A Parameterized Perspective on Uniquely Restricted Matchings

Juhi Chaudhary
Indian Institute of Petroleum and Energy, Visakhapatnam, India

Ignasi Sau
LIRMM, Université de Montpellier, CNRS, Montpellier, France

Meirav Zehavi
Computer Science Department, Ben-Gurion University, Israel

LAGOS 2025, Buenos Aires, Argentina

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

M-saturated vertices

Endpoints of the edges of a matching M.

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

M-saturated vertices

Endpoints of the edges of a matching M. Let us denote them by V_M.

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

M-saturated vertices

Endpoints of the edges of a matching M. Let us denote them by V_M.

In a general matching M, the graph $G[V_M]$ does not have any restriction.

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

M-saturated vertices

Endpoints of the edges of a matching M. Let us denote them by V_M.

In a general matching M, the graph $G[V_M]$ does not have any restriction.

What if we want $G[V_M]$ to satisfy some special property $\mathcal{P}$?

Definitions

Matching in a graph G

A subset of edges $M \subseteq E(G)$ that do not share an endpoint.

M-saturated vertices

Endpoints of the edges of a matching M. Let us denote them by V_M.

In a general matching M, the graph $G[V_M]$ does not have any restriction.

What if we want $G[V_M]$ to satisfy some special property $\mathcal{P}$?

The corresponding matchings are called $\mathcal{P}$-matchings.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\}$

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\}$

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\}$

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\} \rightarrow$ (dis)connected matching.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\} \rightarrow$ (dis)connected matching.
- if $\mathcal{P} = \{\text{being a matching}\}$

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\} \rightarrow$ (dis)connected matching.
- if $\mathcal{P} = \{\text{being a matching}\} \rightarrow$ induced matching.

Examples of $\mathcal{P}$-matchings

$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\} \rightarrow$ (dis)connected matching.
- if $\mathcal{P} = \{\text{being a matching}\} \rightarrow$ induced matching.
- if $\mathcal{P} = \{\text{having exactly one perfect matching}\}$

Examples of $\mathcal{P}$-matchings

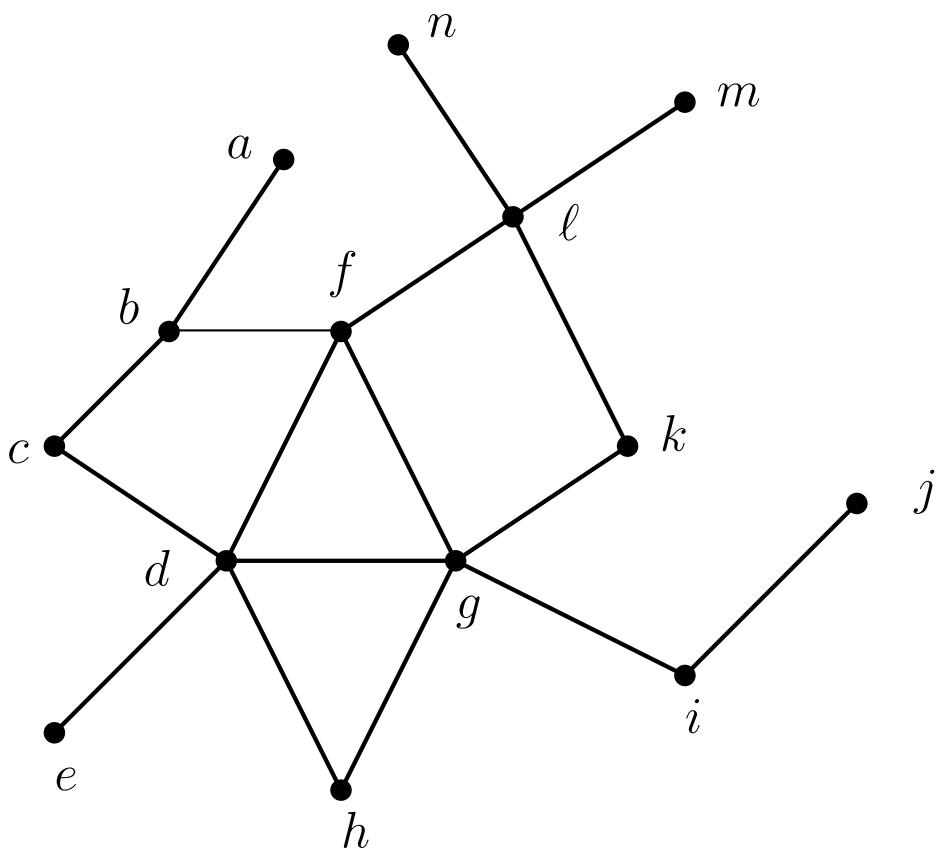
$\mathcal{P}$-matching

A matching M in a graph G is a $\mathcal{P}$-matching if $G[V_M]$ satisfies property $\mathcal{P}$.

