S. Canzar[1] K. Elbassioni[2] G. W. Klau[1] J. Mestre[3]

Tree-Constrained Matching

[1] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

[2] Max-Planck Institut für Informatik, Saarbrücken, Germany

[3] The University of Sydney, Australia

ICALP'11

Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

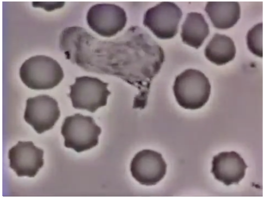
1. Perform image segmentation for each frame

Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

1. Perform image segmentation for each frame

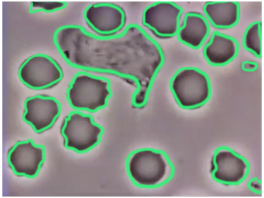


Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

1. Perform image segmentation for each frame

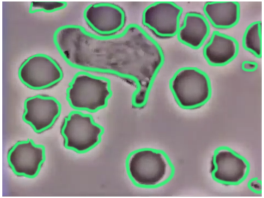


Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

1. Perform image segmentation for each frame
2. Match segments from adjacengt frames

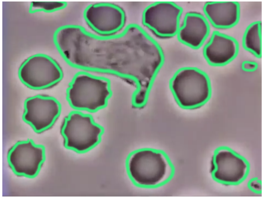


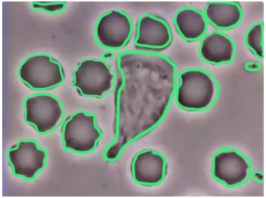
Analysis of Live Cell Video

Given live cell video, we want to track individual cells

Segmentation based methods:

1. Perform image segmentation for each frame
2. Match segments from adjacengt frames



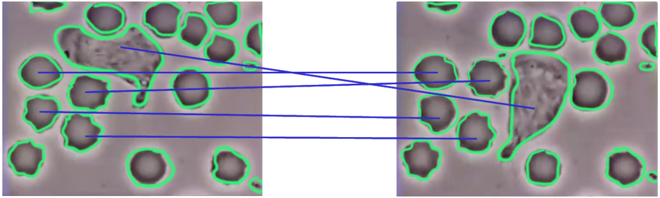


Analysis of Live Cell Video

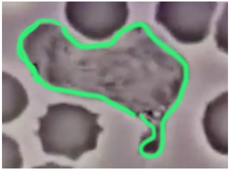
Given live cell video, we want to track individual cells

Segmentation based methods:

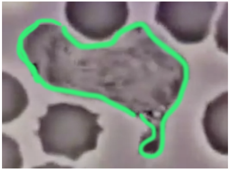
1. Perform image segmentation for each frame
2. Match segments from adjacengt frames

