Edge-partitioning a graph into paths: beyond the Barát-Thomassen conjecture

Julien Bensmail, Ararat Harutyunyan and Stéphan Thomassé

LIP, ÉNS de Lyon, France

AGH University, Kraków
April 14th, 2015

Main problem

G: undirected simple graph.
T: tree with $|E(T)|$ dividing $|E(G)|$.

Main problem

G: undirected simple graph.
T: tree with $|E(T)|$ dividing $|E(G)|$.

Definition: *T-decomposition*

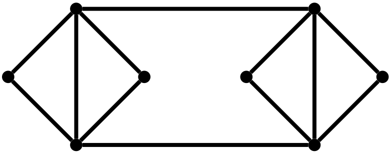
A *T-decomposition* of G is a partition $E_1, ..., E_k$ of $E(G)$ such that E_i induces an isomorphic copy of T for every $i = 1, ..., k$.

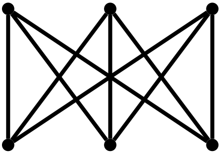
Main problem

G: undirected simple graph.
T: tree with $|E(T)|$ dividing $|E(G)|$.

Definition: *T-decomposition*

A *T-decomposition* of G is a partition $E_1, \ldots, E_k$ of $E(G)$ such that E_i induces an isomorphic copy of T for every $i = 1, \ldots, k$.



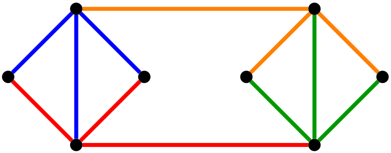


Main problem

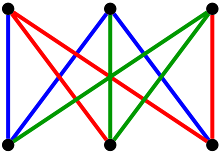
G: undirected simple graph.
T: tree with $|E(T)|$ dividing $|E(G)|$.

Definition: *T-decomposition*

A *T-decomposition* of G is a partition $E_1, ..., E_k$ of $E(G)$ such that E_i induces an isomorphic copy of T for every $i = 1, ..., k$.



S_4-decomposition



P_3-decomposition

The Barát-Thomassen conjecture

Divisibility condition is understood throughout.

Conjecture [Barát, Thomassen – 2006]

For every fixed tree T, there exists a positive constant c_T such that every c_T-edge-connected graph admits a T-decomposition.

The Barát-Thomassen conjecture

Divisibility condition is understood throughout.

Conjecture [Barát, Thomassen – 2006]

For every fixed tree T, there exists a positive constant c_T such that every c_T-edge-connected graph admits a T-decomposition.

Verified for T being:

- a star [Thomassen – 2012],
- a bistar of the form $S_{k,k+1}$ [Thomassen – 2014],
- the tree with degree sequence $(1, 1, 1, 2, 3)$ [Barát, Gerbner – 2014],
- of **diameter at most 4 [Merker – 2015+]**,
- among some family of trees with diameter 5 [Merker – 2015+],

and...

The Barát-Thomassen conjecture

Divisibility condition is understood throughout.

Conjecture [Barát, Thomassen – 2006]

For every fixed tree T, there exists a positive constant c_T such that every c_T-edge-connected graph admits a T-decomposition.

and...

- the path of length 3 [Thomassen – 2008],
- the path of length 4 [Thomassen – 2008],
- a **path of length 2^k [Thomassen – 2014]**,
- the path of length 5 [Botler, Mota, Oshiro, Wakabayashi – 2015],
- **any path [Botler, Mota, Oshiro, Wakabayashi – 2015+]**!

About the result for paths

Theorem [Botler, Mota, Oshiro, Wakabayashi – 2015+]

The Barát-Thomassen conjecture is true for T being any path.

About the proof:

- Generalization of a proof for P_5.

About the result for paths

Theorem [Botler, Mota, Oshiro, Wakabayashi – 2015+]

The Barát-Thomassen conjecture is true for T being any path.

About the proof:

- Generalization of a proof for P_5.
- Technical due to risky path-uncrossing procedures.

About the result for paths

Theorem [Botler, Mota, Oshiro, Wakabayashi – 2015+]

The Barát-Thomassen conjecture is true for T being any path.

About the proof:

- Generalization of a proof for P_5.
- Technical due to risky path-uncrossing procedures.

Our goal: give a somewhat simpler proof with *reasonable* technicalities.

A stronger result

'Stronger' $=$ **degree** is more important than **edge-connectivity**.

A stronger result

'Stronger' $=$ **degree** is more important than **edge-connectivity**.

Theorem [B., Harutyunyan, Thomassé – 2015+]

For every $\ell \geq 1$, every 64-edge-connected graph admits a P_ℓ-decomposition provided its minimum degree is large enough.

A stronger result

'Stronger' $=$ **degree** is more important than **edge-connectivity**.

Theorem [B., Harutyunyan, Thomassé – 2015+]

For every $\ell \geq 1$, every 64-edge-connected graph admits a P_ℓ-decomposition provided its minimum degree is large enough.

More general question

2-edge-connectivity $+$ large minimum degree $\Rightarrow$ path-decomposition???

Overview of the proof

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Overview of the proof

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?

Overview of the proof

Main ideas:

1. See G as a ‘convenient’ system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does ‘convenient’ mean?
2. What does it mean for H to be eulerian?

Overview of the proof

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?

Overview of the proof

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?
3. How to be sure that the trail is properly decomposable?

Overview of the proof

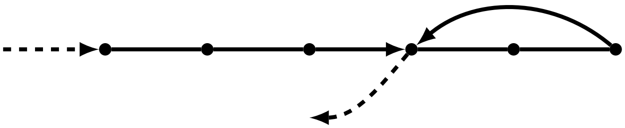
Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?
3. How to be sure that the trail is properly decomposable?

3. is a crucial concern in Botler, Mota, Oshiro and Wakabayashi's proof.



Overview of the proof

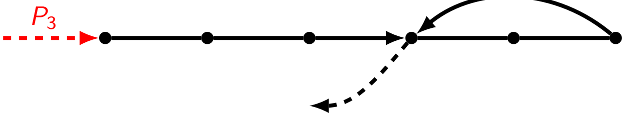
Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?
3. How to be sure that the trail is properly decomposable?

3. is a crucial concern in Botler, Mota, Oshiro and Wakabayashi's proof.



Overview of the proof

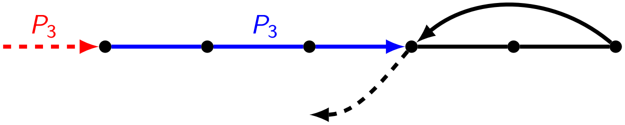
Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?
3. How to be sure that the trail is properly decomposable?

3. is a crucial concern in Botler, Mota, Oshiro and Wakabayashi's proof.



Overview of the proof

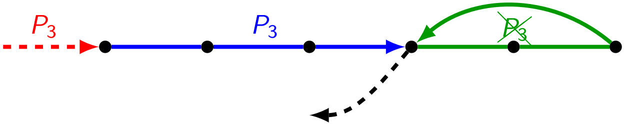
Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Issues:

1. What does 'convenient' mean?
2. What does it mean for H to be eulerian?
2. How to ensure connectedness of H (after 2., especially)?
3. How to be sure that the trail is properly decomposable?

3. is a crucial concern in Botler, Mota, Oshiro and Wakabayashi's proof.



Outline of the talk

Degree assumption $\Rightarrow$ source of degree.

Outline of the talk

Degree assumption $\Rightarrow$ source of degree.

Source of degree $+$ small edge-connectivity $\Rightarrow$ 'convenient' objects.

Outline of the talk

Degree assumption $\Rightarrow$ source of degree.

Source of degree $+$ small edge-connectivity $\Rightarrow$ 'convenient' objects.

- **Part 2**: *sparse* and *dense* subgraphs.

Outline of the talk

Degree assumption $\Rightarrow$ source of degree.

Source of degree + small edge-connectivity $\Rightarrow$ 'convenient' objects.

- **Part 2**: *sparse* and *dense* subgraphs.
- **Part 3**: constructing *systems of edge-disjoint paths*.

Outline of the talk

Degree assumption $\Rightarrow$ source of degree.

Source of degree $+$ small edge-connectivity $\Rightarrow$ 'convenient' objects.

- **Part 2**: *sparse* and *dense* subgraphs.
- **Part 3**: constructing *systems of edge-disjoint paths*.
- **Part 4**: obtaining such systems with a *tree structure*.

