On Decompositions, Generation Methods, and related concepts in the theory of Matching Covered Graphs

Student: Janmenjaya Panda

Guide: Nishad Kothari

Indian Institute of Technology Madras

May 24, 2024

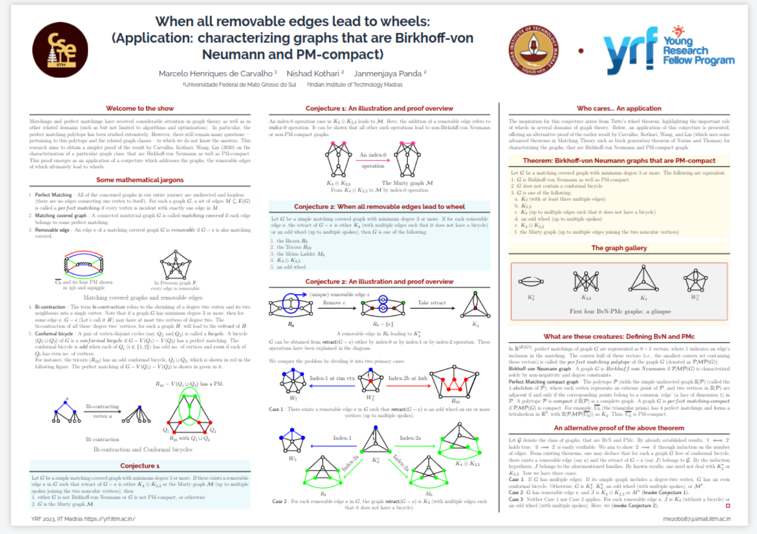
How this proposal came into picture?

How this proposal came into picture?

The story began with:

How this proposal came into picture?

The story began with:



How this proposal came into picture?

How this proposal came into picture?

It was 13/12/2023 and...

How this proposal came into picture?

It was 13/12/2023 and...
en route writing the proof the following graph appreared:

How this proposal came into picture?

It was 13/12/2023 and...
en route writing the proof the following graph appeared:

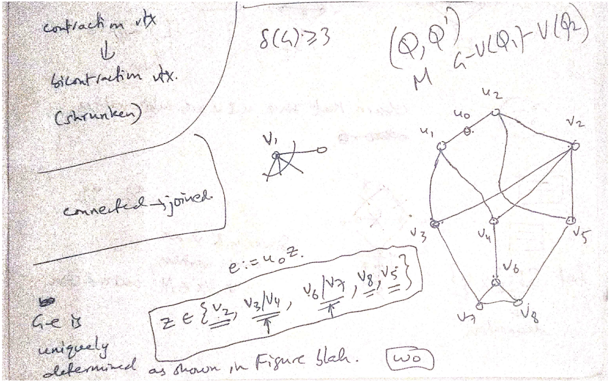


Figure: Check computationally if v_6 and v_7 lie in the same vertex orbit.

Perfect matching: What is that?

Perfect matching: What is that?

All graphs are undirected and loopless.

Perfect matching: What is that?

All graphs are undirected and loopless.

Perfect Matching

Perfect matching: What is that?

All graphs are undirected and loopless.

Perfect Matching

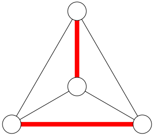
A set M of edges of a graph G is called a *perfect matching* if $|M \cap \partial(v)| = 1$ for each vertex v.

Perfect matching: What is that?

All graphs are undirected and loopless.

Perfect Matching

A set M of edges of a graph G is called a *perfect matching* if $|M \cap \partial(v)| = 1$ for each vertex v.



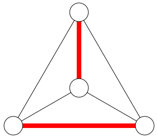
The complete graph K_4 has a perfect matching

Perfect matching: What is that?

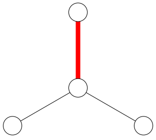
All graphs are undirected and loopless.

Perfect Matching

A set M of edges of a graph G is called a *perfect matching* if $|M \cap \partial(v)| = 1$ for each vertex v.



The complete graph K_4
has a perfect matching



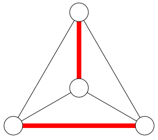
The claw $K_{1,3}$
does not have a perfect matching

Perfect matching: What is that?

All graphs are undirected and loopless.

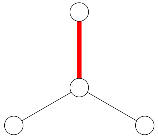
Perfect Matching

A set M of edges of a graph G is called a *perfect matching* if $|M \cap \partial(v)| = 1$ for each vertex v.



The complete graph K_4

has a perfect matching



The claw $K_{1,3}$

does not have a perfect matching

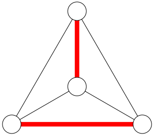
A graph is *matchable* if it has a perfect matching and

Perfect matching: What is that?

All graphs are undirected and loopless.

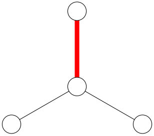
Perfect Matching

A set M of edges of a graph G is called a *perfect matching* if $|M \cap \partial(v)| = 1$ for each vertex v.



The complete graph K_4

has a perfect matching



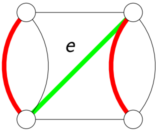
The claw $K_{1,3}$

does not have a perfect matching

A graph is *matchable* if it has a perfect matching and an edge of a graph is *matchable* if it participates in some perfect matching.

Matching covered graphs

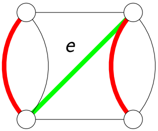
Matching covered graphs



The graph C_4^* has a PM (red).