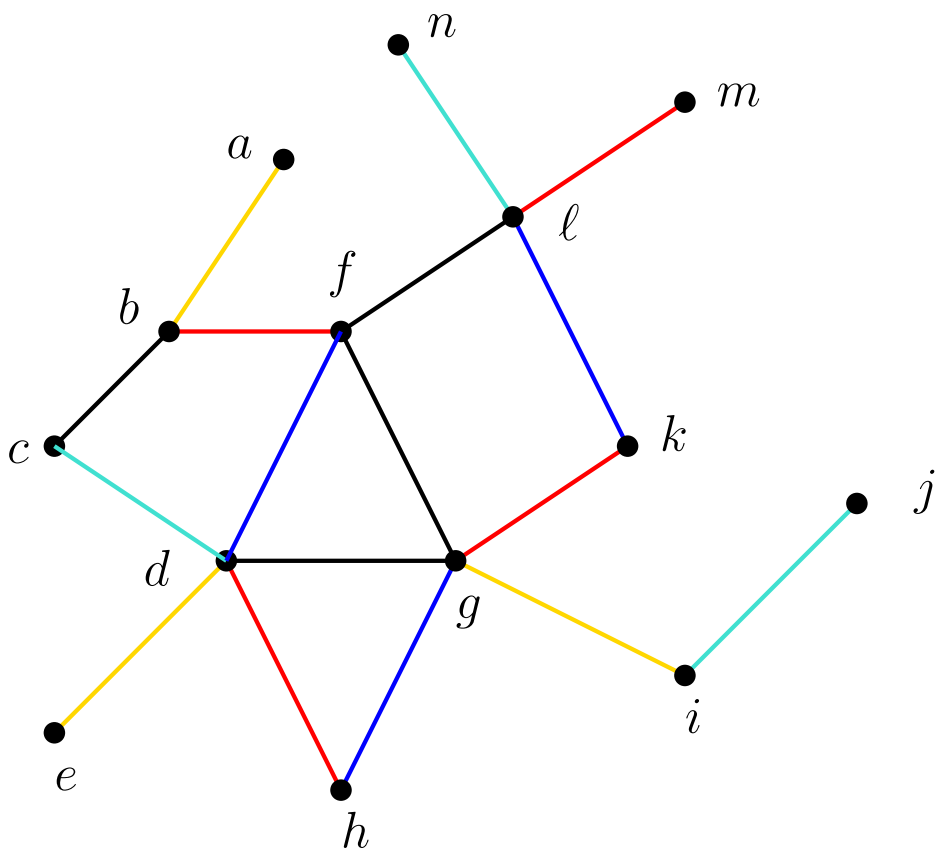
Popular examples

- if $\mathcal{P} = \{\text{being any graph}\} \rightarrow$ (unrestricted) matching.
- if $\mathcal{P} = \{\text{being a forest}\} \rightarrow$ acyclic matching.
- if $\mathcal{P} = \{\text{being (dis)connected}\} \rightarrow$ (dis)connected matching.
- if $\mathcal{P} = \{\text{being a matching}\} \rightarrow$ induced matching.
- if $\mathcal{P} = \{\text{having exactly one perfect matching}\}$
  $\rightarrow$ uniquely restricted matching (URM).