Part 1: Introduction to the problem

Part 2: Fractions of graphs

Part 3: Path-graphs

Part 4: Constructing path-trees

Part 5: Using everything together

Part 6: Conclusion

Sparse and dense subgraphs

α: real number in $[0, 1]$.

Definitions: α-*sparse*, α-*dense*, α-*fraction*

Let H be a spanning subgraph of G. We say that H is α-*sparse* (resp. α-*dense*) if $d_H(v) \leq \alpha d_G(v)$ (resp. $d_H(v) \geq \alpha d_G(v)$) for every $v \in V(G)$. We say that H is an α-*fraction* of G if H is both α-sparse and α-dense.

Sparse and dense subgraphs

α: real number in $[0, 1]$.

Definitions: α-*sparse*, α-*dense*, α-*fraction*

Let H be a spanning subgraph of G. We say that H is α-*sparse* (resp. α-*dense*) if $d_H(v) \leq \alpha d_G(v)$ (resp. $d_H(v) \geq \alpha d_G(v)$) for every $v \in V(G)$. We say that H is an α-*fraction* of G if H is both α-sparse and α-dense.

k-edge-connectivity + large degree $\Rightarrow$ $1/k$-sparse spanning tree.

Theorem [Ellingham, Nam, Voss – 2002]

Every k-edge-connected graph admits a $1/k$-sparse spanning tree.

(with error term $+2$)

On fractions of graphs

Proposition

Every graph G has a $1/2$-fraction (with error term ± 1).

Proof: If G has an even cycle C, remove the edges of C, apply induction and add the edges of a perfect matching of C to the solution. Otherwise, G is either an odd cycle (in which case the conclusion follows), or has a cutvertex z incident to an 'endblock' B which is either an edge or an odd cycle. Then contract B to z, apply induction, and extend the solution by conveniently choosing some edges of B. ■

On fractions of graphs

Proposition

Every graph G has a $1/2$-fraction (with error term ± 1).

Proof: If G has an even cycle C, remove the edges of C, apply induction and add the edges of a perfect matching of C to the solution. Otherwise, G is either an odd cycle (in which case the conclusion follows), or has a cutvertex z incident to an 'endblock' B which is either an edge or an odd cycle. Then contract B to z, apply induction, and extend the solution by conveniently choosing some edges of B. ∎

Corollary

If α has a finite binary extension, then G has an α-fraction.

(with constant additive error term)

Part 1: Introduction to the problem

Part 2: Fractions of graphs

Part 3: Path-graphs

Part 4: Constructing path-trees

Part 5: Using everything together

Part 6: Conclusion

Path-graphs

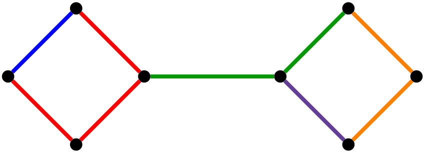
Definition: *path-graph*

A *path-graph* H on $G = (V, E)$ is a couple $(V, \mathcal{P})$ where $\mathcal{P}$ is a set of edge-disjoint paths of G. For every $v \in V$, we define $\mathcal{P}_v$ as the set of paths of $\mathcal{P}$ having v as an endvertex. To H, we associate the (multi)graph H^* on vertex set V and edge set contains the pairs of endvertices of $\mathcal{P}$.

Path-graphs

Definition: *path-graph*

A *path-graph* H on $G = (V, E)$ is a couple $(V, \mathcal{P})$ where $\mathcal{P}$ is a set of edge-disjoint paths of G. For every $v \in V$, we define $\mathcal{P}_v$ as the set of paths of $\mathcal{P}$ having v as an endvertex. To H, we associate the (multi)graph H^* on vertex set V and edge set contains the pairs of endvertices of $\mathcal{P}$.

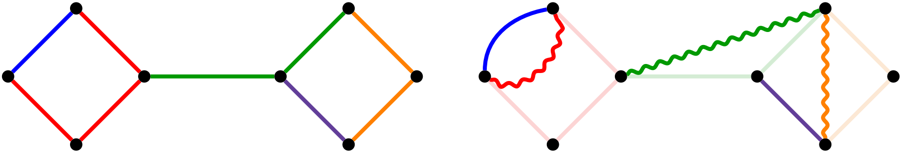


H on G

Path-graphs

Definition: *path-graph*

A *path-graph* H on $G = (V, E)$ is a couple $(V, \mathcal{P})$ where $\mathcal{P}$ is a set of edge-disjoint paths of G. For every $v \in V$, we define $\mathcal{P}_v$ as the set of paths of $\mathcal{P}$ having v as an endvertex. To H, we associate the (multi)graph H^* on vertex set V and edge set contains the pairs of endvertices of $\mathcal{P}$.



H on G $\qquad$ $H*$

Properties of path-graphs

- H **connected** $\Leftrightarrow$ H^* connected.
- H **tree** $\Leftrightarrow$ H^* tree.

Properties of path-graphs

- H **connected** $\Leftrightarrow$ H^* connected.
- H **tree** $\Leftrightarrow$ H^* tree.
- For every $v \in V$, $d_H(v) = d_{H^*}(v)$.
- H **eulerian** $\Leftrightarrow$ H connected + $d_H(v)$ even for every $v \in V$.

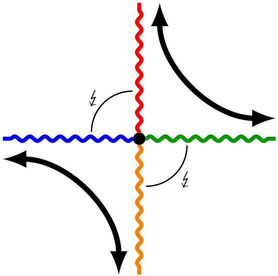
Properties of path-graphs

- H **connected** $\Leftrightarrow$ H^* connected.
- H **tree** $\Leftrightarrow$ H^* tree.
- For every $v \in V$, $d_H(v) = d_{H^*}(v)$.
- H **eulerian** $\Leftrightarrow$ H connected $+$ $d_H(v)$ even for every $v \in V$.
- H **q-path-graph** $\Leftrightarrow$ all paths of $\mathcal{P}$ have length q.
- H ($\leq$ **q**)**-path-graph** $\Leftrightarrow$ all paths of $\mathcal{P}$ have length at most q.
- H ($\geq$ **q**)**-path-graph** $\Leftrightarrow$ all paths of $\mathcal{P}$ have length at least q.
- H $(\mathbf{q_1}, \mathbf{q_2})$**-path-graph** $\Leftrightarrow$ all paths of $\mathcal{P}$ have length q_1 or q_2.

Previous example: disconnected ($\leq$ 3)-path-graph.

Conflicting paths

- **Conflicting paths** ⇔ paths sharing more than just one end.
- **Conflictless trail** ⇔ trail with no subsequent conflicting paths.
- *H* **conflictless eulerian** ⇔ *H* has a conflictless eulerian closed trail.



More terminology for conflicts

Definition: *multiplicity*

For distinct $w, v \in V$, the *multiplicity* of w *around* v is

$$\mathrm{mult}_v(w) := |\{P \in \mathcal{P}_v : w \in P\}|/|\mathcal{P}_v|.$$

The *multiplicity* of H is the maximum multiplicity of its vertices.

More terminology for conflicts

Definition: *multiplicity*

For distinct $w, v \in V$, the *multiplicity* of w *around* v is

$$\mathrm{mult}_v(w) := |\{P \in \mathcal{P}_v : w \in P\}|/|\mathcal{P}_v|.$$

The *multiplicity* of H is the maximum multiplicity of its vertices.

Definitions: *conflict graph*, *conflict ratio*

For every $v \in V$, the *conflict graph* H_v is the graph on vertex set $\mathcal{P}_v$ in which P_1P_2 is an edge if P_1 and P_2 intersect. The *conflict ratio* of H is defined as

$$\max\{(\Delta(H_v) + 1)/|\mathcal{P}_v| : v \in V\}.$$

Remark: Every path is self-conflicting.

Eulerian closed trails and conflict ratio

Eulerian path-graph + reasonable conflict ratio $\Rightarrow$ conflictless eulerian closed trail.

Theorem

Every eulerian path-graph H with conflict ratio at most $1/8$ has a conflictless eulerian closed trail.

Eulerian closed trails and conflict ratio

Eulerian path-graph + reasonable conflict ratio ⇒ conflictless eulerian closed trail.

Theorem

Every eulerian path-graph H with conflict ratio at most $1/8$ has a conflictless eulerian closed trail.

