Computing the Flip Distance of Triangulations

Bachelor-Kolloquium

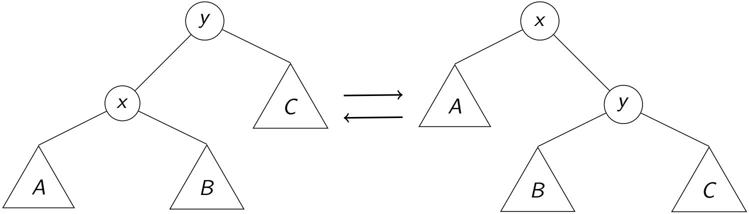
Fabian Lipp

July 5, 2012

Contents

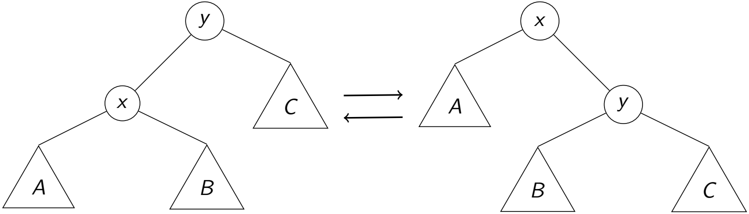
Motivation of the Problem

- Let T_1, T_2 be binary search trees
- Operations: Left rotation, right rotation



Motivation of the Problem

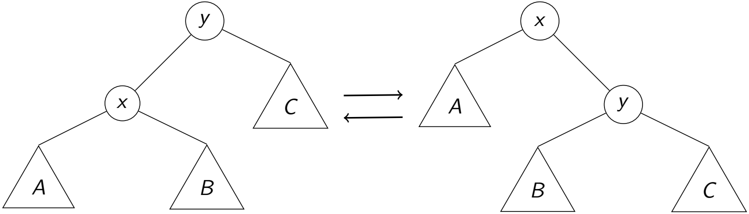
- Let T_1, T_2 be binary search trees
- Operations: Left rotation, right rotation



- Task: Find a shortest sequence of rotations that lead from T_1 to T_2

Motivation of the Problem

- Let T_1, T_2 be binary search trees
- Operations: Left rotation, right rotation



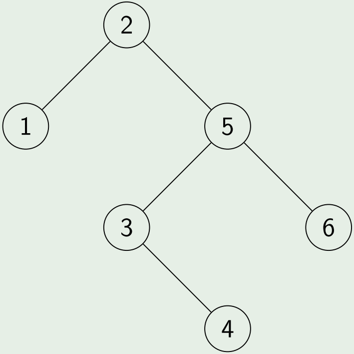
- Task: Find a shortest sequence of rotations that lead from T_1 to T_2
- There is no polynomial-time algorithm known for this problem, but NP-hardness is not proven

Transformation of the Problem

Represent a binary tree with $n-2$ nodes as a triangulation of a polygon with n corners

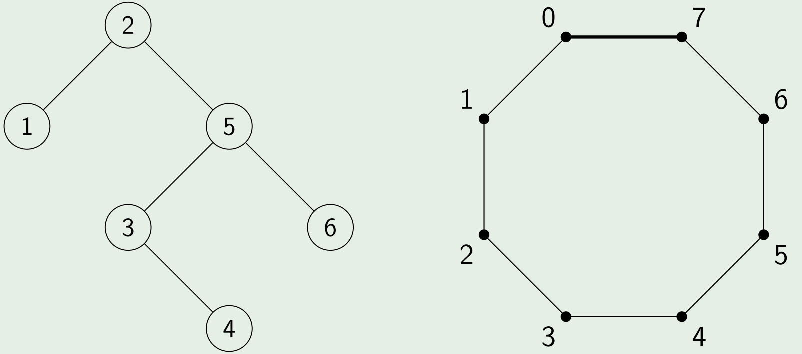
Transformation of the Problem

Represent a binary tree with $n - 2$ nodes as a triangulation of a polygon with n corners



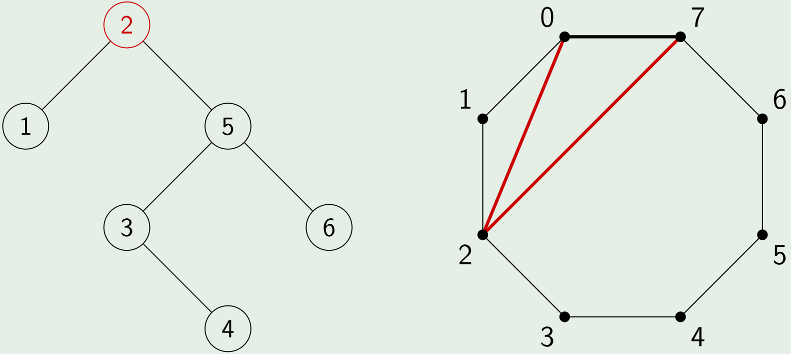
Transformation of the Problem

Represent a binary tree with $n-2$ nodes as a triangulation of a polygon with n corners



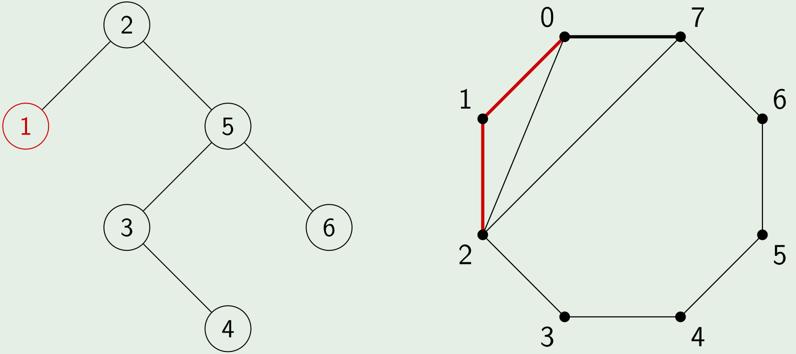
Transformation of the Problem

Represent a binary tree with $n-2$ nodes as a triangulation of a polygon with n corners



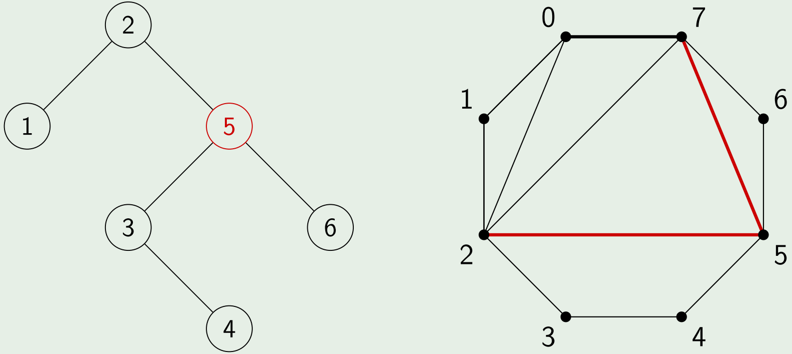
Transformation of the Problem

Represent a binary tree with $n - 2$ nodes as a triangulation of a polygon with n corners



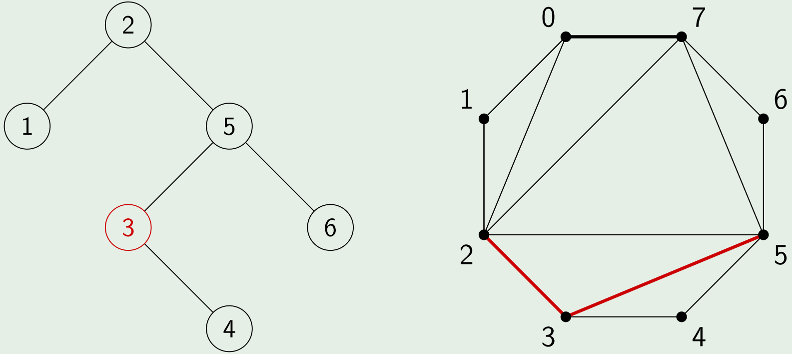
Transformation of the Problem

Represent a binary tree with $n-2$ nodes as a triangulation of a polygon with n corners



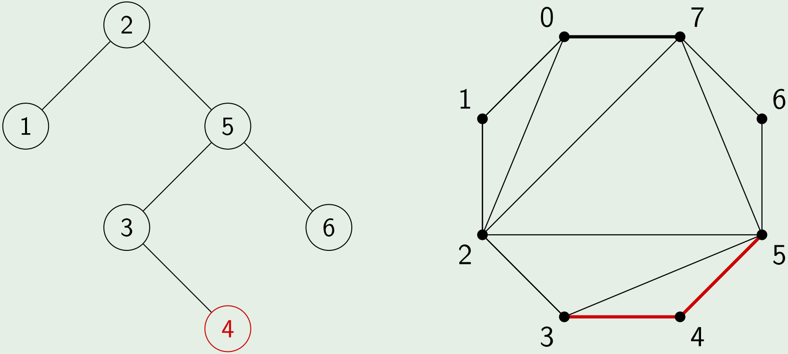
Transformation of the Problem

Represent a binary tree with $n-2$ nodes as a triangulation of a polygon with n corners



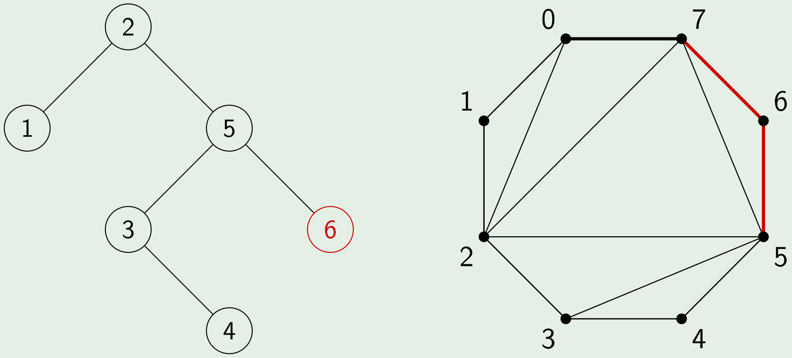
Transformation of the Problem

Represent a binary tree with $n - 2$ nodes as a triangulation of a polygon with n corners



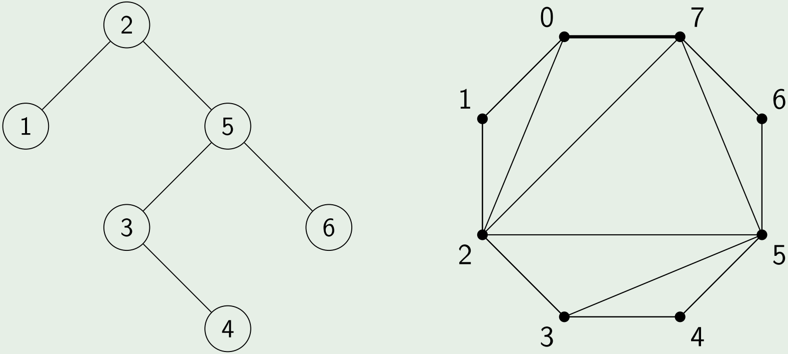
Transformation of the Problem

Represent a binary tree with $n - 2$ nodes as a triangulation of a polygon with n corners