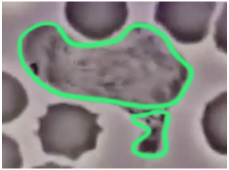


Over/under segmentation

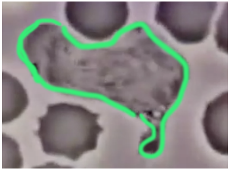


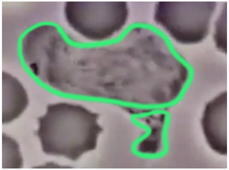
Over/under segmentation

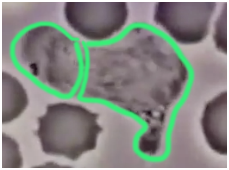




Over/under segmentation







Over/under segmentation

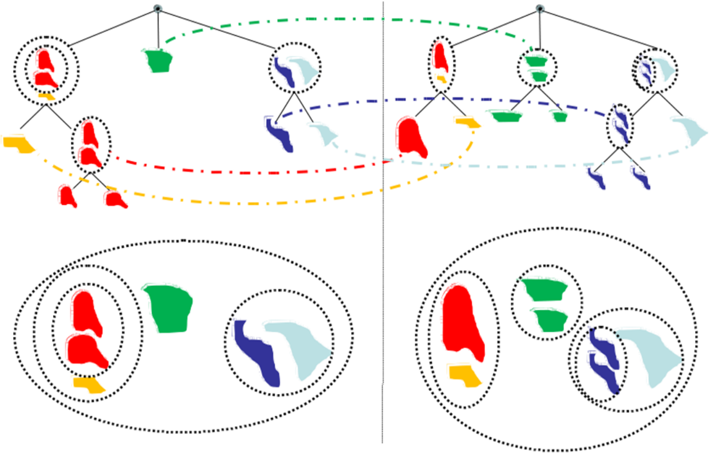
Challenge: Biological cell division vs. over-segmentation

Over/under segmentation

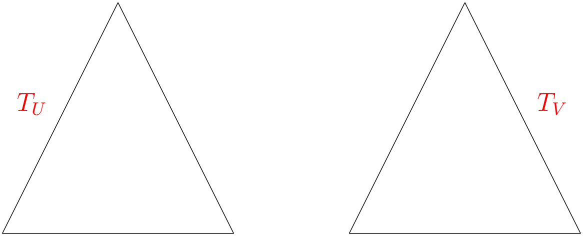
Challenge: Biological cell division vs. over-segmentation

⇒ [Mosig *et al.*, 2009]: Integrate identification and tracking steps!