Proof: Since the antidegree of every vertex in H_v is greater than $|\mathcal{P}_v|/2$, necessarily H_v admits a hamiltonian anticycle (by Dirac's Theorem). So there is a pairing $M_v = P_1P_2, P_3P_4, \ldots$ of the paths in $\mathcal{P}_v$ such that each pair is non-conflicting.

Eulerian closed trails and conflict ratio

Eulerian path-graph + reasonable conflict ratio $\Rightarrow$ conflictless eulerian closed trail.

Theorem

Every eulerian path-graph H with conflict ratio at most $1/8$ has a conflictless eulerian closed trail.

Proof: Since the antidegree of every vertex in H_v is greater than $|\mathcal{P}_v|/2$, necessarily H_v admits a hamiltonian anticycle (by Dirac's Theorem). So there is a pairing $M_v = P_1P_2, P_3P_4, \ldots$ of the paths in $\mathcal{P}_v$ such that each pair is non-conflicting.

Having such a pairing M_v for every $v \in V$ defines a set of conflictless closed trails $T_1, \ldots, T_t$, where a pair $\{P_i, P_{i+1}\}$ means that when entering at a vertex via P_i, we must leave via P_{i+1} (and vice-versa). If $t = 1$, we are done. Otherwise, we merge two trails so that t decreases.

Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■

Merging two conflictless closed trails

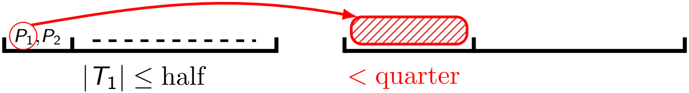
By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■

Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

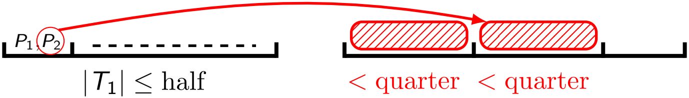
Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■



Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

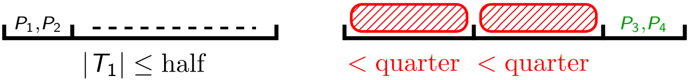
Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■



Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

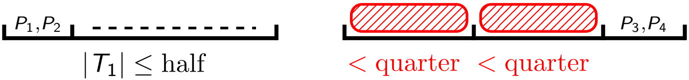
Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■



Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

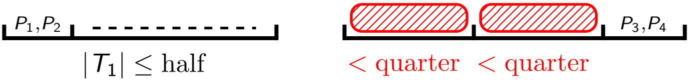
Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■

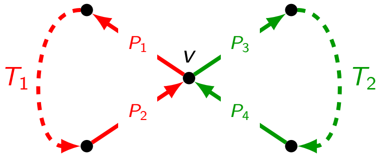


Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ∎

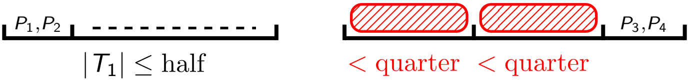


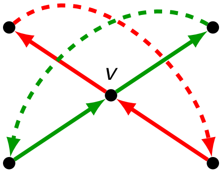


Merging two conflictless closed trails

By our terminology, there is a $v \in V$ whose some paths of $\mathcal{P}_v$ belong to, say, T_1. We may assume that no more than half of its paths appear in T_1.

Assume $\{P_1, P_2\} \in M_v$ and $P_1, P_2 \in T_1$. Because P_1 and P_2 have degree less than $|\mathcal{P}_v|/8$ in H_v, there is a pair $\{P_3, P_4\}$ not in T_1 and such that P_1, P_2, P_3, P_4 are non-conflicting. Then replace $\{P_1, P_2\}$ and $\{P_3, P_4\}$ in M_v by $\{P_1, P_4\}$ and $\{P_2, P_3\}$. Then t decreases by 1, as claimed. ■





To larger paths with reasonable conflicts

Growing paths with still ‘reasonable’ path conflicts?

Lemma

Let q be some fixed positive integer and $c > 0$ be some real number such that $cq < 1/100$. Let $H = (V, \mathcal{P})$ be an α-dense q-path-graph of some graph G with multiplicity at most c and minimum degree k large with respect to $1/c$ and q. Then one can form an $\alpha/5$-dense $2q$-path-graph on G with multiplicity at most $16cq$.

To larger paths with reasonable conflicts

Growing paths with still 'reasonable' path conflicts?

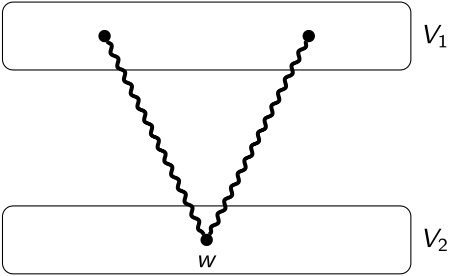
Lemma

Let q be some fixed positive integer and $c > 0$ be some real number such that $cq < 1/100$. Let $H = (V, \mathcal{P})$ be an α-dense q-path-graph of some graph G with multiplicity at most c and minimum degree k large with respect to $1/c$ and q. Then one can form an $\alpha/5$-dense $2q$-path-graph on G with multiplicity at most $16cq$.

Proof idea: Consider a cut (V_1, V_2) of V maximizing the size of the set $\mathcal{P}'$ of edges of H^* 'between' V_1 and V_2. Let $H' = (V, \mathcal{P}')$. Then H' is $\alpha/2$-dense and has multiplicity at most $2c$. Now split H' into two $1/2$-fractions $H'_1 = (V, \mathcal{P}'_1)$ and $H'_2 = (V, \mathcal{P}'_2)$. These two path-graphs are $\alpha/4$-dense and have multiplicity at most $4c$. We use H'_1 only to form a $2q$-path graph on V_1 with required density (almost automatic) and multiplicity (much harder).

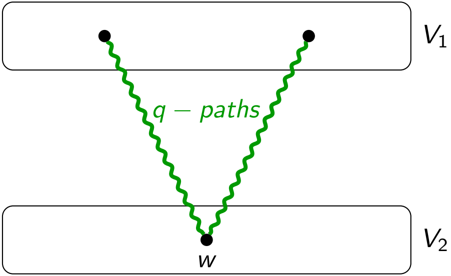
Overview of the proof

We iteratively want to randomly concatenate two q-paths meeting in V_2 at some vertex w to connect two vertices V_1 via a $2q$-paths.



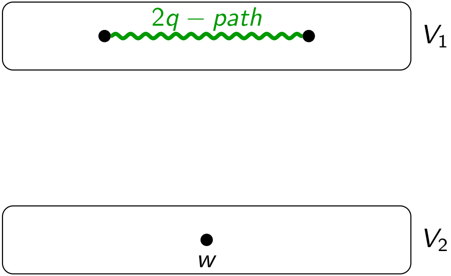
Overview of the proof

We iteratively want to randomly concatenate two q-paths meeting in V_2 at some vertex w to connect two vertices V_1 via a $2q$-paths.



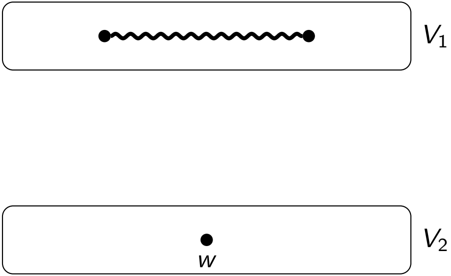
Overview of the proof

We iteratively want to randomly concatenate two q-paths meeting in V_2 at some vertex w to connect two vertices V_1 via a $2q$-paths.



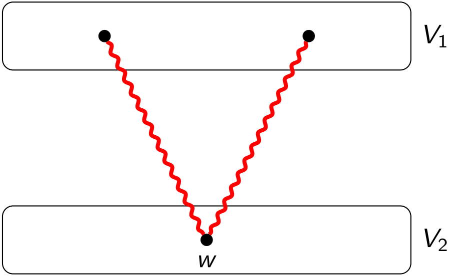
Overview of the proof

We iteratively want to randomly concatenate two q-paths meeting in V_2 at some vertex w to connect two vertices V_1 via a $2q$-paths.



Overview of the proof

We iteratively want to randomly concatenate two q-paths meeting in V_2 at some vertex w to connect two vertices V_1 via a $2q$-paths.



Problem: The two paths may be conflicting, and w can have degree so large that it carries too many path dependencies (making impossible *e.g.* the application of Lovász Local Lemma).

Pairing non-conflicting q-paths

Solution: Group the q-paths arriving at w into small subsets of non-conflicting paths $\rightarrow$ Possible because of the multiplicity assumption.

