Path partitions of regular graphs

Shoham Letzter

joint work with Vytautas Gruslys

University of Cambridge and ETH–ITS

SIAM DM

June 2018

Dirac's theorem

Theorem (Dirac 1952)

Dirac's theorem

Theorem (Dirac 1952)

Let G be a graph on $n \geq 3$ vertices with minimum degree at least $n/2$. Then G contains a Hamilton cycle.

Dirac's theorem

Theorem (Dirac 1952)

Let G be a graph on $n \geq 3$ vertices with minimum degree at least $n/2$. Then G contains a Hamilton cycle.

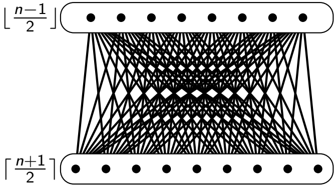
Question. What if $\delta(G) < n/2$?

Dirac's theorem

Theorem (Dirac 1952)

Let G be a graph on $n \geq 3$ vertices with minimum degree at least $n/2$. Then G contains a Hamilton cycle.

Question. What if $\delta(G) < n/2$?

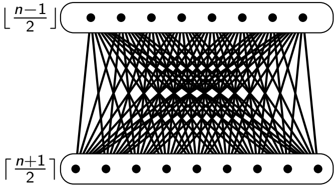


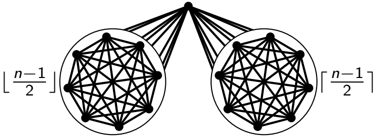
Dirac's theorem

Theorem (Dirac 1952)

Let G be a graph on $n \geq 3$ vertices with minimum degree at least $n/2$. Then G contains a Hamilton cycle.

Question. What if $\delta(G) < n/2$?



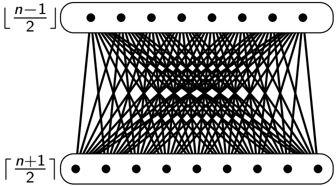


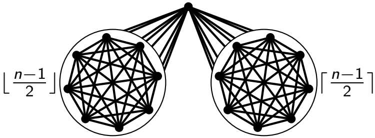
Dirac's theorem

Theorem (Dirac 1952)

Let G be a graph on $n \geq 3$ vertices with minimum degree at least $n/2$. Then G contains a Hamilton cycle.

Question. What if $\delta(G) < n/2$?





Answer. Need to circumvent both examples!

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

Let G be a ***regular*** *t-**connected** graph on n vertices with degree at least $n/(t+1)$. Then G has a Hamilton cycle.*

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

*Let G be a **regular** t-**connected** graph on n vertices with degree at least $n/(t+1)$. Then G has a Hamilton cycle.*

- **Jackson ('80).** $t = 2$.

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

*Let G be a **regular** t-**connected** graph on n vertices with degree at least $n/(t+1)$. Then G has a Hamilton cycle.*

- **Jackson ('80).** $t = 2$.
- **Kühn, Lo, Osthus, Staden ('14).** $t = 3$ (and large n).

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

*Let G be a **regular** t-**connected** graph on n vertices with degree at least $n/(t+1)$. Then G has a Hamilton cycle.*

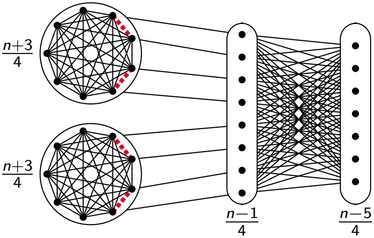
- **Jackson ('80).** $t = 2$.
- **Kühn, Lo, Osthus, Staden ('14).** $t = 3$ (and large n).
- **Jung ('84).** false for $t \geq 4$.

Bollobás and Häggkvist's Conjecture

Conjecture (Bollobás, Häggkvist 70's)

*Let G be a **regular** t-**connected** graph on n vertices with degree at least $n/(t+1)$. Then G has a Hamilton cycle.*

- **Jackson ('80).** $t = 2$.
- **Kühn, Lo, Osthus, Staden ('14).** $t = 3$ (and large n).
- **Jung ('84).** false for $t \geq 4$.



Allowing more cycles

Allowing more cycles

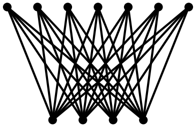
Theorem (Kouider, Lonc '96)

Any graph on n vertices with minimum degree at least d can be ***covered*** *by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*

Allowing more cycles

Theorem (Kouider, Lonc '96)

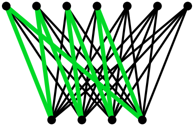
*Any graph on n vertices with minimum degree at least d can be **covered** by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*



Allowing more cycles

Theorem (Kouider, Lonc '96)

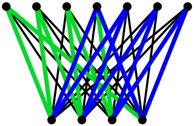
*Any graph on n vertices with minimum degree at least d can be **covered** by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*



Allowing more cycles

Theorem (Kouider, Lonc '96)

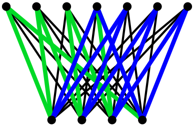
*Any graph on n vertices with minimum degree at least d can be **covered** by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*



Allowing more cycles

Theorem (Kouider, Lonc '96)

*Any graph on n vertices with minimum degree at least d can be **covered** by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*

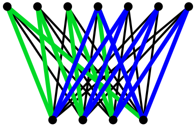


The theorem is tight.

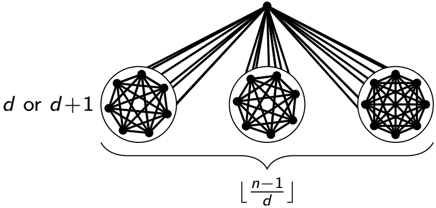
Allowing more cycles

Theorem (Kouider, Lonc '96)

*Any graph on n vertices with minimum degree at least d can be **covered** by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*



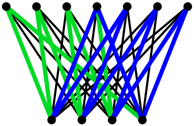
The theorem is tight.