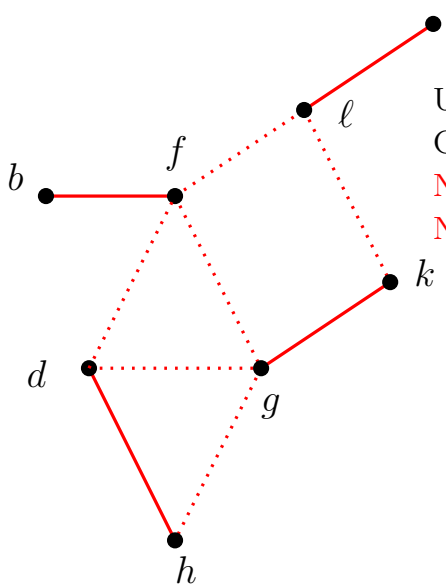
Example



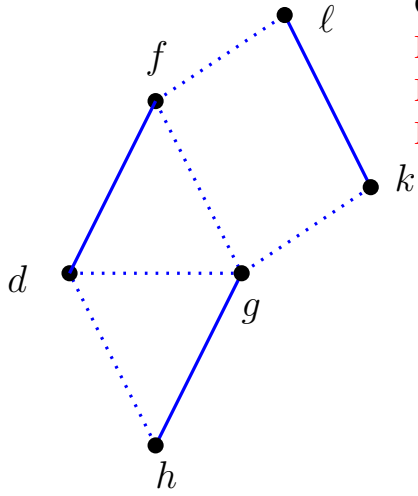
Example



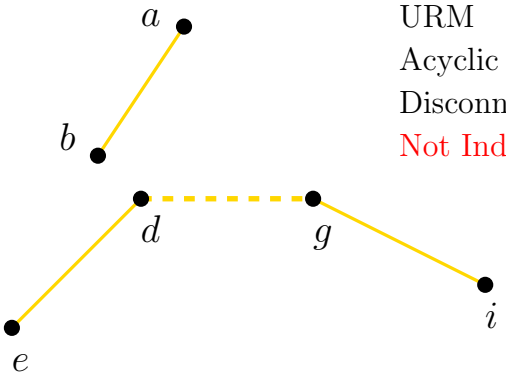
Example



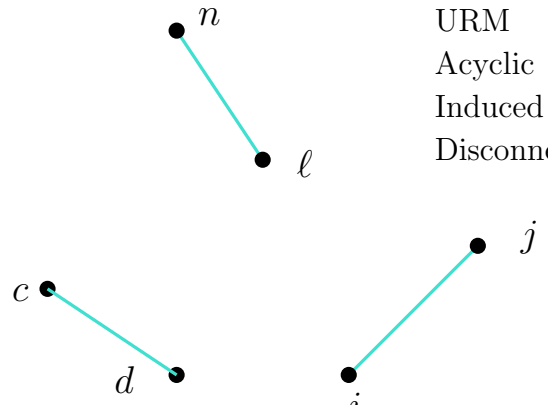
URM
Connected
Not Acyclic
Not Induced



Connected
Not Acyclic
Not Induced
Not URM



URM
Acyclic
Disconnected
Not Induced



URM
Acyclic
Induced
Disconnected

This article: uniquely restricted matchings (URMs)

This article: uniquely restricted matchings (URMs)

URMs have motivation from linear algebra. [Golumbic, Hirst, Lewenstein. 2001]

This article: uniquely restricted matchings (URMs)

URMs have motivation from linear algebra. [Golumbic, Hirst, Lewenstein. 2001]

Uniquely Restricted Matching (URM)
Input: An undirected graph G and a positive integer ℓ.
Question: Does G have a URM of size at least ℓ?

This article: uniquely restricted matchings (URMs)

URMs have motivation from linear algebra. [Golumbic, Hirst, Lewenstein. 2001]

> Uniquely Restricted Matching (URM)
> **Input:** An undirected graph G and a positive integer ℓ.
> **Question:** Does G have a URM of size at least ℓ?

Problem studied in a number of articles, in particular:

Graph class	**Results**	**authors**
bipartite, split threshold, cacti, block	NP-hard linear-time solvable	[Golumbic, Hirst, Lewenstein. 2001]
Bipartite of degree ≤ 3	APX-complete	[Mishra. 2011]
interval, proper int., bipartite perm.	poly time linear time	[Francis, Jacob, Jan. 2018]

This article: uniquely restricted matchings (URMs)

URMs have motivation from linear algebra. [Golumbic, Hirst, Lewenstein. 2001]

> Uniquely Restricted Matching (URM)
> **Input:** An undirected graph G and a positive integer ℓ.
> **Question:** Does G have a URM of size at least ℓ?

Problem studied in a number of articles, in particular:

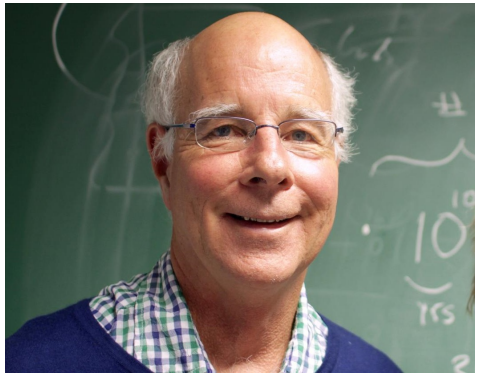
Graph class	**Results**	**authors**
bipartite, split threshold, cacti, block	NP-hard linear-time solvable	[Golumbic, Hirst, Lewenstein. 2001]
Bipartite of degree ≤ 3	APX-complete	[Mishra. 2011]
interval, proper int., bipartite perm.	poly time linear time	[Francis, Jacob, Jan. 2018]

In this article we study the parameterized complexity of the Uniquely Restricted Matching problem.

