On augmenting matchings via bounded-length augmentations

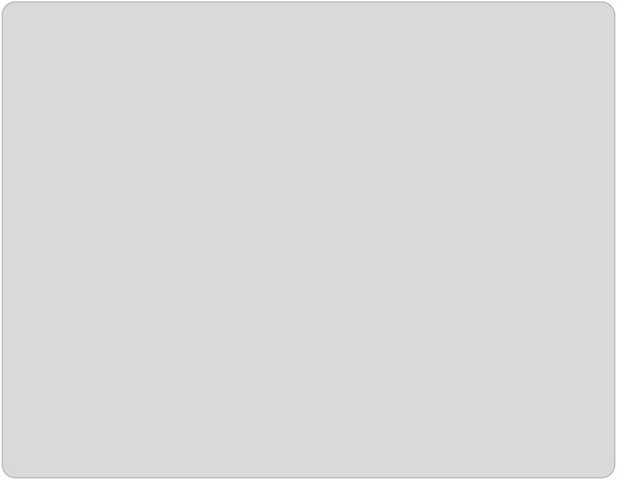
Julien Bensmail, Valentin Garnero, Nicolas Nisse

Université Nice-Sophia-Antipolis, France

LaBRI
April 28th, 2017

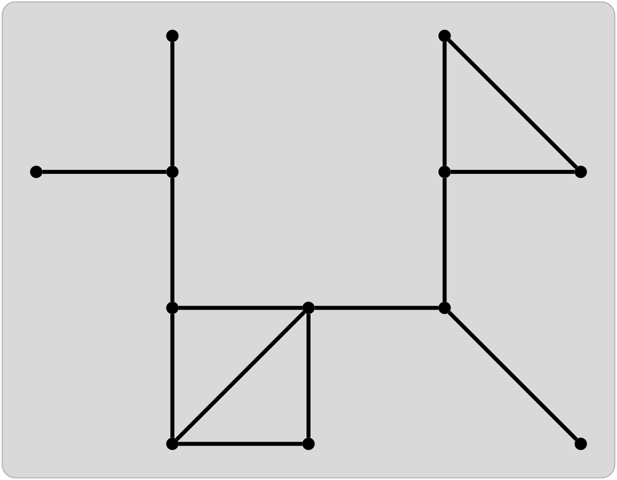
Introduction

Cast



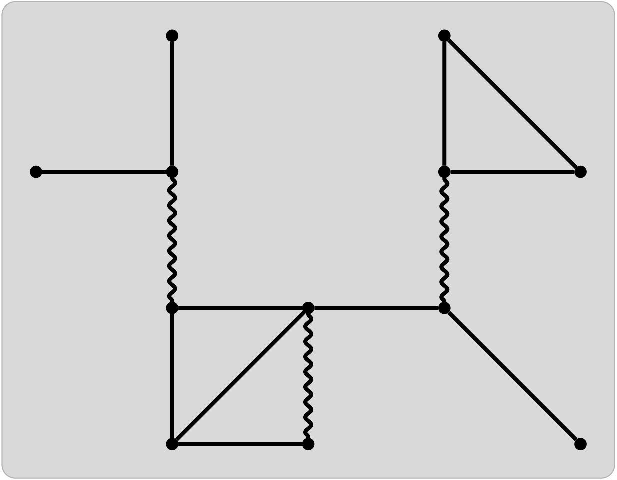
Cast

Graph



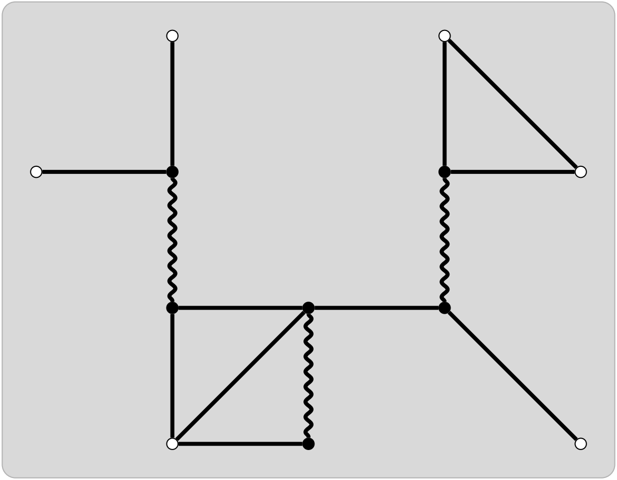
Cast

Graph, Matching



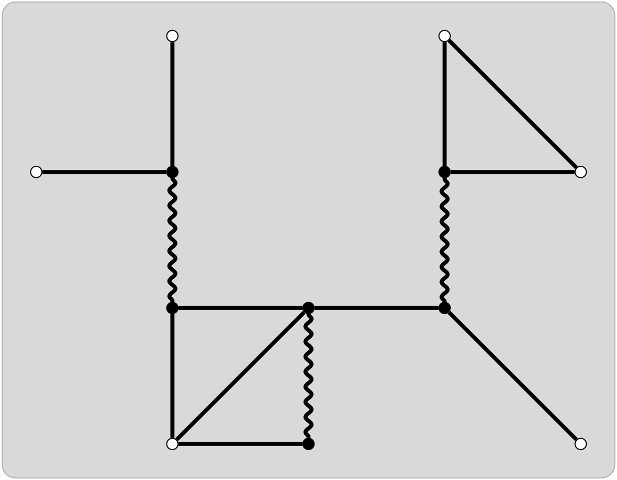
Cast

Graph, Matching
Exposed vertex, Covered vertex



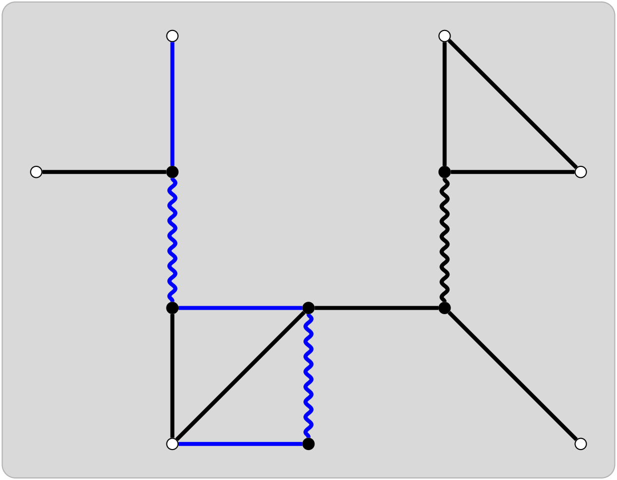
Augmenting a matching

Augmenting path, Augmentation



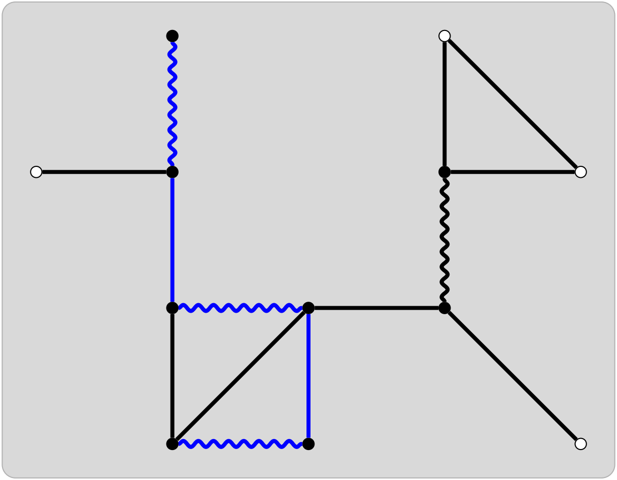
Augmenting a matching

Augmenting path, Augmentation



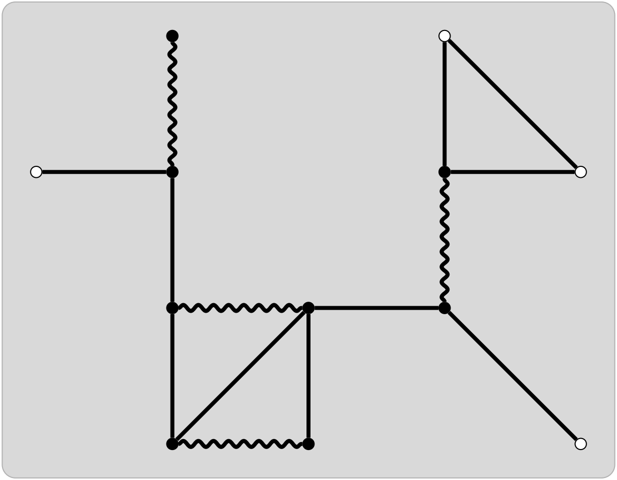
Augmenting a matching

Augmenting path, Augmentation



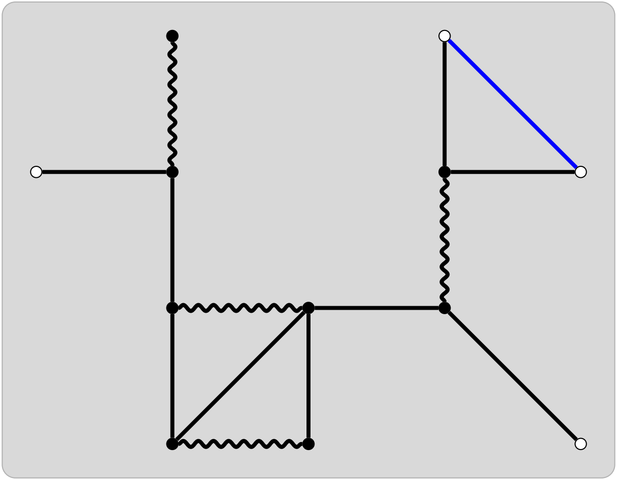
Augmenting a matching

Augmenting path, Augmentation



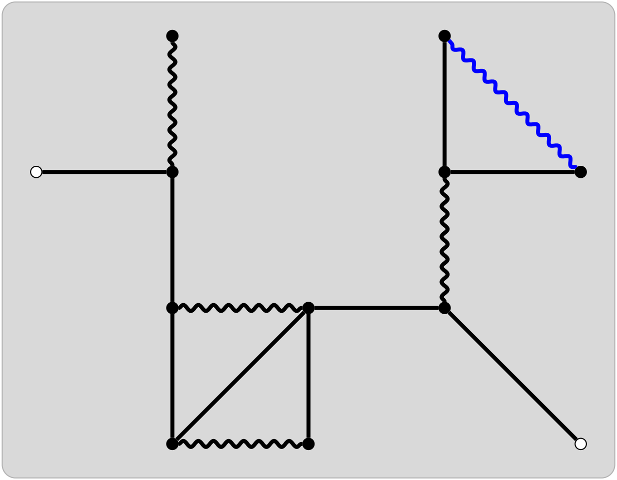
Augmenting a matching