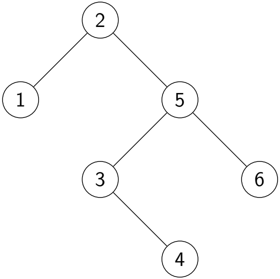


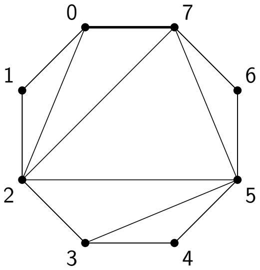
Transformation of the Problem

Represent a binary tree with $n - 2$ nodes as a triangulation of a polygon with n corners

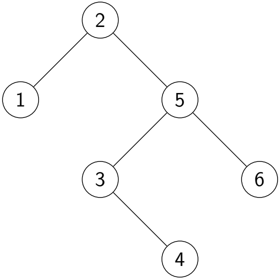


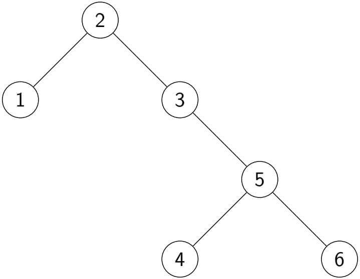
Equivalent of Rotations

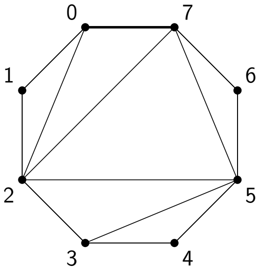




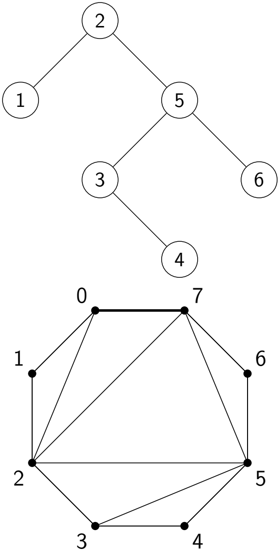
Equivalent of Rotations

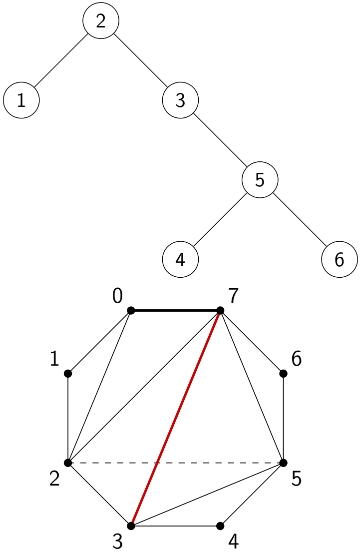






Equivalent of Rotations





Flip Distance

Rotation distance between binary trees is equivalent to flip distance between triangulations

Flip Distance

Rotation distance between binary trees is equivalent to flip distance between triangulations

Definition (Flip Distance)

The *flip distance* $\mathrm{fd}(\pi_1, \pi_2)$ is the minimum number of diagonal flips required to transform π_1 to π_2.

Flip Distance

Rotation distance between binary trees is equivalent to flip distance between triangulations

Definition (Flip Distance)

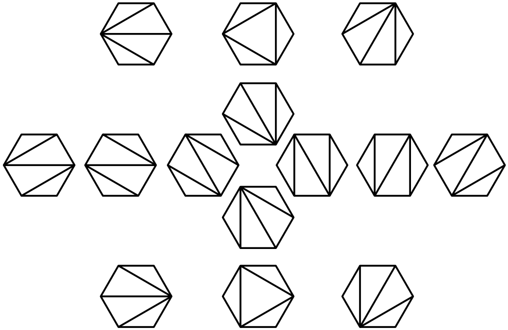
The *flip distance* $\mathrm{fd}(\pi_1, \pi_2)$ is the minimum number of diagonal flips required to transform π_1 to π_2.

Upper bound for two triangulations of the n-gon:

$$\mathrm{fd}(\pi_1, \pi_2) \leq 2n - 6$$

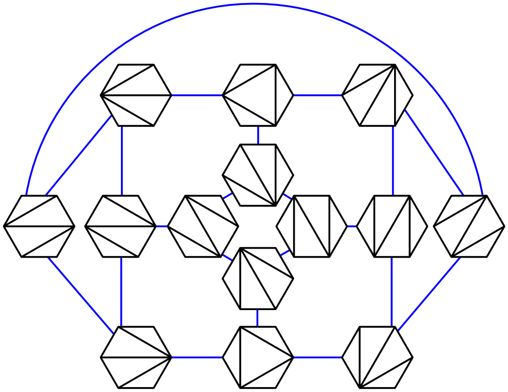
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



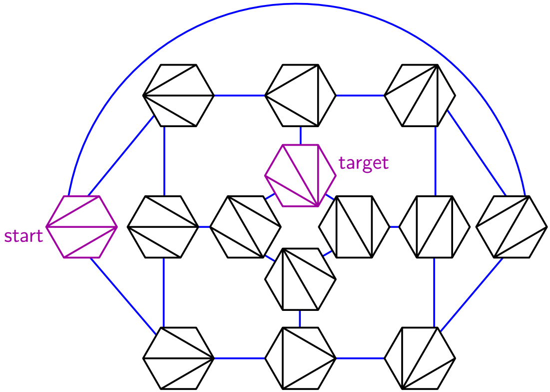
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



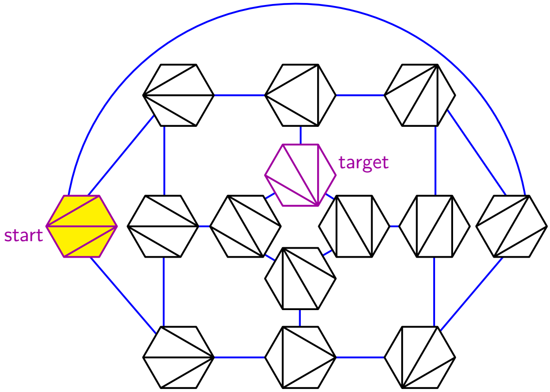
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



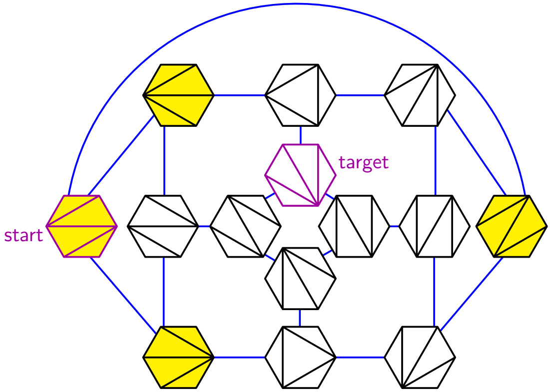
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



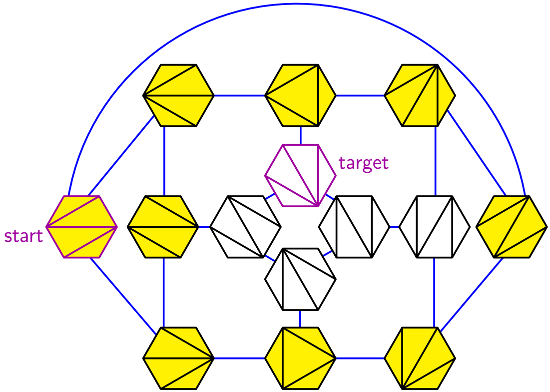
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



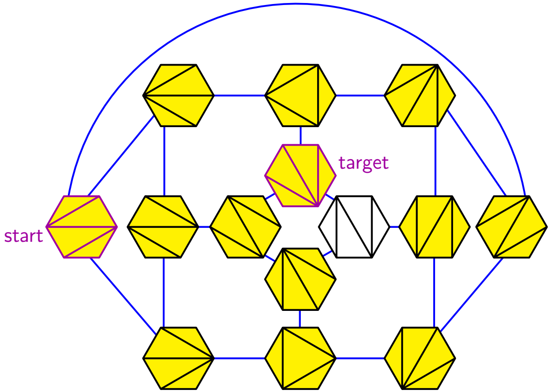
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



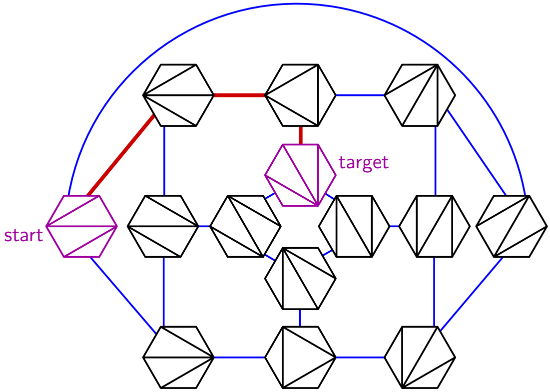
Breadth-First Search

Flip Graph $\mathcal{F}_6$:



Breadth-First Search

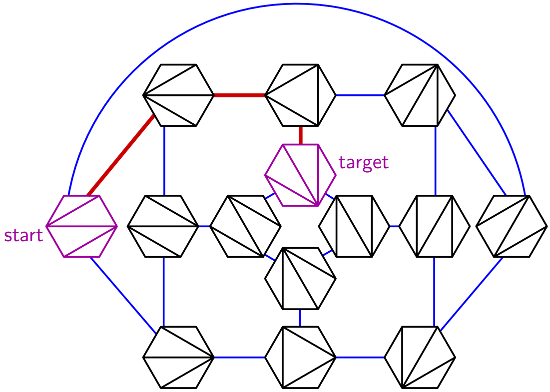
Flip Graph $\mathcal{F}_6$:



$\mathrm{fd}(\mathrm{start}, \mathrm{target}) = 3$

Breadth-First Search

Flip Graph $\mathcal{F}_6$:



$\mathrm{fd}(\mathrm{start}, \mathrm{target}) = 3$

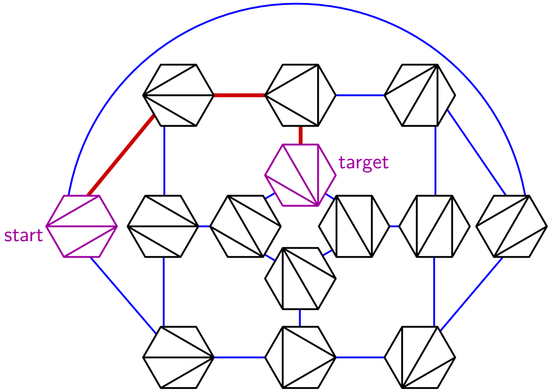
Runtime

Number of nodes in $\mathcal{F}_{n+2}$:

$$C_n = \frac{1}{n+2}\binom{2n}{n}$$

Breadth-First Search

Flip Graph $\mathcal{F}_6$:



$\text{fd}(\text{start}, \text{target}) = 3$

Runtime

Number of nodes in $\mathcal{F}_{n+2}$:

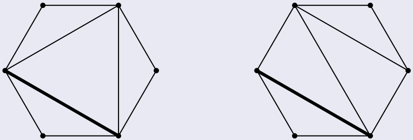
$$C_n = \frac{1}{n+2}\binom{2n}{n}$$

$$C_n \sim \frac{4^n}{\sqrt{\pi n^3}}$$

$\Rightarrow$ Exponential runtime

Simple Improvements

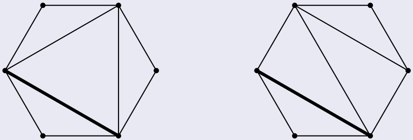
Common Diagonal



Do not flip this diagonal in BFS

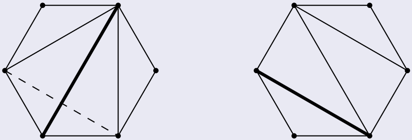
Simple Improvements

Common Diagonal



Do not flip this diagonal in BFS

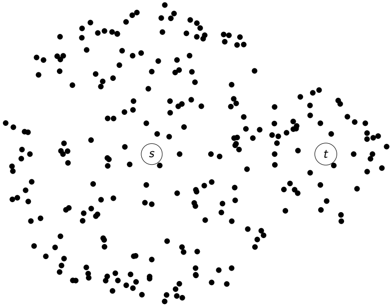
Flip-to-Match Diagonal



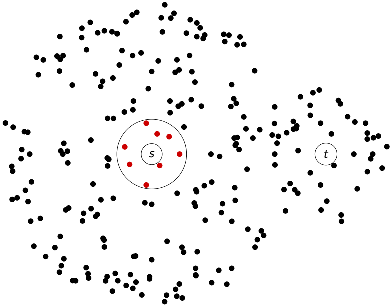
Flip only this diagonal in BFS

Two-Sided BFS

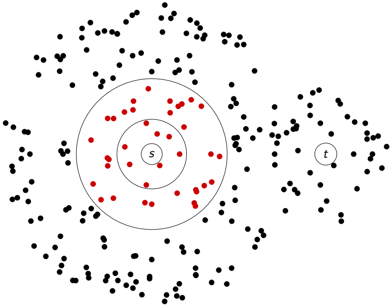
Two-Sided BFS



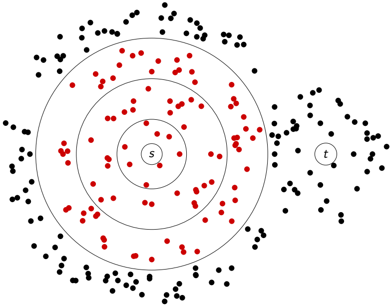
Two-Sided BFS



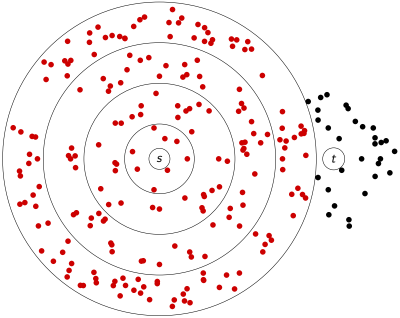
Two-Sided BFS



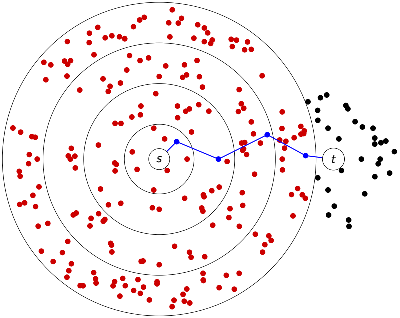
Two-Sided BFS



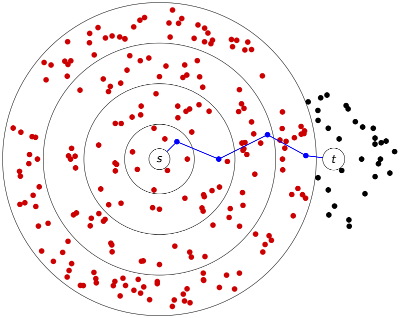
Two-Sided BFS



Two-Sided BFS

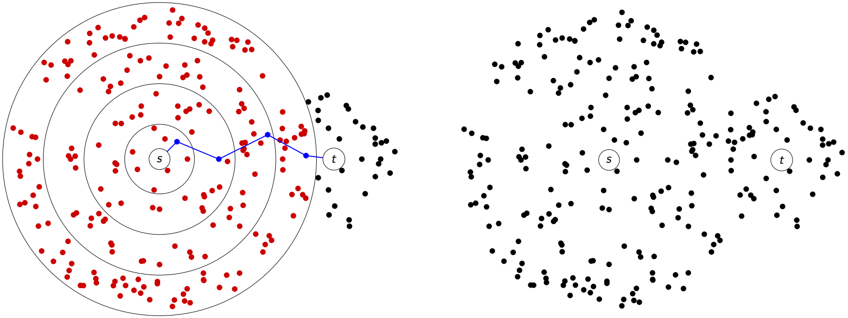


Two-Sided BFS



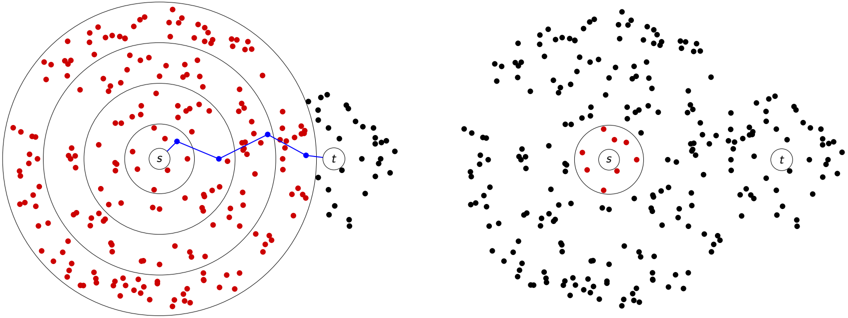
- Start BFS from start and target at the same time

Two-Sided BFS



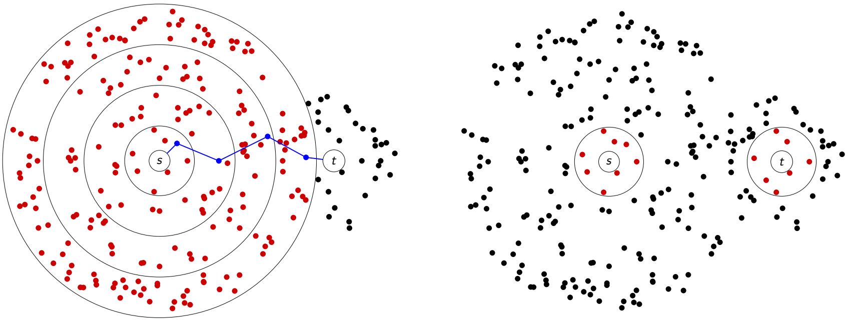
- Start BFS from start and target at the same time

Two-Sided BFS



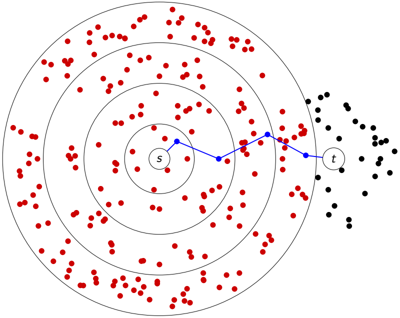
- Start BFS from start and target at the same time

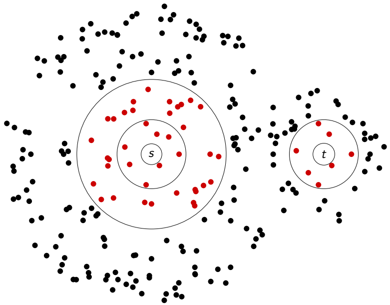
Two-Sided BFS



- Start BFS from start and target at the same time

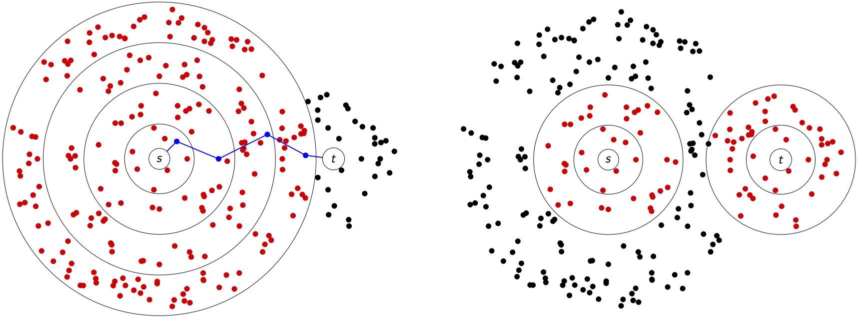
Two-Sided BFS





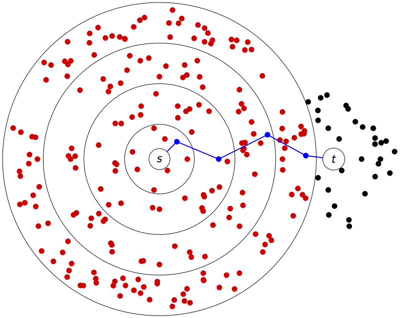
- Start BFS from start and target at the same time

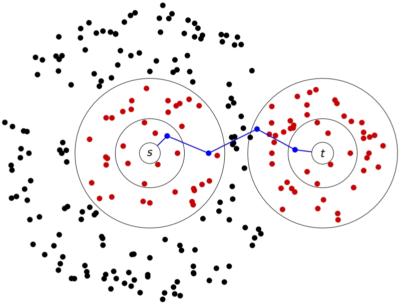
Two-Sided BFS



- Start BFS from start and target at the same time

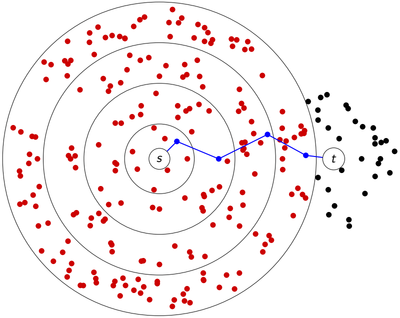
Two-Sided BFS





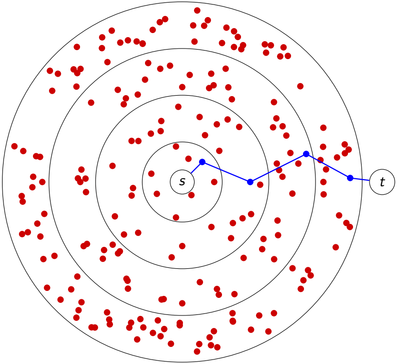
- Start BFS from start and target at the same time

