# Orbits and bichromatic bottleneck matchings of points in convex position

Marko Savić

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

Joint work with Miloš Stojaković

# Geometric perfect matchings

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

Configuration?

- General position
- Special cases
  (Convex position, . . . )

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

Configuration?

- General position
- Special cases
  (Convex position, . . . )

Connections?

- Curves
- Straight-line segments
- Various planar objects

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

Configuration?

- General position
- Special cases
  (Convex position, . . . )

Connections?

- Curves
- Straight-line segments
- Various planar objects

Crossings?

- Allowed
- Not allowed

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

Configuration?

- General position
- Special cases (Convex position, . . . )

Connections?

- Curves
- Straight-line segments
- Various planar objects

Crossings?

- Allowed
- Not allowed

Extremal?

- Minimize total sum of lengths
- Bottleneck (minimize the longest segment)

# Geometric perfect matchings

Vertices?

- Points
- Various planar objects

Configuration?

- General position
- Special cases
  (Convex position, . . . )

Connections?

- Curves
- Straight-line segments
- Various planar objects

Crossings?

- Allowed
- Not allowed

Extremal?

- Minimize total sum of lengths
- Bottleneck (minimize the longest segment)

Partitions

- Monochromatic
- Bichromatic

# Geometric perfect matchings

Vertices?

- **Points**
- Various planar objects

Configuration?

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

Connections?

- Curves
- **Straight-line segments**
- Various planar objects

Crossings?

- Allowed
- **Not allowed**

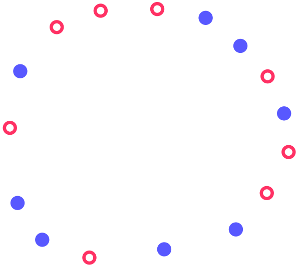
Extremal?

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

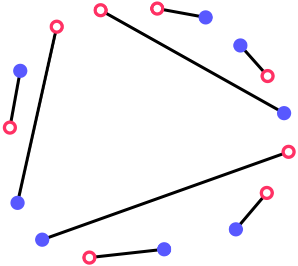
Partitions

- Monochromatic
- **Bichromatic**

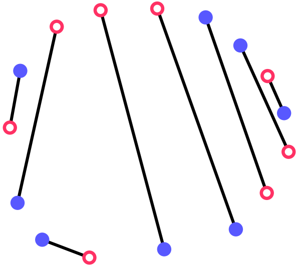
# Bichromatic non-crossing matching, convex position



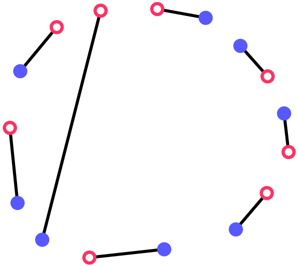
# Bichromatic non-crossing matching, convex position



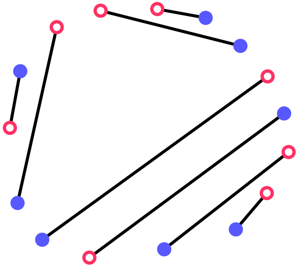
# Bichromatic non-crossing matching, convex position



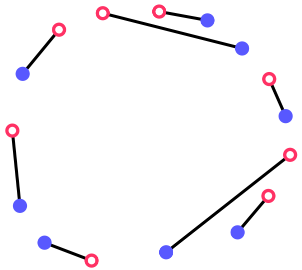
# Bichromatic non-crossing matching, convex position



# Bichromatic non-crossing matching, convex position



# Bichromatic non-crossing matching, convex position



Bottleneck

# Results

# Results

Monochromatic

- (2014 Abu-Affash, Carmi, Katz, Trablesi)
  - General position is NP-hard, no PTAS
  - Factor $2\sqrt{10}$ approximation algorithm
  - Convex position in $O(n^3)$
- (2015 Abu-Affash, Biniaz, Carmi, Maheshwari, Smid)
  - Large but not perfect matchings
- (2016 Savić, Stojaković)
  - Convex position in $O(n^2)$

# Results

Monochromatic

- (2014 Abu-Affash, Carmi, Katz, Trablesi)
  - General position is NP-hard, no PTAS
  - Factor $2\sqrt{10}$ approximation algorithm
  - Convex position in $O(n^3)$
- (2015 Abu-Affash, Biniaz, Carmi, Maheshwari, Smid)
  - Large but not perfect matchings
- (2016 Savić, Stojaković)
  - Convex position in $O(n^2)$