Augmenting path, Augmentation



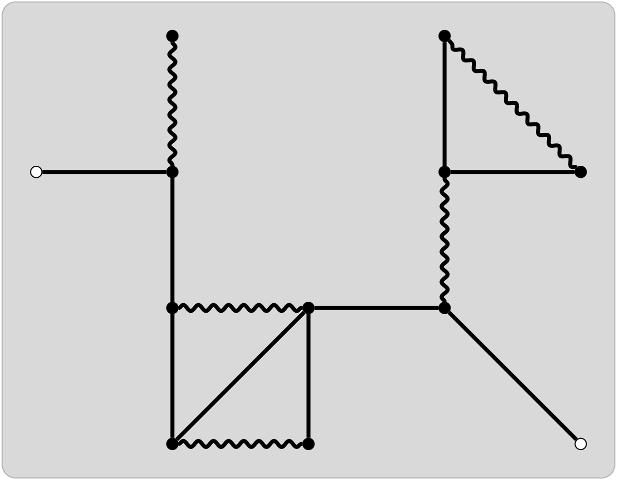
Augmenting a matching

Augmenting path, Augmentation



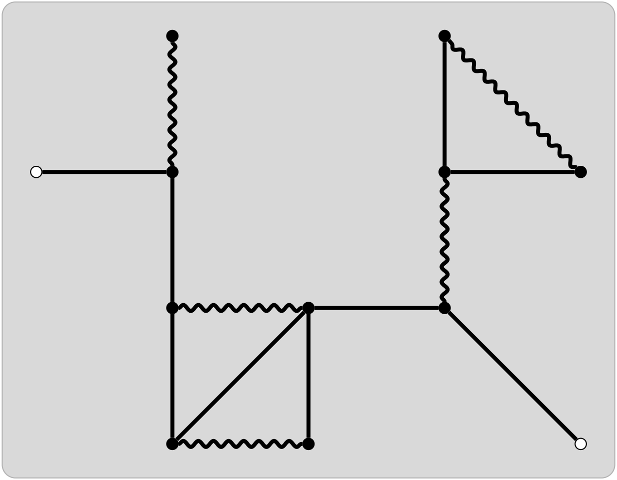
Augmenting a matching

Augmenting path, Augmentation



Augmenting a matching

Augmenting path, Augmentation



Augmentation ⇒ Bigger matching.

Berge and Edmonds' results

Maximum matching = Biggest matching.
$\mu(G)$ = Cardinality of a maximum matching of G.

Berge and Edmonds' results

Maximum matching = Biggest matching.
$\mu(G)$ = Cardinality of a maximum matching of G.

Theorem [Berge, 1957]

Maximum matching $\Leftrightarrow$ No augmenting path.

Berge and Edmonds' results

Maximum matching = Biggest matching.
$\mu(G)$ = Cardinality of a maximum matching of G.

Theorem [Berge, 1957]

Maximum matching $\Leftrightarrow$ No augmenting path.

Finding augmenting paths?

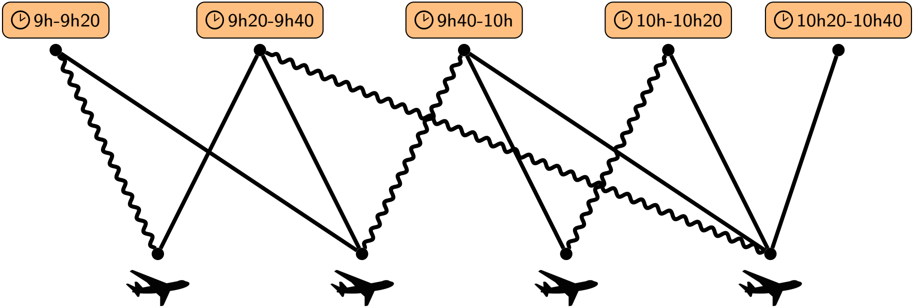
Theorem [Edmonds' Blossom Algorithm, 1965]

Detection in polynomial time.