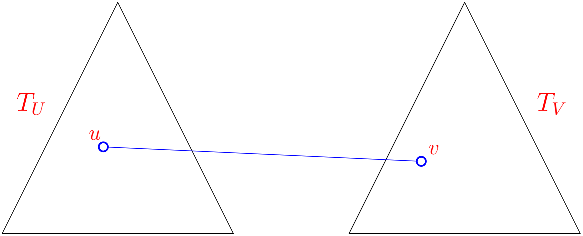
Cosegmentation



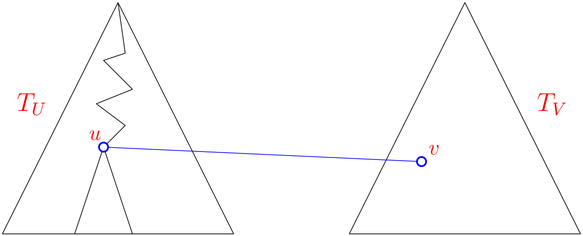
Tree-Constrained Matching



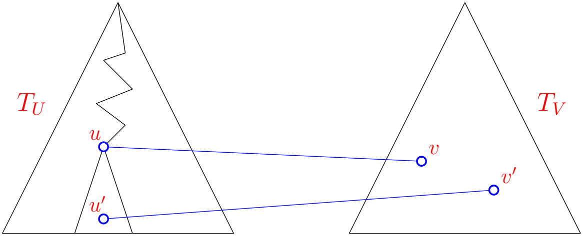
Tree-Constrained Matching



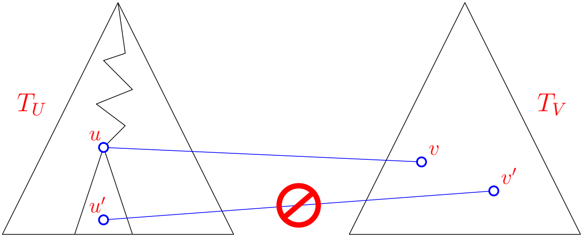
Tree-Constrained Matching



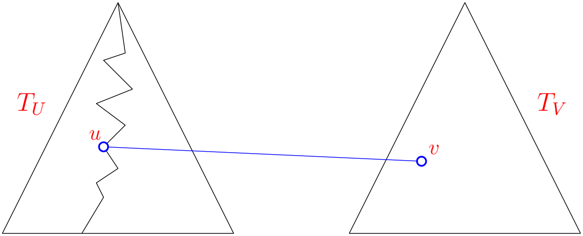
Tree-Constrained Matching



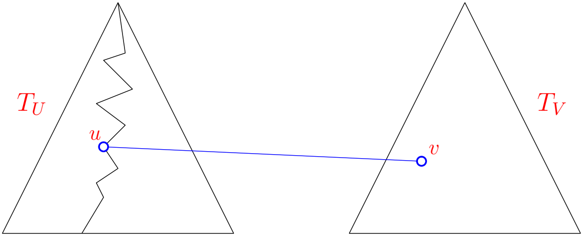
Tree-Constrained Matching



Tree-Constrained Matching

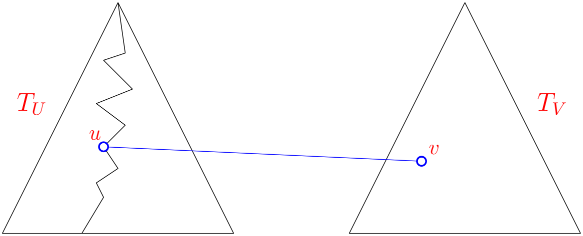


Tree-Constrained Matching



Given a weighted bipartite graph (U, V, E) and trees T_U and T_V over U and V, we want maximum weight matching $\mathcal{M}$ such that matched vertices in T_U and T_V are not comparable

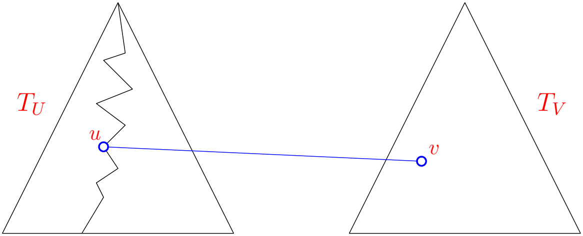
Tree-Constrained Matching



Given a weighted bipartite graph (U, V, E) and trees T_U and T_V over U and V, we want maximum weight matching $\mathcal{M}$ such that matched vertices in T_U and T_V are not comparable

Mosig *et al.* introduced TCM and gave a LP formulation, which they claimed was totally unimodular

Tree-Constrained Matching



Given a weighted bipartite graph (U, V, E) and trees T_U and T_V over U and V, we want maximum weight matching $\mathcal{M}$ such that matched vertices in T_U and T_V are not comparable

Mosig *et al.* introduced TCM and gave a LP formulation, which they claimed was totally unimodular

Unfortunately, as we shall see, this is not the case

MIS in d-Interval Graphs

There is a reduction from TCM to MIS in 2-interval graphs

MIS in d-Interval Graphs

There is a reduction from TCM to MIS in 2-interval graphs

A d-interval is $u = \{[s_1, f_1], [s_2, f_2], \ldots, [s_d, f_d]\}$

MIS in d-Interval Graphs

There is a reduction from TCM to MIS in 2-interval graphs

A d-interval is $u = \{[s_1, f_1], [s_2, f_2], \ldots, [s_d, f_d]\}$

A d-interval graph is the intersection graph of a set of d-intervals

MIS in d-Interval Graphs

There is a reduction from TCM to MIS in 2-interval graphs

A d-interval is $u = \{[s_1, f_1], [s_2, f_2], \ldots, [s_d, f_d]\}$

A d-interval graph is the intersection graph of a set of d-intervals

Maximum weight independent set (MIS) in d-interval graphs:

- If $d = 1$, there is an exact algorithm
- [BHNSS] If $d > 1$, the problem is *APX*-hard, but there is a $2d$-approximation

MIS in d-Interval Graphs

There is a reduction from TCM to MIS in 2-interval graphs

A d-interval is $u = \{[s_1, f_1], [s_2, f_2], \ldots, [s_d, f_d]\}$

A d-interval graph is the intersection graph of a set of d-intervals

Maximum weight independent set (MIS) in d-interval graphs:

- If $d = 1$, there is an exact algorithm
- [BHNSS] If $d > 1$, the problem is *APX*-hard, but there is a $2d$-approximation

