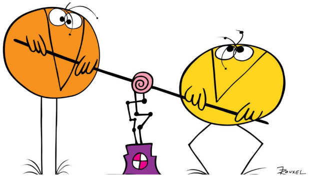
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

How We Solve CG:SHOP Problems

Guilherme D. da Fonseca – LIS, Aix-Marseille Université
Yan Gerard – LIMOS, Université Clermont Auvergne
+ All Shadoks team members



CG:SHOP Competition

Shadoks

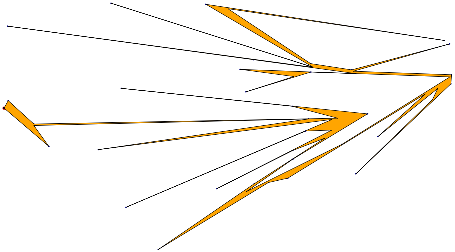
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

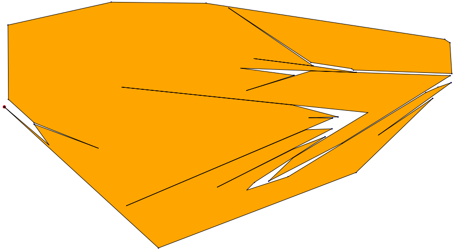
- Part of SoCG (International Symposium on Computational Geometry)
- Annual event that started in 2018–2019
- Hard geometric optimization problems
- Different problem each year
- ~ 200 instances given
- ~ 3 months to compute solutions
- Send our solutions (not the code)
- Score based on the quality of the solutions
- Top teams invited to publish in SoCG proceedings and ACM Journal of Experimental Algorithmics or Computing in Geometry and Topology

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Minimum (or Maximum) Area Polygon:

- Input: A set of points $S \subset \mathbb{R}^2$
- Output: A simple polygon with vertex set S
- Goal: Minimize (or maximize) the area





- Related to Euclidean TSP
- Two categories: minimization, maximization
- We got 2nd place
- Techniques: greedy and local search

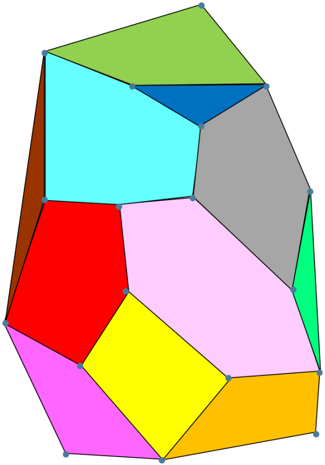
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Minimum Convex Partition:

- Input: A set of points $S \subset \mathbb{R}^2$
- Output: A simple partition of the convex hull of S into convex regions with vertex set S
- Goal: Minimize the number of regions

- We got 4th place
- Used integer programming



11 convex regions

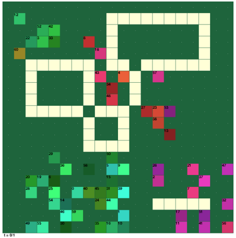
CG:SHOP 2021 (Robots)

Coordinated Motion Planning:

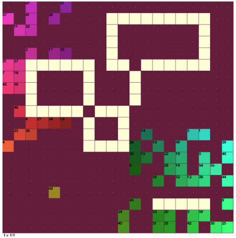
- Input: Sets $S, T \subset \mathbb{Z}^2$ of start and target locations for n robots and possibly a set of obstacles
- Output: A sequence of movements for all robots from start to target avoiding collisions
- Goal: Minimize the total time (makespan) or the total number of movements (energy)

- 1st place in makespan category,
  3rd place in energy category
- Used storage network and conflict optimizer

Start:



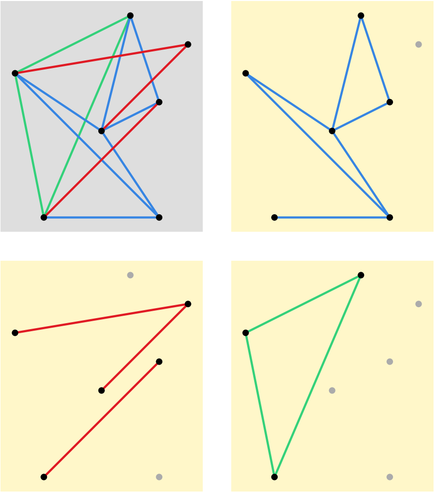
Target:



Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Partition Into Plane Graphs:

- Input: A graph G embedded in the plane with straight edges
- Output: A partition of G into plane graphs
- Goal: Minimize the number of partitions (colors)



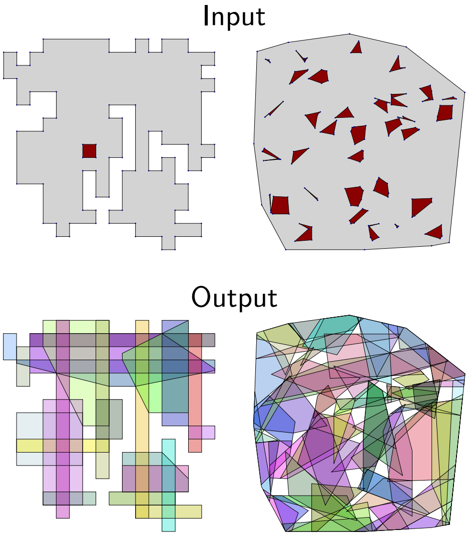
- We won 1st place
- Best solution of all teams to all instances
- Optimal solution to at least 23
- Reused conflict optimizer

CG:SHOP 2023 (ConvexCover)

Convex Covering:

- Input: A polygon with holes P
- Output: A collection of convex polygons whose union is P
- Goal: Minimize the number of convex polygons

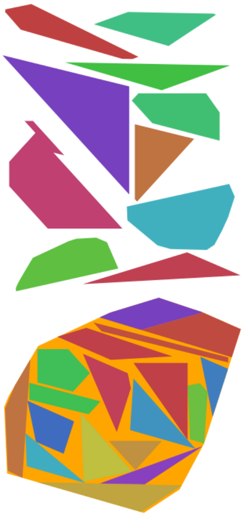
- We won 2nd place
- Best solution among all teams to 128 of 206 instances
- Used integer programming and simulated annealing



Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Knapsack Translational Packing:

- Input: A convex polygon (*container*) and a multi-set of polygons with *values* (*items*)
- Output: A translation of some items that form a packing inside the container
- Goal: Maximize the sum of the values in the output



- We won 1st place
- Used greedy, local search, and integer programming

Initial Solution and Optimized Solution

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Our strategy in every competition:
  1. Find initial feasible solutions
  2. Improve them
- We've seen two types of problems:
  - The quality of the initial solution is irrelevant and the improvements are major
  - The quality of the initial solution is essential and the improvements are minor
- Hard to tell them apart
- If the solution to a subproblem can be incorporated into an existing solution, it is easier to improve

Initial

- Greedy heuristics often produce good initial solutions:
  1. Choose an element to add to the solution
  2. Choose how to incorporate this element
  3. Repeat
- Good data structures to implement it efficiently
- Three examples:
  - `Packing`
  - `Coloring`
  - `MaxPolygon`

Greedy Maximum Area Polygon

Simple idea:

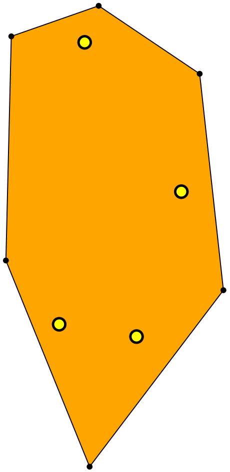
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Simple idea:

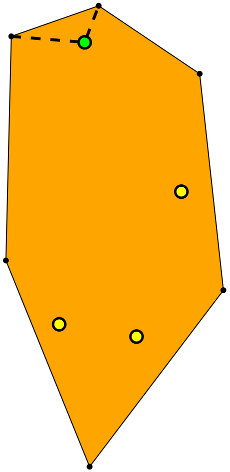
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Simple idea:

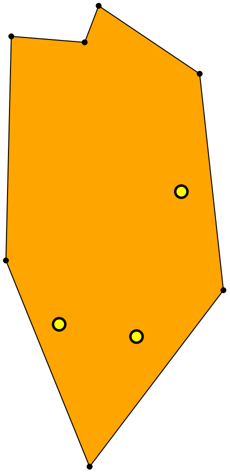
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1p_2q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Simple idea:

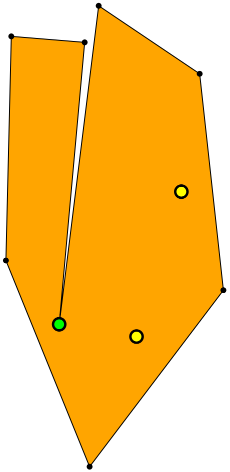
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Simple idea:

- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls

Greedy Maximum Area Polygon

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Simple idea:

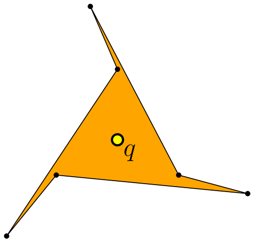
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Simple idea:

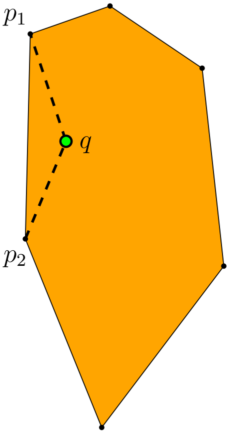
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

- Penalize long edges and favor breaking long edges: $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$ for small α
- Add random noise to circumvent stalls



Greedy Maximum Area Polygon

Simple idea:

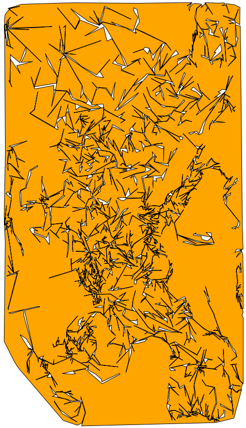
- Start with the convex hull
- Choose the valid vertex-edge pair that decreases the area the least

Issues:

- Long edges block many future vertex-edge pairs
- Stall: May have no valid vertex-edge pair

Solution:

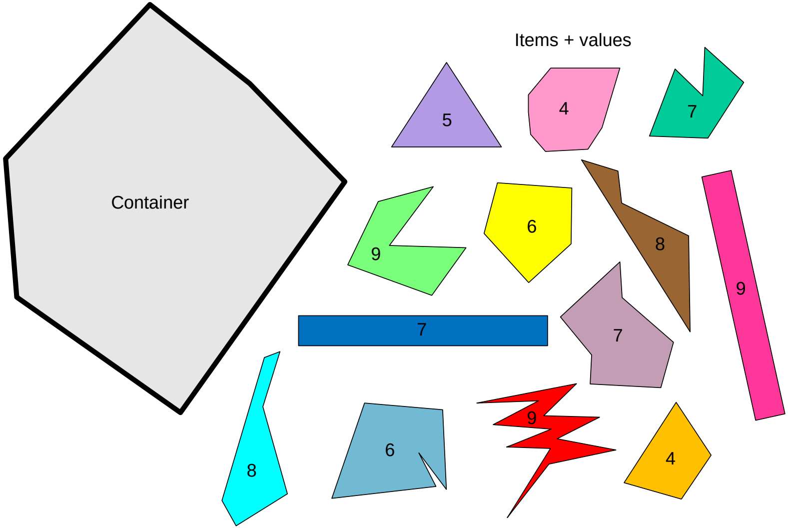
- Penalize long edges and favor breaking long edges:
  $weight(p_1, p_2, q) = area(p_1 p_2 q) + \alpha(\|qp_1\| + \|qp_2\| - \|p_1 p_2\|)$
  for small α
- Add random noise to circumvent stalls



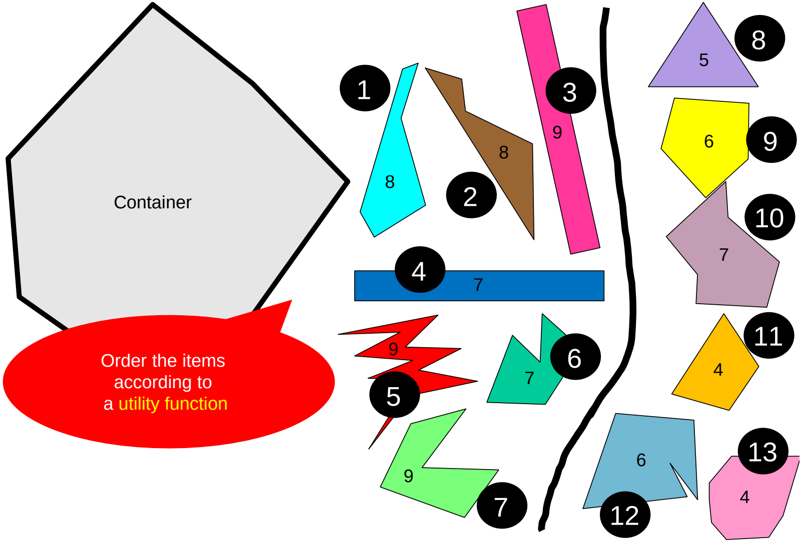
Greedy Packing

Shadoks

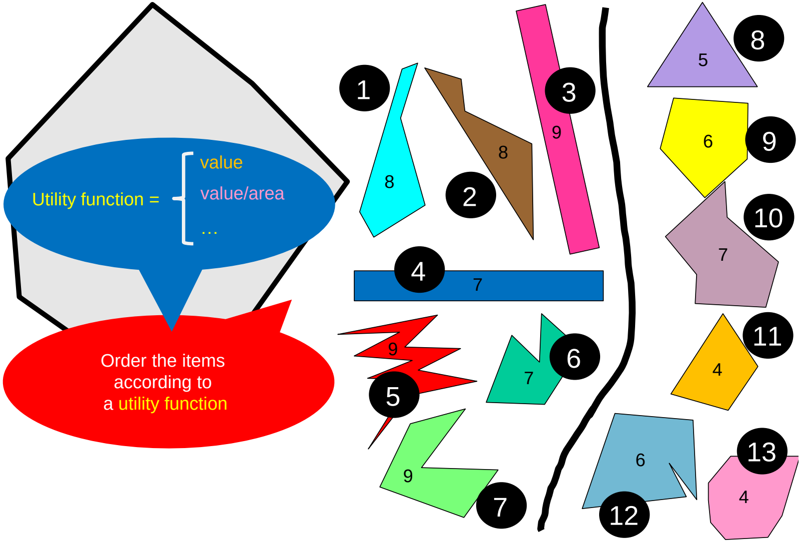
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Greedy Packing

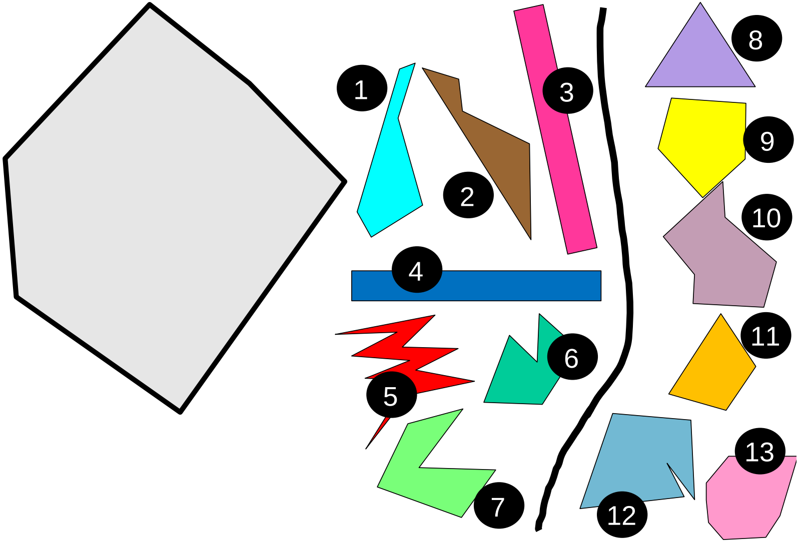


Greedy Packing



Greedy Packing

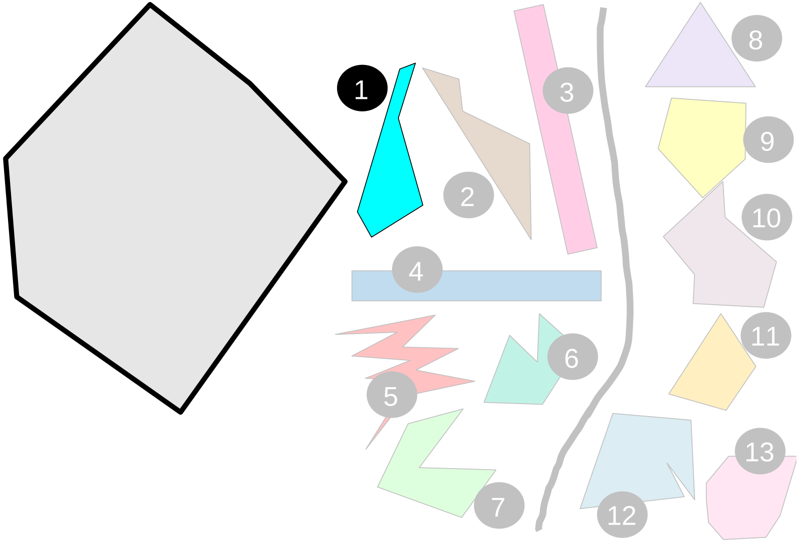
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Greedy Packing

Shadoks

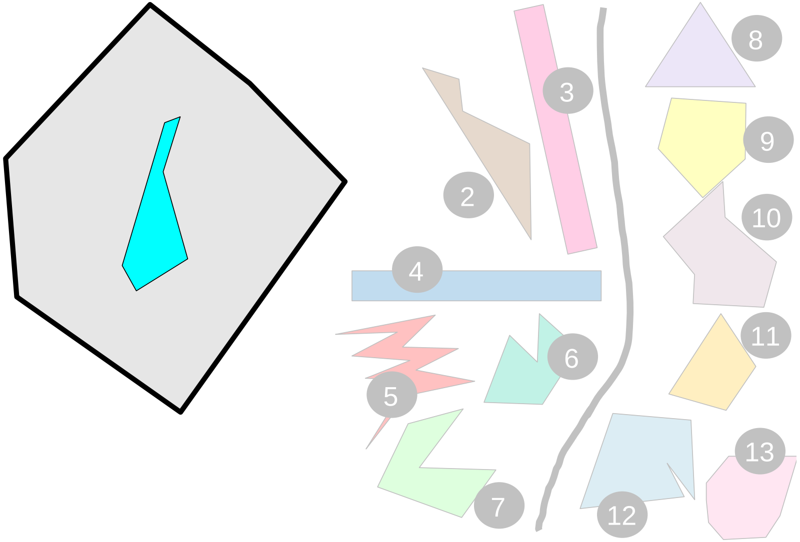
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



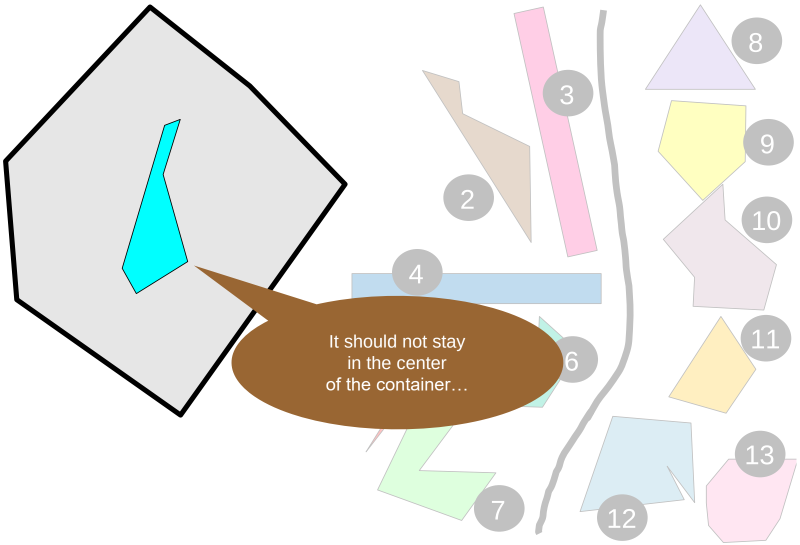
Greedy Packing

Shadoks

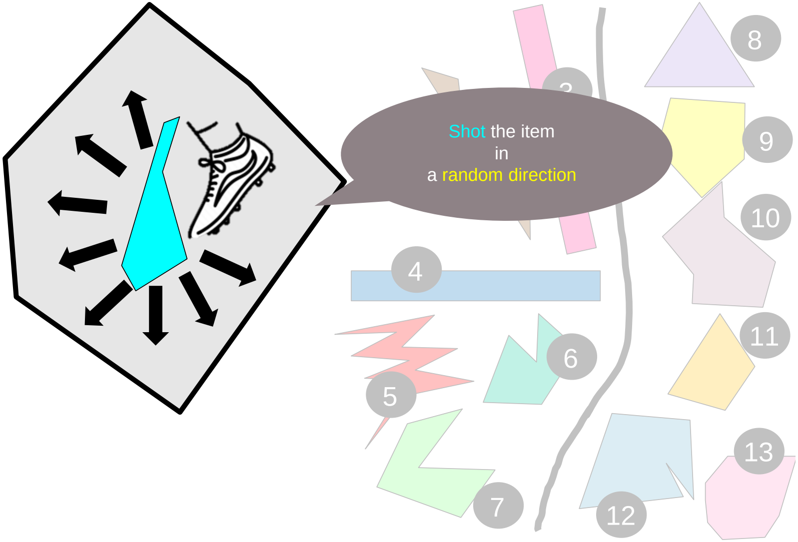
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



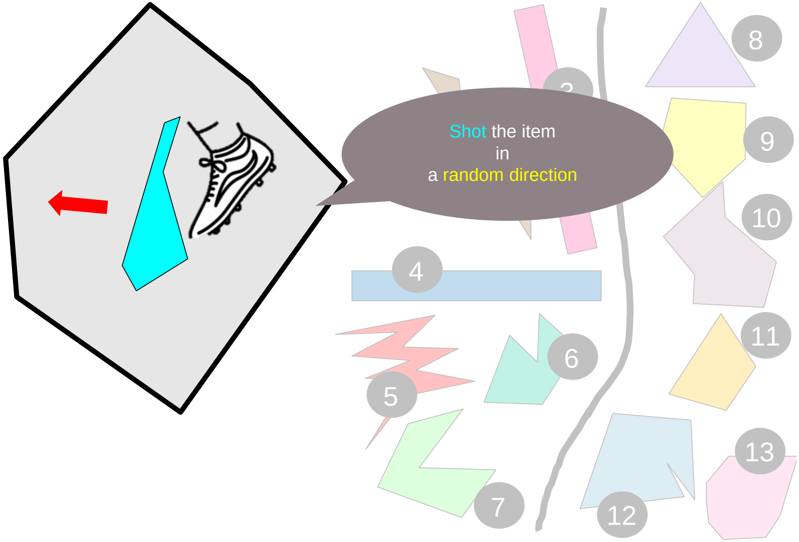
Greedy Packing



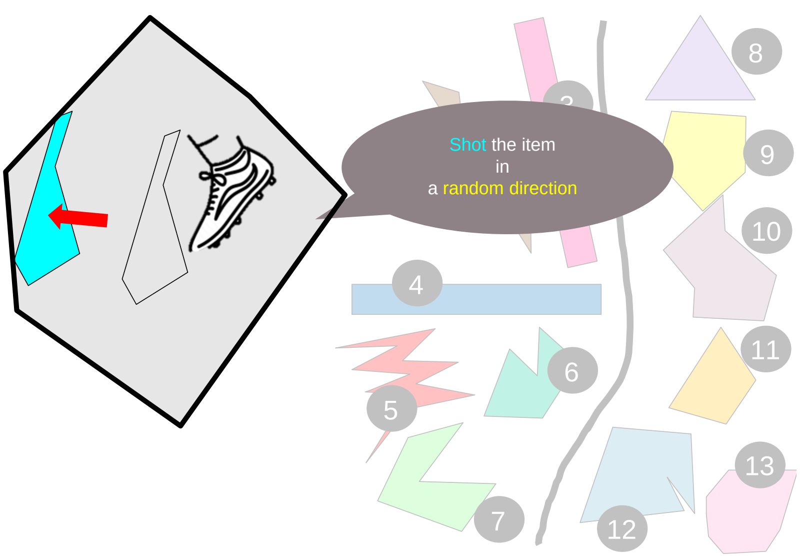
Greedy Packing



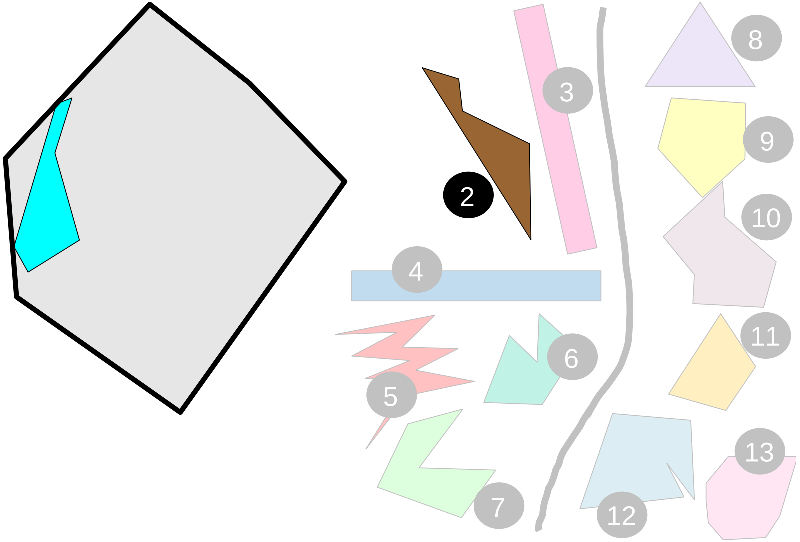
Greedy Packing



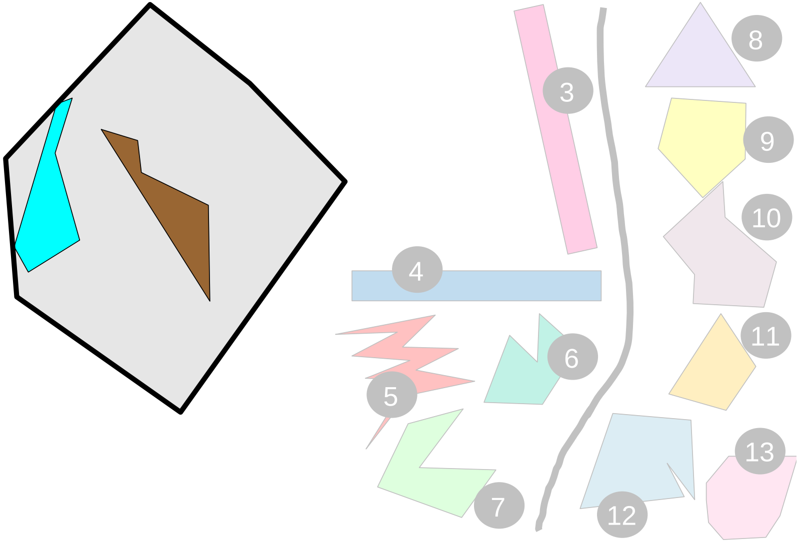
Greedy Packing



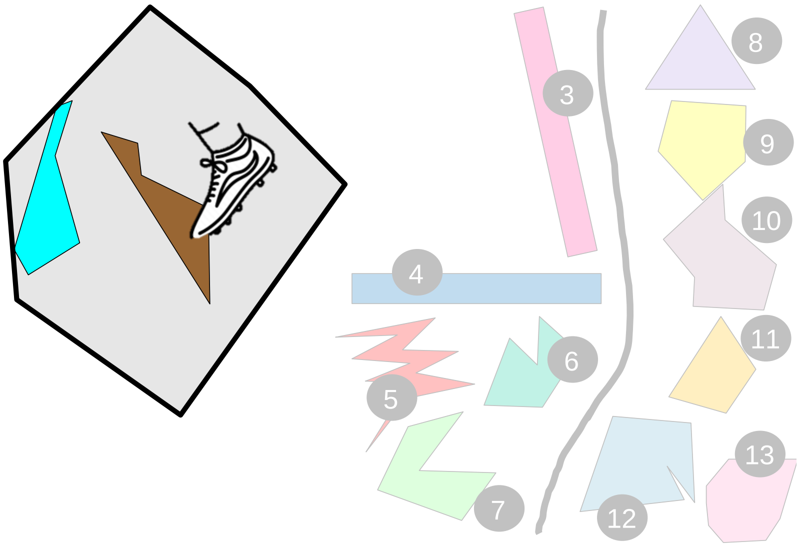
Greedy Packing



Greedy Packing



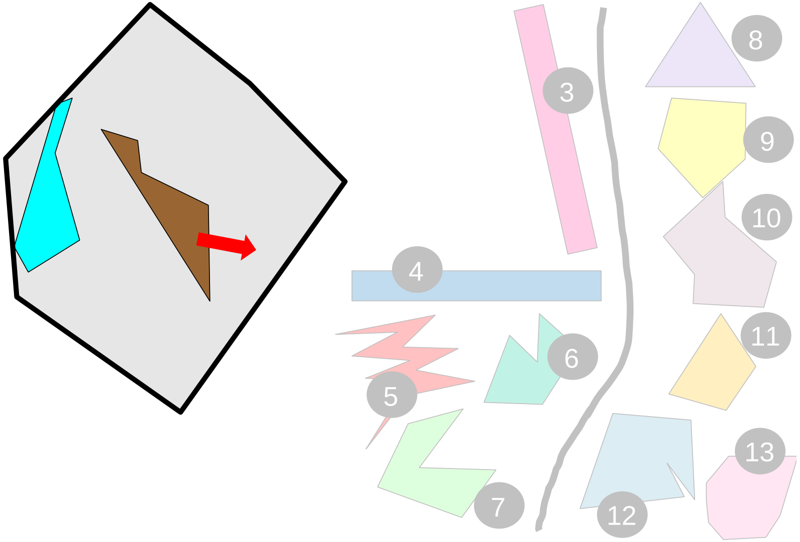
Greedy Packing



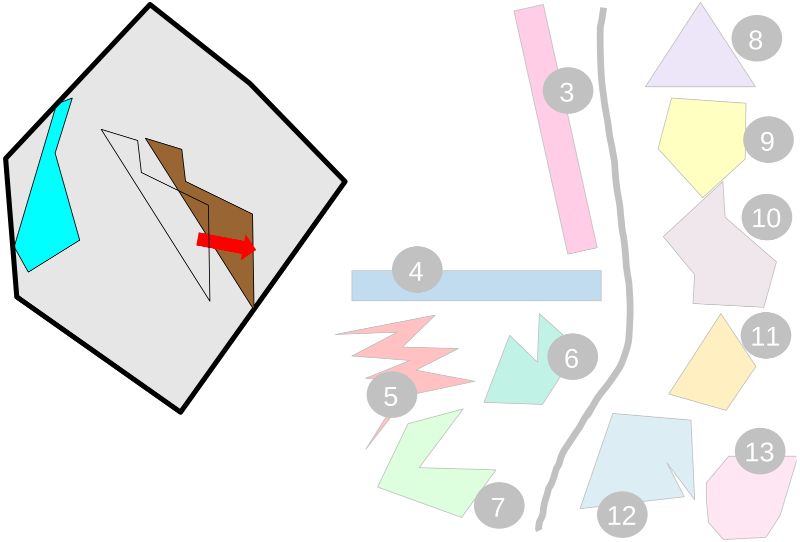
Greedy Packing

Shadoks

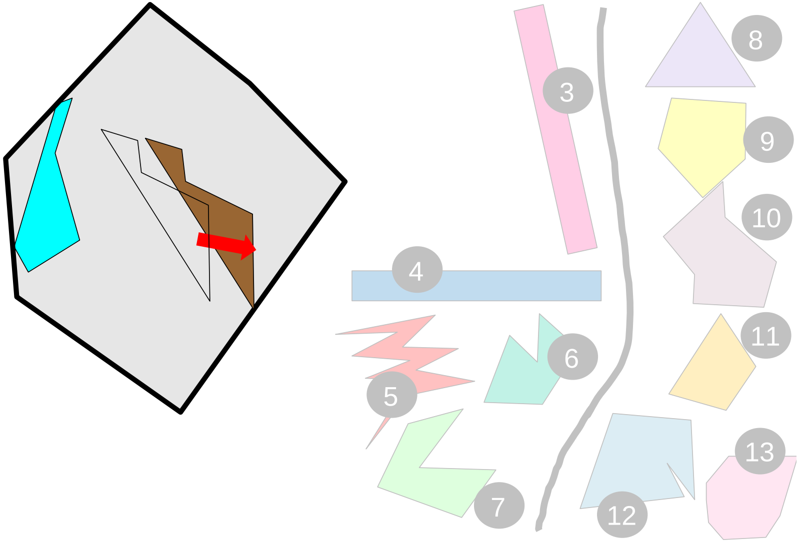
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