Hence, $\mu(G)$ can be determined in poly-time.

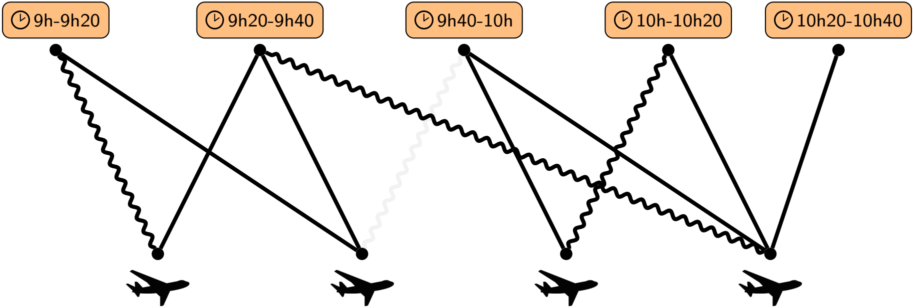
Today's motivation

Plane → Suitable landing slot times (edges) + Scheduled one (matching).



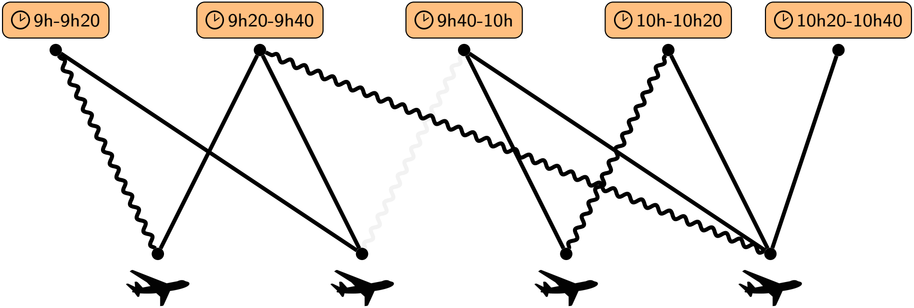
Today's motivation

Issue: Plane 2 is delayed...



Today's motivation

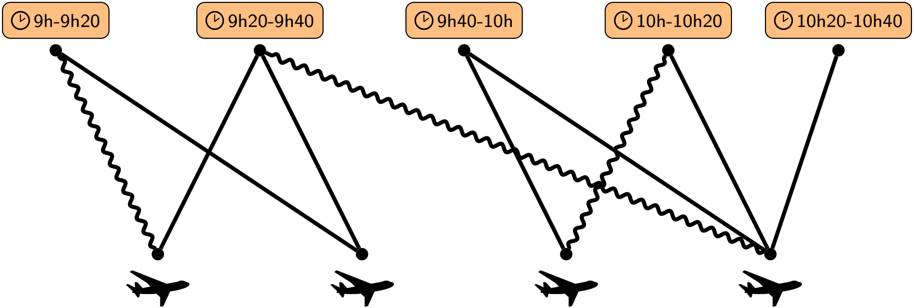
Issue: Plane 2 is delayed...



How to fix that??

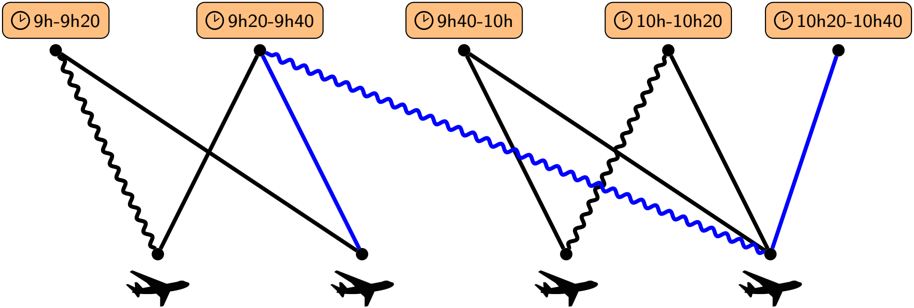
Motivation

Re-scheduling a lot is not acceptable! ⇒ Cannot start over from scratch.
⇒ Modify the matching "locally", via an augmentation.



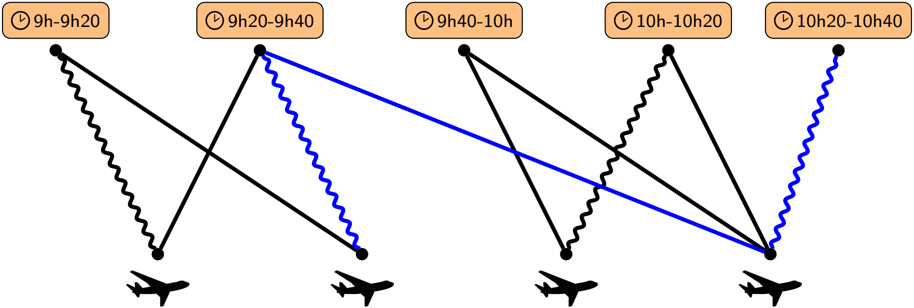
Motivation

Re-scheduling a lot is not acceptable! ⇒ Cannot start over from scratch.
⇒ Modify the matching “locally”, via an augmentation.



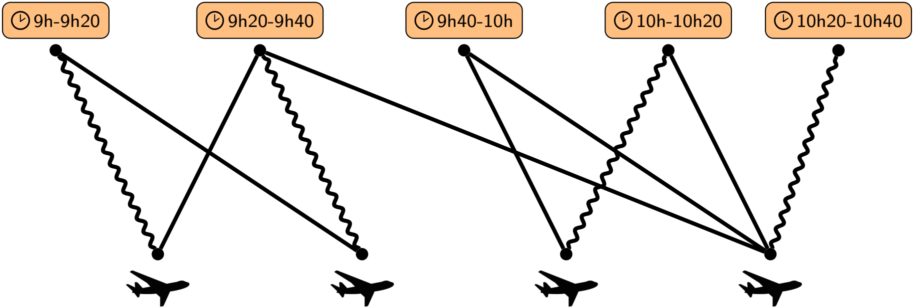
Motivation

Re-scheduling a lot is not acceptable! ⇒ Cannot start over from scratch.
⇒ Modify the matching "locally", via an augmentation.



Motivation

Re-scheduling a lot is not acceptable! ⇒ Cannot start over from scratch.
⇒ Modify the matching "locally", via an augmentation.



More formally

For practical reasons, only

1. 1-paths (= pick a free slot), and
2. 3-paths (= shift a busy slot)

should actually be augmented. Hence (≤ 3)-paths.

More formally

For practical reasons, only

1. 1-paths (= pick a free slot), and
2. 3-paths (= shift a busy slot)

should actually be augmented. Hence (≤ 3)-paths.

