Distinction and Detection Problems in Graphs

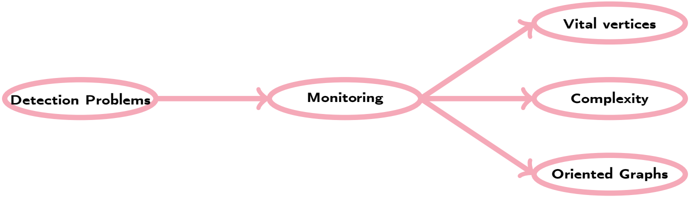
Clara Marcille

Jury:

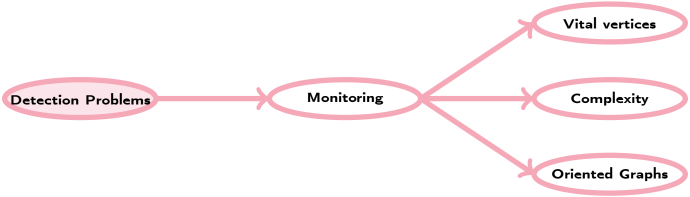
Éric Duchêne (Université Lyon 1)........................reviewer
Nicolas Nisse (Inria d'Université Côte d'Azur).........reviewer
Florent Foucaud (Université Clermont Auvergne)...examiner
Aline Parreau (Université Lyon 1)........................examiner
Cléophée Robin (Université Paris Cité)..................examiner
Éric Sopena (Université de Bordeaux)....................examiner
Julien Bensmail (Université Côte d'Azur).............supervisor
Hervé Hocquard (Université de Bordeaux)............supervisor