Thus, there is a 4-approximation for TCM

Our Results

Show that TCM is APX-hard and disprove claim of Mosig *et al.*

Our Results

Show that TCM is APX-hard and disprove claim of Mosig *et al.*

Bar-Yehuda *et al.* algorithm is in fact a 3-approximation for TCM

Our Results

Show that TCM is APX-hard and disprove claim of Mosig *et al.*

Bar-Yehuda *et al.* algorithm is in fact a 3-approximation for TCM

Give 2-approximation, matching integrality gap of LP formulation

Our Results

Show that TCM is APX-hard and disprove claim of Mosig *et al.*

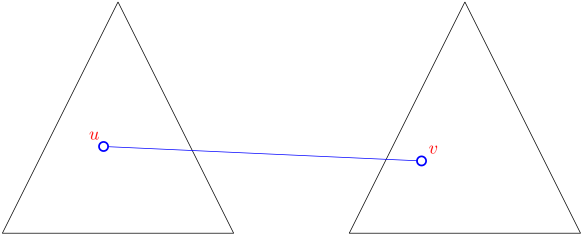
Bar-Yehuda *et al.* algorithm is in fact a 3-approximation for TCM

Give 2-approximation, matching integrality gap of LP formulation

Generalization to posets

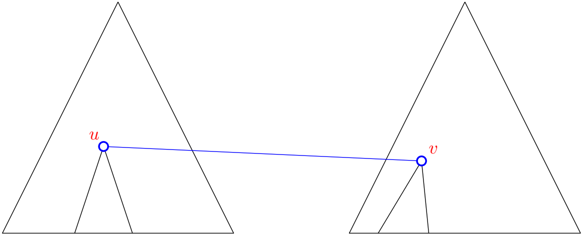
Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V



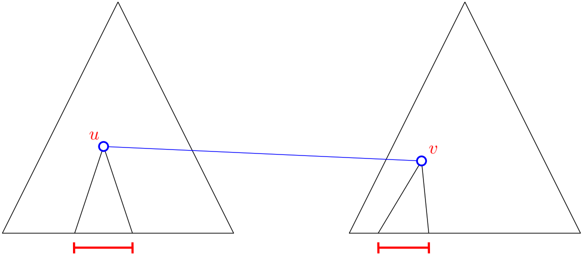
Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V



Reducing TCM to MIS in 2-IG

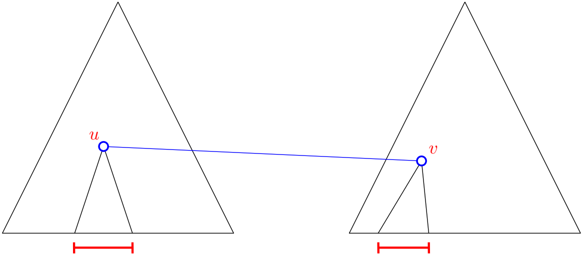
Every edge (u, v) is assigned one interval in T_U and one in T_V



Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V

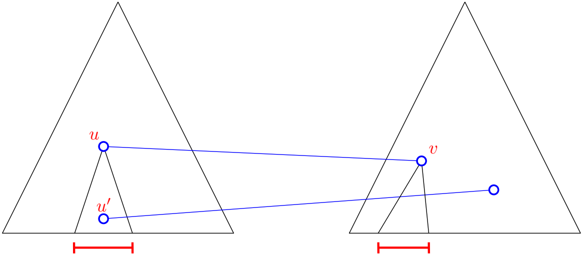
Matching is feasible $\Leftrightarrow$ corresp. set of 2-intervals is independent



Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V

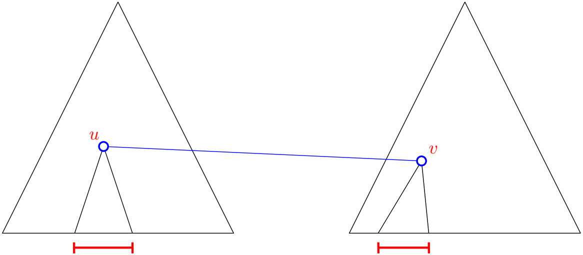
Matching is feasible ⇔ corresp. set of 2-intervals is independent



Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V

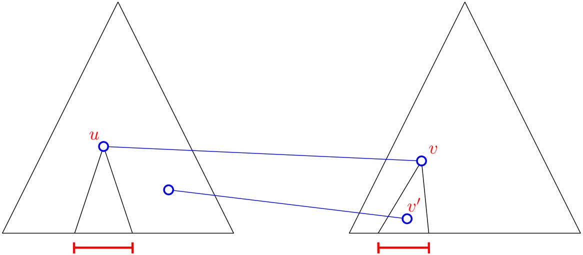
Matching is feasible $\Leftrightarrow$ corresp. set of 2-intervals is independent



Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V

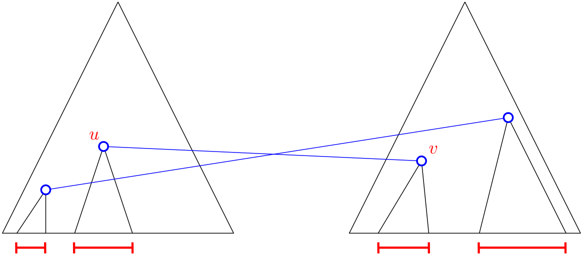
Matching is feasible $\Leftrightarrow$ corresp. set of 2-intervals is independent



Reducing TCM to MIS in 2-IG

Every edge (u, v) is assigned one interval in T_U and one in T_V

Matching is feasible $\Leftrightarrow$ corresp. set of 2-intervals is independent



Bar-Yehuda *et al.* algorithm

$$\max \sum_{u \in V} x_u$$

$$\text{s.t.} \sum_{u: p \in u} x_u \leq 1 \quad \forall \text{ point } p$$

$$x_u \geq 0 \quad \forall d\text{-interval } u$$

Bar-Yehuda *et al.* algorithm

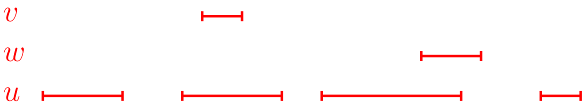
$$\max \sum_{u \in V} x_u$$
$$\text{s.t.} \sum_{u: p \in u} x_u \leq 1 \quad \forall \text{ point } p$$
$$x_u \geq 0 \quad \forall d\text{-interval } u$$

MIS-d-interval(G)

1: let x be optimal LP solution
2: let $\mathcal{S}$ be the empty set
3: **while** G is not empty **do**
4: let u minimize $x(N(u))$
5: add u to $\mathcal{S}$
6: remove $N(u) + u$ from G
7: **return** $\mathcal{S}$

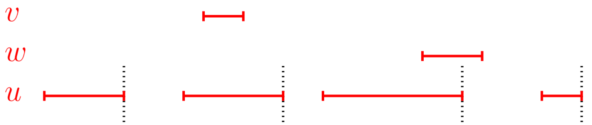
∀ feasible x: $\exists u : x(N(u)) \leq 2d$

$$\sum_{u} x_u \sum_{v \in N(u)} x_v$$



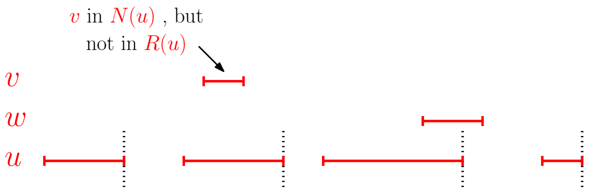
$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

$$\sum_{u} x_u \sum_{v \in N(u)} x_v$$



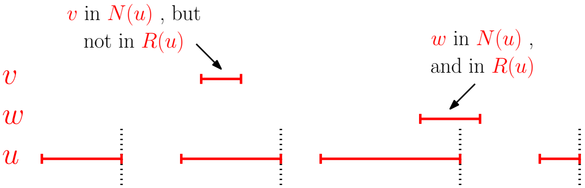
$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