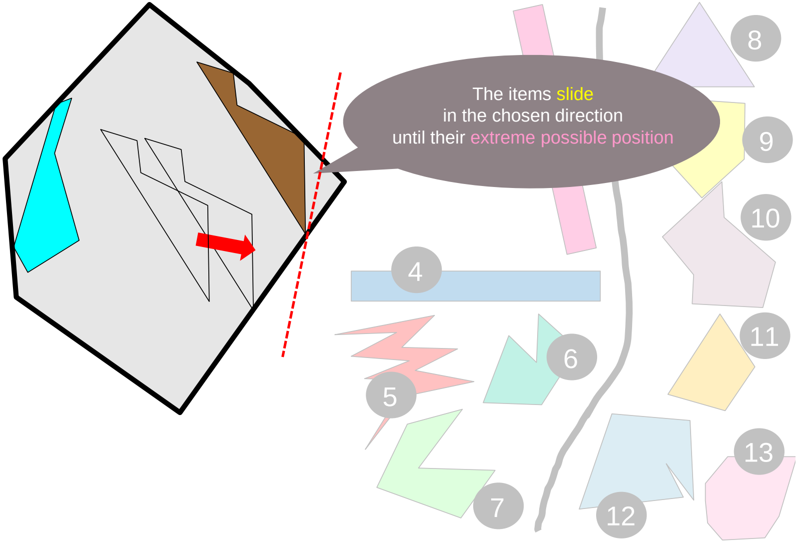
Greedy Packing



Greedy Packing



Greedy Packing



Greedy Packing

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

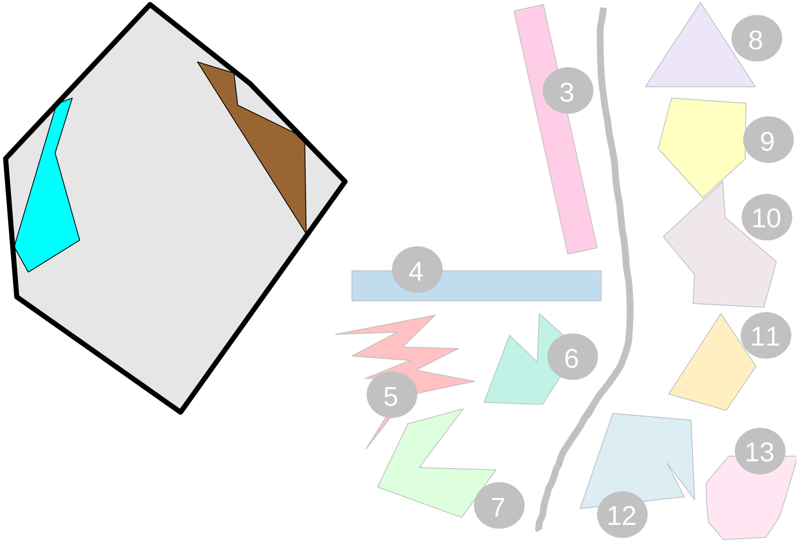
1-IP
Packing
ConvexCover

2-IP
ConvexCover
ConvexPartition

2-Local
MaxPolygon
Packing

2-Conflict
Robots
Coloring

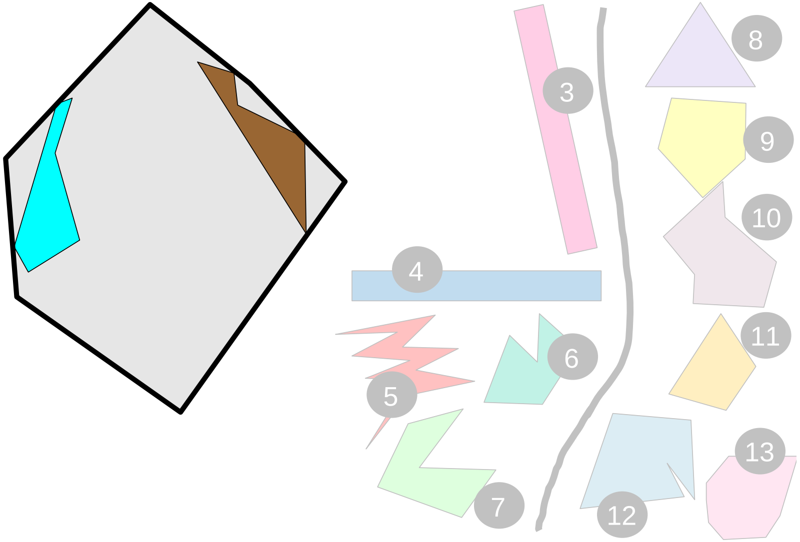
Some results



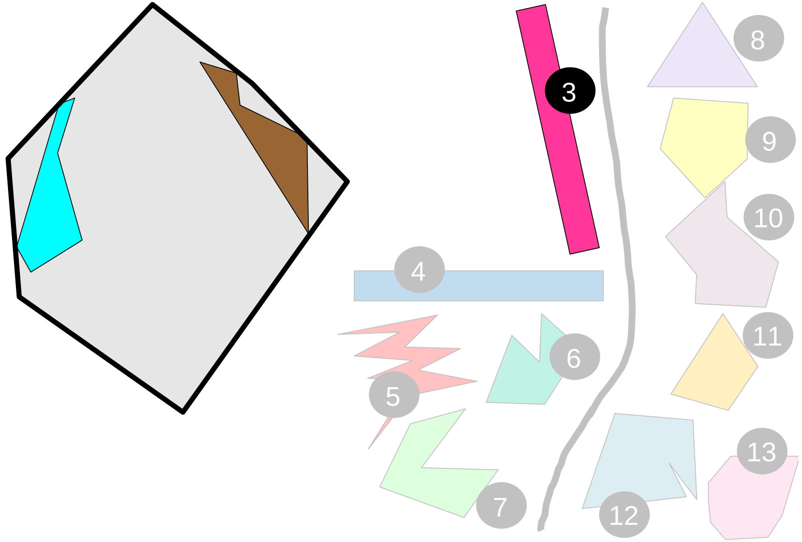
Greedy Packing

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



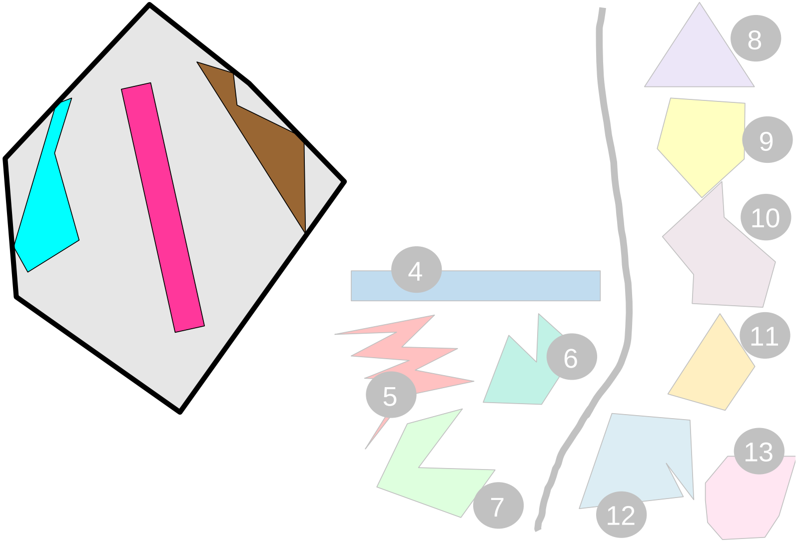
Greedy Packing



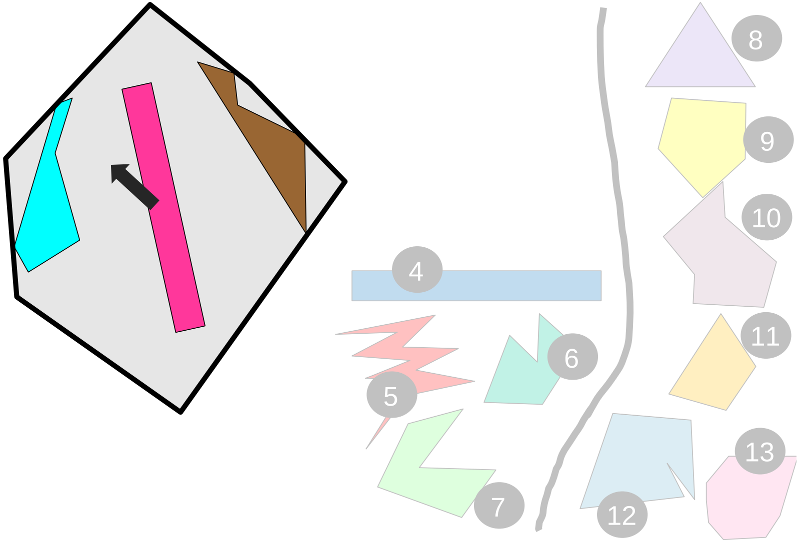
Greedy Packing

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

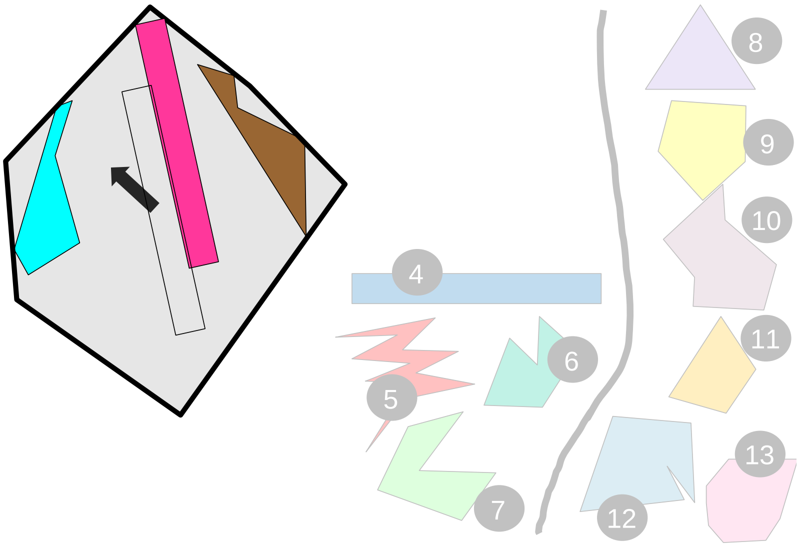


Greedy Packing

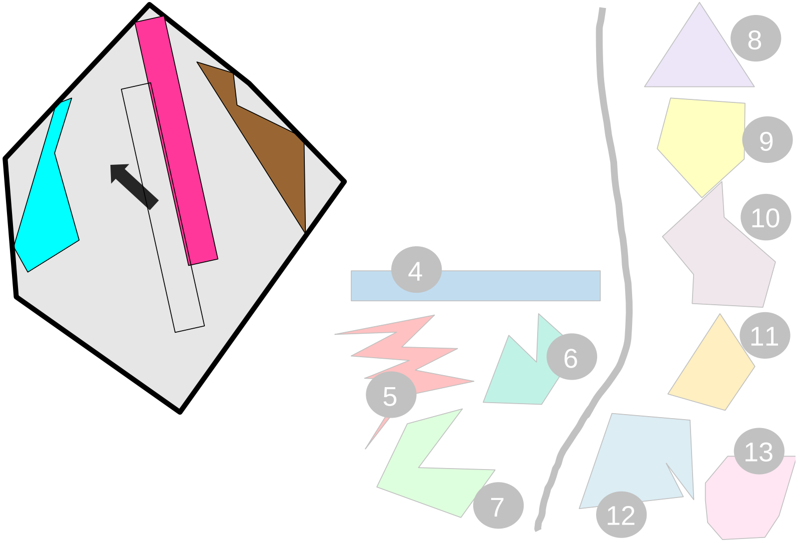


Greedy Packing

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



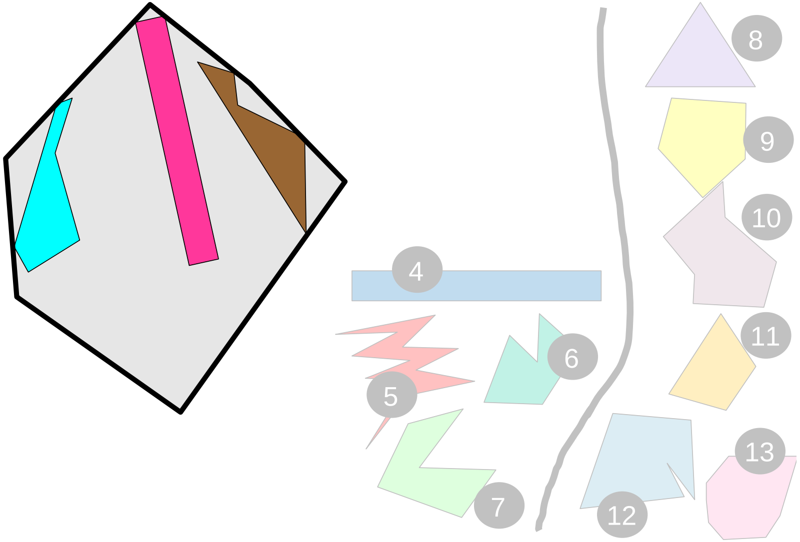
Greedy Packing



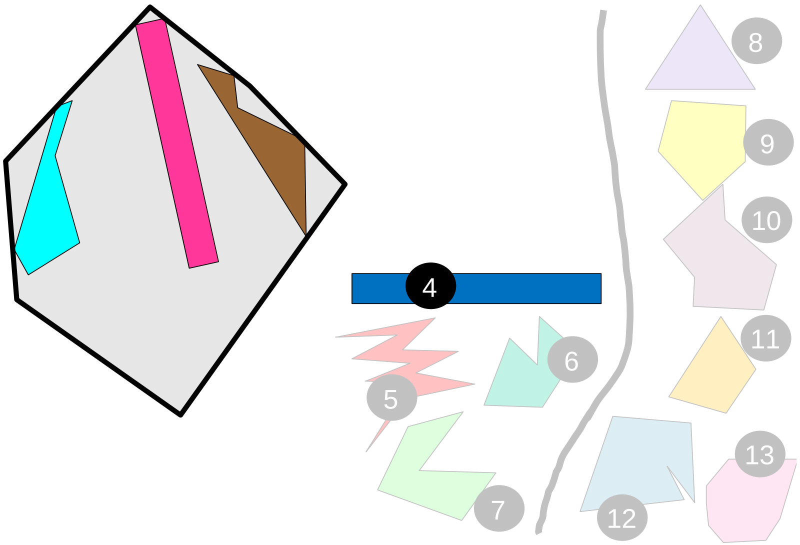
Greedy Packing

Shadoks

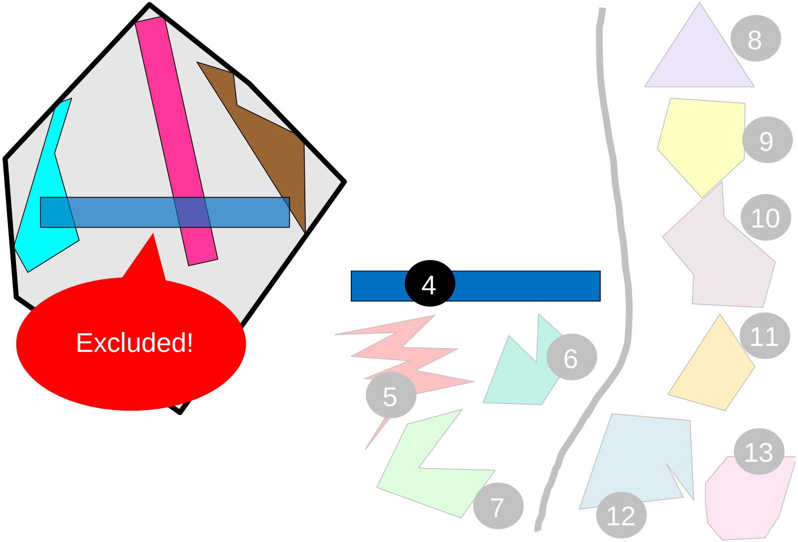
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



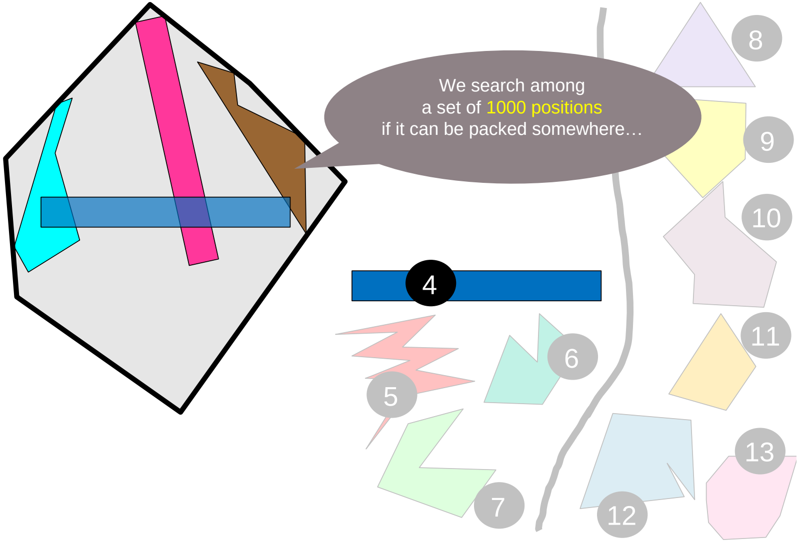
Greedy Packing



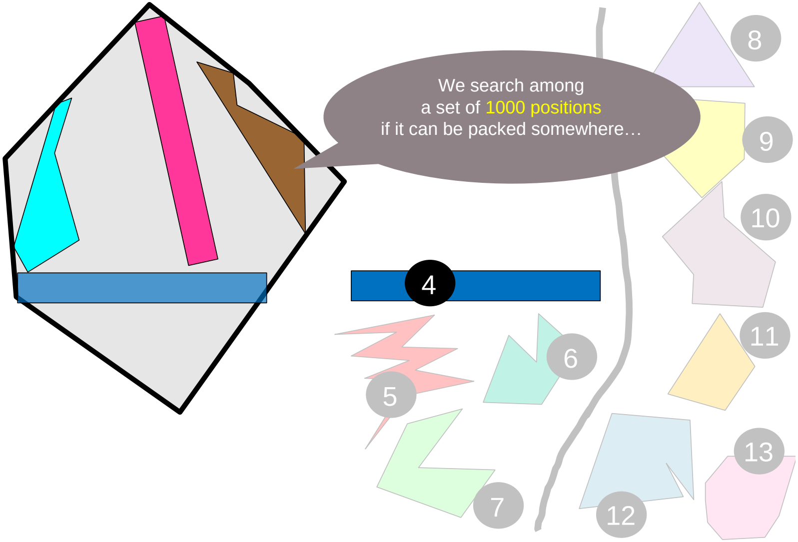
Greedy Packing



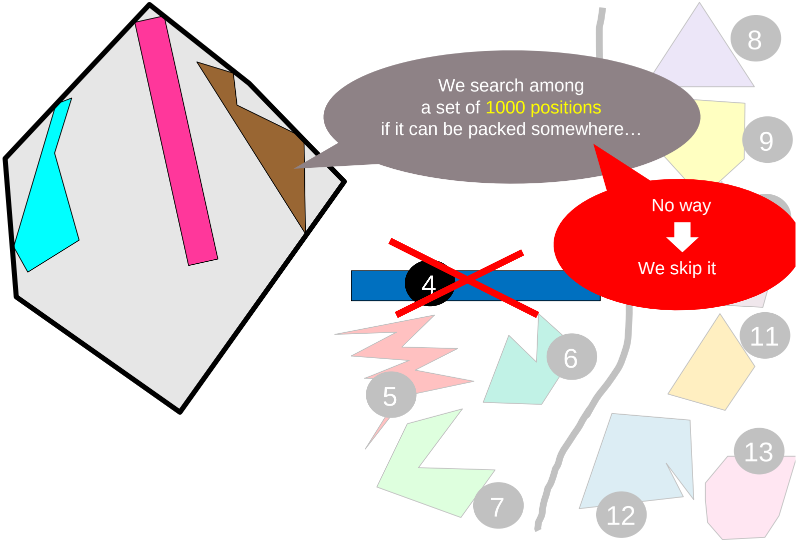
Greedy Packing



Greedy Packing



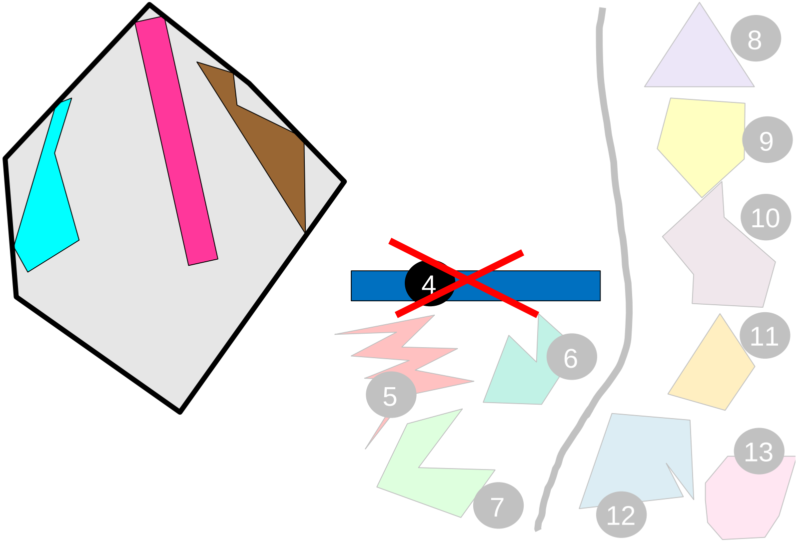
Greedy Packing



Greedy Packing

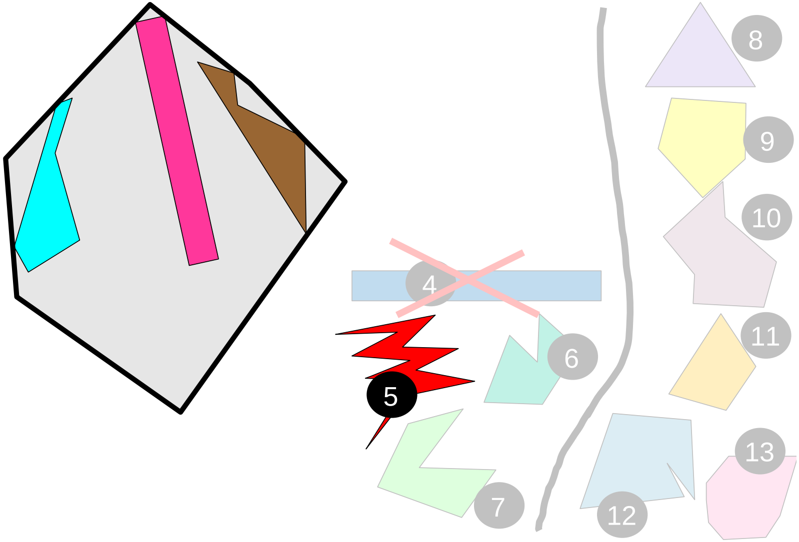
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

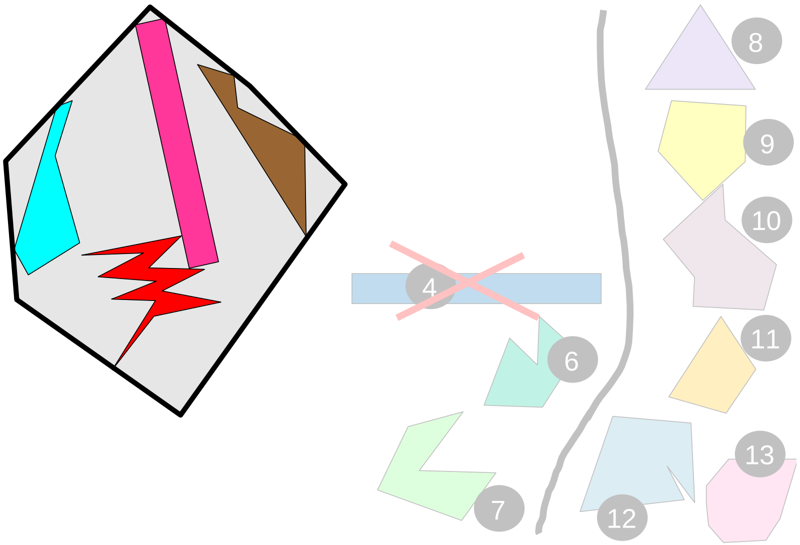


Greedy Packing

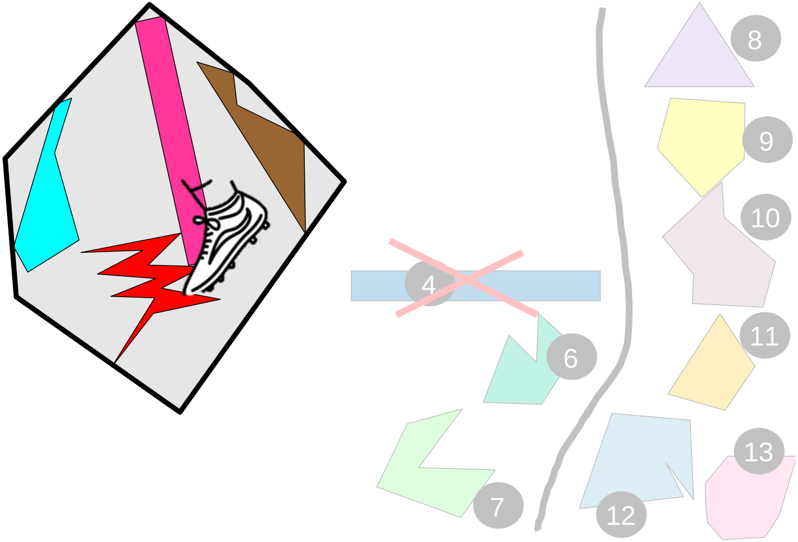
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



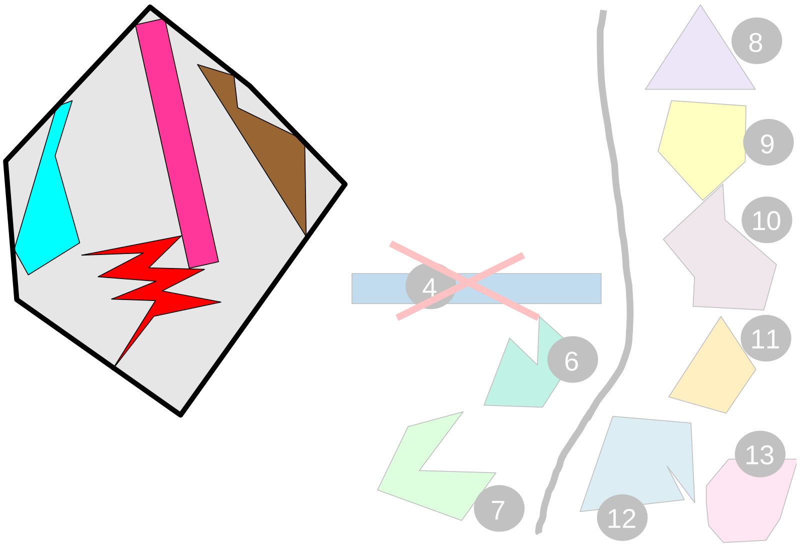
Greedy Packing



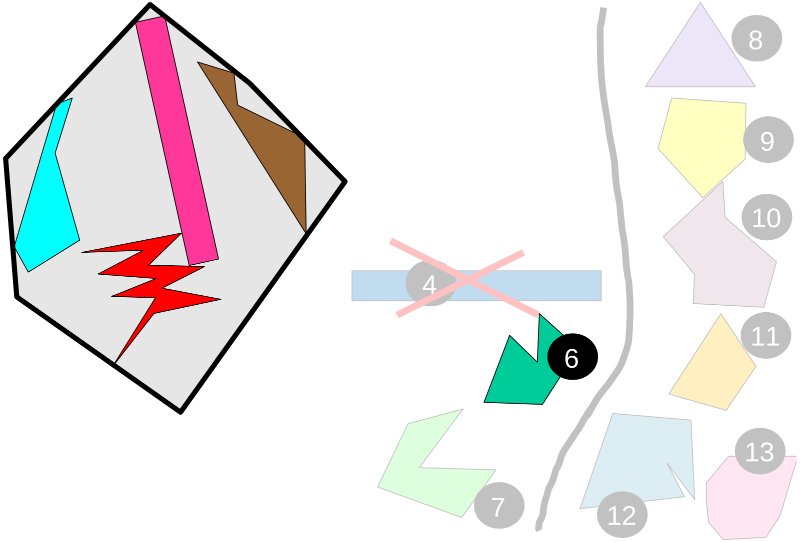
Greedy Packing



Greedy Packing



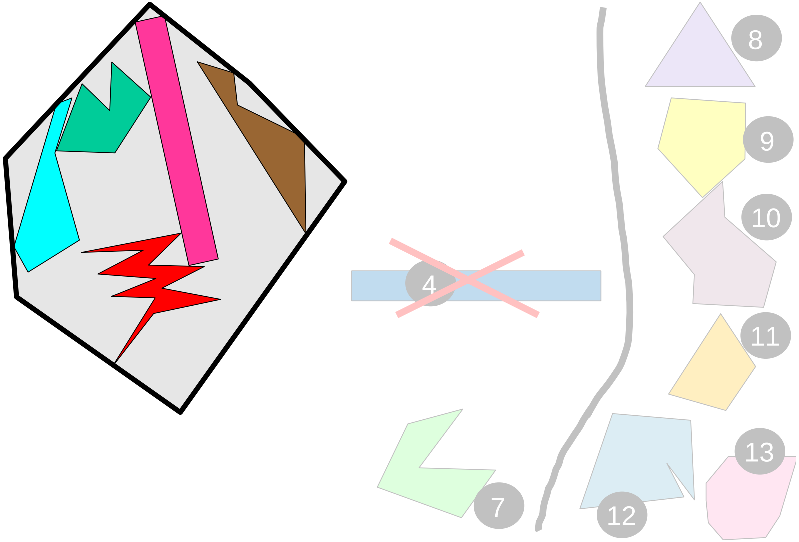
Greedy Packing



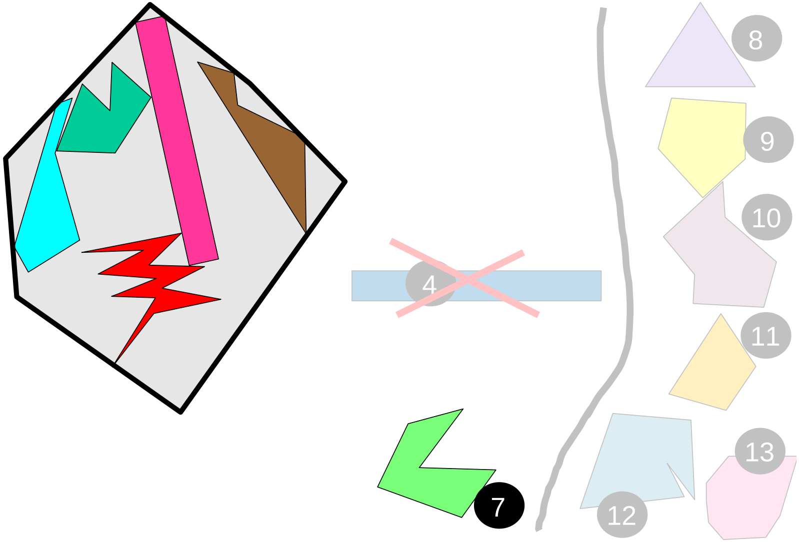
Greedy Packing

Shadoks

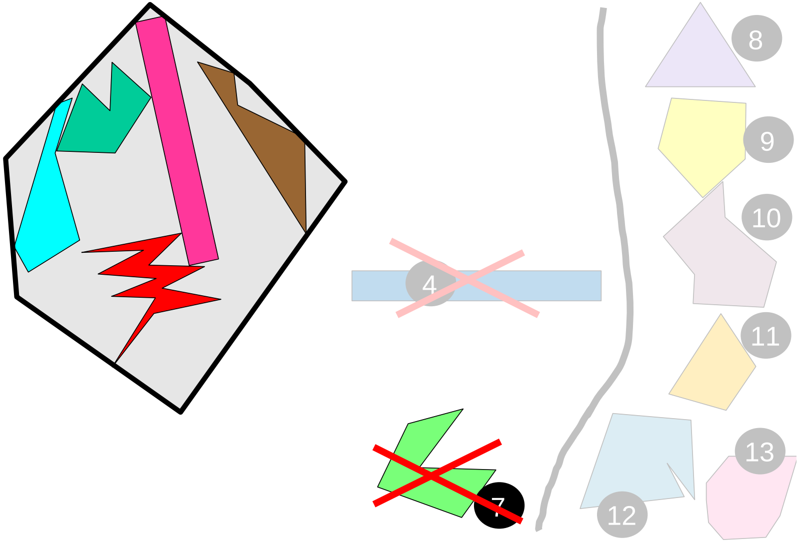
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Greedy Packing