Bichromatic

- (2010 Carlsson, Armbruster)
  - General position is NP-hard
- (2014 Biniaz, Maheshwari, Smid)
  - Convex position in $O(n^3)$
  - Points on a circle in $O(n \log n)$
- (2018+ Savić, Stojaković)
  - Convex position in $O(n^2)$
  - Points on a circle in $O(n)$

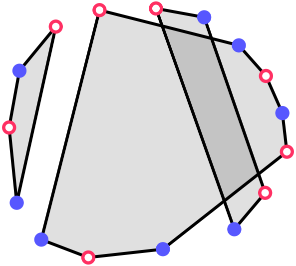
# Orbits

# Orbits

- $(i, j)$ is feasible $\quad \Leftrightarrow \quad \exists(\text{BNCM}\, M), (i, j) \in M$.

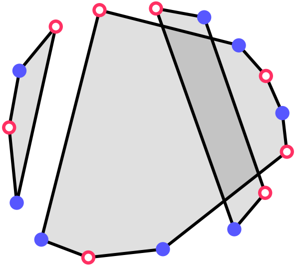
# Orbits

- $(i, j)$ is feasible $\quad \Leftrightarrow \quad \exists(\text{BNCM}M), (i, j) \in M.$
- Orbits - Partition of the point set such that
  $(i, j)$ is feasible $\quad \Leftrightarrow \quad \mathcal{O}(i) = \mathcal{O}(j)$



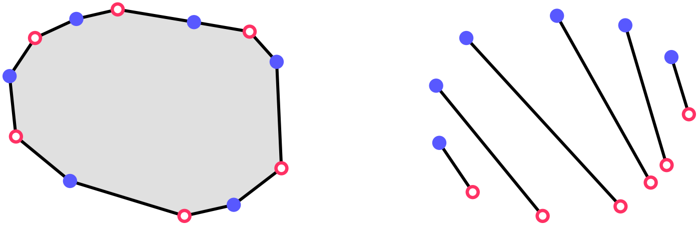
# Orbits

- $(i,j)$ is feasible $\quad\Leftrightarrow\quad \exists(\text{BNCM}M), (i,j) \in M$.
- Orbits - Partition of the point set such that $(i,j)$ is feasible $\quad\Leftrightarrow\quad \mathcal{O}(i) = \mathcal{O}(j)$

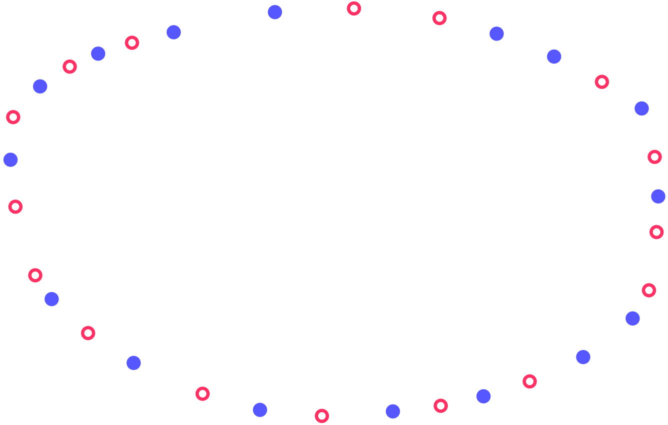


- Orbits can be computed in $O(n)$ time.