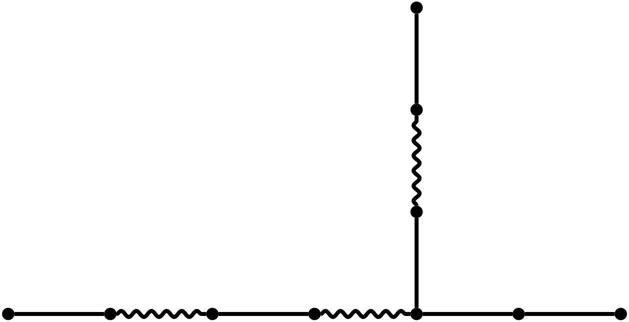
For odd $k \geq 1$, attain a largest matching via ($\leq k$)-augmentations?

$\mu_{\leq k}(G, M)$: Its cardinality for G equipped with M.

Note: $\mu_{\leq 1}(G, \emptyset) = \mu(G)$.

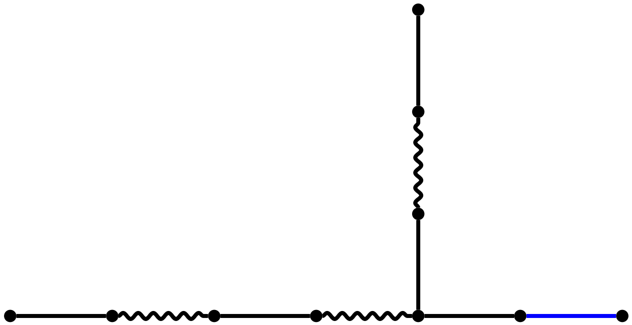
Note: order matters

$k = 5$. First attempt.



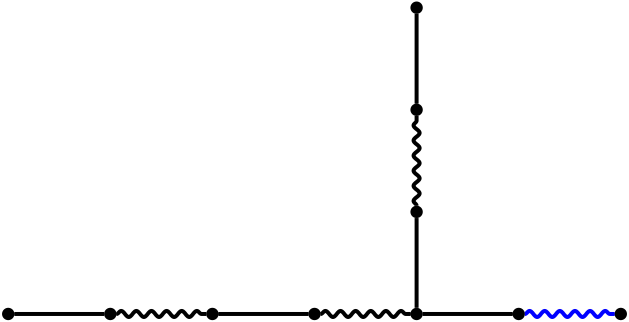
Note: order matters

$k = 5$. First attempt.



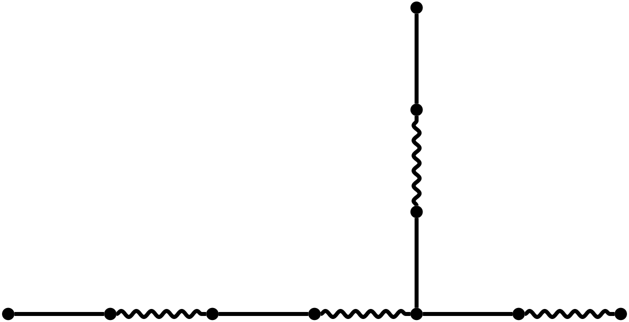
Note: order matters

$k = 5$. First attempt.



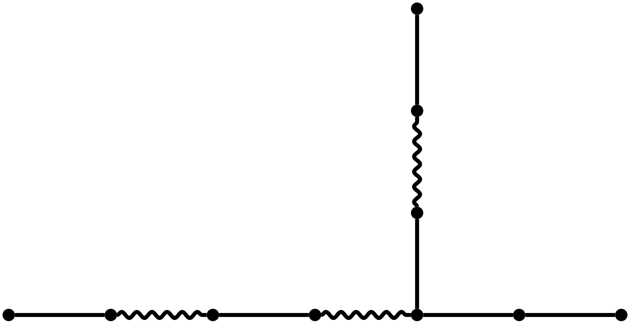
Note: order matters

$k = 5$. First attempt.



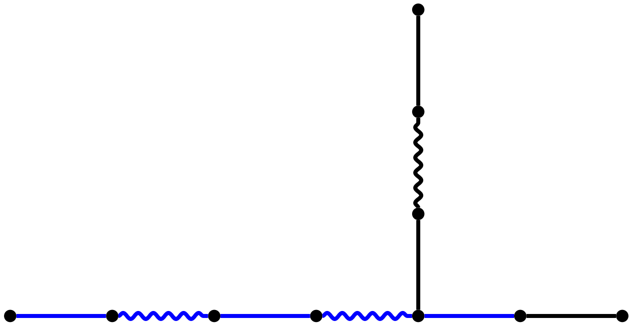
Note: order matters

$k = 5$. Second attempt.



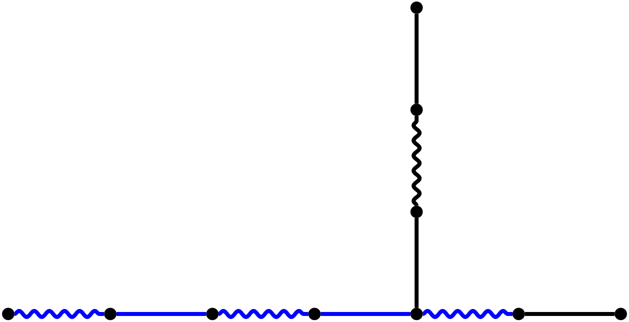
Note: order matters

$k = 5$. Second attempt.



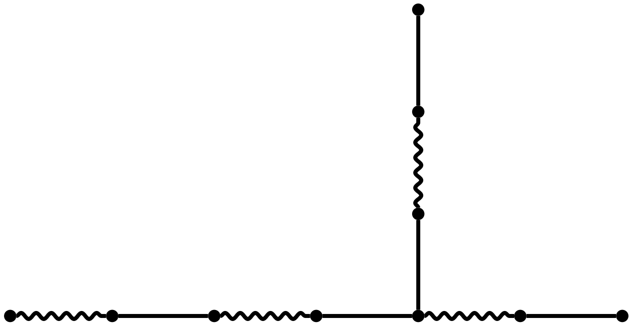
Note: order matters

$k = 5$. Second attempt.



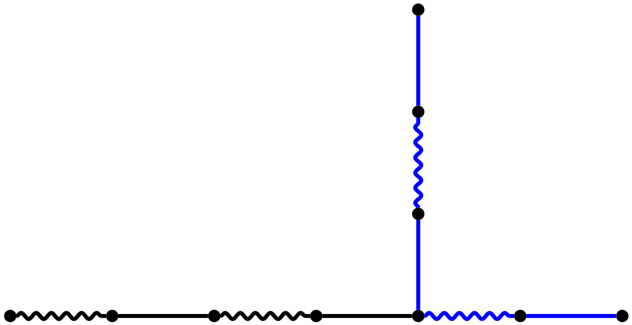
Note: order matters

$k = 5$. Second attempt.



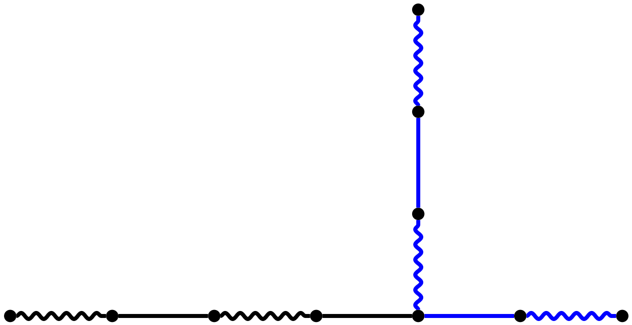
Note: order matters

$k = 5$. Second attempt.