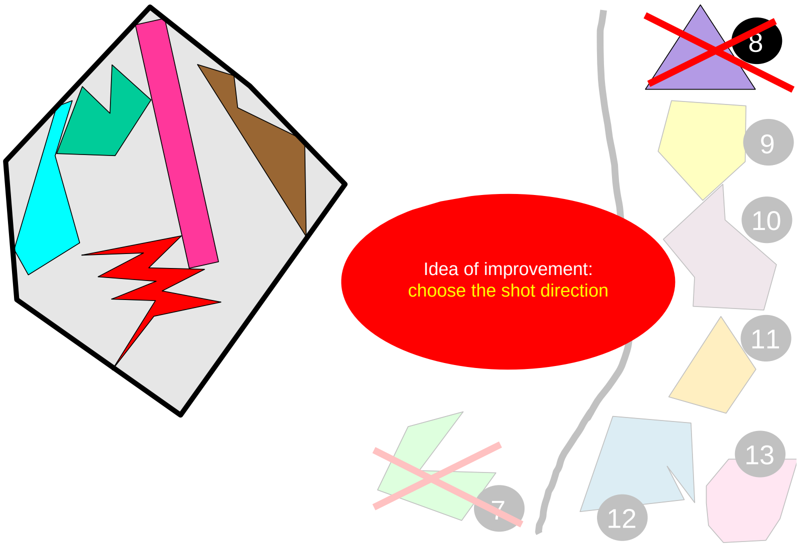
Greedy Packing



Greedy Packing

Shadoks

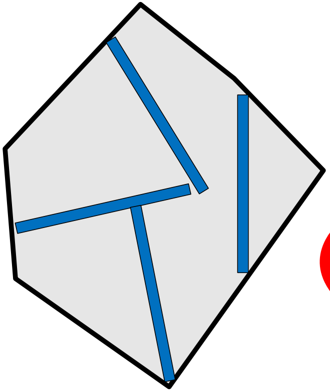
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Choosing a Better Direction

Shadoks

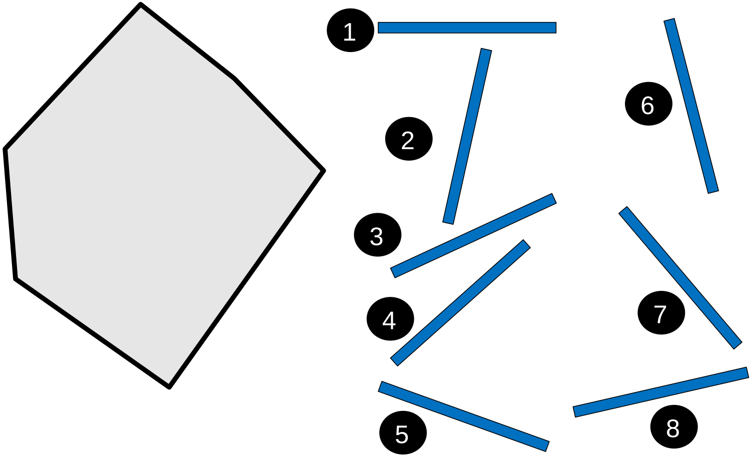
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



For long items:
Random is BAD

Idea of improvement:
choose the shot direction

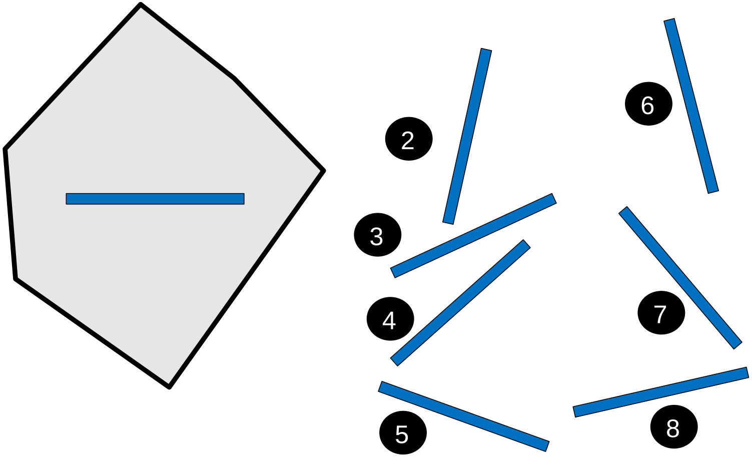
Choosing a Better Direction



Choosing a Better Direction

Shadoks

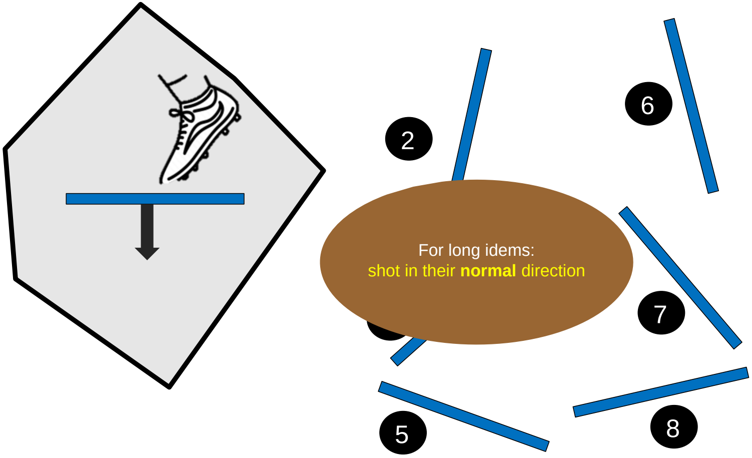
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



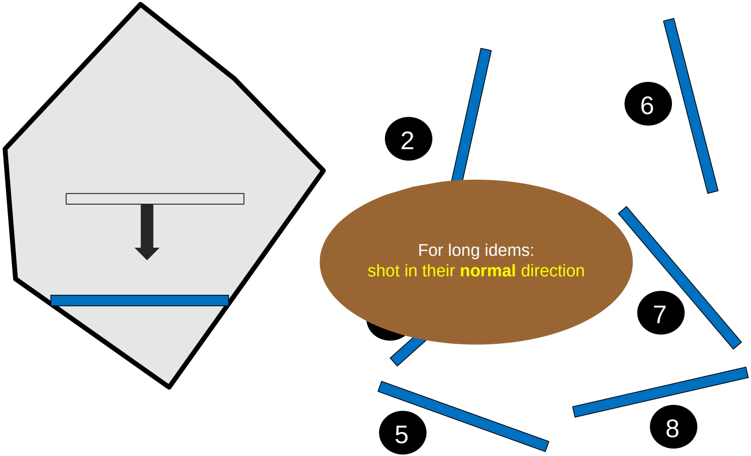
Choosing a Better Direction

Shadoks

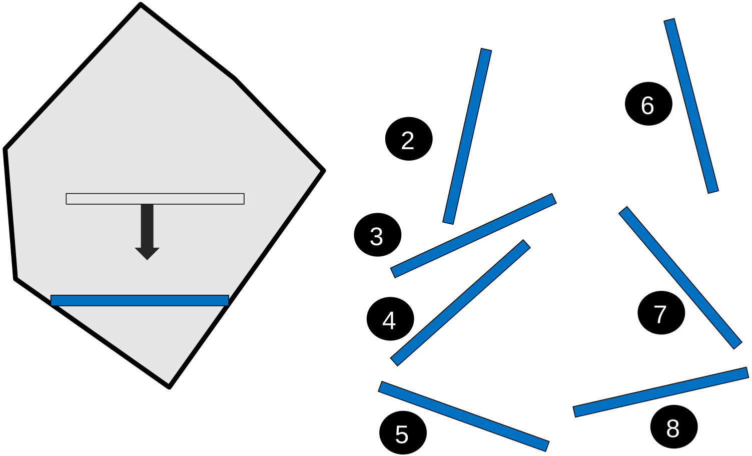
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



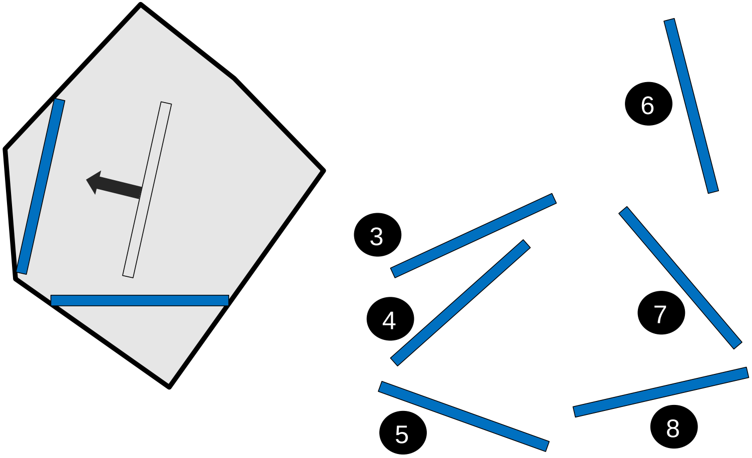
Choosing a Better Direction



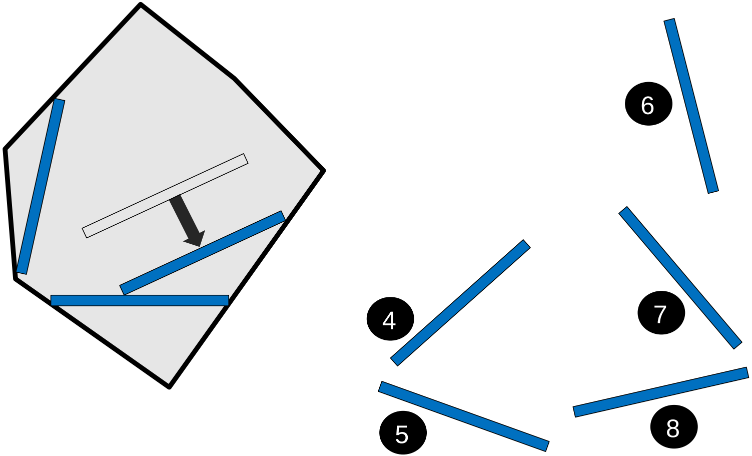
Choosing a Better Direction



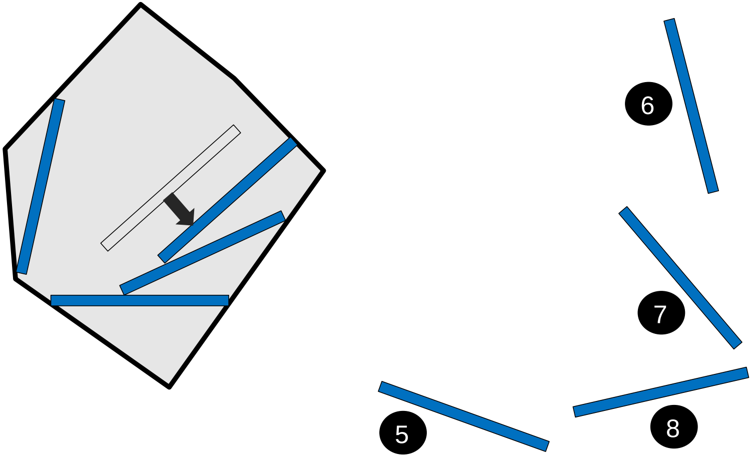
Choosing a Better Direction



Choosing a Better Direction



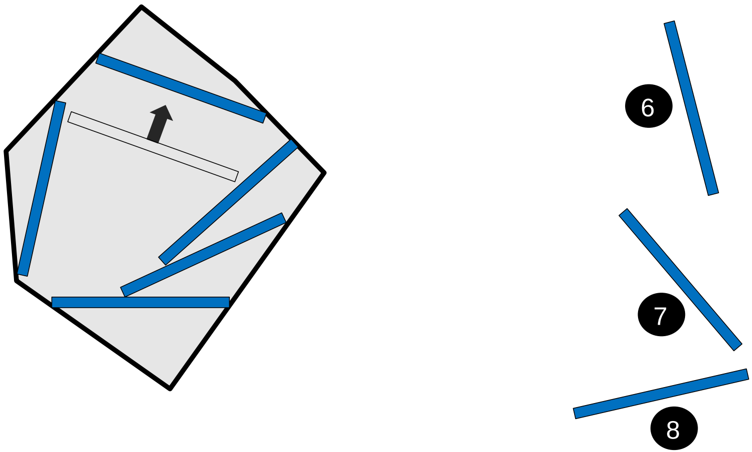
Choosing a Better Direction



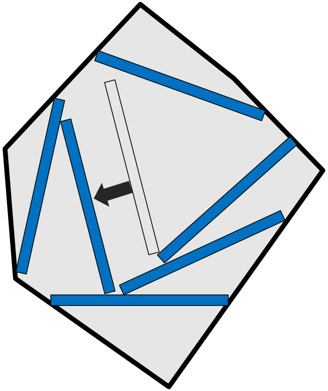
Choosing a Better Direction

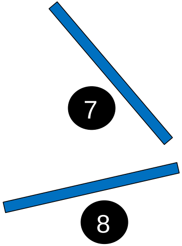
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Choosing a Better Direction

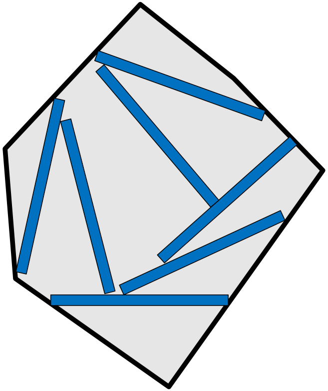


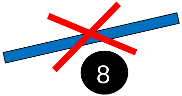


Choosing a Better Direction

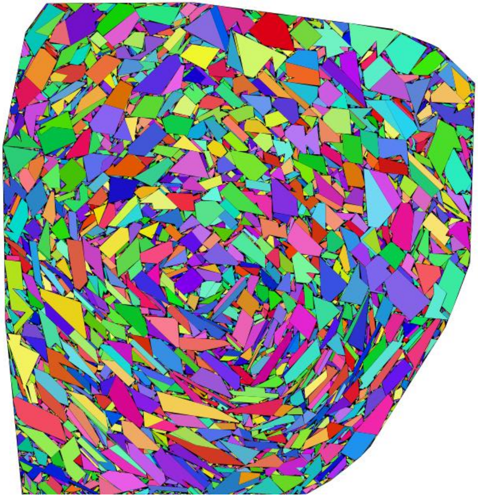
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results





Choosing a Better Direction

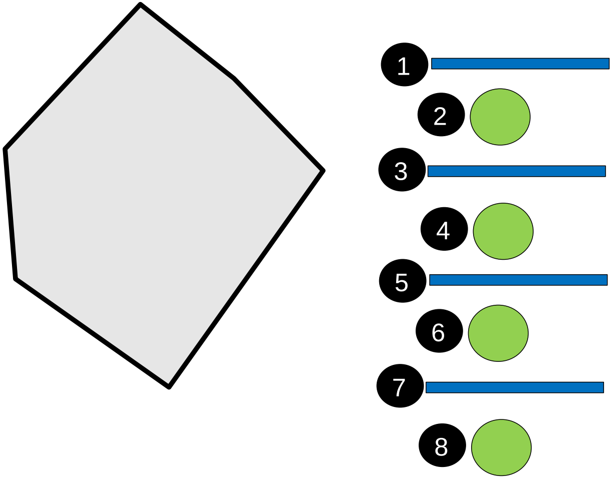


Choosing a Better Direction

Shadoks

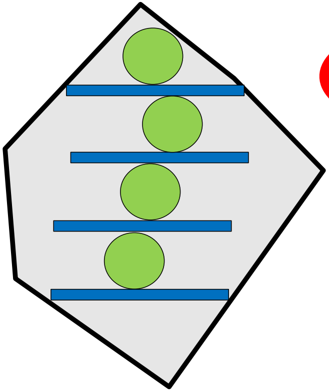
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

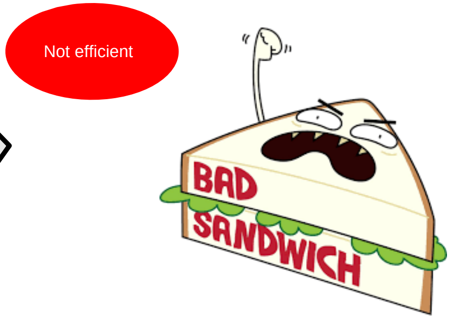
Choosing a Better Direction



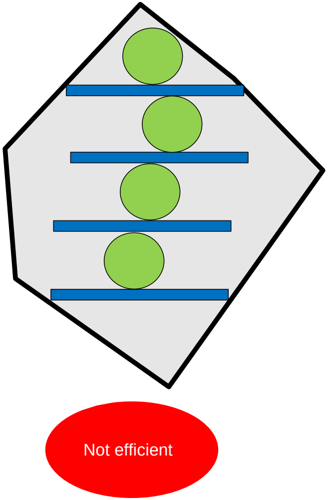
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

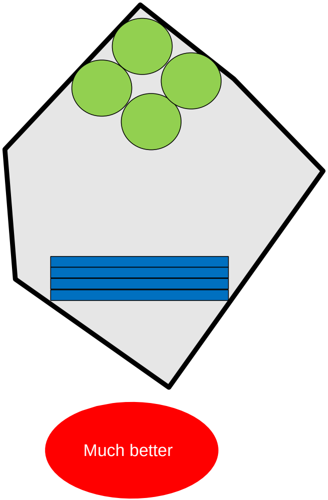
Choosing a Better Direction





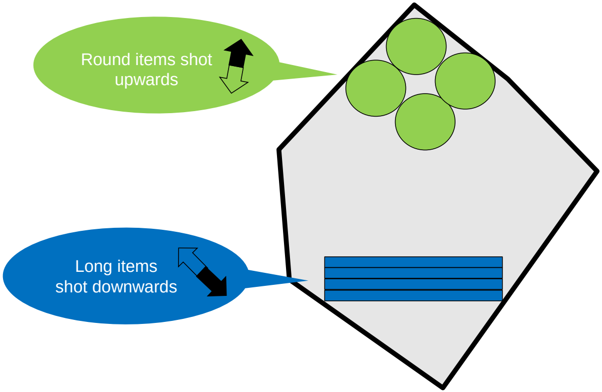
Choosing a Better Direction



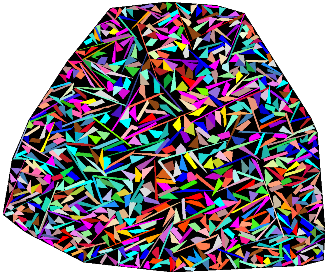


Choosing a Better Direction

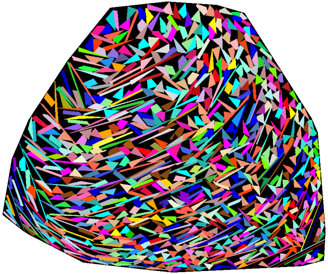
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



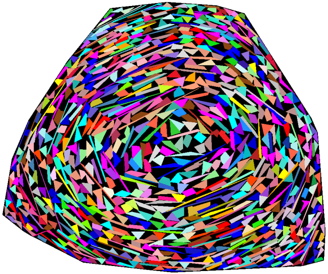
Different Packing Strategies



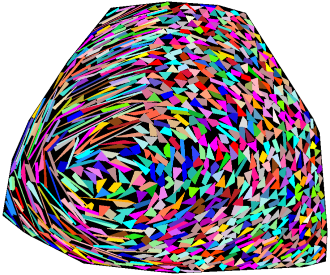
Score= 0,82



Score= 0,85

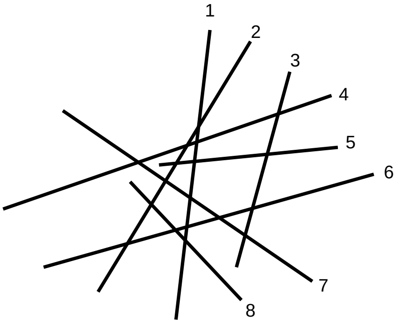


Score= 0,89



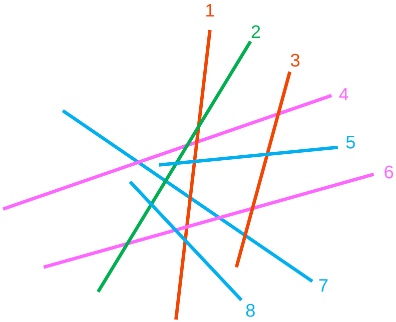
Score= 0,92

Coloring



Input: n segments in the plane

Coloring



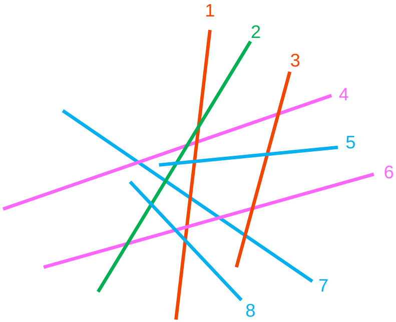
Input: n segments in the plane

Output: segments coloring with a minimum number of colors so that two crossing segments don't have the same color…

Coloring

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

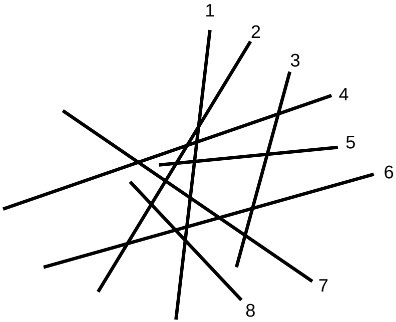


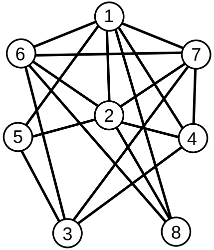
Input: n segments in the plane

Output: segments coloring with a minimum number of colors so that two crossing segments don't have the same color…

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Coloring

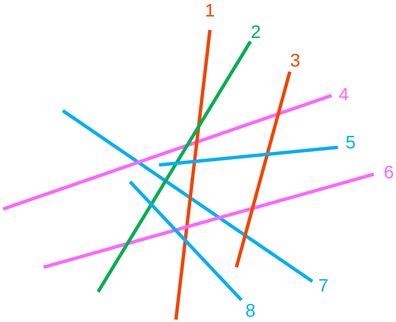


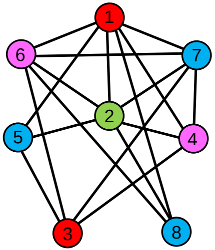


Conflict graph

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Coloring





Any graph coloring algorithm can be used…
or algorithms using geometry

Coloring

Many graph coloring heurisitics…

DSATUR [Brélaz, 1979]

RLF [Leighton, 1979]

Tabu search [Hertz and de Werra, 1987]

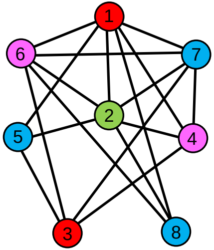
Hybrid evolutionary algorithms

Population-based hybrid algorithms

Reduce and solve algorithms

Modified cuckoo algorithm

Boolean satisfiability formulation



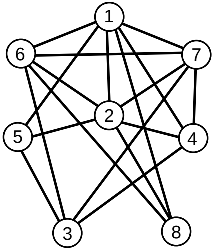
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Shadoks

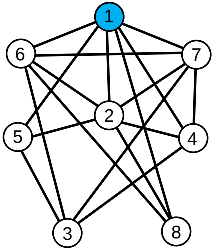
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Color 1

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Coloring

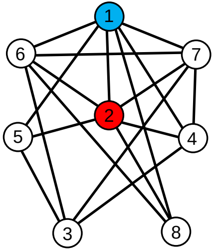
Color 1

Color 2

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Shadoks

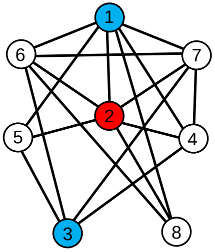
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Color 1 Color 2

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Shadoks

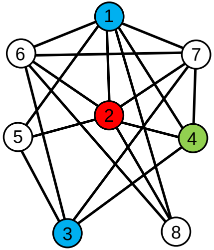
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Color 1 Color 2 Color 3

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



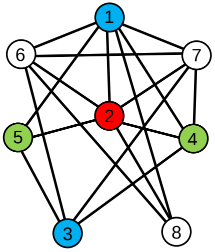
Coloring

Color 1 Color 2 Color 3

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Coloring

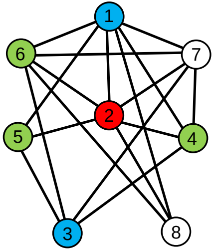
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Color 1 Color 2 Color 3

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



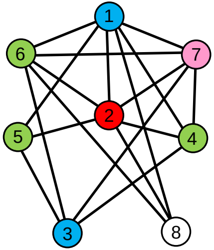
Coloring

Color 1 Color 2 Color 3 Color 4

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



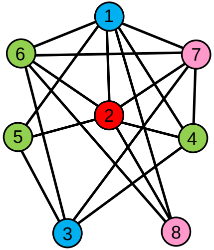
Coloring

Color 1 | Color 2 | Color 3 | Color 4

Most simple heuristic

Initialization: order the vertices

Repeat: take the first uncolored vertex of the list
color it with the first possible color
(create a new color if necessary)



Coloring

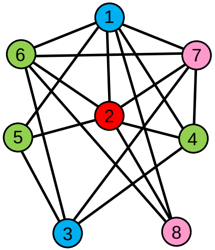
Color 1 | Color 2 | Color 3 | Color 4

Most simple heuristic

Initialization: order the vertices

Repeat: take the irst uncolored vertex of the list
color it wit he first possible color
(create a color if necessary)

Cool, but which order ?

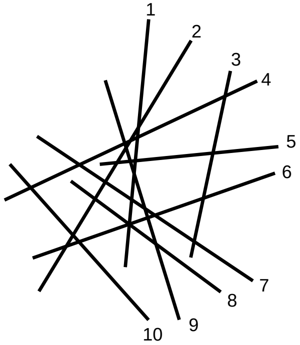


Coloring

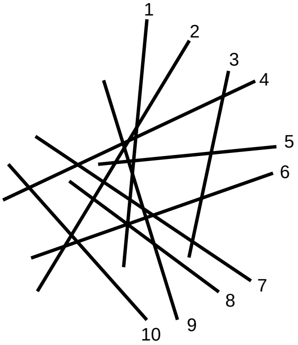
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Cool, but which order ?



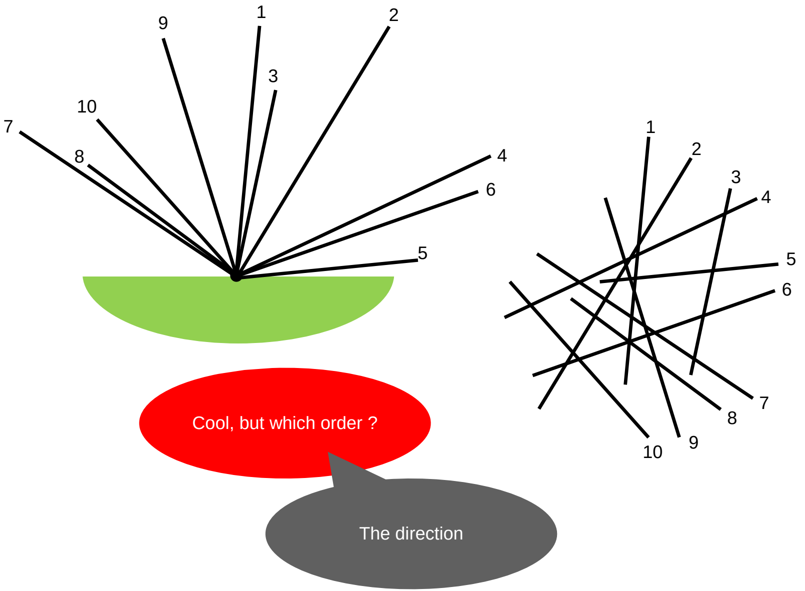
Coloring



Cool, but which order ?