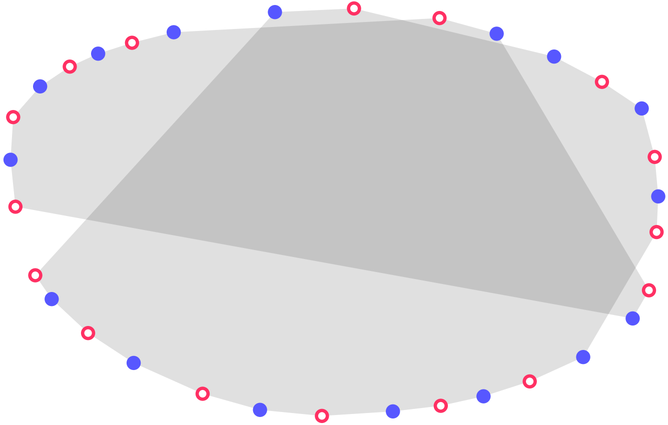
# Orbits - examples



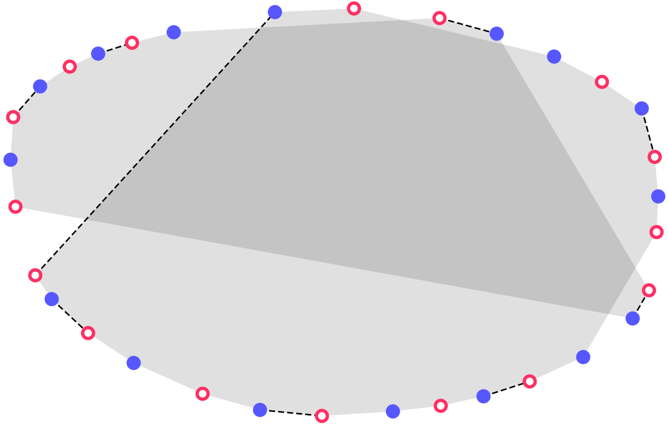
# Orbits - edges, diagonals



# Orbits - edges, diagonals

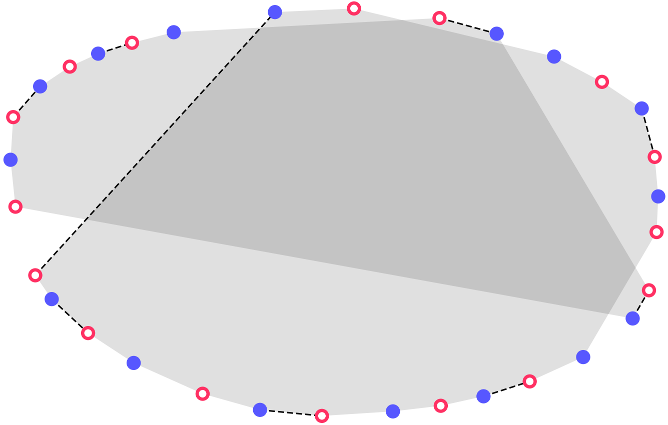


# Orbits - edges, diagonals



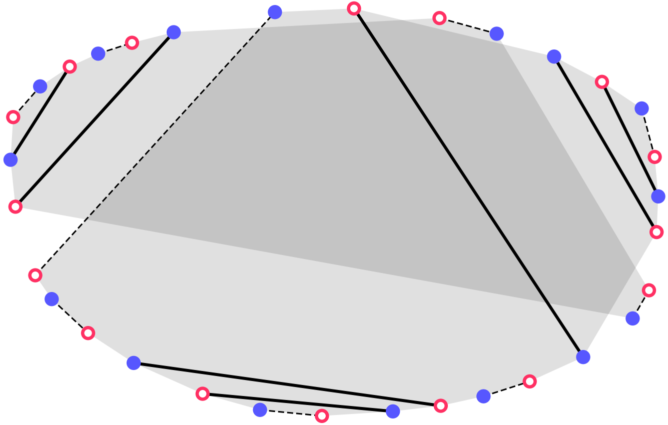
Edges

# Orbits - edges, diagonals



Edges (red-blue, blue-red)

# Orbits - edges, diagonals

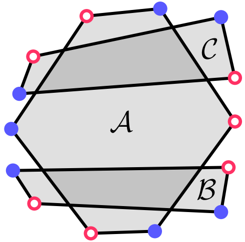


Diagonals

# Orbits - properties

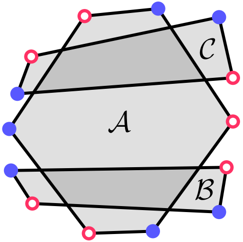
# Orbits - properties

- Orbits are synchronized.



# Orbits - properties

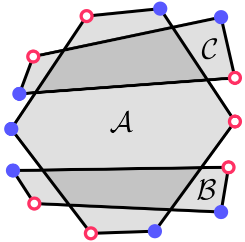
- Orbits are synchronized.




- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.

# Orbits - properties

- Orbits are synchronized.

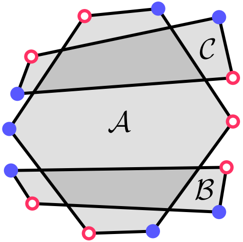



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.

# Orbits - properties

- Orbits are synchronized.

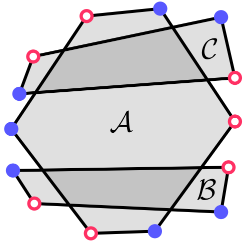



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.
- $\preceq$ is a total order.

# Orbits - properties

- Orbits are synchronized.

