Improved upper bounds for λ-backbone colorings along matchings and stars

Bert Marchal
joint work with:

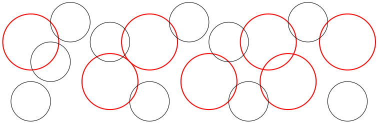
Hajo Broersma, Daniel Paulusma, A.N.M. Salman

January 23, 2007

frequency assignment

setting

- network of transmitters
- interference if they are close and broadcast on the *same* frequency
- stronger transmitters also interfere if they broadcast on *similar* frequencies

set of transmitters 

problem

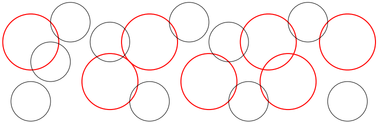
Assign frequency channels to the transmitters in such a way that interference is avoided and bandwidth is minimized.

frequency assignment

setting

- network of transmitters
- interference if they are close and broadcast on the *same* frequency
- stronger transmitters also interfere if they broadcast on *similar* frequencies

set of transmitters



problem

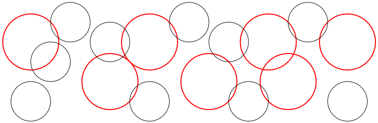
Assign frequency channels to the transmitters in such a way that interference is avoided and bandwidth is minimized.

frequency assignment

setting

- network of transmitters
- interference if they are close and broadcast on the *same* frequency
- stronger transmitters also interfere if they broadcast on *similar* frequencies

set of transmitters



problem

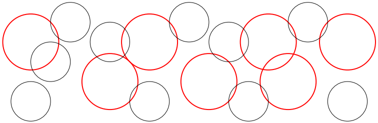
Assign frequency channels to the transmitters in such a way that interference is avoided and bandwidth is minimized.

frequency assignment

setting

- network of transmitters
- interference if they are close and broadcast on the *same* frequency
- stronger transmitters also interfere if they broadcast on *similar* frequencies

set of transmitters



problem

Assign frequency channels to the transmitters in such a way that interference is avoided and bandwidth is minimized.

general model

graph representation

- vertices of G represent transmitters
- vertices are adjacent when corresponding transmitters are likely to interfere
- edges between stronger transmitters that interfere when they broadcast on *similar* frequencies form a subgraph of G

general framework

- graph G_1 and subgraph G_2 of G_1
- determine coloring satisfying restrictions of type 1 in G_1 and of type 2 in G_2 using minimum number of colors

general model

graph representation

- vertices of G represent transmitters
- vertices are adjacent when corresponding transmitters are likely to interfere
- edges between stronger transmitters that interfere when they broadcast on *similar* frequencies form a subgraph of G

general framework

- graph G_1 and subgraph G_2 of G_1
- determine coloring satisfying restrictions of type 1 in G_1 and of type 2 in G_2 using minimum number of colors

general model

graph representation

- vertices of G represent transmitters
- vertices are adjacent when corresponding transmitters are likely to interfere
- edges between stronger transmitters that interfere when they broadcast on *similar* frequencies form a subgraph of G

general framework

- graph G_1 and subgraph G_2 of G_1
- determine coloring satisfying restrictions of type 1 in G_1 and of type 2 in G_2 using minimum number of colors

general model

graph representation

- vertices of G represent transmitters
- vertices are adjacent when corresponding transmitters are likely to interfere
- edges between stronger transmitters that interfere when they broadcast on *similar* frequencies form a subgraph of G

general framework

- graph G_1 and subgraph G_2 of G_1
- determine coloring satisfying restrictions of type 1 in G_1 and of type 2 in G_2 using minimum number of colors

general model

graph representation

- vertices of G represent transmitters
- vertices are adjacent when corresponding transmitters are likely to interfere
- edges between stronger transmitters that interfere when they broadcast on *similar* frequencies form a subgraph of G