The direction

Coloring


9
1
2
3
10
7
8
4
6
5
Cool, but which order ?
The direction
1
2
3
4
5
6
7
8
9
10

Coloring

Shadoks

Introduction
- Two Phases

1-Greedy
- MaxPolygon
- Packing
- Coloring

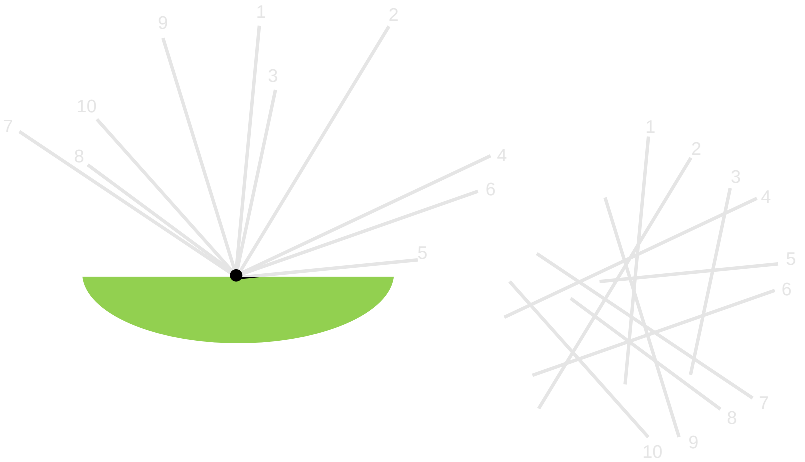
1-IP
- Packing
- ConvexCover

2-IP
- ConvexCover
- ConvexPartition

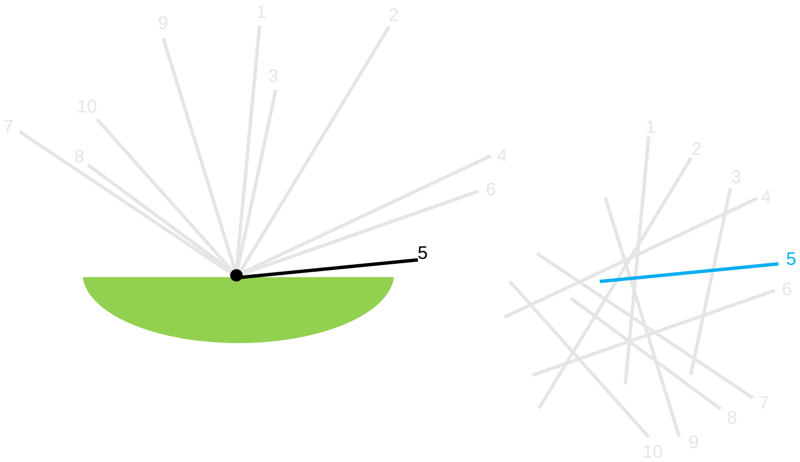
2-Local
- MaxPolygon
- Packing

2-Conflict
- Robots
- Coloring

Some results

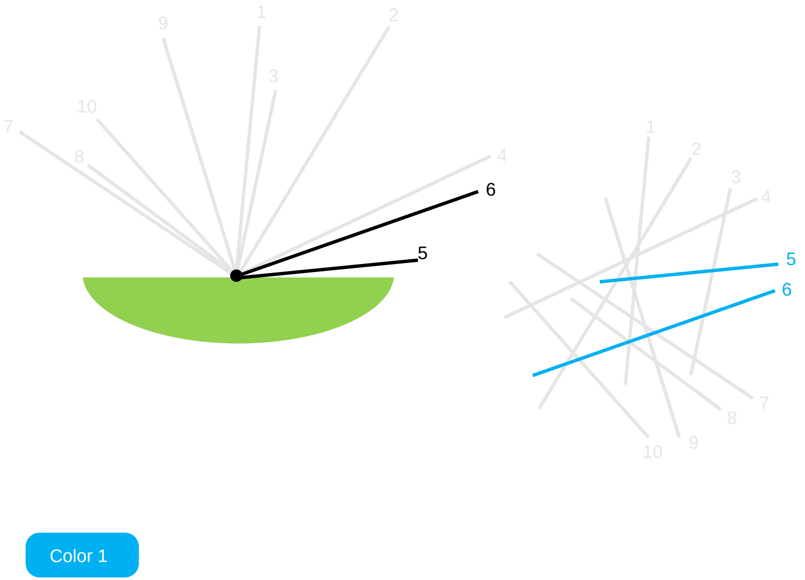


Coloring

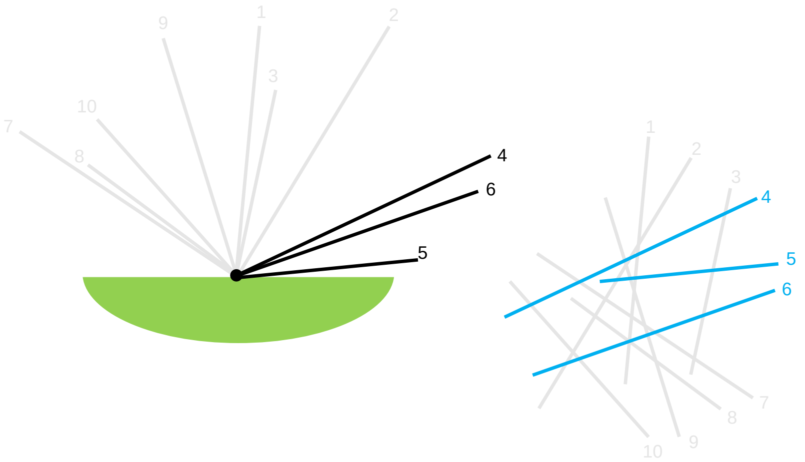


Color 1

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

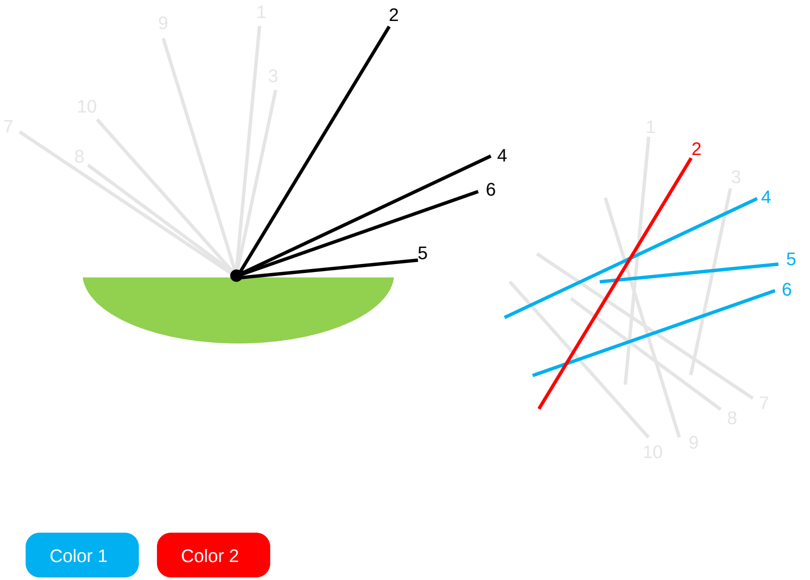


Coloring



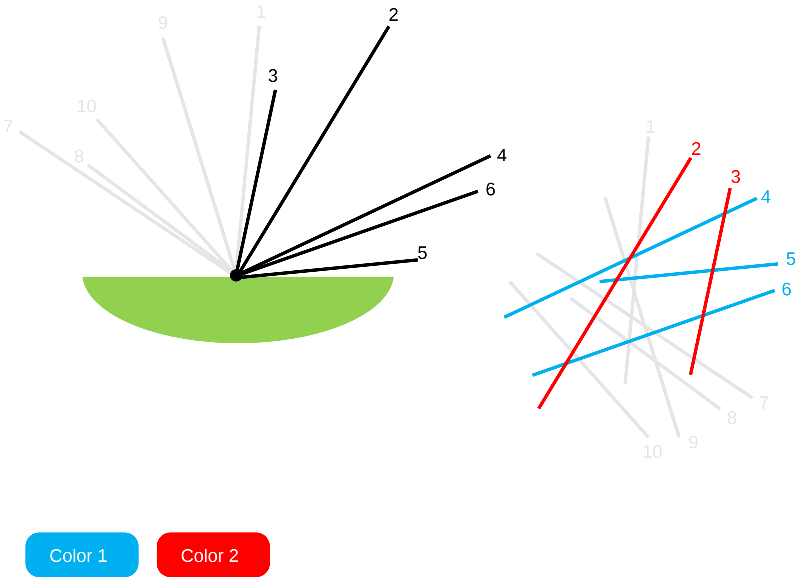
Color 1

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

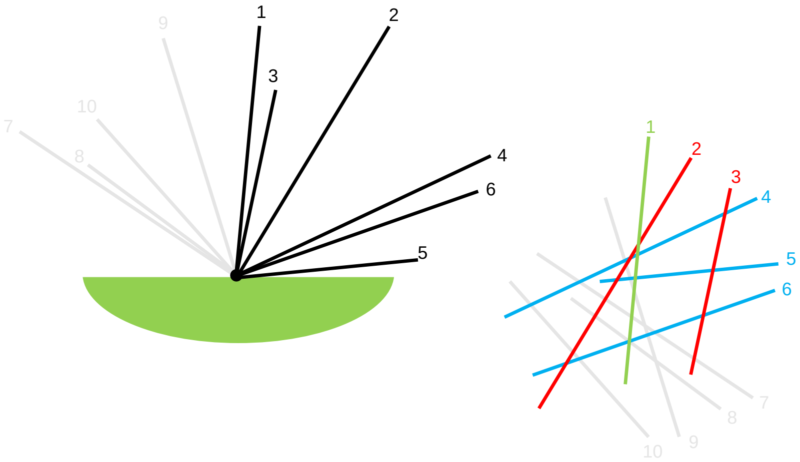


Coloring

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

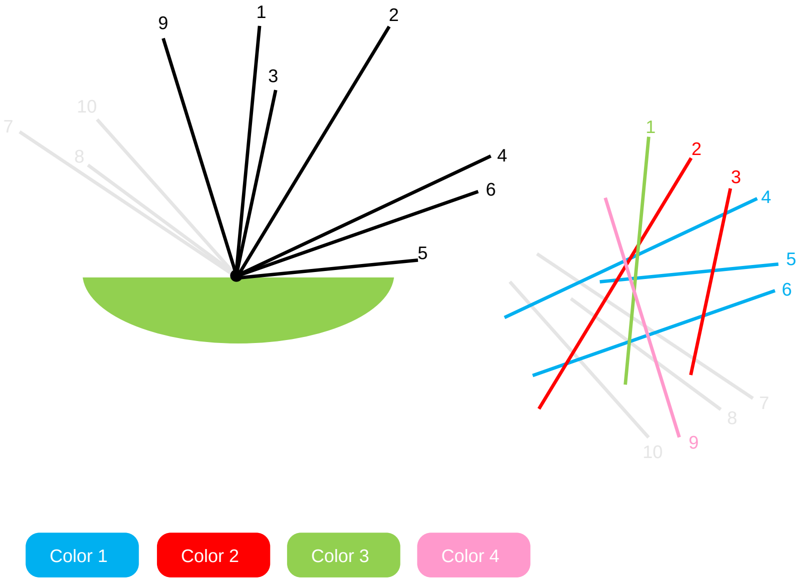


Coloring

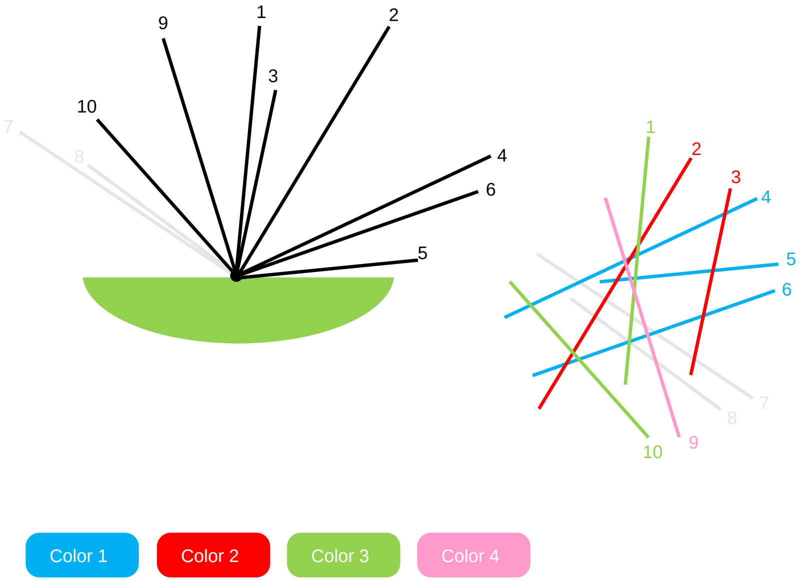


Color 1 Color 2 Color 3

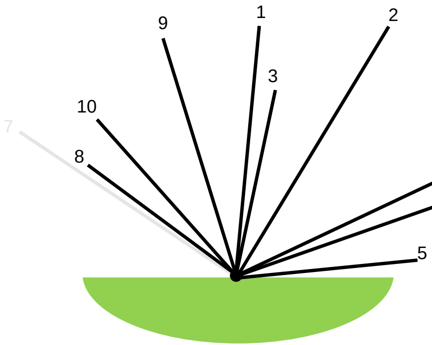
Coloring

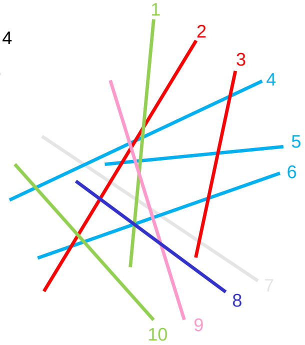


Coloring



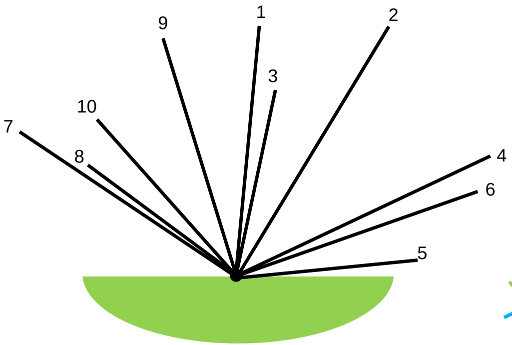
Coloring

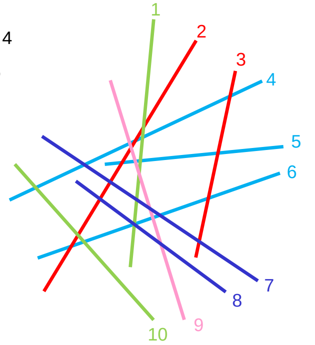




Color 1 Color 2 Color 3 Color 4 Color 5

Coloring





Color 1 Color 2 Color 3 Color 4 Color 5

DSatHull Coloring

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

1-IP
Packing
ConvexCover

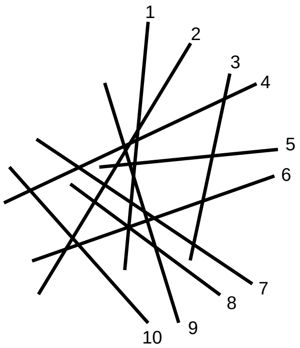
2-IP
ConvexCover
ConvexPartition

2-Local
MaxPolygon
Packing

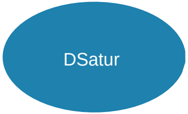
2-Conflict
Robots
Coloring

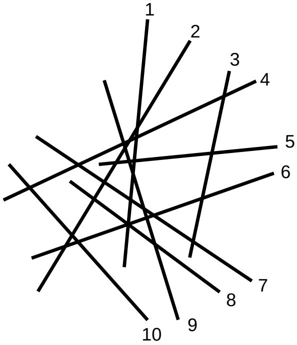
Some results

Another greedy heuristic
(variant of DSatur)



DSatHull Coloring

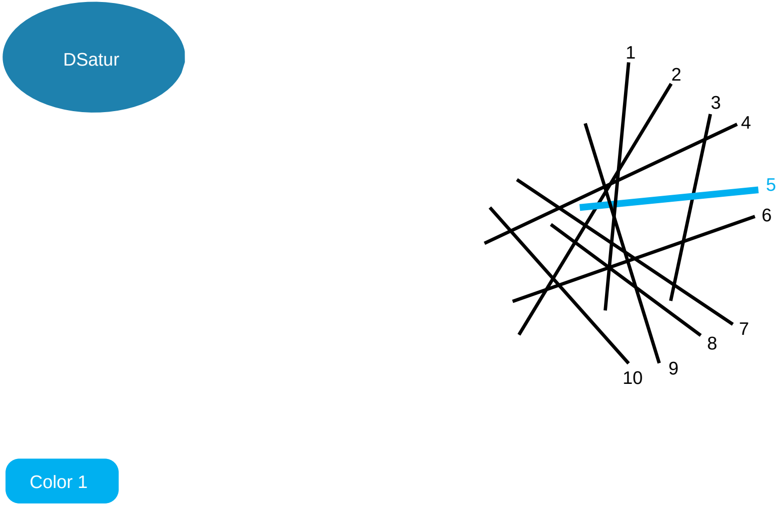




DSatHull Coloring

Shadoks

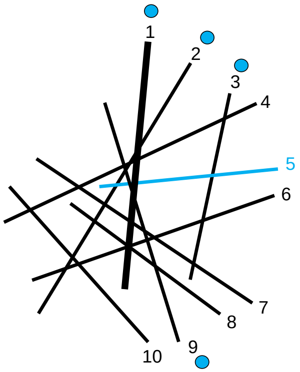
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



DSatHull Coloring

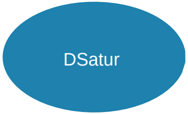
DSatur

Color a segment that crosses the largest number of colors

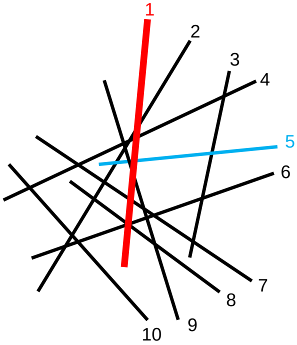


Color 1

DSatHull Coloring



Color a segment that crosses the largest number of colors

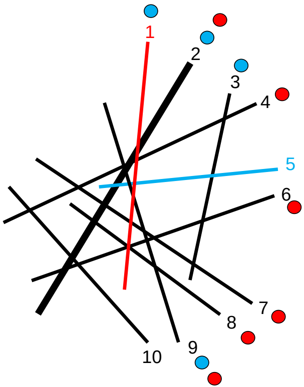


Color 1 Color 2

DSatHull Coloring

DSatur

Color a segment that crosses the largest number of colors



Color 1

Color 2

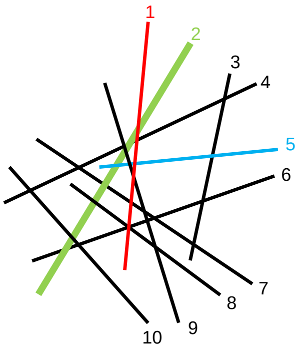
DSatHull Coloring

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

DSatur

Color a segment that crosses the largest number of colors

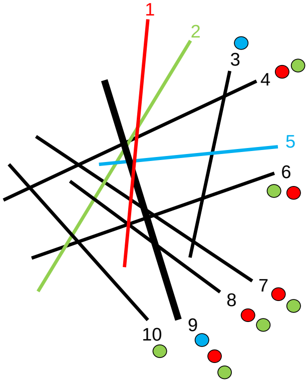


Color 1 Color 2 Color 3

DSatHull Coloring

DSatur

Color a segment that crosses the largest number of colors

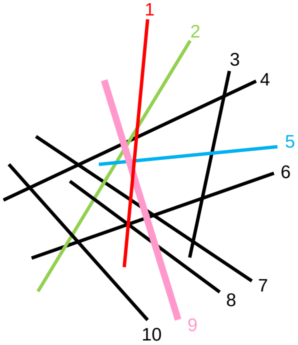


Color 1 Color 2 Color 3

DSatHull Coloring

DSatur

Color a segment that crosses the largest number of colors

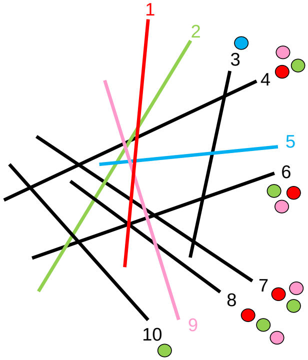


Color 1 | Color 2 | Color 3 | Color 4

DSatHull Coloring

DSatur

Color a segment that crosses the largest number of colors



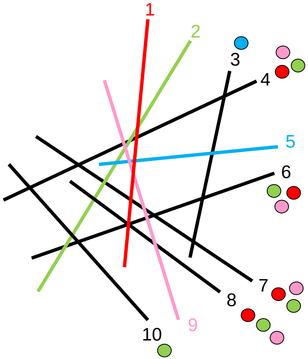
Color 1 | Color 2 | Color 3 | Color 4

DSatHull Coloring

DSatur

Color a segment that crosses the largest number of colors

And we continue...

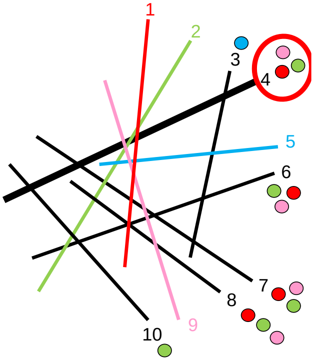


Color 1 | Color 2 | Color 3 | Color 4

DSatHull Coloring

DSatur

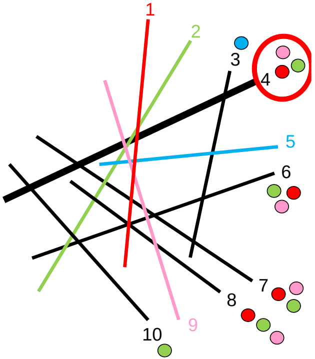
Color a segment that crosses the largest number of colors



Color 1 | Color 2 | Color 3 | Color 4

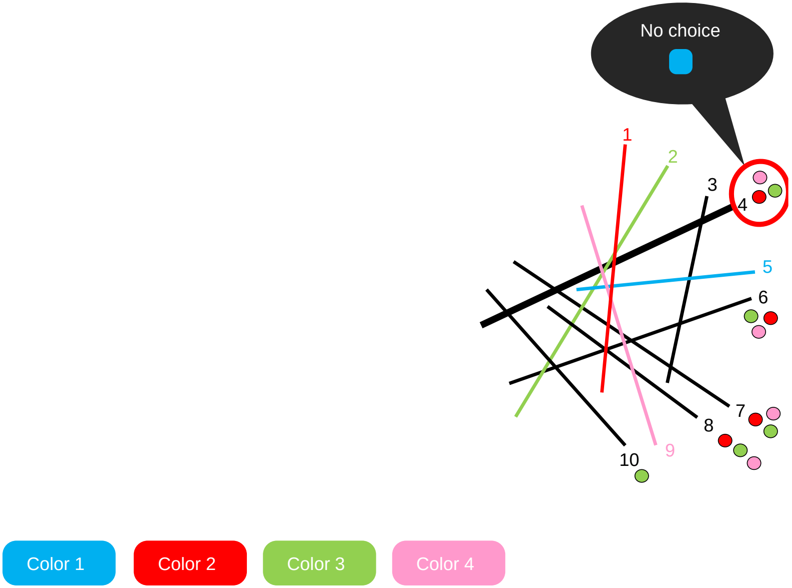
DSatHull Coloring

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

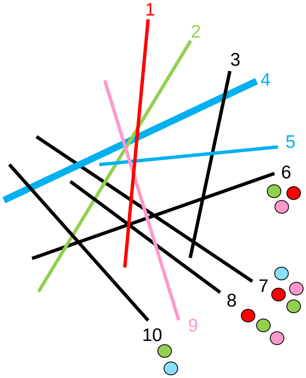


Color 1 Color 2 Color 3 Color 4

DSatHull Coloring

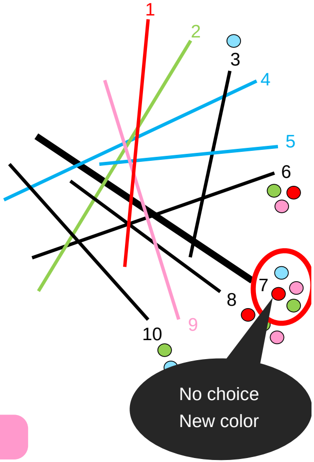


DSatHull Coloring



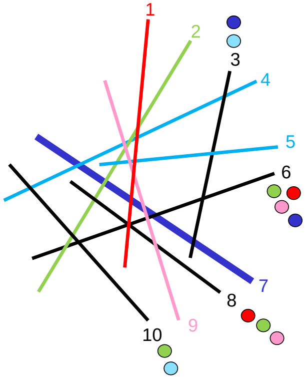
Color 1 Color 2 Color 3 Color 4

DSatHull Coloring



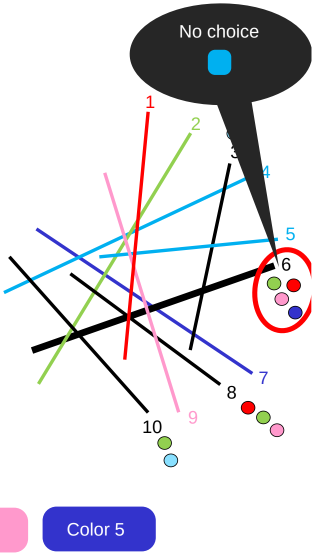
Color 1 | Color 2 | Color 3 | Color 4

DSatHull Coloring



Color 1 | Color 2 | Color 3 | Color 4 | Color 5

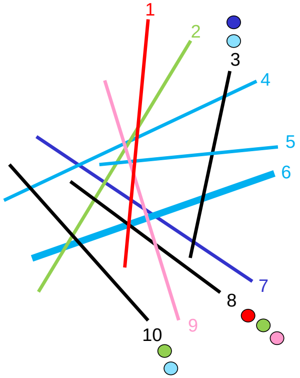
DSatHull Coloring



DSatHull Coloring

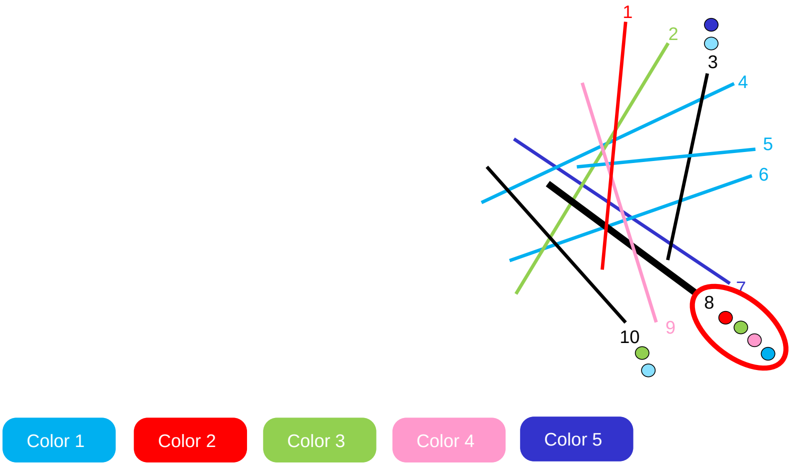
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

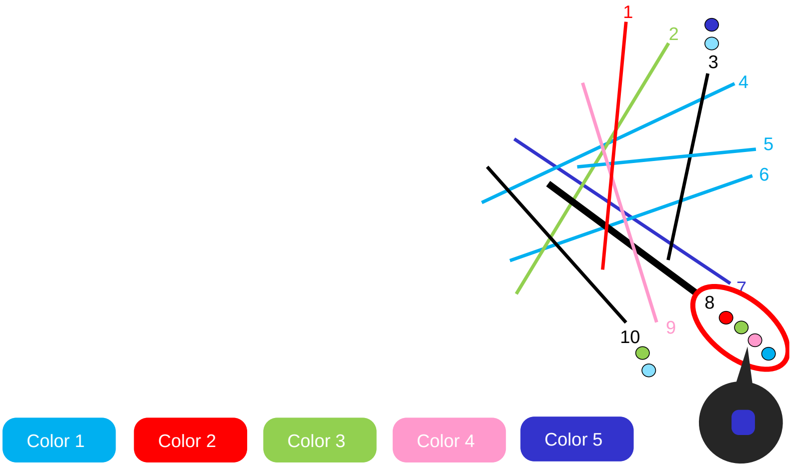


DSatHull Coloring

Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



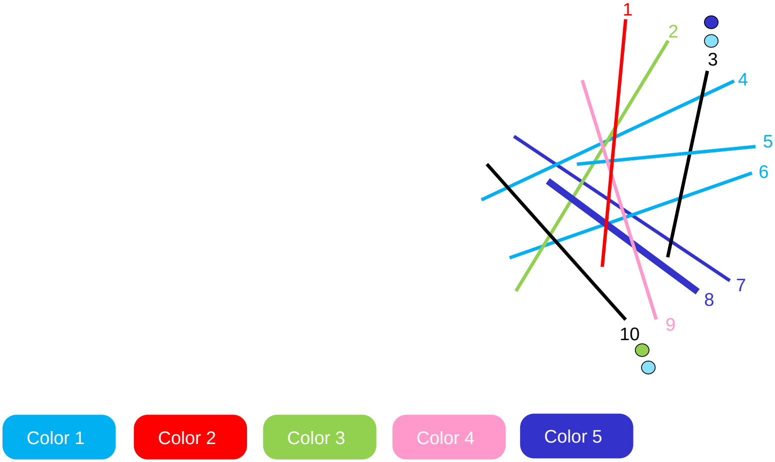
DSatHull Coloring



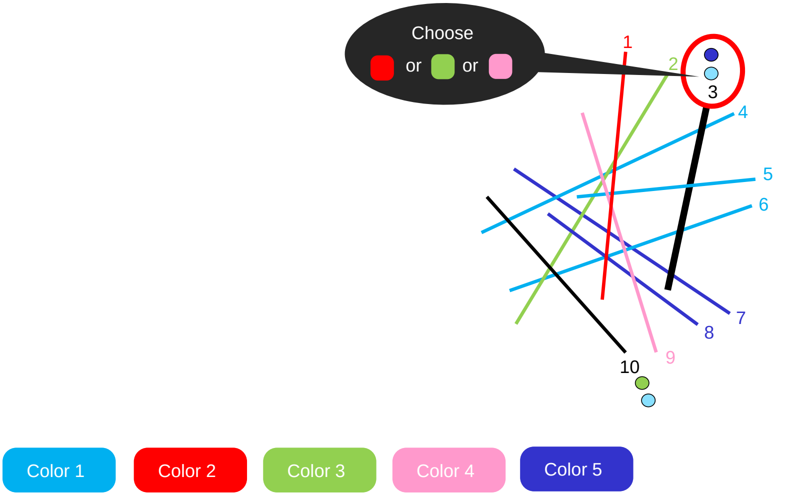
DSatHull Coloring

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

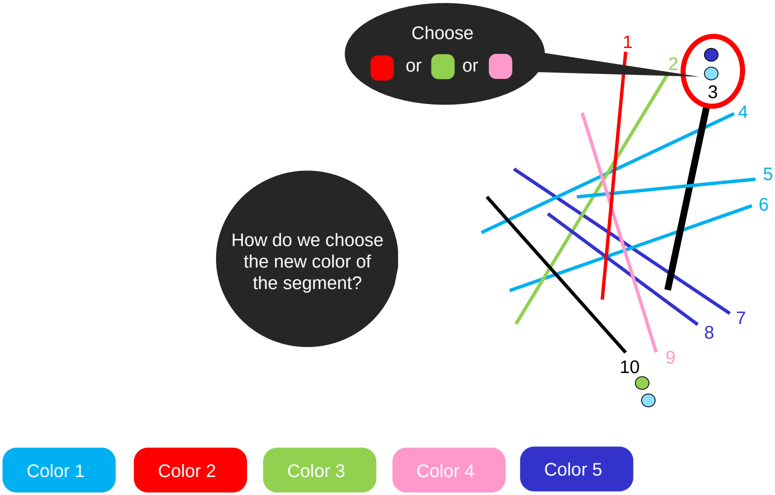


DSatHull Coloring

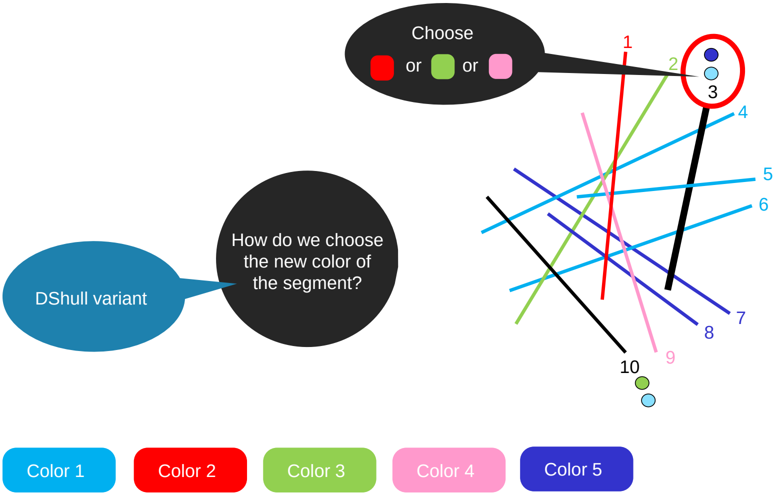


DSatHull Coloring

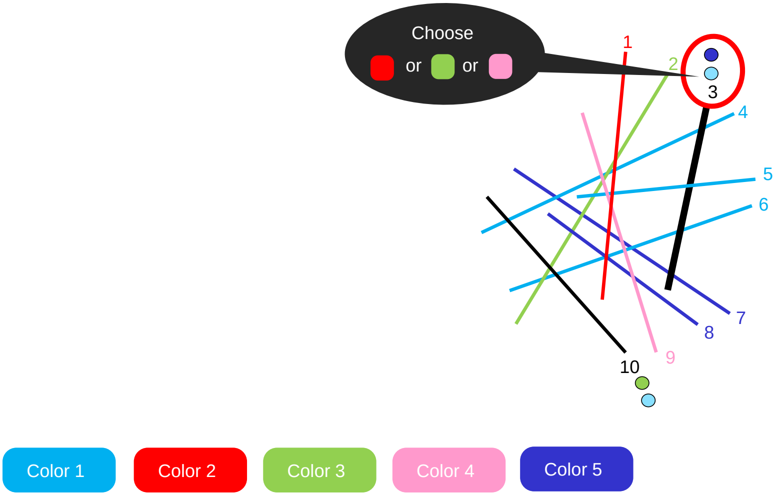
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



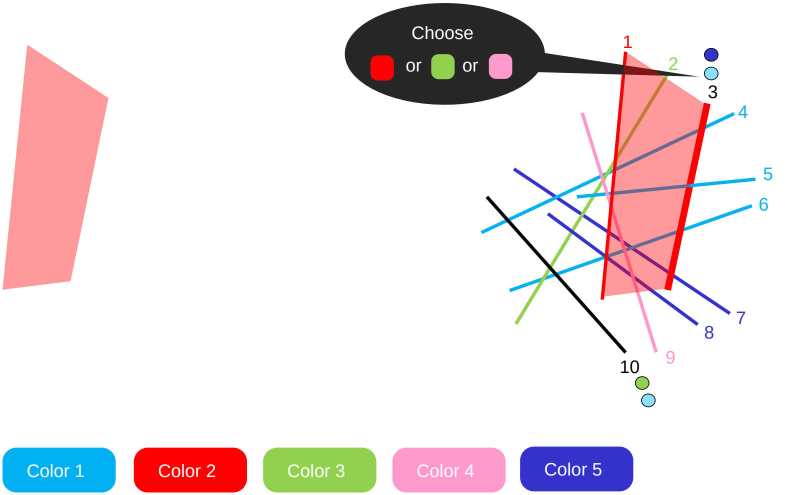
DSatHull Coloring



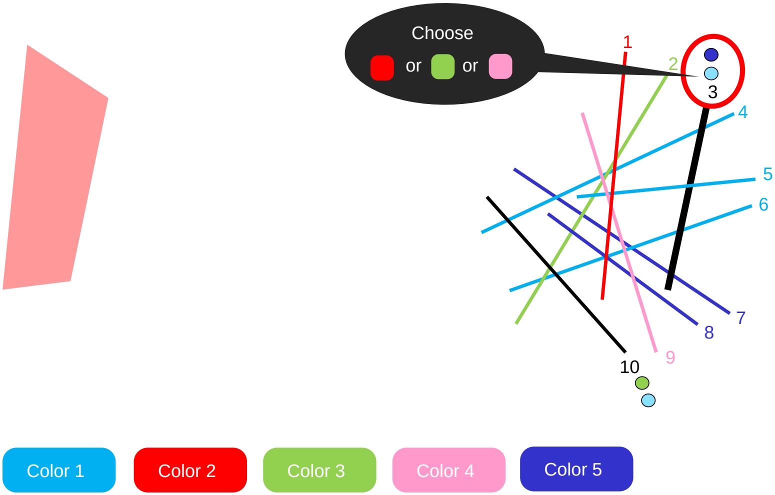
DSatHull Coloring



DSatHull Coloring



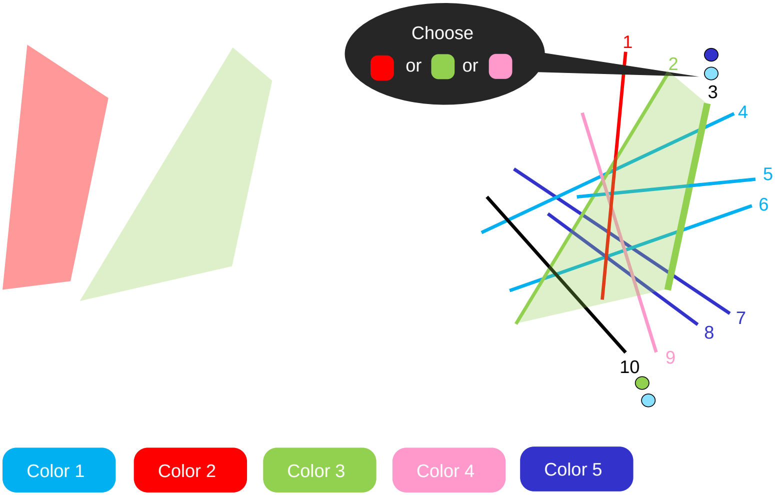
DSatHull Coloring



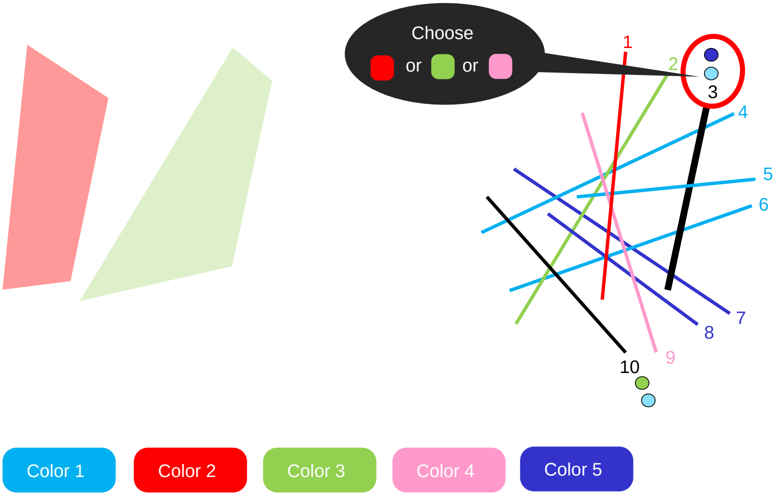
DSatHull Coloring

Shadoks

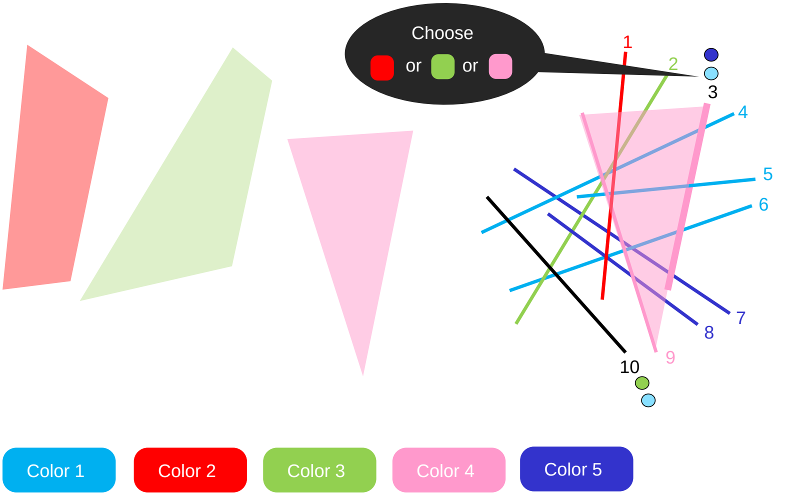
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