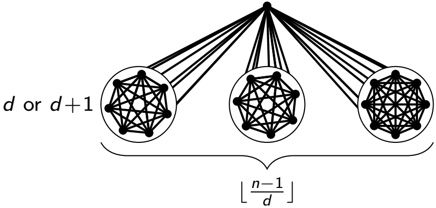
Allowing more cycles

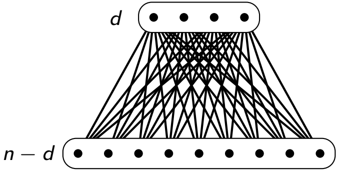
Theorem (Kouider, Lonc '96)

Any graph on n vertices with minimum degree at least d can be ***covered*** *by $\lfloor \frac{n-1}{d} \rfloor$ cycles (edges and vertices allowed).*



The theorem is tight.





Path partitions of regular graphs

Path partitions of regular graphs

Conjecture (Magnant, Martin '09)

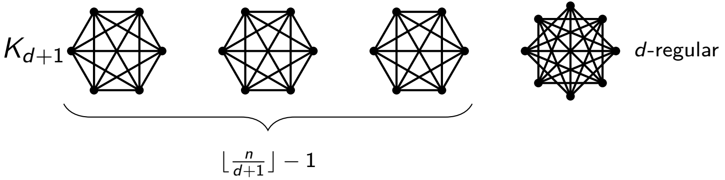
*Let G be a d-**regular** graph on n vertices. Then $V(G)$ can be **partitioned** into at most $\lfloor \frac{n}{d+1} \rfloor$ paths.*

Path partitions of regular graphs

Conjecture (Magnant, Martin '09)

*Let G be a d-**regular** graph on n vertices. Then $V(G)$ can be **partitioned** into at most $\lfloor \frac{n}{d+1} \rfloor$ paths.*

- The conjecture is tight:

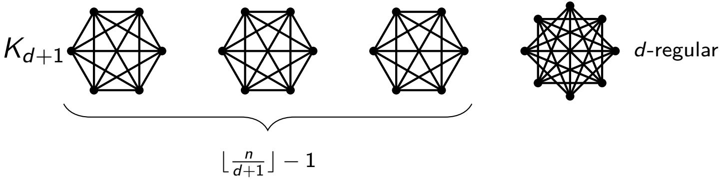


Path partitions of regular graphs

Conjecture (Magnant, Martin '09)

*Let G be a d-**regular** graph on n vertices. Then $V(G)$ can be **partitioned** into at most $\lfloor \frac{n}{d+1} \rfloor$ paths.*

- The conjecture is tight:



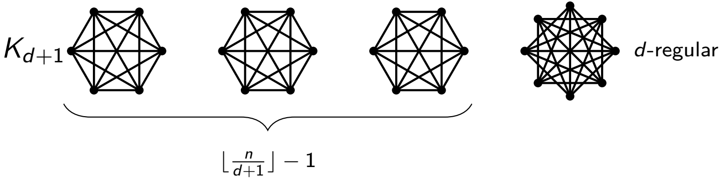
- **Magnant, Martin ('09).** $d \leq 5$.

Path partitions of regular graphs

Conjecture (Magnant, Martin '09)

*Let G be a d-**regular** graph on n vertices. Then $V(G)$ can be **partitioned** into at most $\lfloor \frac{n}{d+1} \rfloor$ paths.*

- The conjecture is tight:



- **Magnant, Martin ('09).** $d \leq 5$.
- **Han ('17).** If $d \geq cn$ then all but $o(n)$ vertices can be covered by $\lfloor \frac{n}{d+1} \rfloor$ vertex-disjoint paths.

Our results

Our results

We prove Magnant and Martin's conjecture for dense graphs.

Our results

We prove Magnant and Martin's conjecture for dense graphs.

Theorem (Gruslys, L. '18+)

Our results

We prove Magnant and Martin's conjecture for dense graphs.

Theorem (Gruslys, L. '18+)

*Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ **cycles**.*

Our results

We prove Magnant and Martin's conjecture for dense graphs.

Theorem (Gruslys, L. '18+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ ***cycles****.*

Theorem (Gruslys, L. '18+)

Let G be a ***bipartite*** *d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{2d} \rfloor$* ***cycles****.*

Hamiltonicity of expanders

Hamiltonicity of expanders

A **sparse cut** in a graph is a partition $\{X, Y\}$ of the vertices, such that $e(X, Y) = o(|X||Y|)$.

Hamiltonicity of expanders

A **sparse cut** in a graph is a partition $\{X, Y\}$ of the vertices, such that $e(X, Y) = o(|X||Y|)$.

Theorem (Kühn, Osthus, Treglown '10)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Suppose that G has no sparse cuts, then G contains a Hamilton cycle.

Hamiltonicity of expanders

A **sparse cut** in a graph is a partition $\{X, Y\}$ of the vertices, such that $e(X, Y) = o(|X||Y|)$.

Theorem (Kühn, Osthus, Treglown '10)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Suppose that G has no sparse cuts, then G contains a Hamilton cycle.

- The proof uses the 'absorbing technique'.

Hamiltonicity of expanders

A **sparse cut** in a graph is a partition $\{X, Y\}$ of the vertices, such that $e(X, Y) = o(|X||Y|)$.

Theorem (Kühn, Osthus, Treglown '10)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Suppose that G has no sparse cuts, then G contains a Hamilton cycle.

- The proof uses the 'absorbing technique'.
- Standard applications use Regularity Lemma.

