Towards Lehel's conjecture for 4-uniform tight cycles

Vincent Pfenninger

University of Birmingham

Joint work with Allan Lo

8th Polish Combinatorial Conference 2020

Edge-Coloured Complete Graphs

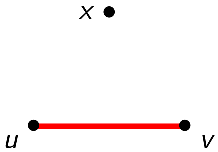
We will consider edge-colourings of the complete graph. That is the assignment of colours to the edges in any way.
By a monochromatic subgraph we mean a subgraph where all the edges are assigned the same colour.

Question

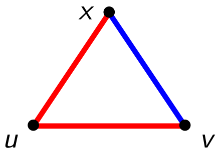
Does every red-blue edge-coloured K_n contain a monochromatic spanning tree?

Edge-Coloured Complete Graphs

Answer: Yes! Assume that there is no blue spanning tree. Hence there are two vertices u and v such that there is no blue path between u and v. In particular, the edge uv is red. Let x be any other vertex.



Answer: Yes! Assume that there is no blue spanning tree. Hence there are two vertices u and v such that there is no blue path between u and v. In particular, the edge uv is red. Let x be any other vertex.
Since there is no blue path from u to v, at least one of the edges xu and xv is red. It follows that there is a red spanning tree.



Monochromatic Cycles?

Lehel's Conjecture

Every red-blue edge-coloured K_n can be partitioned into a red and a blue cycle.

We consider the empty set, a single vertex, and a single edge to be cycles.

Theorem (Łuczak, Rödl, Szemerédi 1998, Allen 2008)

Lehel's conjecture is true for all $n \geq n_0$.

Theorem (Bessy, Thomassé 2010)

Lehel's conjecture is true for all n.

From now on we will assume that n is large enough and that all cycles are vertex-disjoint and monochromatic.

r Colours

Conjecture (Erdős, Gyárfás, Pyber 1991)

Every r-edge-coloured K_n can be partitioned into r monochromatic cycles.

Known results:

- $cr^2 \log r$ cycles (Erdős, Gyárfás, Pyber 1991)
- $100r \log r$ cycles (Gyárfás, Ruszinkó, Sárközy, Szemerédi 2006)
- The conjecture is false. For $r \geq 3$, more than r cycles are needed. (Pokrovskiy 2014)

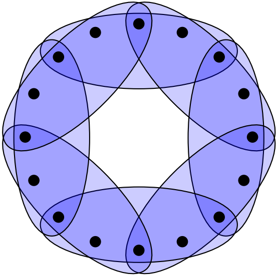
Conjecture (Pokrovskiy 2014)

Every r-edge-coloured K_n contains r monochromatic cycles that cover all but c_r of the vertices.

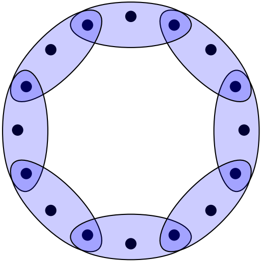
Q: What about hypergraphs?

Cycles in Hypergraphs: Definition

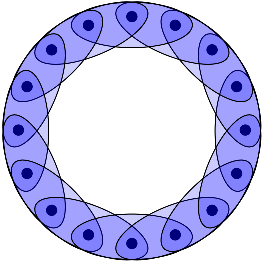
A k-graph H is an pair of sets $(V(H), E(H))$ where $E(H) \subseteq \binom{V(H)}{k}$.
An ℓ-cycle has a cyclic ordering of its vertices such that its edges consist of k consecutive vertices and consecutive edges intersect in ℓ vertices.



$k = 5, \ell = 3$
3-cycle



$k = 3, \ell = 1$
1-cycle
loose cycle



$k = 3, \ell = 2$
$(k-1)$-cycle
tight cycle

Results for Loose Cycles and ℓ-Cycles

In k-graphs we consider any set of vertices of size at most k to be a degenerate cycle.
For r colours:

Theorem (Sárközy 2014)

Every r-edge-coloured K_n^k can be partitioned into at most $50rk\log(rk)$ monochromatic loose cycles.

For 2 colours:

Theorem (Bustamante, Stein 2018)

If $0 < \ell \leq k/2$ and $k - \ell$ divides n, then every red-blue edge-coloured K_n^k contains a red and a blue ℓ-cycle that are disjoint and cover all but at most $4(k - \ell)$ of the vertices.