Parameterized complexity in a nutshell

Idea Measure the complexity of an algorithm in terms of the input size and an additional parameter.

This theory started in the late 80's, by Downey and Fellows:



Today, it is a well-established and very active area.

Parameterized problems

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

Parameterized problems

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$,
where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

A parameterized problem is fixed-parameter tractable (FPT) if it can be solved in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f.

Parameterized problems

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$,
where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

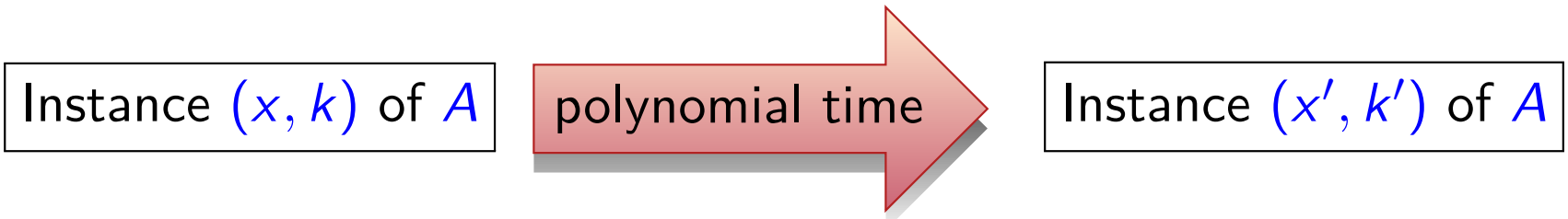
A parameterized problem is fixed-parameter tractable (FPT) if it can be solved in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f.

W[1]-hardness: strong evidence of not being FPT.

Kernelization

Idea polynomial-time preprocessing.

A kernel for a parameterized problem A is an algorithm such that:



1. (x, k) is a YES-instance of $A \Leftrightarrow (x', k')$ is a YES-instance of A.
2. $|x'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$.

The function g is called the size of the kernel.

If g is a polynomial (linear), then we have a polynomial (linear) kernel.

Kernelization

Idea polynomial-time preprocessing.

A kernel for a parameterized problem A is an algorithm such that:

1. (x, k) is a YES-instance of $A \Leftrightarrow (x', k')$ is a YES-instance of A.
2. $|x'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$.

The function g is called the size of the kernel.

If g is a polynomial (linear), then we have a polynomial (linear) kernel.

Fact: A problem is FPT $\Leftrightarrow$ it admits a kernel

Do all FPT problems admit polynomial kernels?

Fact: A problem is FPT $\Leftrightarrow$ it admits a kernel

Do all FPT problems admit polynomial kernels?

Do all FPT problems admit polynomial kernels?

Fact: A problem is FPT $\Leftrightarrow$ it admits a kernel

Do all FPT problems admit polynomial kernels? NO!

Theorem (Bodlaender, Downey, Fellows, Hermelin. 2009)

Deciding whether a graph has a PATH *with* $\geq k$ *vertices is* FPT *but does not admit a polynomial kernel, unless* NP $\subseteq$ coNP/poly.

Do all FPT problems admit polynomial kernels?

Fact: A problem is FPT $\Leftrightarrow$ it admits a kernel

Do all FPT problems admit polynomial kernels? NO!

Theorem (Bodlaender, Downey, Fellows, Hermelin. 2009)

Deciding whether a graph has a PATH *with* $\geq k$ *vertices is* FPT *but does not admit a polynomial kernel, unless* NP $\subseteq$ coNP/poly.

Major goal in parameterized complexity:

Which FPT problems admit polynomial kernels?

Param. comp. of Uniquely Restricted Matching

Only a few recent results about the parameterized complexity of Uniquely Restricted Matching: [Chaudhary, Zehavi. 2025]

Param. comp. of Uniquely Restricted Matching

Only a few recent results about the parameterized complexity of Uniquely Restricted Matching: [Chaudhary, Zehavi. 2025]

- The problem cannot be approximated in FPT time within any constant factor.

Param. comp. of Uniquely Restricted Matching

Only a few recent results about the parameterized complexity of Uniquely Restricted Matching: [Chaudhary, Zehavi. 2025]

- The problem cannot be approximated in FPT time within any constant factor. In particular, it is W[1]-hard by the solution size.