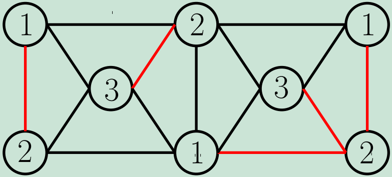
general framework

- graph G_1 and subgraph G_2 of G_1
- determine coloring satisfying restrictions of type 1 in G_1 and of type 2 in G_2 using minimum number of colors

examples

example 1

G_2 (red edges) is a subgraph of G_1



type 1: adjacent vertices have different color numbers
type 2: adjacent vertices have different color numbers

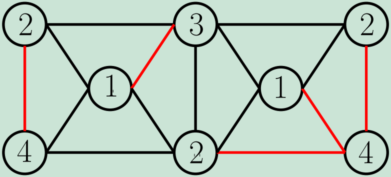
example 2

examples

example 1

example 2

G_2 (red edges) is a subgraph of G_1



type 1: adjacent vertices have different color numbers
type 2: adjacent vertices have color numbers that differ by at least 2

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)

$BBC_\lambda(G, H)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ $(\lambda \geq 2)$
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

- smallest number ℓ for which there exists a λ-backbone coloring of (G, H) with color numbers $1, \ldots, \ell$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

- smallest number ℓ for which there exists a λ-backbone coloring of (G, H) with color numbers $1, \ldots, \ell$
- upper bounds in terms of $\chi(G)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

- smallest number ℓ for which there exists a λ-backbone coloring of (G, H) with color numbers $1, \ldots, \ell$
- upper bounds in terms of $\chi(G)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ ($\lambda \geq 2$)
- other neighbors just have different color numbers

$BBC_\lambda(G, H)$

- smallest number ℓ for which there exists a λ-backbone coloring of (G, H) with color numbers $1, \ldots, \ell$
- upper bounds in terms of $\chi(G)$

definitions

backbone

- backbone of network is substructure of transmitters that are more crucial than the rest
- denoted by H and forms a spanning subgraph of the network graph G

λ-backbone coloring of (G, H)

- neighbors in H have color numbers that differ by at least λ $(\lambda \geq 2)$
- other neighbors just have different color numbers

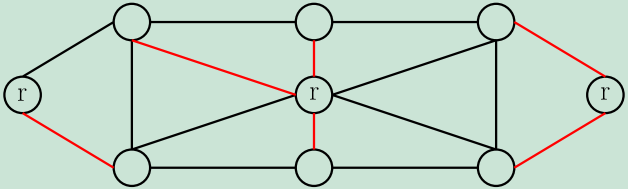
$BBC_\lambda(G, H)$

- smallest number ℓ for which there exists a λ-backbone coloring of (G, H) with color numbers $1, \ldots, \ell$
- upper bounds in terms of $\chi(G)$

star backbones

- star S_q is complete 2-partite graph with root r and q leaves
- star backbone is spanning collection of pairwise disjoint stars

example



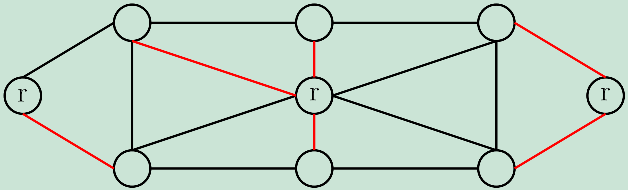
graph G with star backbone S

- application: e.g. sensor networks

star backbones

- star S_q is complete 2-partite graph with root r and q leaves
- star backbone is spanning collection of pairwise disjoint stars

example



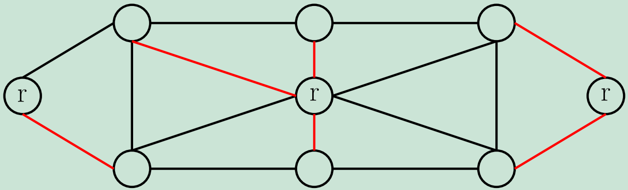
graph G with star backbone S

- application: e.g. sensor networks

star backbones

- star S_q is complete 2-partite graph with root r and q leaves
- star backbone is spanning collection of pairwise disjoint stars