Hamiltonicity of expanders

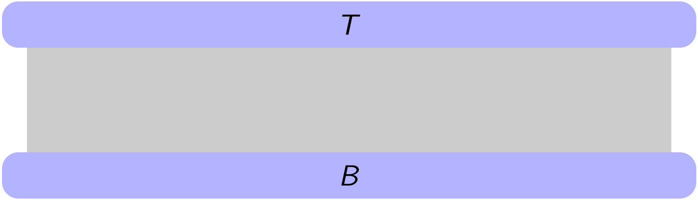
A **sparse cut** in a graph is a partition $\{X, Y\}$ of the vertices, such that $e(X, Y) = o(|X||Y|)$.

Theorem (Kühn, Osthus, Treglown '10)

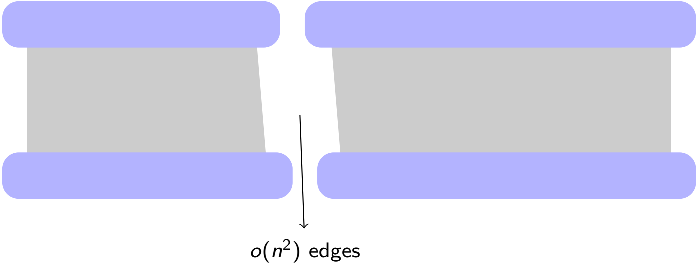
Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Suppose that G has no sparse cuts, then G contains a Hamilton cycle.

- The proof uses the 'absorbing technique'.
- Standard applications use Regularity Lemma. Here it is possible to avoid it, using an argument by Lo and Patel ('15) which uses the 'Rotation-Extension' technique.

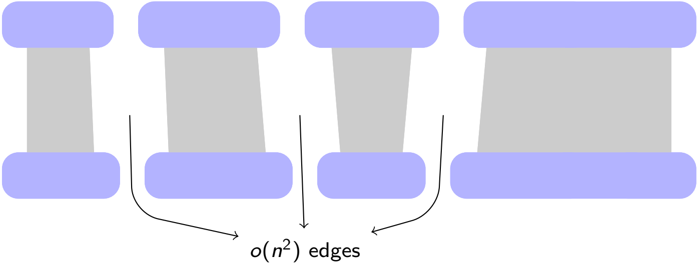
Partition vertices into clusters



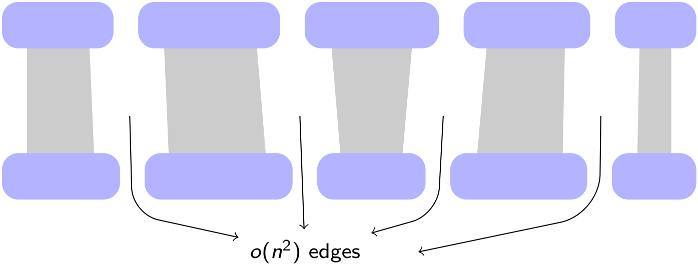
Partition vertices into clusters



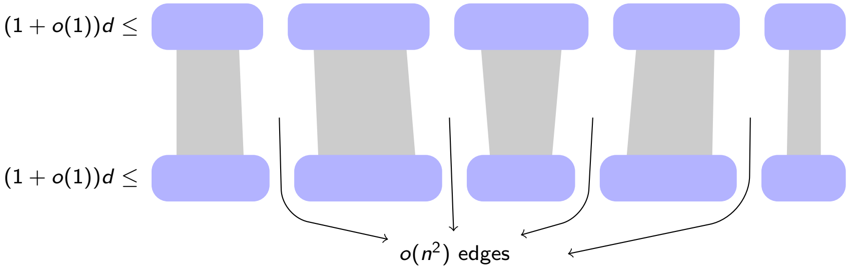
Partition vertices into clusters



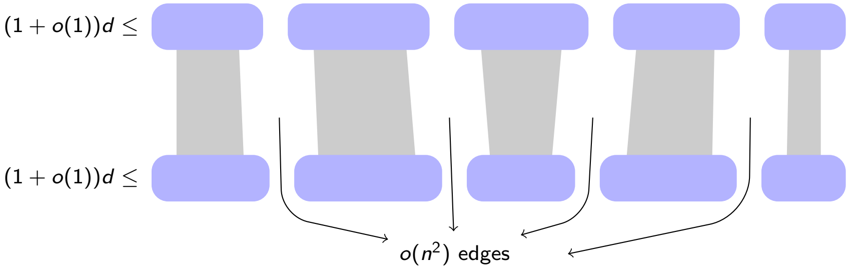
Partition vertices into clusters



Partition vertices into clusters

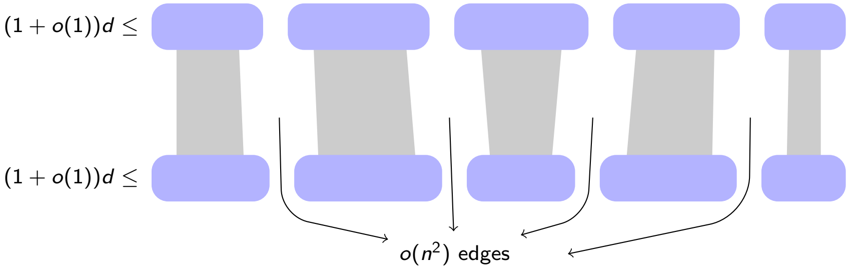


Partition vertices into clusters



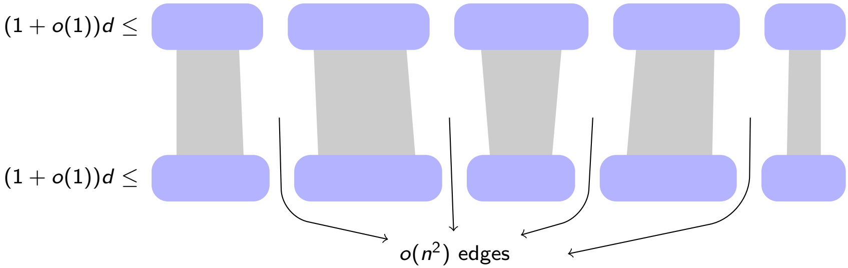
- we have at most $\frac{n}{2d(1+o(1))} \leq \lfloor n/2d \rfloor + 1$ clusters;