Param. comp. of Uniquely Restricted Matching

Only a few recent results about the parameterized complexity of Uniquely Restricted Matching: [Chaudhary, Zehavi. 2025]

- The problem cannot be approximated in FPT time within any constant factor. In particular, it is W[1]-hard by the solution size.
- It admits a linear kernel on planar graphs by the solution size.

Param. comp. of Uniquely Restricted Matching

Only a few recent results about the parameterized complexity of Uniquely Restricted Matching: [Chaudhary, Zehavi. 2025]

- The problem cannot be approximated in FPT time within any constant factor. In particular, it is W[1]-hard by the solution size.
- It admits a linear kernel on planar graphs by the solution size.

Our goal: deeper analysis of the parameterized complexity of URM.

Our results and sketch of techniques (part 1/3)

The problem is W[1]-hard by the solution size ℓ on general graphs.

[Chaudhary, Zehavi. 2025]

Can we identify relevant classes of graphs where the problem is FPT?

Our results and sketch of techniques (part 1/3)

The problem is W[1]-hard by the solution size ℓ on general graphs.

[Chaudhary, Zehavi. 2025]

Can we identify relevant classes of graphs where the problem is FPT?

Theorem 1

For every line graph G, the URM *problem can be solved in time $2^{\mathcal{O}(\ell)} \cdot |V(G)|$ when parameterized by the solution size ℓ.*

Our results and sketch of techniques (part 1/3)

Theorem 1

For every line graph G, the URM *problem can be solved in time $2^{\mathcal{O}(\ell)} \cdot |V(G)|$ when parameterized by the solution size ℓ.*

Our results and sketch of techniques (part 1/3)

Theorem 1

For every line graph G, the URM *problem can be solved in time $2^{\mathcal{O}(\ell)} \cdot |V(G)|$ when parameterized by the solution size ℓ.*

This is an easy consequence of the following lemma:

Lemma

Let H be a graph and $G = L(H)$ be the line graph of H. Then, G has a URM of size ℓ ***if and only if*** *H contains ℓ edge-disjoint paths $W_1, \ldots, W_\ell$, each with 2 edges, such that $\bigcup_{i \in [\ell]} W_i$ is a forest, and no two distinct paths W_i and W_j (for $i \neq j$) together form a $K_{1,4}$.*

Our results and sketch of techniques (part 1/3)

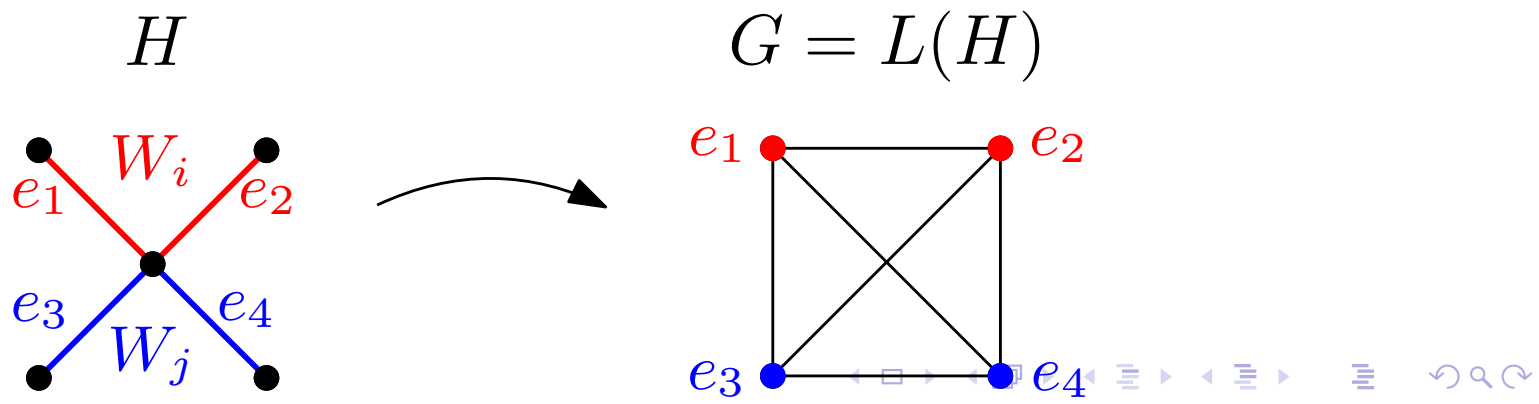
Theorem 1

For every line graph G, the URM *problem can be solved in time $2^{\mathcal{O}(\ell)} \cdot |V(G)|$ when parameterized by the solution size ℓ.*

This is an easy consequence of the following lemma:

Lemma

Let H be a graph and $G = L(H)$ be the line graph of H. Then, G has a URM of size ℓ ***if and only if*** *H contains ℓ edge-disjoint paths $W_1, \ldots, W_\ell$, each with 2 edges, such that $\bigcup_{i \in [\ell]} W_i$ is a forest, and no two distinct paths W_i and W_j (for $i \neq j$) together form a $K_{1,4}$.*



Our results and sketch of techniques (part 2/3)

One of the most successful structural parameters is treewidth (noted tw).

