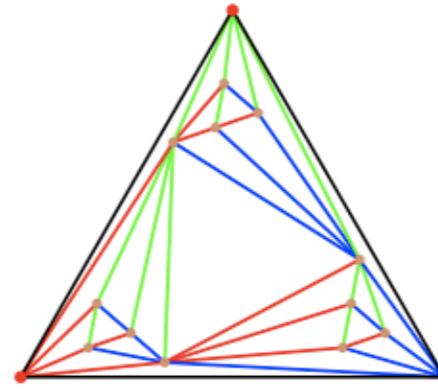
maximize T

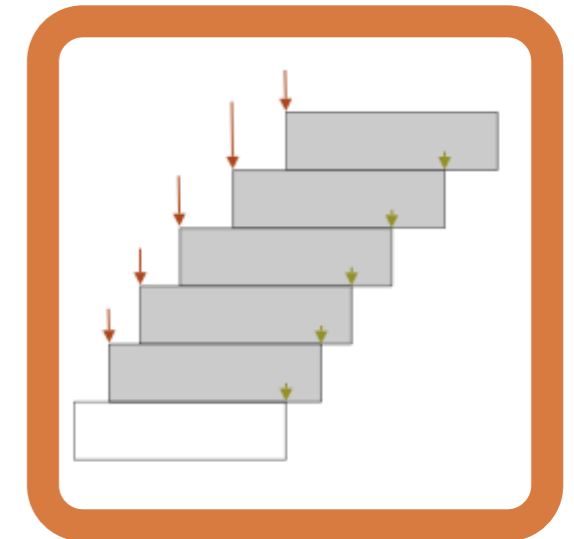
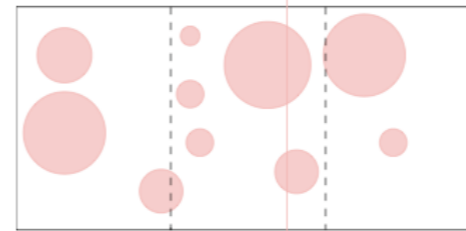
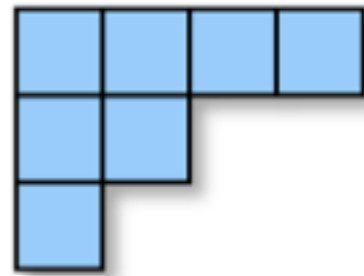
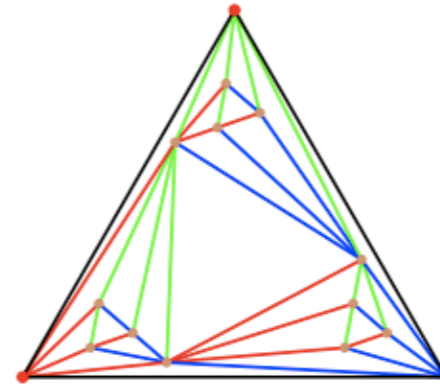
$$x_{j,C} = 0 \text{ if } p(j,C) < T$$

$$\sum_C x_{j,C} = 1 \text{ for all } j$$

$$\sum_{j,C \mid i \text{ in } C} x_{j,C} = 1 \text{ for all } i$$

Algorithmic
version

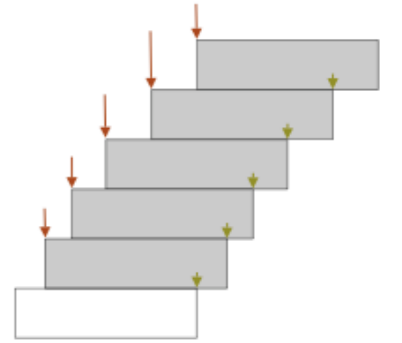




Maximum Overhang

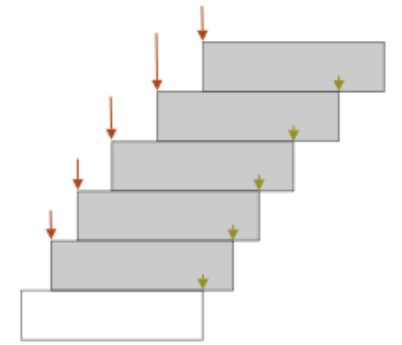
by M. Paterson, Y. Peres, M. Thorup, P. Winkler, and U. Zwick

Maximum Overhang

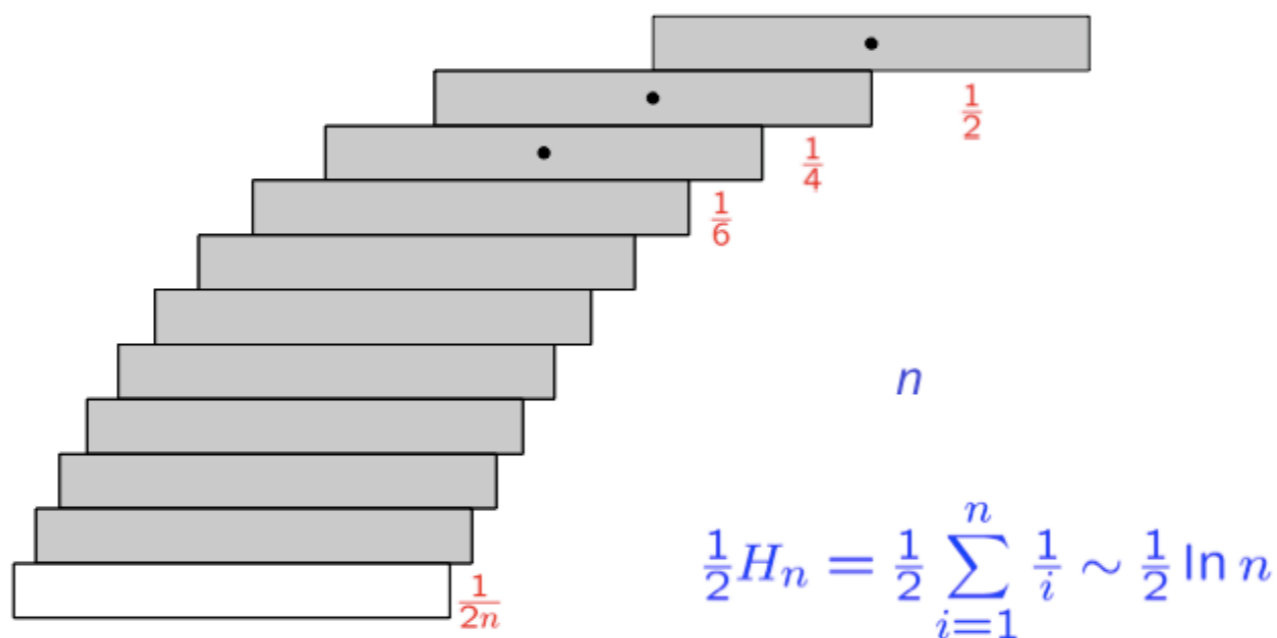


- Given n blocks of unit length, what is the maximum overhang that can be achieved?
- [Folk] If we use a stack then exactly $\frac{1}{2} H_n$
- [PZ] If we use a wall then overhang is $\Omega(n^{1/3})$