Generalization of Hajnal-Szemerédi Theorem

Let G be some graph of order n. Then, for every integer $t \geq \Delta(G) + 1$, the set $V(G)$ can be partitioned into $V_1, ..., V_t$ such that each V_i is an independent set of size $\lfloor \frac{n}{t} \rfloor$ or $\lceil \frac{n}{t} \rceil$.

Pairing non-conflicting q-paths

Solution: Group the q-paths arriving at w into small subsets of non-conflicting paths $\rightarrow$ Possible because of the multiplicity assumption.

Generalization of Hajnal-Szemerédi Theorem

Let G be some graph of order n. Then, for every integer $t \geq \Delta(G) + 1$, the set $V(G)$ can be partitioned into $V_1, ..., V_t$ such that each V_i is an independent set of size $\lfloor \frac{n}{t} \rfloor$ or $\lceil \frac{n}{t} \rceil$.

Now randomly pairing the q-paths arriving at every vertex w of V_2, we get a $2q$-path graph H_1'' spanning V_1. Multiplicity around every vertex v of V_1 is shown to be smaller than $16cq$ by combining LLL and Chernoff's bound.

Using H_2' instead H_1', we also obtain a $2q$-path graph H_2'' spanning V_2. ■

Constructing larger paths with reasonable conflicts

From repeated applications, we get:

Theorem

Let p be some integer and $0 < c < 1$ be some real number. There is an integer k depending on p and c such that every graph G with minimum degree at least k admits a $1/5^p$-dense 2^p-path graph H with multiplicity at most c.

Part 1: Introduction to the problem

Part 2: Fractions of graphs

Part 3: Path-graphs

Part 4: Constructing path-trees

Part 5: Using everything together

Part 6: Conclusion

Constructing $(1, \ell)$-trees

2-edge-connectivity + large degree $\Rightarrow$ $(1, \ell)$-(path-)tree with 'small' degree.

Theorem

Every 2-edge-connected graph G admits a subcubic spanning $(1, 2)$-tree.

Constructing $(1, \ell)$-trees

2-edge-connectivity + large degree $\Rightarrow$ $(1, \ell)$-(path-)tree with 'small' degree.

Theorem

Every 2-edge-connected graph G admits a subcubic spanning $(1, 2)$-tree.

Proof idea: Assume G is minimal and perform a DFS from some vertex. This defines some *forward edges*. The other edges of G are *backward edges*.

Constructing $(1, \ell)$-trees

2-edge-connectivity + large degree $\Rightarrow$ $(1, \ell)$-(path-)tree with 'small' degree.

Theorem

Every 2-edge-connected graph G admits a subcubic spanning $(1, 2)$-tree.

Proof idea: Assume G is minimal and perform a DFS from some vertex. This defines some *forward edges*. The other edges of G are *backward edges*.

To every vertex v of G, we initially associate the empty $(1, 2)$-tree X on $\{v\}$. The procedure mainly consists in iteratively considering vertices at the highest depth, and 'merging' their corresponding $(1, 2)$-trees somehow. This is done with preserving 2-edge-connectivity.

Constructing $(1, \ell)$-trees

2-edge-connectivity + large degree $\Rightarrow$ $(1, \ell)$-(path-)tree with 'small' degree.

Theorem

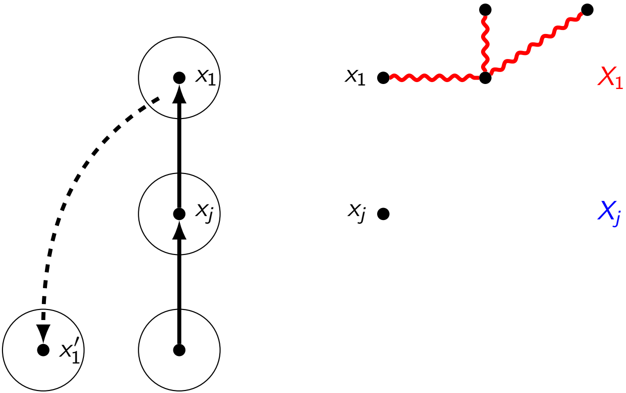
Every 2-edge-connected graph G admits a subcubic spanning $(1, 2)$-tree.

Proof idea: Assume G is minimal and perform a DFS from some vertex. This defines some *forward edges*. The other edges of G are *backward edges*.

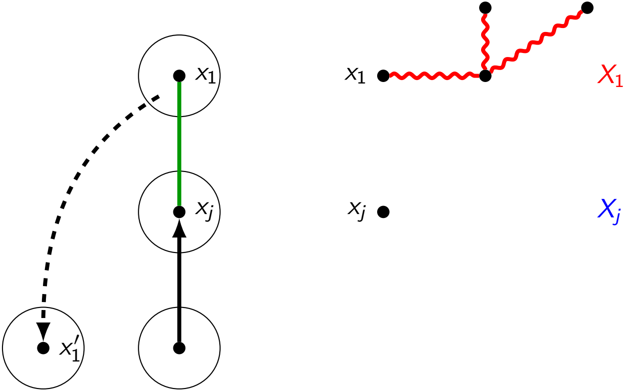
To every vertex v of G, we initially associate the empty $(1, 2)$-tree X on $\{v\}$. The procedure mainly consists in iteratively considering vertices at the highest depth, and 'merging' their corresponding $(1, 2)$-trees somehow. This is done with preserving 2-edge-connectivity.

The key to respect the degree condition is that there are only two possibilities for increasing the degree of a vertex in a merged $(1, 2)$-tree, namely by adding its incident forward and backward edges. ■