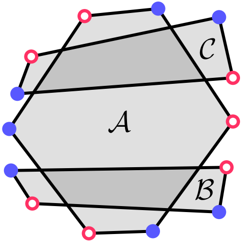



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.
- $\preceq$ is a total order.
- Orbit graph $\mathcal{G}$

# Orbits - properties

- Orbits are synchronized.

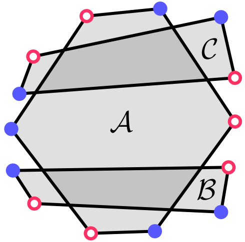



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.
- $\preceq$ is a total order.
- Orbit graph $\mathcal{G}$
  - $V(\mathcal{G}) =$ orbits
  - $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G}) \quad \Leftrightarrow \quad \mathcal{A} \preceq \mathcal{B} \wedge$ ($\mathcal{A}$ and $\mathcal{B}$ cross).

# Orbits - properties

- Orbits are synchronized.

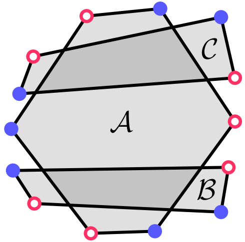



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.
- $\preceq$ is a total order.
- Orbit graph $\mathcal{G}$
  - $V(\mathcal{G}) =$ orbits
  - $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G}) \quad \Leftrightarrow \quad \mathcal{A} \preceq \mathcal{B} \wedge$ ($\mathcal{A}$ and $\mathcal{B}$ cross).
  - Hamiltonian path in each weakly connected component.

# Orbits - properties

- Orbits are synchronized.

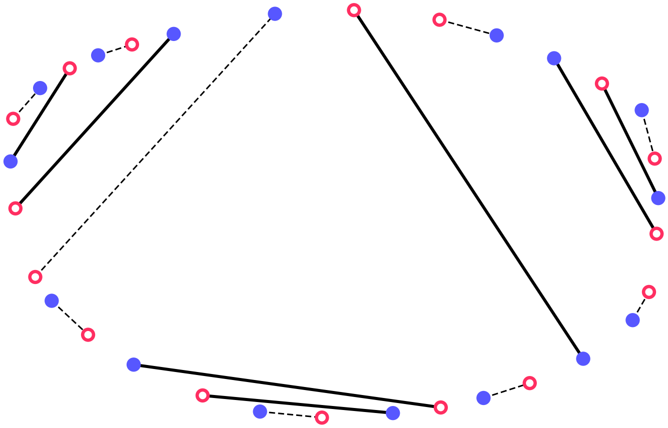



$\mathcal{C} \preceq \mathcal{A} \preceq \mathcal{B}$

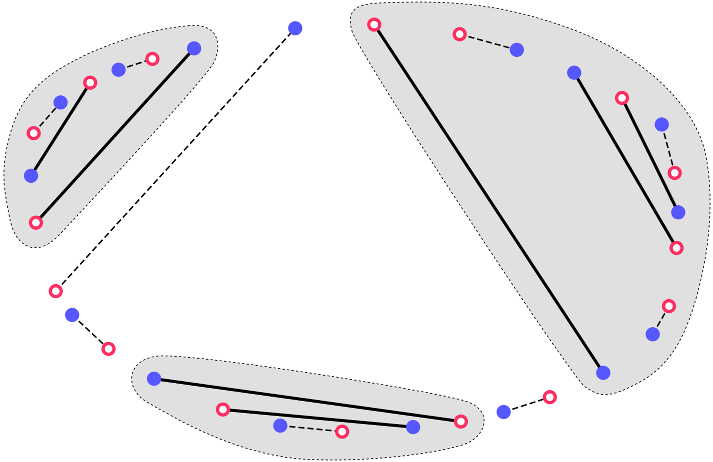
- $\mathcal{A} \preceq \mathcal{B} \quad \Leftrightarrow \quad$ No points of $\mathcal{B}$ behind red-blue edges of $\mathcal{A}$.
- $\preceq$ is a total order.
- Orbit graph $\mathcal{G}$
  - $V(\mathcal{G}) =$ orbits
  - $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G}) \quad \Leftrightarrow \quad \mathcal{A} \preceq \mathcal{B} \wedge$ ($\mathcal{A}$ and $\mathcal{B}$ cross).
  - Hamiltonian path in each weakly connected component.
- Total order and all Hamiltonian paths can be computed in $O(n)$ time.