Partition vertices into clusters



- we have at most $\frac{n}{2d(1+o(1))} \leq \lfloor n/2d \rfloor + 1$ clusters;
- by a variant of the Hamiltonicity Theorem, every **balanced** subgraph of a cluster, obtained be removing $o(n)$ vertices, is Hamiltonian;

Partition vertices into clusters

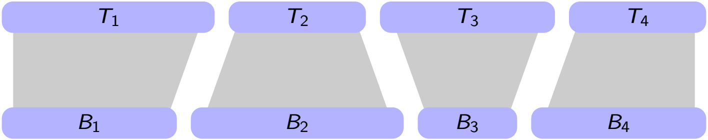


- we have at most $\frac{n}{2d(1+o(1))} \le \lfloor n/2d \rfloor + 1$ clusters;
- by a variant of the Hamiltonicity Theorem, every **balanced** subgraph of a cluster, obtained be removing $o(n)$ vertices, is Hamiltonian;
- such a partition was also used by KLOS.

Balancing matchings

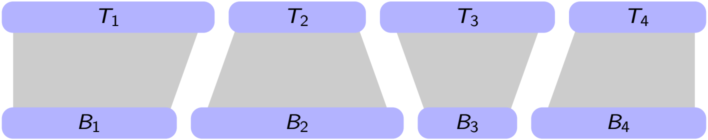
Balancing matchings

Denote the vertices of the i-th cluster by $B_i \cup T_i$.



Balancing matchings

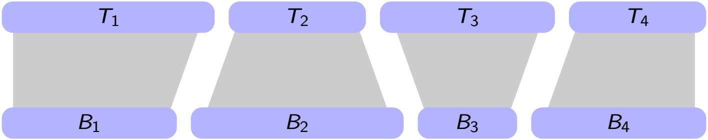
Denote the vertices of the i-th cluster by $B_i \cup T_i$.



A **balancing matching** is a matching M s.t.

Balancing matchings

Denote the vertices of the i-th cluster by $B_i \cup T_i$.

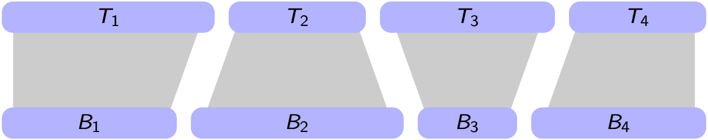


A **balancing matching** is a matching M s.t.

(a) $|T_i \setminus V(M)| = |B_i \setminus V(M)|$.

Balancing matchings

Denote the vertices of the i-th cluster by $B_i \cup T_i$.



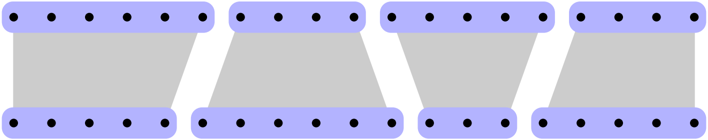
A **balancing matching** is a matching M s.t.

(a) $|T_i \setminus V(M)| = |B_i \setminus V(M)|$.

(b) $|M| = o(n)$.

Balancing matchings

Denote the vertices of the i-th cluster by $B_i \cup T_i$.



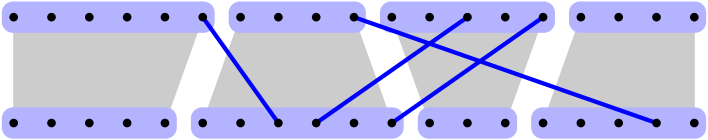
A **balancing matching** is a matching M s.t.

(a) $|T_i \setminus V(M)| = |B_i \setminus V(M)|$.

(b) $|M| = o(n)$.

Balancing matchings

Denote the vertices of the i-th cluster by $B_i \cup T_i$.



A **balancing matching** is a matching M s.t.

(a) $|T_i \setminus V(M)| = |B_i \setminus V(M)|$.

(b) $|M| = o(n)$.

Combining two balancing matchings

Combining two balancing matchings

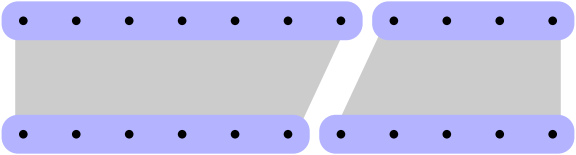
New aim.

Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

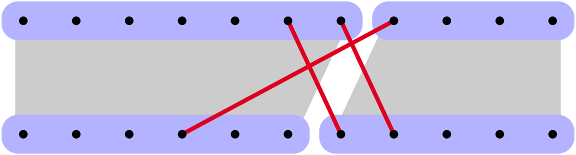
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



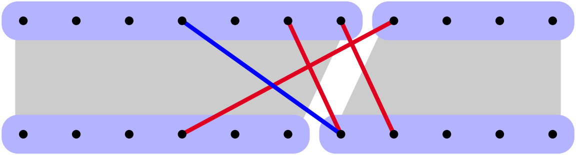
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



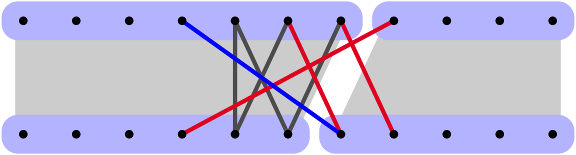
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



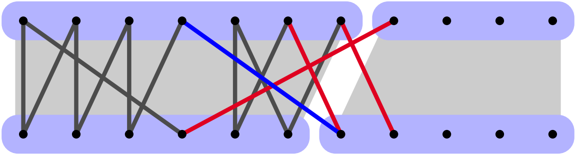
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



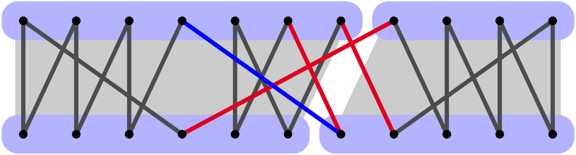
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



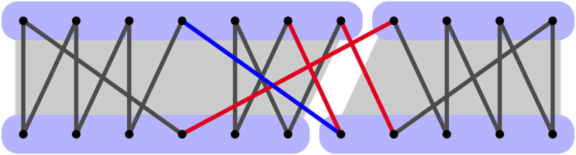
Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



Combining two balancing matchings

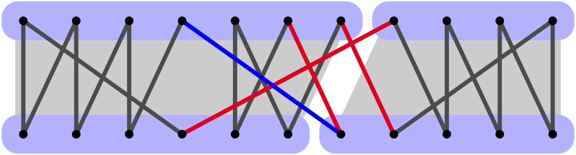
New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



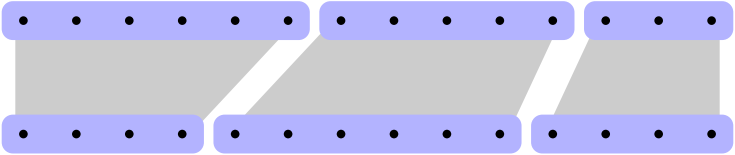
It is easy to find one balancing matching.

Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

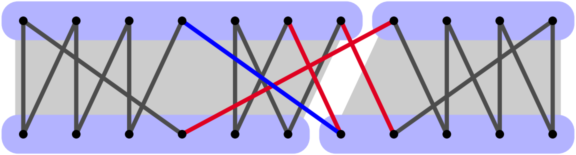


It is easy to find one balancing matching.

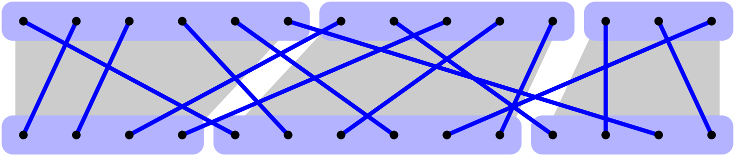


Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

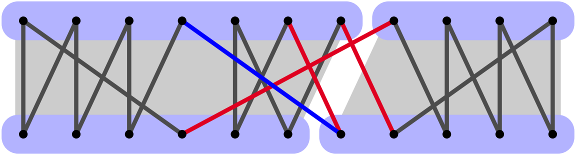


It is easy to find one balancing matching.

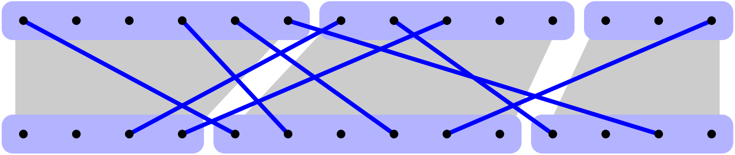


Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

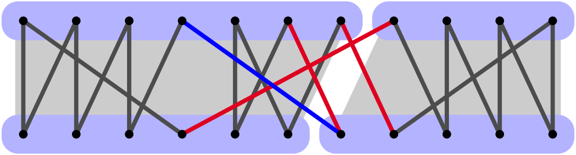


It is easy to find one balancing matching.

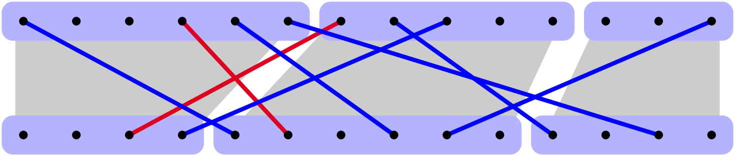


Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

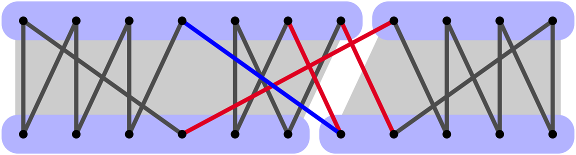


It is easy to find one balancing matching.

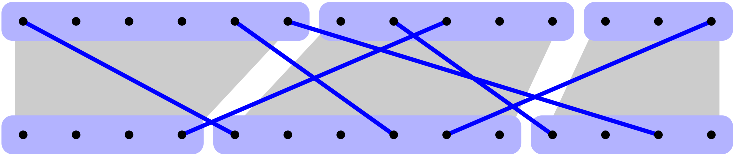


Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.

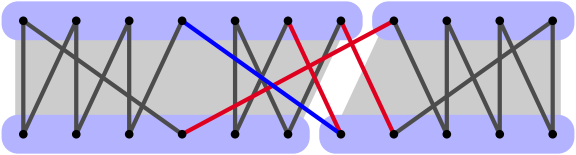


It is easy to find one balancing matching.