PhD. Defence, 24 June 2025

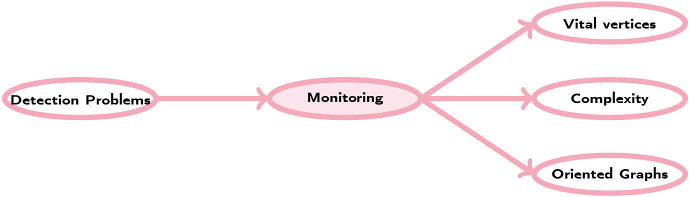
Detection Problems in Graphs



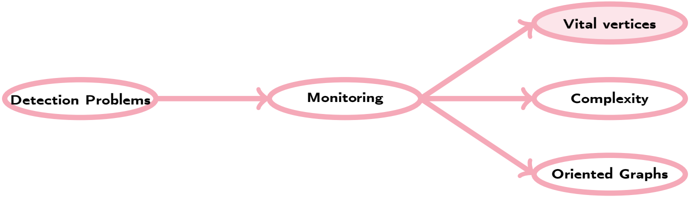
Detection Problems in Graphs



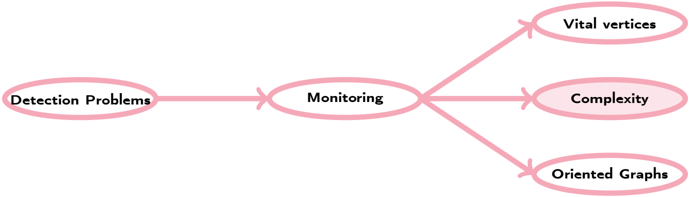
Detection Problems in Graphs



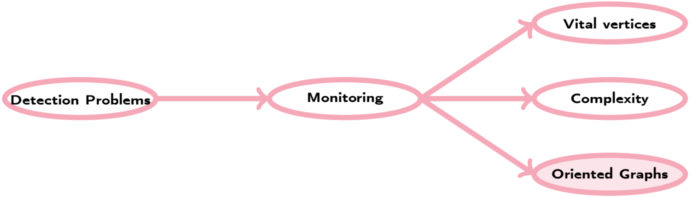
Detection Problems in Graphs



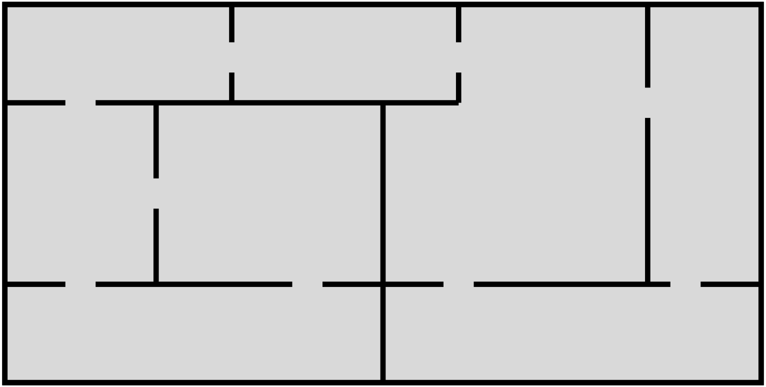
Detection Problems in Graphs



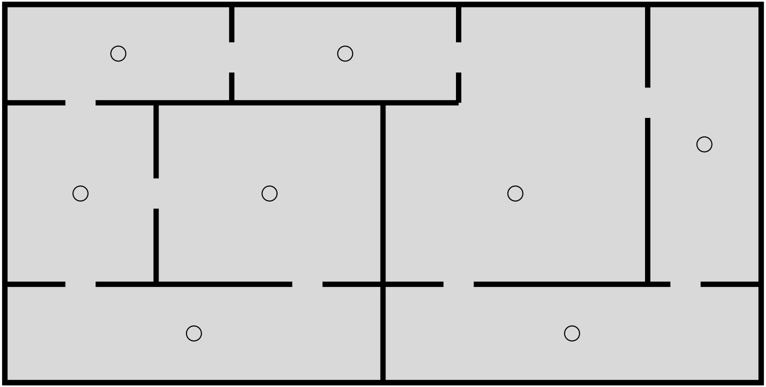
Detection Problems in Graphs



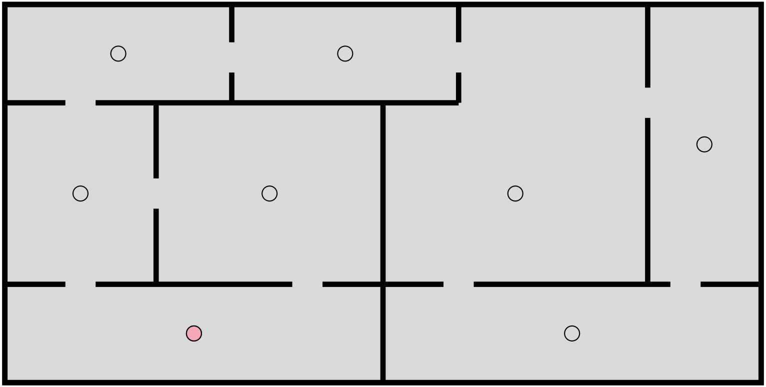
Detecting Events in Graphs



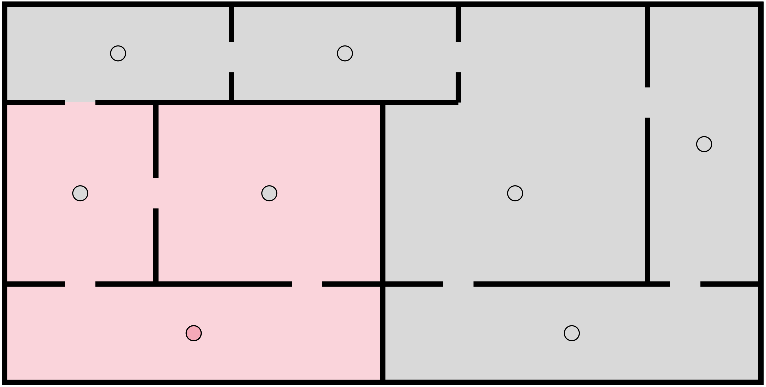
Detecting Events in Graphs



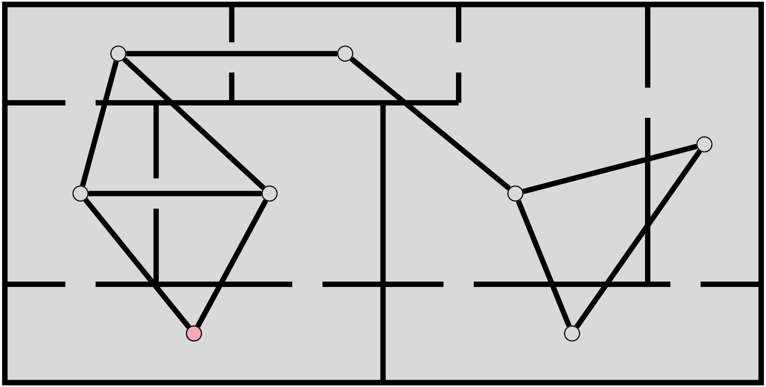
Detecting Events in Graphs



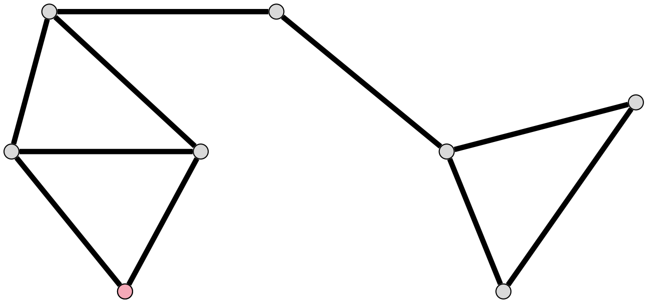
Detecting Events in Graphs



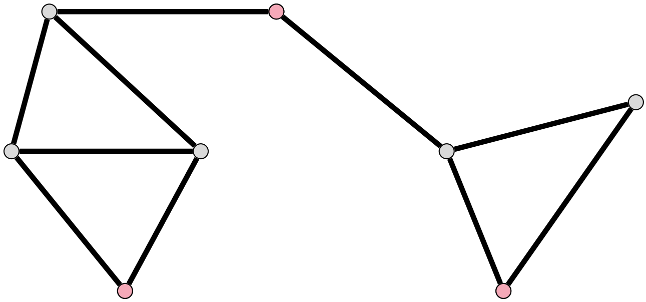
Detecting Events in Graphs



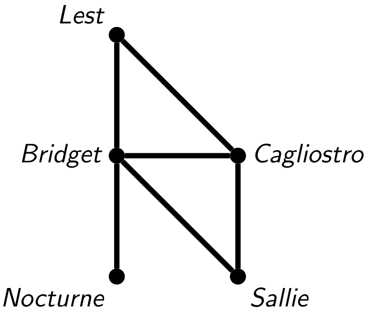
Detecting Events in Graphs



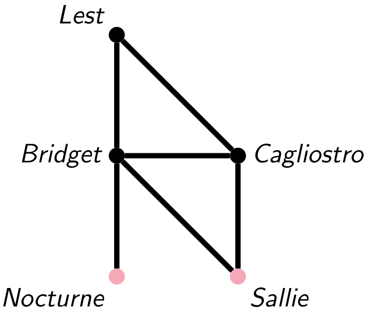
Detecting Events in Graphs



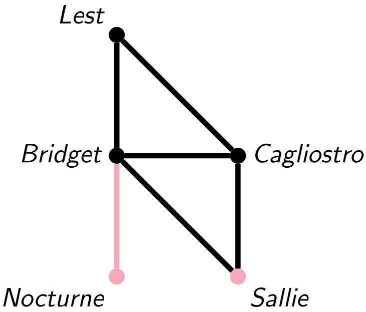
Monitoring Networks through Distances



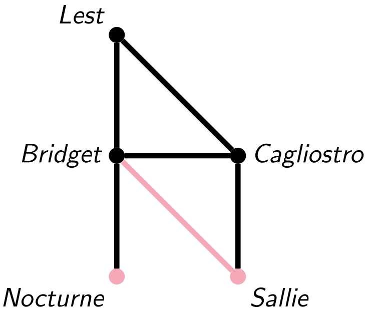
Monitoring Networks through Distances



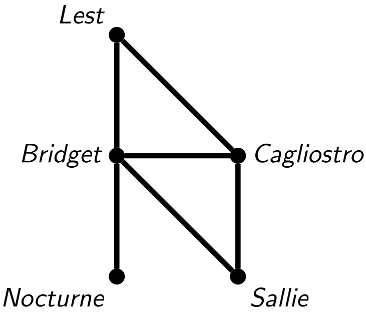
Monitoring Networks through Distances



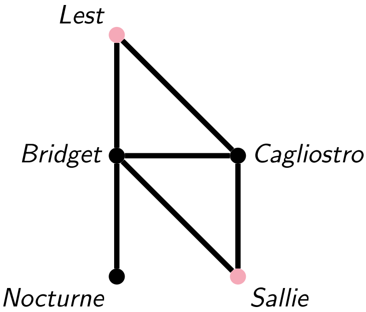
Monitoring Networks through Distances



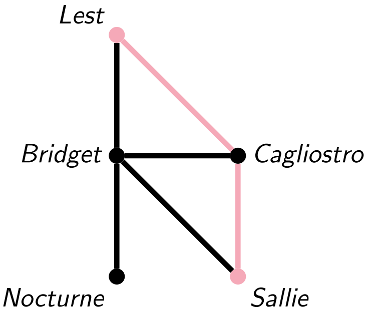
Monitoring Networks through Distances



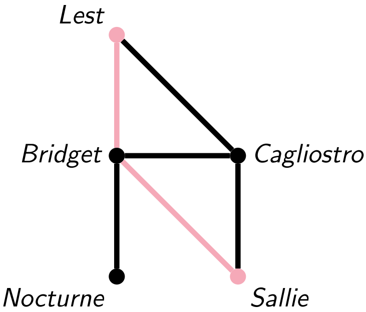
Monitoring Networks through Distances



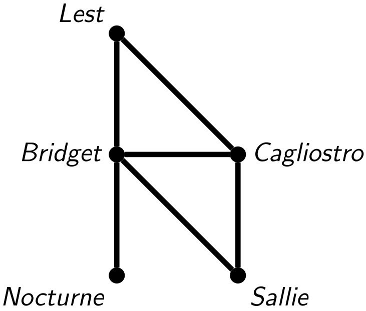
Monitoring Networks through Distances



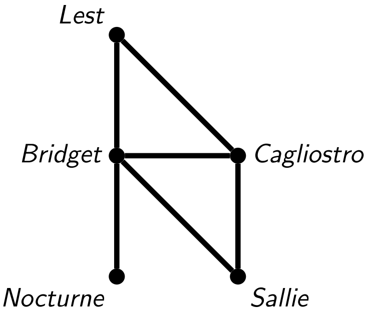
Monitoring Networks through Distances



Monitoring Networks through Distances



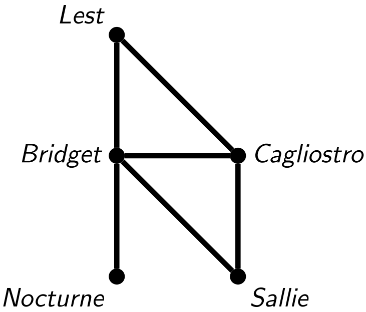
Monitoring Networks through Distances



We make the following hypotheses:

- messages have a timestamp;
- every step takes the same time;
- messages take a shortest path;
- everyone knows their distance to others.

Monitoring Networks through Distances



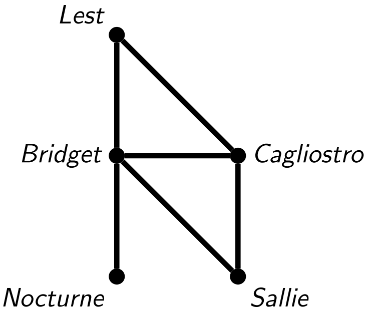
We make the following hypotheses:

- messages have a timestamp;
- every step takes the same time;
- messages take a shortest path;
- everyone knows their distance to others.

Link with networks

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know a failure happened.

Monitoring Networks through Distances



We make the following hypotheses:

- messages have a timestamp;
- every step takes the same time;
- messages take a shortest path;
- everyone knows their distance to others.

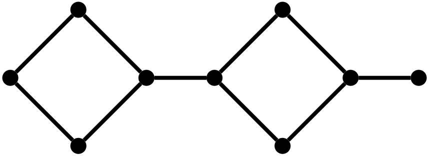
Link with networks

This object simulates probes monitoring a network: if the value of the ping between two probes increases, then one can know a failure happened.

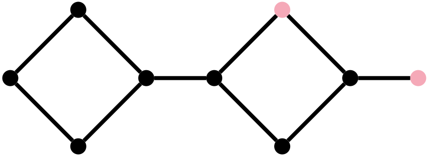
Goal

We want to minimise the number of probes.

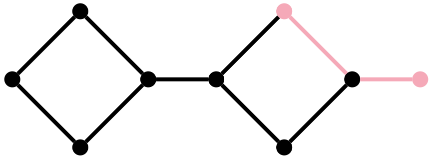
Monitoring Edge Geodetics



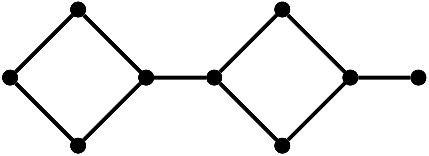
Monitoring Edge Geodetics



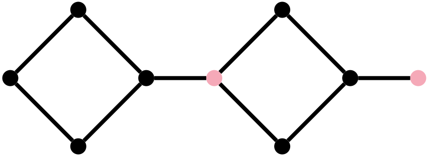
Monitoring Edge Geodetics



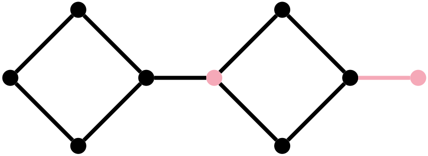
Monitoring Edge Geodetics



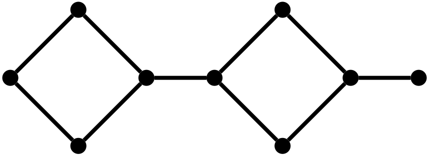
Monitoring Edge Geodetics



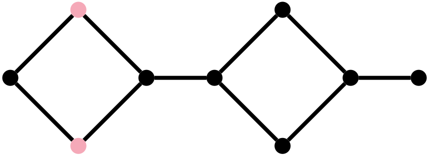
Monitoring Edge Geodetics



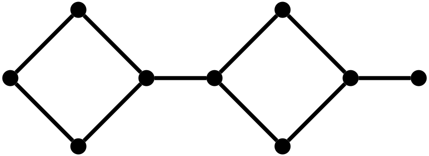
Monitoring Edge Geodetics



Monitoring Edge Geodetics



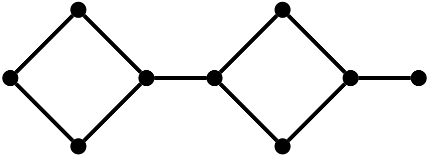
Monitoring Edge Geodetics



Definition (MEG-set) [FNRS23]

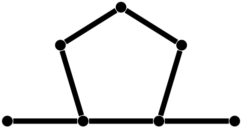
A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.

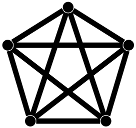
Monitoring Edge Geodetics



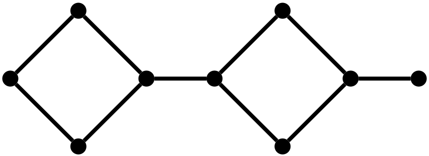
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



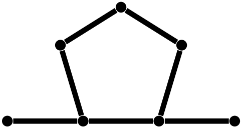


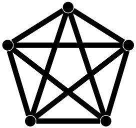
Monitoring Edge Geodetics



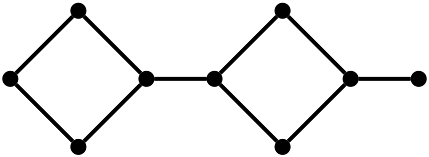
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



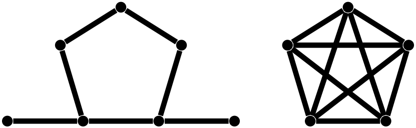


Monitoring Edge Geodetics

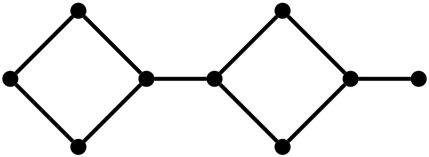


Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.

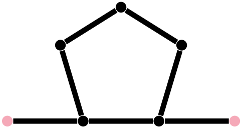


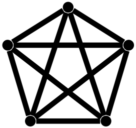
Monitoring Edge Geodetics



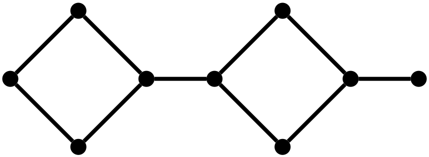
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



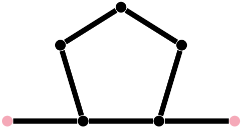


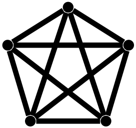
Monitoring Edge Geodetics



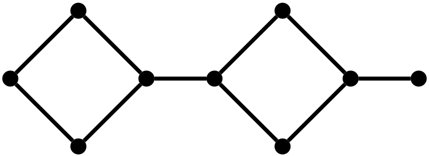
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



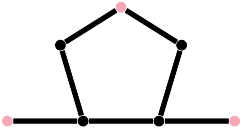


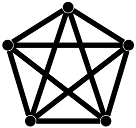
Monitoring Edge Geodetics



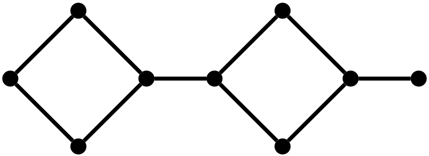
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



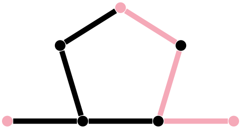


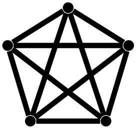
Monitoring Edge Geodetics



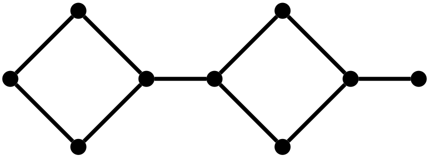
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



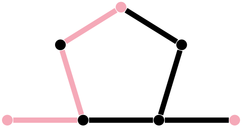


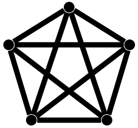
Monitoring Edge Geodetics



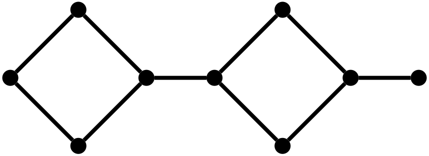
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



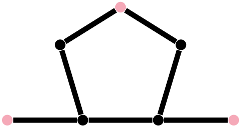


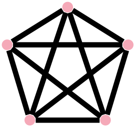
Monitoring Edge Geodetics



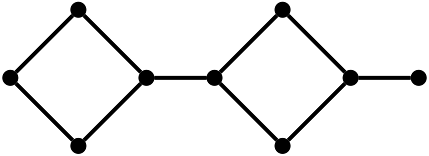
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



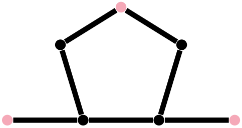


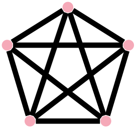
Monitoring Edge Geodetics



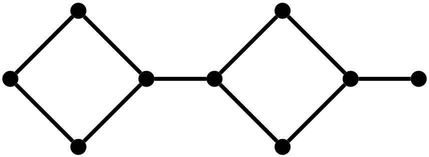
Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



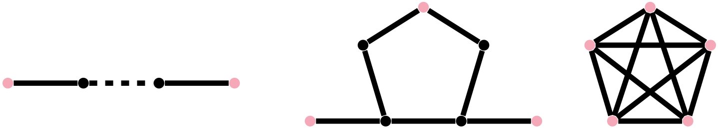


Monitoring Edge Geodetics



Definition (MEG-set) [FNRS23]

A set M of vertices *monitors* an edge e if e lies on all shortest paths between two vertices of M.