# BBNCM - cascades

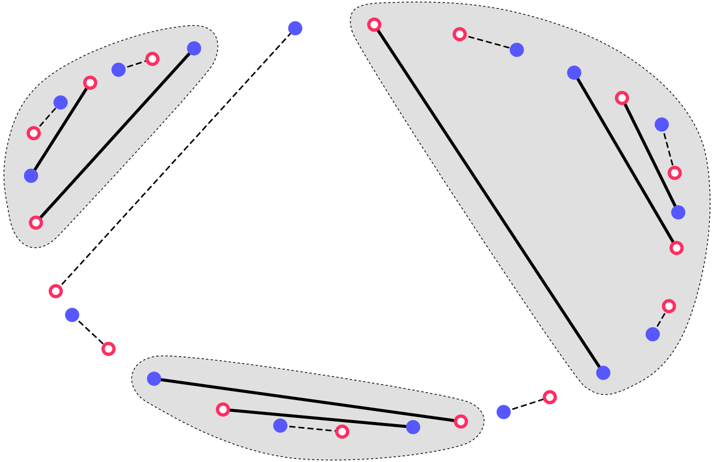
# BBNCM - cascades



# BBNCM - cascades

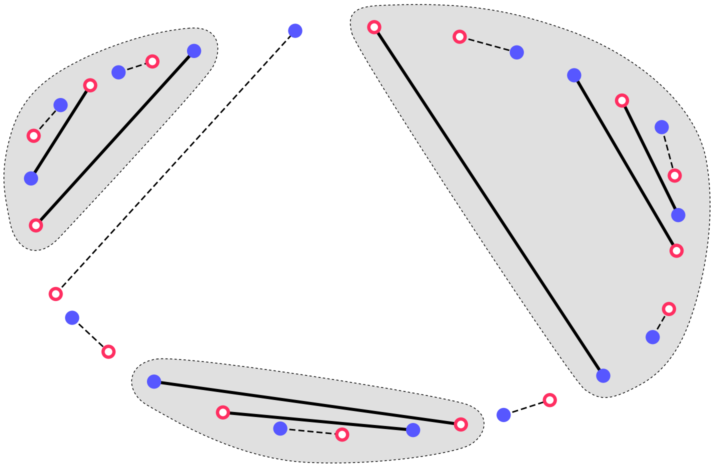


# BBNCM - cascades



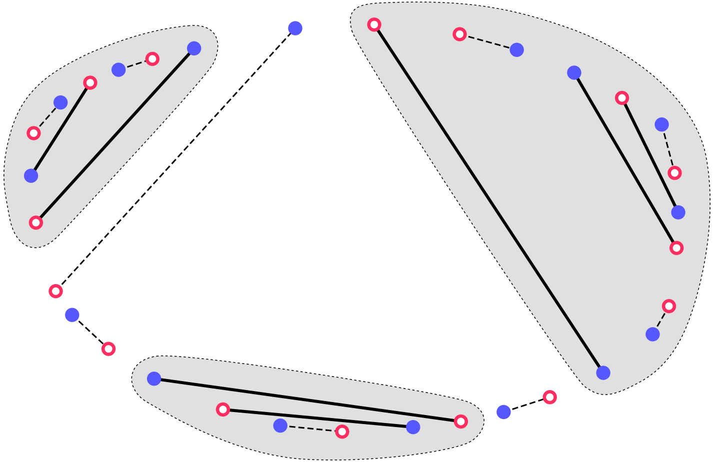
- There is a bottleneck matching with at most 3 cascades.

# BBNCM - cascades



- There is a bottleneck matching with at most 3 cascades.
- Subproblem: Find an optimal single-cascade matching in a subset of consecutive points.

# BBNCM - cascades



- There is a bottleneck matching with at most 3 cascades.
- Subproblem: Find an optimal single-cascade matching in a subset of consecutive points.
  - All subproblems can be solved in $O(n^2)$ total time.

# How to find a BBNCM?

# How to find a BBNCM?

- At most one cascade

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.
- Three cascades

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.
- Three cascades
  - Split in three parts and use subproblems.

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.
- Three cascades
  - Split in three parts and use subproblems.
    - $O(n^3)$ time - not an improvement.

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.
- Three cascades
  - Split in three parts and use subproblems.
    - $O(n^3)$ time - not an improvement.
  - It is enough to consider only $O(n^2)$ splits!