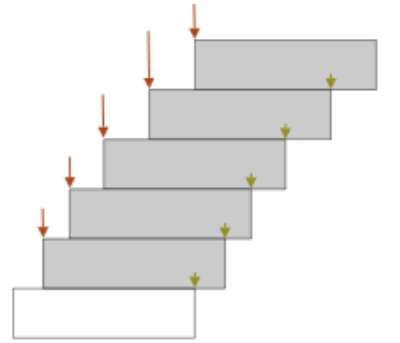
Maximum Overhang



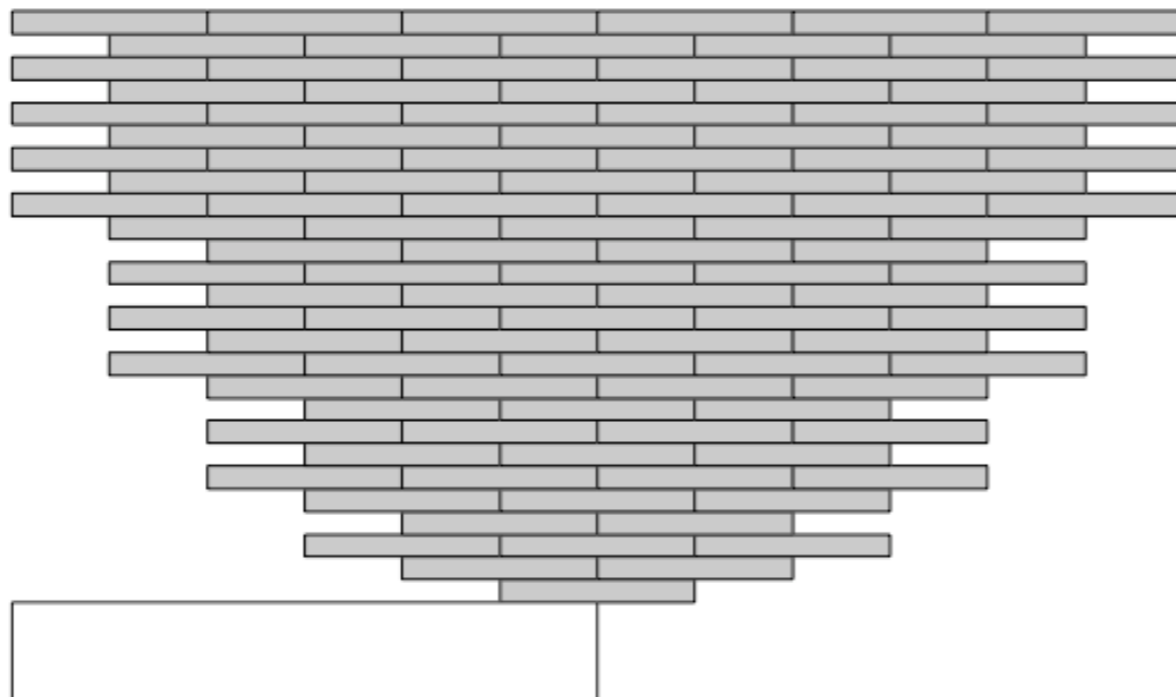
- Given n blocks of unit length, what is the maximum overhang that can be achieved?
- [Folk] If we use a stack then exactly $\frac{1}{2} H_n$
- [PZ] If we use a wall then overhang is $\Omega(n^{1/3})$



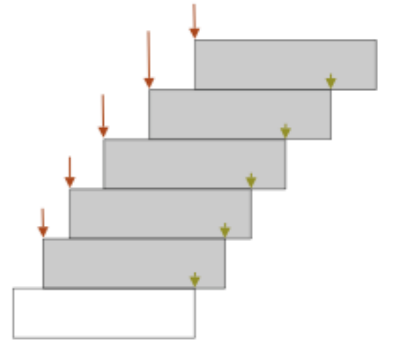
Maximum Overhang



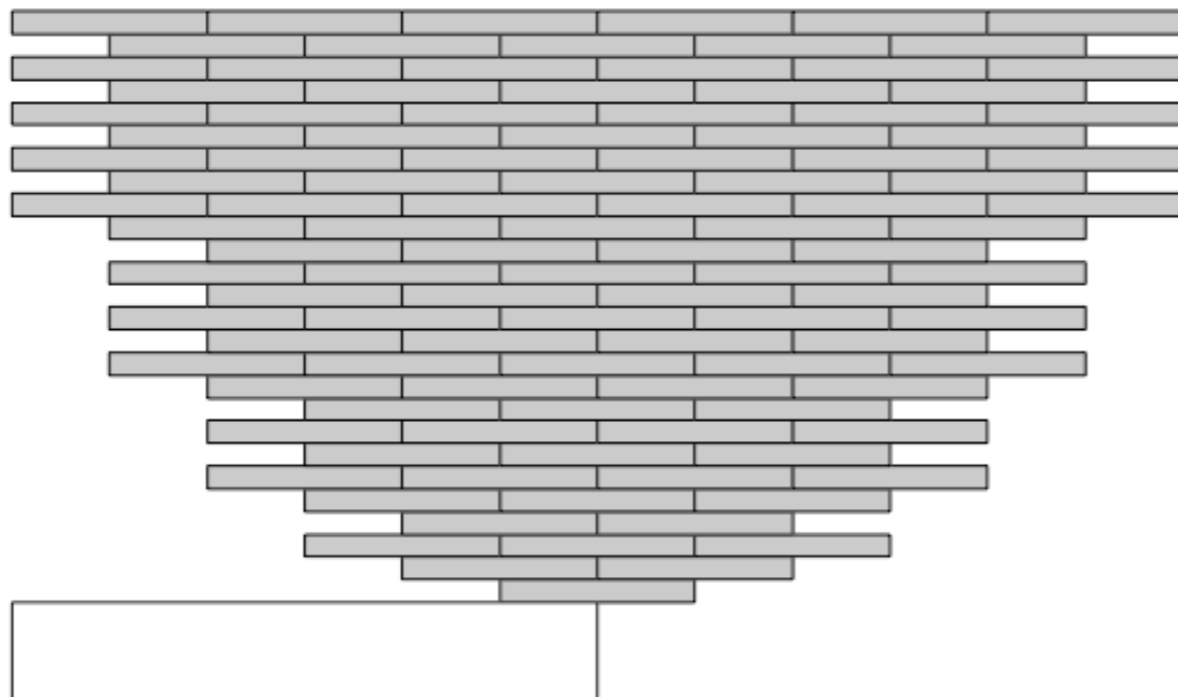
- Given n blocks of unit length, what is the maximum overhang that can be achieved?
- [Folk] If we use a stack then exactly $\frac{1}{2} H_n$
- [PZ] If we use a wall then overhang is $\Omega(n^{1/3})$



Maximum Overhang

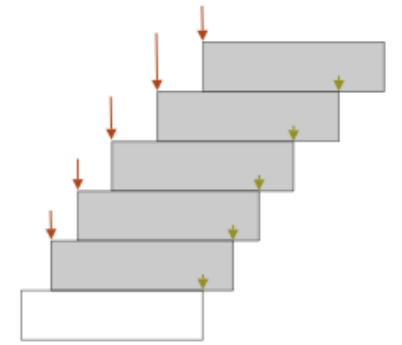


- Given n blocks of unit length, what is the maximum overhang that can be achieved?
- [Folk] If we use a stack then exactly $\frac{1}{2} H_n$
- [PZ] If we use a wall then overhang is $\Omega(n^{1/3})$