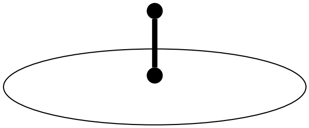
Theorem [Haslegrave, 2023]

Deciding for a graph G and a natural number k whether $meg(G) \leq k$ is NP-complete.

Vital Vertices

Definition

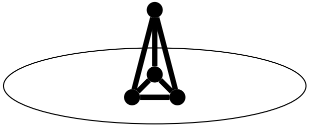
A vertex of G is *vital* if it belongs to all MEG-sets of G.



Vital Vertices

Definition

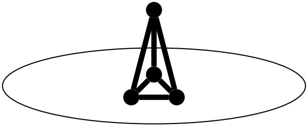
A vertex of G is *vital* if it belongs to all MEG-sets of G.

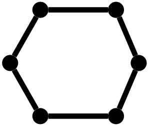


Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.

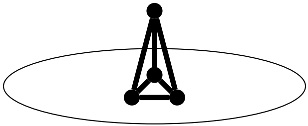


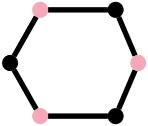


Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.

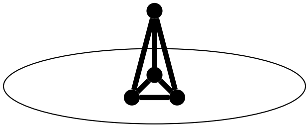


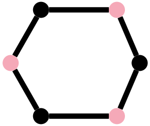


Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.

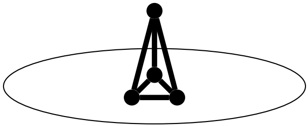


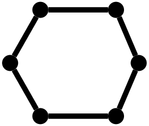


Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.





Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.

Theorem [FMMSST24]

We can find the set of vital vertices of a graph in polynomial time.

Vital Vertices

Definition

A vertex of G is *vital* if it belongs to all MEG-sets of G.

Theorem [FMMSST24]

We can find the set of vital vertices of a graph in polynomial time.

The set of vital vertices
is enough for:

- cographs;
- split graphs;
- block graphs;
- interval graphs.

Vital Vertices

Definition

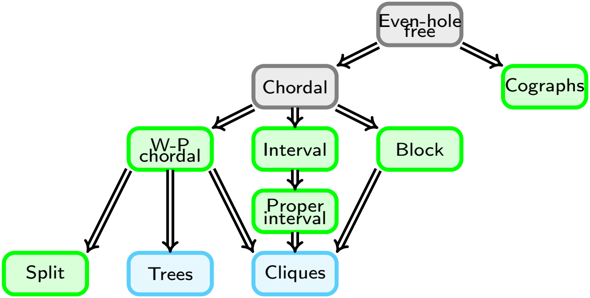
A vertex of G is *vital* if it belongs to all MEG-sets of G.

Theorem [FMMSST24]

We can find the set of vital vertices of a graph in polynomial time.

The set of vital vertices is enough for:

- cographs;
- split graphs;
- block graphs;
- interval graphs.



Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Main Theorem [FMMSST24]

The set of vital vertices of an interval graph is a MEG-set.

Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Main Theorem [FMMSST24]

The set of vital vertices of an interval graph is a MEG-set.

Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Main Theorem [FMMSST24]

The set of vital vertices of an interval graph is a MEG-set.

Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Main Theorem [FMMSST24]

The set of vital vertices of an interval graph is a MEG-set.

Idea of the Interval Graphs Result

Lemma [FMMSST24]

In an interval graph, a vertex is vital if and only if its neighbourhood has diameter at most 4.

Lemma [FMMSST24]

All interval graphs have a representation where the leftmost and rightmost intervals are vital vertices.

Main Theorem [FMMSST24]

The set of vital vertices of an interval graph is a MEG-set.

Complexity Aspects of MEG-sets

Theorem [Haslegrave, 2023]

Deciding for a graph G and a natural number k whether $meg(G) \leq k$ is NP-complete.

Complexity Aspects of MEG-sets

Theorem [Haslegrave, 2023]

Deciding for a graph G and a natural number k whether $meg(G) \leq k$ is NP-complete.

Our results ([FMSST25]):

- MEG-set is XP by solution size.
- MEG-set is FPT by clique-width plus diameter.
- MEG-set is FPT by tree-width on chordal graphs.

Parameterised Complexity

Complexity Aspects of MEG-sets

Theorem [Haslegrave, 2023]

Deciding for a graph G and a natural number k whether $meg(G) \leq k$ is NP-complete.

Our results ([FMSST25]):

- MEG-set is XP by solution size.
- MEG-set is FPT by clique-width plus diameter.
- MEG-set is FPT by tree-width on chordal graphs.

For a graph G, there is a polynomial-time algorithm to:

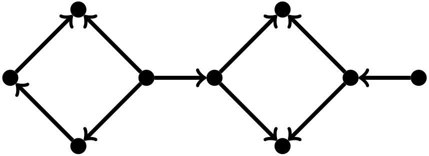
- find a MEG-set of G of size $meg(G) \cdot \sqrt{n \ln m}$, but
- not of size $c \cdot meg(G)$ for any $c > 1$.

Parameterised Complexity

Approximation Algorithms

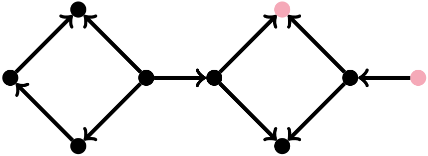
Oriented version

In fact, we only need to be able to define paths to study monitoring.



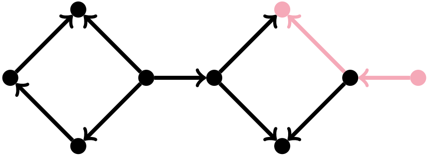
Oriented version

In fact, we only need to be able to define paths to study monitoring.



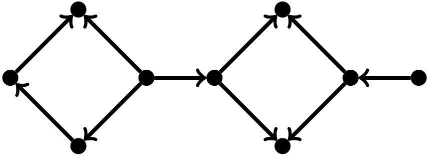
Oriented version

In fact, we only need to be able to define paths to study monitoring.



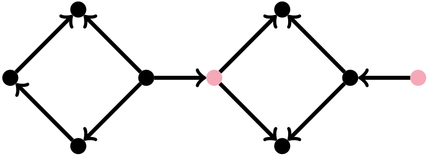
Oriented version

In fact, we only need to be able to define paths to study monitoring.



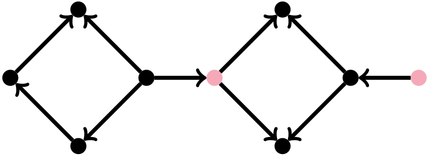
Oriented version

In fact, we only need to be able to define paths to study monitoring.



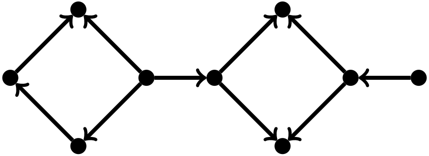
Oriented version

In fact, we only need to be able to define paths to study monitoring.



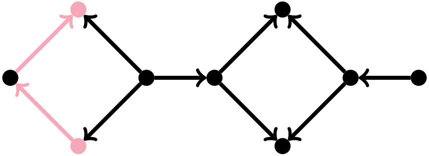
Oriented version

In fact, we only need to be able to define paths to study monitoring.



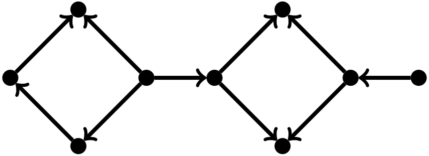
Oriented version

In fact, we only need to be able to define paths to study monitoring.



Oriented version

In fact, we only need to be able to define paths to study monitoring.

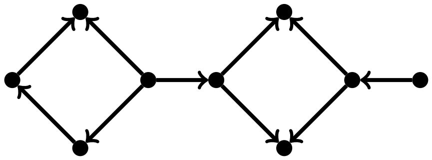


Definition [DFMPS25+]

A set M of vertices *monitors* an arc $\overrightarrow{a}$ if $\overrightarrow{a}$ lies on all shortest paths between two vertices of M.

Oriented version

In fact, we only need to be able to define paths to study monitoring.



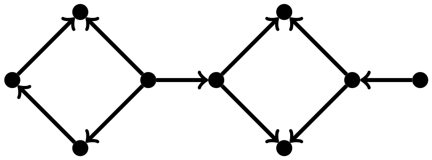
Problem	Complexity
k-mag	NP
Is-Extremal	P
MAG^+	NP
Vital	P

Definition [DFMPS25⁺]

A set M of vertices *monitors* an arc $\overrightarrow{a}$ if $\overrightarrow{a}$ lies on all shortest paths between two vertices of M.

Oriented version

In fact, we only need to be able to define paths to study monitoring.



Problem	Complexity
k-mag	NP
Is-Extremal	P
MAG^+	NP
Vital	P

Definition [DFMPS25+]

A set M of vertices *monitors* an arc $\overrightarrow{a}$ if $\overrightarrow{a}$ lies on all shortest paths between two vertices of M.

Complete characterisation for:

- paths;
- cycles;
- tournaments;
- trees;
- transitive graphs.

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;
- generalised notions to oriented graphs.

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;
- generalised notions to oriented graphs.

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;
- generalised notions to oriented graphs.

Some open perspectives:

- hardness of MEG-sets on chordal graphs;

Conclusion on Monitoring

Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;
- generalised notions to oriented graphs.