# How to find a BBNCM?

- At most one cascade
  - It is a solution to one of the subproblems.
- Two cascades
  - Not possible.
- Three cascades
  - Split in three parts and use subproblems.
    - $O(n^3)$ time - not an improvement.
  - It is enough to consider only $O(n^2)$ splits!
    - $O(n^2)$ time!

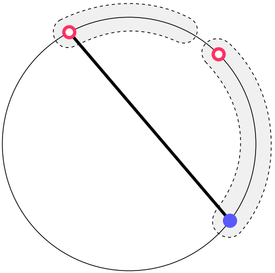
# Circle

# Circle

- There is a bottleneck matching using edges only.

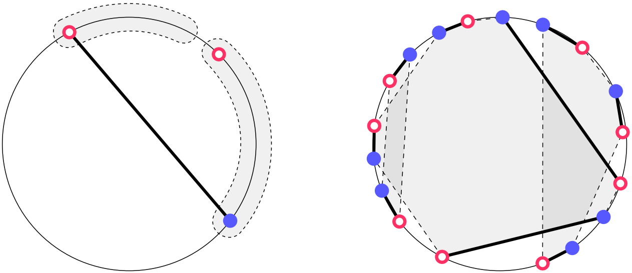
# Circle

- There is a bottleneck matching using edges only.



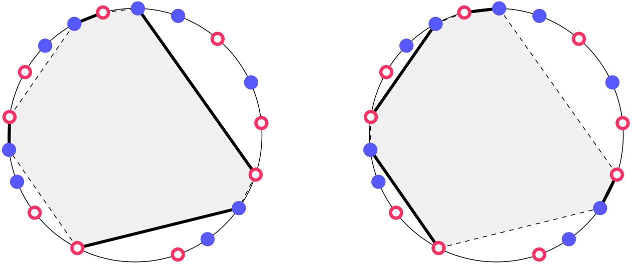
# Circle

- There is a bottleneck matching using edges only.



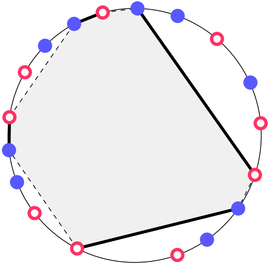
# Circle

- Two ways to choose every second edge from an orbit



# Circle

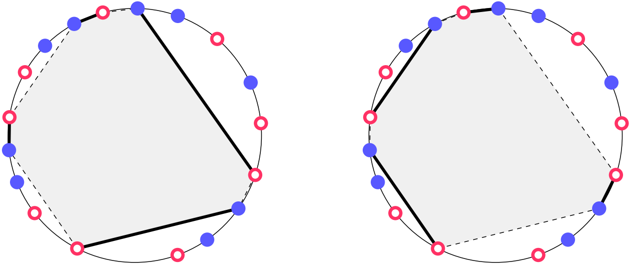
- Two ways to choose every second edge from an orbit



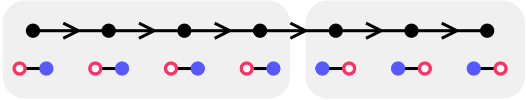
- If $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G})$ then we cannot choose blue-red edges from $\mathcal{A}$ and red-blue edges from $\mathcal{B}$.

# Circle

- Two ways to choose every second edge from an orbit

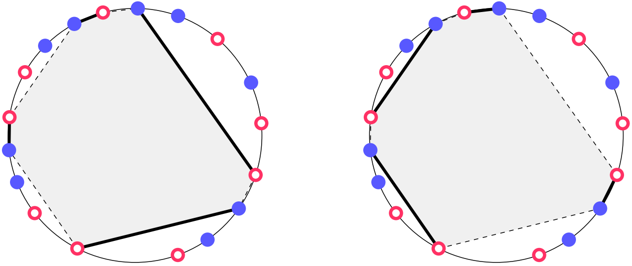


- If $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G})$ then we cannot choose blue-red edges from $\mathcal{A}$ and red-blue edges from $\mathcal{B}$.
- We traverse Hamiltonian paths for each component.

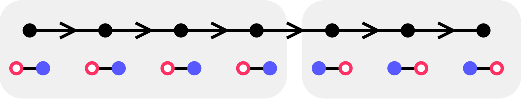


# Circle

- Two ways to choose every second edge from an orbit



- If $(\mathcal{A}, \mathcal{B}) \in E(\mathcal{G})$ then we cannot choose blue-red edges from $\mathcal{A}$ and red-blue edges from $\mathcal{B}$.
- We traverse Hamiltonian paths for each component.



- $O(n)$ time!

Thank you!