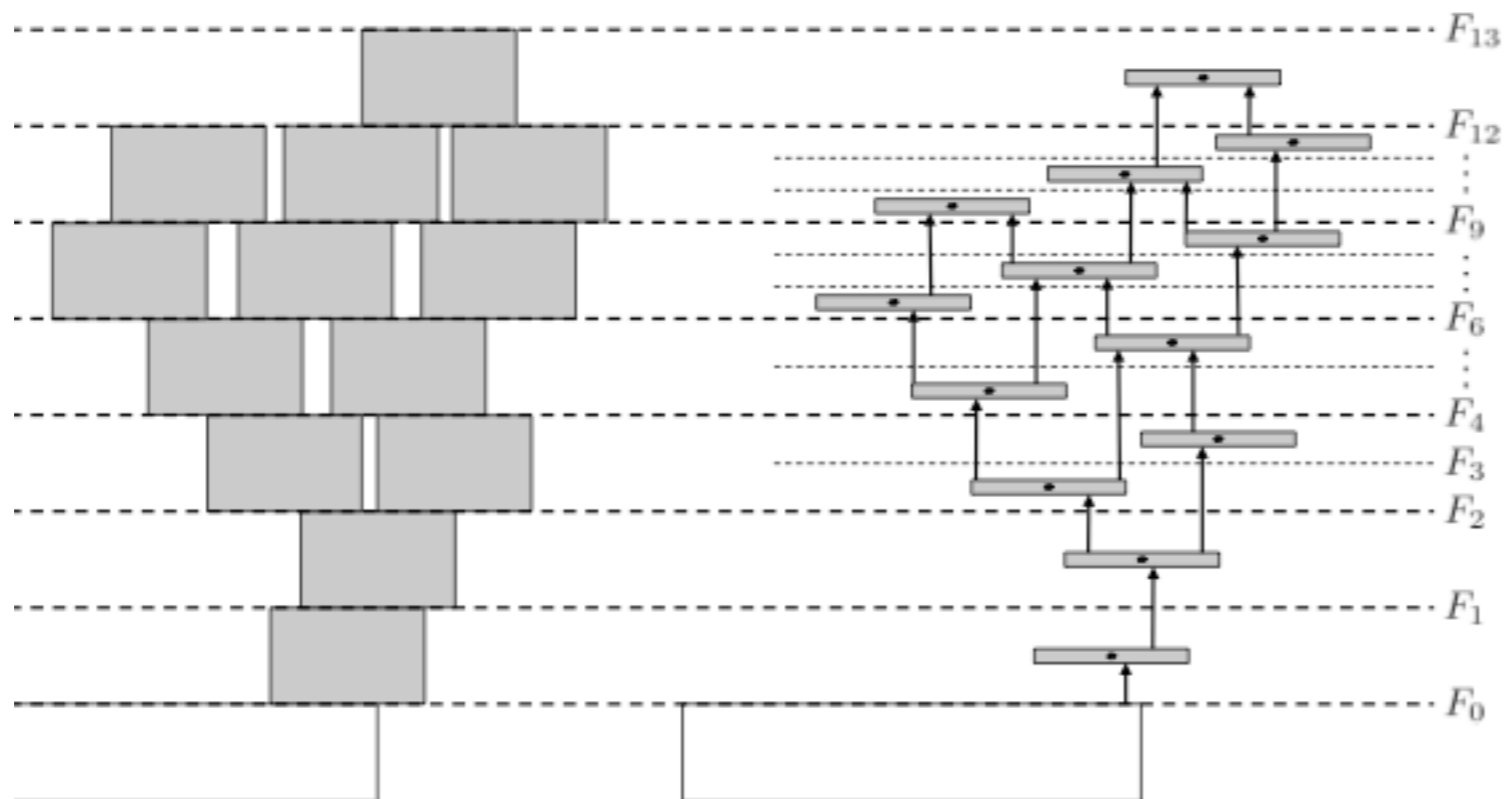


Max. overhang
is $O(n^{1/3})$

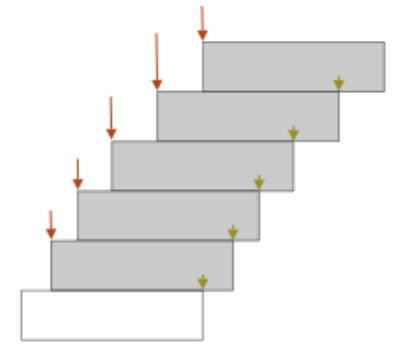
Maximum Overhang



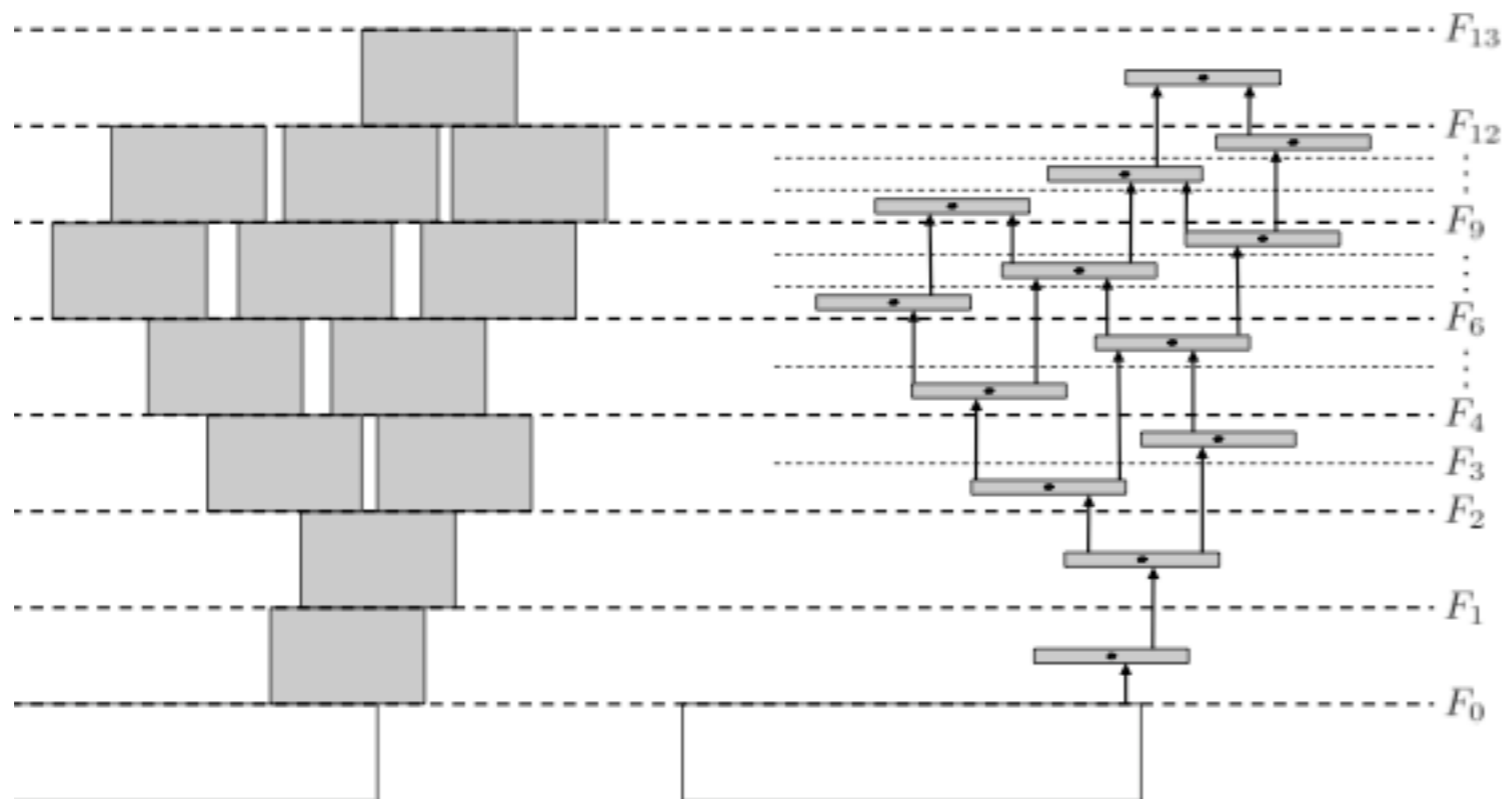
- Think of blocks as operations that distribute forces
- Want move l unit of force to the right
- Overhang of $6n^{1/3}$ requires more than n operations