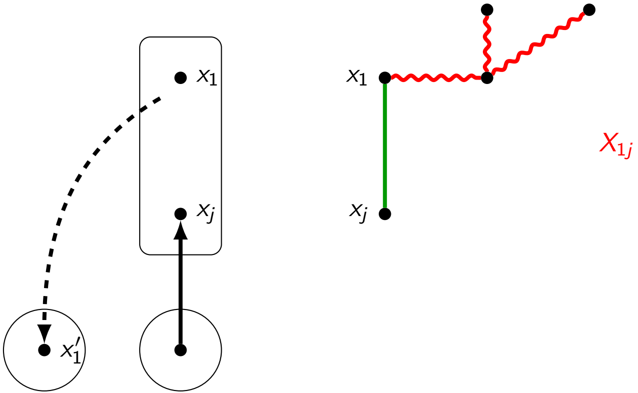
Sample cases – One child



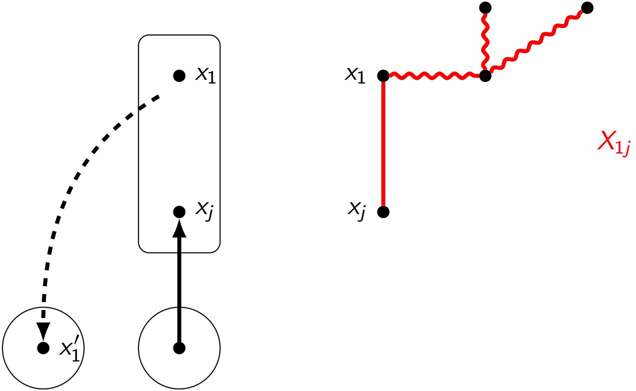
Sample cases – One child



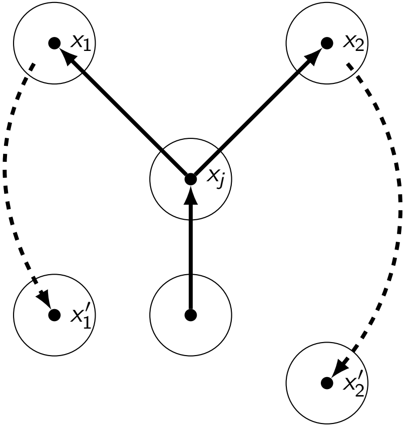
Sample cases – One child

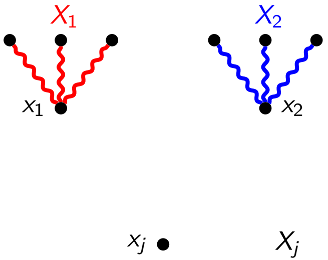


Sample cases – One child

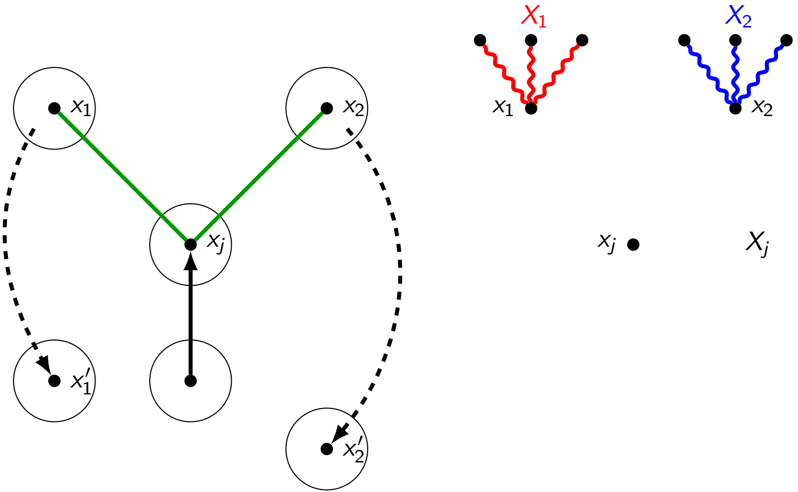


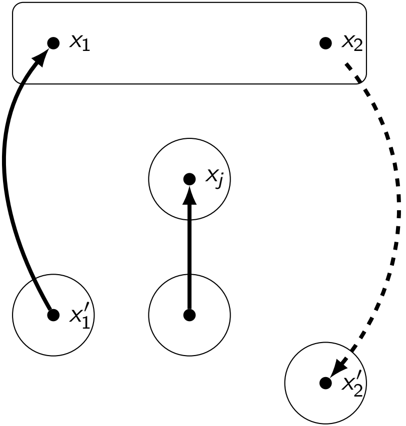
Sample cases – Two children

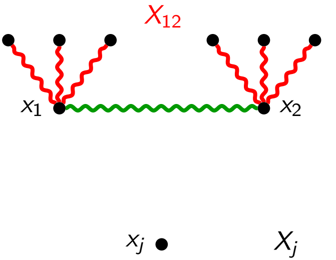




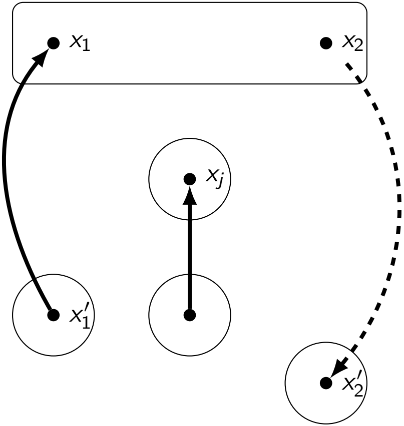
Sample cases – Two children

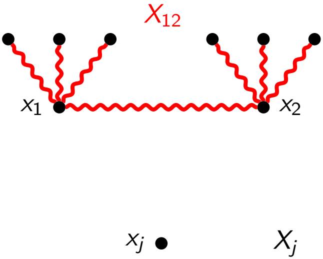



x_1
x_2
x_j
x'_1
x'_2


X_{12}
x_1
x_2
x_j
X_j

Sample cases – Two children





From $(1,2)$-trees to $(1,k)$-trees

spanning $(1,k)$-tree + disjoint 'source' of degree = spanning $(1,k+1)$-tree.

Theorem

Let T be a spanning $(1,k)$-tree of some graph $G=(V,E)$, and let H be some additional graph on V, edge-disjoint from G, and satisfying $d_H(v) \geq 2(d_T(v)+2k)$ for every $v \in V$. Then $G \cup H$ is spanned by a $(1,k+1)$-tree T'.

From $(1,2)$-trees to $(1,k)$-trees

spanning $(1,k)$-tree $+$ disjoint 'source' of degree $=$ spanning $(1,k+1)$-tree.

Theorem

Let T be a spanning $(1,k)$-tree of some graph $G=(V,E)$, and let H be some additional graph on V, edge-disjoint from G, and satisfying $d_H(v) \geq 2(d_T(v)+2k)$ for every $v \in V$. Then $G \cup H$ is spanned by a $(1,k+1)$-tree T'.

Proof idea: Same kind of proof. Start from the leaves of T and iteratively concatenate incident k-paths of T with some edges of H in order to form $(k+1)$-paths. This is always possible by the assumption on the degrees in H.

From $(1, 2)$-trees to $(1, k)$-trees

spanning $(1, k)$-tree + disjoint 'source' of degree = spanning $(1, k+1)$-tree.

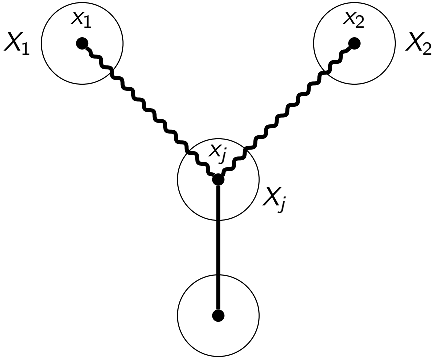
Theorem

Let T be a spanning $(1, k)$-tree of some graph $G = (V, E)$, and let H be some additional graph on V, edge-disjoint from G, and satisfying $d_H(v) \geq 2(d_T(v) + 2k)$ for every $v \in V$. Then $G \cup H$ is spanned by a $(1, k+1)$-tree T'.

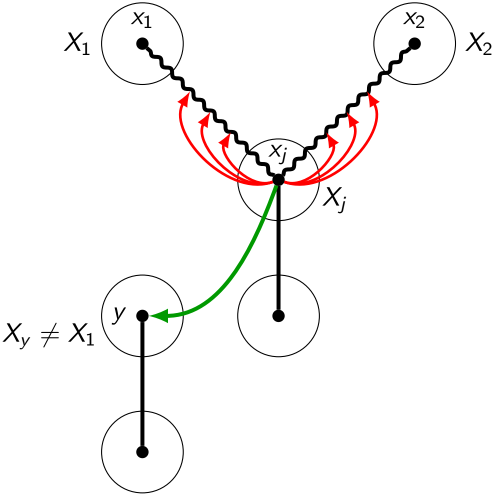
Proof idea: Same kind of proof. Start from the leaves of T and iteratively concatenate incident k-paths of T with some edges of H in order to form $(k+1)$-paths. This is always possible by the assumption on the degrees in H.

To make sure that the edges of H are equitably used and not 'saturated' by some vertex, we orient them in a balanced way beforehand (hence defining *private edges* for every vertex). ■

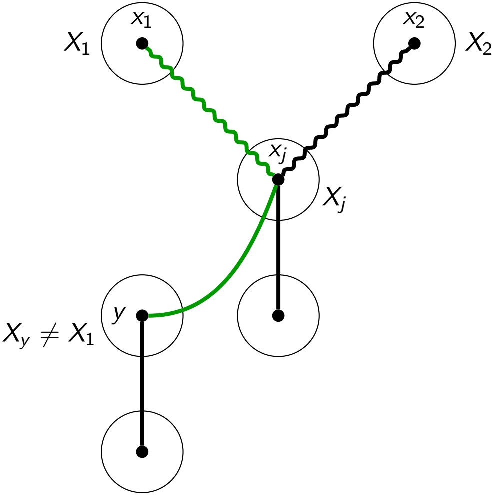
Sample case – Two children



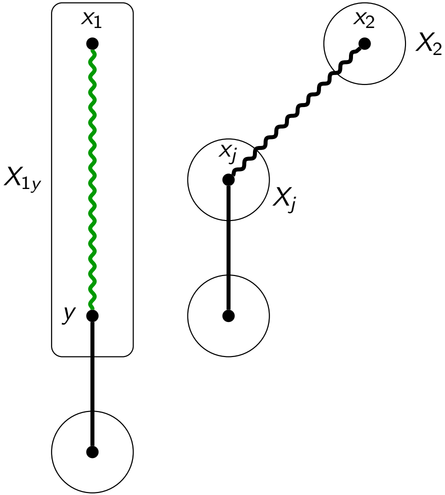
Sample case – Two children



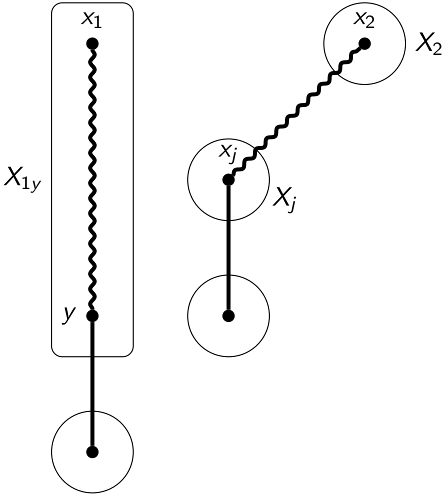
Sample case – Two children



Sample case – Two children



Sample case – Two children



$(1, k)$-trees – Summary

2-edge-connectivity + disjoint 'source' of degree = spanning $(1, \ell)$-tree.