example



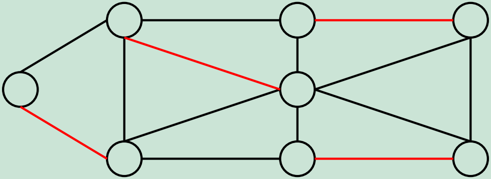
graph G with star backbone S

- application: e.g. sensor networks

matching backbones

- matching backbone is spanning collection of pairwise disjoint copies of S_1

example



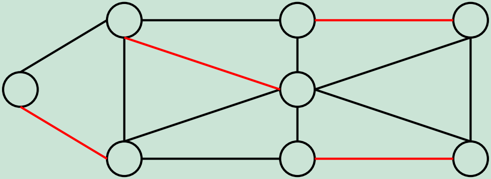
graph G with matching backbone M

- application: e.g. military scenarios

matching backbones

- matching backbone is spanning collection of pairwise disjoint copies of S_1

example



graph G with matching backbone M

- application: e.g. military scenarios

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G, M)$ roughly grow like $(2 - \frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G, S)$ roughly grow like $(2 - \frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G, T) \leq \chi(G) + 2$

idea

additive constant (depending on λ) in split graphs ?

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G, M)$ roughly grow like $(2 - \frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G, S)$ roughly grow like $(2 - \frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G, T) \leq \chi(G) + 2$

idea

additive constant (depending on λ) in split graphs ?

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G, M)$ roughly grow like $(2 - \frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G, S)$ roughly grow like $(2 - \frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G, T) \leq \chi(G) + 2$

idea

additive constant (depending on λ) in split graphs ?

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G,M)$ roughly grow like $(2-\frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G,S)$ roughly grow like $(2-\frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G,T) \leq \chi(G)+2$

idea

additive constant (depending on λ) in split graphs ?

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G, M)$ roughly grow like $(2 - \frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G, S)$ roughly grow like $(2 - \frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G, T) \leq \chi(G) + 2$

idea

additive constant (depending on λ) in split graphs ?

previous results

Broersma et al. (2004)

- ub for $BBC_\lambda(G, M)$ roughly grow like $(2 - \frac{1}{\lambda})\chi(G)$
- ub for $BBC_\lambda(G, S)$ roughly grow like $(2 - \frac{2}{\lambda+1})\chi(G)$
- multiplicative factor (bad!)

Broersma, Fomin, Golovach and Woeginger (2003)

- for *split graphs* G and tree backbones T, $BBC_2(G, T) \leq \chi(G) + 2$

idea

additive constant (depending on λ) in split graphs ?

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

split graphs

definition

A split graph is a graph whose vertex set can be partitioned into a *clique* and an *independent set* with possibly edges in between.

motivation

1. nice structural properties
2. every graph can be turned into a split graph

application:

- consider sensor network with weak sensors (heat, smoke) and strong sensors (PC's, Laptops)
- weak sensors do not interfere with one another
- strong sensors interfere with one another and with weak sensors in their vicinity

upper bounds on $BBC_\lambda(G,S)$

Theorem

Let $\lambda \geq 2$ and let G be a split graph with $\chi(G) = k \geq 2$. For every star backbone S of G,

$$BBC_\lambda(G,S) \leq \begin{cases} k+\lambda & \text{if } k=3 \text{ and } \lambda \geq 2 \text{ or } k \geq 4,\ \lambda = 2 \\ k+\lambda-1 & \text{in the other cases.} \end{cases}$$

The bounds are tight.

$\Rightarrow$ additive constant depending on λ

example

split graph G and star backbone S with $\chi(G) = 5$ and $\lambda = 2 \Rightarrow$ $BBC_2(G,S) \leq 7$

upper bounds on $BBC_\lambda(G,S)$

Theorem

Let $\lambda \geq 2$ and let G be a split graph with $\chi(G) = k \geq 2$. For every star backbone S of G,

$$BBC_\lambda(G,S) \leq \begin{cases} k+\lambda & \text{if } k=3 \text{ and } \lambda \geq 2 \text{ or } k \geq 4,\ \lambda = 2 \\ k+\lambda-1 & \text{in the other cases.} \end{cases}$$

The bounds are tight.