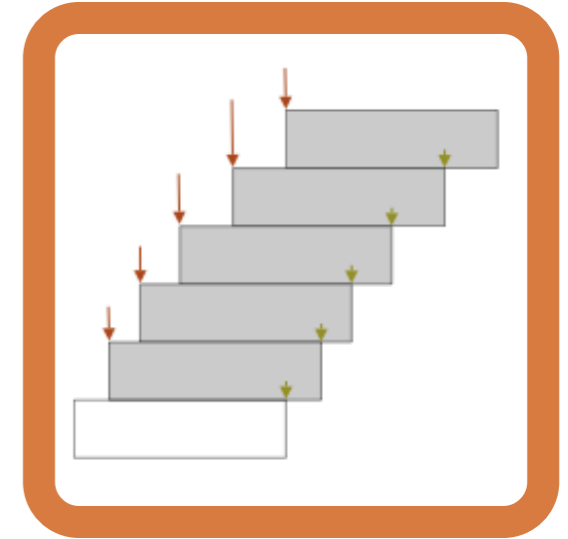
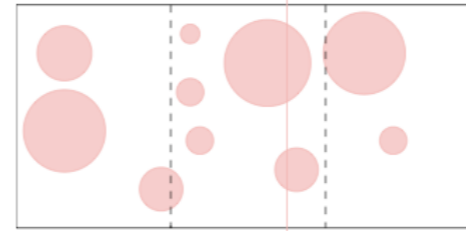
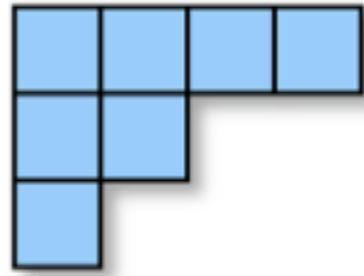
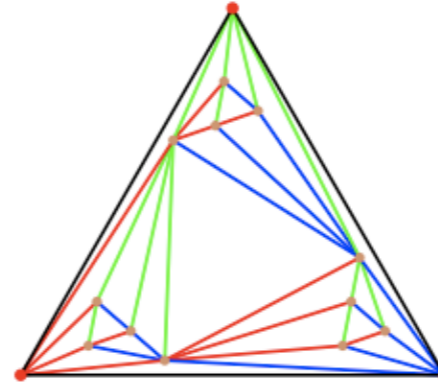
Maximum Overhang

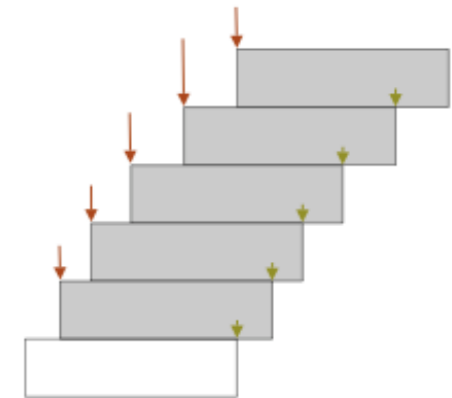
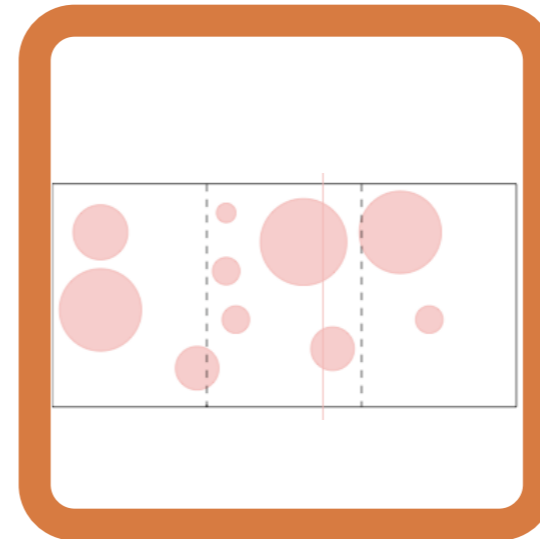
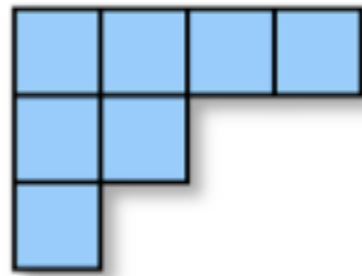
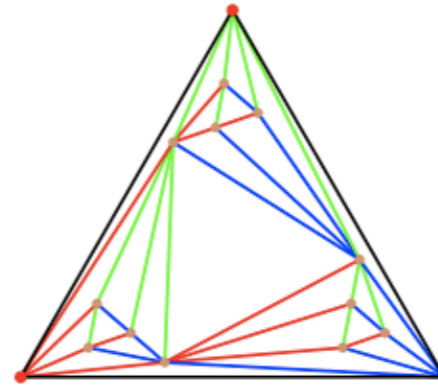


- Think of blocks as operations that distribute forces
- Want move 1 unit of force to the right
- Overhang of $6n^{1/3}$ requires more than n operations



- 1) Small gap
- 2) Friction

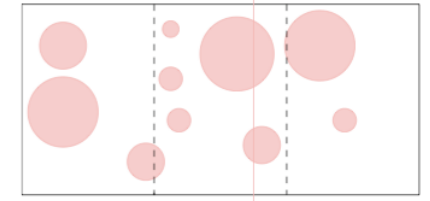




On Clustering to Minimize the Sum of Radii

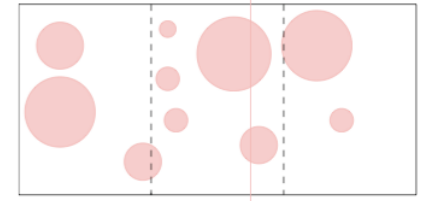
by M. Gibson, G. Kanade, E. Krohn, I. Pirwami, and K. Varadarajan

Clustering to minimize sum of radii



- Given a set of points in Euclidean space, find **k** cluster to minimize sum of cluster radii
- NP-hard when minimizing powers of radii [A et. al]

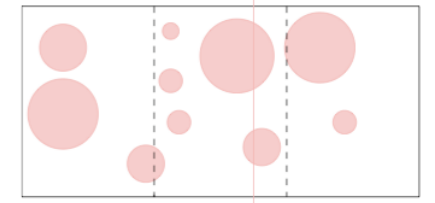
Clustering to minimize sum of radii



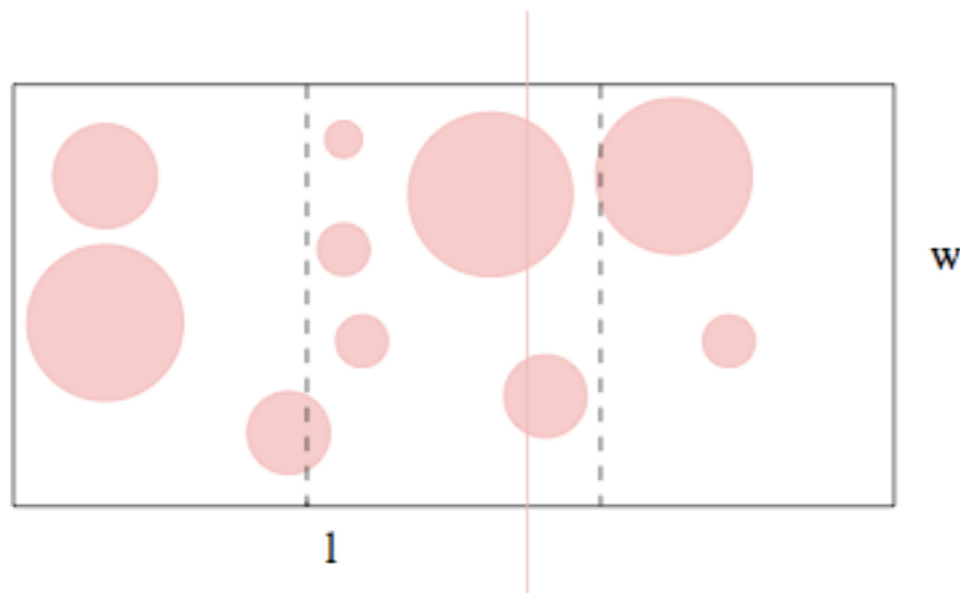
- Given a set of points in Euclidean space, find k clusters to minimize sum of cluster radii
- NP-hard when minimizing powers of radii [A et. al]