Results for Tight Cycles

For r colours:

Theorem (Bustamante, Corsten, Frankl, Pokrovskiy, Skokan 2020)

Every r-edge-coloured K_n^k can be partitioned into at most $c(r,k)$ monochromatic tight cycles.

For 2 colours in the 3-uniform case:

Theorem (Bustamante, Hàn, Stein 2019)

Every red-blue edge-coloured K_n^3 contains a red and a blue tight cycle that together cover $(1-o(1))n$ vertices.

Theorem (Garbe, Mycroft, Lang, Lo, Sanhueza-Matamala 2020+)

Every red-blue edge-coloured K_n^3 can be partitioned into two monochromatic tight cycles.

Our Results

How many monochromatic tight cycles do we need to almost partition K_n^k?
In the 4-uniform case 2 cycles is enough:

Theorem (Lo, P 2020+)

Every red-blue edge-coloured K_n^4 contains a red and a blue tight cycle that together cover $(1 - o(1))n$ vertices.

In the 5-uniform case, we proved that 4 cycles are enough:

Theorem (Lo, P 2020+)

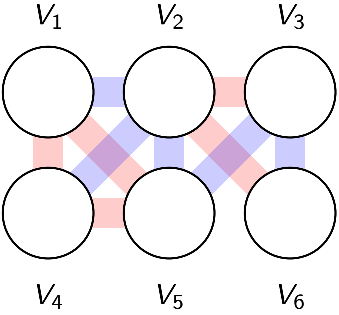
Every red-blue edge-coloured K_n^5 contains four monochromatic tight cycles that together cover $(1 - o(1))n$ vertices.

We prove these results by using a hypergraph version of Łuczak's Connected Matching Method.

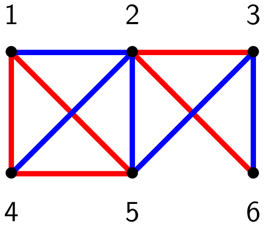
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.

Regularity Partition



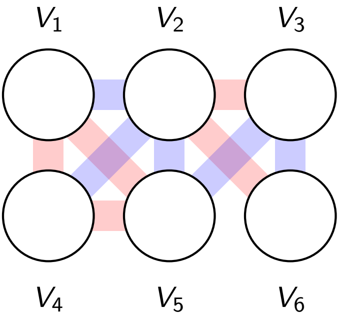
Reduced Graph



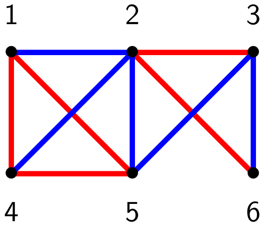
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.

Regularity Partition



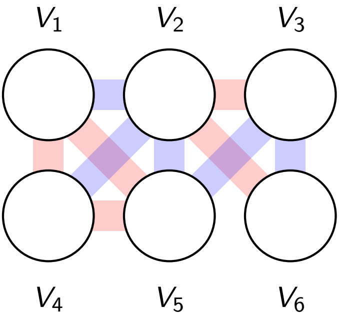
Reduced Graph



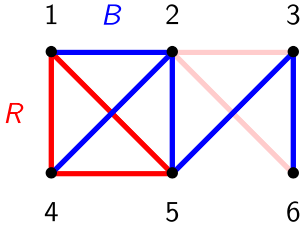
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.

Regularity Partition

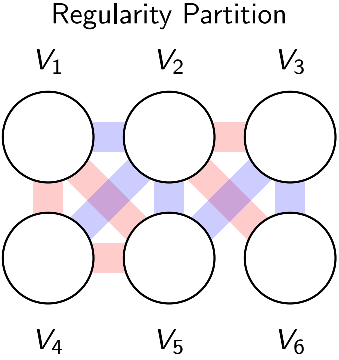


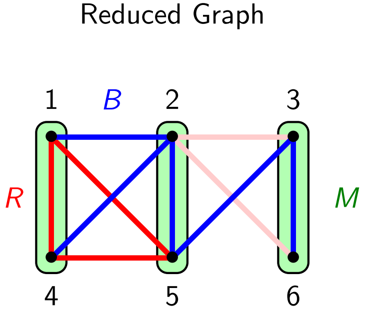
Reduced Graph



Łuczak's Connected Matching Method (Graph Version)

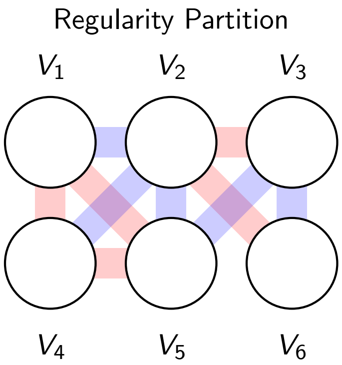
1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.