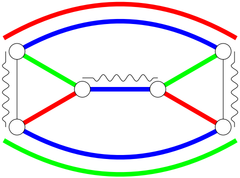
But e does not participate in any PM.

Matching covered graphs



The graph C_4^* has a PM (red).

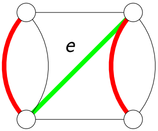
But e does not participate in any PM.



The triangular prism $\overline{C_6}$ and

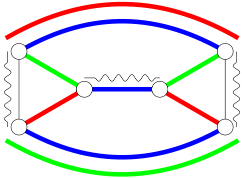
its four PMs (in RGB and squiggle)

Matching covered graphs



The graph C_4^* has a PM (red).

But e does not participate in any PM.



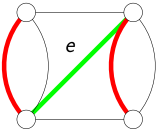
The triangular prism $\overline{C_6}$ and

its four PMs (in RGB and squiggle)

Matching covered graph

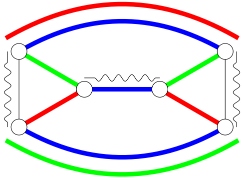
A graph G is *matching covered*, if

Matching covered graphs



The graph C_4^* has a PM (red).

But e does not participate in any PM.



The triangular prism $\overline{C_6}$ and

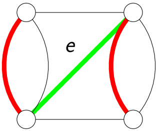
its four PMs (in RGB and squiggle)

Matching covered graph

A graph G is *matching covered*, if

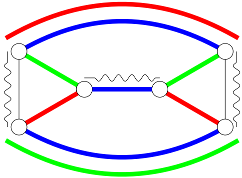
1. G is connected and $|V(G)| \geq 2$

Matching covered graphs



The graph C_4^* has a PM (red).

But e does not participate in any PM.



The triangular prism $\overline{C_6}$ and

its four PMs (in RGB and squiggle)

Matching covered graph

A graph G is *matching covered*, if

1. G is connected and $|V(G)| \geq 2$
2. each edge of G belongs to some perfect matching.

But, why matching covered graphs?

But, why matching covered graphs?

Apart from a reduced dimension for perfect matching polytope, matching covered graphs are well respected creatures.

But, why matching covered graphs?

Apart from a reduced dimension for perfect matching polytope, matching covered graphs are well respected creatures.

- Counting number of PMs

But, why matching covered graphs?

Apart from a reduced dimension for perfect matching polytope, matching covered graphs are well respected creatures.

- Counting number of PMs
- Rich theory

But, why matching covered graphs?

Apart from a reduced dimension for perfect matching polytope, matching covered graphs are well respected creatures.

- Counting number of PMs
- Rich theory
- A lot of well known open problems, such as Barnette's conjecture, Berge-Fulkerson conjecture, and so on

But, why matching covered graphs?

Apart from a reduced dimension for perfect matching polytope, matching covered graphs are well respected creatures.

- Counting number of PMs
- Rich theory
- A lot of well known open problems, such as Barnette's conjecture, Berge-Fulkerson conjecture, and so on

What is this proposal about?

What is this proposal about?

To implement:

What is this proposal about?

To implement:

- Canonical partition

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition
- Notable Families of bricks and braces

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition
- Notable Families of bricks and braces
- Dependency relation and removable classes

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition
- Notable Families of bricks and braces
- Dependency relation and removable classes
- Ear decompositions

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition
- Notable Families of bricks and braces
- Dependency relation and removable classes
- Ear decompositions
- Brick and brace generation

What is this proposal about?

To implement:

- Canonical partition
- Tight cut decomposition
- Notable Families of bricks and braces
- Dependency relation and removable classes
- Ear decompositions
- Brick and brace generation

in the theory of matching covered graphs.

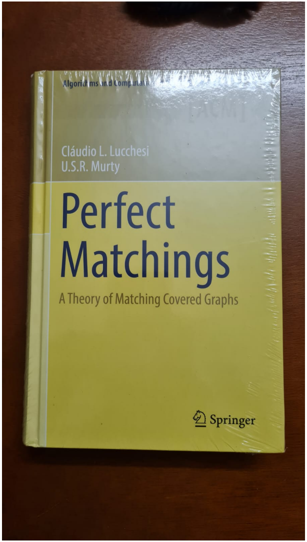
Inspiration

Inspiration

This proposal has been inspired by:

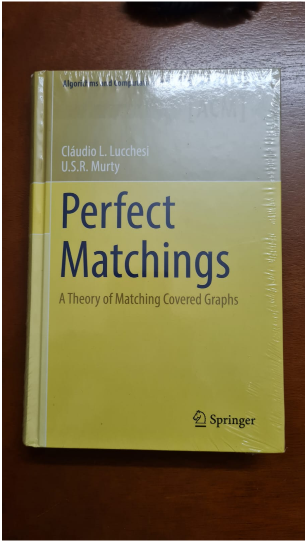
Inspiration

This proposal has been inspired by:



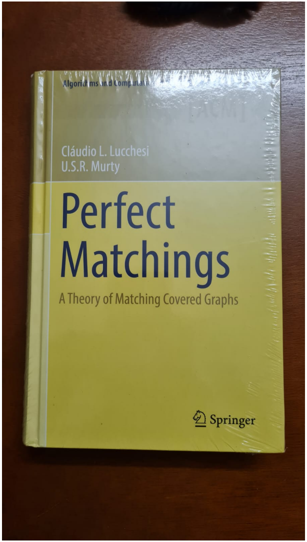
Inspiration

This proposal has been inspired by:



"PERFECT MATCHINGS:

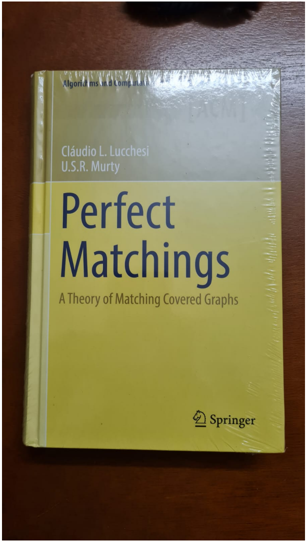
This proposal has been inspired by:



"PERFECT MATCHINGS:
A THEORY OF MATCHING COVERED GRAPHS"

Inspiration

This proposal has been inspired by:



"PERFECT MATCHINGS:
A THEORY OF MATCHING COVERED GRAPHS"
Lucchesi and Murty, 2024

How to check if a graph is matching covered?

How to check if a graph is matching covered?

To recall:

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph

Given a matchable graph G and a perfect matching M.

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph (DIY)

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M,

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph (DIY)

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M, there exists an M-alternating ab-path in G starting and ending with edges in M.

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph (DIY)

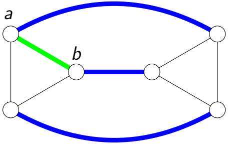
Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M, there exists an M-alternating ab-path in G starting and ending with edges in M.

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph (DIY)

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M, there exists an M-alternating ab-path in G starting and ending with edges in M.



A matchable graph G and a PM M

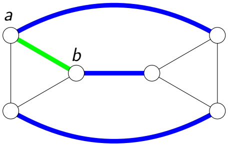
Is ab matchable?

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

Matching covered graph (DIY)

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M, there exists an M-alternating ab-path in G starting and ending with edges in M.



A matchable graph G and a PM M

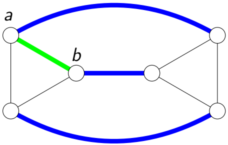
Is ab matchable?

How to check if a graph is matching covered?

To recall: A connected nontrivial graph is *matching covered* if each edge participate in some perfect matching.

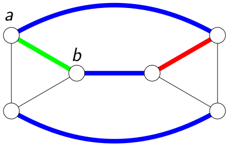
Matching covered graph (DIY)

Given a matchable graph G and a perfect matching M.
G is matching covered, if and only if for each edge ab not in M, there exists an M-alternating ab-path in G starting and ending with edges in M.



A matchable graph G and a PM M

Is ab matchable?



Yes

The req. M-alternating path in $G - ab$

Maximal barriers and canonical Partition

Maximal barriers and canonical Partition

- It turns out that maximal barriers and canonical partition are useful in finding tight cuts.

Maximal barriers and canonical Partition

- It turns out that maximal barriers and canonical partition are useful in finding tight cuts.

Barrier in a matchable graph

For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the # of odd components in graph H.

Maximal barriers and canonical Partition

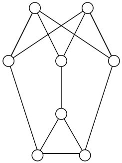
- It turns out that maximal barriers and canonical partition are useful in finding tight cuts.

Barrier in a matchable graph

For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the # of odd components in graph H.



An example of a barrier

- For a graph G, a barrier B is a maximal barrier if C is not a barrier for each C such that $B \subset C \subseteq V$.

Maximal barriers and canonical Partition

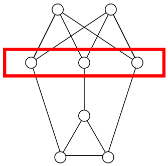
- It turns out that maximal barriers and canonical partition are useful in finding tight cuts.

Barrier in a matchable graph

For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the # of odd components in graph H.



An example of a barrier

Maximal barriers and canonical Partition

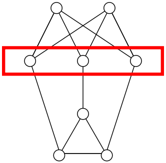
- It turns out that maximal barriers and canonical partition are useful in finding tight cuts.

Barrier in a matchable graph

For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the $\#$ of odd components in graph H.



An example of a barrier

- For a graph G, a barrier B is a maximal barrier if C is not a barrier for each C such that $B \subset C \subseteq V$.

Maximal barriers and canonical Partition

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

The maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its *canonical partition*.

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

The maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its *canonical partition*.

- But, how to compute them in polytime?

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

The maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its *canonical partition*.

- But, how to compute them in polytime?
- We use the following theorem:

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

The maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its *canonical partition*.

- But, how to compute them in polytime?
- We use the following theorem:

Theorem: Lucchesi Murty, 2024

Maximal barriers and canonical Partition

Canonical Partition Theorem; Kotzig 1959

The maximal barriers of a matching covered graph G partition its vertex set, and this partition is called its *canonical partition*.

- But, how to compute them in polytime?
- We use the following theorem:

Theorem: Lucchesi Murty, 2024

Let u and v be any two vertices in a matchable graph G. Then the graph $G - u - v$ is not matchable if and only if there is a barrier of G which contains both u and v.

Tight cut

Tight cut

Before the tight cut decomposition,

Tight cut

Before the tight cut decomposition, we shall know, what a tight cut is.

Tight cut

Before the tight cut decomposition, we shall know, what a tight cut is. Before that, what is a cut?

Before the tight cut decomposition, we shall know, what a tight cut is. Before that, what is a cut?

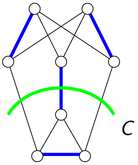
Cut

A subset C of the edge set E of a graph G is called a *cut* if $C = \partial(X)$, for nonempty subset X of the vertex set V.