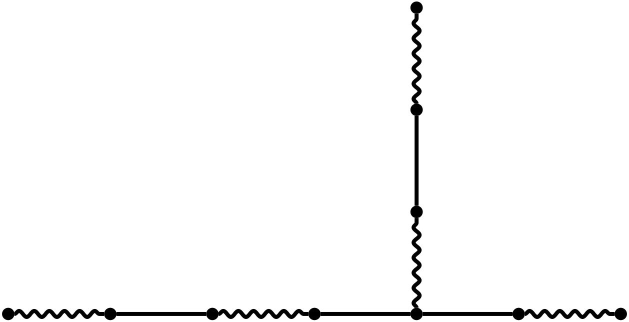
Note: order matters

$k = 5$. Second attempt.



Note: order matters

$k = 5$. Second attempt.



First dichotomy

($\leq k$)-MATCHING PROBLEM – ($\leq k$)-MP
Input: A graph G, and a matching M of G.
Question: What is the value of $\mu_{\leq k}(G, M)$?

First dichotomy

($\leq k$)-Matching Problem – ($\leq k$)-MP
Input: A graph G, and a matching M of G.
Question: What is the value of $\mu_{\leq k}(G, M)$?

Dichotomy on k:

Theorem [Nisse, Salch, Weber, 2015+]

($\leq k$)-MP is

- in P for $k = 1, 3$;
- NP-hard for every odd $k \geq 5$.

Latter statement true for planar bipartite graphs with $\Delta \leq 3$ and arb. large girth.

Towards a second dichotomy

Summary:

- For $k = 1, 3$, the problem is settled.
- For odd $k \geq 5$, NP-hard for graphs close to trees.

Towards a second dichotomy

Summary:

- For $k = 1, 3$, the problem is settled.
- For odd $k \geq 5$, NP-hard for graphs close to trees.

Complexity of $(\leq k)$-MP for trees?

Towards a second dichotomy

Summary:

- For $k = 1, 3$, the problem is settled.
- For odd $k \geq 5$, NP-hard for graphs close to trees.

Complexity of $(\leq k)$-MP for trees?

Today's talk:

- $(\leq k)$-MP is in P for caterpillars, subdivided stars, "sparse trees", etc.

Towards a second dichotomy

Summary:

- For $k = 1, 3$, the problem is settled.
- For odd $k \geq 5$, NP-hard for graphs close to trees.

Complexity of $(\leq k)$-MP for trees?

Today's talk:

- $(\leq k)$-MP is in P for caterpillars, subdivided stars, "sparse trees", etc.
- A modified version is NP-complete for trees.

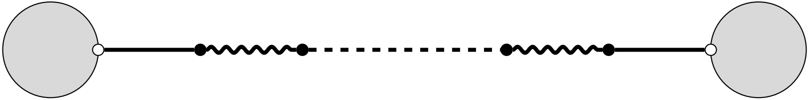
Positive results

Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

1st key idea: Consider exposed vertices joined **only once** by an augmenting path.

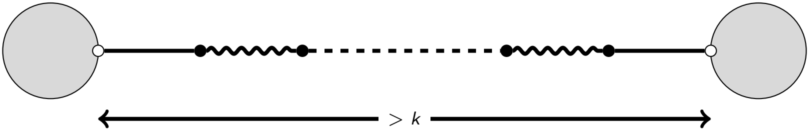


Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

1st key idea: Consider exposed vertices joined **only once** by an augmenting path.

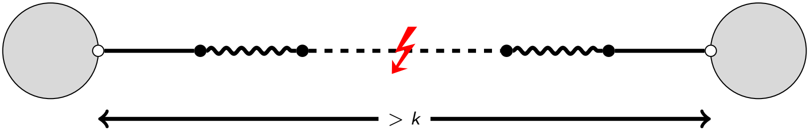


Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

1st key idea: Consider exposed vertices joined **only once** by an augmenting path.

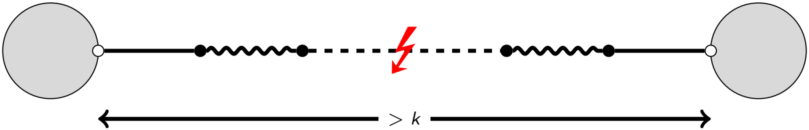


Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

1st key idea: Consider exposed vertices joined **only once** by an augmenting path.



$\Rightarrow$ Decompose the problem into two sub-problems.
In a path $\Rightarrow$ Exposed vertices have one on the left/right at distance $\leq k$.

Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

2nd key idea: We can augment paths joining "consecutive" exposed vertices only.

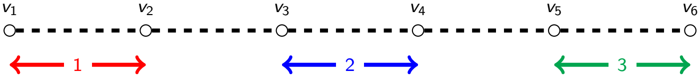
Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

2nd key idea: We can augment paths joining "consecutive" exposed vertices only.

3 $\Rightarrow$ The paths $v_1 \ldots v_2$, $v_3 \ldots v_4$ and $v_5 \ldots v_6$ have length $\leq k$ and alternate. So



yield the same matching.

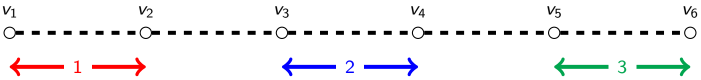
Easy case: paths

Theorem [Nisse, Salch, Weber, 2015+]

$(\leq k)$-MP is in P for paths.

2nd key idea: We can augment paths joining "consecutive" exposed vertices only.

3 ⇒ The paths $v_1 \ldots v_2$, $v_3 \ldots v_4$ and $v_5 \ldots v_6$ have length $\leq k$ and alternate. So



yield the same matching.

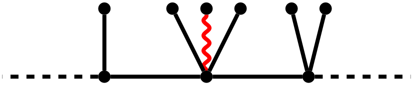
⇒ In a path, just go from left to right, and augment paths when possible. ∎

Caterpillars

Theorem [B., Garnero, Nisse, 2017+]

($\leq k$)-MP is in P for caterpillars.

Remark: Matched leaf edge $\Rightarrow$ Simplification.

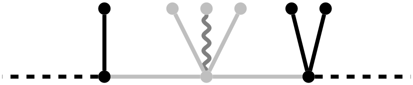


Caterpillars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for caterpillars.

Remark: Matched leaf edge $\Rightarrow$ Simplification.

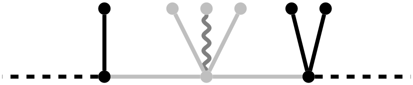


Caterpillars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for caterpillars.

Remark: Matched leaf edge $\Rightarrow$ Simplification.



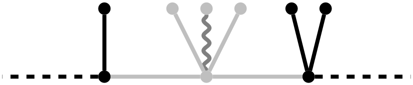
$\Rightarrow$ Being adjacent to two leaves is useless.

Caterpillars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for caterpillars.

Remark: Matched leaf edge $\Rightarrow$ Simplification.



$\Rightarrow$ Being adjacent to two leaves is useless.