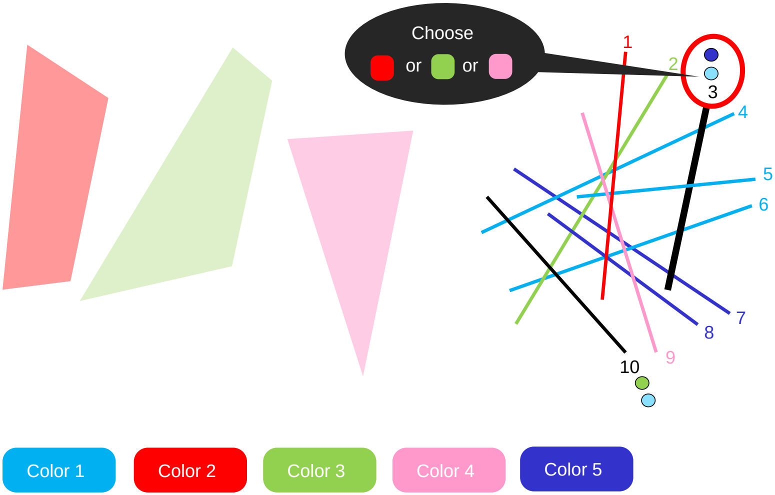
DSatHull Coloring



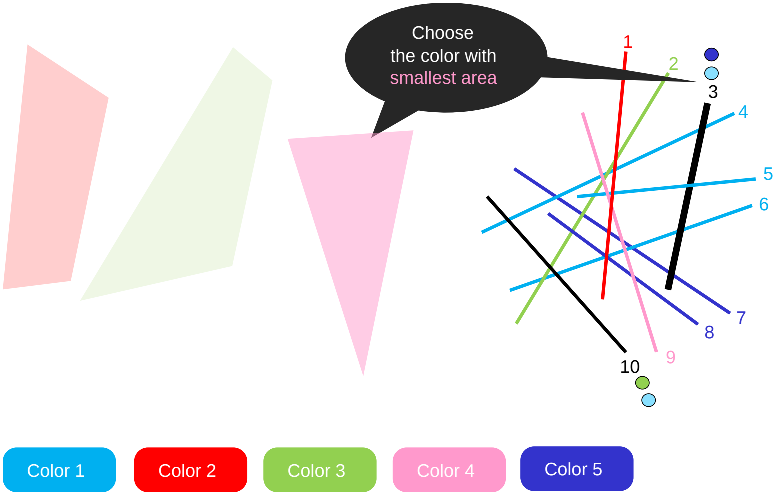
DSatHull Coloring



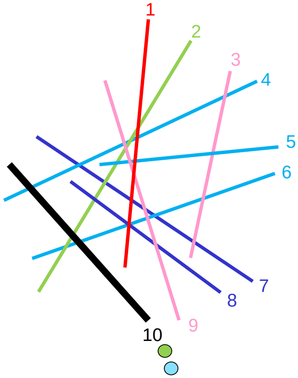
DSatHull Coloring



DSatHull Coloring

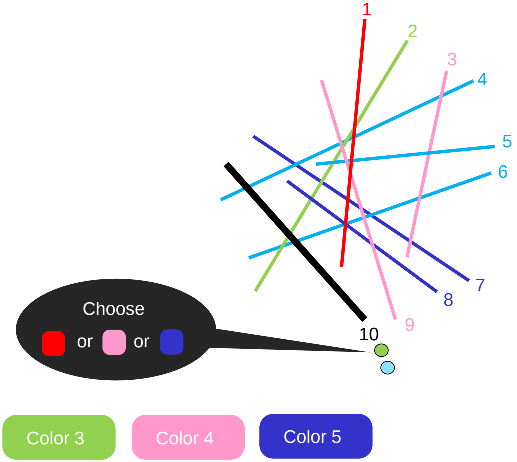


DSatHull Coloring

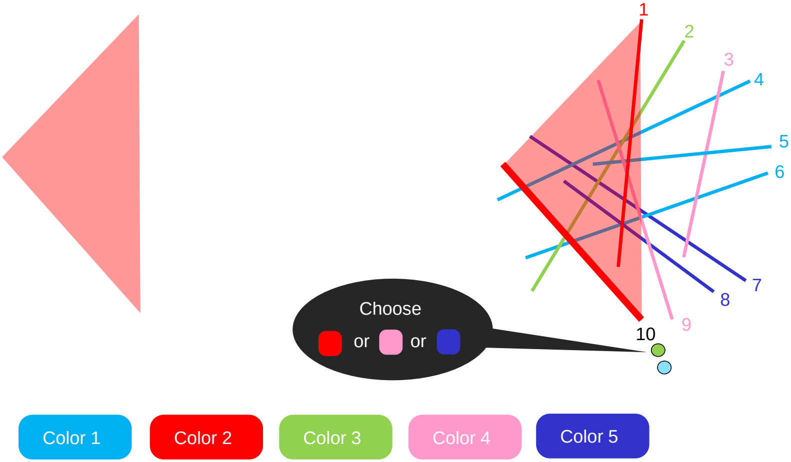


Color 1 | Color 2 | Color 3 | Color 4 | Color 5

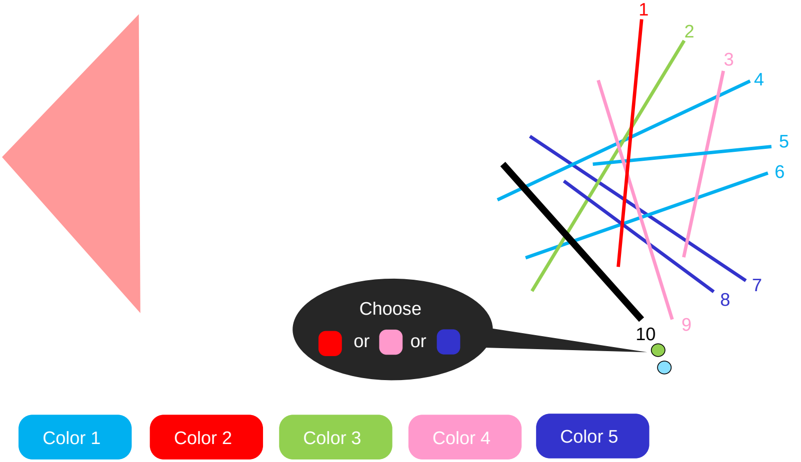
DSatHull Coloring



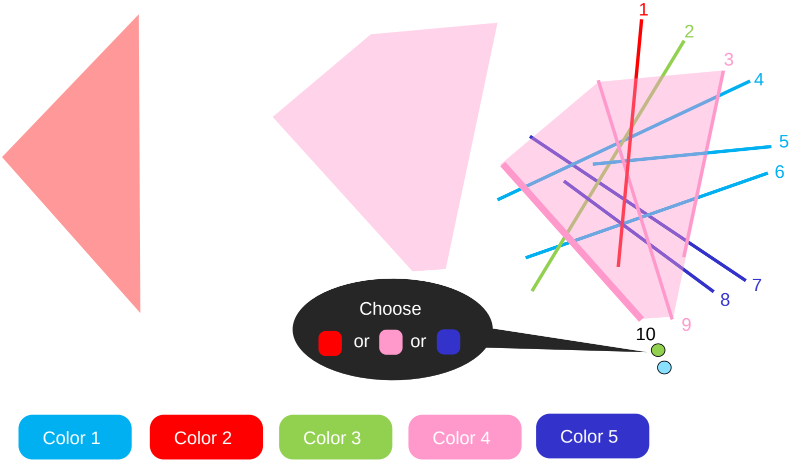
DSatHull Coloring



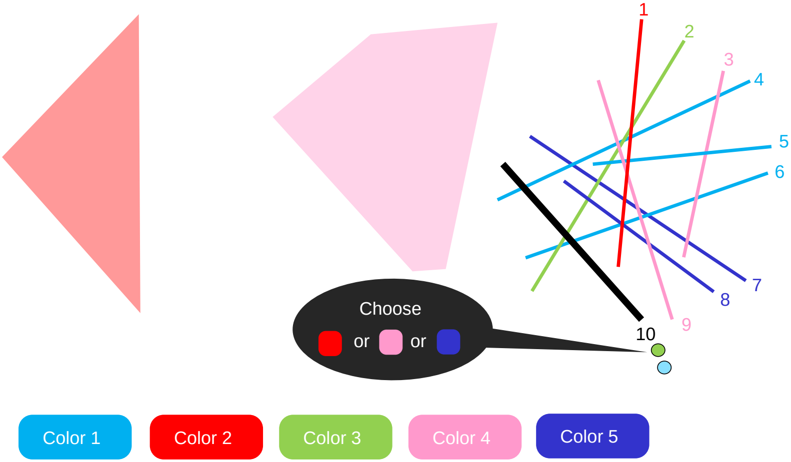
DSatHull Coloring



DSatHull Coloring



DSatHull Coloring



DSatHull Coloring

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

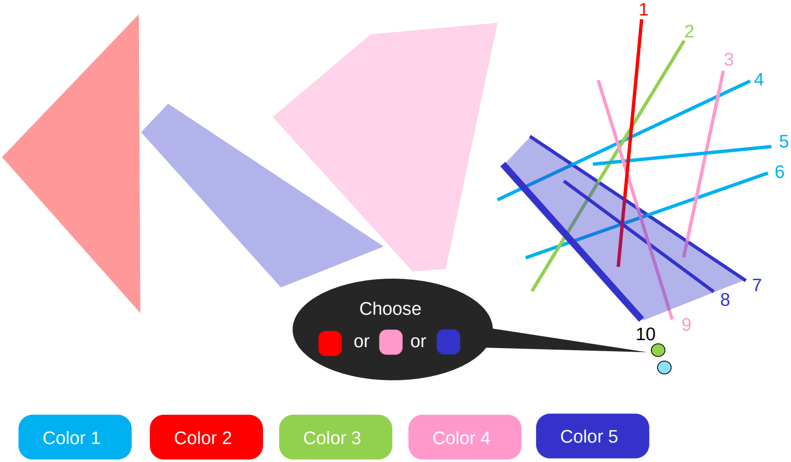
1-IP
Packing
ConvexCover

2-IP
ConvexCover
ConvexPartition

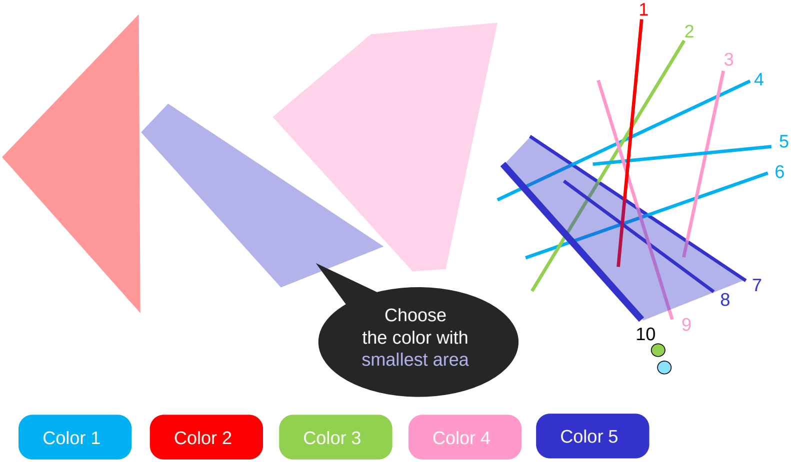
2-Local
MaxPolygon
Packing

2-Conflict
Robots
Coloring

Some results



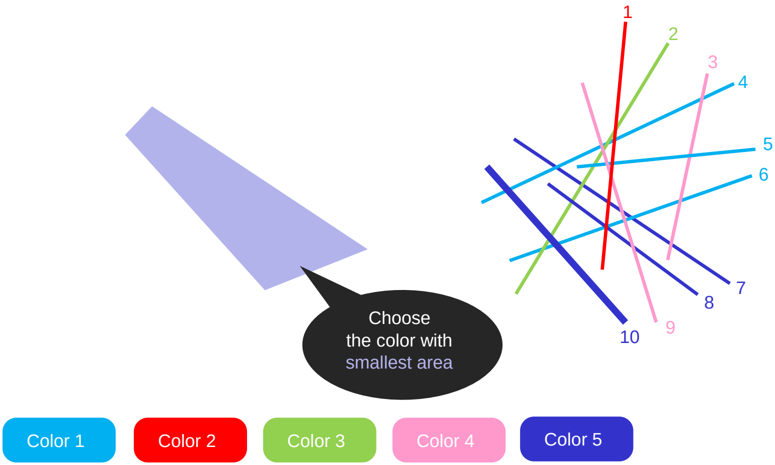
DSatHull Coloring



DSatHull Coloring

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

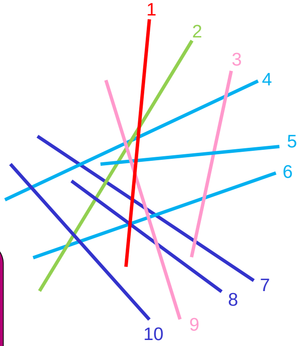


DSatHull Coloring

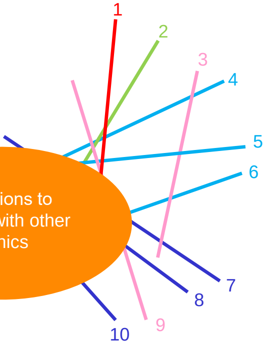
Greedy algorithms

Fast and simple (easy and fast to code)

Can be randomized to provide many solutions



DSatHull Coloring



… solutions to optimize with other technics

Greedy algorithms

Fast and simple (easy and fast to code)

Can be randomized to provide many solutions

Integer Programming (IP)

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Integer Programming (IP)

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Integer programming (IP):

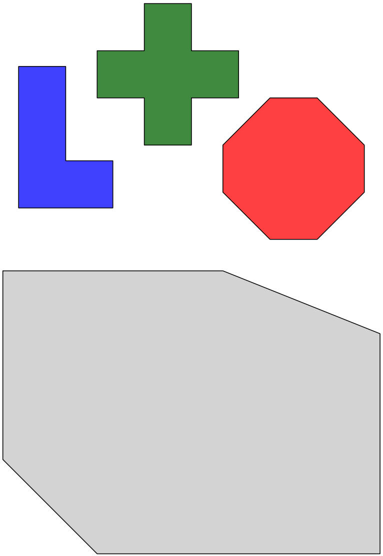
- Variables: Take integer values
- Constraints: A set of linear inequalities
- Objective: A linear function to maximize or minimize

Solvers:

- Commercial solvers are quite efficient (CPLEX, Gurobi...)
- Some open source alternatives (GLPK, COIN-OR...)
- Optimal solutions or small gap are guaranteed
- Cannot solve huge problems

Packing as Independent Set

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

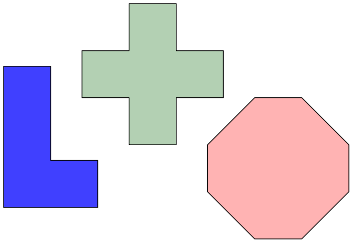


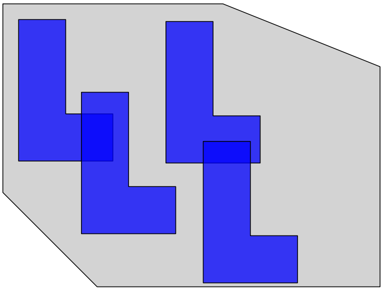
Packing as Independent Set

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing





Packing as Independent Set

Shadoks

Introduction
- Two Phases

1-Greedy
- MaxPolygon
- Packing
- Coloring

1-IP
- Packing
- ConvexCover

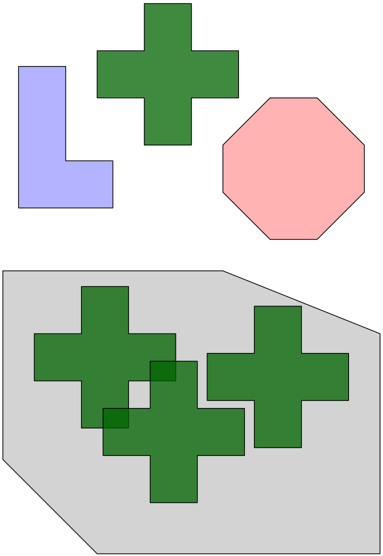
2-IP
- ConvexCover
- ConvexPartition

2-Local
- MaxPolygon
- Packing

2-Conflict
- Robots
- Coloring

Some results

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

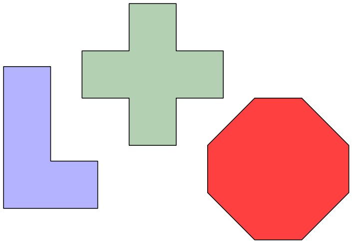


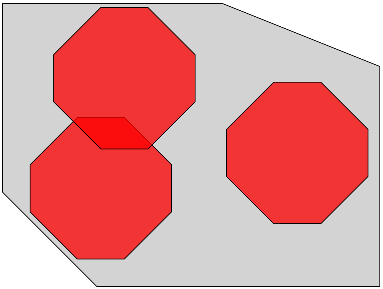
Packing as Independent Set

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing



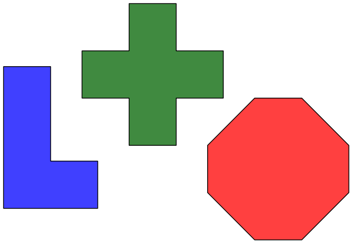


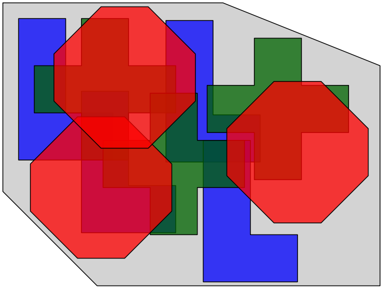
Packing as Independent Set

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing



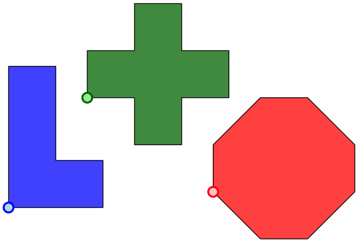


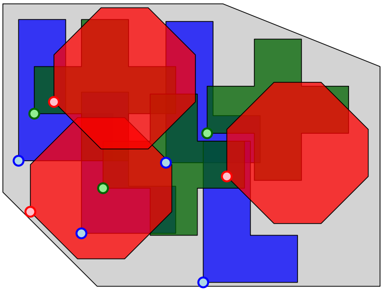
Packing as Independent Set

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

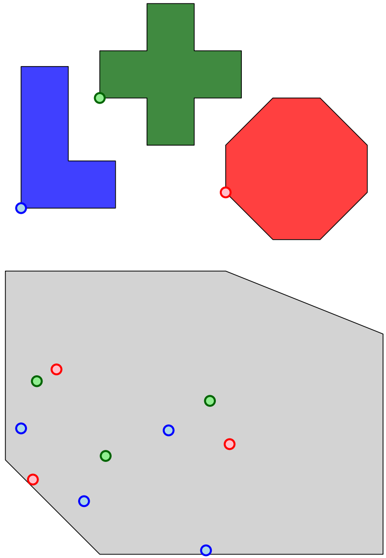
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing





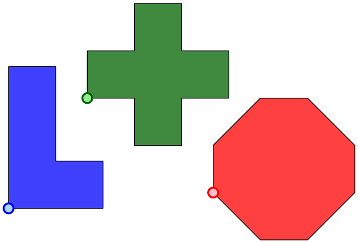
Packing as Independent Set

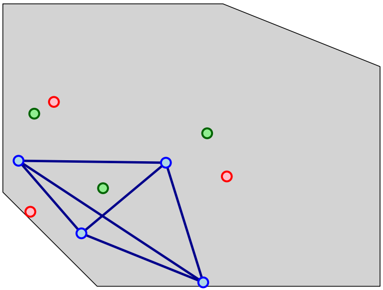
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing



Packing as Independent Set

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing



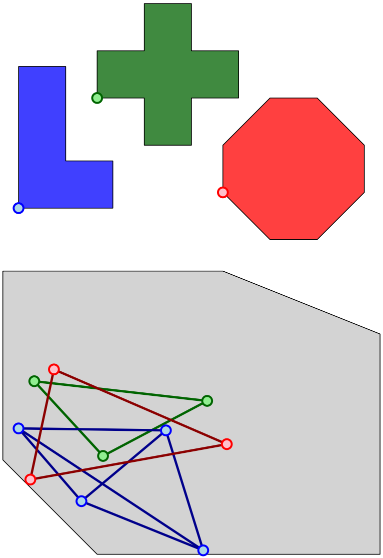


Packing as Independent Set

Shadoks

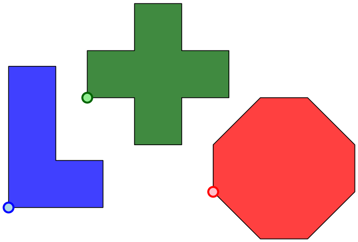
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

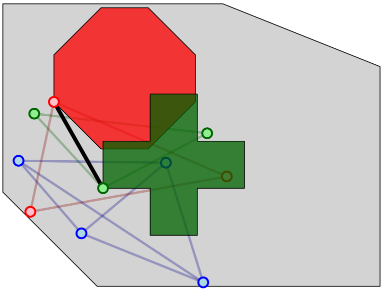
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing



Packing as Independent Set

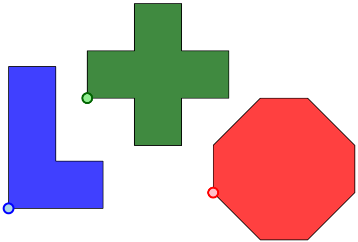
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

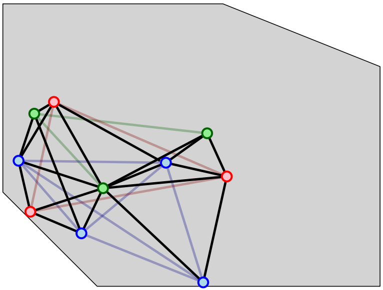




Packing as Independent Set

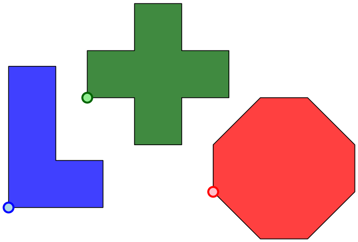
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

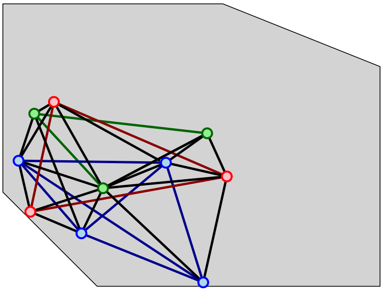




Packing as Independent Set

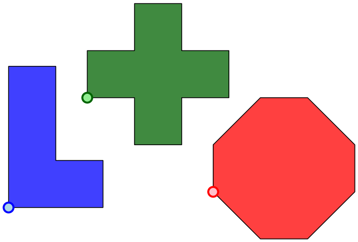
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

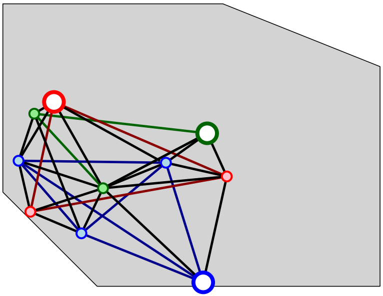




Packing as Independent Set

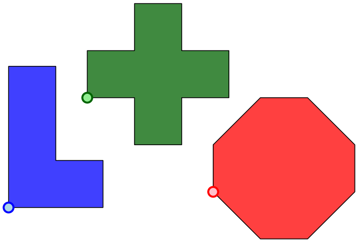
- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing

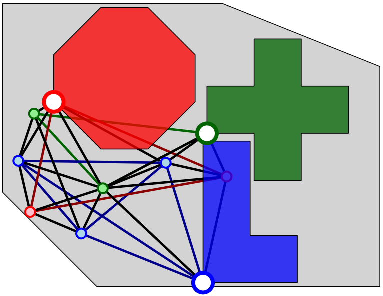




Packing as Independent Set

- Goal: Reduce a geometric problem to a combinatorial problem
- Create some random translations of each item inside the container
- Each translated item is a vertex
- A clique for translations of the same item
- Edges for overlapping translations
- Solve maximum independent set
- Use the corresponding packing





Maximum Independent Set as Linear Programming

Shadoks

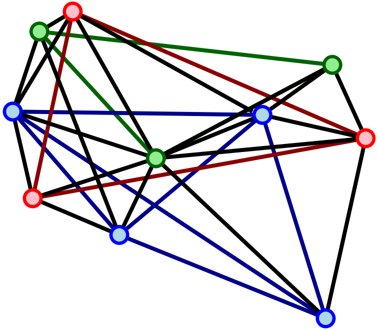
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Variables: vertices
  0: not in the solution
  1: in the solution
- Objective: Maximize sum of variables
- Constraints: edge uv becomes $u + v \leq 1$
- Better constraint for clique C

$$\sum_{v \in C} v \leq 1$$

- And for k copies of the same item

$$\sum_{v \in C} v \leq k$$



Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

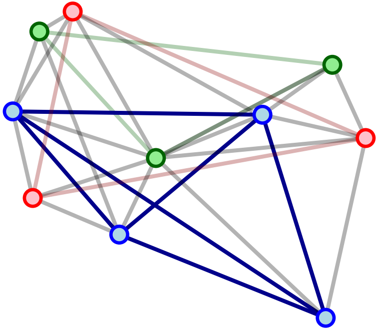
Maximum Independent Set as Linear Programming

- Variables: vertices
  0: not in the solution
  1: in the solution
- Objective: Maximize sum of variables
- Constraints: edge uv becomes $u + v \leq 1$
- Better constraint for clique C

$$\sum_{v \in C} v \leq 1$$

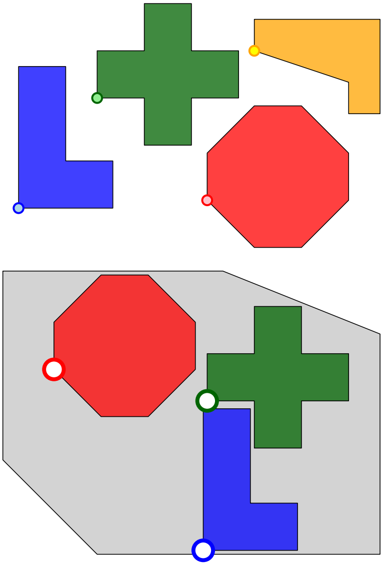
- And for k copies of the same item

$$\sum_{v \in C} v \leq k$$



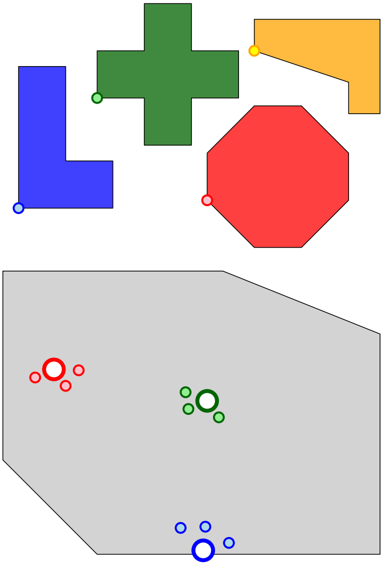
IP Limitation

- CPLEX can only handle ~ 1000 vertices
- Not enough copies of each item
- Solution: We repeat with some translations near the previous solution
- Also random translations of every item
- If we are lucky, we pack more items



IP Limitation

- CPLEX can only handle ~ 1000 vertices
- Not enough copies of each item
- Solution: We repeat with some translations near the previous solution
- Also random translations of every item
- If we are lucky, we pack more items

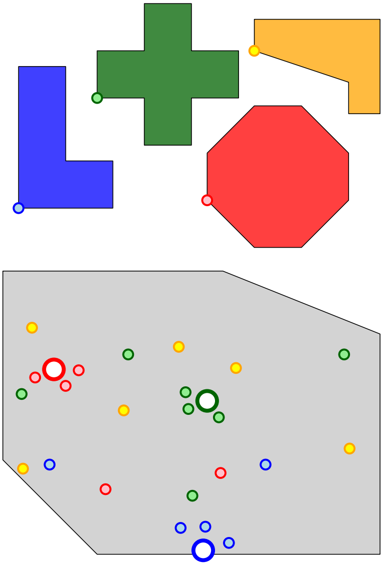


IP Limitation

Shadoks

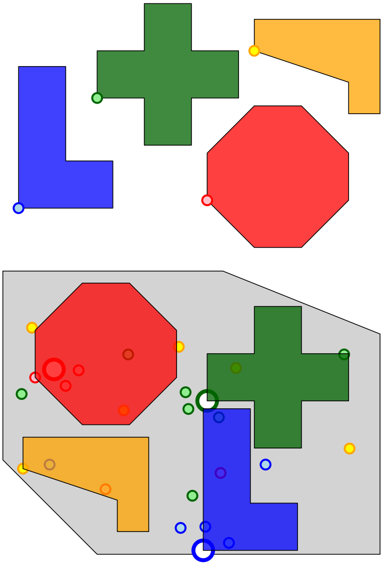
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- CPLEX can only handle ~ 1000 vertices
- Not enough copies of each item
- Solution: We repeat with some translations near the previous solution
- Also random translations of every item
- If we are lucky, we pack more items



IP Limitation

- CPLEX can only handle ~ 1000 vertices
- Not enough copies of each item
- Solution: We repeat with some translations near the previous solution
- Also random translations of every item
- If we are lucky, we pack more items



Convex Covering in Two Steps

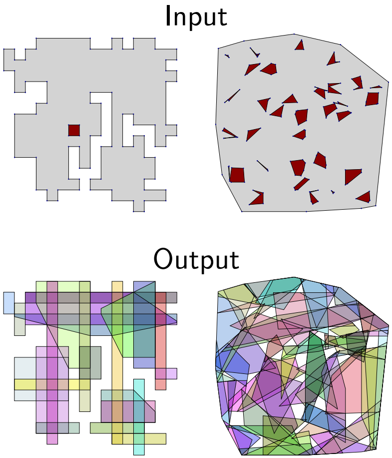
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- We want to cover a polygon (with holes) using convex polygons
- Goal: Reduce a geometric problem to a combinatorial problem

1. Create a collection of many large convex polygons
2. Find a small subset of the collection

- We show how to solve step 2

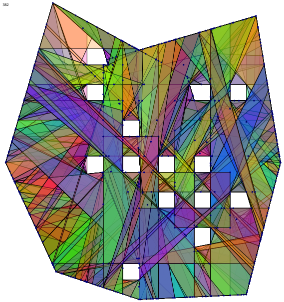


Convex Covering in Two Steps

- We want to cover a polygon (with holes) using convex polygons
- Goal: Reduce a geometric problem to a combinatorial problem

1. Create a collection of many large convex polygons
2. Find a small subset of the collection

- We show how to solve step 2



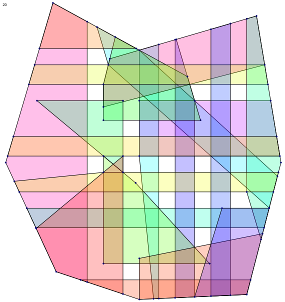
382 convex polygons

Convex Covering in Two Steps

- We want to cover a polygon (with holes) using convex polygons
- Goal: Reduce a geometric problem to a combinatorial problem

1. Create a collection of many large convex polygons
2. Find a small subset of the collection

- We show how to solve step 2

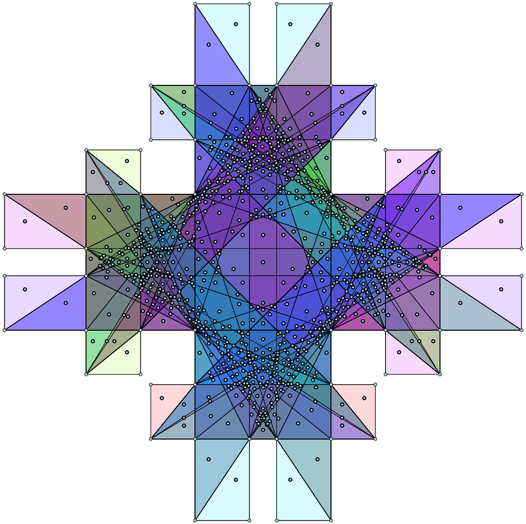


20 convex polygons

Convex Covering as Set Cover

$\mathcal{C}$: Convex polygons from phase 1
P: Instance polygon with holes

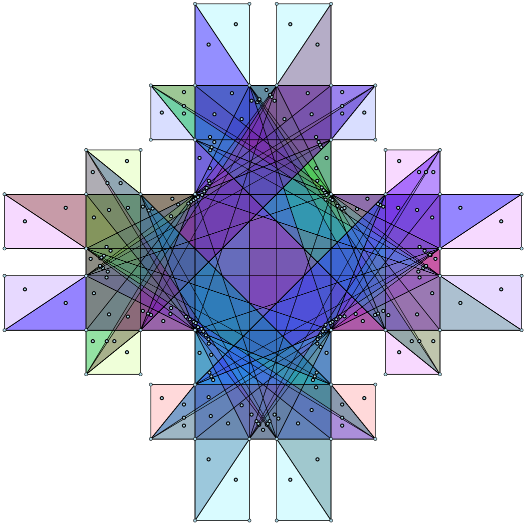
- $(\mathcal{C}, P)$ define a set system
- P has infinitely many points
- First attempt: reduce P to a quadratic number of witnesses, one point per arrangement cell
- Too many witnesses!
- Building the arrangement is slow!