$$\sum_u x_u \sum_{v \in N(u)} x_v$$



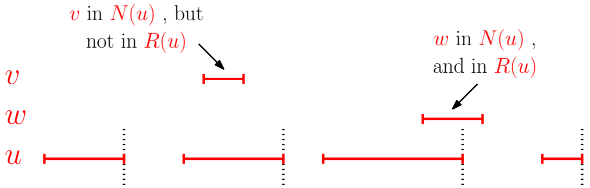
$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

$$\sum_{u} x_u \sum_{v \in N(u)} x_v$$



$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

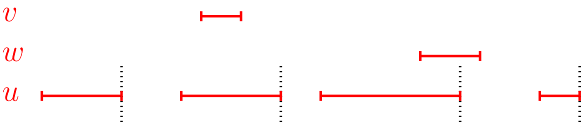
$$\sum_{u} x_u \sum_{v \in N(u)} x_v \leq 2 \sum_{u} x_u \sum_{v \in R(u)} x_v$$



$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

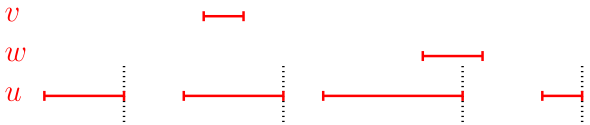
$$\sum_{u} x_u \sum_{v \in N(u)} x_v \leq 2 \sum_{u} x_u \sum_{v \in R(u)} x_v$$

$$\leq 2 \sum_{u} x_u \, d$$



$\forall$ feasible x: $\exists u : x(N(u)) \leq 2d$

$$\sum_u x_u \sum_{v \in N(u)} x_v \leq 2 \sum_u x_u \sum_{v \in R(u)} x_v$$
$$\leq 2 \sum_u x_u \, d$$
$$\leq 2d \sum_u x_u$$



4-approximation for TCM

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{e \text{ on } P} x_e \leq 1 \quad \forall \text{desc. path } P$$

$$x_e \geq 0 \quad \forall \text{ edge } e$$

4-approximation for TCM

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{e \text{ on } P} x_e \leq 1 \quad \forall \text{desc. path } P$$

$$x_e \geq 0 \quad \forall \text{ edge } e$$

TCM(U,V,E)

```
let x be optimal LP solution
let 𝓜 be the empty set
while E is not empty do
    let e minimize x(N(e))
    add e to 𝓜
    remove N(e) + e from E
return 𝓜
```

4-approximation for TCM

$$\max \sum_{e \in E} x_e$$

$$\text{s.t.} \sum_{e \text{ on } P} x_e \leq 1 \quad \forall \text{desc. path } P$$

$$x_e \geq 0 \quad \forall \text{ edge } e$$

$N(e)$ is the set of edges in conflict with e

TCM(U,V,E)

1: let x be optimal LP solution
2: let $\mathcal{M}$ be the empty set
3: **while** E is not empty **do**
4: let e minimize $x(N(e))$
5: add e to $\mathcal{M}$
6: remove $N(e) + e$ from E
7: **return** $\mathcal{M}$

$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

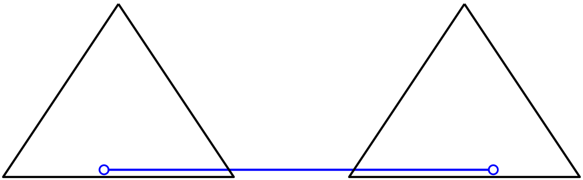
If we have a leaf-to-leaf edge, we are done

∀ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

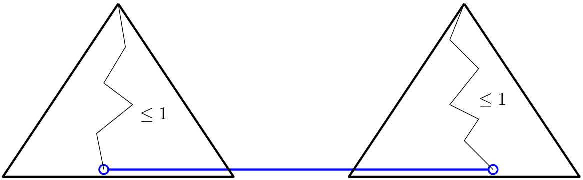


$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done



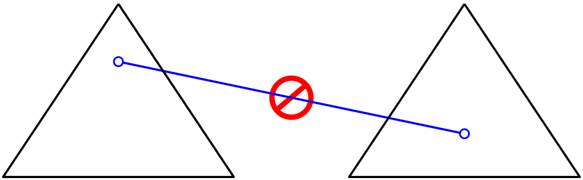
∀ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