Our results and sketch of techniques (part 2/3)

One of the most successful structural parameters is treewidth (noted tw).

FPT algorithms in time $2^{\mathcal{O}(\mathsf{tw})} \cdot |V(G)|$ were obtained for Induced Matching, Acyclic Matching, and c-Disconnected Matching.

[Chaudhary, Zehavi. 2023]

[Lampis, Vasilakis. 2025]

Our results and sketch of techniques (part 2/3)

One of the most successful structural parameters is treewidth (noted tw).

FPT algorithms in time $2^{\mathcal{O}(\mathsf{tw})} \cdot |V(G)|$ were obtained for Induced Matching, Acyclic Matching, and c-Disconnected Matching.

[Chaudhary, Zehavi. 2023]

[Lampis, Vasilakis. 2025]

Theorem 2

Uniquely Restricted Matching can be solved in $\mathcal{O}(2^{\mathsf{tw}^2/2} \cdot |V(G)|)$ time when parameterized by the treewidth tw *of the input graph G.*

Our results and sketch of techniques (part 2/3)

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathsf{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

Our results and sketch of techniques (part 2/3)

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathrm{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

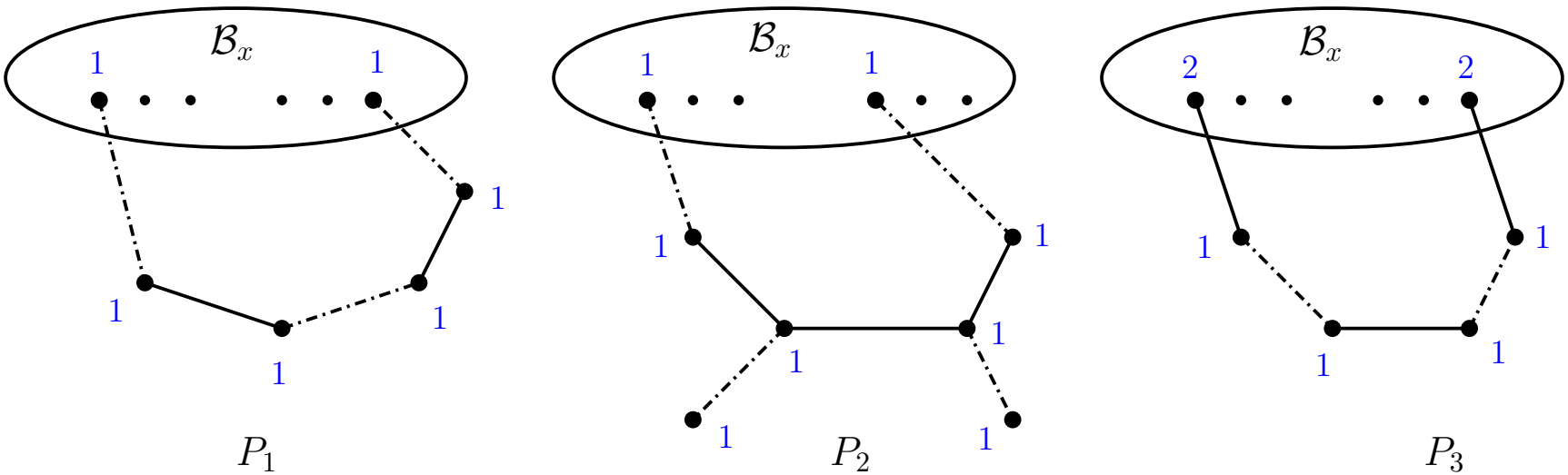
A matching M in a graph G is uniquely restricted **if and only if** there is no alternating cycle with respect to M in G. [Golumbic, Hirst, Lewenstein. 2001]

Our results and sketch of techniques (part 2/3)

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathsf{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

A matching M in a graph G is uniquely restricted **if and only if** there is no alternating cycle with respect to M in G. [Golumbic, Hirst, Lewenstein. 2001]



Our results and sketch of techniques (part 3/3)

Another popular (strong) parameter: vertex cover number.