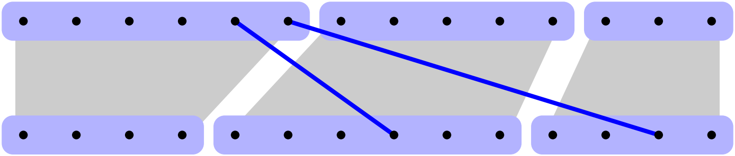


Combining two balancing matchings

New aim. find two edge-disjoint balancing matchings M_1 and M_2 whose union does not have cycles.



It is easy to find one balancing matching.



Matchings in G_σ

Matchings in G_σ

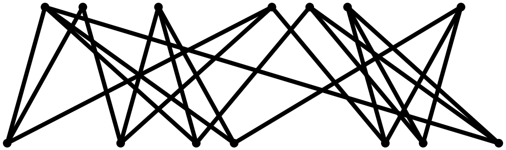
Let σ be an ordering of $V(G)$.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

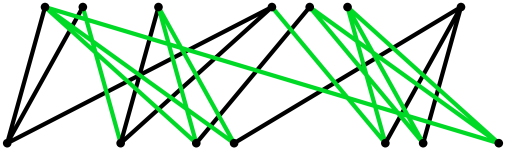
Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.



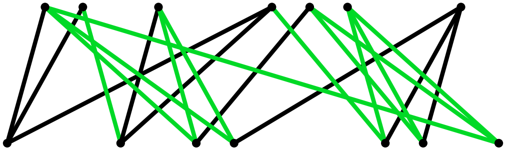
Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.



Matchings in G_σ

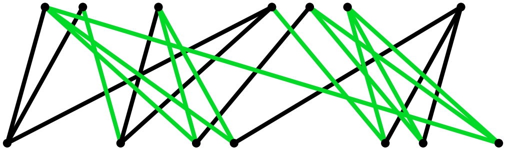
Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.



Let $\bar{\sigma}$ be the reverse ordering of σ.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

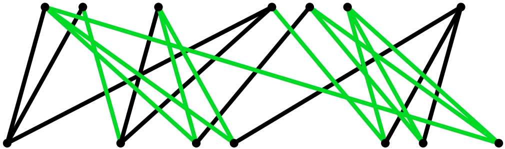


Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

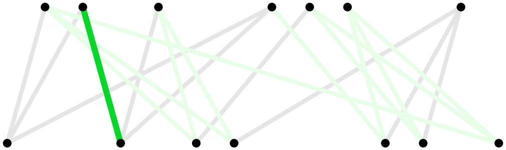


Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

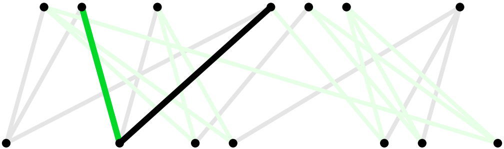


Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

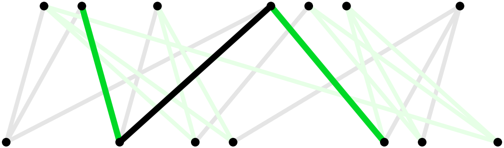


Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.

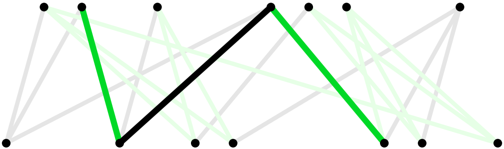


Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.



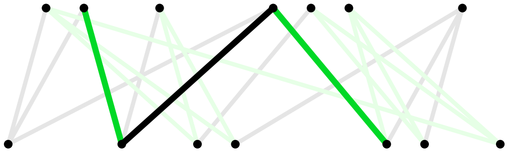
Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

Suffices to show.

Matchings in G_σ

Let σ be an ordering of $V(G)$. G_σ is the subgraph whose edges are edges tb of G, where $t \in T$, $b \in B$ and $\sigma(t) < \sigma(b)$.



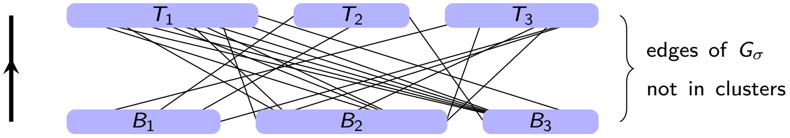
Let $\bar{\sigma}$ be the reverse ordering of σ. Note

- $G_{\bar{\sigma}} = G \setminus G_\sigma$;
- let M_1 and M_2 be matchings in G_σ and $G_{\bar{\sigma}}$. Then $M_1 \cup M_2$ has no cycles.

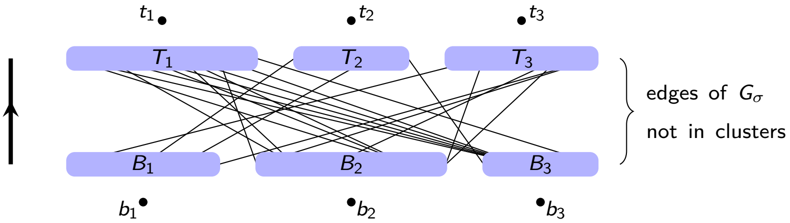
Suffices to show. If σ is random, G_σ has a balancing matching with probability larger than $1/2$.