Otherwise, no internal-to-internal edges



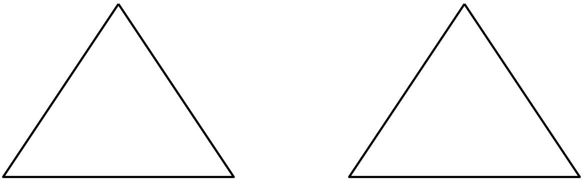
∀ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

Otherwise, no internal-to-internal edges



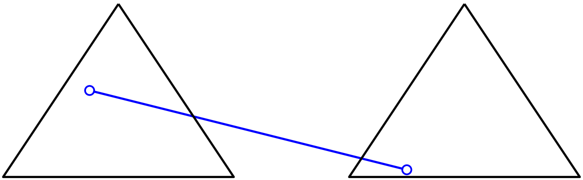
$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

Otherwise, no internal-to-internal edges



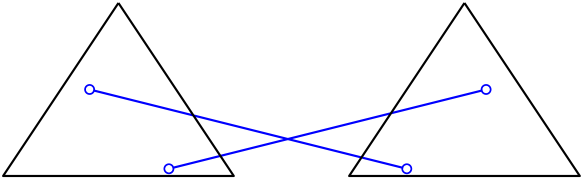
$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

Otherwise, no internal-to-internal edges



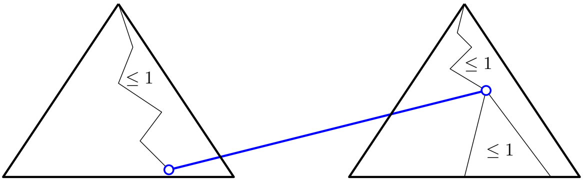
$\forall$ basic feasible x: $\exists e : x(N(e)) \leq 3$

For ease of analysis, assume that

- For all edges $x_e > 0$
- No leaf is unmatched

If we have a leaf-to-leaf edge, we are done

Otherwise, no internal-to-internal edges



Some comments

We don't need to find bfs at each step

Some comments

We don't need to find bfs at each step

We can handle weighted MIS by using fractional local ratio (FLR)

Some comments

We don't need to find bfs at each step

We can handle weighted MIS by using fractional local ratio (FLR)

There are instances where every edge has FLR $3 - o(1)$

Some comments

We don't need to find bfs at each step

We can handle weighted MIS by using fractional local ratio (FLR)

There are instances where every edge has FLR $3 - o(1)$

But we can get a 2-approximation with one more idea

Some comments

We don't need to find bfs at each step

We can handle weighted MIS by using fractional local ratio (FLR)

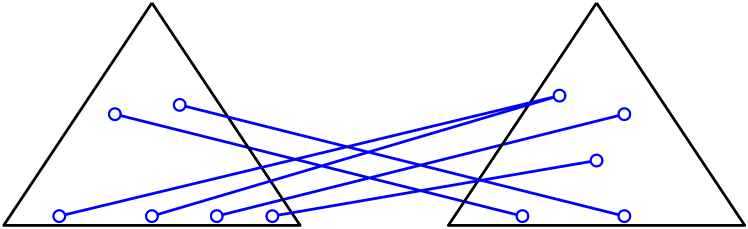
There are instances where every edge has FLR $3 - o(1)$

But we can get a 2-approximation with one more idea

Integrality gap of LP formulation is $2 - o(1)$

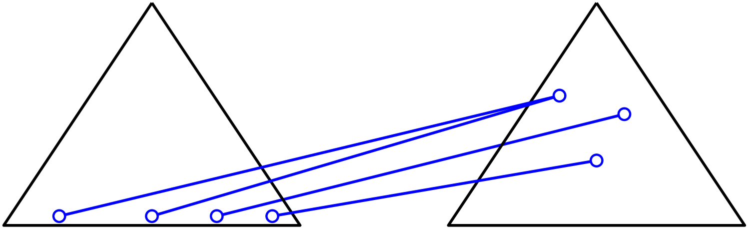
2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