1009 witnesses for 82 convex polygons

Vertex Witnesses

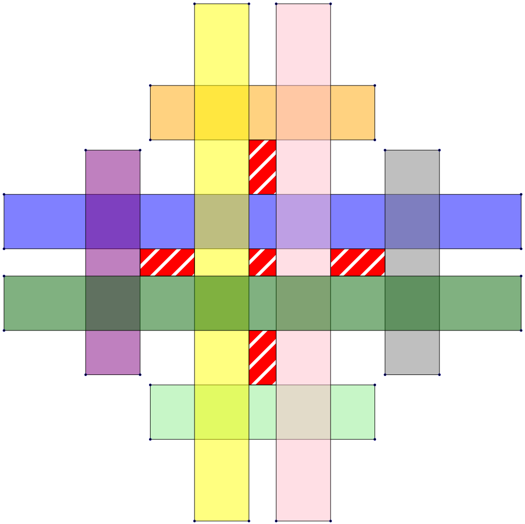
- Solution: only place witnesses near vertices of P
- Does not guarantee that P is covered
- Two possible fixes:
  - Add a witnesses inside each uncovered area and repeat (generally better, but slower)
  - Cover the uncovered area using some quick heuristic (faster and sometimes better)



200 witnesses for 82 convex polygons

Vertex Witnesses

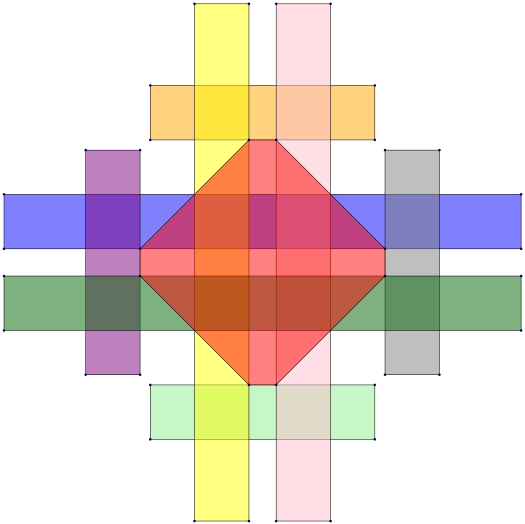
- Solution: only place witnesses near vertices of P
- Does not guarantee that P is covered
- Two possible fixes:
  - Add a witnesses inside each uncovered area and repeat (generally better, but slower)
  - Cover the uncovered area using some quick heuristic (faster and sometimes better)



5 uncovered regions 8 convex polygons

Vertex Witnesses

- Solution: only place witnesses near vertices of P
- Does not guarantee that P is covered
- Two possible fixes:
  - Add a witnesses inside each uncovered area and repeat (generally better, but slower)
  - Cover the uncovered area using some quick heuristic (faster and sometimes better)



0 uncovered regions 9 convex polygons

Solving Combinatorial Set Cover with IP

Use integer programming (IP):

- Each set becomes a binary variable
  0: Not in the solution
  1: In the solution
- Each witness becomes a constraint:
  the sum of the sets that contain it must be at least 1
- Objective: minimize the sum of all variables

- Very fast for small to medium instances
- Solutions often guaranteed optimal
- On some large instances: slow and very bad solutions

Solving Combinatorial Set Cover with IP

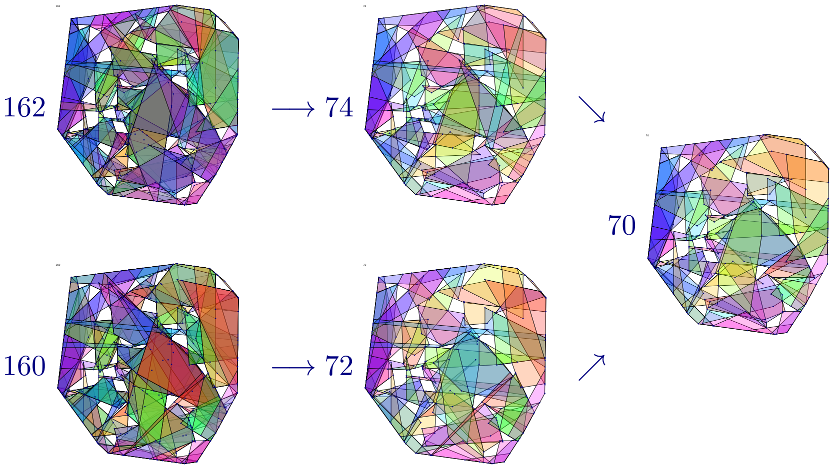
Use integer programming (IP):

- Each set becomes a binary variable
  0: Not in the solution
  1: In the solution
- Each witness becomes a constraint:
  the sum of the sets that contain it must be at least 1
- Objective: minimize the sum of all variables

- Very fast for small to medium instances
- Solutions often guaranteed optimal
- On some large instances: slow and very bad solutions

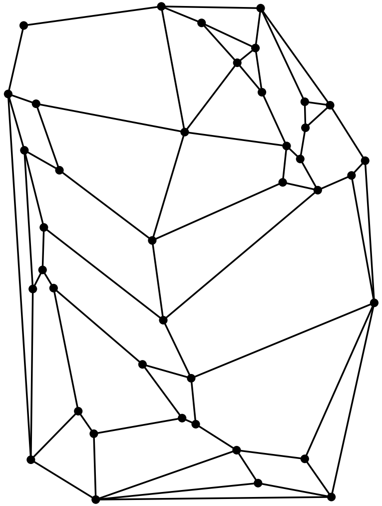
Improving Convex Cover with IP

We use multiple good solutions to get a better one



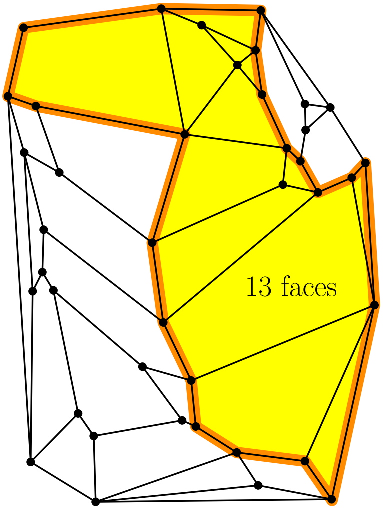
Improving Convex Partition with IP

- Convex partition can easily be modelled as IP
- CPLEX can only solve small instances
- We improve a solution using CPLEX:
  1. Select some adjacent cells
  2. Remove selected cells
  3. Solve the polygon in the selected area
- The initial solution is not very relevant, we may start with a triangulation



Improving Convex Partition with IP

- Convex partition can easily be modelled as IP
- CPLEX can only solve small instances
- We improve a solution using CPLEX:
  1. Select some adjacent cells
  2. Remove selected cells
  3. Solve the polygon in the selected area
- The initial solution is not very relevant, we may start with a triangulation

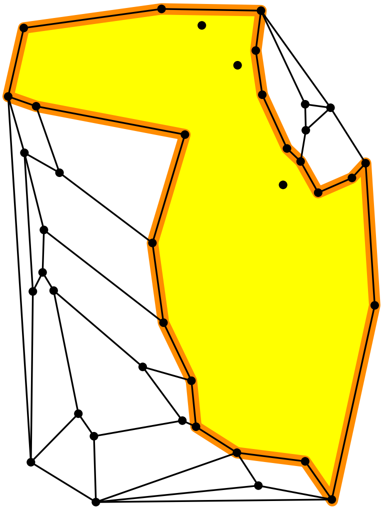


Improving Convex Partition with IP

Shadoks

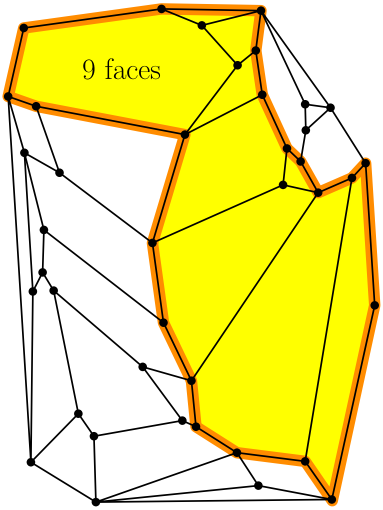
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

- Convex partition can easily be modelled as IP
- CPLEX can only solve small instances
- We improve a solution using CPLEX:
  1. Select some adjacent cells
  2. Remove selected cells
  3. Solve the polygon in the selected area
- The initial solution is not very relevant, we may start with a triangulation



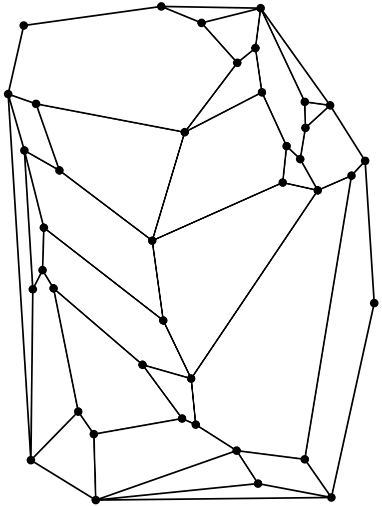
Improving Convex Partition with IP

- Convex partition can easily be modelled as IP
- CPLEX can only solve small instances
- We improve a solution using CPLEX:
  1. Select some adjacent cells
  2. Remove selected cells
  3. Solve the polygon in the selected area
- The initial solution is not very relevant, we may start with a triangulation



Improving Convex Partition with IP

- Convex partition can easily be modelled as IP
- CPLEX can only solve small instances
- We improve a solution using CPLEX:
  1. Select some adjacent cells
  2. Remove selected cells
  3. Solve the polygon in the selected area
- The initial solution is not very relevant, we may start with a triangulation

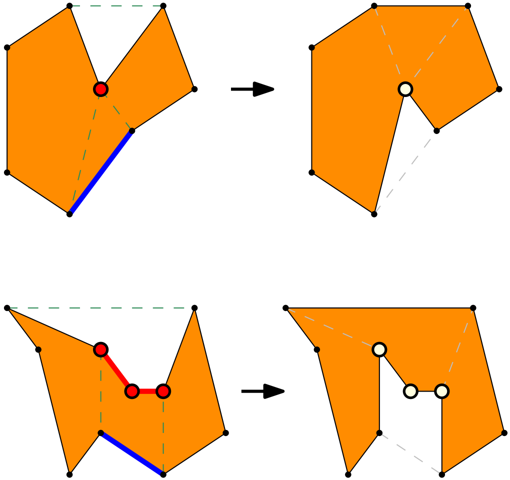


Shadoks

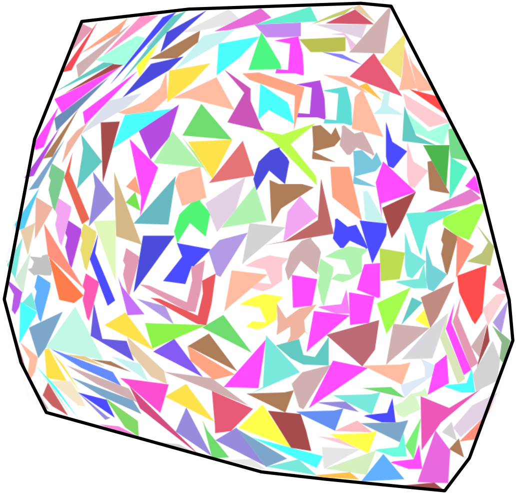
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Improving Maximum Area Polygon with Local Search

- We try to improve many solutions
- Sometimes a worse solution may improve better
- We move one or more vertices in the polygon order in order to increase the area
- For this problem, changes are small

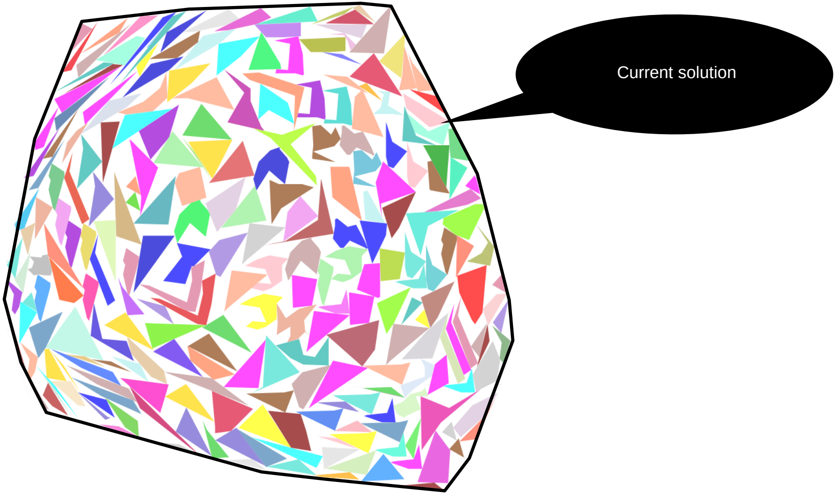


Packing Local Improvement

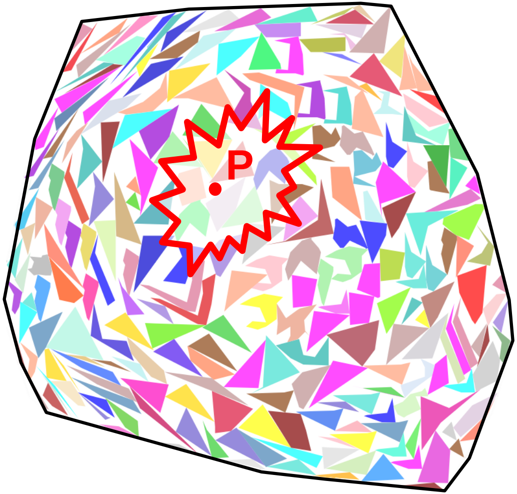


Local Search for Packing

Packing Local Improvement

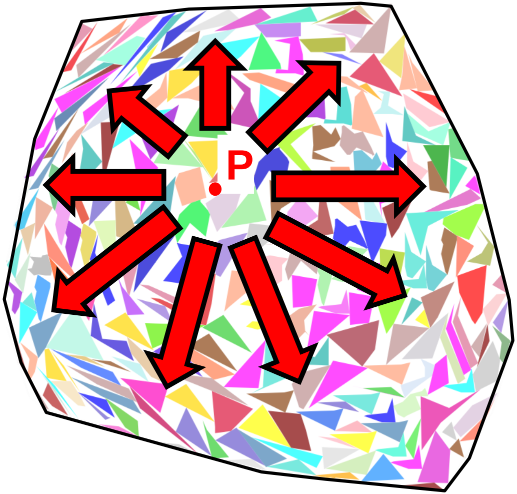


Packing Local Improvement



1) Choose **a random point P** in the solution

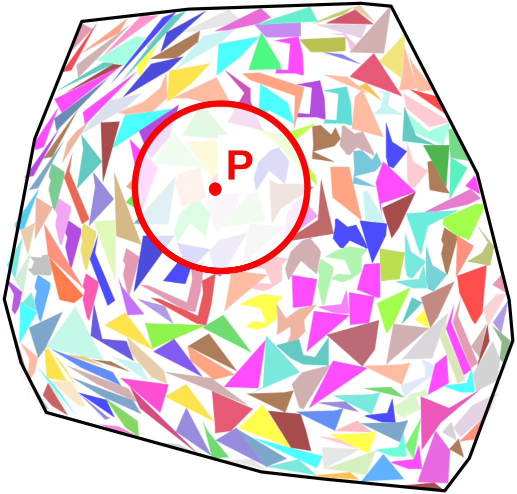
Packing Local Improvement



1) Choose **a random point P** in the solution

2) Push every packed item away from **P**
(from the farthest to the closest)

Packing Local Improvement

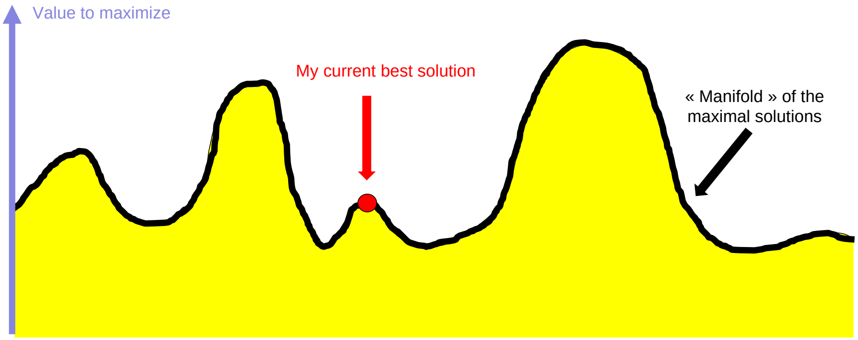


1) Choose **a random point P** in the solution

2) Push every packed item away from **P**
(from the farthest to the closest)

3) Try to pack a new item around **P**
(if it crosses some packed items but there could be a benefit, we pack the new item)

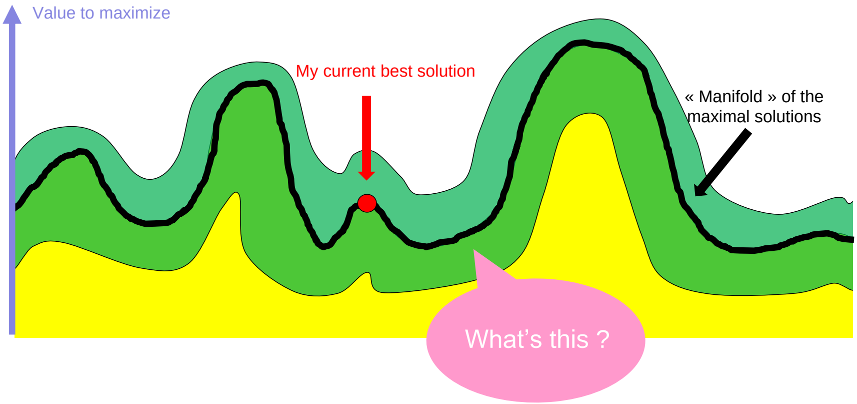
More than Local Search ?



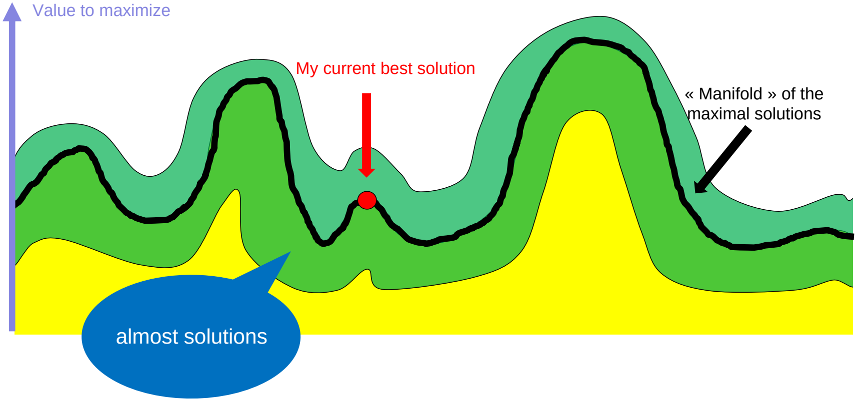
More than Local Search ?

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



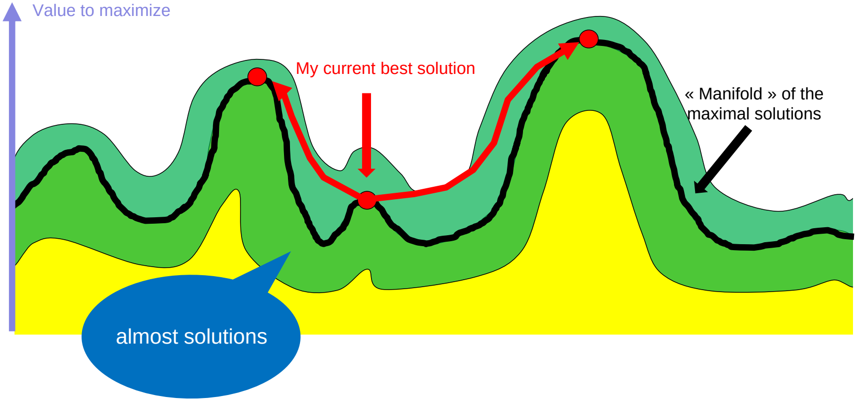
More than Local Search ?



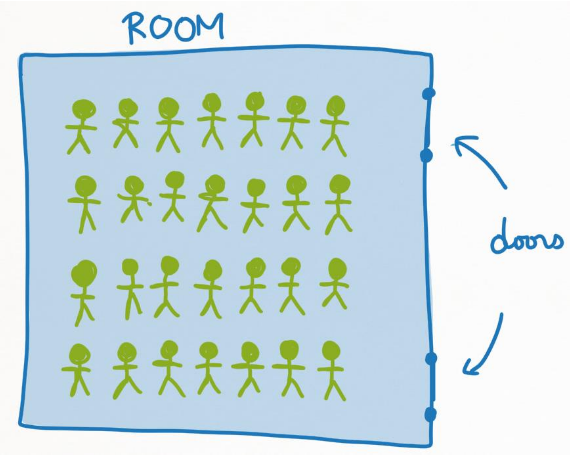
More than Local Search ?

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

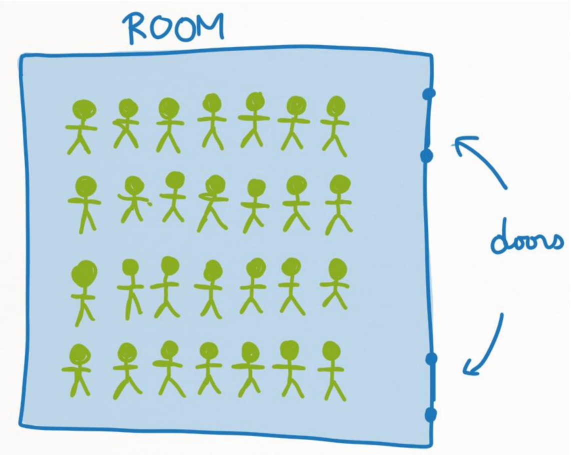


Conflict



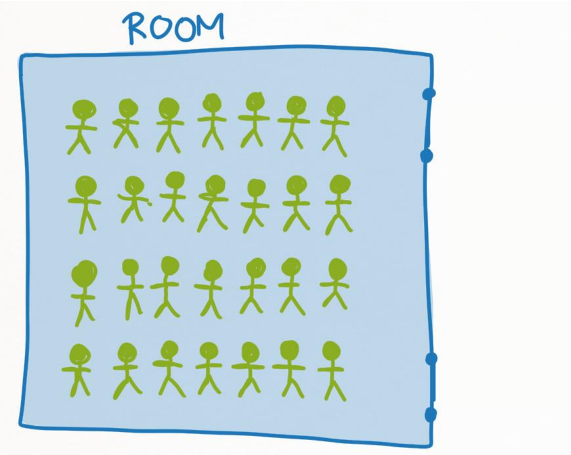
28 agents in a room: find the best consensus of a complex combinatorial problem…

Conflict



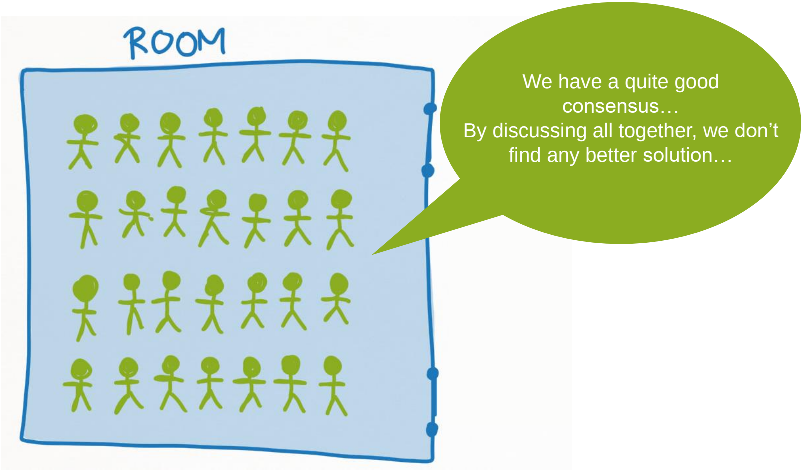
28 agents in a room: find the best consensus of a complex combinatorial problem…

Conflict



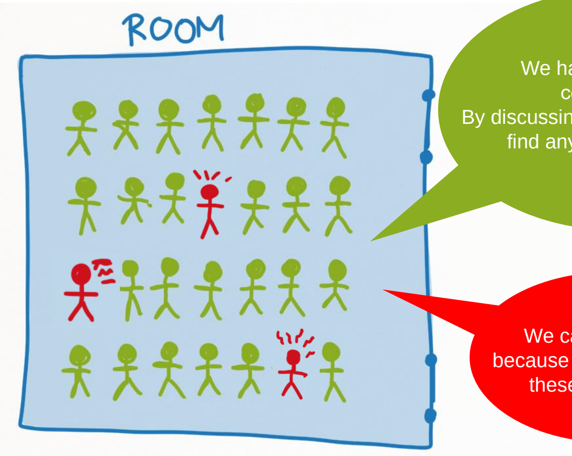
28 agents in a room: find the best consensus of a complex combinatorial problem…

Conflict



28 agents in a room: find the best consensus of a complex combinatorial problem…

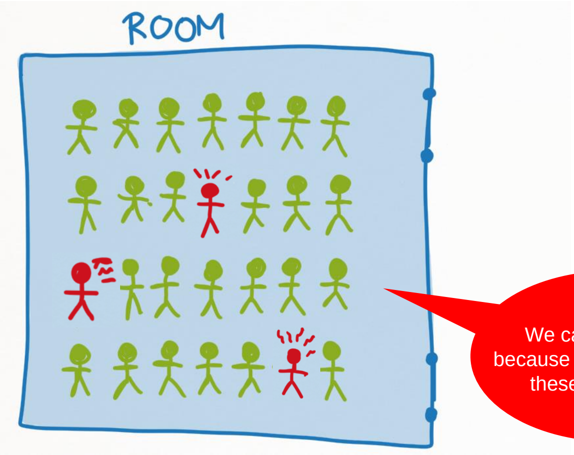
Conflict



We have a quite good consensus…
By discussing all together, we don't find any better solution…

We cannot do better because we are blocked by these three guys!!!!

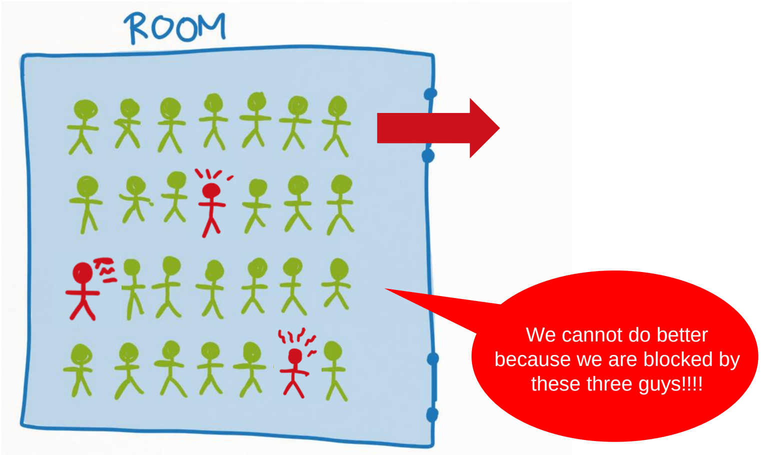
Conflict



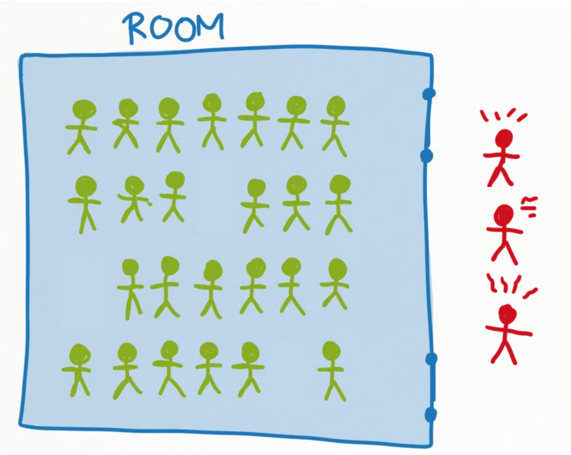
We cannot do better because we are blocked by these three guys!!!!