Some open perspectives:

- hardness of MEG-sets on chordal graphs;
- investigate solution size as a parameter;

Conclusion on Monitoring

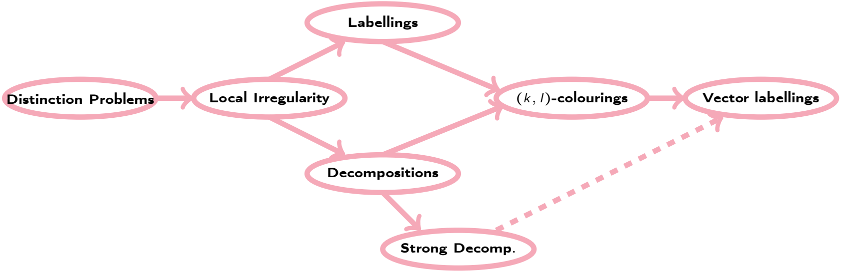
Our results on monitoring:

- polynomial characterisation of vital vertices;
- polynomial-time algorithm to find minimum MEG-sets for several classes of graphs;
- parameterised algorithm for this problem;
- generalised notions to oriented graphs.

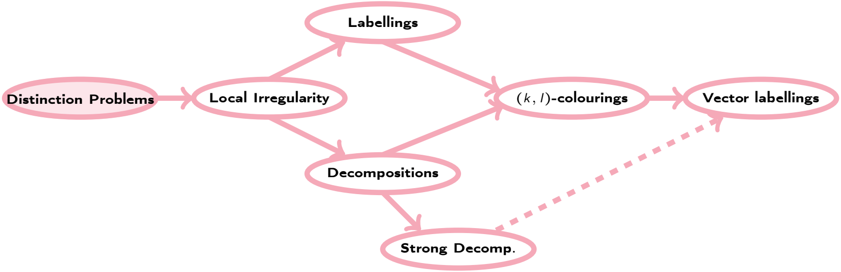
Some open perspectives:

- hardness of MEG-sets on chordal graphs;
- investigate solution size as a parameter;
- investigate useless vertices.

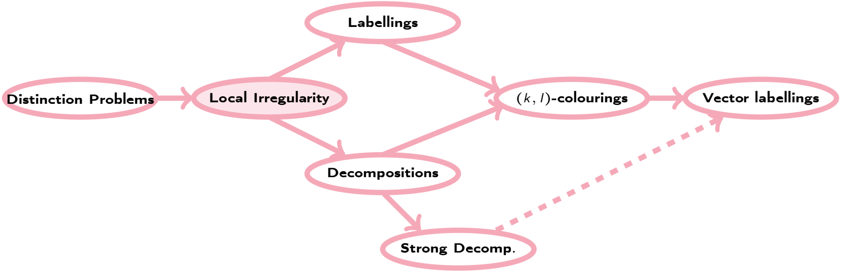
Distinction Problems in Graphs



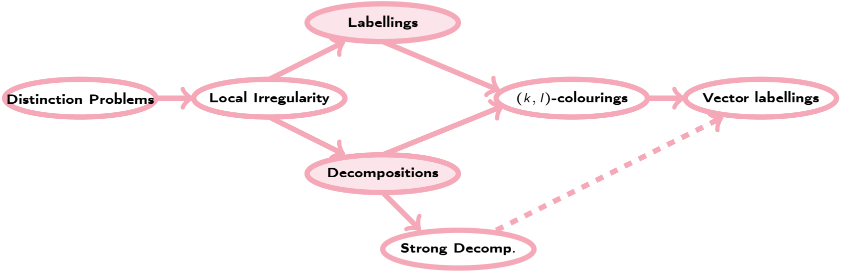
Distinction Problems in Graphs



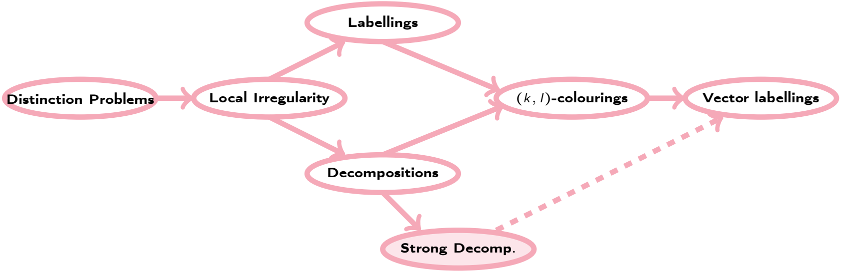
Distinction Problems in Graphs



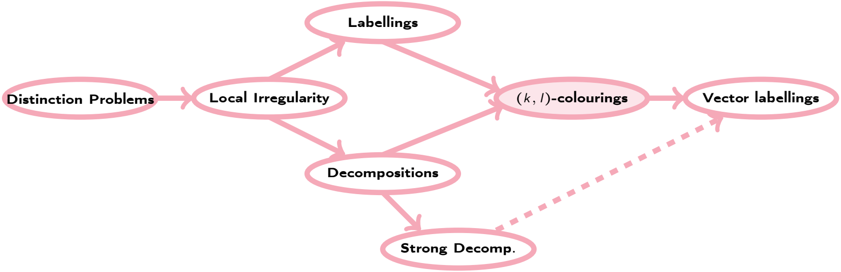
Distinction Problems in Graphs



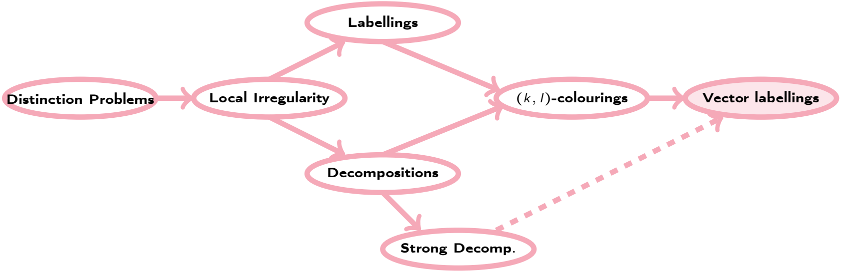
Distinction Problems in Graphs



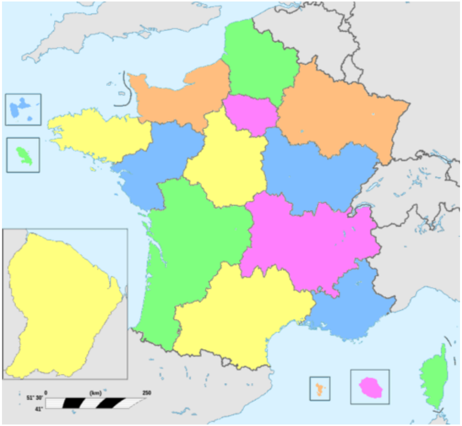
Distinction Problems in Graphs



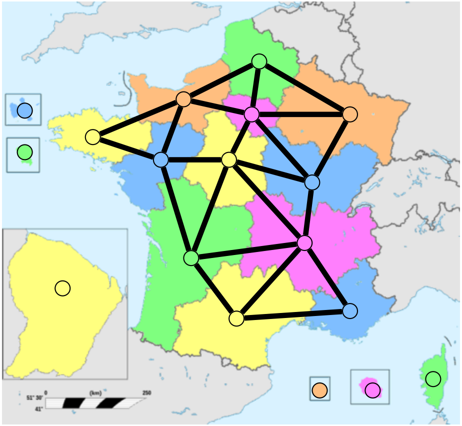
Distinction Problems in Graphs



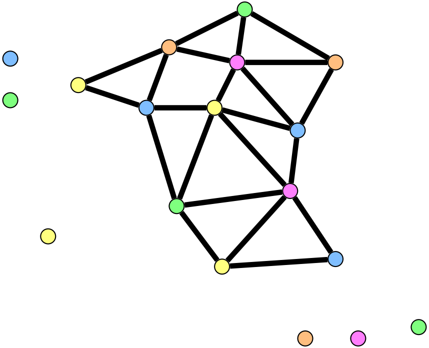
Distinguishing Vertices in Graphs



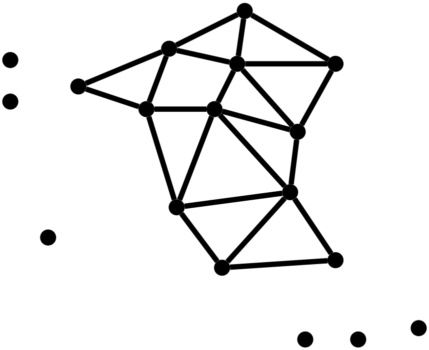
Distinguishing Vertices in Graphs



Distinguishing Vertices in Graphs



Distinguishing Vertices in Graphs



Local Irregularity

Definition

A graph is *locally irregular* if every two adjacent vertices have different degrees.

Local Irregularity

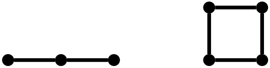
Definition

A graph is *locally irregular* if every two adjacent vertices have different degrees.

Local Irregularity

Definition

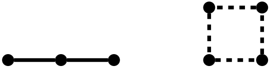
A graph is *locally irregular* if every two adjacent vertices have different degrees.



Local Irregularity

Definition

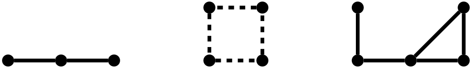
A graph is *locally irregular* if every two adjacent vertices have different degrees.



Local Irregularity

Definition

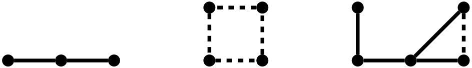
A graph is *locally irregular* if every two adjacent vertices have different degrees.



Local Irregularity

Definition

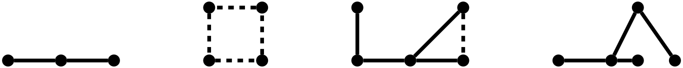
A graph is *locally irregular* if every two adjacent vertices have different degrees.



Local Irregularity

Definition

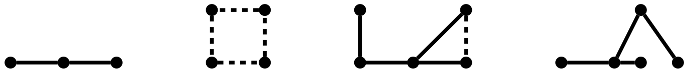
A graph is *locally irregular* if every two adjacent vertices have different degrees.



Local Irregularity

Definition

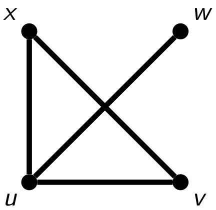
A graph is *locally irregular* if every two adjacent vertices have different degrees.

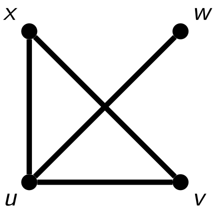


Observation

Not all graphs are locally irregular.