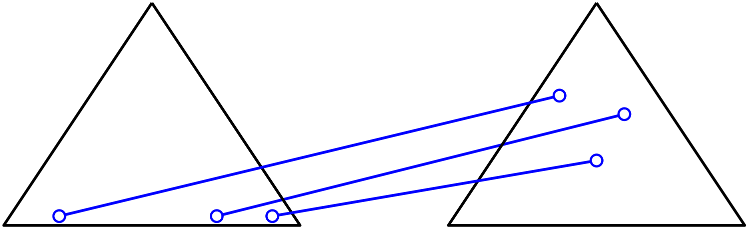
2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



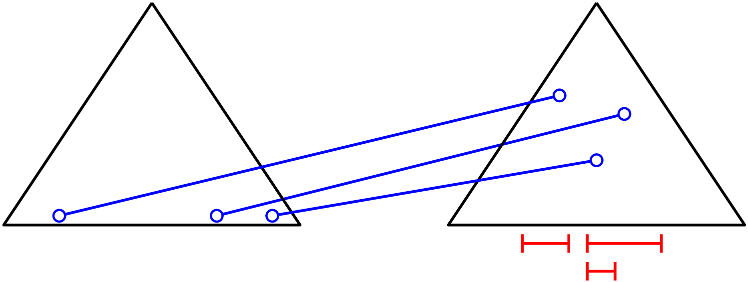
2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



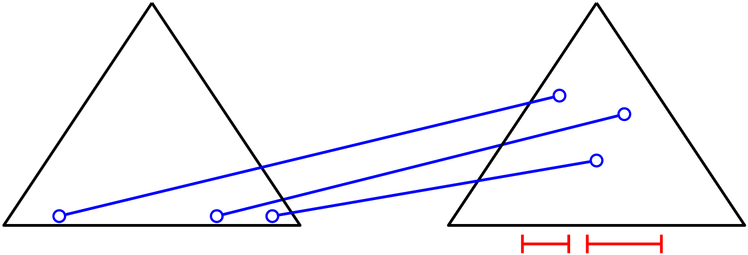
2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



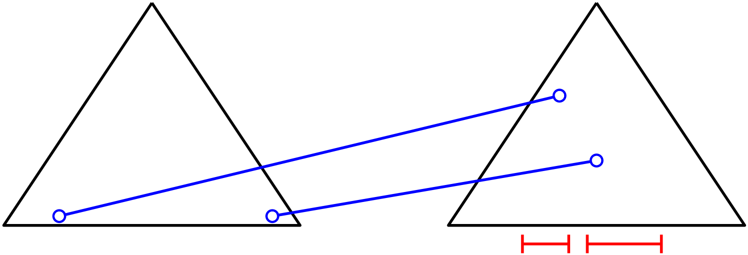
2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



2-approximation

Idea: Exploit bfs structure if $\forall e : x(N(e)) > 2$



Generalization to posets

To model uncertainty in hierarchical clustering, we can use posets instead of trees

Generalization to posets

To model uncertainty in hierarchical clustering, we can use posets instead of trees

We want matched vertices to be incomparable

Generalization to posets

To model uncertainty in hierarchical clustering, we can use posets instead of trees

We want matched vertices to be incomparable

Give 4ρ-approximation, where ρ is a parameter of poset

Generalization to posets

To model uncertainty in hierarchical clustering, we can use posets instead of trees

We want matched vertices to be incomparable

Give 4ρ-approximation, where ρ is a parameter of poset

It cannot be approximated to $2^{\log^{1-\epsilon} \rho}$, for any $\epsilon > 0$, unless $NP \subseteq DTIME(n^{\text{polylog}\, n})$

Thank you for your attention!