$\Rightarrow$ additive constant depending on λ

example

split graph G and star backbone S with $\chi(G) = 5$ and $\lambda = 2 \Rightarrow BBC_2(G,S) \leq 7$

upper bounds on $BBC_\lambda(G,S)$

Theorem

Let $\lambda \geq 2$ and let G be a split graph with $\chi(G) = k \geq 2$. For every star backbone S of G,

$$BBC_\lambda(G,S) \leq \begin{cases} k+\lambda & \text{if } k=3 \text{ and } \lambda \geq 2 \text{ or } k \geq 4,\ \lambda = 2 \\ k+\lambda-1 & \text{in the other cases.} \end{cases}$$

The bounds are tight.

$\Rightarrow$ additive constant depending on λ

example

split graph G and star backbone S with $\chi(G) = 5$ and $\lambda = 2 \Rightarrow$ $BBC_2(G,S) \leq 7$

upper bounds on $BBC_\lambda(G, M)$

Theorem

Let $\lambda \geq 2$ and let G be a split graph with $\chi(G) = k \geq 2$. For every matching backbone M of G, it holds that $BBC_\lambda(G, M) \leq$

$$\begin{cases} \lambda + 1 & \text{if } k = 2 & (i) \\ k + 1 & \text{if } k \geq 4 \text{ and } \lambda \leq \min\{\frac{k}{2}, \frac{k+5}{3}\} & (ii) \\ k + 2 & \text{if } k = 9 \text{ or } k \geq 11 \text{ and } \frac{k+6}{3} \leq \lambda \leq \lceil \frac{k}{2} \rceil & (iii) \\ \lceil \frac{k}{2} \rceil + \lambda & \text{if } k = 3, 5, 7 \text{ and } \lambda \geq \lceil \frac{k}{2} \rceil & (iv) \\ \lceil \frac{k}{2} \rceil + \lambda + 1 & \text{if } k = 4, 6 \text{ or } k \geq 8 \text{ and } \lambda \geq \lceil \frac{k}{2} \rceil + 1. & (v) \end{cases}$$

The bounds are tight.

$\Rightarrow$ additive constant again!

upper bounds on $BBC_\lambda(G, M)$

Theorem

Let $\lambda \geq 2$ and let G be a split graph with $\chi(G) = k \geq 2$. For every matching backbone M of G, it holds that $BBC_\lambda(G, M) \leq$

$$\begin{cases} \lambda + 1 & \text{if } k = 2 & (i) \\ k + 1 & \text{if } k \geq 4 \text{ and } \lambda \leq \min\{\frac{k}{2}, \frac{k+5}{3}\} & (ii) \\ k + 2 & \text{if } k = 9 \text{ or } k \geq 11 \text{ and } \frac{k+6}{3} \leq \lambda \leq \lceil \frac{k}{2} \rceil & (iii) \\ \lceil \frac{k}{2} \rceil + \lambda & \text{if } k = 3, 5, 7 \text{ and } \lambda \geq \lceil \frac{k}{2} \rceil & (iv) \\ \lceil \frac{k}{2} \rceil + \lambda + 1 & \text{if } k = 4, 6 \text{ or } k \geq 8 \text{ and } \lambda \geq \lceil \frac{k}{2} \rceil + 1. & (v) \end{cases}$$

The bounds are tight.

$\Rightarrow$ additive constant again!

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

- for split graphs
- for large classes of non-split graphs as well, e.g. graphs with $|V(G)| - \omega(G) \leq 2\omega(G)$

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

- for split graphs
- for large classes of non-split graphs as well, e.g. graphs with $|V(G)| - \alpha(G) \leq 2\omega(G)$

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

- for split graphs
- for large classes of non-split graphs as well, e.g. graphs with $|V(G)| - \alpha(G) \leq 2\omega(G)$

old and new upper bounds compared

previous results:

- for any graph G with matching or star backbone
- upper bounds are roughly a factor times $\chi(G)$

new results:

- for split graphs G with matching or star backbone
- upper bounds are additive constant away from $\chi(G)$

When are the new bounds better?