Corollary

For every $\ell \geq 1$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected graph and H is some additional graph on V with minimum degree k_ℓ, then $G \cup H$ is spanned by a $(1, \ell + 1)$-tree T where $d_T(v) \leq d_H(v)$ for every $v \in V$.

$(1, k)$-trees – Summary

2-edge-connectivity $+$ disjoint 'source' of degree $=$ spanning $(1, \ell)$-tree.

Corollary

For every $\ell \geq 1$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected graph and H is some additional graph on V with minimum degree k_ℓ, then $G \cup H$ is spanned by a $(1, \ell + 1)$-tree T where $d_T(v) \leq d_H(v)$ for every $v \in V$.

Proof: First deduce a subcubic $(1, 2)$-tree T_2 spanning G. Then consider a sequence of disjoint small fraction $H_1, ..., H_{\ell-1}$ of H, where each H_i is an ε_i-fraction of H. By the assumption on k_ℓ, we can assume $\varepsilon_{i+1} \geq 4\varepsilon_i$.

$(1, k)$-trees – Summary

2-edge-connectivity + disjoint 'source' of degree = spanning $(1, \ell)$-tree.

Corollary

For every $\ell \geq 1$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected graph and H is some additional graph on V with minimum degree k_ℓ, then $G \cup H$ is spanned by a $(1, \ell + 1)$-tree T where $d_T(v) \leq d_H(v)$ for every $v \in V$.

Proof: First deduce a subcubic $(1, 2)$-tree T_2 spanning G. Then consider a sequence of disjoint small fraction $H_1, \ldots, H_{\ell-1}$ of H, where each H_i is an ε_i-fraction of H. By the assumption on k_ℓ, we can assume $\varepsilon_{i+1} \geq 4\varepsilon_i$.

Using H_1, from T_2 we can deduce a $(1, 3)$-tree T_3 spanning G. Note that $d_{T_3}(v) \leq d_{T_2}(v) + d_{H_1}(v)$ for every vertex $v \in V$. Since $4d_{H_1}(v) \leq d_{H_2}(v)$, we can use H_2 to extend T_3 to a $(1, 4)$-tree T_4 spanning G. Due to the choice of the ε_i's, this process can be repeated until we get T. ■

Path-trees in bipartite graphs

Path-trees with paths of lengths multiple of ℓ?

Theorem

For every even $\ell \geq 2$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected bipartite graph with vertex partition (A, B) and H is some additional bipartite graph with vertex bipartition (A, B) and minimum degree k_ℓ, then $G \cup H$ admits a $(\ell, 2\ell)$-tree T spanning A where $d_T(v) \leq d_H(v)$ for every $v \in V$.

Path-trees in bipartite graphs

Path-trees with paths of lengths multiple of ℓ?

Theorem

For every even $\ell \geq 2$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected bipartite graph with vertex partition (A, B) and H is some additional bipartite graph with vertex bipartition (A, B) and minimum degree k_ℓ, then $G \cup H$ admits a $(\ell, 2\ell)$-tree T spanning A where $d_T(v) \leq d_H(v)$ for every $v \in V$.

Proof idea: Using a tiny ε-fraction of H, we can obtain a $(1, \ell + 1)$-tree T' spanning G verifying $d_{T'}(v) \leq \varepsilon d_H(v)$ for every $v \in V$.

Path-trees in bipartite graphs

Path-trees with paths of lengths multiple of ℓ?

Theorem

For every even $\ell \geq 2$, there exists k_ℓ such that if $G = (V, E)$ is a 2-edge-connected bipartite graph with vertex partition (A, B) and H is some additional bipartite graph with vertex bipartition (A, B) and minimum degree k_ℓ, then $G \cup H$ admits a $(\ell, 2\ell)$-tree T spanning A where $d_T(v) \leq d_H(v)$ for every $v \in V$.

Proof idea: Using a tiny ε-fraction of H, we can obtain a $(1, \ell+1)$-tree T' spanning G verifying $d_{T'}(v) \leq \varepsilon d_H(v)$ for every $v \in V$.

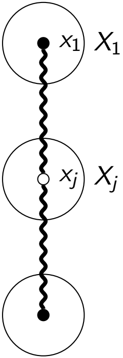
We may assume $4\varepsilon \leq 1/5^p$, where $p = \lceil \log_2 \ell \rceil$. Also, we can deduce a $1/5^p$-dense 2^p-path-graph H' on H with multiplicity at most $1/16.2^p$. Orienting H'^* in a balanced way, every vertex v is the origin of at least $d_H(v)/2.5^p$ private 2^p-paths of H' (with multiplicity at most $1/8.2^p$). Furthermore, $2^p \geq \ell - 1$.

Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell - 1)$-path of x. This forms an ℓ or 2ℓ-path. ■

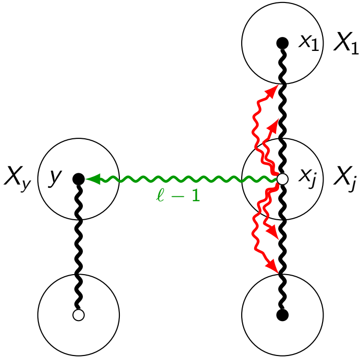
Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell - 1)$-path of x. This forms an ℓ or 2ℓ-path. ■



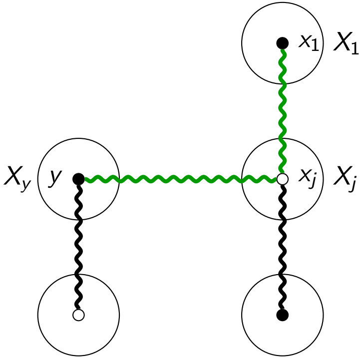
Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell-1)$-path of x. This forms an ℓ or 2ℓ-path. ■



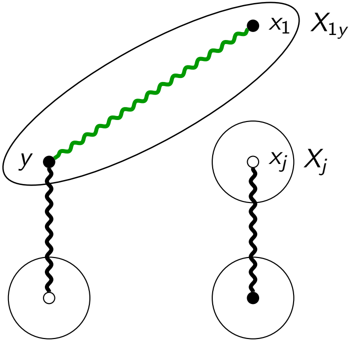
Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell - 1)$-path of x. This forms an ℓ or 2ℓ-path. ∎



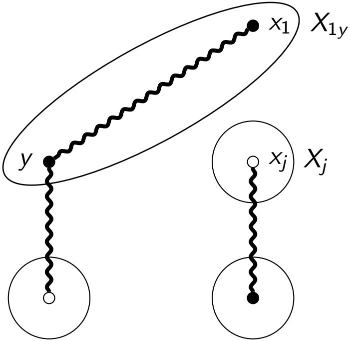
Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell - 1)$-path of x. This forms an ℓ or 2ℓ-path. ■



Path-trees in bipartite graphs

To obtain T, we once again start from every vertex v of A being its own $(\ell, 2\ell)$-tree X_v. We then iteratively consider a leaf u in T' with parent x, and concatenate the path associated to ux with some private $(\ell-1)$-path of x. This forms an ℓ or 2ℓ-path. ■



Overview, second try

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Overview, second try

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Refined to...

1. Turn G into a $(\geq \ell)$-path graph H.

Overview, second try

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Refined to...

1. Turn G into a $(\geq \ell)$-path graph H.
2. Make H conflictless eulerian by removing some P_ℓ's.

Overview, second try

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Refined to...

1. Turn G into a $(\geq \ell)$-path graph H.
2. Make H conflictless eulerian by removing some P_ℓ's.
3. Achieve the decomposition by following the conflictless eulerian trail.

Overview, second try

Main ideas:

1. See G as a 'convenient' system H of induced paths.
2. Remove some P_ℓ's from G so that H is eulerian.
3. Finish the decomposition by following a eulerian trail of H.