Creating Irregularity

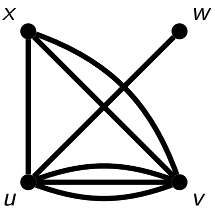


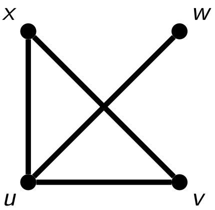


Multigraphs

Decompositions

Creating Irregularity

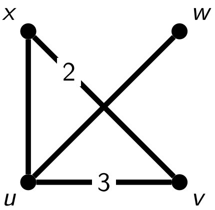


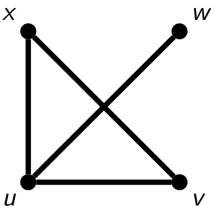


Multigraphs

Decompositions

Creating Irregularity

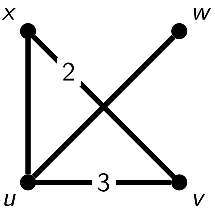


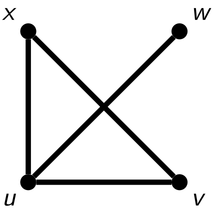


Multigraphs

Decompositions

Creating Irregularity



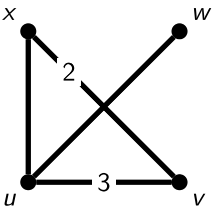


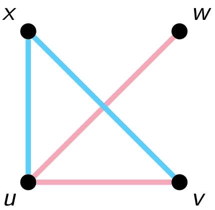
Exceptions	K_2
Lower Bound	3 [DW, 2011]
Upper Bound	3 [Keusch, 2024]

Multigraphs

Decompositions

Creating Irregularity



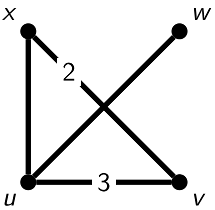


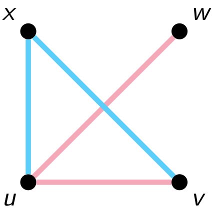
Exceptions	K_2
Lower Bound	3 [DW, 2011]
Upper Bound	3 [Keusch, 2024]

Multigraphs

Decompositions

Creating Irregularity





Exceptions	K_2
Lower Bound	3 [DW, 2011]
Upper Bound	3 [Keusch, 2024]

Exceptions	Polynomial [BB$^+$15]
Lower Bound	4 [SŠ21]
Upper Bound	220 [LPS18]

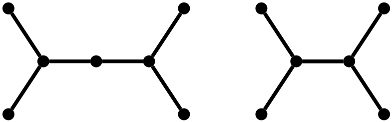
Multigraphs

Decompositions

Strong Local Irregularity

Definition [BM25+]

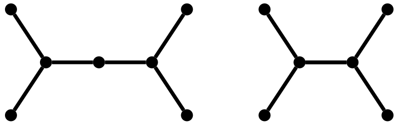
A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



Strong Local Irregularity

Definition [BM25+]

A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



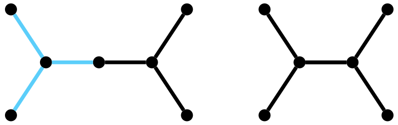
S.l.i. decomposition

If a graph is not s.l.i., then we can try to decompose it into s.l.i. graphs.

Strong Local Irregularity

Definition [BM25+]

A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



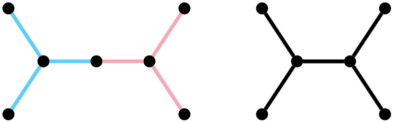
S.l.i. decomposition

If a graph is not s.l.i., then we can try to decompose it into s.l.i. graphs.

Strong Local Irregularity

Definition [BM25+]

A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



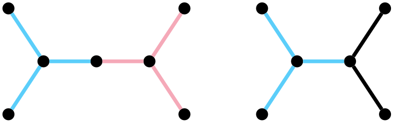
S.l.i. decomposition

If a graph is not s.l.i., then we can try to decompose it into s.l.i. graphs.

Strong Local Irregularity

Definition [BM25+]

A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



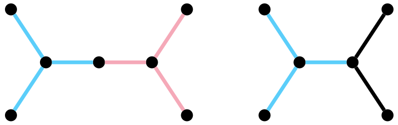
S.l.i. decomposition

If a graph is not s.l.i., then we can try to decompose it into s.l.i. graphs.

Strong Local Irregularity

Definition [BM25+]

A graph is *strongly locally irregular* (s.l.i. for short) if the degrees of every two adjacent vertices differ by at least 2.



S.l.i. decomposition

If a graph is not s.l.i., then we can try to decompose it into s.l.i. graphs.

Theorem [BM25+]

Deciding whether a graph can be decomposed into s.l.i. graphs is NP-complete.

Our Results on s.l.i. Decompositions

Theorem [BM25+]

Deciding whether $\chi_{\mathrm{s.l.i.}}(G) \leq 2$ holds for a bipartite graph G is NP-complete.

Our Results on s.l.i. Decompositions

Theorem [BM25+]

Deciding whether $\chi_{\mathrm{s.l.i.}}(G) \leq 2$ holds for a bipartite graph G is NP-complete.

Class	Decomposability	Lower Bound	Upper Bound
All graphs	NP	16	Unknown
Trees	P	5	16
Subcubic	P	7	7
$\delta \geq 5$	P	Unknown	Unknown

$\chi_{\mathrm{s.l.i.}}$ of Trees

Theorem [BM25+]

If T is a s.l.i. decomposable tree, then $\chi_{\mathrm{s.l.i.}}(T) \leq 16$.

$\chi_{\text{s.l.i.}}$ of Trees

Theorem [BM25⁺]

If T is a s.l.i. decomposable tree, then $\chi_{\text{s.l.i.}}(T) \leq 16$.

Theorem [BM25⁺]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\text{s.l.i.}}(T) \leq k$ holds.

$\chi_{\mathrm{s.l.i.}}$ of Trees

Theorem [BM25+]

If T is a s.l.i. decomposable tree, then $\chi_{\mathrm{s.l.i.}}(T) \leq 16$.

Theorem [BM25+]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\mathrm{s.l.i.}}(T) \leq k$ holds.

In a rooted tree, a *shrub* is the subtree induced by an edge.

$\chi_{\text{s.l.i.}}$ of Trees

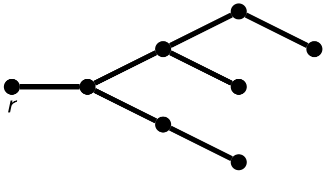
Theorem [BM25⁺]

If T is a s.l.i. decomposable tree, then $\chi_{\text{s.l.i.}}(T) \leq 16$.

Theorem [BM25⁺]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\text{s.l.i.}}(T) \leq k$ holds.

In a rooted tree, a *shrub* is the subtree induced by an edge.



$\chi_{\mathrm{s.l.i.}}$ of Trees

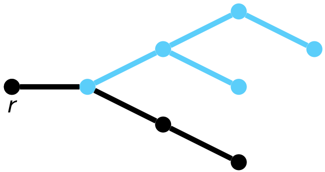
Theorem [BM25+]

If T is a s.l.i. decomposable tree, then $\chi_{\mathrm{s.l.i.}}(T) \leq 16$.

Theorem [BM25+]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\mathrm{s.l.i.}}(T) \leq k$ holds.

In a rooted tree, a *shrub* is the subtree induced by an edge.



$\chi_{\text{s.l.i.}}$ of Trees

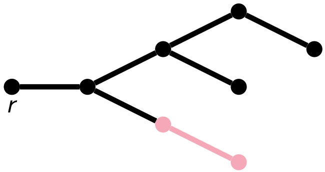
Theorem [BM25+]

If T is a s.l.i. decomposable tree, then $\chi_{\text{s.l.i.}}(T) \leq 16$.

Theorem [BM25+]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\text{s.l.i.}}(T) \leq k$ holds.

In a rooted tree, a *shrub* is the subtree induced by an edge.



$\chi_{\text{s.l.i.}}$ of Trees

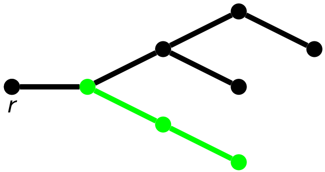
Theorem [BM25+]

If T is a s.l.i. decomposable tree, then $\chi_{\text{s.l.i.}}(T) \leq 16$.

Theorem [BM25+]

For any tree T and any fixed integer $k \geq 1$, we can determine in time $\mathcal{O}(n^k)$ whether $\chi_{\text{s.l.i.}}(T) \leq k$ holds.

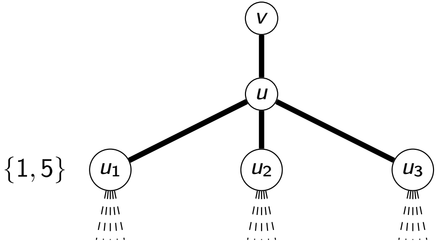
In a rooted tree, a *shrub* is the subtree induced by an edge.



Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

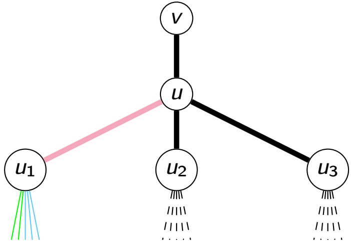


Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

$\{1, 5\}$

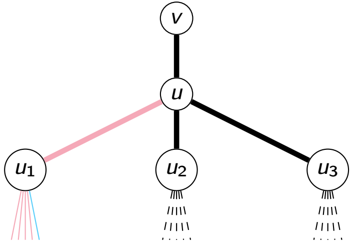


Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

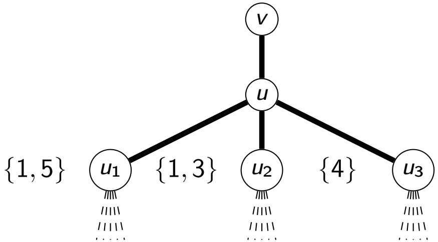
$\{1, 5\}$



Mending Shrub Cuttings Together

Dynamic programming

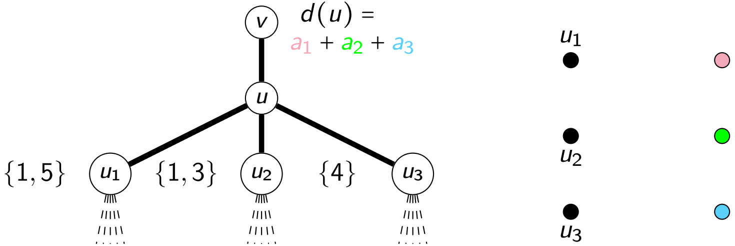
For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.



Mending Shrub Cuttings Together

Dynamic programming

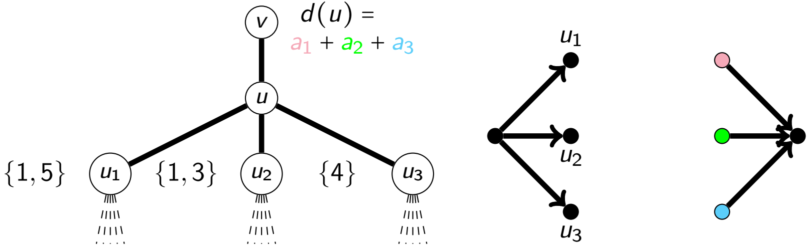
For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.



Mending Shrub Cuttings Together

Dynamic programming

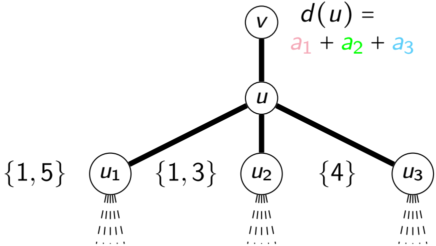
For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

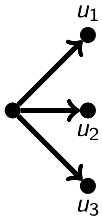


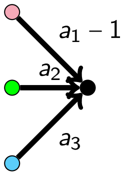
Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.