Our results and sketch of techniques (part 3/3)

Another popular (strong) parameter: vertex cover number.

Acyclic Matching does not admit a polynomial kernel when parameterized by the vertex cover number plus the size of the matching unless NP $\subseteq$ coNP/poly. [Chaudhary, Zehavi. 2025]

Our results and sketch of techniques (part 3/3)

Another popular (strong) parameter: vertex cover number.

Acyclic Matching does not admit a polynomial kernel when parameterized by the vertex cover number plus the size of the matching unless NP $\subseteq$ coNP/poly. [Chaudhary, Zehavi. 2025]

Theorem 3

Uniquely Restricted Matching does not admit a polynomial kernel when parameterized by the vertex cover number plus the size of the matching unless NP $\subseteq$ coNP/poly.

OR-cross-compositions

Useful technique for kernel lower bounds: [Bodlaender, Jansen, Kratsch. 2011]

OR-cross-compositions

Useful technique for kernel lower bounds: [Bodlaender, Jansen, Kratsch. 2011]

Let $L \subseteq \Sigma^*$ be a problem and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem.

OR-cross-compositions

Useful technique for kernel lower bounds: [Bodlaender, Jansen, Kratsch. 2011]

Let $L \subseteq \Sigma^*$ be a problem and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L OR-cross-composes into Q if there is an algorithm which, given t instances $x_1, x_2, \ldots, x_t$ of L, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ of Q in time polynomial in $\Sigma_{i=1}^{t} |x_i|$ such that:

OR-cross-compositions

Useful technique for kernel lower bounds: [Bodlaender, Jansen, Kratsch. 2011]

Let $L \subseteq \Sigma^*$ be a problem and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L OR-cross-composes into Q if there is an algorithm which, given t instances $x_1, x_2, \ldots, x_t$ of L, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ of Q in time polynomial in $\Sigma_{i=1}^{t} |x_i|$ such that:

1. $(x^*, k^*) \in Q \quad \Leftrightarrow \quad x_i \in L$ for some $i \in [t]$,
2. $k^\star$ is bounded by a polynomial in $\max_{i=1}^{t} |x_i| + \log t$.

OR-cross-compositions

Useful technique for kernel lower bounds: [Bodlaender, Jansen, Kratsch. 2011]

Let $L \subseteq \Sigma^*$ be a problem and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L OR-cross-composes into Q if there is an algorithm which, given t instances $x_1, x_2, \ldots, x_t$ of L, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ of Q in time polynomial in $\Sigma_{i=1}^{t} |x_i|$ such that:

1. $(x^*, k^*) \in Q \iff x_i \in L$ for some $i \in [t]$,
2. k^* is bounded by a polynomial in $\max_{i=1}^{t} |x_i| + \log t$.

Theorem (Bodlaender, Jansen, Kratsch. 2011)

If some problem L is NP-hard (under Karp reduction) and there exists an OR-cross-composition from L into some parameterized problem Q, then there is no polynomial kernel for Q unless $\mathrm{NP} \subseteq \mathrm{coNP/poly}$.

Reduction: OR-cross-composition from Exact-3-Cover (NP-hard):

Exact-3-Cover:

Instance: A set $\mathcal{U}$ with $|\mathcal{U}| = 3c$, where $c \in \mathbb{N}$, and a collection $\mathcal{X}$ of 3-element subsets of $\mathcal{U}$.