Refined to...

1. Turn G into a $(\geq \ell)$-path graph H.
2. Make H conflictless eulerian by removing some P_ℓ's.
3. Achieve the decomposition by following the conflictless eulerian trail.

Remark: We may suppose ℓ is even.

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

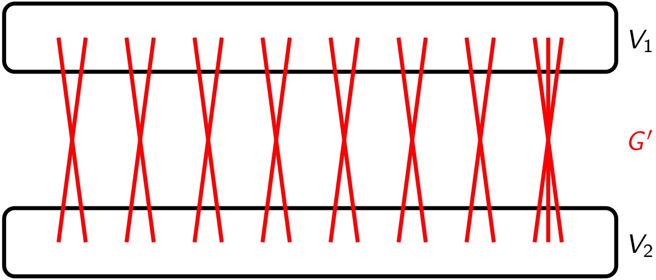
Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Set $G' := (V, F)$.

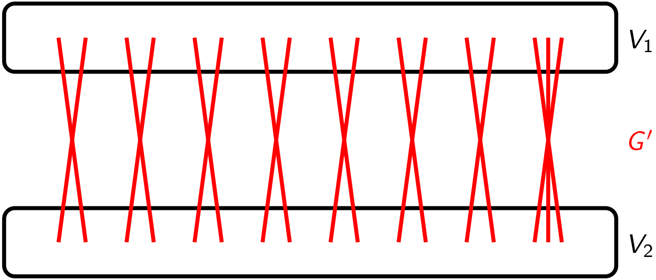


Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Set $G' := (V, F)$.



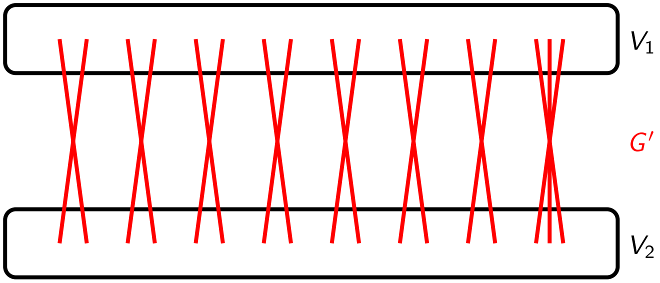
Max cut $\Rightarrow$ G' is 32-edge-connected and $\delta(G') \gg \ell$.

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Set $G' := (V, F)$.



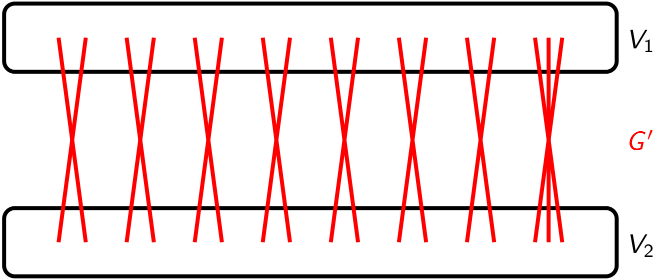
Max cut $\Rightarrow$ G' is 32-edge-connected and $\delta(G') \gg \ell$.
$\Rightarrow$ there exist 16 edge-disjoint spanning trees of G' [Tutte – 1965].

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Set $G' := (V, F)$.



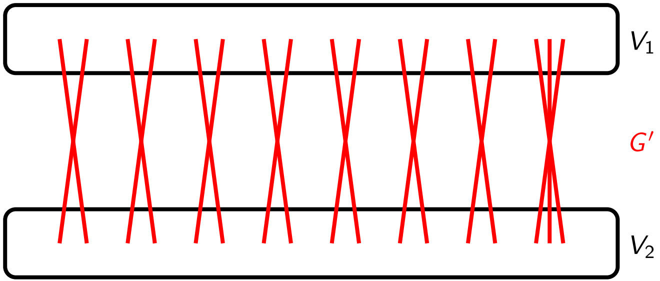
Max cut $\Rightarrow$ G' is 32-edge-connected and $\delta(G') \gg \ell$.
$\Rightarrow$ there exist 16 edge-disjoint spanning trees of G' [Tutte – 1965].
$\Rightarrow$ $G' = G'_1, \ldots, G'_8$ (2-edge-connected).

Setting up

G: 64-edge-connected, $\delta(G) \gg \ell$.

Start from a maximum cut (V_1, V_2) of G, and let F be the edges across the cut.

Set $G' := (V, F)$.



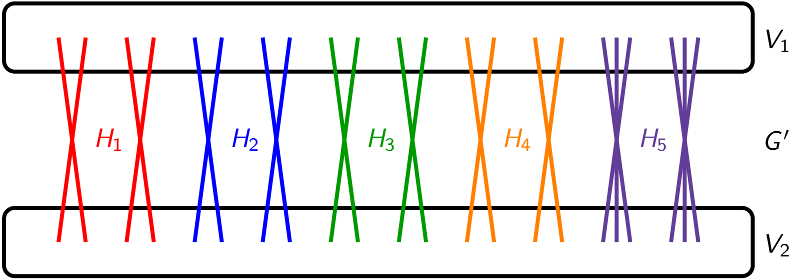
Max cut $\Rightarrow$ G' is 32-edge-connected and $\delta(G') \gg \ell$.
$\Rightarrow$ there exist 16 edge-disjoint spanning trees of G' [Tutte – 1965].
$\Rightarrow$ $G' = G'_1, \ldots, G'_8$ (2-edge-connected).
$\Rightarrow$ there exist $T_1, \ldots, T_8$ 1/2-sparse spanning trees.

Setting up

Set $H_1 := T_1 \cup T_2$, $H_2 := T_3 \cup T_4$, $H_3 := T_5 \cup T_6$, $H_4 := T_7 \cup T_8$, and H_5 the rest.

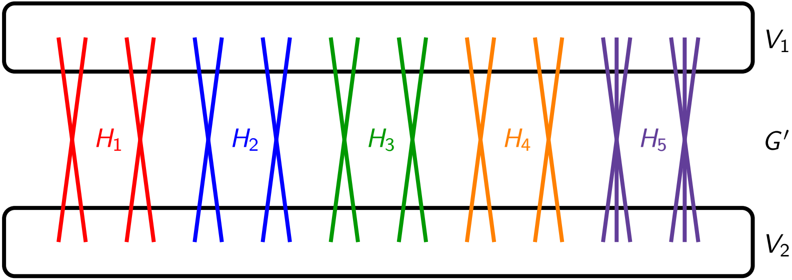
Setting up

Set $H_1 := T_1 \cup T_2$, $H_2 := T_3 \cup T_4$, $H_3 := T_5 \cup T_6$, $H_4 := T_7 \cup T_8$, and H_5 the rest.



Setting up

Set $H_1 := T_1 \cup T_2$, $H_2 := T_3 \cup T_4$, $H_3 := T_5 \cup T_6$, $H_4 := T_7 \cup T_8$, and H_5 the rest.



$$G' = \underbrace{H_1, H_2, H_3, H_4}_{} + \underbrace{H_5}_{}$$

- connected
- bridgeless
- 1/2-sparse

- 1/2-dense

Creating $(\ell, 2\ell)$-trees over V_1 and V_2

H_1 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_1' spanning V_1.
H_2 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_2' spanning V_1.

Creating $(\ell, 2\ell)$-trees over V_1 and V_2

H_1 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_1' spanning V_1.
H_2 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_2' spanning V_1.

H_3 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_3' spanning V_2.
H_4 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_4' spanning V_2.

Creating $(\ell, 2\ell)$-trees over V_1 and V_2

H_1 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_1' spanning V_1.
H_2 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_2' spanning V_1.

H_3 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_3' spanning V_2.
H_4 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_4' spanning V_2.

Because $\delta(H_5) \gg \ell$, we can find a proper $(\ell + 1)$-path P in H_5.

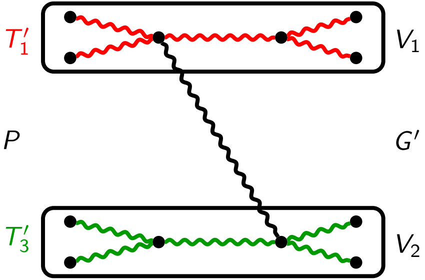
Creating $(\ell, 2\ell)$-trees over V_1 and V_2

H_1 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_1' spanning V_1.
H_2 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_2' spanning V_1.