- for split graphs
- for large classes of non-split graphs as well, e.g. graphs with $|V(G)| - \alpha(G) \leq 2\omega(G)$

splitting set

- consider split graph $G = (V, E)$ with matching backbone M
- partition V in a largest clique C and an independent set I
- let $nn(v)$ denote the set of non-neighbors of vertex v
- let $mn(v)$ denote the unique matching neighbor of vertex v

definition

A splitting set S is a subset of I such that $\bigcup_{v \in S} nn(v)$ does not intersect with $\bigcup_{v \in S} mn(v)$.

splitting set

- consider split graph $G = (V, E)$ with matching backbone M
- partition V in a largest clique C and an independent set I
- let $nn(v)$ denote the set of non-neighbors of vertex v
- let $mn(v)$ denote the unique matching neighbor of vertex v

definition

A splitting set S is a subset of I such that $\bigcup_{v \in S} nn(v)$ does not intersect with $\bigcup_{v \in S} mn(v)$.

splitting set

- consider split graph $G = (V, E)$ with matching backbone M
- partition V in a largest clique C and an independent set I
- let $nn(v)$ denote the set of non-neighbors of vertex v
- let $mn(v)$ denote the unique matching neighbor of vertex v

definition

A splitting set S is a subset of I such that $\bigcup_{v \in S} nn(v)$ does not intersect with $\bigcup_{v \in S} mn(v)$.

splitting set

- consider split graph $G = (V, E)$ with matching backbone M
- partition V in a largest clique C and an independent set I
- let $nn(v)$ denote the set of non-neighbors of vertex v
- let $mn(v)$ denote the unique matching neighbor of vertex v

definition

A splitting set S is a subset of I such that $\bigcup_{v \in S} nn(v)$ does not intersect with $\bigcup_{v \in S} mn(v)$.

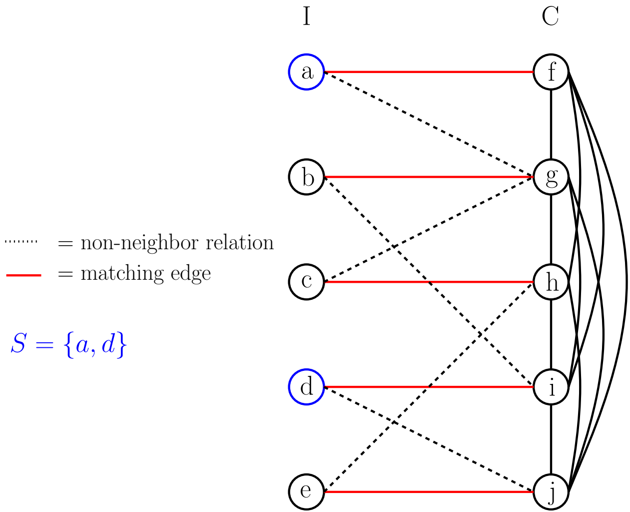
splitting set

- consider split graph $G = (V, E)$ with matching backbone M
- partition V in a largest clique C and an independent set I
- let $nn(v)$ denote the set of non-neighbors of vertex v
- let $mn(v)$ denote the unique matching neighbor of vertex v

definition

A splitting set S is a subset of I such that $\bigcup_{v \in S} nn(v)$ does not intersect with $\bigcup_{v \in S} mn(v)$.

example splitting set



$S = \{a, d\}$

splitting set lemma

Lemma

Given (G, M), let $k = \omega(G) = |C|$ and let $i = |I|$. If every vertex in I has exactly one non-neighbor in C and $\lceil \frac{k}{3} \rceil \geq p$, then (G, M) has a splitting set S with $|S| = p - \frac{k-i}{2}$ such that there are no matching edges between elements of the set of non-neighbors of vertices of S.

constructive proof!