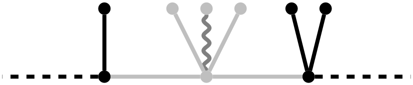
Focus on caterpillars with $\Delta = 3$ ($\sim$ paths).

Caterpillars

Theorem [B., Garnero, Nisse, 2017+]

($\leq k$)-MP is in P for caterpillars.

Remark: Matched leaf edge $\Rightarrow$ Simplification.



$\Rightarrow$ Being adjacent to two leaves is useless.

Focus on caterpillars with $\Delta = 3$ ($\sim$ paths).

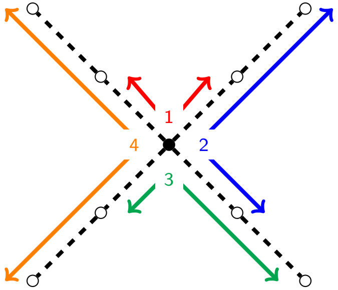
Essentially, again go from left to right. ∎

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

Claim: Augmentations through the root should behave in a path way:

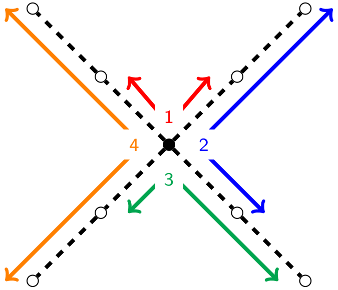


Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

($\leq k$)-MP is in P for subdivided stars.

Claim: Augmentations through the root should behave in a path way:



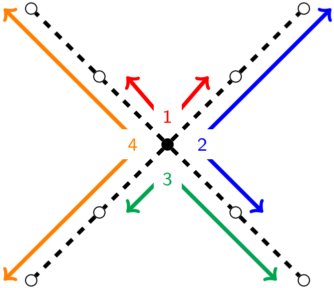
equivalent to

Subdivided stars

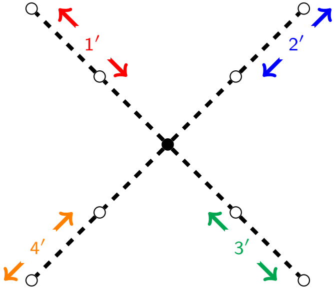
Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

Claim: Augmentations through the root should behave in a path way:



equivalent to



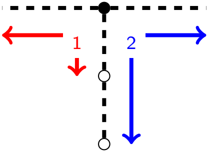
(because 1, 2, 3 and 4 are augmenting $(\leq k)$-paths.)

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

For an inner-branch, an equivalent augmentation can be performed

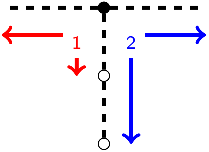


Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

For an inner-branch, an equivalent augmentation can be performed



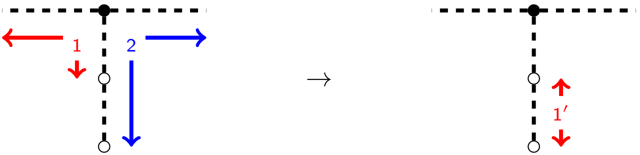
$\rightarrow$

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

For an inner-branch, an equivalent augmentation can be performed



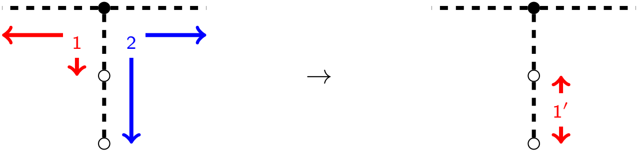
... and it retains the parity of the number of exposed vertices along that branch.

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

For an inner-branch, an equivalent augmentation can be performed



... and it retains the parity of the number of exposed vertices along that branch.

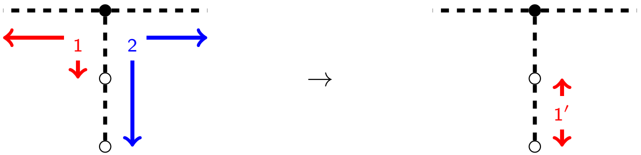
Augmentations through the root → Alter the parity of the end-branches only.

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

($\leq k$)-MP is in P for subdivided stars.

For an inner-branch, an equivalent augmentation can be performed



... and it retains the parity of the number of exposed vertices along that branch.

Augmentations through the root → Alter the parity of the end-branches only.

Along a branch with α exp. vertices, $\lfloor \alpha/2 \rfloor$ augmentations can be done (path):

- α even: all exposed can be matched along.
- otherwise: all but one.

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

So, we can reach a maximum matching by essentially:

1. Augment root paths to match two vertices from different branches;

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

So, we can reach a maximum matching by essentially:

1. Augment root paths to match two vertices from different branches;
2. Then finish off along the branches.

... and the first step is only useful if the two end-branches are "odd".

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

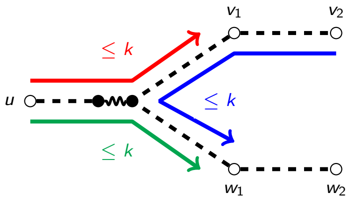
$(\leq k)$-MP is in P for subdivided stars.

So, we can reach a maximum matching by essentially:

1. Augment root paths to match two vertices from different branches;
2. Then finish off along the branches.

... and the first step is only useful if the two end-branches are "odd".

The first end-branch is the one having the "root" matching. Accessibility of a second odd branch can be checked via a BFS in an auxiliary graph:



Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

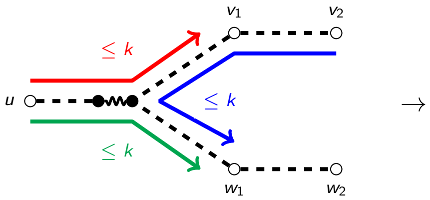
$(\leq k)$-MP is in P for subdivided stars.

So, we can reach a maximum matching by essentially:

1. Augment root paths to match two vertices from different branches;
2. Then finish off along the branches.

... and the first step is only useful if the two end-branches are “odd”.

The first end-branch is the one having the “root” matching. Accessibility of a second odd branch can be checked via a BFS in an auxiliary graph:



Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

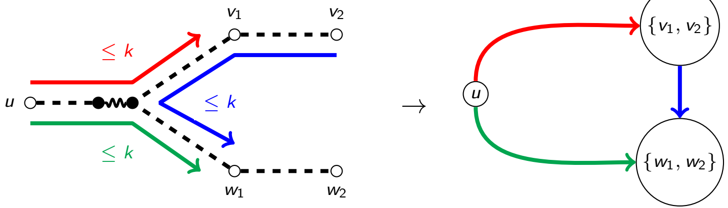
$(\leq k)$-MP is in P for subdivided stars.

So, we can reach a maximum matching by essentially:

1. Augment root paths to match two vertices from different branches;
2. Then finish off along the branches.

... and the first step is only useful if the two end-branches are "odd".

The first end-branch is the one having the "root" matching. Accessibility of a second odd branch can be checked via a BFS in an auxiliary graph:



Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

To summarize:

1. If necessary, do an augmentation involving the root.

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

To summarize:

1. If necessary, do an augmentation involving the root.
2. If possible, join two odd branches via root augmentations.

Subdivided stars

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for subdivided stars.