Can be solved
in $n^{O(k)}$ time

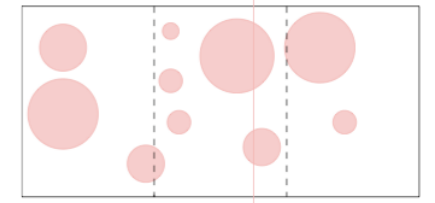
Clustering to minimize sum of radii



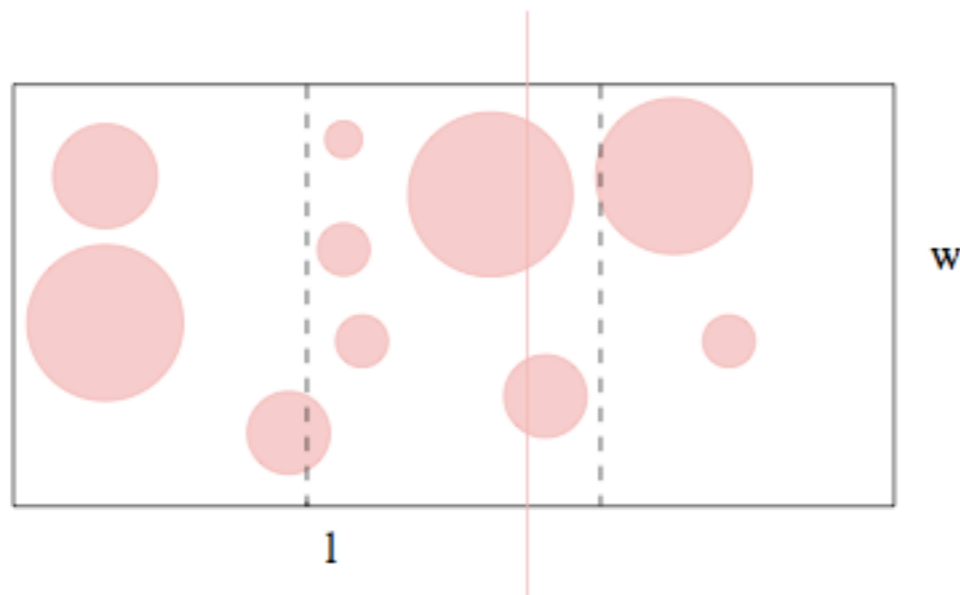
- ⦿ Problem is separable \Rightarrow Dynamic programming
- ⦿ Two issues must be dealt with:
 - Critical and canonical lines
 - How to manage constraints from other subproblems



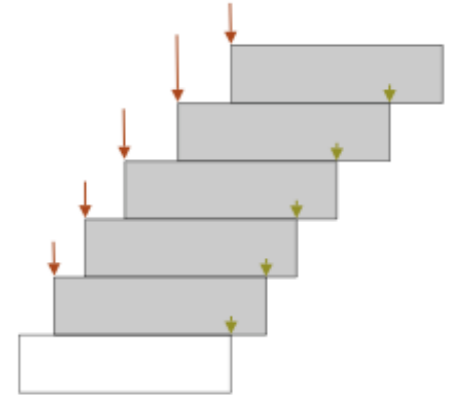
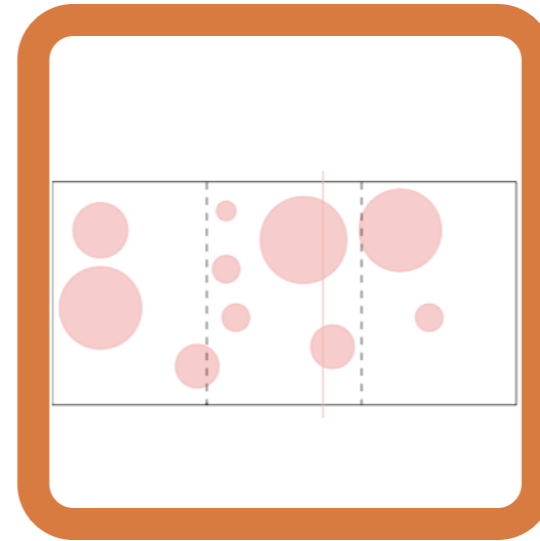
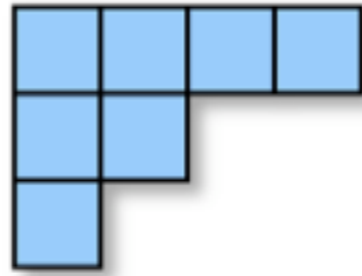
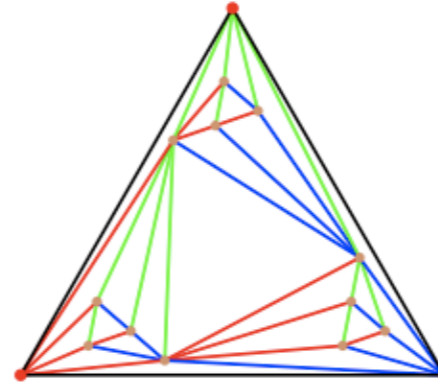
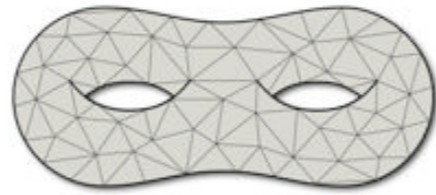
Clustering to minimize sum of radii

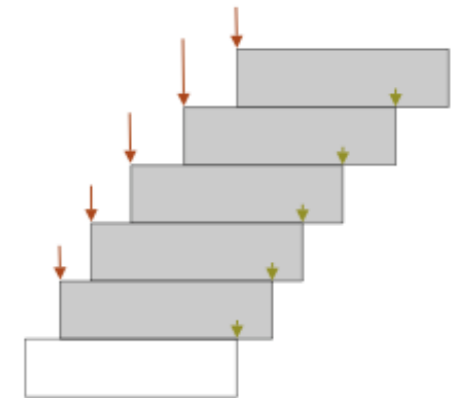
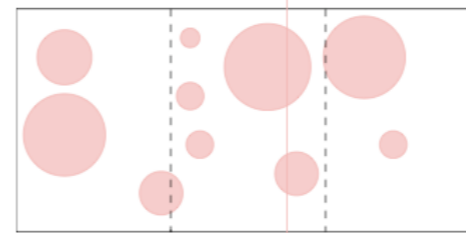
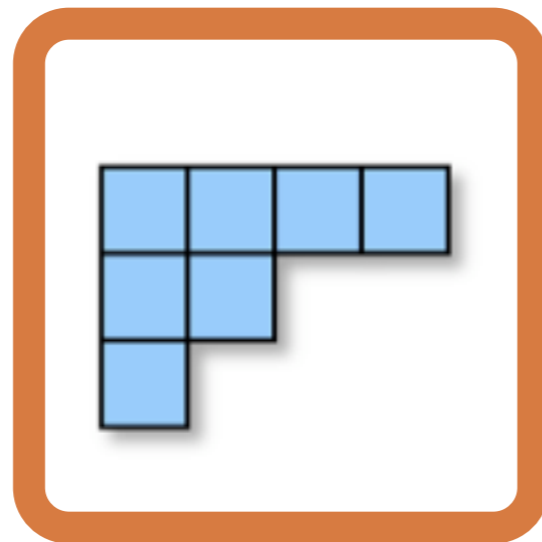
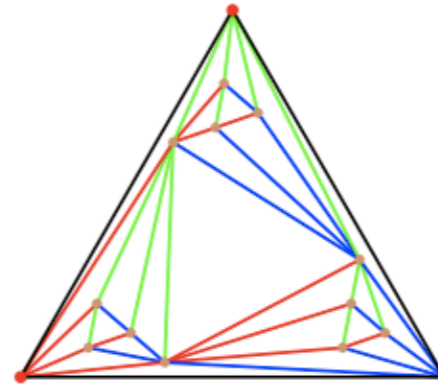