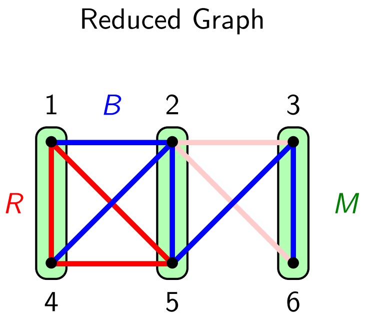




Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.
3. Using the matching M find a red and a blue cycle in the K_n that cover almost all the vertices.

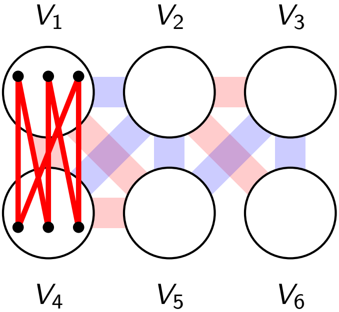




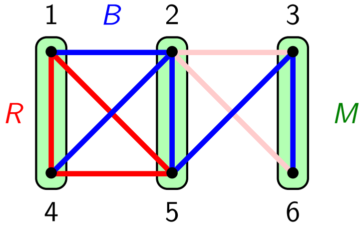
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.
3. Using the matching M find a red and a blue cycle in the K_n that cover almost all the vertices.

Regularity Partition



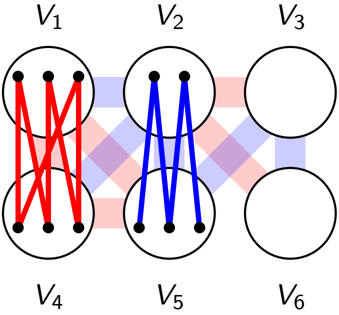
Reduced Graph



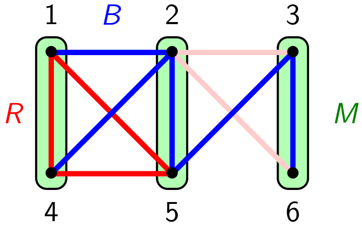
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.
3. Using the matching M find a red and a blue cycle in the K_n that cover almost all the vertices.

Regularity Partition



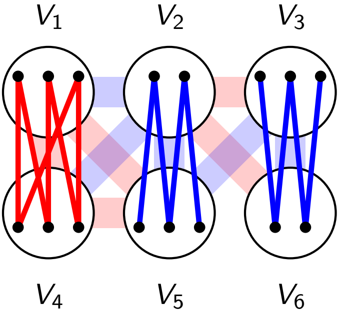
Reduced Graph



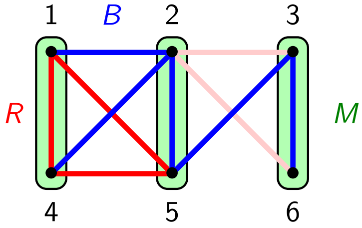
Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.
3. Using the matching M find a red and a blue cycle in the K_n that cover almost all the vertices.

Regularity Partition

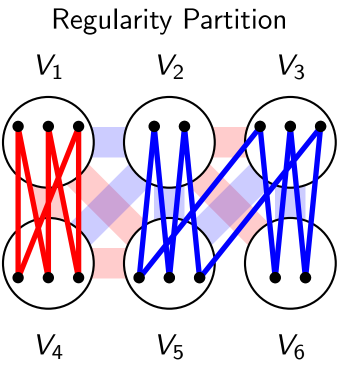


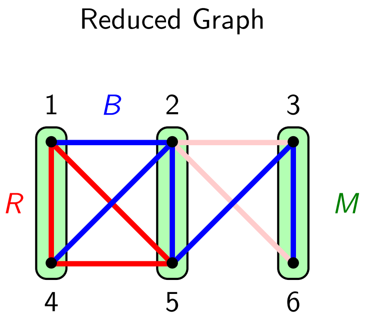
Reduced Graph



Łuczak's Connected Matching Method (Graph Version)

1. Apply the regularity lemma to the graph induced by the red edges.
2. In the reduced graph, pick a red component R and a blue component B such that $R \cup B$ contains a large matching M.
3. Using the matching M find a red and a blue cycle in the K_n that cover almost all the vertices.