Before the tight cut decomposition, we shall know, what a tight cut is. Before that, what is a cut?

Cut

A subset C of the edge set E of a graph G is called a *cut* if $C = \partial(X)$, for nonempty subset X of the vertex set V.



An example of a cut C

Tight cut

Before the tight cut decomposition, we shall know, what a tight cut is:

Tight cut

Before the tight cut decomposition, we shall know, what a tight cut is:

Tight cut

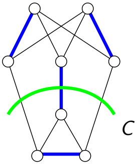
A cut C of a matching covered graph G is tight if $|M \cap C| = 1$, for every perfect matching M of G.

Tight cut

Before the tight cut decomposition, we shall know, what a tight cut is:

Tight cut

A cut C of a matching covered graph G is tight if $|M \cap C| = 1$, for every perfect matching M of G.



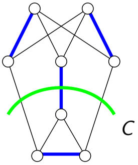
An example of a tight cut C

Tight cut

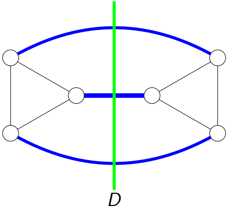
Before the tight cut decomposition, we shall know, what a tight cut is:

Tight cut

A cut C of a matching covered graph G is tight if $|M \cap C| = 1$, for every perfect matching M of G.



An example of a tight cut C



D: A cut that is not tight

Cuts and cut-contractions

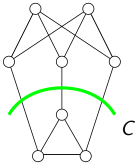
Cuts and cut-contractions

Cuts and cut-contractions

Given any cut $C := \partial(X)$ of a connected graph G, where X is a nonempty proper subset of V , we refer to the two graphs $G/(X \to x)$ and $G/(\overline{X} \to x)$ as the C-contractions of G, where $\overline{X} := V - X$.

Cuts and cut-contractions

Given any cut $C := \partial(X)$ of a connected graph G, where X is a nonempty proper subset of V , we refer to the two graphs $G/(X \to x)$ and $G/(\overline{X} \to x)$ as the C-contractions of G, where $\overline{X} := V - X$.



The graph G with the cut C leads to two C-contractions, that are — K_4 and $K_{3,3}$.

Towards the tight cut decomposition

A small DIY for all of you:

Towards the tight cut decomposition

A small DIY for all of you:

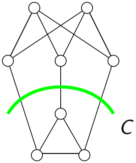
Getting smaller MCGs thru tight cuts

Let G be a graph and C be a tight cut of it. Suppose that G_1 and G_2 are the C-contractions of G. The graph G is matching covered iff each of G_1 and G_2 are matching covered.

A small DIY for all of you:

Getting smaller MCGs thru tight cuts

Let G be a graph and C be a tight cut of it. Suppose that G_1 and G_2 are the C-contractions of G. The graph G is matching covered iff each of G_1 and G_2 are matching covered.



The graph G shown has a non-trivial tight cut C, that leads to two C-contractions, that are — K_4 and $K_{3,3}$, both of which are free of tight cuts.

Bricks and braces

For a matching covered graph that is free of non-trivial tight cuts, it is called a *brick* if it is non-bipartite and a *brace* if it is bipartite.

For example: K_4 is a brick and $K_{3,3}$ is a brace.

A natural question.

A natural question.

Are the results of two TCD procedure unique?

A matching covered graph might have several tight cuts. Will they lead to different list of bricks and braces if we follow two different sequence of tight cut decomposition procedure?

Lovász Tight cut decomposition theorem

Lovász Tight cut decomposition theorem

The unique decomposition theorem: Lovász 1987

Lovász Tight cut decomposition theorem

The unique decomposition theorem: Lovász 1987

Any two applications of the tight cut decomposition procedure to a matching covered graph G produce the same list of bricks and braces, up to multiple edges.

Lovász Tight cut decomposition theorem

The unique decomposition theorem: Lovász 1987

Any two applications of the tight cut decomposition procedure to a matching covered graph G produce the same list of bricks and braces, up to multiple edges.

- The list underlying simple graphs of the bricks and the braces of a matching covered graph are its invariant.

Why is it interesting?

Why is it interesting?

- Several properties concerning problems in matching theory are preserved under tight cut decomposition thus reducing the problems to bricks and braces.
- For example, pfaffian orientations are useful in counting the number of PMs.

Why is it interesting?

- Several properties concerning problems in matching theory are preserved under tight cut decomposition thus reducing the problems to bricks and braces.
- For example, pfaffian orientations are useful in counting the number of PMs.

A cool result

A matching covered graph is pfaffian iff each of its bricks and braces is pfaffian.

Enough! How to implement these?

Enough! How to implement these?

The implementation problem

Enough! How to implement these?

The implementation problem

Given a matching covered graph; either

1. deduce that it is a brick or a brace, or
2. find a nontrivial tight cut.

Enough! How to implement these?

The implementation problem

Given a matching covered graph; either

1. deduce that it is a brick or a brace, or
2. find a nontrivial tight cut.

It turns out that finding arbitrary tight cuts (if any) in a matching covered graph is not an easy task.

How to do in polytime?

How to do in polytime?

Edmonds-Lovász-Pulleyblank 1982

If a matching covered graph has a nontrivial tight cut, then it has a non trivial tight cut that is either a barrier cut, or a two-separation cut.

How to do in polytime?

Edmonds-Lovász-Pulleyblank 1982

If a matching covered graph has a nontrivial tight cut, then it has a non trivial tight cut that is either a barrier cut, or a two-separation cut.

