# Faster bottleneck non-crossing matchings of points in convex position

Marko Savić and Miloš Stojaković

Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad

## Geometric perfect matchings

Vertices?
- Various planar objects
- **Points**

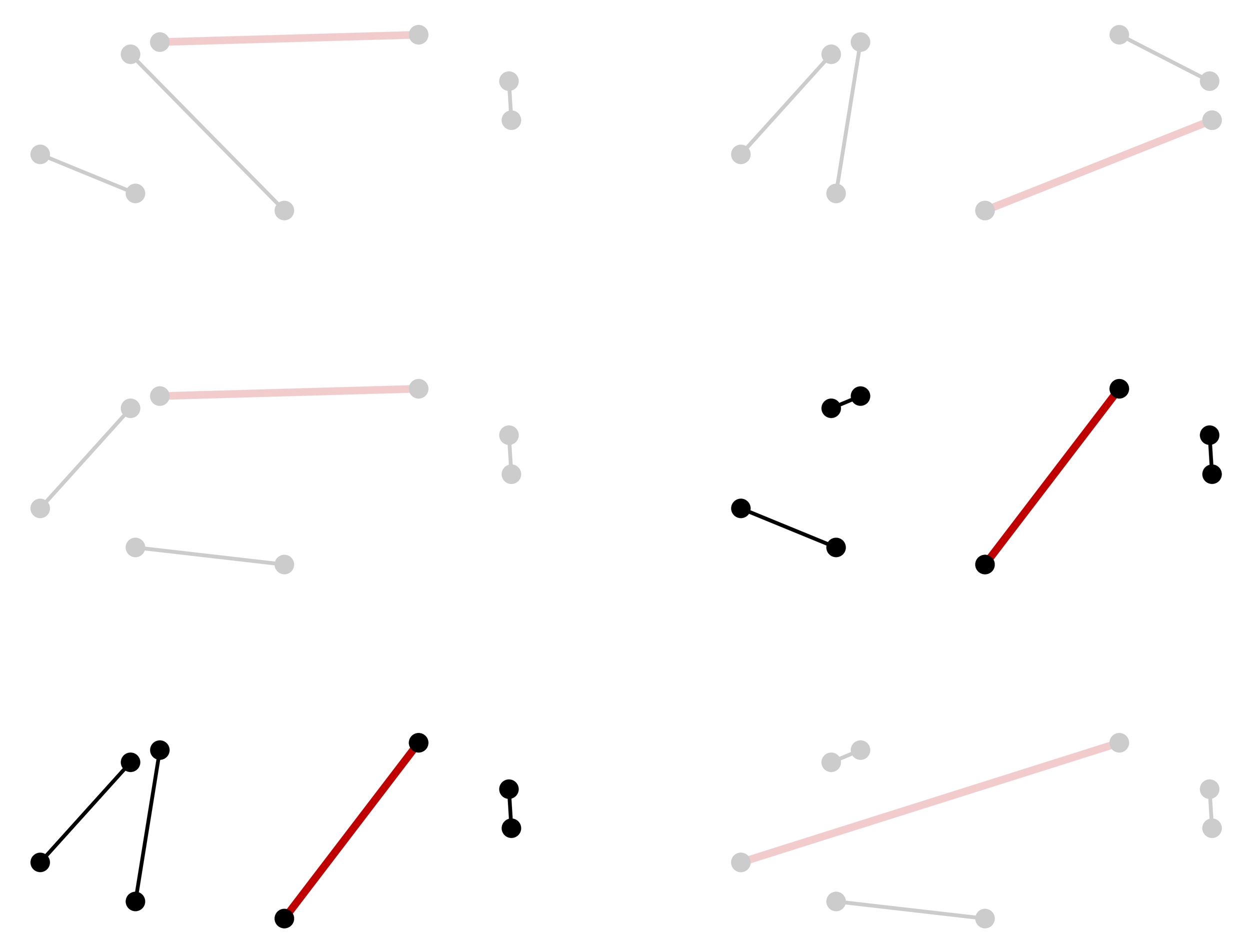
Connections?
- Curves
- **Straight-line segments**

Configuration?
- General position
- Special cases (**Convex position**, ...)