- ⦿ Problem is separable \Rightarrow Dynamic programming
- ⦿ Two issues must be dealt with:
 - Critical and canonical lines
 - How to manage constraints from other subproblems



Improve running
time

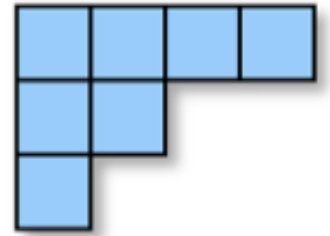




Matroid Intersection, Pointer Chasing and Young's Seminormal Representation of S_n

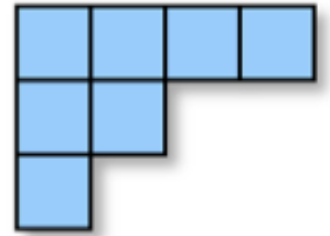
by N. Harvey

Matroid Intersection Lowerbounds



- Matroids are well-behaved optimization problems
- Defined by (E, I) where I is a subset of 2^E , given implicitly by an oracle
- Matroid intersection of (E, I_1) and (E, I_2) can be solved in $O(nr^{1/2})$ queries [C]

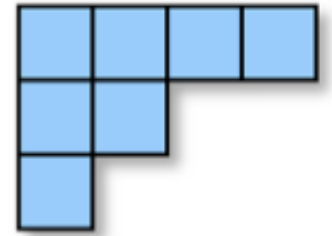
Matroid Intersection Lowerbounds



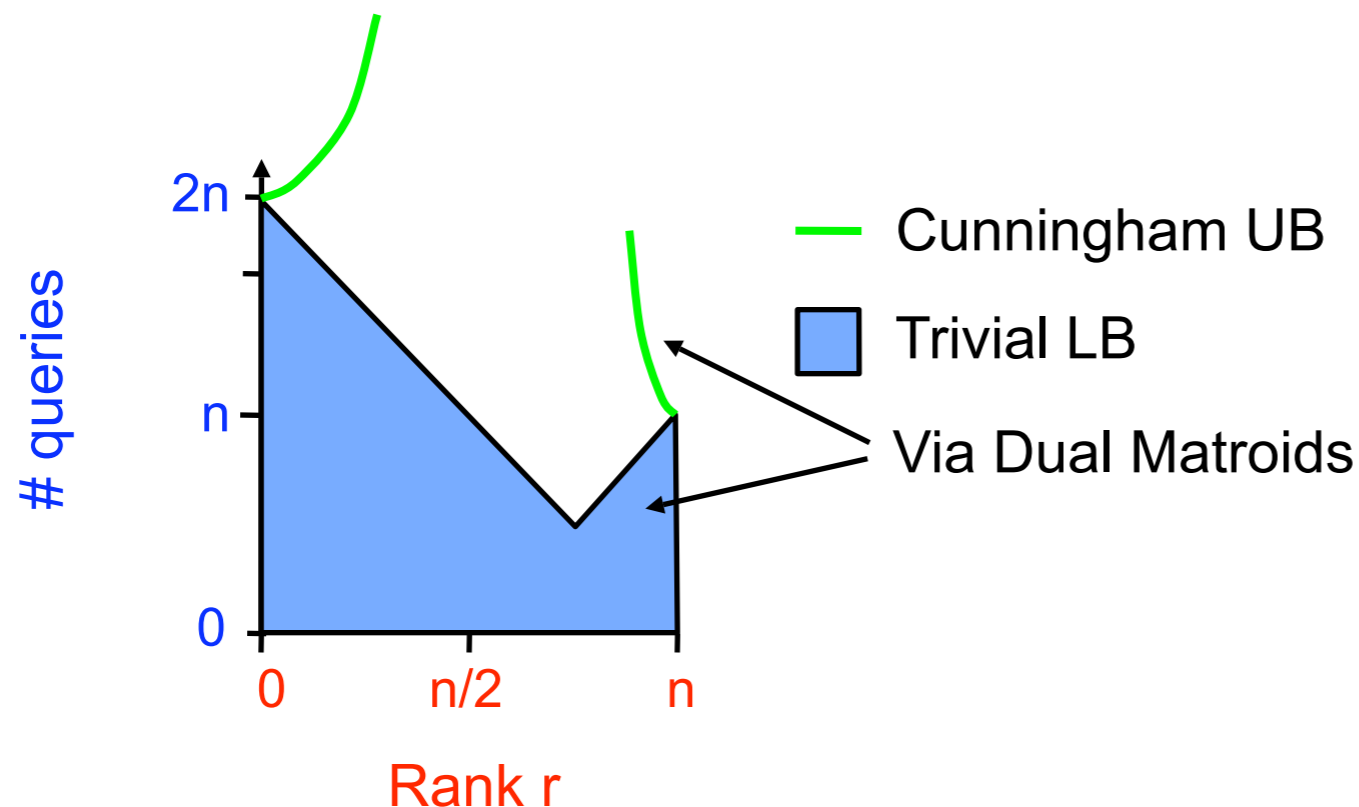
- Matroids are well-behaved optimization problems
- Defined by (E, I) where I is a subset of 2^E , given implicitly by an oracle
- Matroid intersection of (E, I_1) and (E, I_2) can be solved in $O(nr^{1/2})$ queries [C]

First nontrivial
lowerbound

Matroid Intersection Lowerbounds

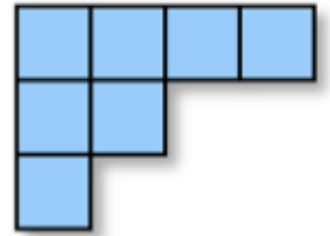


- Matroids are well-behaved optimization problems
- Defined by (E, I) where I is a subset of 2^E , given implicitly by an oracle
- Matroid intersection of (E, I_1) and (E, I_2) can be solved in $O(nr^{1/2})$ queries [C]