- We will see some example subsequently.

How to do in polytime?

Edmonds-Lovász-Pulleyblank 1982

If a matching covered graph has a nontrivial tight cut, then it has a non trivial tight cut that is either a barrier cut, or a two-separation cut.

- We will see some example subsequently.
- Note that, there might be cuts that are neither a barrier cut nor a 2-separation cut.

How to do in polytime?

Edmonds-Lovász-Pulleyblank 1982

If a matching covered graph has a nontrivial tight cut, then it has a non trivial tight cut that is either a barrier cut, or a two-separation cut.

- We will see some example subsequently.
- Note that, there might be cuts that are neither a barrier cut nor a 2-separation cut. Again, we will see examples as we go on.

What is a barrier cut?

To recall:

What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

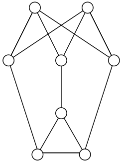
where $o(H)$ denotes the $\#$ of odd components in graph H.

What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the $\#$ of odd components in graph H.



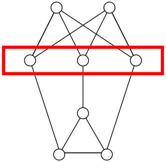
The set shown in red constitutes a barrier; the corresponding nontrivial odd comp. is shown in blue. The tight cut C shown in green is a nontrivial barrier cut.

What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the # of odd components in graph H.



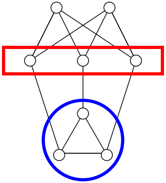
The set shown in red constitutes a barrier; the corresponding nontrivial odd comp. is shown in blue. The tight cut C shown in green is a nontrivial barrier cut.

What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the $\#$ of odd components in graph H.



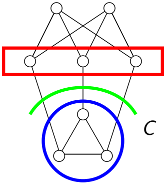
The set shown in red constitutes a barrier; the corresponding nontrivial odd comp. is shown in blue. The tight cut C shown in green is a nontrivial barrier cut.

What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the # of odd components in graph H.

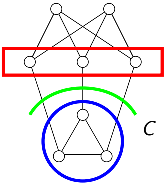


What is a barrier cut?

To recall: For a matchable graph G, a subset B of the vertex set V is a barrier if

$$|B| = o(G - B)$$

where $o(H)$ denotes the $\#$ of odd components in graph H.



The set shown in red constitutes a barrier; the corresponding nontrivial odd comp. is shown in blue. The tight cut C shown in green is a nontrivial barrier cut.

What if there are no nontrivial barrier cuts?

What if there are no nontrivial barrier cuts?

Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers,

What if there are no nontrivial barrier cuts?

Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers, that is — $\emptyset$ and singleton vertex set are the only barriers of a bicritical graph.

Clearly, bricks are bicritical, but is each bicritical graph a brick?

What if there are no nontrivial barrier cuts?

Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers, that is — $\emptyset$ and singleton vertex set are the only barriers of a bicritical graph.

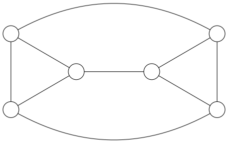
Clearly, bricks are bicritical, but is each bicritical graph a brick?

What if there are no nontrivial barrier cuts?

Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers, that is — $\emptyset$ and singleton vertex set are the only barriers of a bicritical graph.

Clearly, bricks are bicritical, but is each bicritical graph a brick?



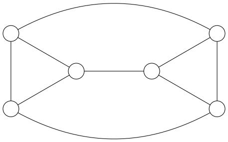
A bicritical graph that is a brick

What if there are no nontrivial barrier cuts?

Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers, that is — $\emptyset$ and singleton vertex set are the only barriers of a bicritical graph.

Clearly, bricks are bicritical, but is each bicritical graph a brick?

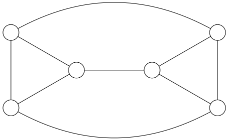


A bicritical graph that is a brick

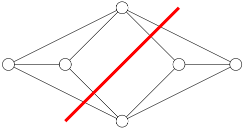
Bicritical graph

A matching covered graph is *bicritical* if it is free of nontrivial barriers, that is — $\emptyset$ and singleton vertex set are the only barriers of a bicritical graph.

Clearly, bricks are bicritical, but is each bicritical graph a brick?



A bicritical graph that is a brick



A bicritical graph that is not a brick

2-separation cut

2-separation cut

- Consider a matching covered graph G

2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.

2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.
- What you can say about the # of odd components of $G - u - v$?

2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.
- What you can say about the # of odd components of $G - u - v$?
- By Tutte's theorem, either it is 2 or 0.

2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.
- What you can say about the $\#$ of odd components of $G - u - v$?
- By Tutte's theorem, either it is 2 or 0.
- If it is two and we have a nontrivial odd component, then we have a nontrivial barrier cut.

2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.
- What you can say about the # of odd components of $G - u - v$?
- By Tutte's theorem, either it is 2 or 0.
- If it is two and we have a nontrivial odd component, then we have a nontrivial barrier cut.
- If it is zero, then we will get a bunch of even components (at least two).

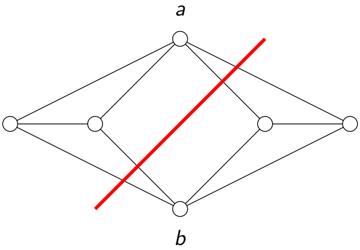
2-separation cut

- Consider a matching covered graph G
- As the name suggest, we have a 2 vertex cut $\{u, v\}$ in G.
- What you can say about the # of odd components of $G - u - v$?
- By Tutte's theorem, either it is 2 or 0.
- If it is two and we have a nontrivial odd component, then we have a nontrivial barrier cut.
- If it is zero, then we will get a bunch of even components (at least two).