H_3 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_3' spanning V_2.
H_4 + tiny c-fraction of H_5 = c-sparse $(\ell, 2\ell)$-tree T_4' spanning V_2.

Because $\delta(H_5) \gg \ell$, we can find a proper $(\ell + 1)$-path P in H_5.

$T = T_1' \cup T_3' \cup P$ is a sparse $(\geq \ell)$-tree spanning G.



Going on

We may suppose H_5 is still $1/2$-dense in G'. Because of the degree assumption, it has a $1/5^p$-dense 2^p-path-graph H with multiplicity at most c , where p satisfies $\ell \leq 2^p < 2\ell$.

Going on

We may suppose H_5 is still $1/2$-dense in G'. Because of the degree assumption, it has a $1/5^p$-dense 2^p-path-graph H with multiplicity at most c, where p satisfies $\ell \le 2^p < 2\ell$.

So far, G is decomposed, saved some ℓ-paths, into:

$$\underbrace{H} + \underbrace{T} + \underbrace{T_2'} + \underbrace{T_4'} + \underbrace{R}$$

- H:
  - 2^p-path-graph
  - $1/5^p$-dense
  - multiplicity c
- T:
  - $[\ell, 2\ell]$-tree
  - c-sparse
- T_2':
  - $(\ell, 2\ell)$-tree on V_1
  - c-sparse
- T_4':
  - $(\ell, 2\ell)$-tree on V_2
  - c-sparse
- R:
  - rest

Going on

We may suppose H_5 is still $1/2$-dense in G'. Because of the degree assumption, it has a $1/5^p$-dense 2^p-path-graph H with multiplicity at most c , where p satisfies $\ell \leq 2^p < 2\ell$.

So far, G is decomposed, saved some ℓ-paths, into:

$$\underbrace{H} + \underbrace{T} + \underbrace{T'_2} + \underbrace{T'_4} + \underbrace{R}$$

- H: • 2^p-path-graph • $1/5^p$-dense • multiplicity c
- T: • $[\ell, 2\ell]$-tree • c-sparse
- T'_2: • $(\ell, 2\ell)$-tree on V_1 • c-sparse
- T'_4: • $(\ell, 2\ell)$-tree on V_2 • c-sparse
- R: • rest

As long as possible, remove ℓ-paths from R. Call G_R what remains.

Theorem [Thomassen – 2008]

G_R admits a $(< \ell)$-path-graph with maximum degree at most $\ell - 1$.

Growing the paths of G_R

Call G_R this path-graph.

Growing the paths of G_R

Call G_R this path-graph.

Consider a 'tiny' $4c\ell^3$-fraction of H' of H, and orient H'^* in a balanced way. Then every vertex v is the origin of a private set of $K = 2c\ell^3 d_H(v)$ paths $P'_1, ..., P'_K$ in H'.

Growing the paths of G_R

Call G_R this path-graph.

Consider a 'tiny' $4c\ell^3$-fraction of H' of H, and orient H'^* in a balanced way. Then every vertex v is the origin of a private set of $K = 2c\ell^3 d_H(v)$ paths $P'_1, ..., P'_K$ in H'.

Since the multiplicity of H is at most c, every path P'_i is conflicting at v with at most $c|P'_i|d_H(v) \leq 2c\ell d_H(v)$ other paths of H'_v. Among $P'_1, ..., P'_K$, one can hence find $K/2c\ell d_H(v) = \ell^2$ non-conflicting paths. Then use these paths to extend those of H_r starting at v. Then we transform all paths of G_R into paths of length in between ℓ and 3ℓ.

Growing the paths of G_R

Call G_R this path-graph.

Consider a 'tiny' $4c\ell^3$-fraction of H' of H, and orient H'^* in a balanced way. Then every vertex v is the origin of a private set of $K = 2c\ell^3 d_H(v)$ paths $P'_1, ..., P'_K$ in H'.

Since the multiplicity of H is at most c, every path P'_i is conflicting at v with at most $c|P'_i|d_H(v) \leq 2c\ell d_H(v)$ other paths of H'_v. Among $P'_1, ..., P'_K$, one can hence find $K/2c\ell d_H(v) = \ell^2$ non-conflicting paths. Then use these paths to extend those of H_r starting at v. Then we transform all paths of G_R into paths of length in between ℓ and 3ℓ.

Call H'_R the resulting $(\geq \ell)$-path graph. In particular, this graph is sparse in H.

Getting a decomposition

Our objects are:

$$\underbrace{H}_{} + \underbrace{T}_{} + \underbrace{T'_2}_{} + \underbrace{T'_4}_{} + \underbrace{H'_R}_{}$$

- H:
  - 2^p-path-graph
  - $1/5^p$-dense
  - multiplicity c
- T:
  - $[\ell, 2\ell]$-tree
  - c-sparse
- T'_2:
  - $(\ell, 2\ell)$-tree on V_1
  - c-sparse
- T'_4:
  - $(\ell, 2\ell)$-tree on V_2
  - c-sparse
- H'_R:
  - $[\ell, 3\ell]$-path-graph
  - $4c\ell^3$-sparse

and form H_F. The final objective is to invoke a eulerian closed trail argument.

Getting a decomposition

Our objects are:

$$\underbrace{H}_{} + \underbrace{T}_{} + \underbrace{T_2'}_{} + \underbrace{T_4'}_{} + \underbrace{H_R'}_{}$$

- H:
  - 2^p-path-graph
  - $1/5^p$-dense
  - multiplicity c
- T:
  - $[\ell, 2\ell]$-tree
  - c-sparse
- T_2':
  - $(\ell, 2\ell)$-tree on V_1
  - c-sparse
- T_4':
  - $(\ell, 2\ell)$-tree on V_2
  - c-sparse
- H_R':
  - $[\ell, 3\ell]$-path-graph
  - $4c\ell^3$-sparse

and form H_F. The final objective is to invoke a eulerian closed trail argument.

So that all degrees of V_1 (resp. V_2) but maybe one are even, we can remove ℓ- or 2ℓ-paths from T_2' (resp. T_4'). In case V_1 and V_2 both have a remaining vertex with odd degree, with join them via a dummy edge.

Getting a decomposition

Our objects are:

$$\underbrace{H}_{\substack{\bullet\ 2^p\text{-path-graph}\\ \bullet\ 1/5^p\text{-dense}\\ \bullet\ \text{multiplicity } c}} + \underbrace{T}_{\substack{\bullet\ [\ell,2\ell]\text{-tree}\\ \bullet\ c\text{-sparse}}} + \underbrace{T_2'}_{\substack{\bullet\ (\ell,2\ell)\text{-tree on } V_1\\ \bullet\ c\text{-sparse}}} + \underbrace{T_4'}_{\substack{\bullet\ (\ell,2\ell)\text{-tree on } V_2\\ \bullet\ c\text{-sparse}}} + \underbrace{H_R'}_{\substack{\bullet\ [\ell,3\ell]\text{-path-graph}\\ \bullet\ 4c\ell^3\text{-sparse}}}$$

and form H_F. The final objective is to invoke a eulerian closed trail argument.

So that all degrees of V_1 (resp. V_2) but maybe one are even, we can remove ℓ- or 2ℓ-paths from T_2' (resp. T_4'). In case V_1 and V_2 both have a remaining vertex with odd degree, with join them via a dummy edge.

Now, since the multiplicity of H is arbitrarily small and $T \cup T_2' \cup T_4' \cup H_R'$ is, say, $4c\ell^3$-sparse, H_F is a $(\leq 3\ell)$-path-graph with multiplicity less than $1/24\ell$. Then a conflictless eulerian closed trail exists in H_F. Going along it, we finish the decomposition into ℓ-paths. ■

Conclusion

- For paths, 64-edge-connectivity suffices...

Conclusion

- For paths, 64-edge-connectivity suffices...
- ... but how far can this be pushed?

Conclusion

- For paths, 64-edge-connectivity suffices...
- ... but how far can this be pushed?
- Actually 48 works (save two trees for constructing T).

Conclusion

- For paths, 64-edge-connectivity suffices...
- ... but how far can this be pushed?
- Actually 48 works (save two trees for constructing T).
- Similar approaches for other kinds of trees?

Conclusion

- For paths, 64-edge-connectivity suffices...
- ... but how far can this be pushed?
- Actually 48 works (save two trees for constructing T).
- Similar approaches for other kinds of trees?

Thank you for your attention.