Idea for the Proof in the 4-Uniform Case

Theorem (Lo, P 2020+)

Every red-blue edge-coloured K_n^4 contains a red and a blue tight cycle that together cover $(1 - o(1))n$ vertices.

Using Łuczak's idea we reduce the problem to proving the following.

Lemma

Every red-blue edge-coloured (almost) complete 4-graph H contains a red tight component[1] R and a blue tight component B such that $R \cup B$ contains a large matching.

Question

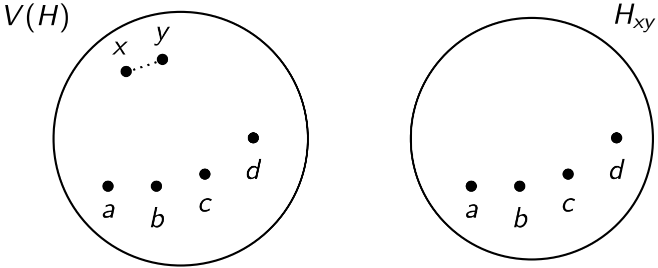
How do we choose the tight components R and B?

To do this we construct an auxiliary graph (the blueprint).

[1] A *tight component* is a set of edges F such that, for any $e, f \in F$, there exist $e_1, \dots, e_t \in F$ with $e_1 = e$, $e_t = f$, and $|e_i \cap e_{i+1}| = 3$.

Constructing the Blueprint

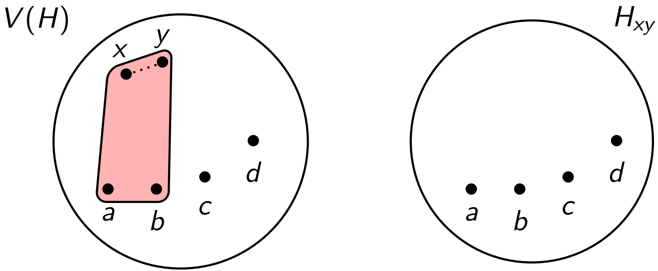
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

Constructing the Blueprint

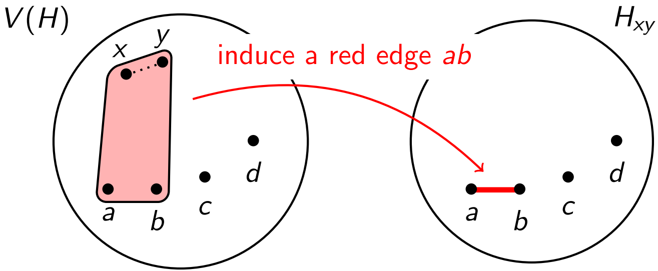
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

Constructing the Blueprint

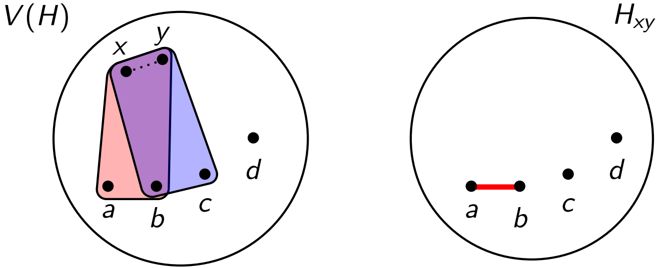
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

Constructing the Blueprint

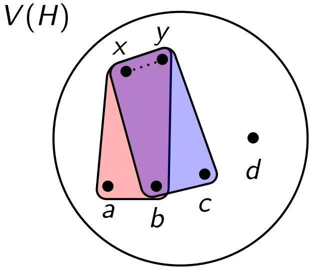
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