2-separation cut

Bricks are free of 2 separation cuts as well. Morevoer, they are 3-connected.

2-separation cut



The illustration of a 2-sepration cut

Bricks are free of 2 separation cuts as well. Morevoer, they are 3-connected.

Are tight cuts always a barrier cut or a 2-sep cut?

- No,

Are tight cuts always a barrier cut or a 2-sep cut?

- No, not all tight cuts are either barrier cuts or 2-separation cuts.

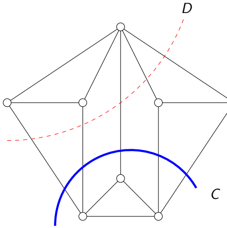


Figure: The cut C (shown in bold blue) is neither a barrier cut nor a 2-separation cut, but D (shown in dashed red) is a 2-separation cut

Are tight cuts always a barrier cut or a 2-sep cut?

- No, not all tight cuts are either barrier cuts or 2-separation cuts.

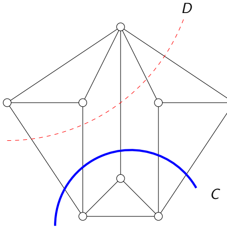


Figure: The cut C (shown in bold blue) is neither a barrier cut nor a 2-separation cut, but D (shown in dashed red) is a 2-separation cut

- Finding an arbitrary tight cut is difficult, but finding a barrier cut or a 2-separation cut is polytime computable.

Are tight cuts always a barrier cut or a 2-sep cut?

- No, not all tight cuts are either barrier cuts or 2-separation cuts.

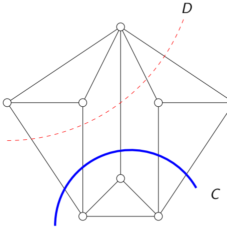


Figure: The cut C (shown in bold blue) is neither a barrier cut nor a 2-separation cut, but D (shown in dashed red) is a 2-separation cut

- Finding an arbitrary tight cut is difficult, but finding a barrier cut or a 2-separation cut is polytime computable.
- The ELP theorem guarantees the existence of either a nontrivial barrier or a non trivial 2-separation cut, if at all the matching covered graph has a nontrivial tight cut.

Implementing tight cut decomposition procedure

Implementing tight cut decomposition procedure

MCG G

Implementing tight cut decomposition procedure

$$\begin{array}{ccc} & \text{MCG } G & \\ \swarrow & & \\ G \text{ is non-bipartite} & & \end{array}$$

Implementing tight cut decomposition procedure

$$
\begin{array}{ccc}
 & & \text{MCG } G \\
 & \swarrow & \\
G \text{ is non-bipartite} & & \\
\downarrow & & \\
\text{Is } G \text{ a brick?} & &
\end{array}
$$

Implementing tight cut decomposition procedure

MCG G

G is non-bipartite

Is G a brick?

Yes

Implementing tight cut decomposition procedure

MCG G

$\swarrow$

G is non-bipartite

$\downarrow$

Is G a brick?

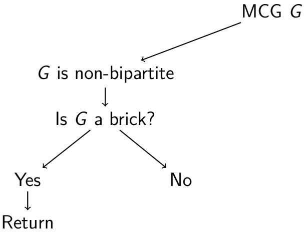
$\swarrow$

Yes

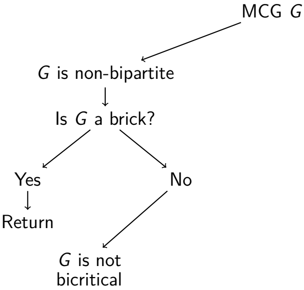
$\downarrow$

Return

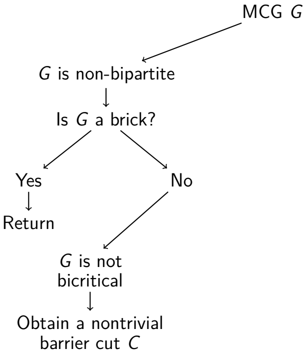
Implementing tight cut decomposition procedure



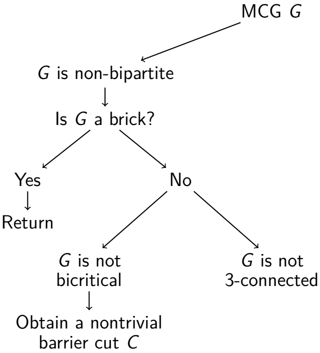
Implementing tight cut decomposition procedure



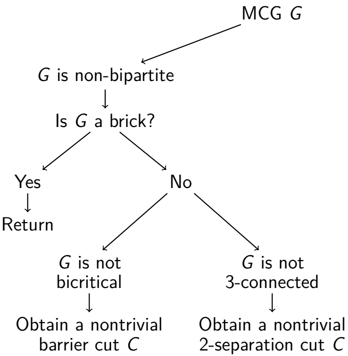
Implementing tight cut decomposition procedure



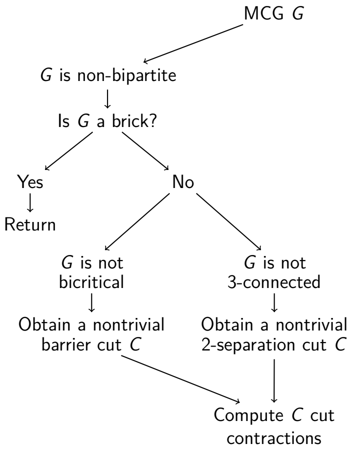
Implementing tight cut decomposition procedure