Two-Sided BFS

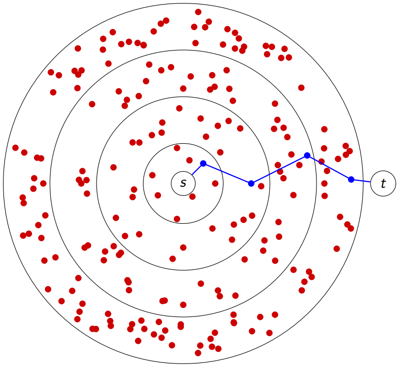


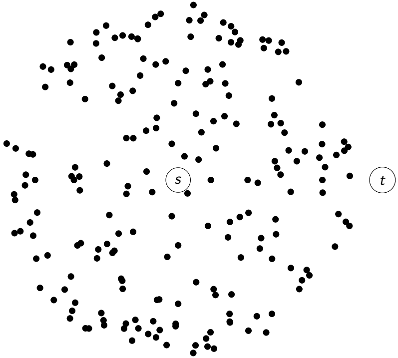
- Start BFS from start and target at the same time
- Reduces the number of visited nodes

A* search algorithm

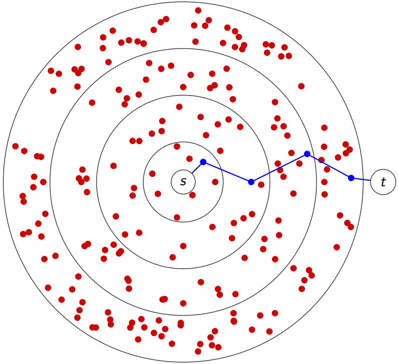


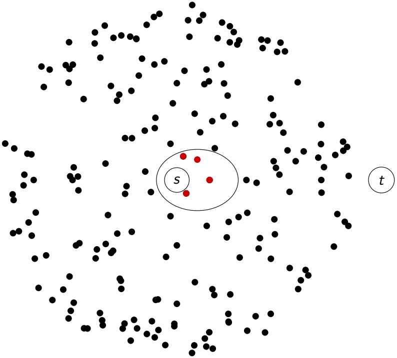
A* search algorithm



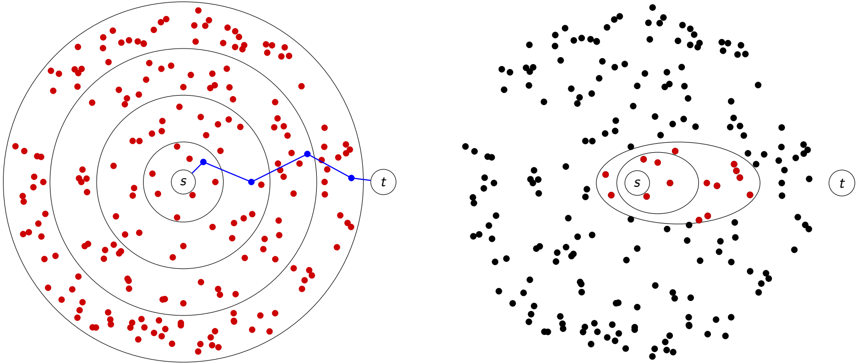


A* search algorithm

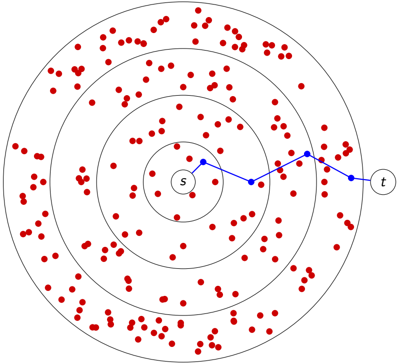


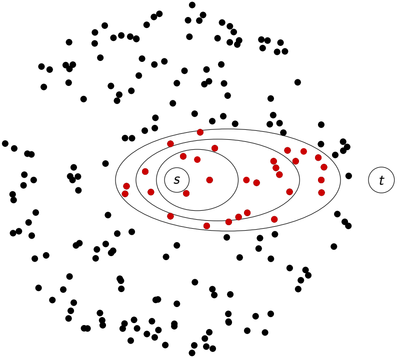


A* search algorithm

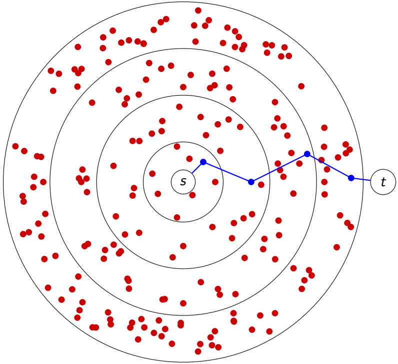


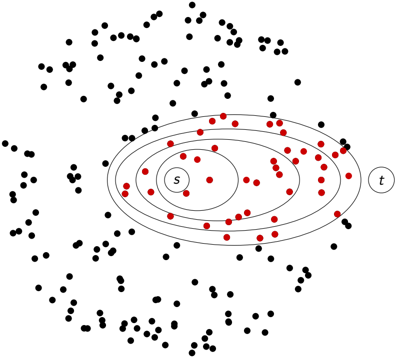
A* search algorithm



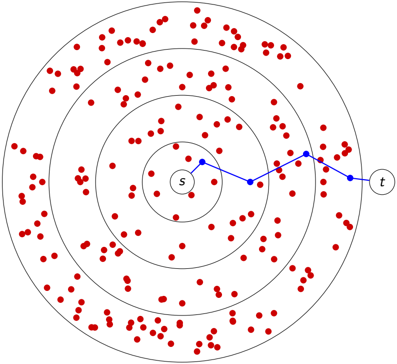


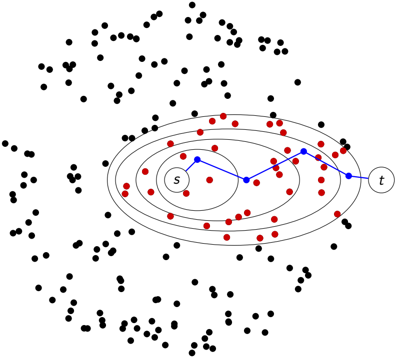
A* search algorithm



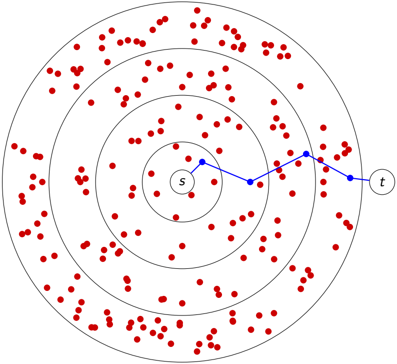


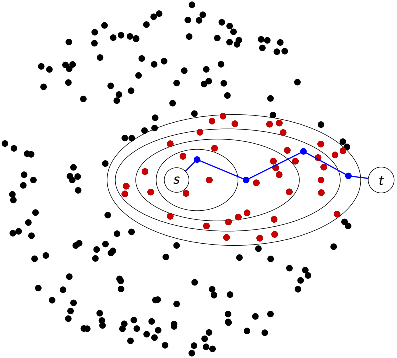
A* search algorithm





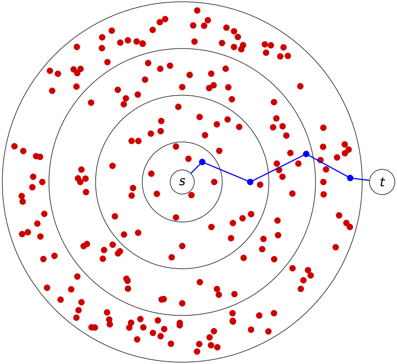
A* search algorithm

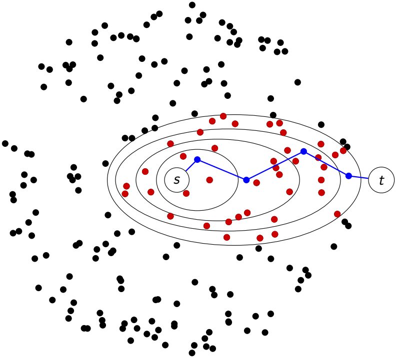




- Reduces the number of visited nodes

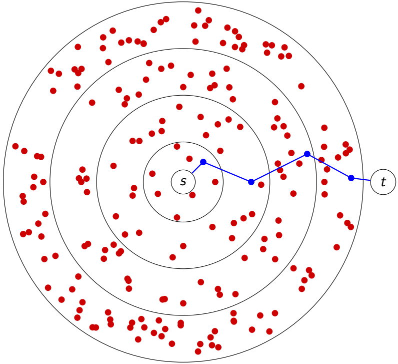
A* search algorithm

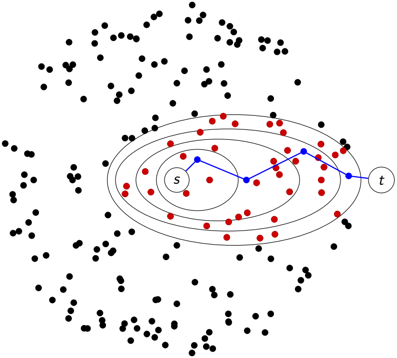




- Reduces the number of visited nodes
- Needs lower bound for distance from an arbitrary node to target