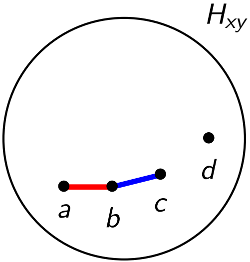


1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

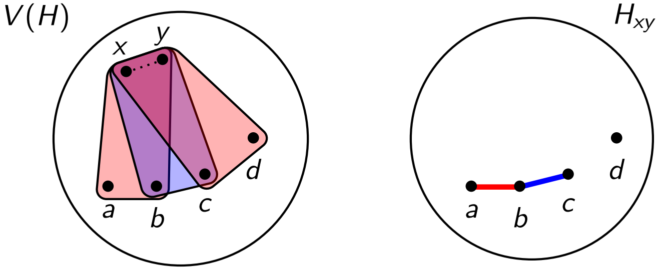




1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

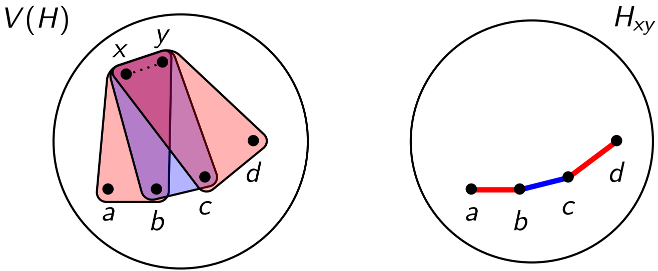
Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

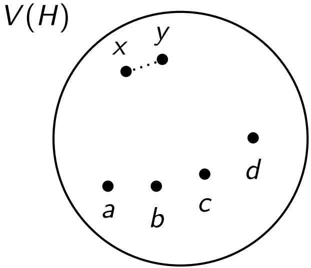
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

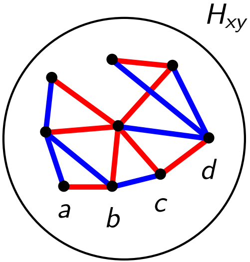


1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

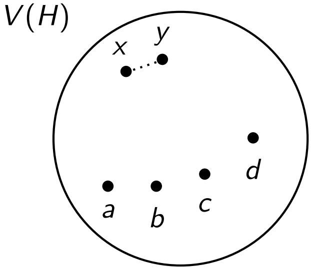


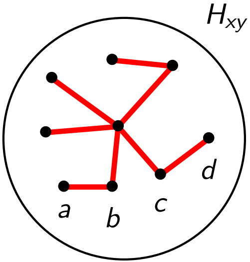


1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

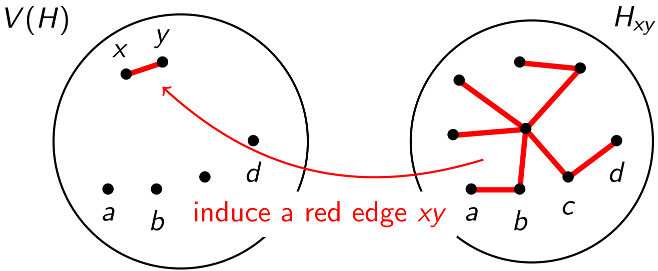




1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

Constructing the Blueprint

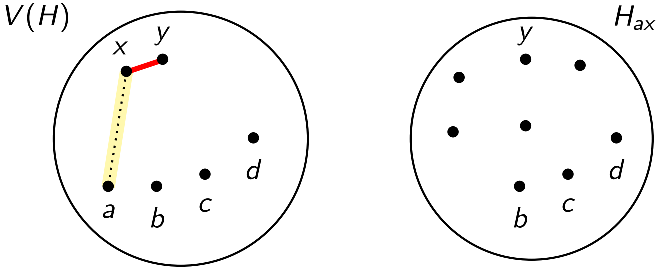
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

Constructing the Blueprint

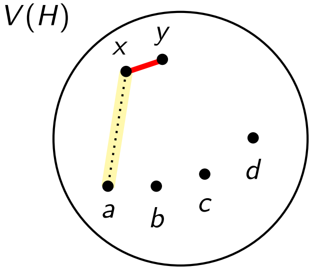
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