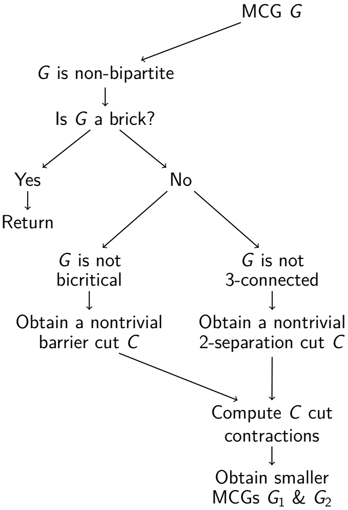
Implementing tight cut decomposition procedure



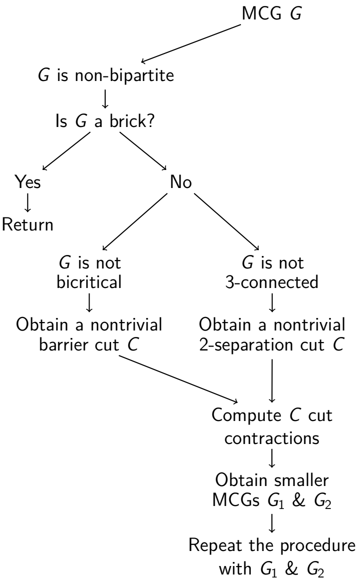
Implementing tight cut decomposition procedure



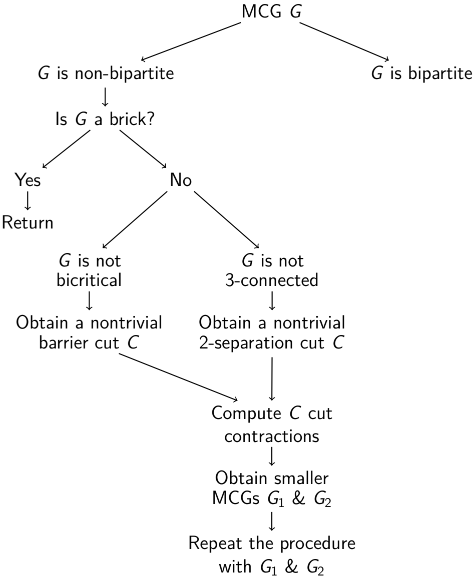
Implementing tight cut decomposition procedure



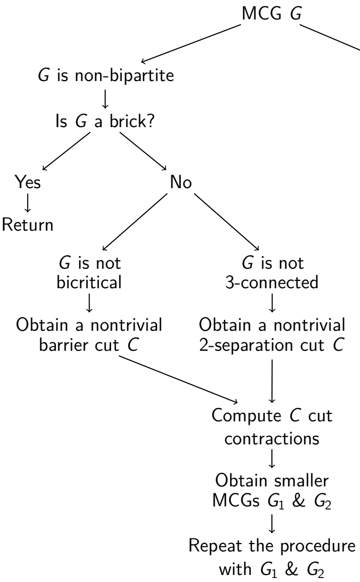
Implementing tight cut decomposition procedure



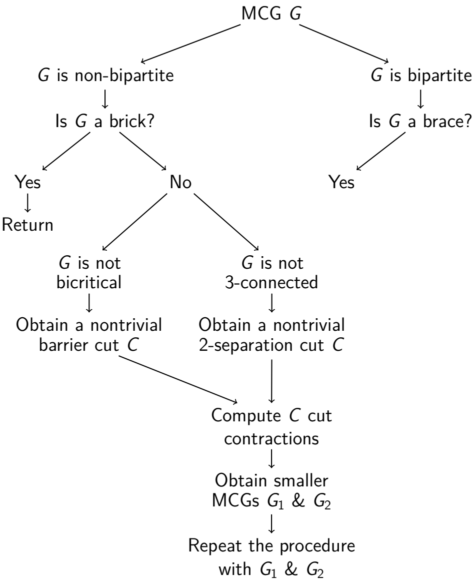
Implementing tight cut decomposition procedure



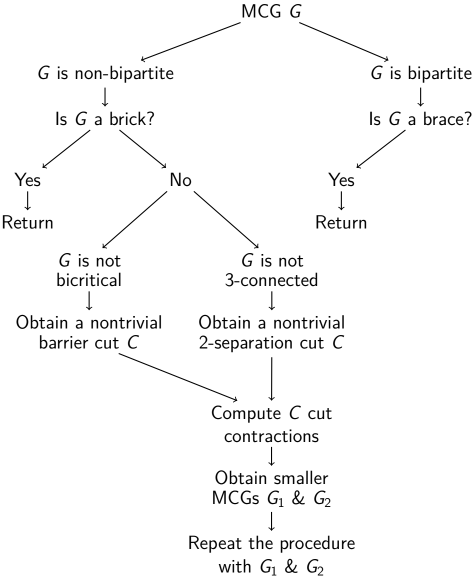
Implementing tight cut decomposition procedure