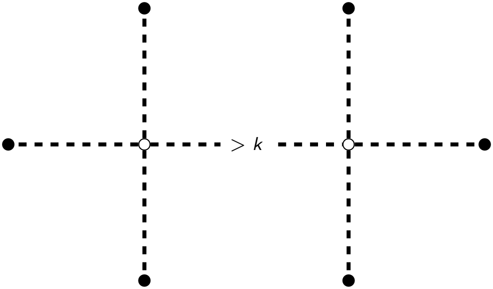
To summarize:

1. If necessary, do an augmentation involving the root.
2. If possible, join two odd branches via root augmentations.
3. Finally, match the remaining exposed vertices on the branches.

$\Rightarrow$ Polynomial-time algorithm. ■

Going to sparse trees

k-**sparse tree:** Vertices with degree ≥ 3 are at distance $> k$.

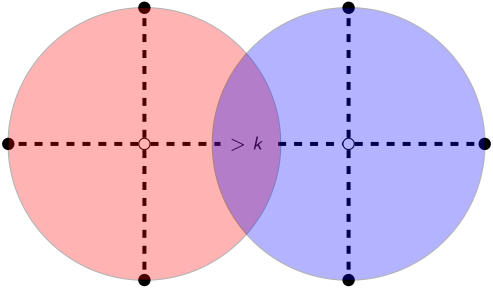


$(\leq k)$-MP for k-sparse trees

Theorem [B., Garnero, Nisse, 2017+]

$(\leq k)$-MP is in P for k-sparse trees.

Idea: Consider subdivided stars, and build a solution from bottom to top. ■



Negative results

Original intention

NP-hardness proof: Need some forcing mechanisms.

For $(\leq k)$-MP in trees, sounds hard because of the "$\leq k$" requirement.

Original intention

NP-hardness proof: Need some forcing mechanisms.

For $(\leq k)$-MP in trees, sounds hard because of the "$\leq k$" requirement.

$\Rightarrow$ What if we augment k-paths only?

Original intention

NP-hardness proof: Need some forcing mechanisms.

For $(\leq k)$-MP in trees, sounds hard because of the "$\leq k$" requirement.

$\Rightarrow$ What if we augment k-paths only?

$(= k)$-Matching Problem – $(= k)$-MP
Input: A graph G, and a matching M of G.
Question: What is the value of $\mu_{=k}(G, M)$?

Original intention

NP-hardness proof: Need some forcing mechanisms.

For $(\leq k)$-MP in trees, sounds hard because of the "$\leq k$" requirement.

$\Rightarrow$ What if we augment k-paths only?

$(= k)$-Matching Problem – $(= k)$-MP
Input: A graph G, and a matching M of G.
Question: What is the value of $\mu_{=k}(G, M)$?

Good news: Some properties of $(\leq k)$-MP derive to $(= k)$-MP:

- NP-hardness for odd $k \geq 5$;
- all polynomial-time algorithms for classes of trees.

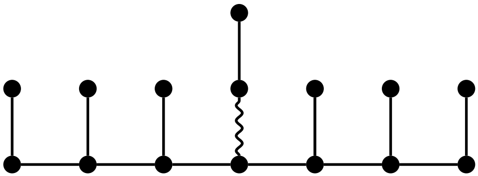
On (≤ 3)-MP and $(= 3)$-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



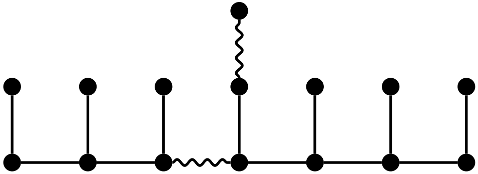
On (≤ 3)-MP and ($= 3$)-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

($= 3$)-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



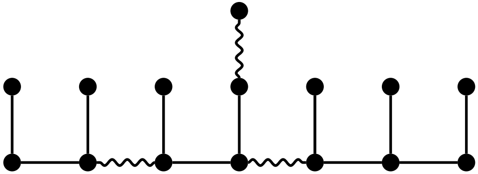
On (≤ 3)-MP and ($= 3$)-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

($= 3$)-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



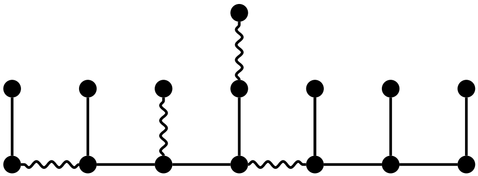
On (≤ 3)-MP and ($= 3$)-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

($= 3$)-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



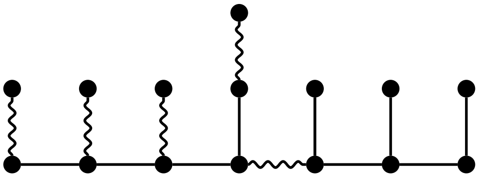
On (≤ 3)-MP and $(= 3)$-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



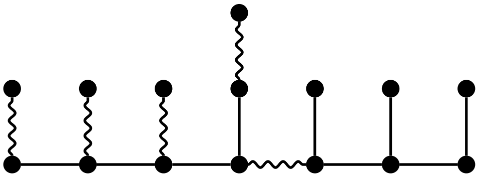
On (≤ 3)-MP and ($= 3$)-MP

Recall that (≤ 3)-MP is in P.

Theorem [B., Garnero, Nisse, 2017+]

($= 3$)-MP is NP-hard.

Proof: Reduction from 3-SAT. Just need variable gadget:



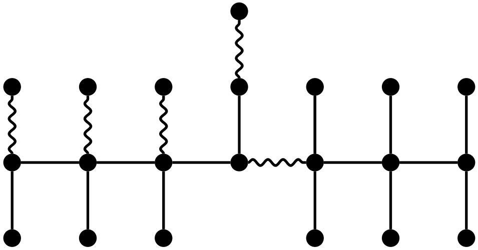
Longest sequence: Matched edges on all spikes of a single side.

On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Attach a leaf to the base of every spike. Previous remark still applies.



On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Add a variable gadget G_i for each x_i. Pushing left=True. Pushing right=False.

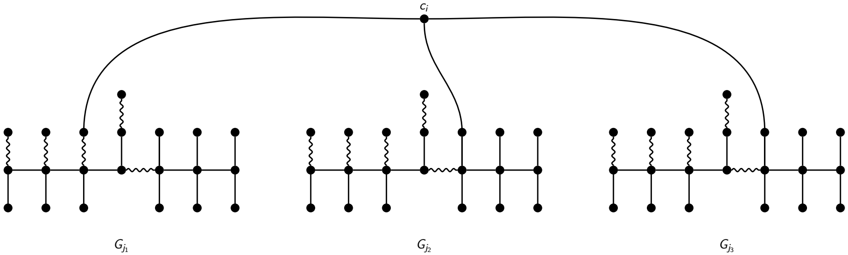
On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Add a variable gadget G_i for each x_i. Pushing left=True. Pushing right=False.

Next add a clause vertex c_i for every clause C_i, and, for every distinct literal ℓ_j it contains, join c_i and one non-used spike of G_i (left if positive, right otherwise).