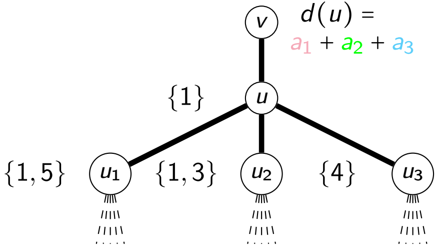


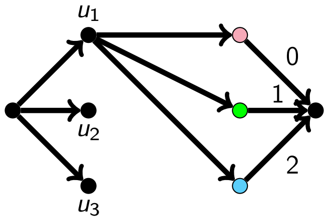


Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

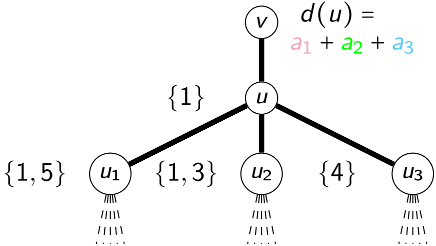


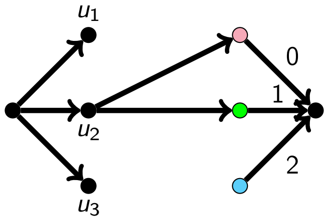


Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

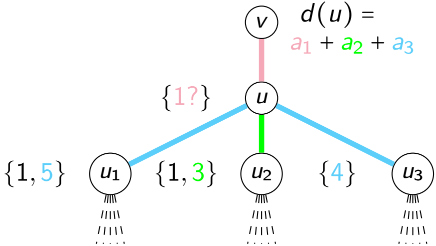


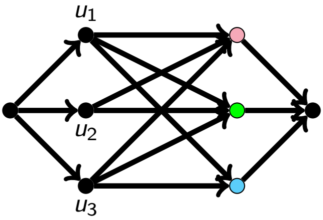


Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.

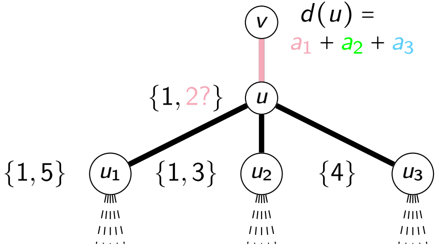


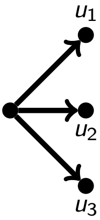


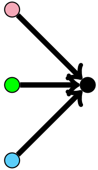
Mending Shrub Cuttings Together

Dynamic programming

For a shrub uu_i, we compute all the possible degrees of u_i in the colour of uu_i in some *loosely s.l.i. decomposition*.



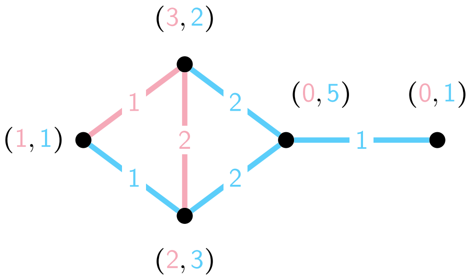




Having both a Labelling and a Decomposition

Intuition [BB+19]

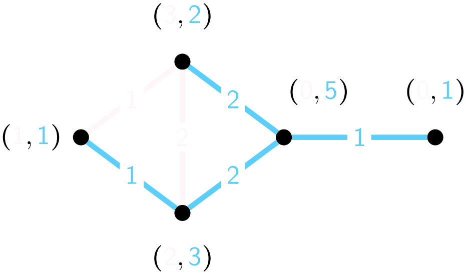
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

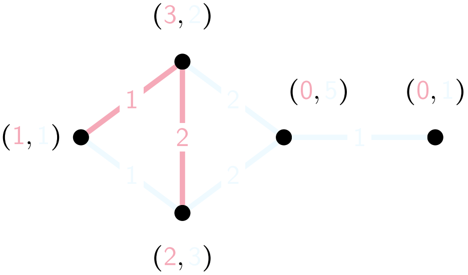
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

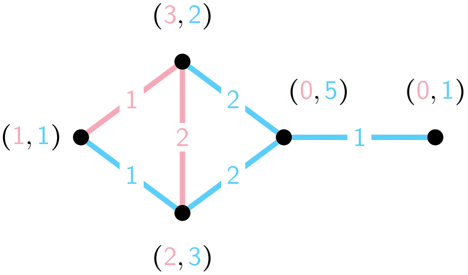
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

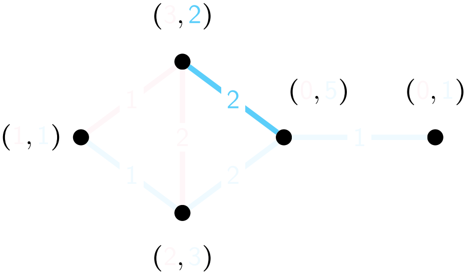
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

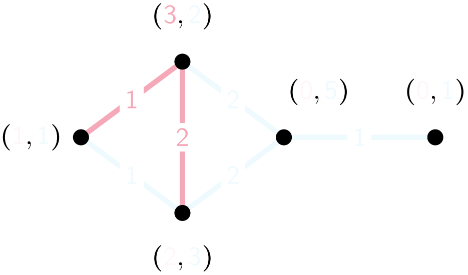
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

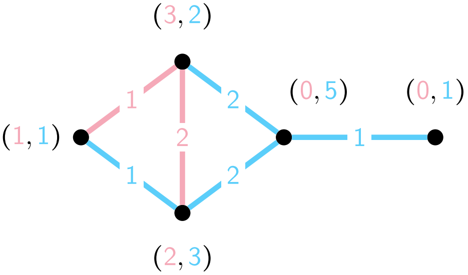
If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



Having both a Labelling and a Decomposition

Intuition [BB+19]

If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



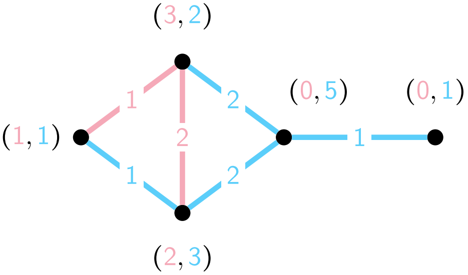
Strong $(2, 2)$ Conjecture [BB+19]

"All" graphs admit a strong $(2, 2)$-colouring.

Having both a Labelling and a Decomposition

Intuition [BB+19]

If we use both a k-decomposition and a l-labelling, then we may reach better bounds.



[BHMM25] We prove it for:

- cacti graphs;
- subcubic outerplanar graphs;
- graphs of *maximum average degree* less than $\frac{9}{4}$;
- powers of cycles;
- complete k-partite graphs.

Strong $(2,2)$ Conjecture [BB+19]

"All" graphs admit a strong $(2,2)$-colouring.

Polarised Labellings of Graphs

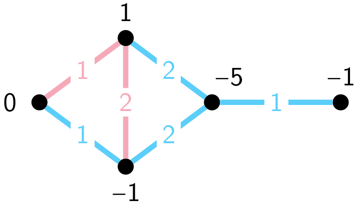
Intuition [BHMM25]

We interpret the two colours as a polarity.

Polarised Labellings of Graphs

Intuition [BHMM25]

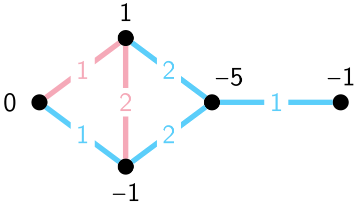
We interpret the two colours as a polarity.



Polarised Labellings of Graphs

Intuition [BHMM25]

We interpret the two colours as a polarity.



Properness

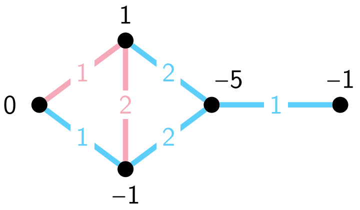
We can consider either:

- the sum, or
- the absolute value.

Polarised Labellings of Graphs

Intuition [BHMM25]

We interpret the two colours as a polarity.



Properness

We can consider either:

- the sum, or
- the absolute value.

Signature

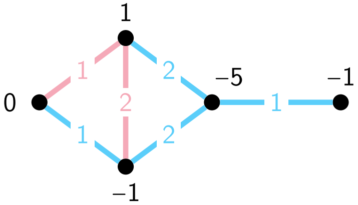
The signature is either:

- fixed, or
- can be chosen.

Polarised Labellings of Graphs

Intuition [BHMM25]

We interpret the two colours as a polarity.



Properness

We can consider either:

- the sum, or
- the absolute value.

Signature

The signature is either:

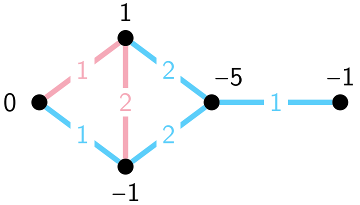
- fixed, or
- can be chosen.

	proper	abs-proper
fixed signature	≤ 5	≤ 9
free signature	≤ 3 [Keu24]	≤ 3 [Keu24]

Polarised Labellings of Graphs

Intuition [BHMM25]

We interpret the two colours as a polarity.



Properness

We can consider either:

- the sum, or
- the absolute value.

Signature

The signature is either:

- fixed, or
- can be chosen.

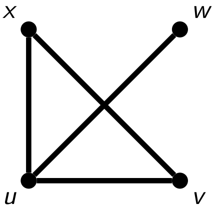
	proper	abs-proper
fixed signature	≤ 5	≤ 9
free signature	≤ 3 [Keu24]	≤ 3 [Keu24]

Conjecture (optimal)

← These upper bounds are 2.

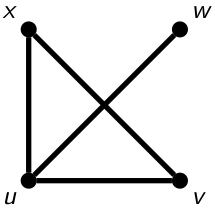
(k, l)-colourings in a Euclidean Space

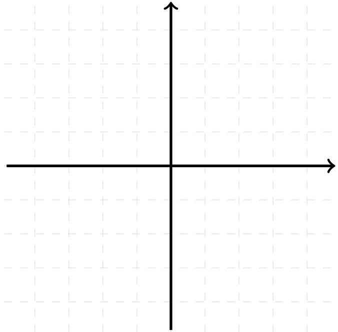
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.



(k, l)-colourings in a Euclidean Space

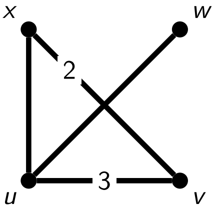
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

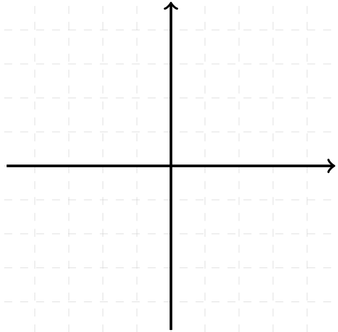




(k, l)-colourings in a Euclidean Space

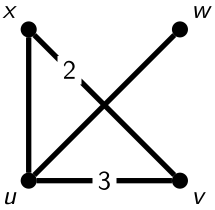
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.





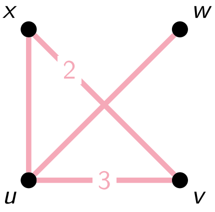
(k, l)-colourings in a Euclidean Space

We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.