A flow network

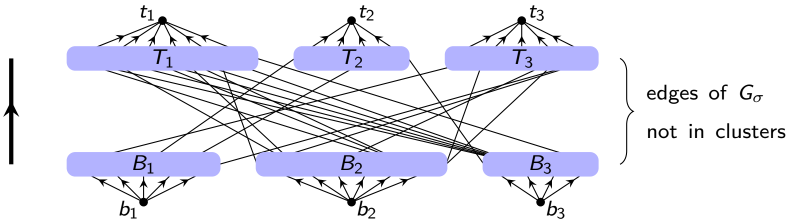
A flow network



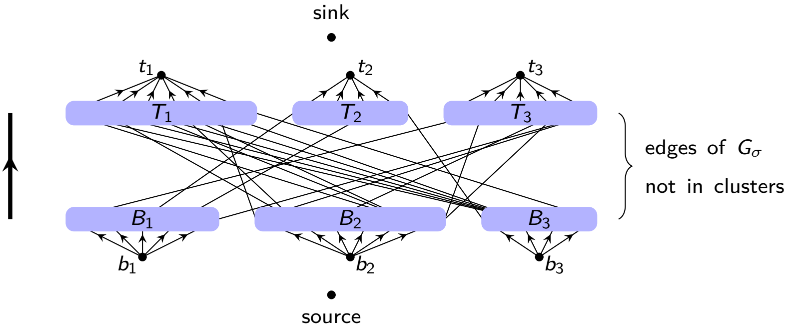
A flow network



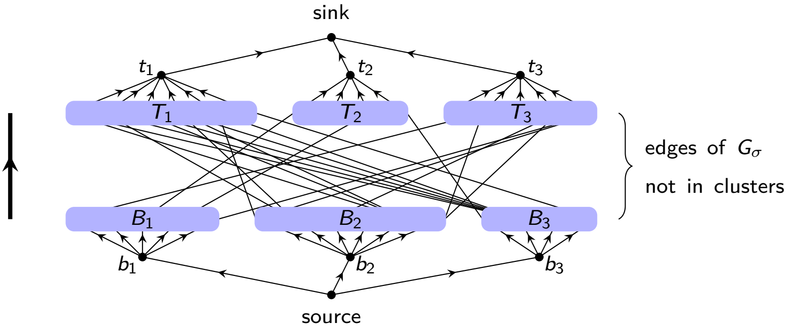
A flow network



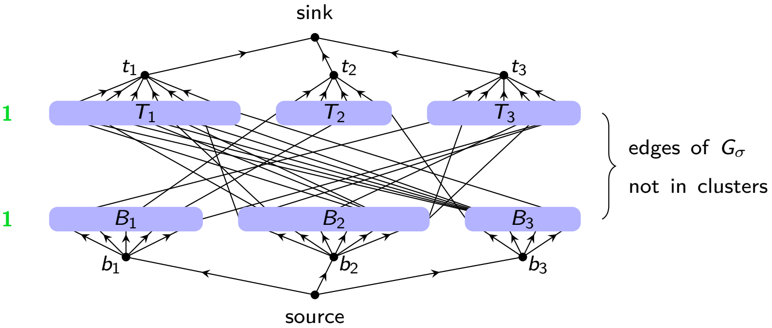
A flow network



A flow network

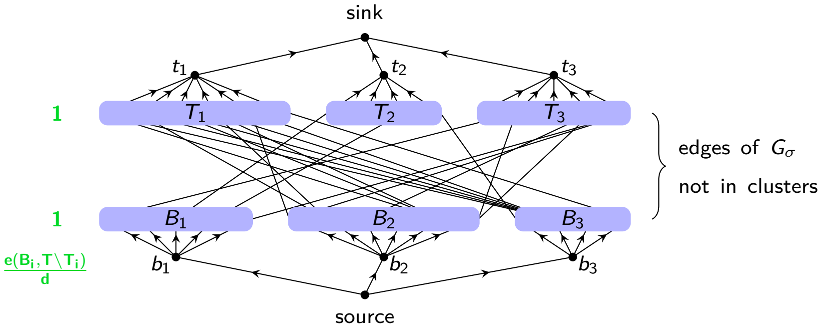


A flow network



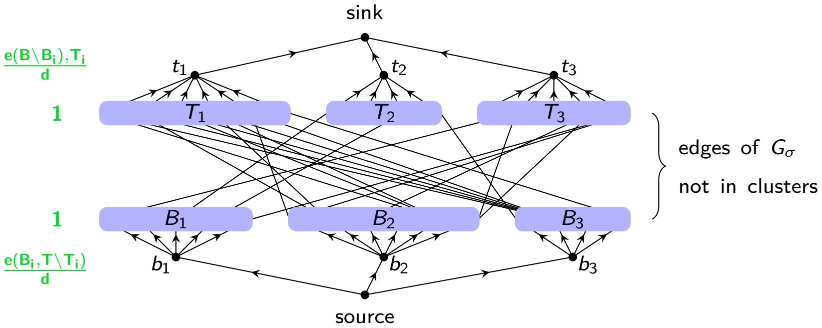
- vertices of G receive capacity 1,

A flow network



- vertices of G receive capacity 1,
- b_i has capacity $\frac{e(B_i, T \setminus T_i)}{d}$,

A flow network



- vertices of G receive capacity 1,
- b_i has capacity $\frac{e(B_i,T\setminus T_i)}{d}$,
- t_j has capacity $\frac{e(B\setminus B_i,T_i)}{d}$.

Finding a balancing matching in G_σ

Finding a balancing matching in G_σ

Lemma

With probability $> \frac{1}{2}$, *there is a flow with value* $\sum_{i \neq j} \frac{e(B_i, T_j)}{d} - \frac{9}{10}$.

Finding a balancing matching in G_σ

Lemma

With probability $> \frac{1}{2}$, *there is a flow with value* $\sum_{i \neq j} \frac{e(B_i, T_j)}{d} - \frac{9}{10}$.

- view such a flow as a fractional matching;

Finding a balancing matching in G_σ

Lemma

With probability $> \frac{1}{2}$, *there is a flow with value* $\sum_{i \neq j} \frac{e(B_i, T_j)}{d} - \frac{9}{10}$.