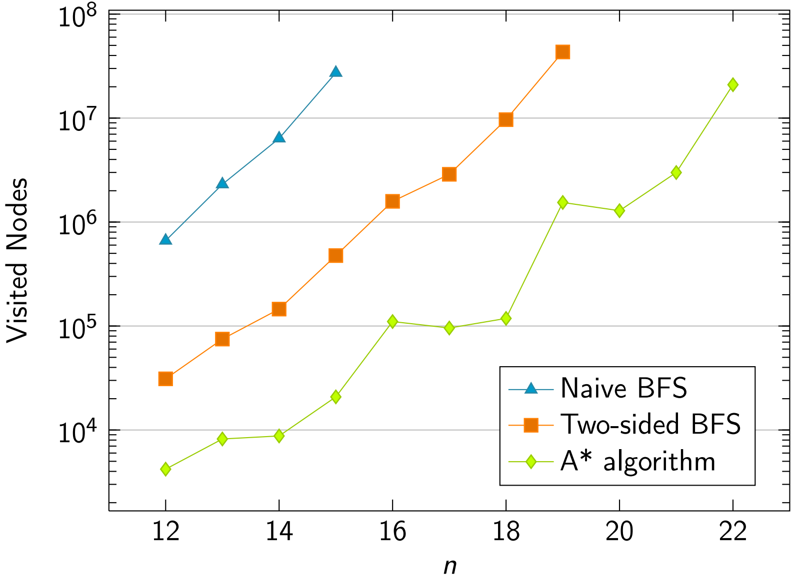
A* search algorithm



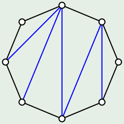


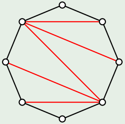
- Reduces the number of visited nodes
- Needs lower bound for distance from an arbitrary node to target
- Easy lower bound: number of diagonals that are not common