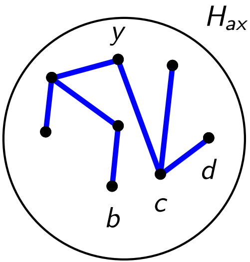


1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

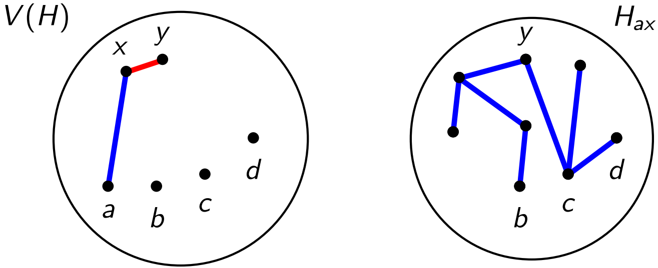




1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

Constructing the Blueprint

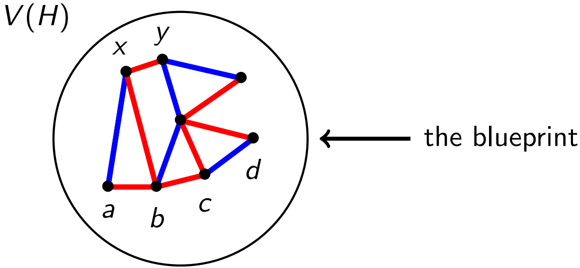
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



1. For each pair xy in $V(H)$, we consider the link graph H_{xy} that has an edge ab for each edge $abxy$ in H and the edge ab inherits the colour of the edge $abxy$.
2. Since H_{xy} is almost complete, it contains a large monochromatic component. We colour the edge xy in the blueprint accordingly. We colour all pairs of vertices in $V(H)$ in this way.

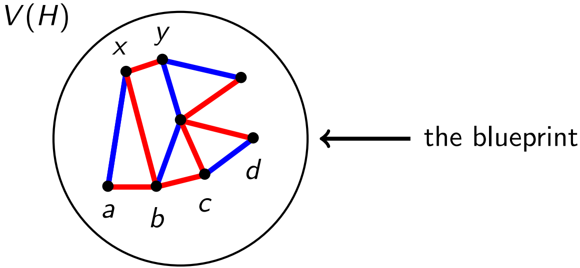
Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



Constructing the Blueprint

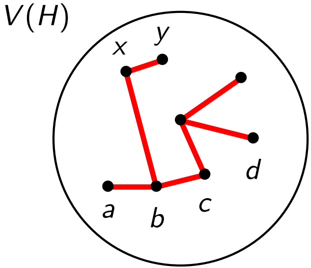
To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



3. After deleting some vertices and edges, components in the blueprint correspond to tight components in H.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.

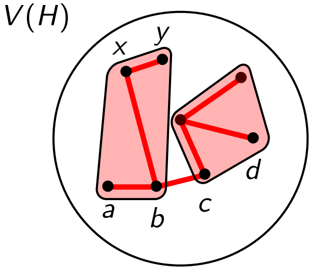


large monochromatic component in the blueprint

3. After deleting some vertices and edges, components in the blueprint correspond to tight components in H.
4. Since the blueprint is almost complete, it has a large monochromatic component. This component corresponds to a large monochromatic component in H which will be our initial choice for R or B.

Constructing the Blueprint

To find R and B we define an auxiliary red-blue edge-coloured graph (the blueprint) on the same vertex set as H.



matching in the red tight component R

5. We then find a matching in R. If that matching is not big enough, we use an additional argument to find a blue tight component B such that $R \cup B$ contains a large matching.

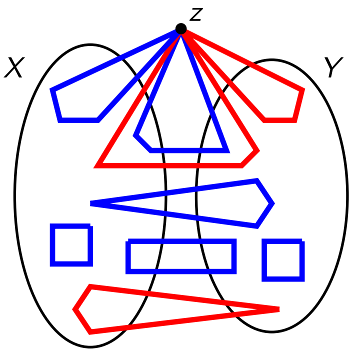
Open Question

Question

Can every red-blue edge-coloured K_n^4 be partitioned into two monochromatic tight cycles?

The following example shows that we need to allow the two tight cycles to possibly have the same colour.

$$|X| = 3m + 2$$



$$|Y| = m + 2$$

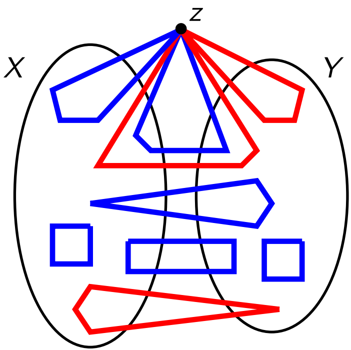
Open Question

Question

Can every red-blue edge-coloured K_n^4 be partitioned into two monochromatic tight cycles?

The following example shows that we need to allow the two tight cycles to possibly have the same colour.

$|X| = 3m + 2$



$|Y| = m + 2$

Thank you!