- view such a flow as a fractional matching;
- it is $\frac{9}{10}$-**almost-balancing**;

Finding a balancing matching in G_σ

Lemma

With probability $> \frac{1}{2}$, *there is a flow with value* $\sum_{i \neq j} \frac{e(B_i, T_j)}{d} - \frac{9}{10}$.

- view such a flow as a fractional matching;
- it is $\frac{9}{10}$-**almost-balancing**;
- by weight shifting, find a $\frac{9}{10}$-almost-balancing integer-valued fractional matching,

Finding a balancing matching in G_σ

Lemma

With probability $> \frac{1}{2}$, *there is a flow with value* $\sum_{i \neq j} \frac{e(B_i, T_j)}{d} - \frac{9}{10}$.

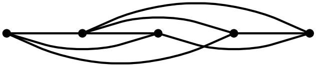
- view such a flow as a fractional matching;
- it is $\frac{9}{10}$-**almost-balancing**;
- by weight shifting, find a $\frac{9}{10}$-almost-balancing integer-valued fractional matching, i.e. a balancing matching.

2-lifts

2-lifts

Let G be a graph and σ an ordering.

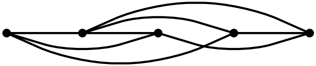
G



2-lifts

Let G be a graph and σ an ordering. Define $L_\sigma(G)$ by

G



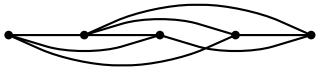
$L_\sigma(G)$

2-lifts

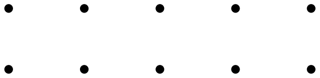
Let G be a graph and σ an ordering. Define $L_\sigma(G)$ by

- $V(L_\sigma(G)) = \{v_0, v_1 : v \in V(G)\}$,

G



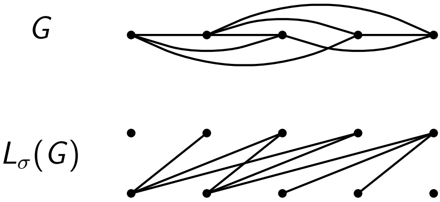
$L_\sigma(G)$



2-lifts

Let G be a graph and σ an ordering. Define $L_\sigma(G)$ by

- $V(L_\sigma(G)) = \{v_0, v_1 : v \in V(G)\}$,
- $E(L_\sigma(G)) = \{v_0 u_1 : vu \in E(G) \text{ and } \sigma(v) < \sigma(u)\}$.



Proof for non-bipartite graphs

Proof for non-bipartite graphs

- partition the vertices into clusters,

Proof for non-bipartite graphs

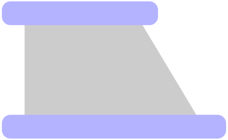
- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,

Proof for non-bipartite graphs

- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,
- in $L_\sigma(G)$, almost bipartite clusters become two clusters,

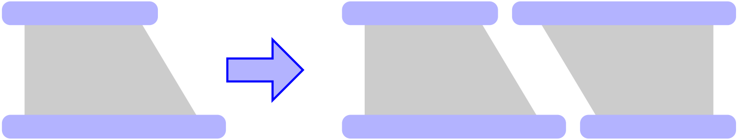
Proof for non-bipartite graphs

- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,
- in $L_\sigma(G)$, almost bipartite clusters become two clusters,



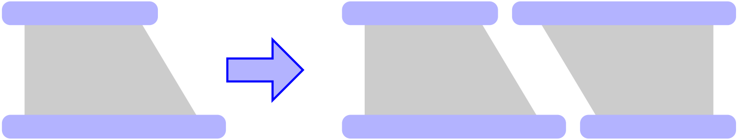
Proof for non-bipartite graphs

- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,
- in $L_\sigma(G)$, almost bipartite clusters become two clusters,



Proof for non-bipartite graphs

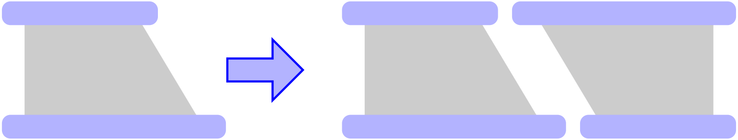
- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,
- in $L_\sigma(G)$, almost bipartite clusters become two clusters,



- far-from-bipartite clusters become one balanced cluster,

Proof for non-bipartite graphs

- partition the vertices into clusters,
- distinguish **almost-bipartite** and **far-from-bipartite** clusters,
- in $L_\sigma(G)$, almost bipartite clusters become two clusters,



- far-from-bipartite clusters become one balanced cluster,
- show that with positive probability, $L_\sigma(G)$ contains a balancing matching.

Open Problems

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

- improve the lower bound on d.

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

- improve the lower bound on d. We can probably obtain the result for $d \geq \frac{n}{\log \log \log \log n}$;

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

- improve the lower bound on d. We can probably obtain the result for $d \geq \frac{n}{\log\log\log\log n}$;
- is there a version for regular directed graphs?

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

- improve the lower bound on d. We can probably obtain the result for $d \geq \frac{n}{\log \log \log \log n}$;
- is there a version for regular directed graphs?
- directed or bipartite versions of Bollobás-Häggkvist (for 3-connected graphs).

Open Problems

Theorem (Gruslys, L. '17+)

Let G be a d-regular graph on n vertices, where $d \geq cn$ and $n \geq n_0(c)$. Then $V(G)$ can be partitioned into at most $\lfloor \frac{n}{d+1} \rfloor$ cycles.

Open Problems.

- improve the lower bound on d. We can probably obtain the result for $d \geq \frac{n}{\log \log \log \log n}$;
- is there a version for regular directed graphs?
- directed or bipartite versions of Bollobás-Häggkvist (for 3-connected graphs).

Thank you for listening!