Effects of these improvements

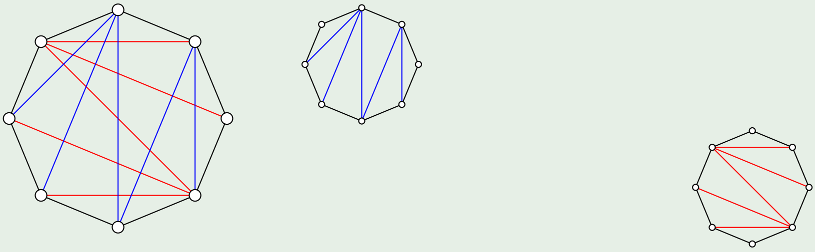


HighestDifference

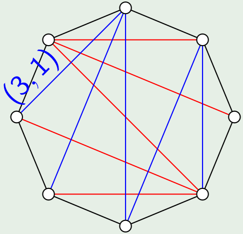


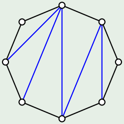


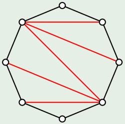
HighestDifference



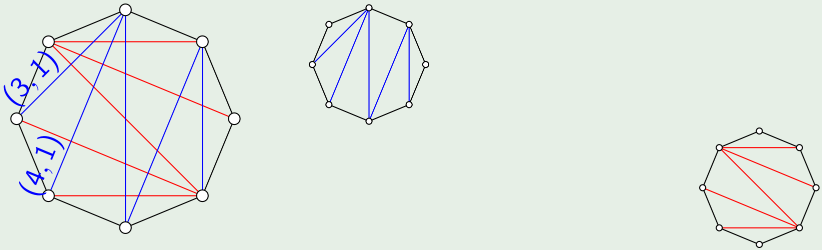
HighestDifference



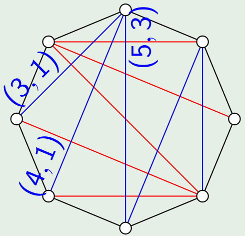


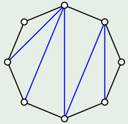


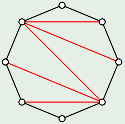
HighestDifference



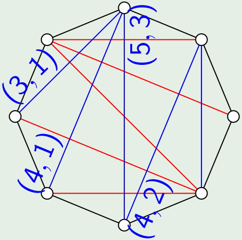
HighestDifference

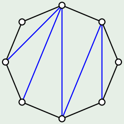


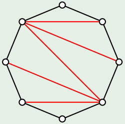




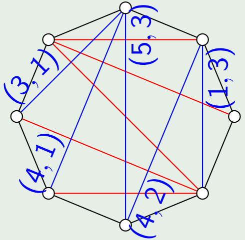
HighestDifference

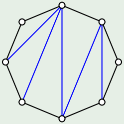


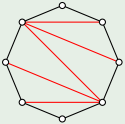




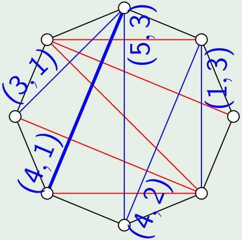
HighestDifference

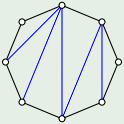


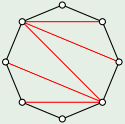




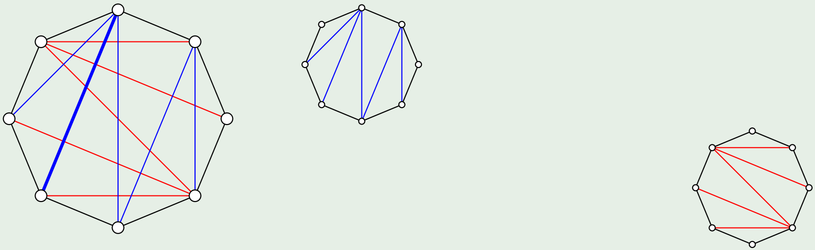
HighestDifference



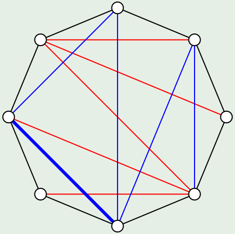


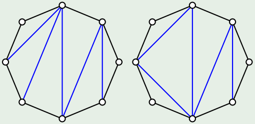


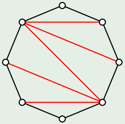
HighestDifference



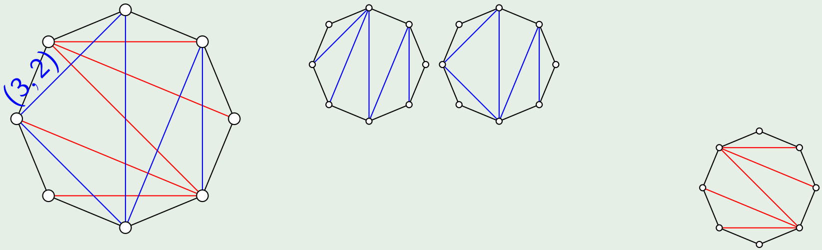
HighestDifference



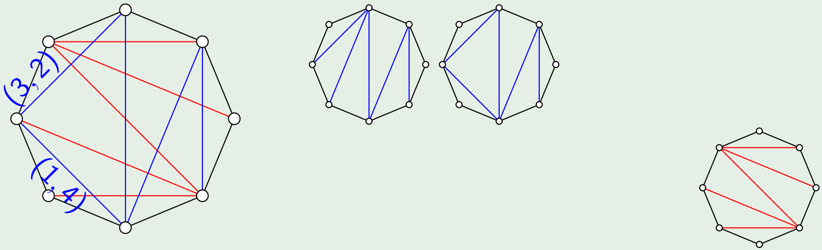




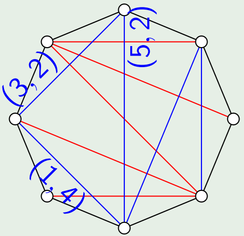
HighestDifference

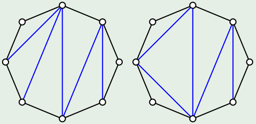


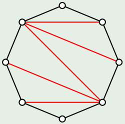
HighestDifference



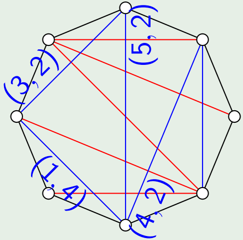
HighestDifference

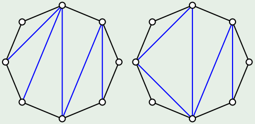


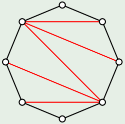




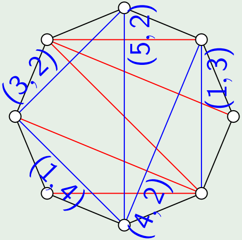
HighestDifference

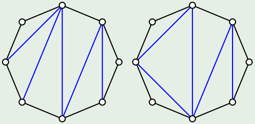


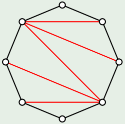




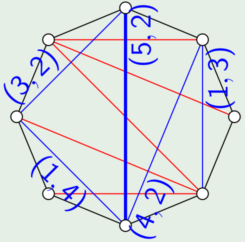
HighestDifference

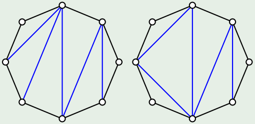


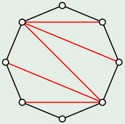




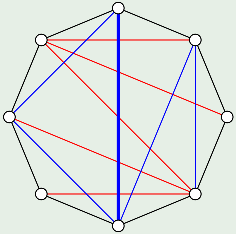
HighestDifference

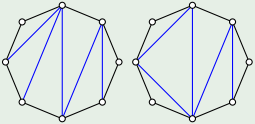


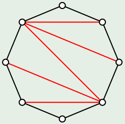




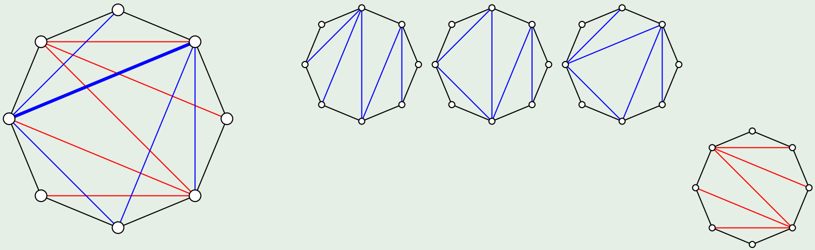
HighestDifference



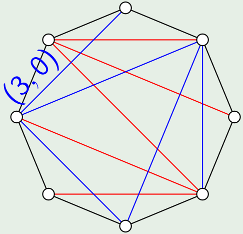


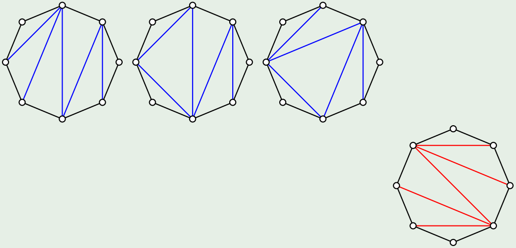


HighestDifference

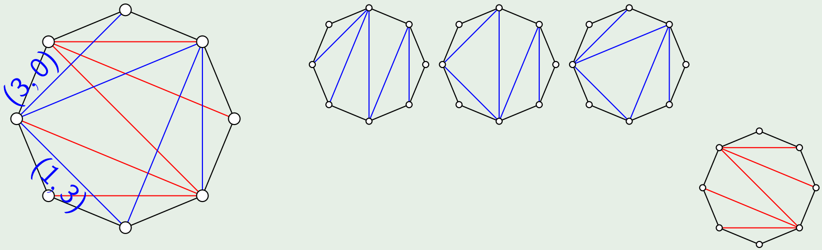


HighestDifference

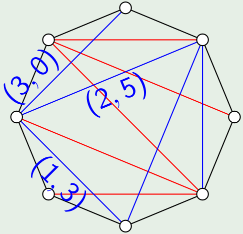


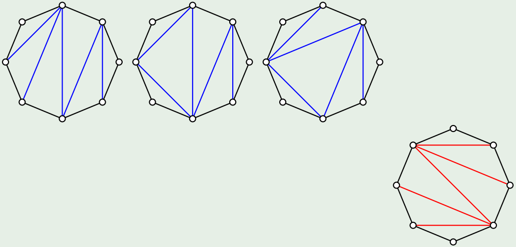


HighestDifference

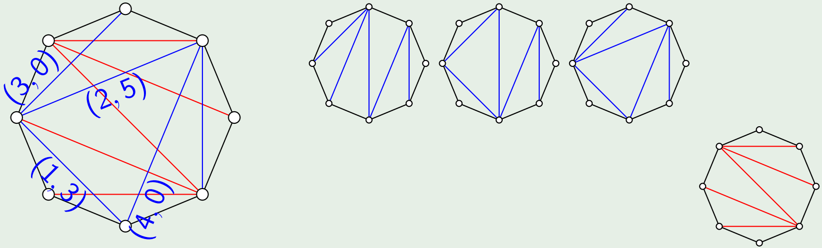


HighestDifference

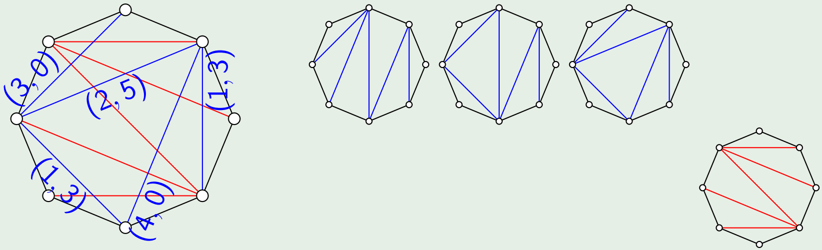




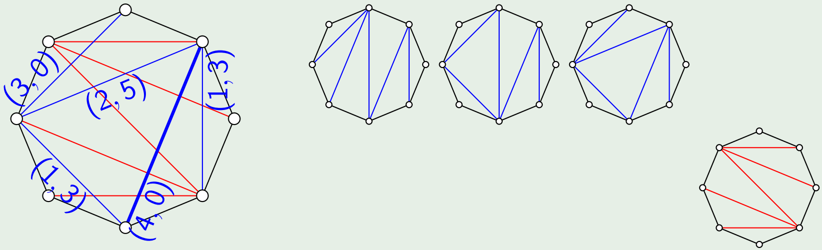
HighestDifference