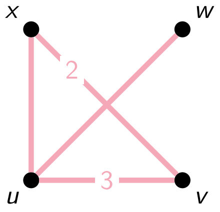
(k, l)-colourings in a Euclidean Space

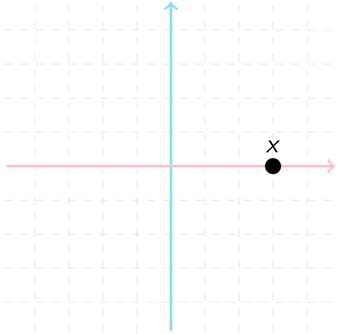
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.



(k, l)-colourings in a Euclidean Space

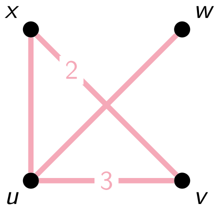
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

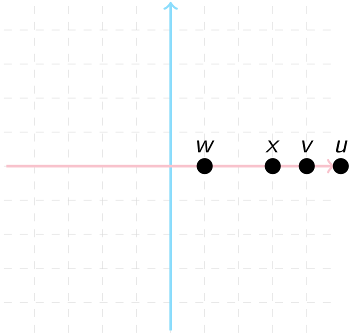




(k, l)-colourings in a Euclidean Space

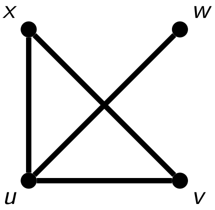
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.





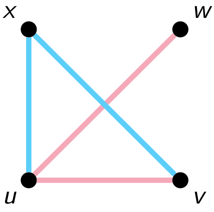
(k, l)-colourings in a Euclidean Space

We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.



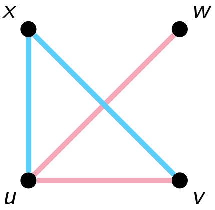
(k, l)-colourings in a Euclidean Space

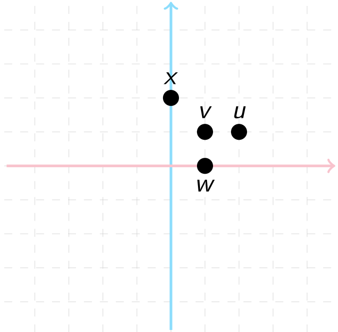
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.



(k, l)-colourings in a Euclidean Space

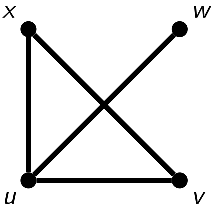
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.





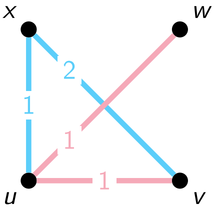
(k,l)-colourings in a Euclidean Space

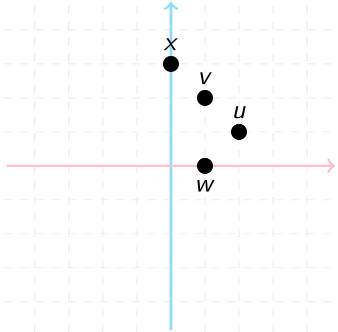
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k,l)-colourings.



(k, l)-colourings in a Euclidean Space

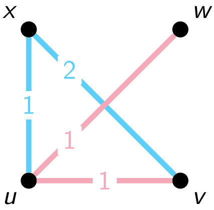
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

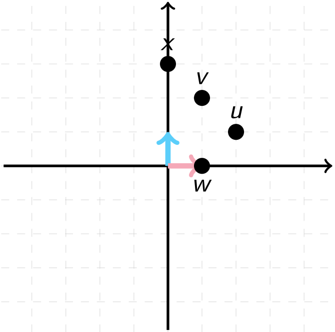




(k, l)-colourings in a Euclidean Space

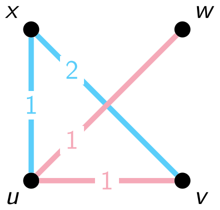
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

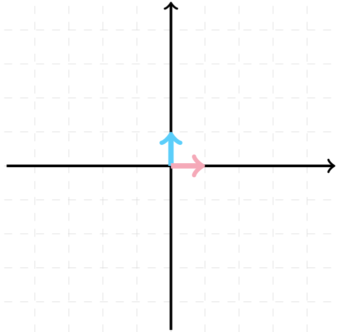




(k, l)-colourings in a Euclidean Space

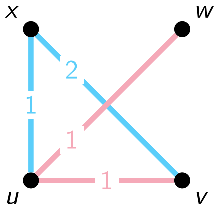
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

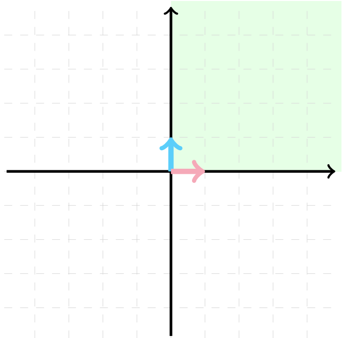




(k, l)-colourings in a Euclidean Space

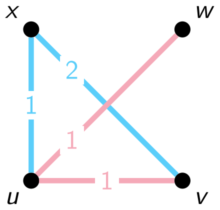
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

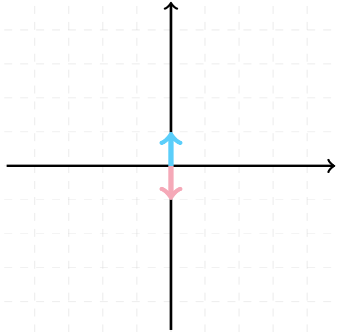




(k, l)-colourings in a Euclidean Space

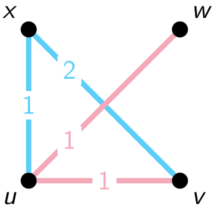
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

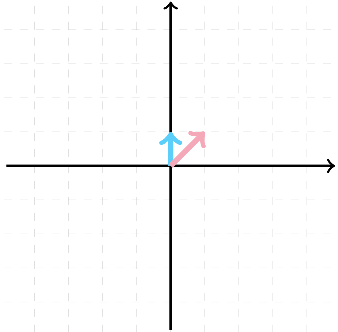




(k, l)-colourings in a Euclidean Space

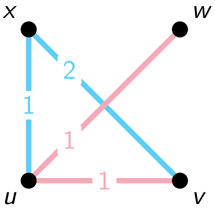
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

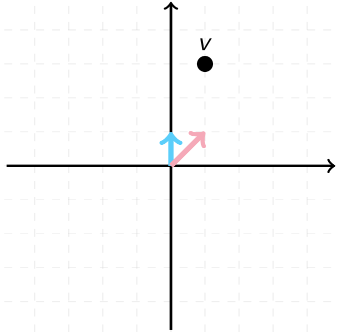




(k, l)-colourings in a Euclidean Space

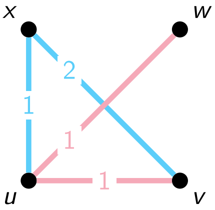
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

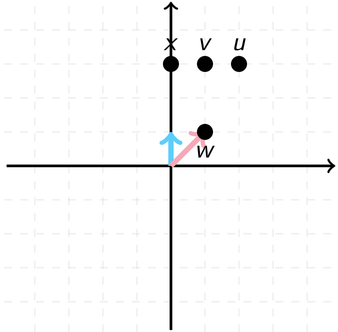




(k, l)-colourings in a Euclidean Space

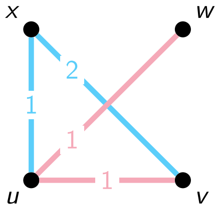
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

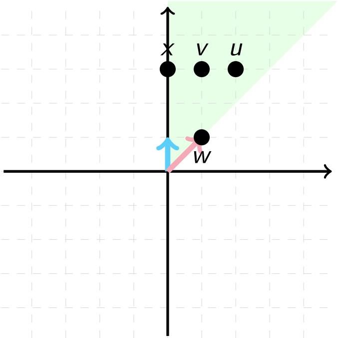




(k, l)-colourings in a Euclidean Space

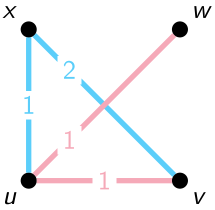
We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.

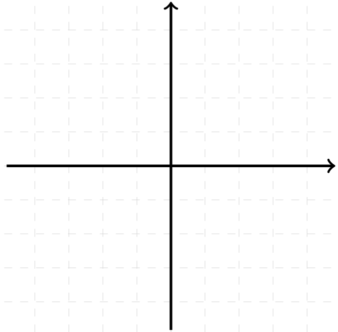




(k, l)-colourings in a Euclidean Space

We also introduce a way to encapsulate all previous definitions, as a generalisation of (k, l)-colourings.





Theorem (informal)

Given a graph G and two vectors, deciding whether G can be properly edge-coloured with these two vectors is NP-hard.

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2,2)$ Conjecture;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Some open perspectives:

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Some open perspectives:

- investigate properties of strong irregularity and links with other known parameters;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Some open perspectives:

- investigate properties of strong irregularity and links with other known parameters;
- the Strong $(2, 2)$ Conjecture is still open to this day;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2, 2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Some open perspectives:

- investigate properties of strong irregularity and links with other known parameters;
- the Strong $(2, 2)$ Conjecture is still open to this day;
- prove that "all" graphs admit good polarised 2-labellings;

Conclusion on Irregularity

Our work on local irregularity:

- stronger irregularity and decompositions;
- results supporting the Strong $(2,2)$ Conjecture;
- a general formalism for unified statements of previously known results;
- a way to accommodate irregularity for signed graphs.

Some open perspectives:

- investigate properties of strong irregularity and links with other known parameters;
- the Strong $(2,2)$ Conjecture is still open to this day;
- prove that "all" graphs admit good polarised 2-labellings;
- investigate the full extent of vectors as a mean of irregularity.

Conclusion

In this thesis, we tackled multiple problems with:

- structural approaches (studying simple cases, establishing structural properties of solutions), and
- algorithmic approaches (proving hardness, designing parameterised algorithms).

This process can be applied on several variants of a problem, with two goals:

- establishing relations between the variants to prove more results;
- providing a gathering framework to give insights on previous results.

Published Works

Works on Irregularity:

- Going Wide with the 1-2-3 Conjecture, BHM, *DAM*
- On inducing degenerate sums through 2-labellings, BHM, *G&C*
- Adding direction constraints to the 1-2-3 Conjecture, BHM, *TCS*
- On 1-2-3 Conjecture-like problems in 2-edge-coloured graphs, BHMM, *DM*
- Irregularity Notions for Digraphs, BFHM, *G&C*

Journal

- The Weak $(2,2)$-Labelling Problem for Graphs with Forbidden Induced Structures, BHM *CALDAM23*
- An Improved Bound for Equitable Proper Labellings, BM, *IWOCA24*

Conferences

Works on Monitoring:

- Bounds and extremal graphs for monitoring edge-geodetic sets in graphs, FMSST, *DAM*

Journal

- Monitoring Edge-Geodetic Sets in Graphs: Extremal Graphs, Bounds, Complexity, FMMSST, *CALDAM24*

Conferences

What the Future Holds

Submitted works:

- The Strong (2, 2)-Conjecture for more classes of graphs, BBGM;
- An Improved Bound for Equitable Proper Labellings, BM;
- Strongly Locally Irregular Graphs and Decompositions, BM;
- Pushing Vertices to Make Graphs Irregular, BMO;
- Graph Irregularity via Edge Deletions, BCFMO;

- Monitoring arc-geodetic sets of oriented graphs, DFMPS;
- Algorithms and complexity for monitoring edge-geodetic sets in graphs, FMSST;

What the Future Holds

Submitted works:

- The Strong (2, 2)-Conjecture for more classes of graphs, BBGM;
- An Improved Bound for Equitable Proper Labellings, BM;
- Strongly Locally Irregular Graphs and Decompositions, BM;
- Pushing Vertices to Make Graphs Irregular, BMO;
- Graph Irregularity via Edge Deletions, BCFMO;
- Monitoring arc-geodetic sets of oriented graphs, DFMPS;
- Algorithms and complexity for monitoring edge-geodetic sets in graphs, FMSST;

Current perspectives:

- exploring the vector formalism for irregularity in inner product spaces;
- colourings of anti-prismatic graphs;
- parameterised complexity of identifying some χ-bounded classes of graphs;
- algorithms for MEG-sets.