splitting set lemma

Lemma

Given (G, M), let $k = \omega(G) = |C|$ and let $i = |I|$. If every vertex in I has exactly one non-neighbor in C and $\lceil \frac{k}{3} \rceil \geq p$, then (G, M) has a splitting set S with $|S| = p - \frac{k-i}{2}$ such that there are no matching edges between elements of the set of non-neighbors of vertices of S.

constructive proof!

matching case: $k = 10$, $\lambda = 5$

Thm: $BBC_5(G, M) \leq 11$

Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

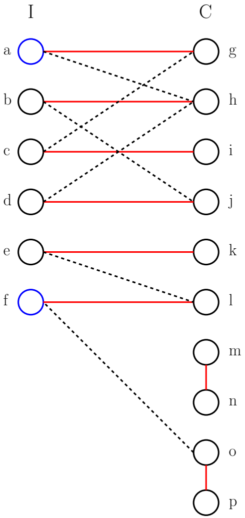
C_3: one ev of each me in C that has no ev in C_1

C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:



matching case: $k = 10$, $\lambda = 5$

Thm: $BBC_5(G, M) \leq 11$

Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

C_3: one ev of each me in C that has no ev in C_1

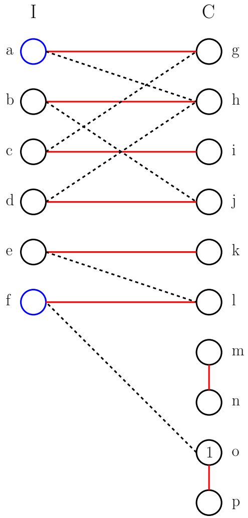
C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1



matching case: $k = 10$, $\lambda = 5$

Thm: $BBC_5(G, M) \leq 11$

Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

C_3: one ev of each me in C that has no ev in C_1

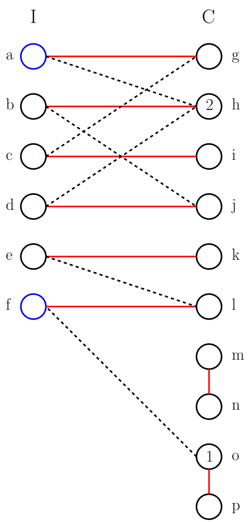
C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2



matching case: $k = 10$, $\lambda = 5$

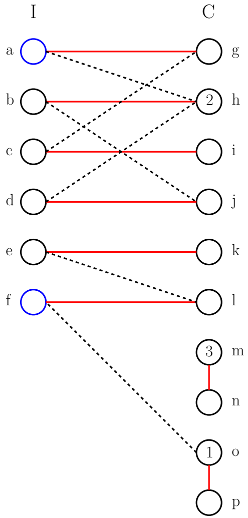
Thm: $BBC_5(G, M) \leq 11$
Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S
C_2: mn in I, nn in S
C_3: one ev of each me in C that has no ev in C_1
C_4: mn in I, no mn or nn in S
C_5: mn in S
C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3



matching case: $k = 10$, $\lambda = 5$

Thm: $BBC_5(G, M) \leq 11$
Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

C_3: one ev of each me in C that has no ev in C_1

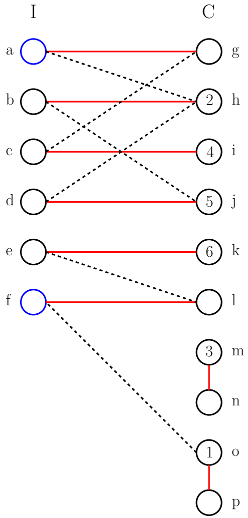
C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3
4. $C_4 = \{i, j, k\}$, color with 4, 5 and 6