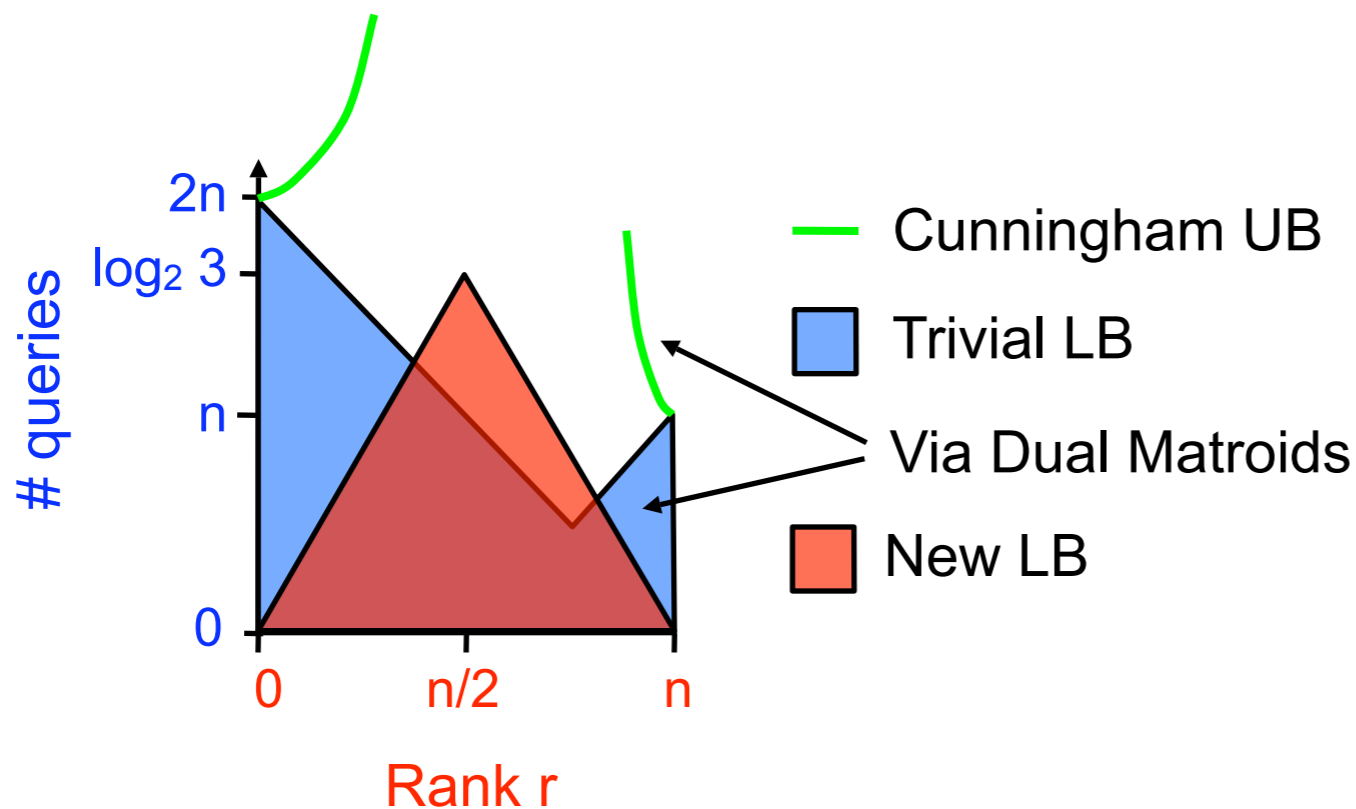


First nontrivial lowerbound

Matroid Intersection Lowerbounds

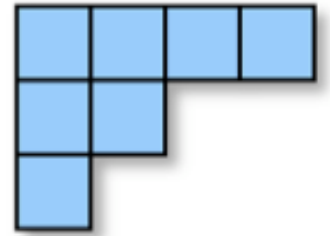


- Matroids are well-behaved optimization problems
- Defined by (E, I) where I is a subset of 2^E , given implicitly by an oracle
- Matroid intersection of (E, I_1) and (E, I_2) can be solved in $O(nr^{1/2})$ queries [C]

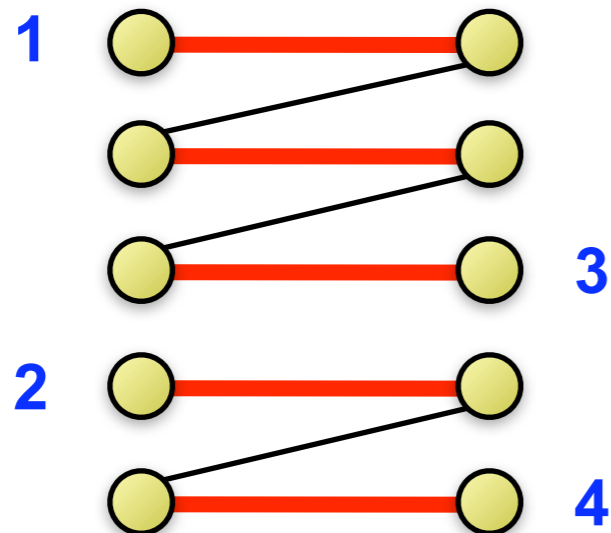


First nontrivial lowerbound

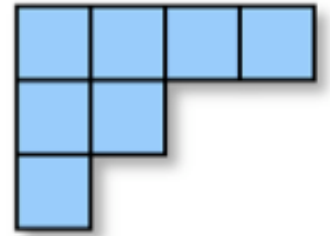
Matroid Intersection Lowerbounds



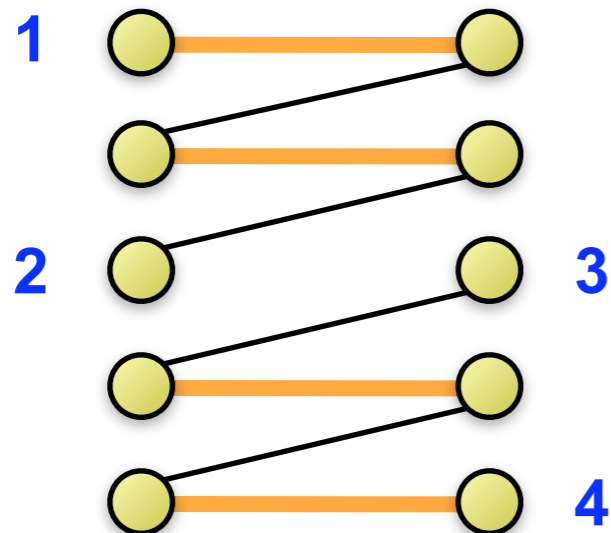
- Perfect matching in **2-almost-regular** bipartite graphs
- Pointer chasing problem
- Communication complexity: Compute rank of communication matrix and apply **[MS]**
- Diagonalize matrix via basis change



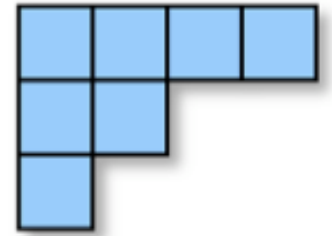
Matroid Intersection Lowerbounds



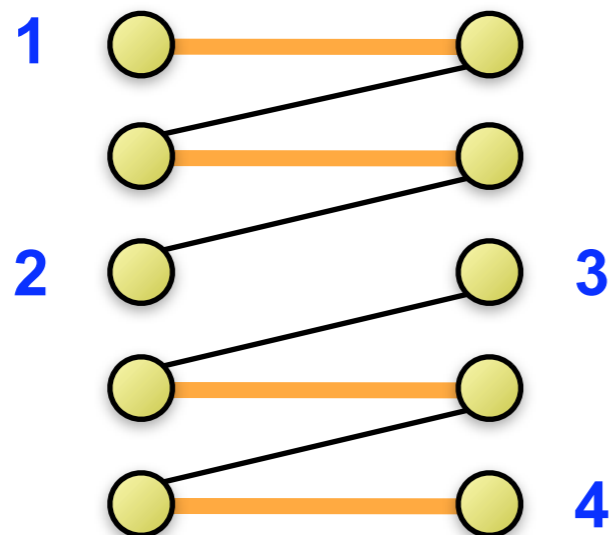
- Perfect matching in **2-almost-regular** bipartite graphs
- Pointer chasing problem
- Communication complexity: Compute rank of communication matrix and apply **[MS]**
- Diagonalize matrix via basis change



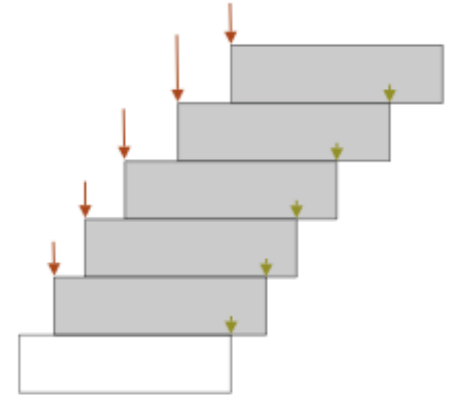
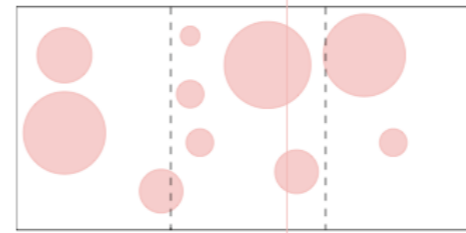
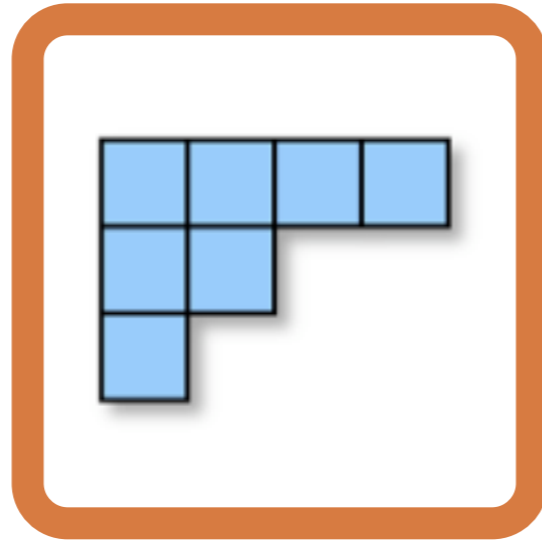
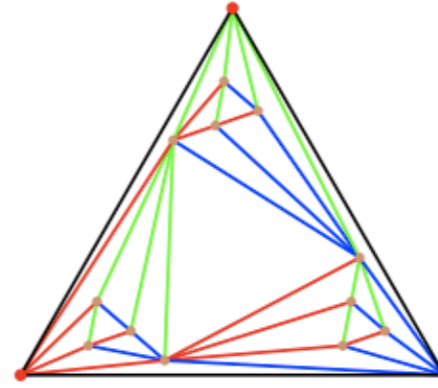
Matroid Intersection Lowerbounds