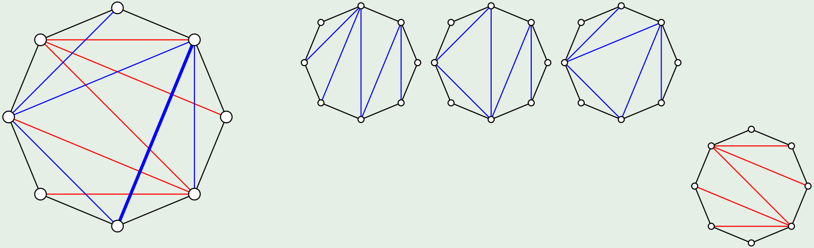
HighestDifference



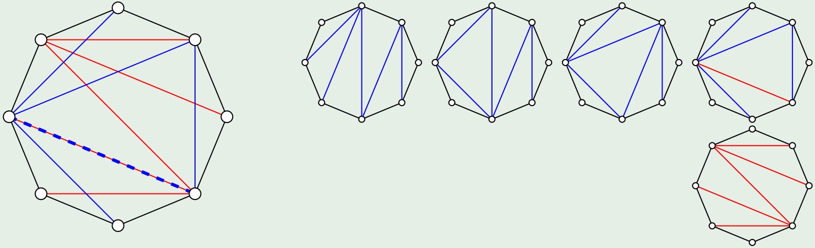
HighestDifference



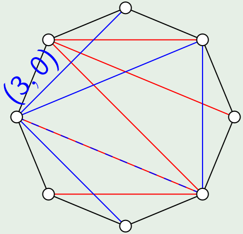
HighestDifference

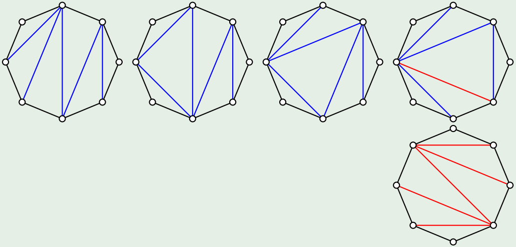


HighestDifference

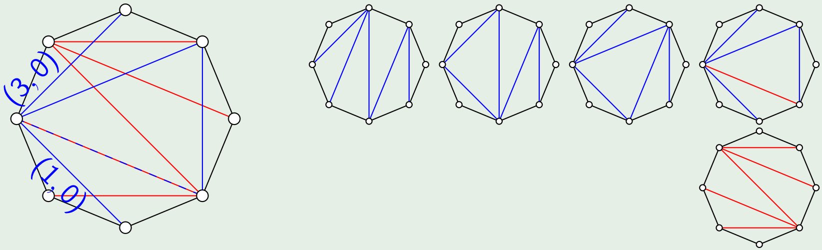


HighestDifference

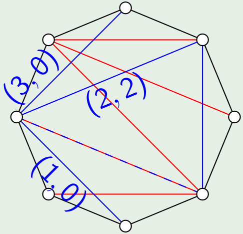


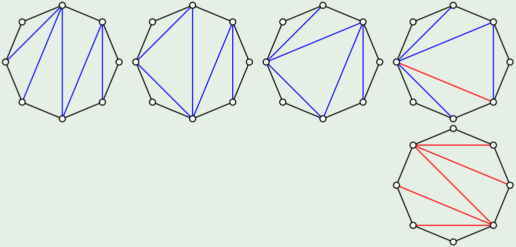


HighestDifference

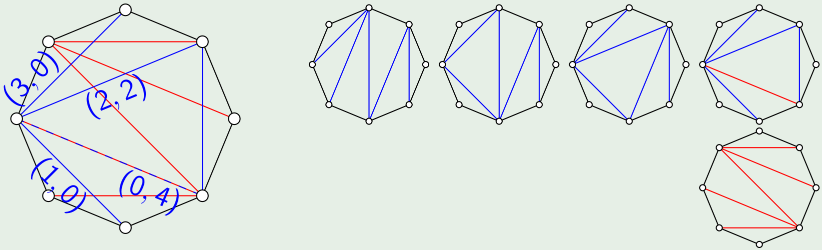


HighestDifference

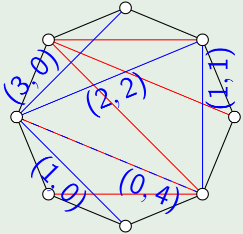


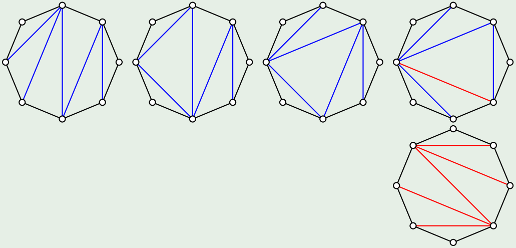


HighestDifference

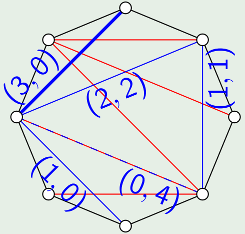


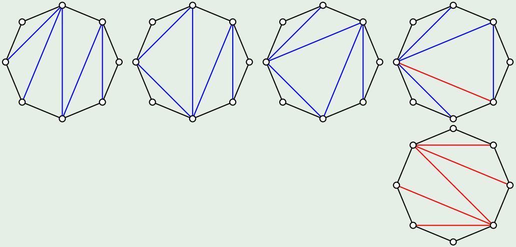
HighestDifference



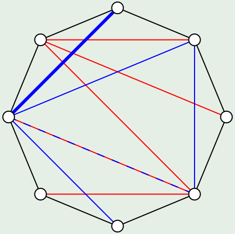


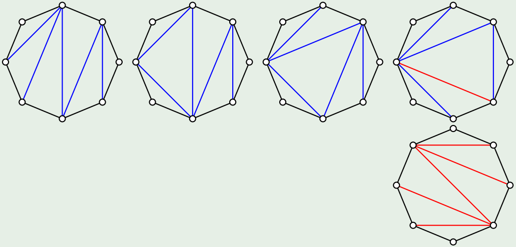
HighestDifference



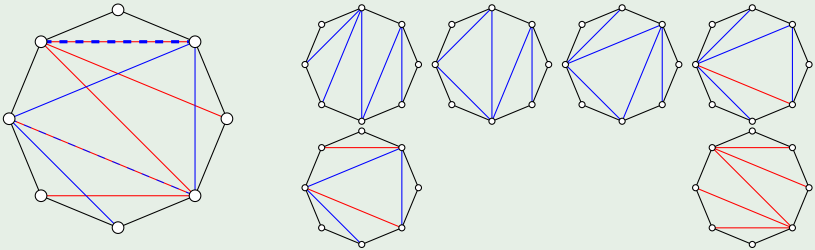


HighestDifference

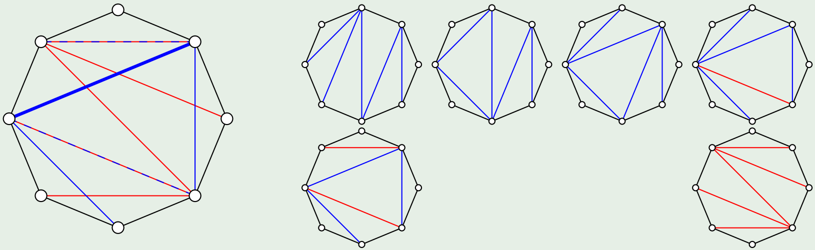




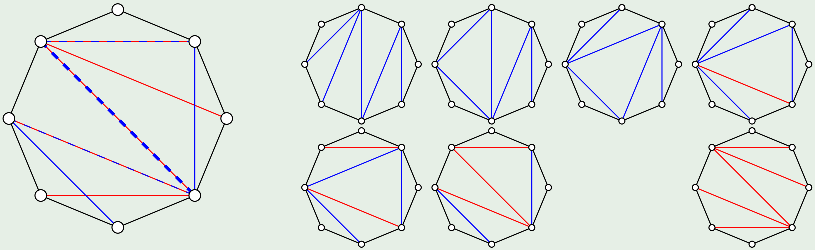
HighestDifference



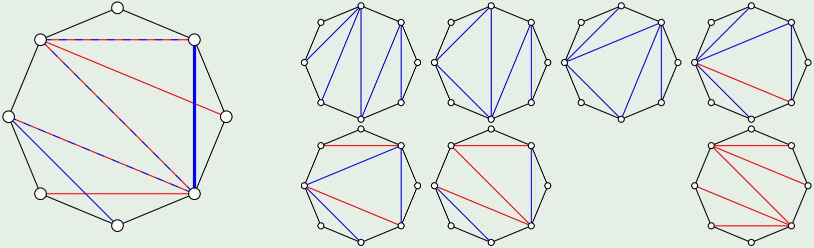
HighestDifference



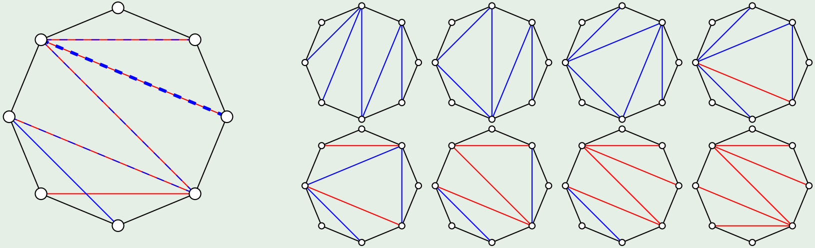
HighestDifference



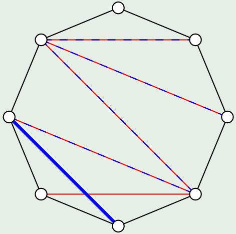
HighestDifference

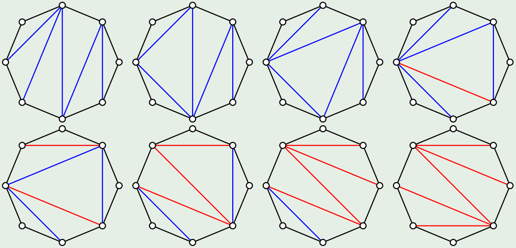


HighestDifference

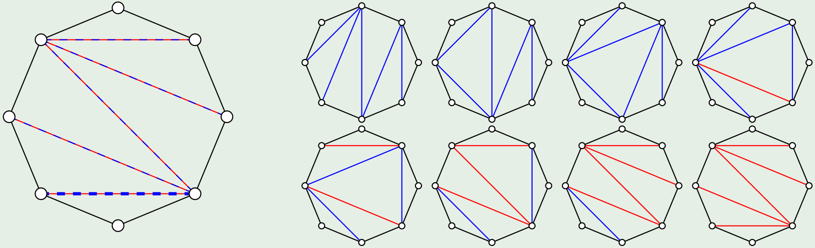


HighestDifference

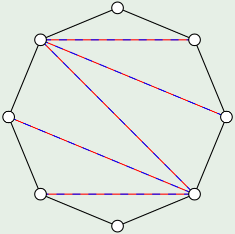


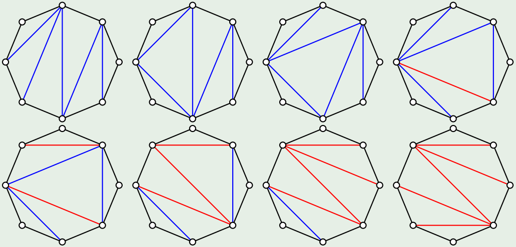


HighestDifference

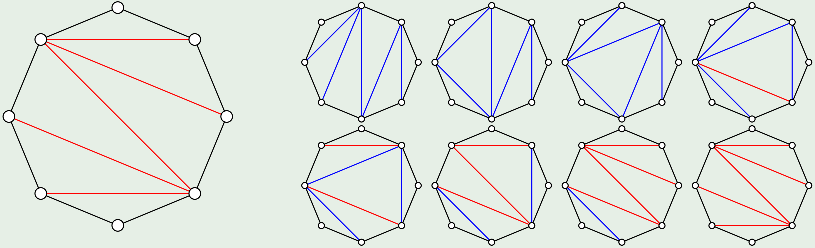


HighestDifference

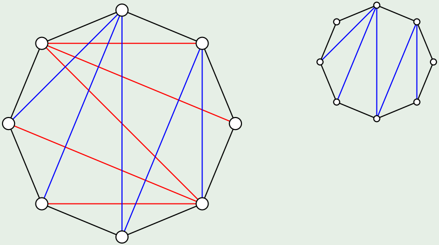




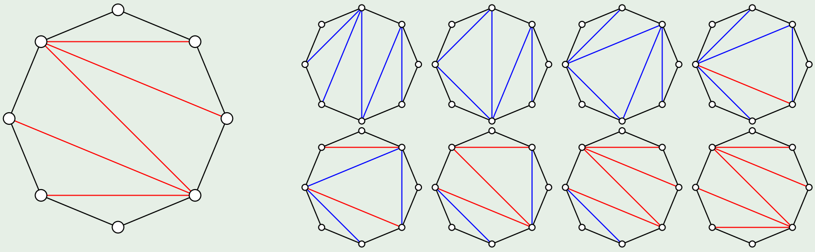
HighestDifference



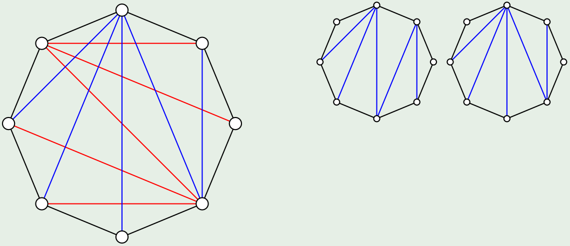
Optimal flip sequence



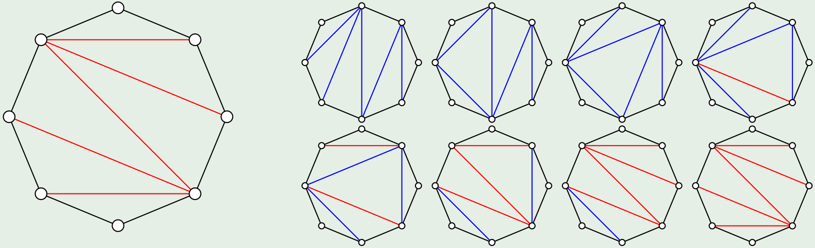
HighestDifference



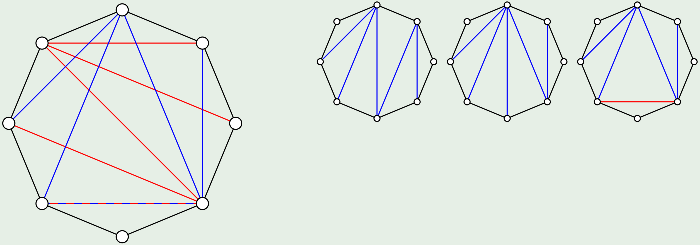
Optimal flip sequence



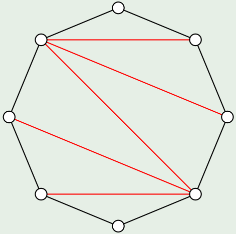
HighestDifference

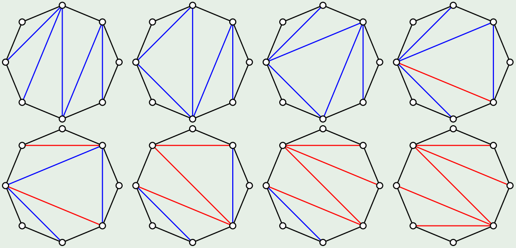


Optimal flip sequence

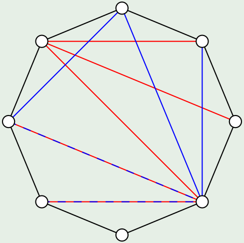


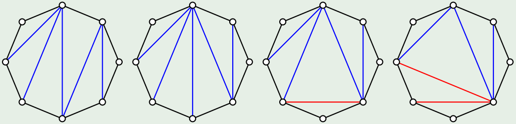
HighestDifference



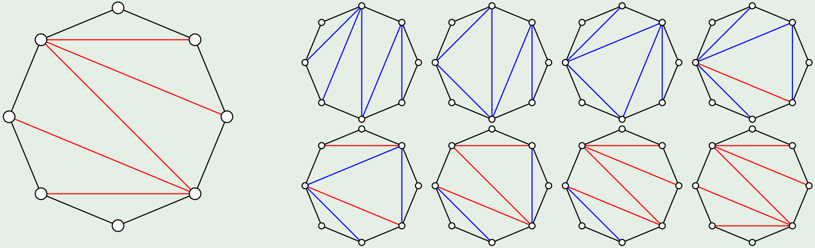


Optimal flip sequence

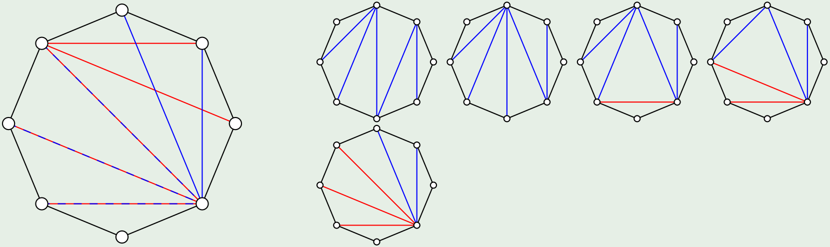




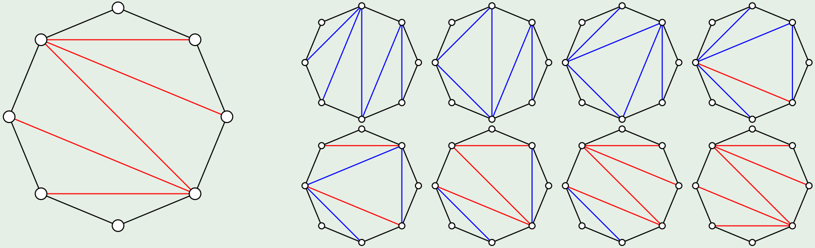
HighestDifference



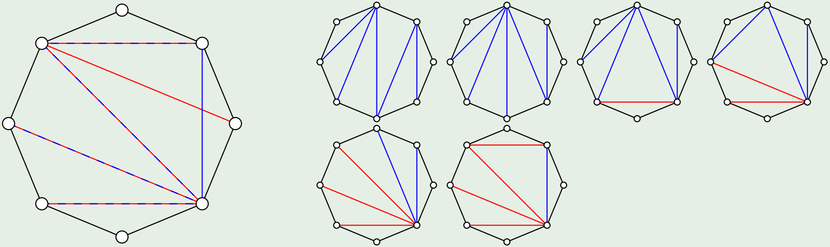
Optimal flip sequence



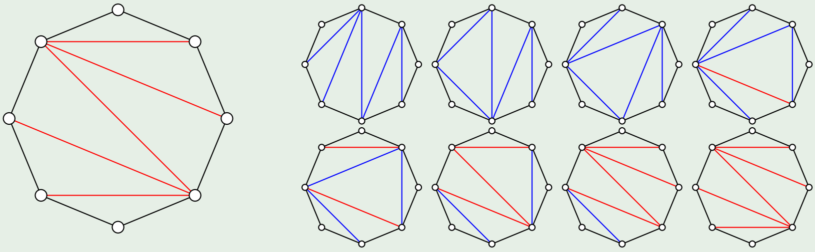
HighestDifference



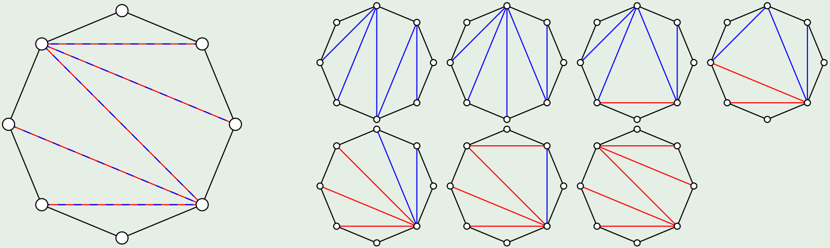
Optimal flip sequence



HighestDifference



Optimal flip sequence



Variants of the Heuristic

Input: start triangulation s, target triangulation t
Output: short flip sequence from s to t