What the Future Holds

Submitted works:

- The Strong (2, 2)-Conjecture for more classes of graphs, BBGM;
- An Improved Bound for Equitable Proper Labellings, BM;
- Strongly Locally Irregular Graphs and Decompositions, BM;
- Pushing Vertices to Make Graphs Irregular, BMO;
- Graph Irregularity via Edge Deletions, BCFMO;
- Monitoring arc-geodetic sets of oriented graphs, DFMPS;
- Algorithms and complexity for monitoring edge-geodetic sets in graphs, FMSST;

Current perspectives:

- exploring the vector formalism for irregularity in inner product spaces;
- colourings of anti-prismatic graphs;
- parameterised complexity of identifying some χ-bounded classes of graphs;
- algorithms for MEG-sets.

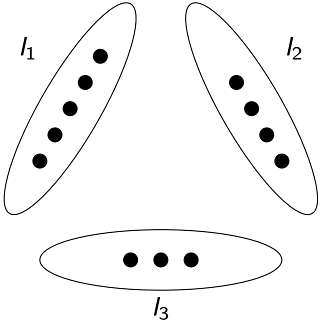
Thank you!

Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

Case $k = 3$



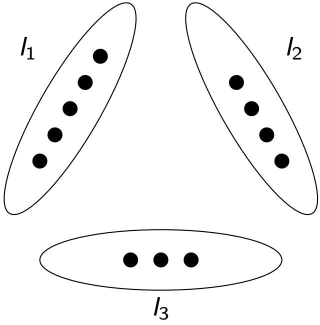
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 > n_2 \geq n_3$



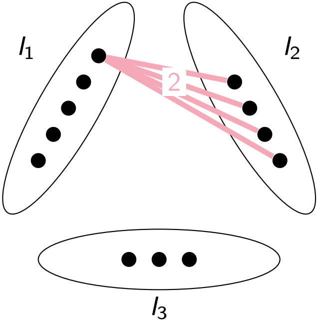
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 > n_2 \geq n_3$



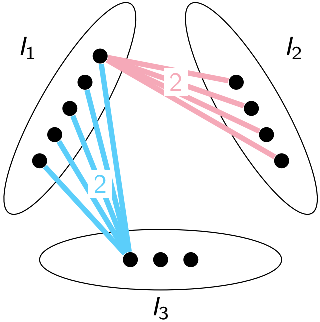
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 > n_2 \geq n_3$



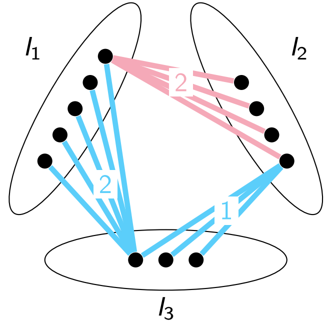
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 > n_2 \geq n_3$



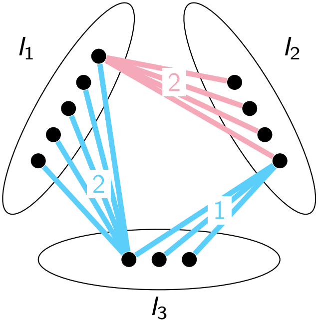
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 > n_2 \geq n_3$



$I_1 : (2n_2, 2n_3)$
$I_2 : (2n_1, n_3)$
$I_3 : (0, 2n_1 + n_2)$

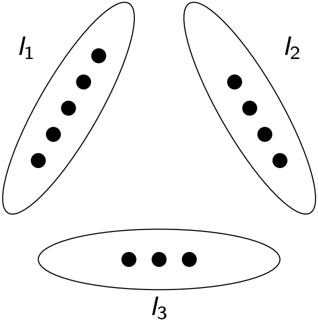
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 > n_3$



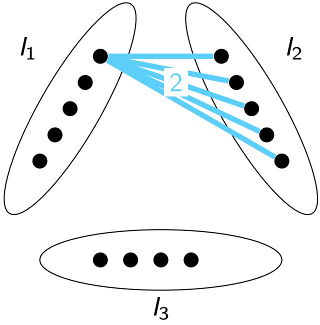
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 > n_3$



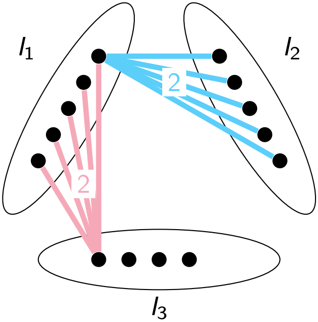
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 > n_3$



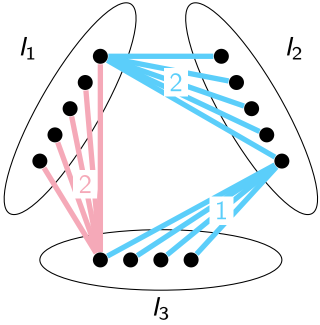
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 > n_3$



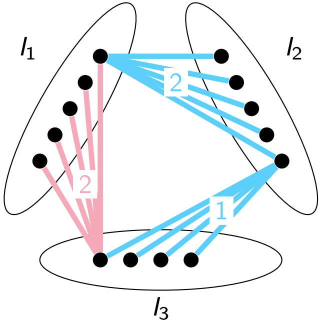
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 > n_3$



$I_1 : (2n_3, 2n_2)$
$I_2 : (0, 2n_2 + n_3)$
$I_3 : (2n_1, n_2)$

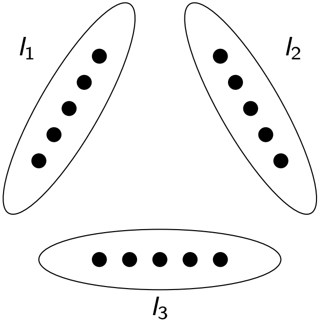
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



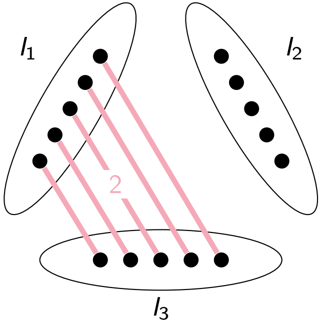
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2, 2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



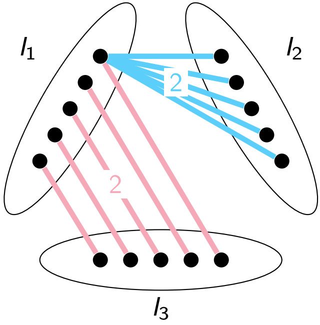
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



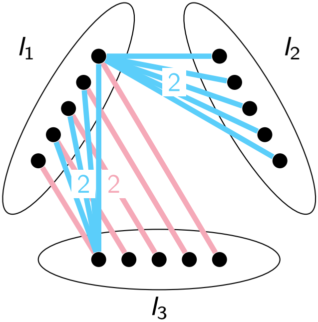
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



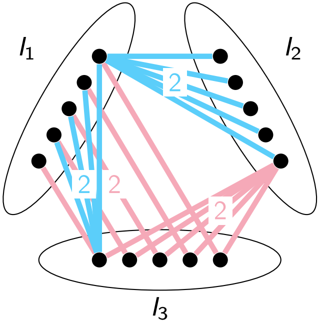
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



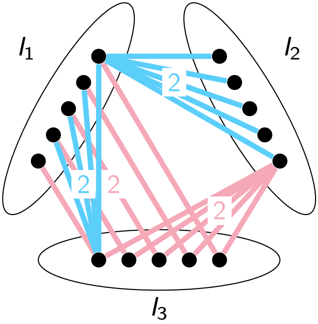
Proof of one of our Results

Theorem

If G is a complete k-partite graph other than K_2 and K_3, then G admits a strong $(2,2)$-colouring.

All vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Case $k = 3$
$n_1 = n_2 = n_3$



$I_1 : (2, 4n_1 - 2)$
$I_2 : (2n_1, 2n_1)$
$I_3 : (2n_1 + 2, 2n_1 - 2)$

Proof of one of our Results (part 2)

Theorem

If G is a complete 3-partite graph other than K_3, then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Proof of one of our Results (part 2)

Theorem

If G is a complete 3-partite graph other than K_3, then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue sum and no two adjacent vertices have red sum equal to 0.

Induction Hypothesis

If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.

Proof of one of our Results (part 2)

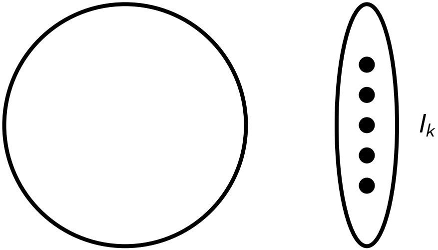
Induction Hypothesis

If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.

Proof of one of our Results (part 2)

Induction Hypothesis

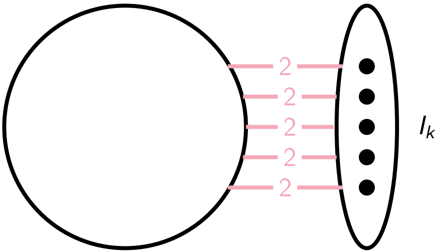
If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.



Proof of one of our Results (part 2)

Induction Hypothesis

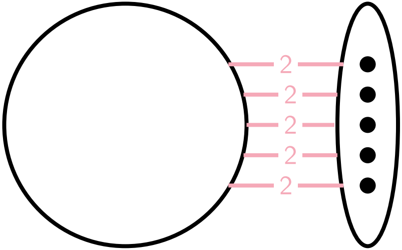
If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.



Proof of one of our Results (part 2)

Induction Hypothesis

If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.

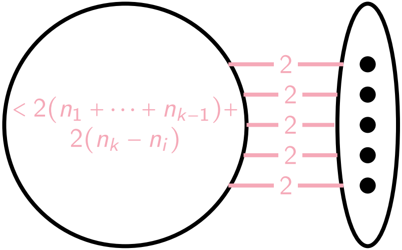


$I_k : (2(n_1 + \cdots + n_{k-1}), 0)$

Proof of one of our Results (part 2)

Induction Hypothesis

If G is a complete k-partite graph for $k \geq 4$ odd (resp. even), then G admits a strong $(2,2)$-colouring such that all vertices have a non-zero blue (resp. red) sum and no two adjacent vertices have red (resp. blue) sum equal to 0.



$I_k : (2(n_1 + \cdots + n_{k-1}), 0)$