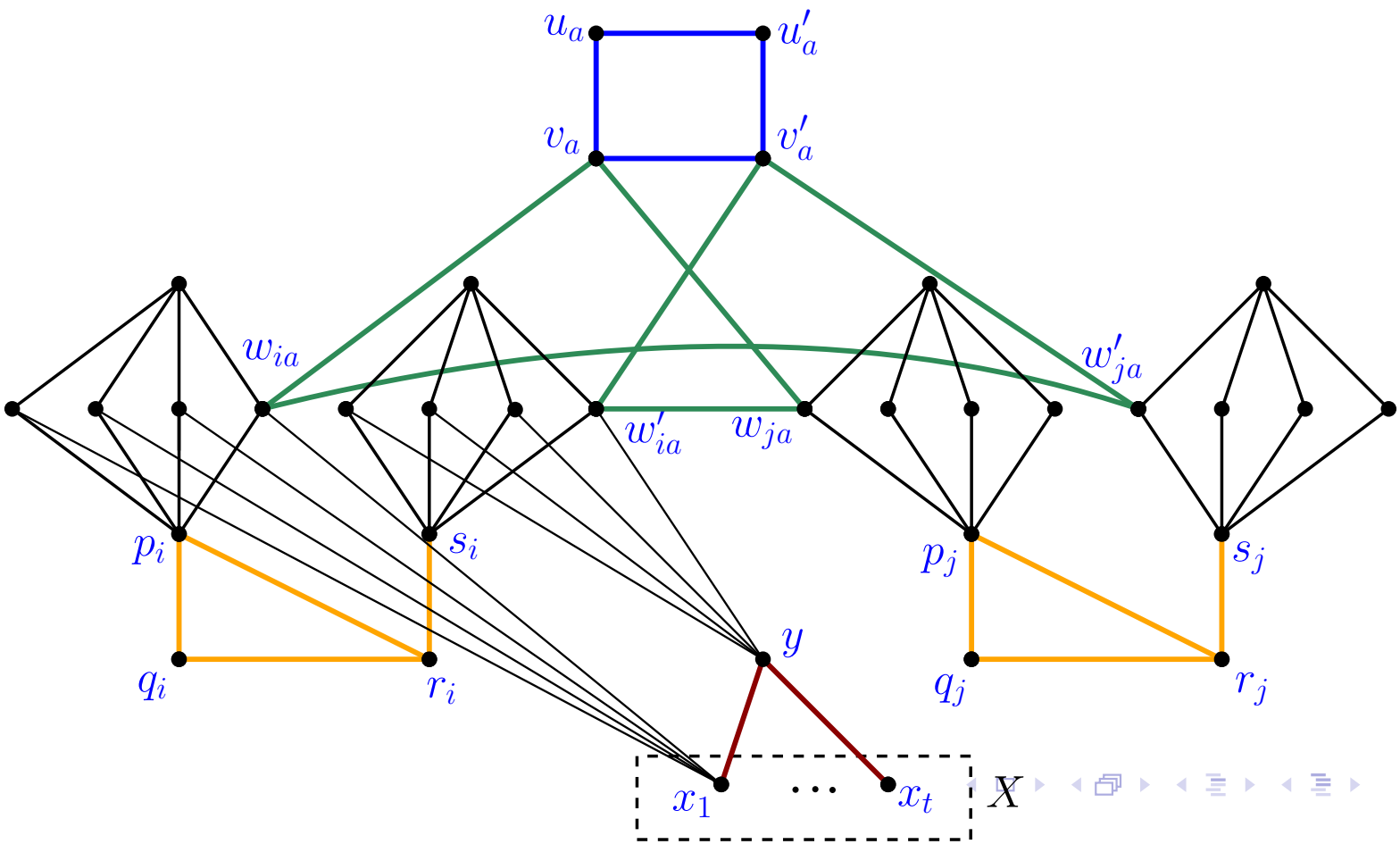
Question: Does there exist a subcollection $\mathcal{X}' \subseteq \mathcal{X}$ such that every element of $\mathcal{U}$ appears in exactly one member of $\mathcal{X}'$?

Reduction: OR-cross-composition from Exact-3-Cover (NP-hard):

Exact-3-Cover:

Instance: A set $\mathcal{U}$ with $|\mathcal{U}| = 3c$, where $c \in \mathbb{N}$, and a collection $\mathcal{X}$ of 3-element subsets of $\mathcal{U}$.

Question: Does there exist a subcollection $\mathcal{X}' \subseteq \mathcal{X}$ such that every element of $\mathcal{U}$ appears in exactly one member of $\mathcal{X}'$?



Further research

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathrm{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

1. Can we get running time $2^{\mathcal{O}(\mathrm{tw})} \cdot |V(G)|$?

Further research

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathrm{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

1 Can we get running time $2^{\mathcal{O}(\mathrm{tw})} \cdot |V(G)|$?

2 Investigate cliquewidth as the parameter, recently studied for Induced Matching and Acyclic Matching. [Lampis, Vasilakis. 2025]

Further research

Theorem 2

Uniquely Restricted Matching *can be solved in* $\mathcal{O}(2^{\mathrm{tw}^2/2} \cdot |V(G)|)$ *time when parameterized by the treewidth* tw *of the input graph* G.

1. Can we get running time $2^{\mathcal{O}(\mathrm{tw})} \cdot |V(G)|$?

2. Investigate cliquewidth as the parameter, recently studied for Induced Matching and Acyclic Matching. [Lampis, Vasilakis. 2025]

3. Study below-guarantee parameters of the form $\mathrm{UB} - \ell$, where UB is an upper bound on the uniquely restricted matching number. Explored for Induced Matching and Acyclic Matching.

[Koana. 2023]

[Chaudhary, Zehavi. 2025]

Gràcies!