Crossings?
- Allowed
- **Not allowed**

Extremal?
- Minimize total sum of lengths
- Maximize total sum of lengths
- **Bottleneck (minimize the longest segment)**

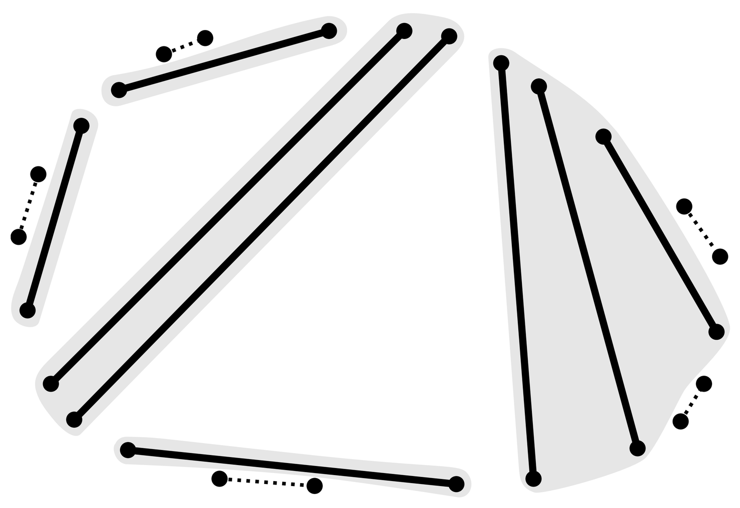
## Bottleneck non-crossing matchings, convex position



## Bottleneck non-crossing matchings - results

- (2010 Carlsson, Armbruster)
  - General position for bichromatic point set is NP-hard
- (2014 Abu-Affash, Carmi, Katz, Trablesi)
  - General position for monochromatic point set is NP-hard, no PTAS
  - Factor $2\sqrt{10}$ approximation algorithm
  - Convex position in $O(n^3)$
- (2016+ Savić, Stojaković)
  - Convex position in $O(n^2)$

## Edges, Diagonals, Cascades



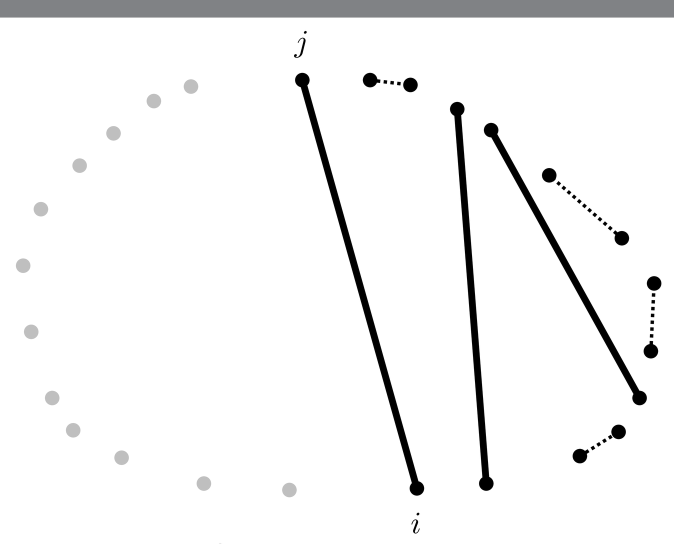
- Edge – neighbouring match.
- Diagonal – any other match.
- Cascade – sequence of "parallel" diagonals.

**Lemma:** There is a bottleneck matching with at most three cascades.

## Subproblems

$\textsc{Matching}(i, j)$

Optimal matching of points $\{i, \dots, j\}$ with a single cascade "parallel" to $(i, j)$.