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

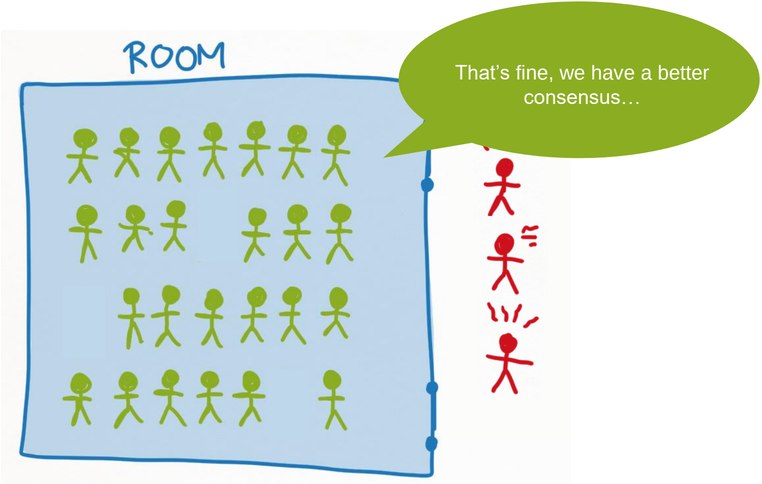
Conflict



Conflict



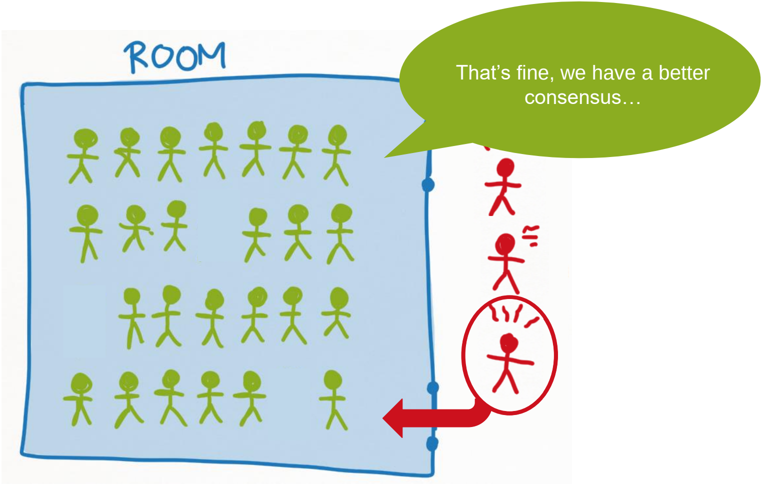
Conflict



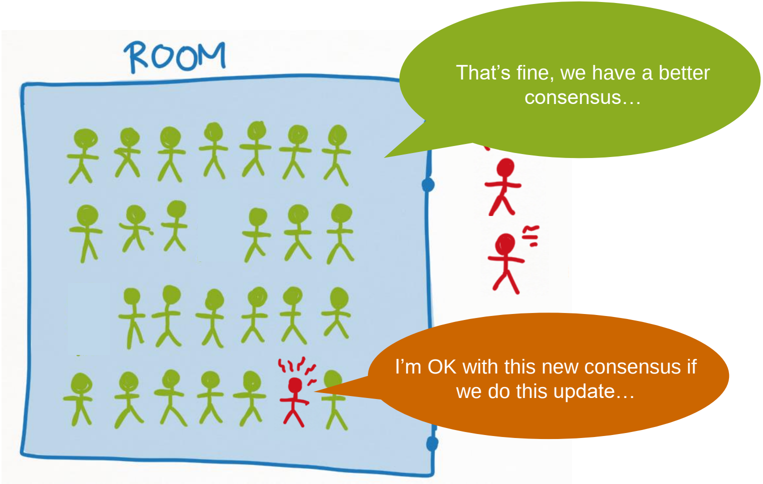
Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Conflict



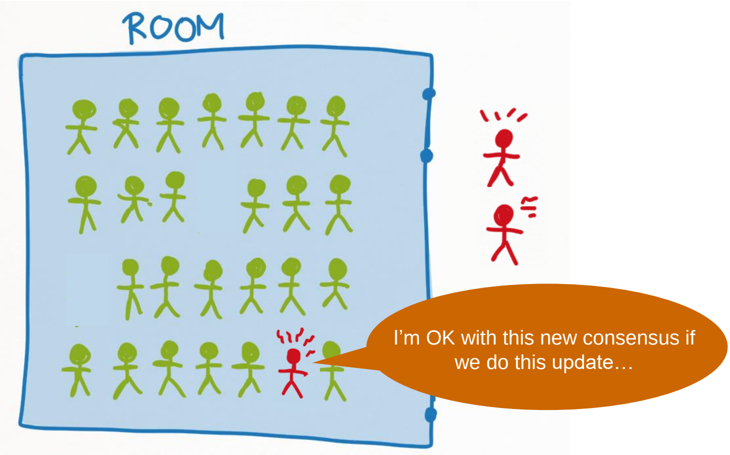
Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

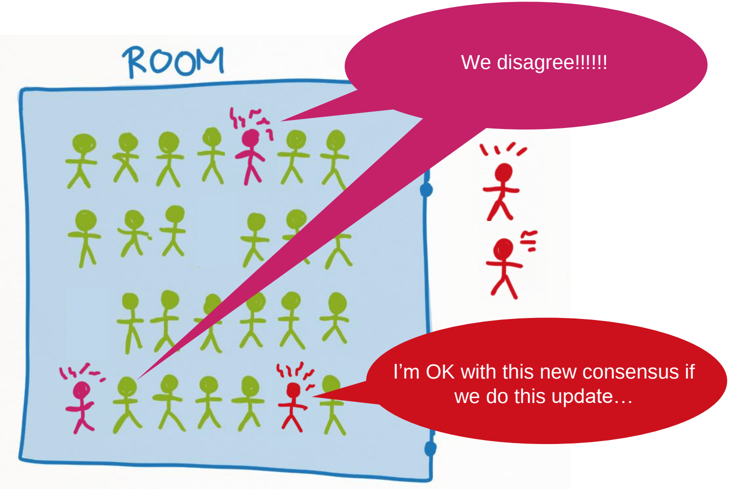
Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

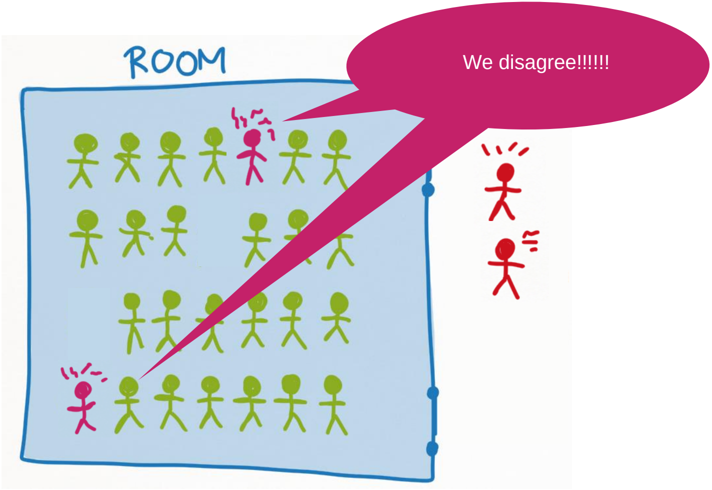


Conflict

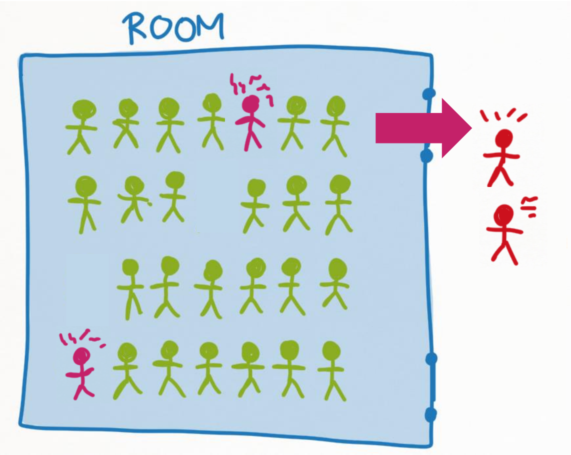

ROOM
We disagree!!!!!!
I'm OK with this new consensus if we do this update…

Conflict

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



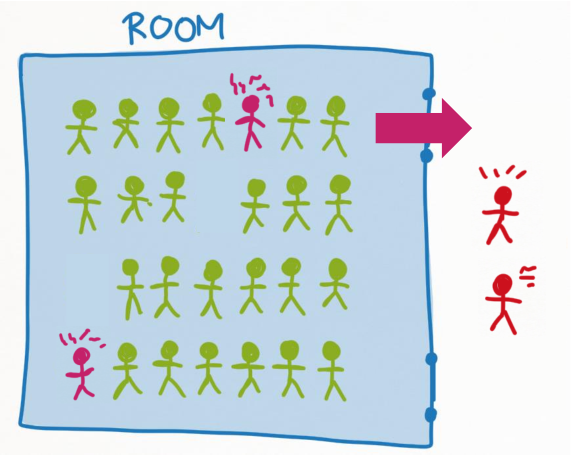
Conflict



Conflict

Shadoks

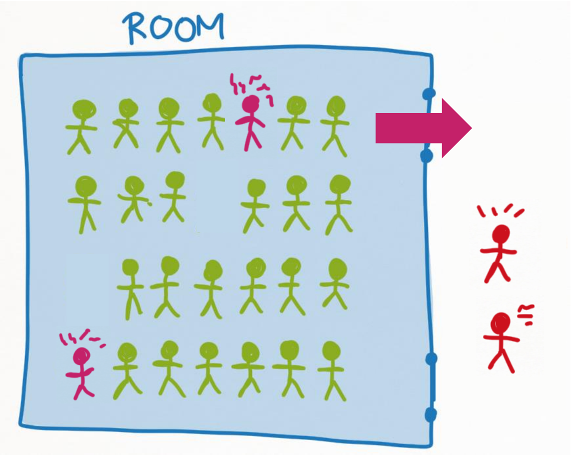
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



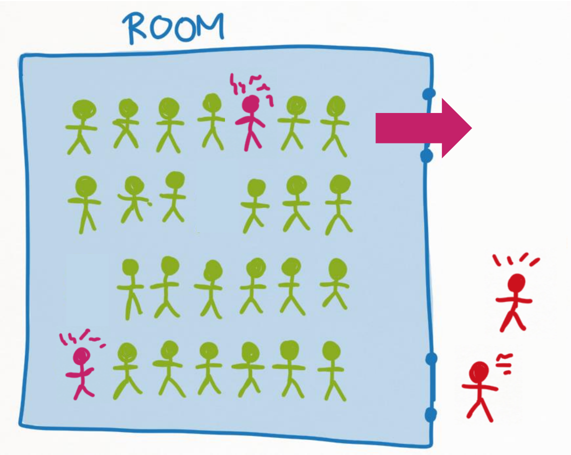
Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



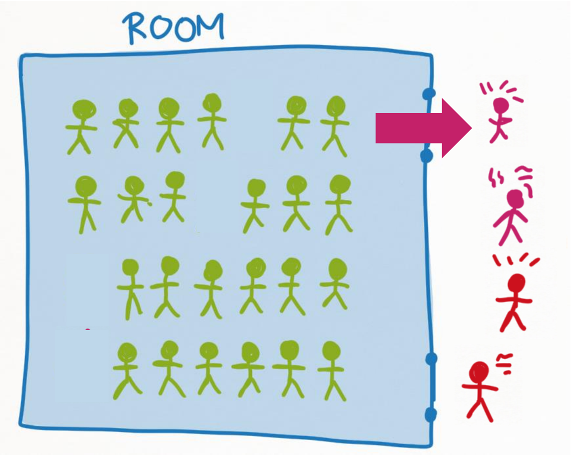
Conflict



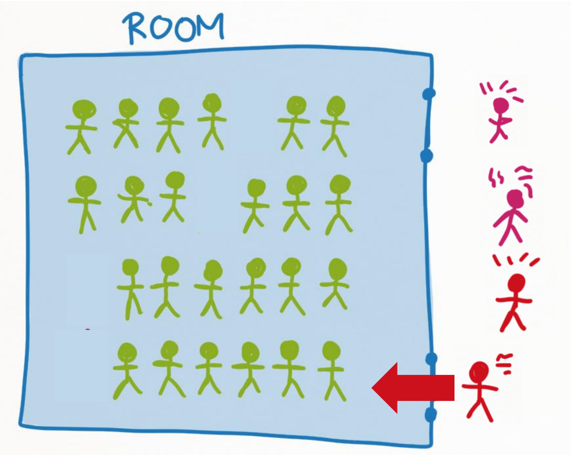
Conflict

Shadoks

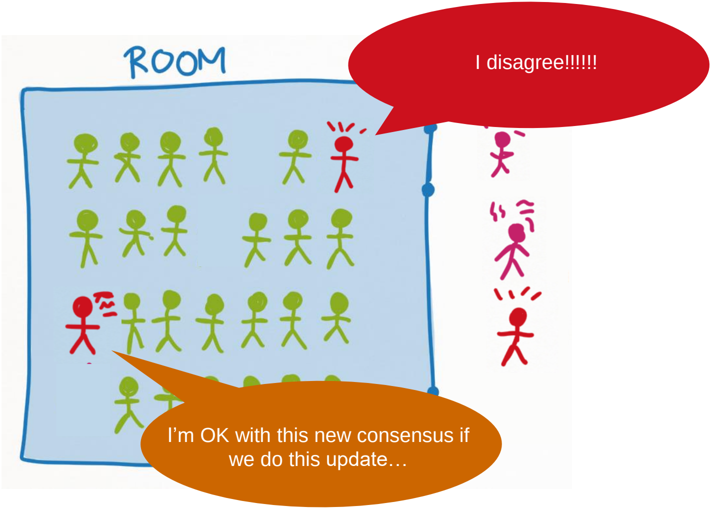
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Conflict



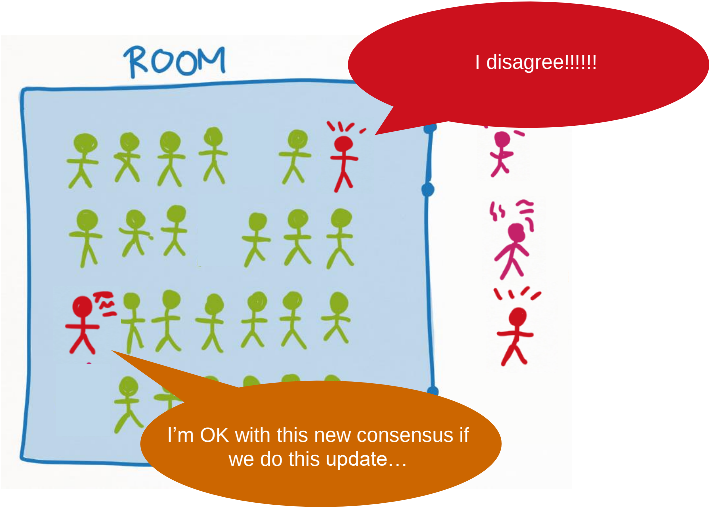
Conflict


ROOM
I disagree!!!!!!
I'm OK with this new consensus if we do this update…

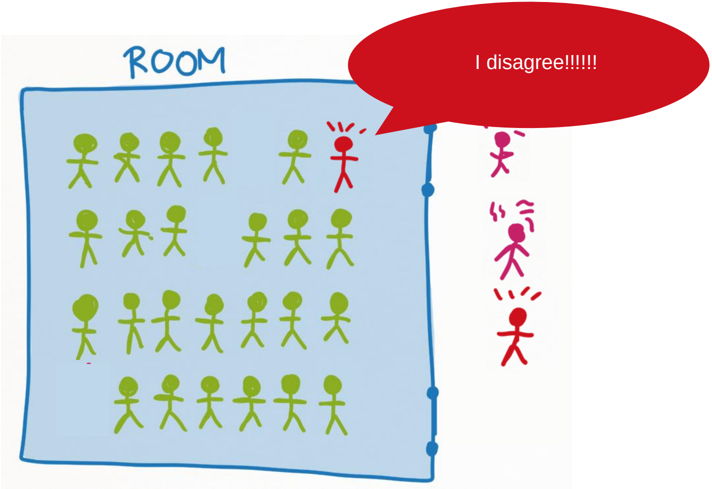
Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

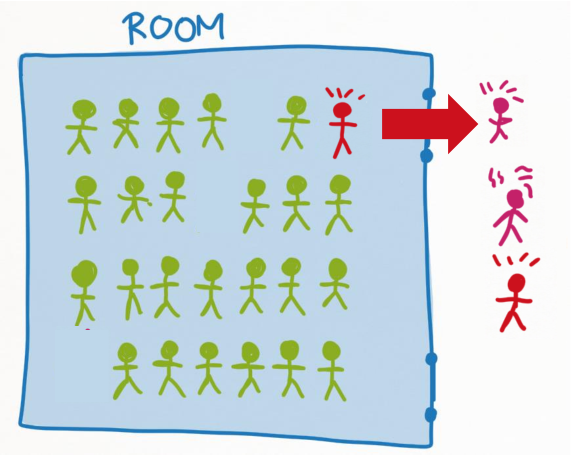


Conflict



Conflict

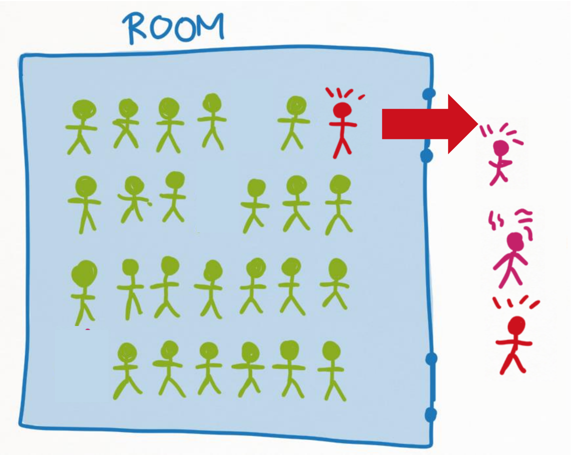
Shadoks
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

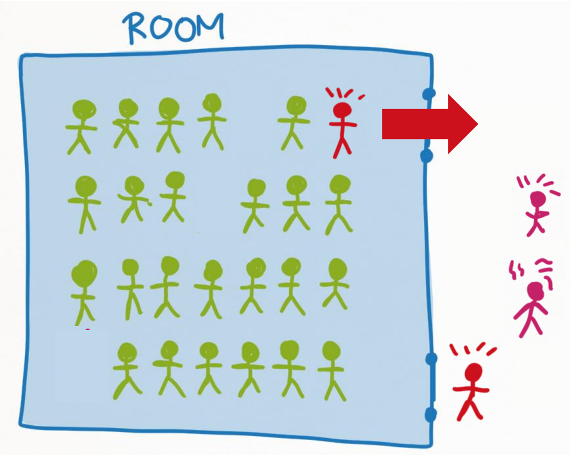


28 agents in a room: find the best consensus to a complex combinatorial problem…

Conflict

Shadoks

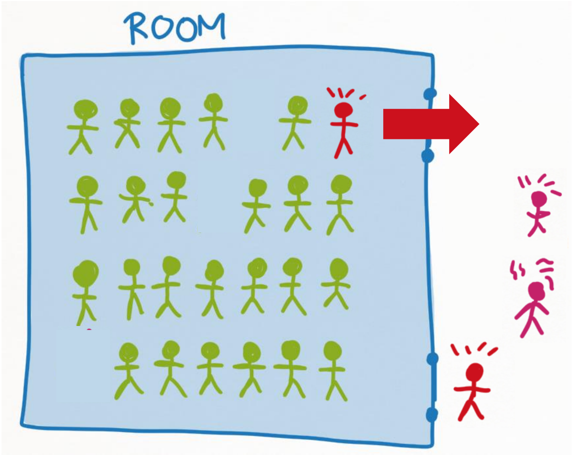
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Conflict

Shadoks

Introduction
- Two Phases

1-Greedy
- MaxPolygon
- Packing
- Coloring

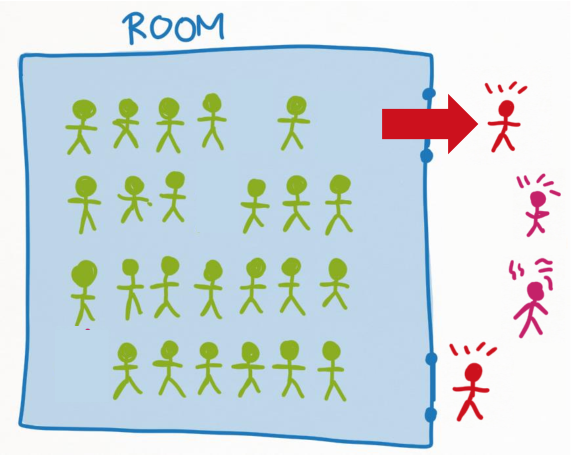
1-IP
- Packing
- ConvexCover

2-IP
- ConvexCover
- ConvexPartition

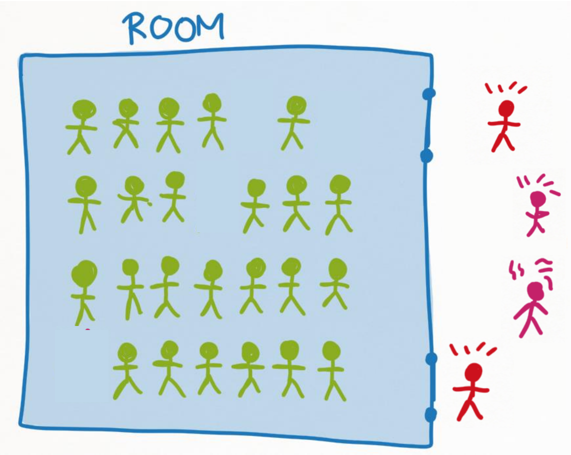
2-Local
- MaxPolygon
- Packing

2-Conflict
- Robots
- Coloring

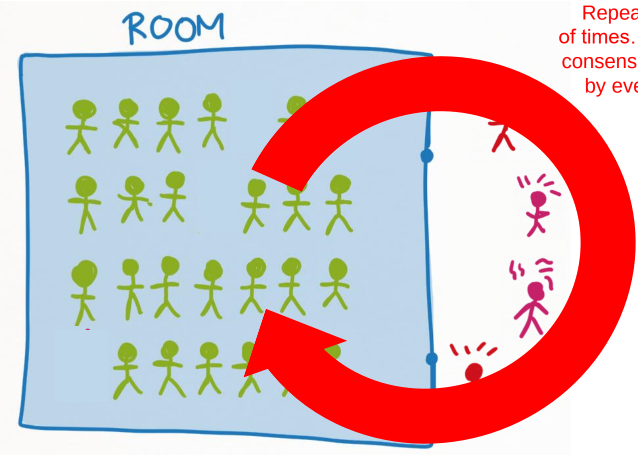
Some results



Conflict



Conflict



Repeat thousands of times… until the new consensus is accepted by every agent….

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

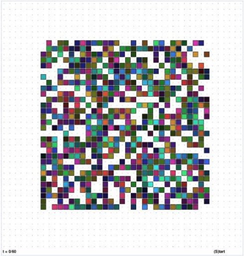
1-IP
Packing
ConvexCover

2-IP
ConvexCover
ConvexPartition

2-Local
MaxPolygon
Packing

2-Conflict
Robots
Coloring

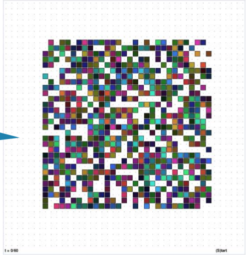
Some results



What can we do to optimize a solution ?

Conflict

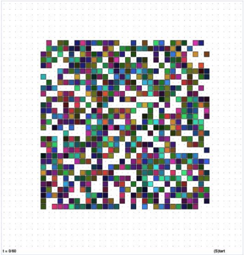
Solution of makespan 61



Conflict

Solution of makespan 61

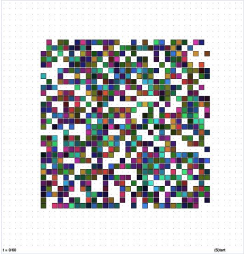
Remove the robots arriving at their target at t=61



All the other robots arrive at t≤60

Conflict

Solution of makespan 61

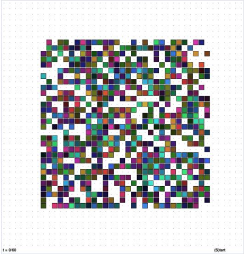


Remove the robots arriving at their target at t=61

- Put the removed robots in a queue Q

Conflict

Solution of makespan 61



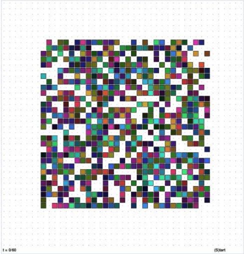
Remove the robots arriving at their target at t=61

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Conflict

Solution of makespan 61



Remove the robots arriving at their target at t=61

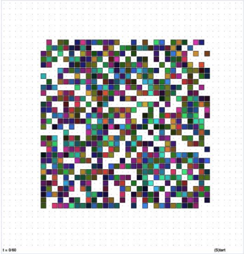
- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

Solution of makespan 61



Remove the robots arriving at their target at t=61

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

Conflict optimizer

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

Loops

robot r_0 excludes robot r_1
robot r_1 excludes robot r_0
robot r_0 excludes robot r_1
....

Conflict optimizer

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

We add weights
(each robot has a weight)

Loops

robot r_0 excludes robot r_1
robot r_1 excludes robot r_0
robot r_0 excludes robot r_1
….

Conflict optimizer

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
search for a path P arriving at time t=60
with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

We add weights
(each robot has a weight)

Loops
robot r_0 excludes robot r_1
robot r_1 excludes robot r_0
robot r_0 excludes robot r_1
….

Conflict optimizer

- Put the removed robots in a queue Q

Minimize the total weight of the robots in conflict…

- Pick the first robot r_0 of the queue
search for a path P arriving at t… t=60
with a minimum number of conflicts with other robots

Add the path of the first …ot r_0 in the current …ion…
…emove the robots in conflict with this path…

Conflict

We add weights
(each robot has a weight)

Loops
robot r_0 excludes robot r_1
robot r_1 excludes robot r_0
robot r_0 excludes robot r_1
....

Each time that a robot is
removed from the solution:
increase its weight…

- Put the removed robots in a queue Q

Minimize the total weight of
the robots in conflict…

- Pick the first robot r_0 of the queue
search for a path P arriving at t…e t=60
with a minimum number of conflicts with other robots

Add the path of the first
…ot r_0 in the current
…ion…
…emove the robots in
conflict with this path…

Conflict

Each time that a robot is removed from the solution: increase its weight…

- Put the removed robots in a queue Q

Minimize the total weight of the robots in conflict…

- Pick the first robot r_0 of the queue
search for a path P arriving at t… t=60
with a minimum number of conflicts with other robots

Add the path of the first …ot r_0 in the current …ion…
…emove the robots in conflict with this path…

Conflict

If there is a loop,
it increases the weights
and the algorithm goes out
of the loop…

Each time that a robot is
removed from the solution:
increase its weight…

- Put the removed robots in a queue Q

Minimize the total weight of
the robots in conflict…

- Pick the first robot r_0 of the queue
search for a path P arriving at t… t=60
with a minimum number of conflicts with other robots

Add the path of the first
…ot r_0 in the current
…ion…
…emove the robots in
conflict with this path…

Conflict

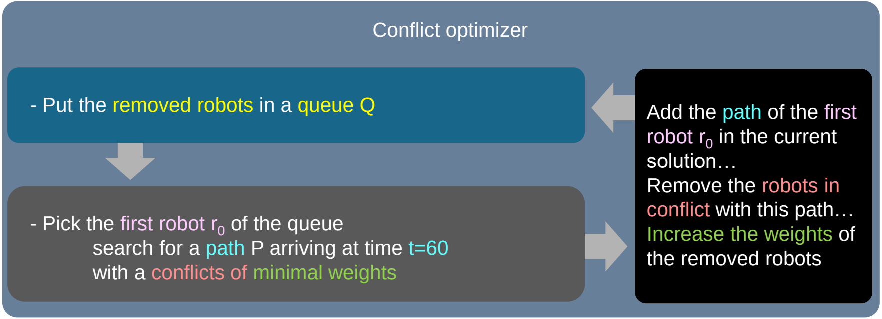
Conflict optimizer

- Put the removed robots in a queue Q

- Pick the first robot r_0 of the queue
  search for a path P arriving at time t=60
  with a minimum number of conflicts with other robots

Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

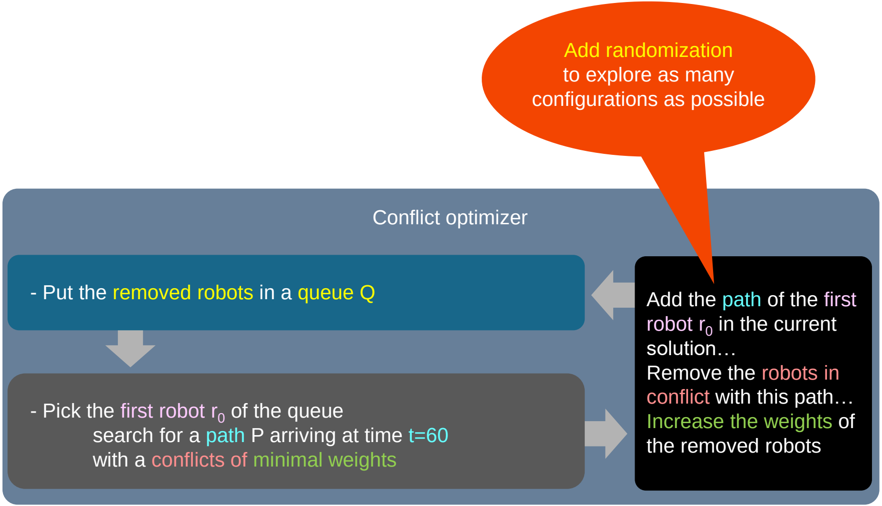
Conflict


Conflict optimizer
- Put the removed robots in a queue Q
- Pick the first robot r0 of the queue
search for a path P arriving at time t=60
with a conflicts of minimal weights
Add the path of the first robot r0 in the current solution…
Remove the robots in conflict with this path…
Increase the weights of the removed robots

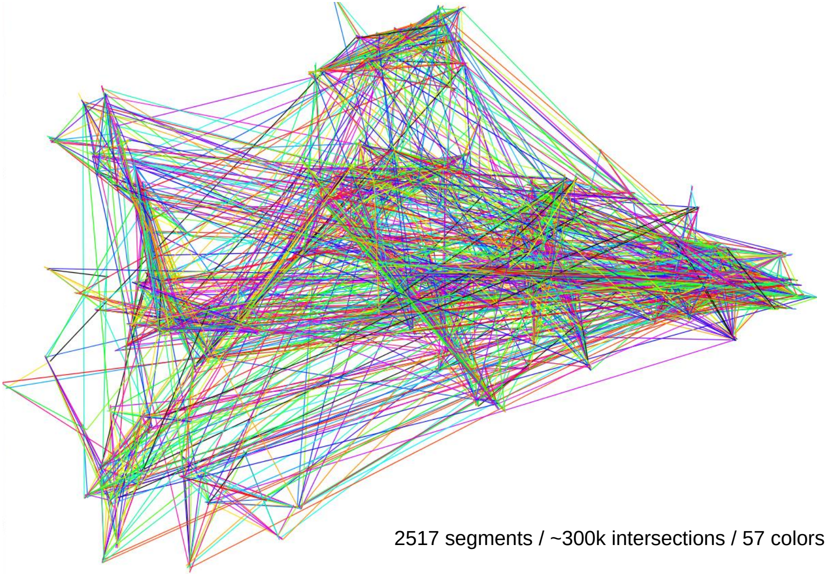
Conflict

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

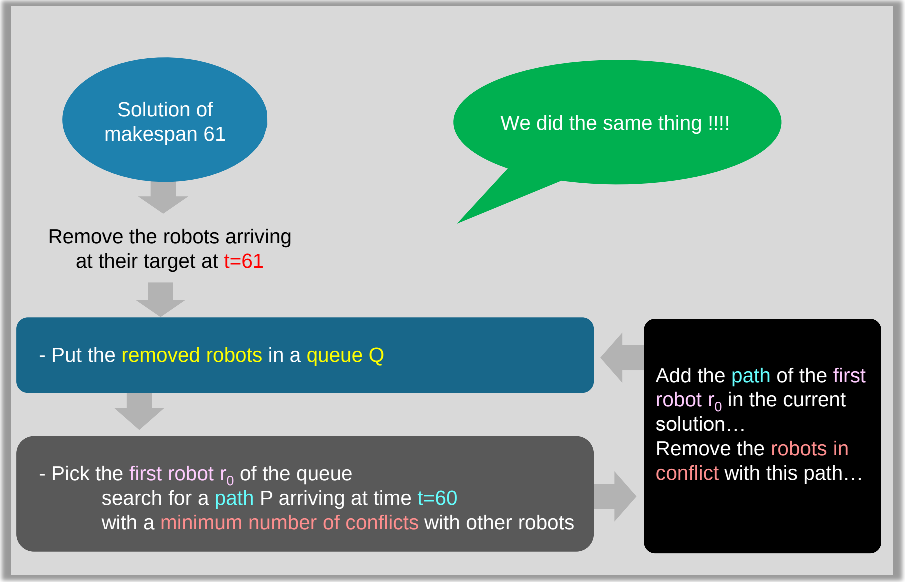


Conflict



2517 segments / ~300k intersections / 57 colors

Conflict


Solution of makespan 61
We did the same thing !!!!
Remove the robots arriving at their target at t=61
- Put the removed robots in a queue Q
- Pick the first robot r_0 of the queue
search for a path P arriving at time t=60
with a minimum number of conflicts with other robots
Add the path of the first robot r_0 in the current solution…
Remove the robots in conflict with this path…

Conflict

Shadoks

Introduction
- Two Phases

1-Greedy
- MaxPolygon
- Packing
- Coloring

1-IP
- Packing
- ConvexCover

2-IP
- ConvexCover
- ConvexPartition

2-Local
- MaxPolygon
- Packing

2-Conflict
- Robots
- Coloring

Some results

Solution with 61 colors

Remove one color

- Put the segments without color in a queue Q

- Pick the first segment s_0 of the queue
  choose its new color
  with a minimum conflict weight with other segments

Color s_0 with its new color…
Uncolor the segments in conflict with s_0 …
And increase their weights

Conflict

Solution with 61 colors

Remove one color

- Put the segments without color in a queue Q

- Pick the first segment s_0 of the queue
  choose its new color
  with a minimum conflict weight with other segments

Color s_0 with its new color…
Uncolor the segments in conflict with s_0 …
And increase their weights