matching case: $k = 10$, $\lambda = 5$

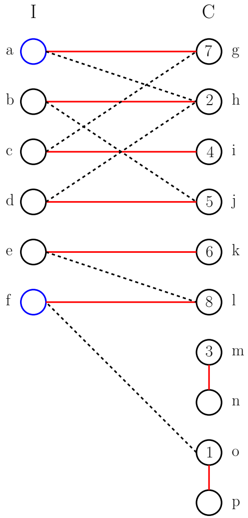
Thm: $BBC_5(G, M) \leq 11$
Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S
C_2: mn in I, nn in S
C_3: one ev of each me in C that has no ev in C_1
C_4: mn in I, no mn or nn in S
C_5: mn in S
C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3
4. $C_4 = \{i, j, k\}$, color with 4, 5 and 6
5. $C_5 = \{g, l\}$, color with 7 and 8



matching case: $k = 10$, $\lambda = 5$

Thm: $BBC_5(G, M) \leq 11$

Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

C_3: one ev of each me in C that has no ev in C_1

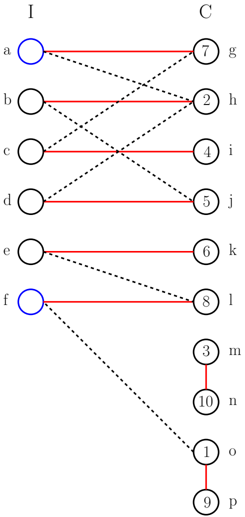
C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3
4. $C_4 = \{i, j, k\}$, color with 4, 5 and 6
5. $C_5 = \{g, l\}$, color with 7 and 8
6. $C_6 = \{n, p\}$, color with 9 and 10



matching case: $k = 10,\ \lambda = 5$

Thm: $BBC_5(G, M) \leq 11$

Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S

C_2: mn in I, nn in S

C_3: one ev of each me in C that has no ev in C_1

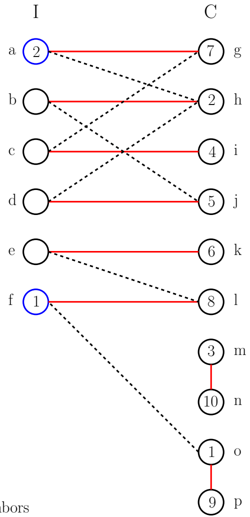
C_4: mn in I, no mn or nn in S

C_5: mn in S

C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3
4. $C_4 = \{i, j, k\}$, color with 4, 5 and 6
5. $C_5 = \{g, l\}$, color with 7 and 8
6. $C_6 = \{n, p\}$, color with 9 and 10
7. color S by assigning same color as non-neighbors



matching case: $k = 10$, $\lambda = 5$

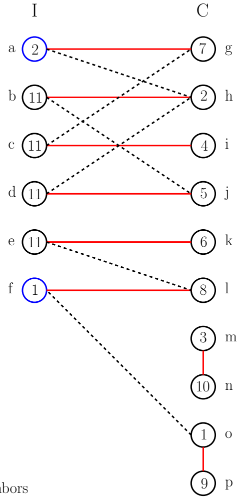
Thm: $BBC_5(G, M) \leq 11$
Lemma: $S = \{a, f\}$

partition of C:

C_1: mn in C, nn in S
C_2: mn in I, nn in S
C_3: one ev of each me in C that has no ev in C_1
C_4: mn in I, no mn or nn in S
C_5: mn in S
C_6: not in C_1 or C_3,mn in C

coloring algorithm:

1. $C_1 = \{o\}$ color with 1
2. $C_2 = \{h\}$ color with 2
3. $C_3 = \{m\}$ color with 3
4. $C_4 = \{i, j, k\}$, color with 4, 5 and 6
5. $C_5 = \{g, l\}$, color with 7 and 8
6. $C_6 = \{n, p\}$, color with 9 and 10
7. color S by assigning same color as non-neighbors
8. color rest of I with 11



directions for further research

- obtain good upper bounds on $BBC_\lambda(G, T)$, where G is a split graph and T is a tree.
- obtain good upper bounds on $BBC_\lambda(G, H)$ for general G and H.

directions for further research

- obtain good upper bounds on $BBC_\lambda(G, T)$, where G is a split graph and T is a tree.
- obtain good upper bounds on $BBC_\lambda(G, H)$ for general G and H.