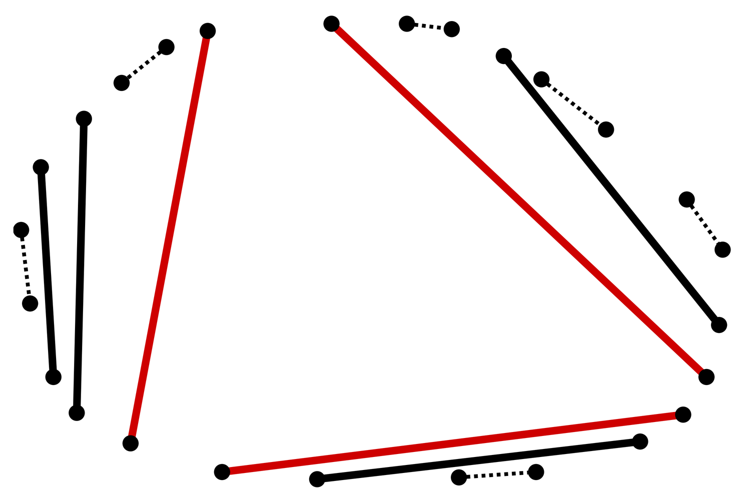



- **Lemma:** All subproblems can be solved in $O(n^2)$ total time.
- $(i, j)$ is necessary if it is contained in all solutions to $\textsc{Matching}(i, j)$.

## How to find a bottleneck matching?

- No cascades – Only two possible cases.
- Single cascade – A solution to one of the subproblems.
- Two cascades – Impossible.
- Three cascades
  - For all triples $(i, j, k)$ of vertices, combine the three subproblems defined by them. – $O(n^3)$ time - not an improvement.
  - Choose $(i, j)$ only from a set of at most linear size. – $O(n^2)$ time!

## Inner diagonals

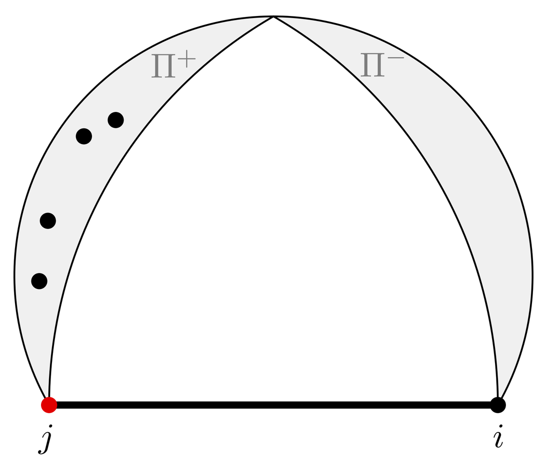


- **Lemma:** There is a bottleneck matching whose every inner diagonal is necessary.
- **Lemma:** There is a bottleneck matching with at least one inner diagonal being a candidate diagonal.
  - Candidate diagonal := necessary diagonal $\wedge$ (turning angle $\leq 2\pi/3$)

## Polarity

For a candidate diagonal $(i, j)$, we look at the points $\{i + 1, \dots, j - 1\}$.

- **Lemma:** They all must lie either in $\Pi^+$ or $\Pi^-$.




$(i, j)$ has positive polarity, and $j$ is pole.

- **Lemma:** No two candidate diagonals of the same polarity can have the same point as a pole.
- **Corollary:** There are $O(n)$ candidate diagonals.

## Algorithm

We search only through matchings in which one of the inner diagonals is a candidate diagonal.

- Solve subproblems
  - While doing so, find necessary and candidate diagonals.
- For each candidate diagonal $(i, j)$
  - For each point $k$ not in $\{i, \dots, j\}$
    - check if
      $\textsc{Matching}(i, j) \cup$
      $\textsc{Matching}(j + 1, k) \cup$
      $\textsc{Matching}(k + 1, i - 1)$
      is best so far.

Total running time is $O(n^2)$.