Conflict

Solution with 61 colors

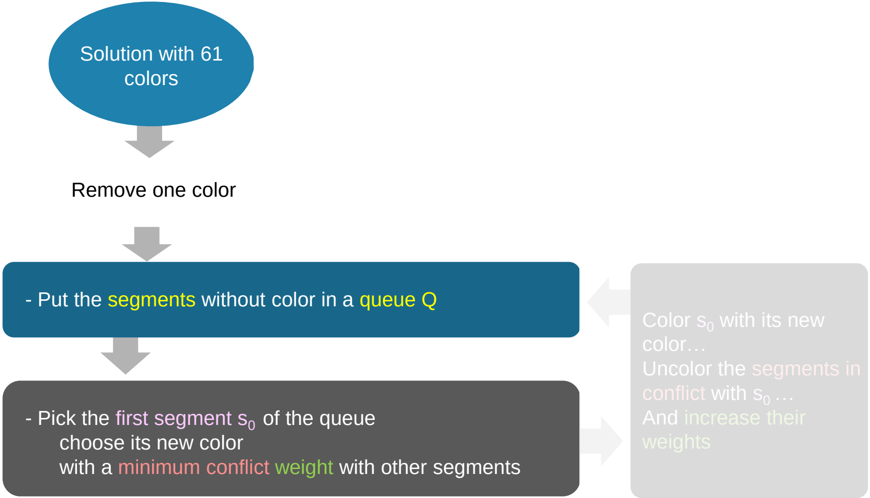
Remove one color

- Put the segments without color in a queue Q

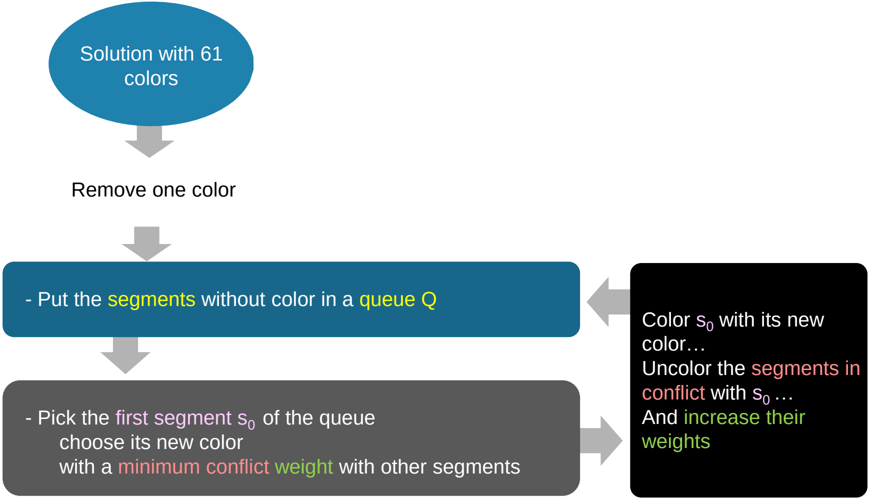
- Pick the first segment s_0 of the queue
  choose its new color
  with a minimum conflict weight with other segments

Color s_0 with its new color…
Uncolor the segments in conflict with s_0 …
And increase their weights

Conflict


Solution with 61 colors
Remove one color
- Put the segments without color in a queue Q
- Pick the first segment s0 of the queue
choose its new color
with a minimum conflict weight with other segments
Color s0 with its new color…
Uncolor the segments in conflict with s0 …
And increase their weights

Conflict


Solution with 61 colors
Remove one color
- Put the segments without color in a queue Q
- Pick the first segment s0 of the queue
choose its new color
with a minimum conflict weight with other segments
Color s0 with its new color…
Uncolor the segments in conflict with s0 …
And increase their weights

Conflict

Solution with 61 colors

Conflict optimizer

Remove one color

- Put the segments without color in a queue Q

- Pick the first segment s_0 of the queue
  choose its new color
  with a minimum conflict weight with other segments

Color s_0 with its new color…
Uncolor the segments in conflict with s_0 …
And increase their weights

Conflict

Solution with 61 colors

Remove one color

Conflict optimizer

Improvements (DFS…)

- Put the segments without color in a queue Q

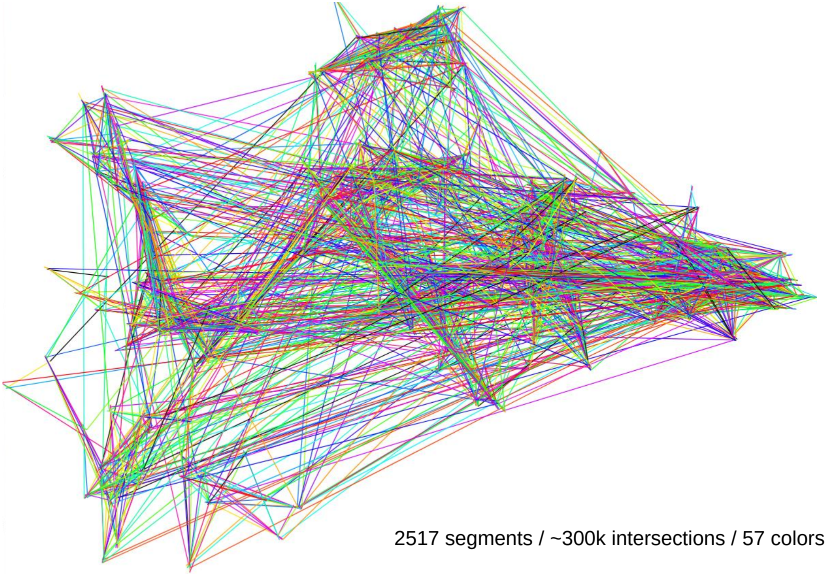
- Pick the first segment s_0 of the queue
  choose its new color
  with a minimum conflict weight with other segments

Color s_0 with its new color…
Uncolor the segments in conflict with s_0 …
And increase their weights

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Conflict



2517 segments / ~300k intersections / 57 colors

What did we get with these algorithms?

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

1-IP
Packing
ConvexCover

2-IP
ConvexCover
ConvexPartition

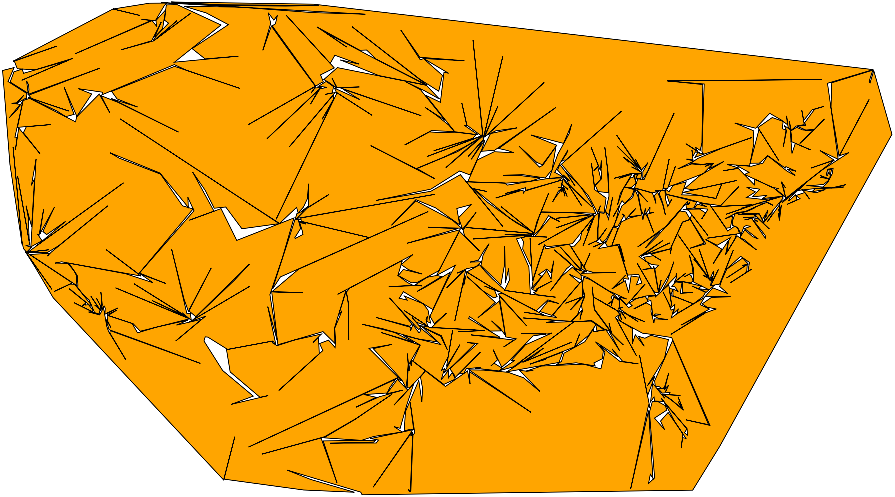
2-Local
MaxPolygon
Packing

2-Conflict
Robots
Coloring

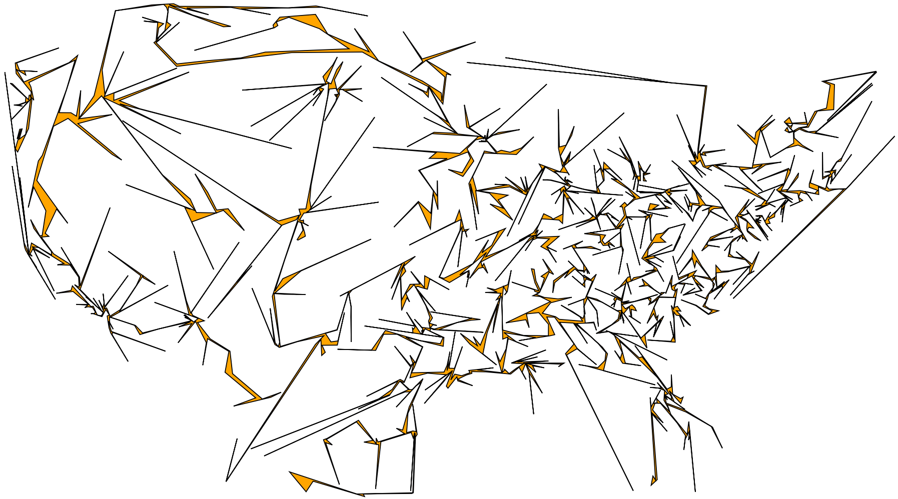
Some results

The (very hard) problem
of going
on vacation

Maximum Area Polygon

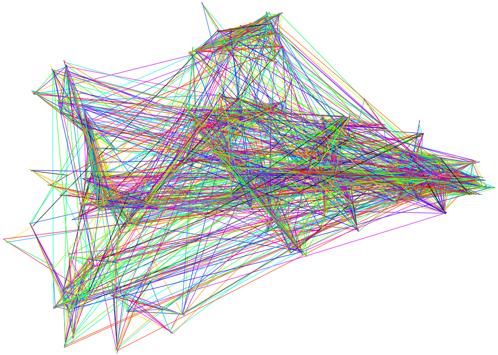


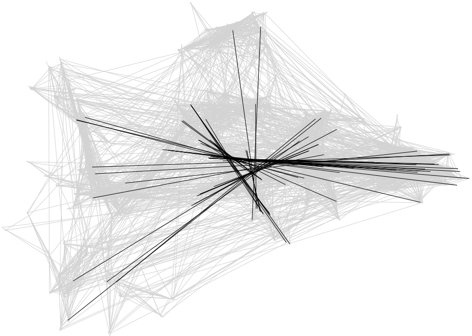
Minimum Area Polygon



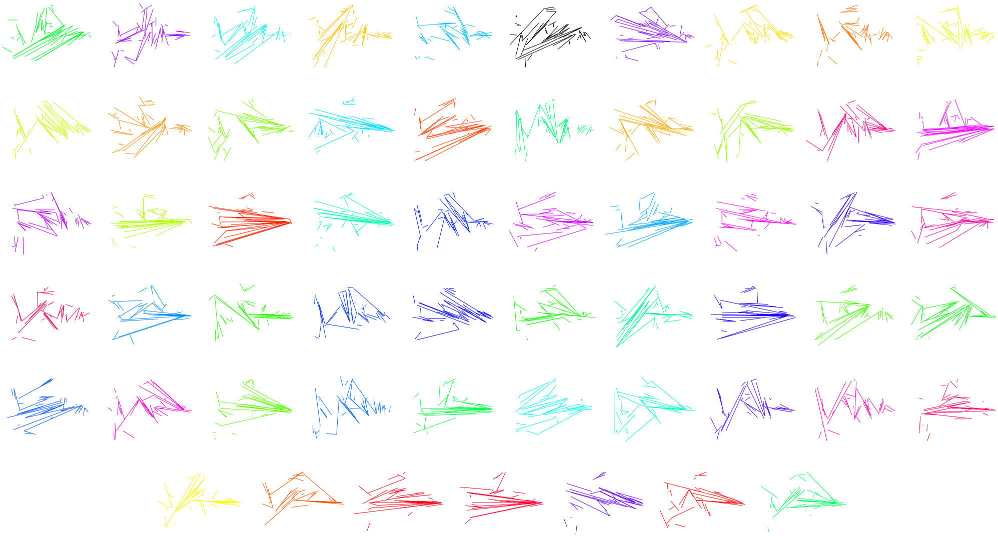
Segment Intersection Graph Coloring and Clique

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

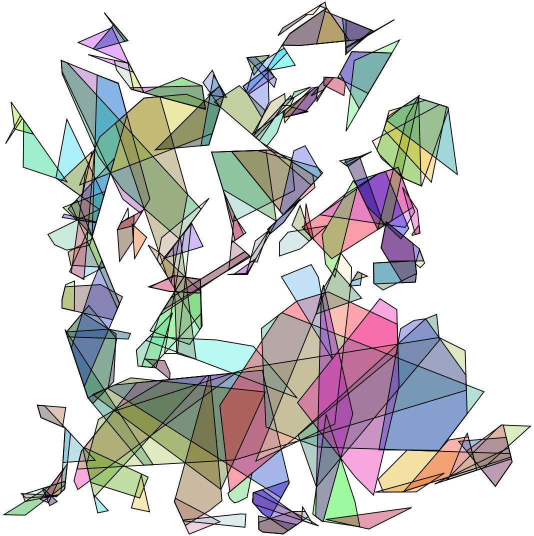




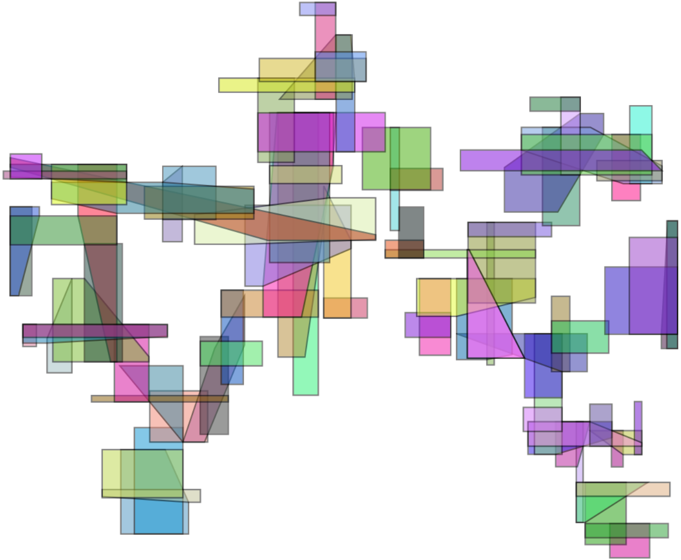
Each Color Class (Interior-Disjoint Segments)



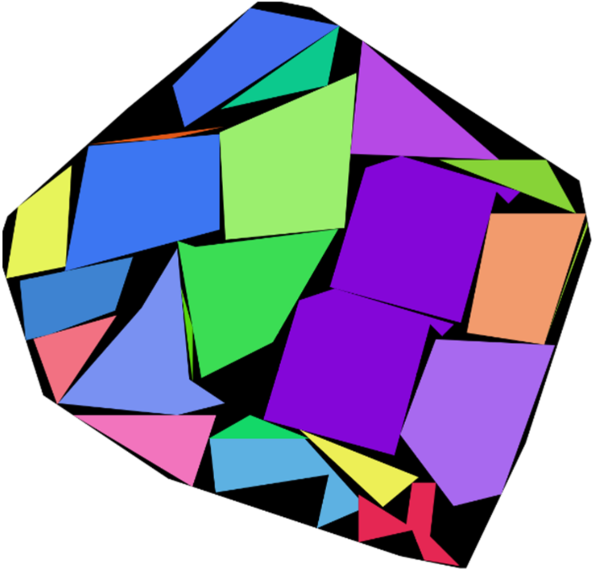
Convex Cover



Convex Cover



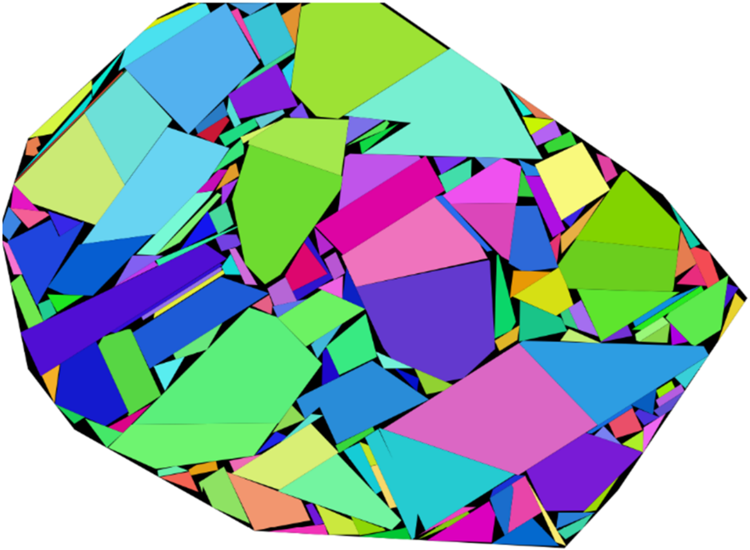
Packing (50 items)



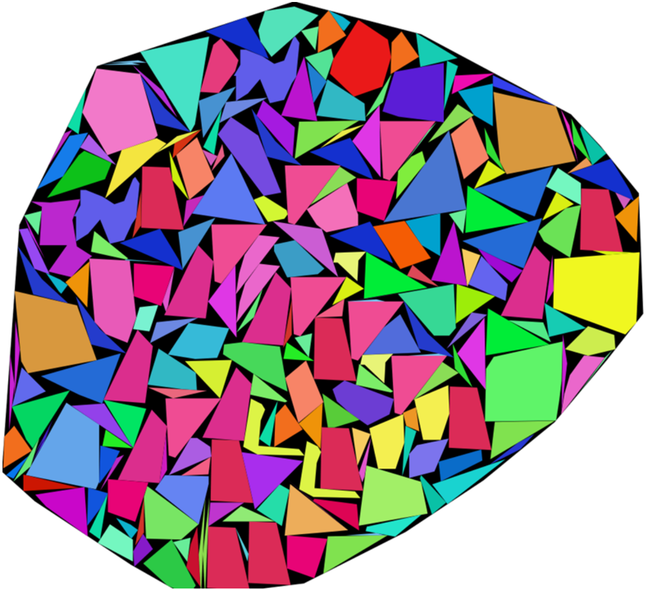
Packing (335 items)

Shadoks

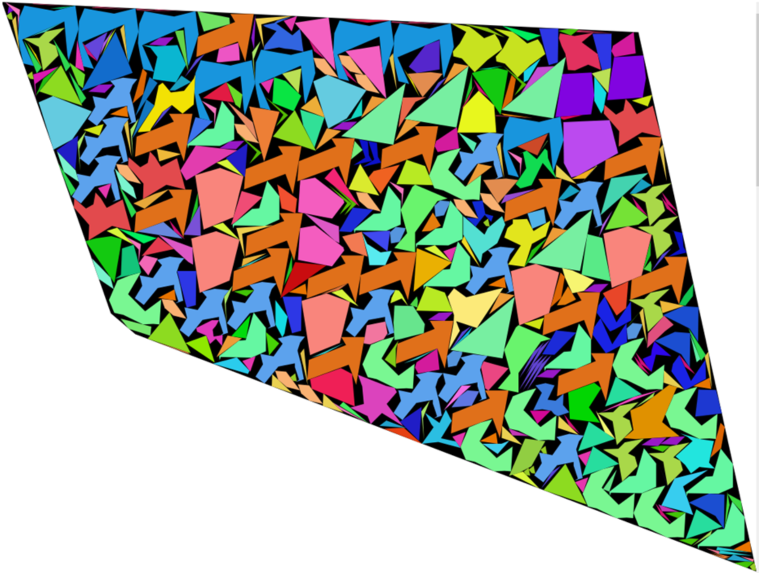
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



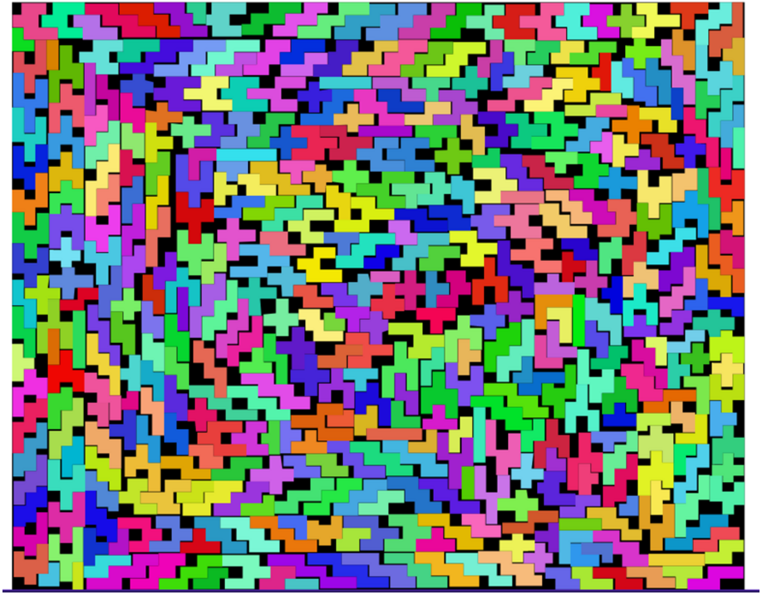
Packing (500 items)



Packing 1000 items)



Packing (1240 items)



Packing (1672 items)

Shadoks

Introduction
Two Phases

1-Greedy
MaxPolygon
Packing
Coloring

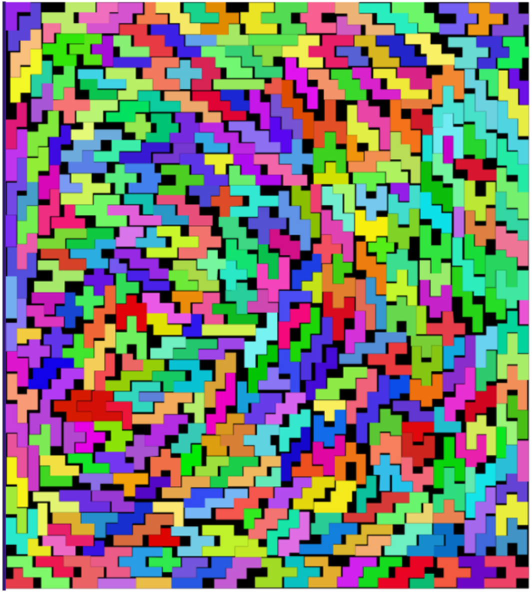
1-IP
Packing
ConvexCover

2-IP
ConvexCover
ConvexPartition

2-Local
MaxPolygon
Packing

2-Conflict
Robots
Coloring

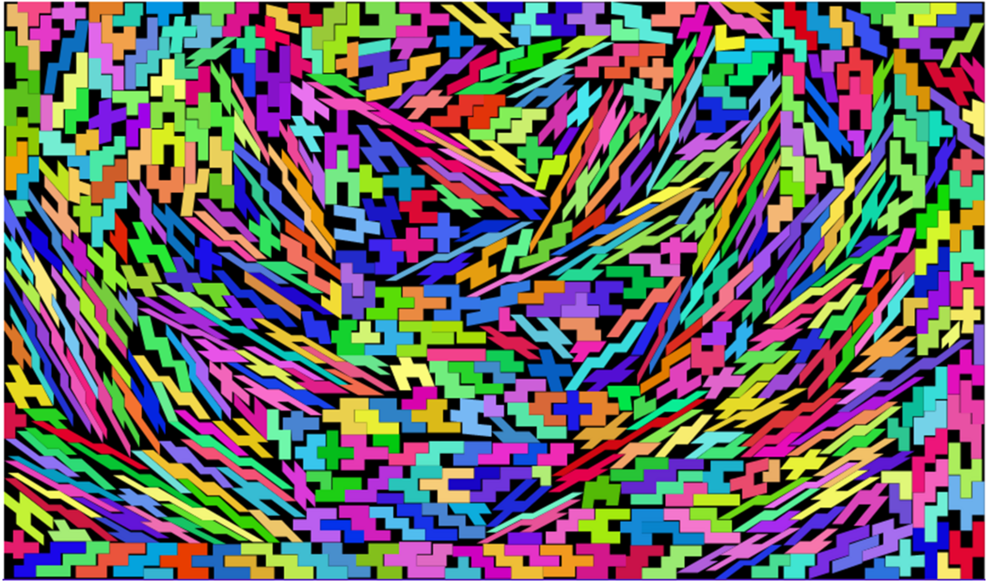
Some results



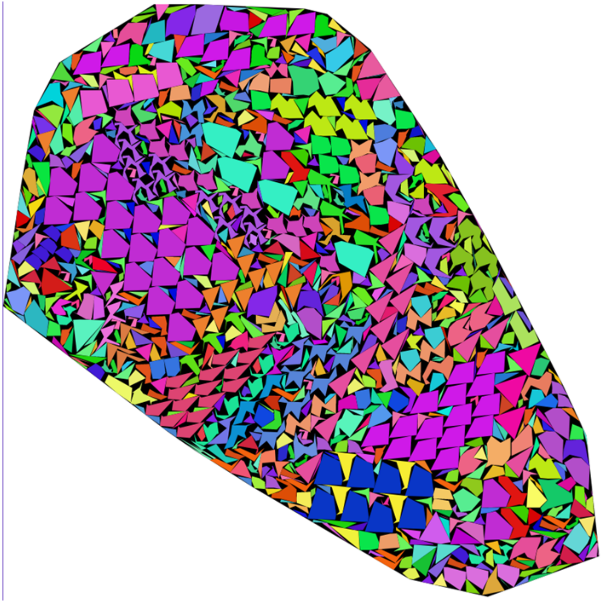
Packing (1685 items)

Shadoks

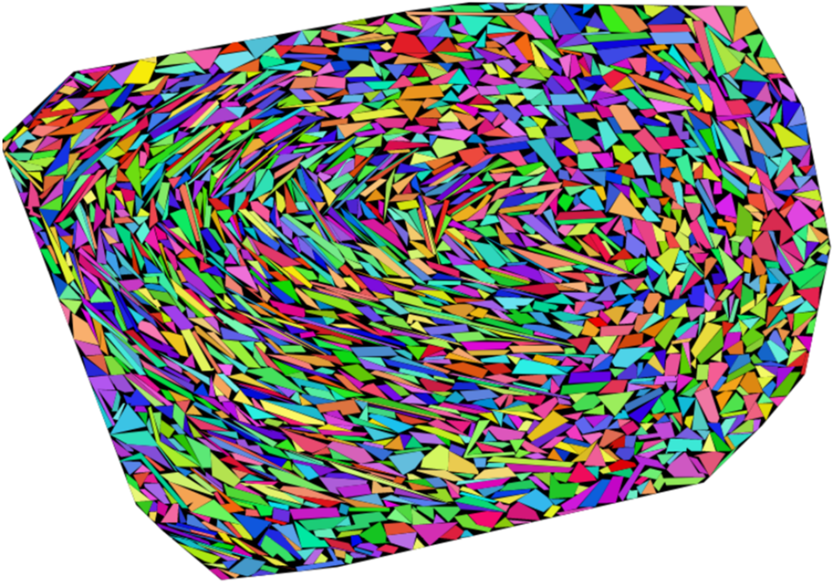
Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



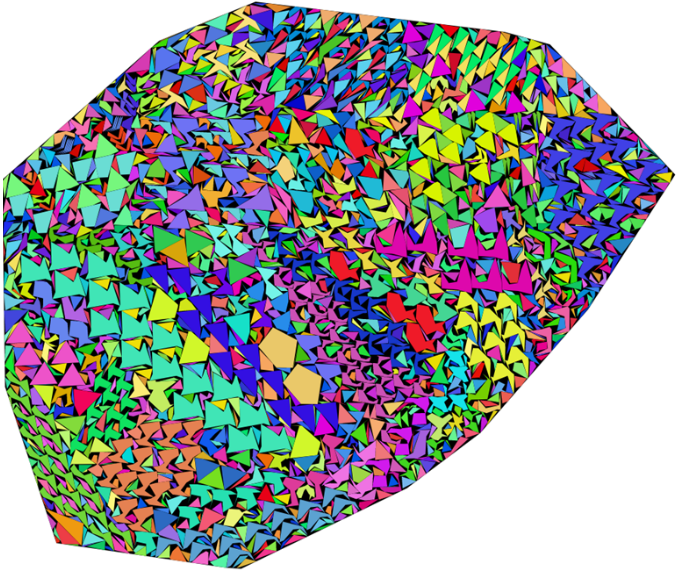
Packing (2000 items)



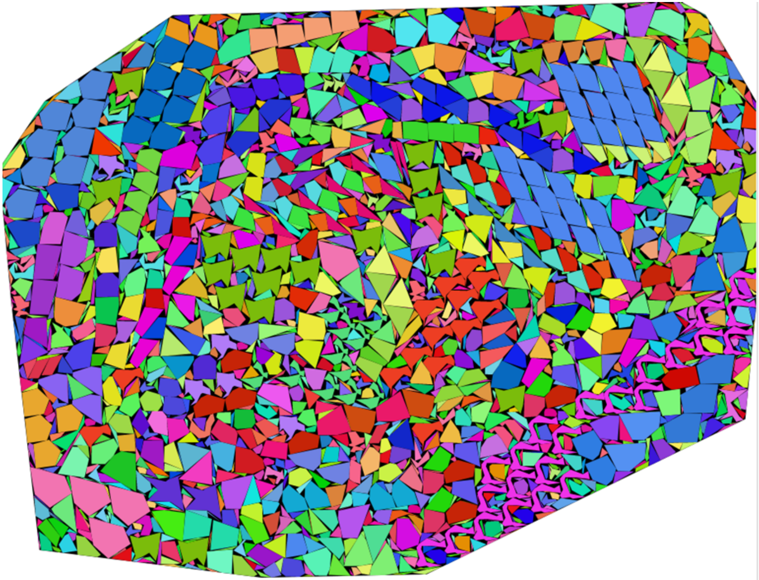
Packing (3000 items)



Packing (5000 items)



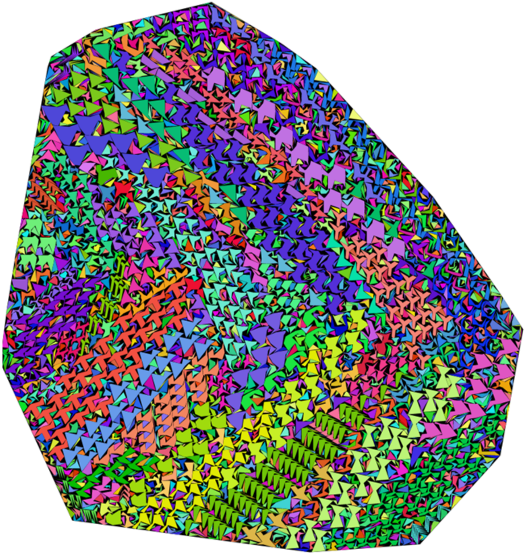
Packing (10000 items)



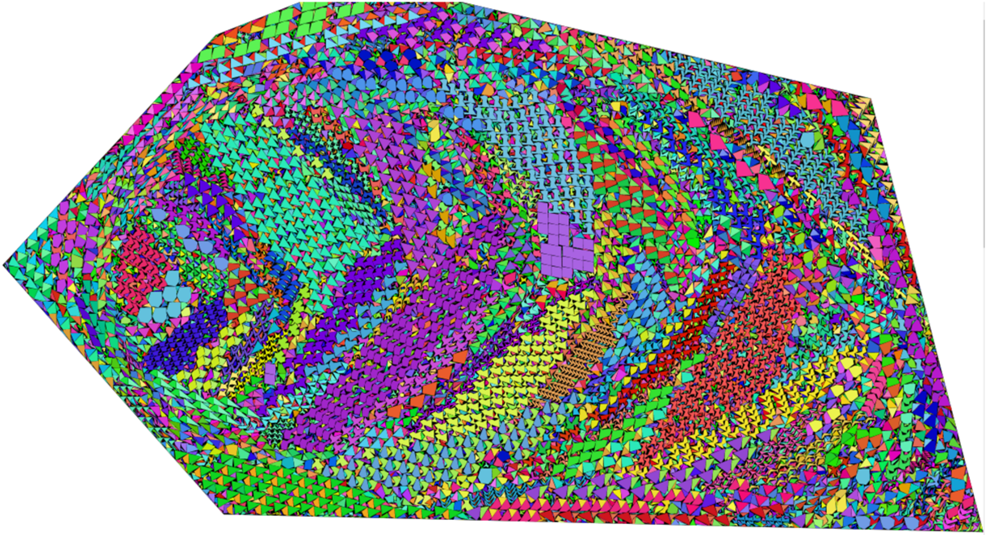
Packing (10000 items)

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results



Packing (50000 items)



Packing (50000 items)

Shadoks

Introduction
- Two Phases

1-Greedy
- MaxPolygon
- Packing
- Coloring

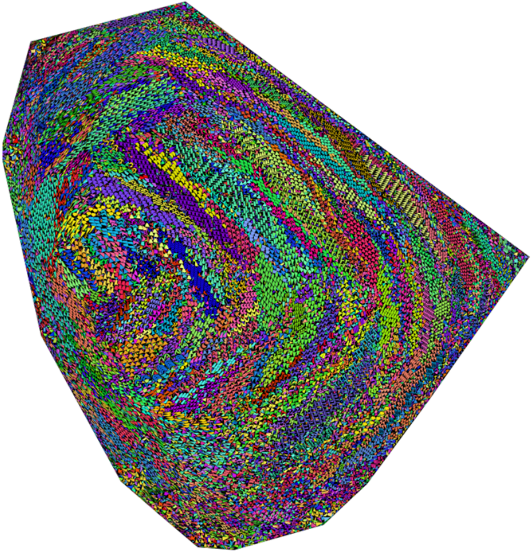
1-IP
- Packing
- ConvexCover

2-IP
- ConvexCover
- ConvexPartition

2-Local
- MaxPolygon
- Packing

2-Conflict
- Robots
- Coloring

Some results



Moving Robots Video

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Thank You!

Shadoks

Introduction
Two Phases
1-Greedy
MaxPolygon
Packing
Coloring
1-IP
Packing
ConvexCover
2-IP
ConvexCover
ConvexPartition
2-Local
MaxPolygon
Packing
2-Conflict
Robots
Coloring
Some results

Art by @maryanneshakyhand