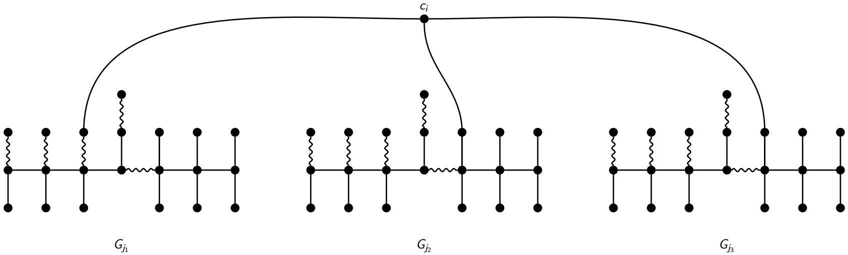
On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Add a variable gadget G_i for each x_i. Pushing left=True. Pushing right=False.

Next add a clause vertex c_i for every clause C_i, and, for every distinct literal ℓ_j it contains, join c_i and one non-used spike of G_i (left if positive, right otherwise).



$\Rightarrow$ One additional augmentation covering c_i can be done.

On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Maximum $\#$ of 3-augmentations:

1. For every G_i, push the matching to the left (x_i true) or to the right (x_i false).
2. For every c_i, do an additional augmentation (if made true by a literal).

On (≤ 3)-MP and $(= 3)$-MP

Theorem [B., Garnero, Nisse, 2017+]

$(= 3)$-MP is NP-hard.

Maximum # of 3-augmentations:

1. For every G_i, push the matching to the left (x_i true) or to the right (x_i false).
2. For every c_i, do an additional augmentation (if made true by a literal).

$\Rightarrow$ Maximum $\mu_{=3}$ achievable is

$$(\#\text{variables} \cdot \#\text{spikes}) + \#\text{clauses},$$

which is attainable iff F is satisfiable. ■

About the reduction

We have $\Delta \leq 4$ in the reduction.

About the reduction

We have $\Delta \leq 4$ in the reduction. Furthermore:

- If F planar, then the reduced graph is planar.

About the reduction

We have $\Delta \leq 4$ in the reduction. Furthermore:

- If F planar, then the reduced graph is planar.
- All cycles go through c_i's and variables gadgets:

About the reduction

We have $\Delta \leq 4$ in the reduction. Furthermore:

- If F planar, then the reduced graph is planar.
- All cycles go through c_i's and variables gadgets:
  - Conveniently choose the joined spikes $\Rightarrow$ Bipartite.
  - Same $\Rightarrow$ Arbitrarily large girth.

About the reduction

We have $\Delta \leq 4$ in the reduction. Furthermore:

- If F planar, then the reduced graph is planar.
- All cycles go through c_i's and variables gadgets:
  - Conveniently choose the joined spikes $\Rightarrow$ Bipartite.
  - Same $\Rightarrow$ Arbitrarily large girth.

$+$ by slight modifications, we can also guarantee $\Delta \leq 3$.

$(= k)$-MP in trees for non-fixed k

Modified version:

$(=)$-Matching Problem – $(=)$-MP
Input: A graph G, a matching M of G, and an odd $k \geq 1$.
Question: What is the value of $\mu_{=k}(G, M)$?

$(=k)$-MP in trees for non-fixed k

Modified version:

$(=)$-Matching Problem – $(=)$-MP
Input: A graph G, a matching M of G, and an odd $k \geq 1$.
Question: What is the value of $\mu_{=k}(G, M)$?

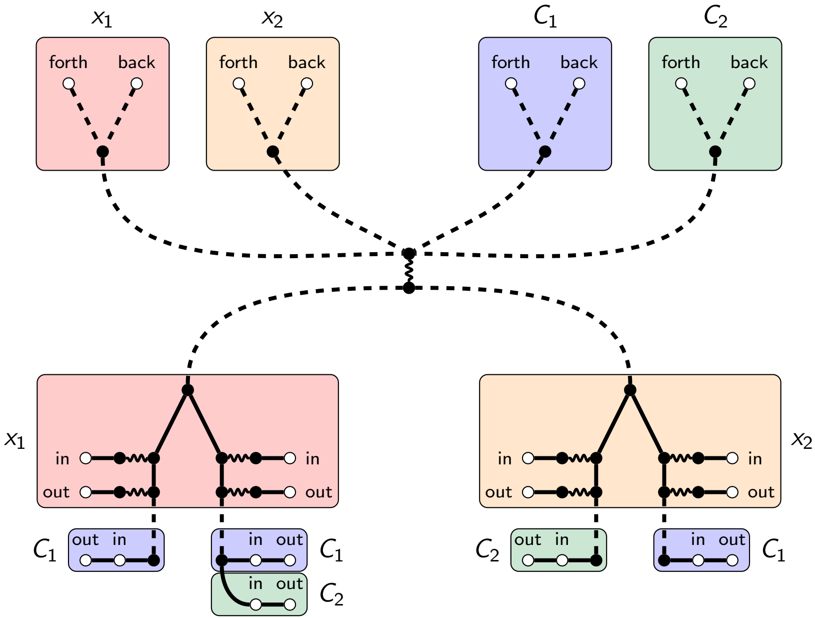
Negative result for trees:

Theorem [B., Garnero, Nisse, 2017+]

$(=)$-MP is NP-hard for trees.

Proof (sketch): Reduction from 3-SAT.

$(=)$-MP in trees



x_1
forth
back
x_2
forth
back
C_1
forth
back
C_2
forth
back
x_1
in
out
in
out
C_1
out in
in out
C_1
in out
C_2
x_2
in
out
in
out
C_2
out in
in out
C_1

(=)-MP in trees

Theorem [B., Garnero, Nisse, 2017+]

(=)-MP is NP-hard for trees.

Lengths of the dashed paths chosen so that:

- for each x_i, open either the *true* or *false* gate;
- for each C_i, reach only the arrival points.

(=)-MP in trees

Theorem [B., Garnero, Nisse, 2017+]

(=)-MP is NP-hard for trees.

Lengths of the dashed paths chosen so that:

- for each x_i, open either the *true* or *false* gate;
- for each C_i, reach only the arrival points.

⇒ Needed k depends on #clauses and #variables. ■

Conclusion

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.

Conclusion and perspectives

- Status of ($\leq k$)-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?
  - subdivided combs?

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?
  - subdivided combs?
  - etc.

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?
  - subdivided combs?
  - etc.
- Dynamic programming yields algorithms.

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?
  - subdivided combs?
  - etc.
- Dynamic programming yields algorithms.
- What about $(= k)$-MP in trees?
- Appealing case: $k = 3$.

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
    - trees with $\Delta \leq 3$?
    - subdivided combs?
    - etc.
- Dynamic programming yields algorithms.
- What about $(= k)$-MP in trees?
- Appealing case: $k = 3$.
- Other classes of graphs?
- e.g. interval graphs, other sparse classes, etc.

Conclusion and perspectives

- Status of $(\leq k)$-MP still unclear for trees.
- What about:
  - trees with $\Delta \leq 3$?
  - subdivided combs?
  - etc.
- Dynamic programming yields algorithms.
- What about $(= k)$-MP in trees?
- Appealing case: $k = 3$.
- Other classes of graphs?
- e.g. interval graphs, other sparse classes, etc.

Thank you for your attention!