Variants of the Heuristic

Input: start triangulation s, target triangulation t
Output: short flip sequence from s to t

Variants

MostIntersections Choose diagonal with maximum #(Intersections before flipping)

Variants of the Heuristic

Input: start triangulation s, target triangulation t
Output: short flip sequence from s to t

Variants

MostIntersections Choose diagonal with maximum
#(Intersections before flipping)

FewestIntersectionsAfter Choose Diagonal with minimum
#(Intersections after flipping)

Variants of the Heuristic

Input: start triangulation s, target triangulation t
Output: short flip sequence from s to t

Variants

MostIntersections Choose diagonal with maximum $\#$(Intersections before flipping)

FewestIntersectionsAfter Choose Diagonal with minimum $\#$(Intersections after flipping)

HighestDifference Choose diagonal with maximum $\Delta := \#$(Intersections before flipping) $-$ $\#$(Intersections after flipping).

Variants of the Heuristic

Input: start triangulation s, target triangulation t
Output: short flip sequence from s to t

Variants

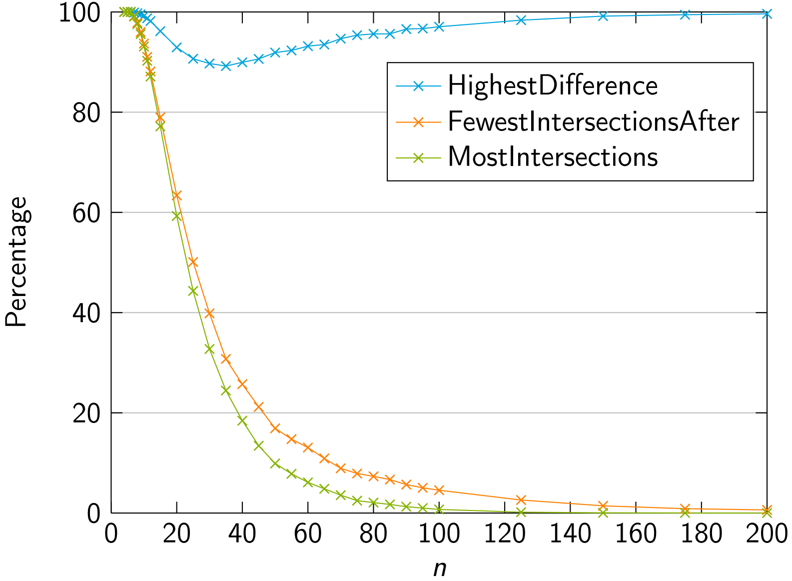
MostIntersections Choose diagonal with maximum $\#$(Intersections before flipping)

FewestIntersectionsAfter Choose Diagonal with minimum $\#$(Intersections after flipping)

HighestDifference Choose diagonal with maximum $\Delta := \#$(Intersections before flipping) $-$ $\#$(Intersections after flipping).

Runtime: $O(n^2)$

Comparing the Variants



Heuristics of Baril and Pallo

- First Heuristic by Pallo (2000):
  - Works on weight sequences representing binary trees
  - Constructs a lattice containing the trees
  - Length of path in lattice is upper bound for rotation distance

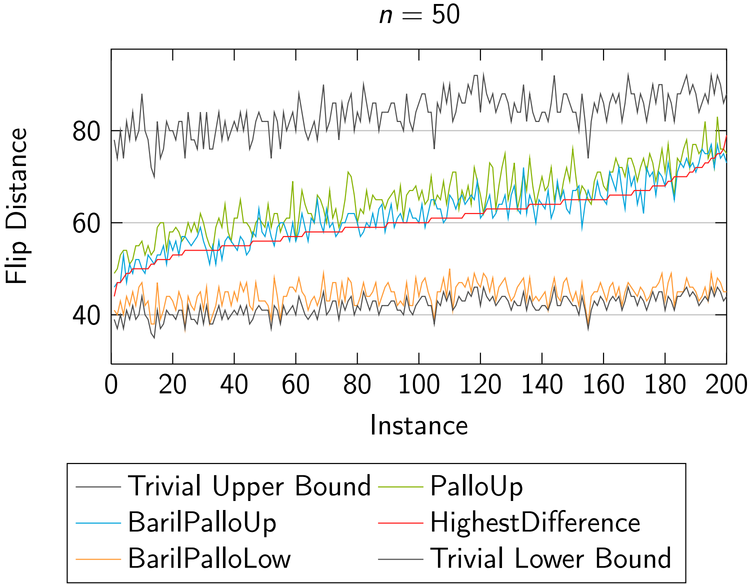
Heuristics of Baril and Pallo

- First Heuristic by Pallo (2000):
  - Works on weight sequences representing binary trees
  - Constructs a lattice containing the trees
  - Length of path in lattice is upper bound for rotation distance
- Heuristic by Baril and Pallo (2006):
  - Works on triangulations
  - Local transformation to create an additional common edge
  - Provides Upper and Lower Bound for the Flip Distance

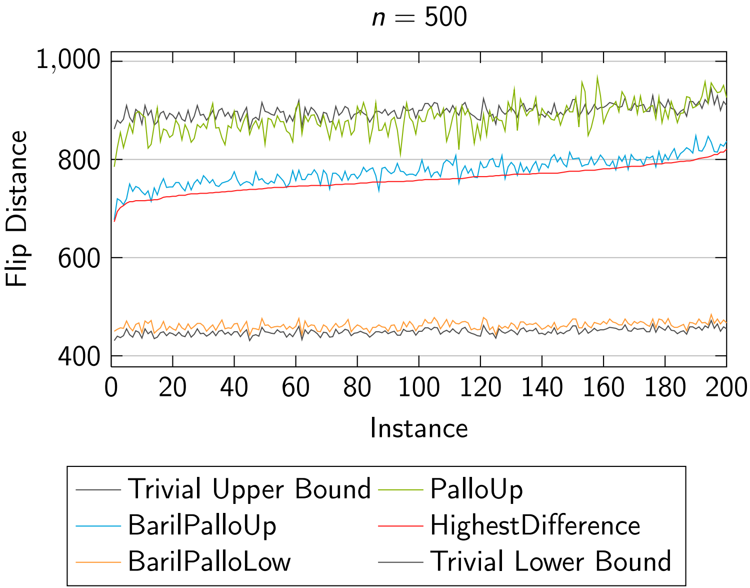
Heuristics of Baril and Pallo

- First Heuristic by Pallo (2000):
  - Works on weight sequences representing binary trees
  - Constructs a lattice containing the trees
  - Length of path in lattice is upper bound for rotation distance
- Heuristic by Baril and Pallo (2006):
  - Works on triangulations
  - Local transformation to create an additional common edge
  - Provides Upper and Lower Bound for the Flip Distance
- Runtime: $O(n^3)$

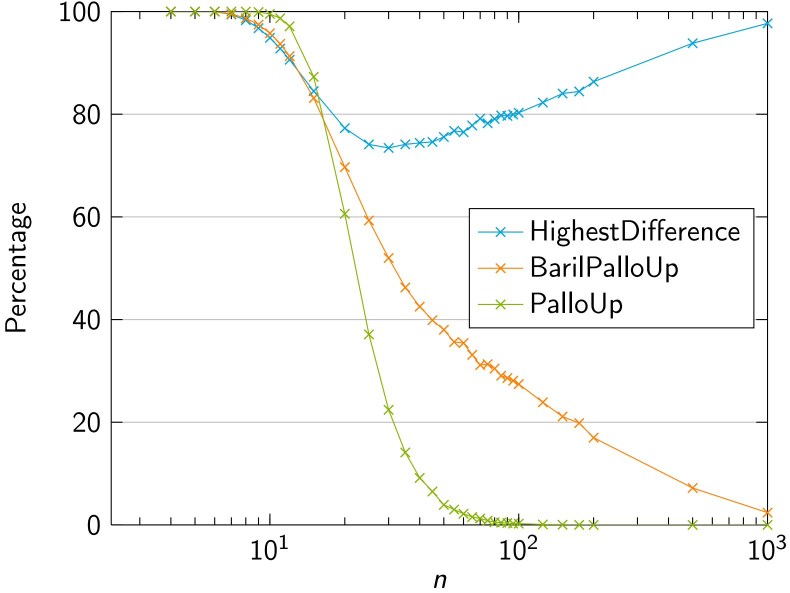
Comparing to other Heuristics



Comparing to other Heuristics



Comparing to other Heuristics



Conclusion

Summary

- Exact Algorithms: Improvements to BFS
- Heuristics: HighestDifference
  - Efficient compared to other heuristics
  - Easy to implement

Open Problems

- Polynomial-time algorithm
- NP-hardness proof
- Algorithms with approximation ratio < 2

References

Jean-Luc Baril and Jean-Marcel Pallo.
Efficient lower and upper bounds of the diagonal-flip distance between triangulations.
Information Processing Letters, 100(4):131–136, 2006.

Jean Pallo.
An efficient upper bound of the rotation distance of binary trees.
Information Processing Letters, 73(3–4):87–92, 2000.

Daniel D. Sleator, Robert E. Tarjan, and William P. Thurston.
Rotation distance, triangulations, and hyperbolic geometry.
Journal of the American Mathematical Society, 1(3):647–681, 1988.