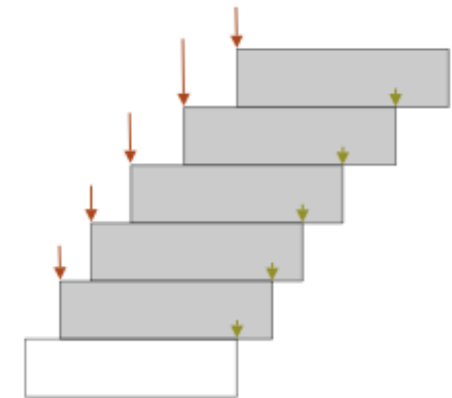
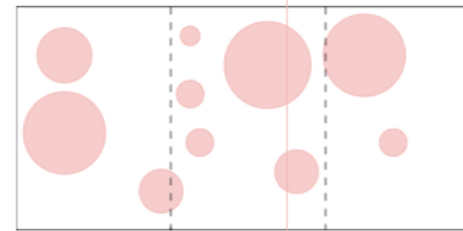
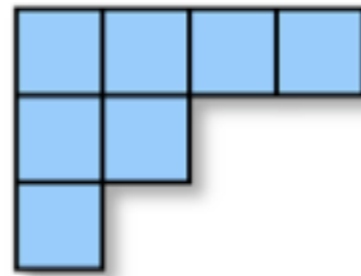
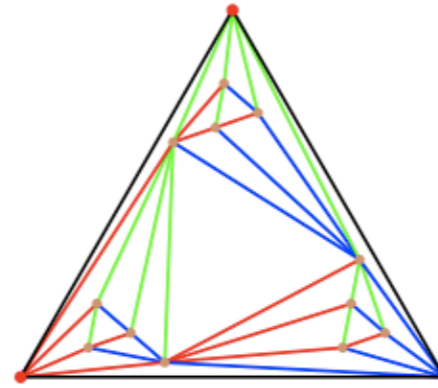


- Perfect matching in **2-almost-regular** bipartite graphs
- Pointer chasing problem
- Communication complexity: Compute rank of communication matrix and apply **[MS]**
- Diagonalize matrix via basis change



1) more “complex” matroids
2) avoid CC





The Hiring Problem and Lake Wobegon Strategies

by Broder, Kirsch, Kumar, Mitzenmacher, Upfal, and Vassilvitskii

The Hiring Problem



- Variation of the secretary problem
- First employee in $(0, 1)$ is fixed
- Applicants come from $U(0, 1)$
- Lake Wabegon strategy: Hire above *average*
 - mean
 - median

The Hiring Problem

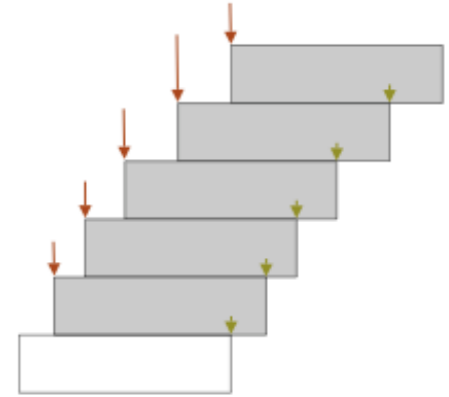
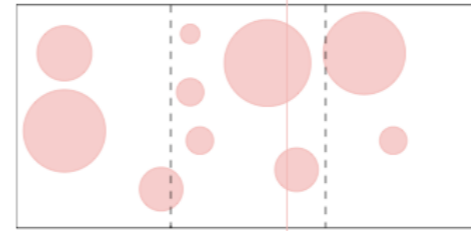
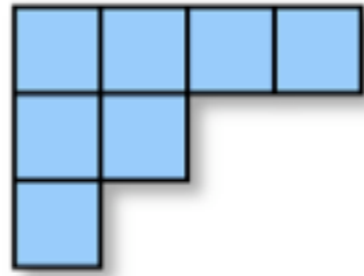
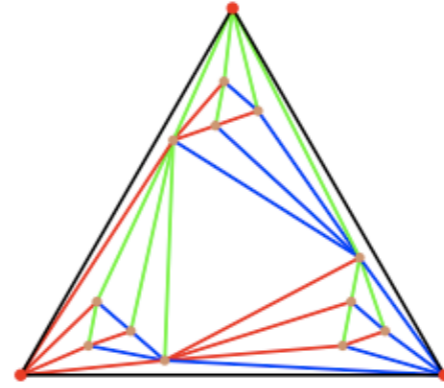


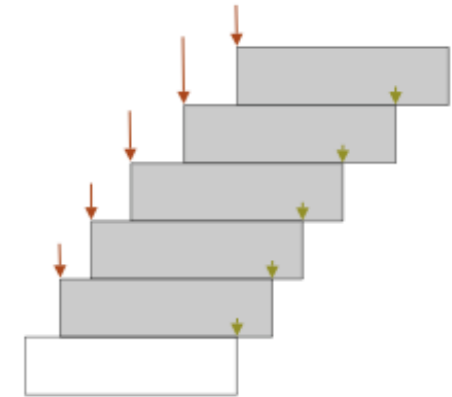
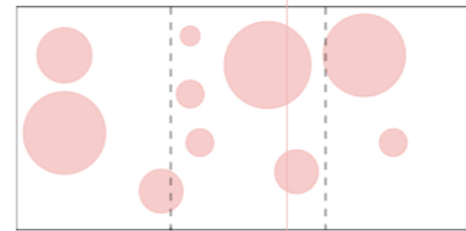
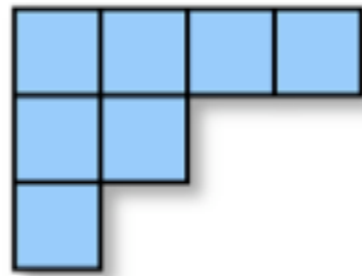
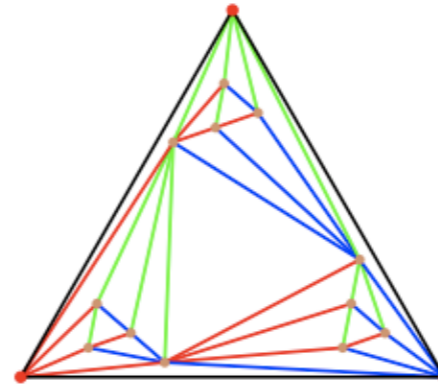
- Variation of the secretary problem
- First employee in $(0, 1)$ is fixed
- Applicants come from $U(0, 1)$
- Lake Wabegon strategy: Hire above *average*
 - mean
 - median

1) mean \neq median
2) first employee's
influence

Acknowledgements

- Raghavan S. Dhandapani, NYU
- Jeff Erickson, UIUC
- Nick Harvey, MIT
- Gaurav Kanade, UIOWA
- Uri Zwick, TAU





Thanks
for your
attention!