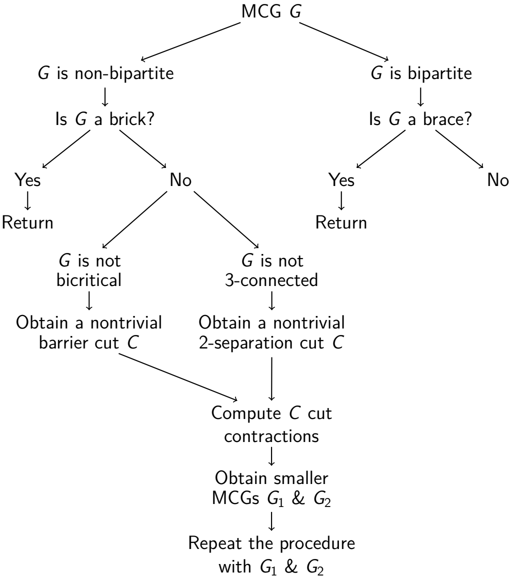
Implementing tight cut decomposition procedure



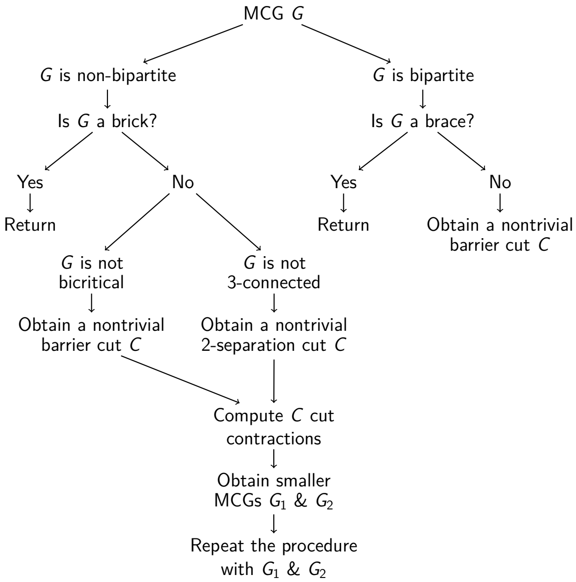
Implementing tight cut decomposition procedure



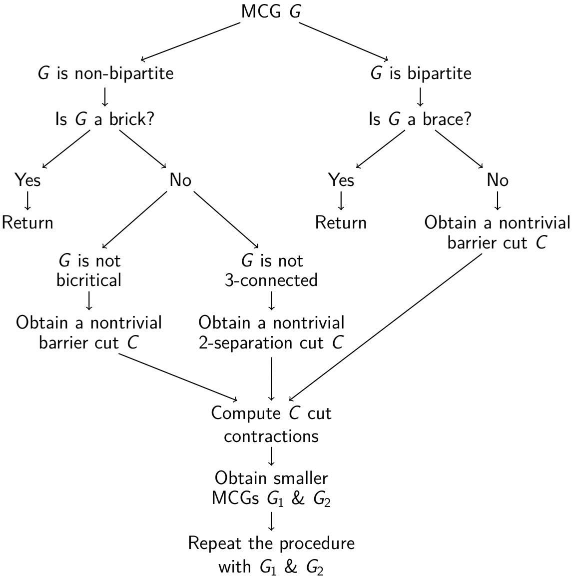
Implementing tight cut decomposition procedure



Implementing tight cut decomposition procedure



Implementing tight cut decomposition procedure



Notable families of bricks and braces

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph
- Möbius Ladder graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph
- Möbius Ladder graph
- Biwheel graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph
- Möbius Ladder graph
- Biwheel graph
- Truncated Biwheel graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph
- Möbius Ladder graph
- Biwheel graph
- Truncated Biwheel graph
- Staircase graph

Notable families of bricks and braces

These families play significant roles in the works of McCuaig and Norine and Thomas.

- Circular Ladder graph
- Wheel graph
- Möbius Ladder graph
- Biwheel graph
- Truncated Biwheel graph
- Staircase graph

Notable families: Circular Ladder graph

Circular Ladder graph

Notable families: Circular Ladder graph

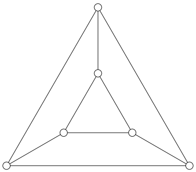
Circular Ladder graph

The *Circular ladder graph*, aka *Prism graph* $\mathbb{P}_{2n}$, for $n \geqslant 3$, is the graph obtained from two disjoint cycles

$$u_1 u_2 u_3 \dots u_n u_1 \text{ and } v_1 v_2 \dots v_n v_1$$

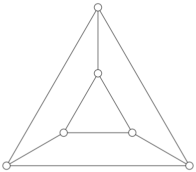
of length n by the addition of the n edges $u_i v_i$, $i = 1, 2, \dots, n$.

Notable families: Circular Ladder graph

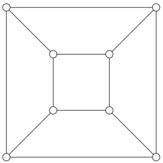


(a) CircularLadderGraph(3)

Notable families: Circular Ladder graph

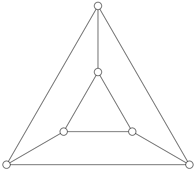


(a) CircularLadderGraph(3)

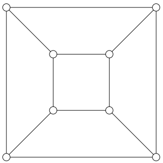


(b) CirclularLadderGraph(4)

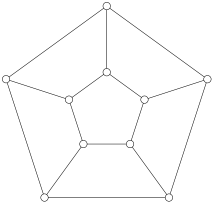
Notable families: Circular Ladder graph



(a) CircularLadderGraph(3)

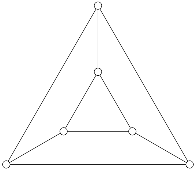


(b) CirclularLadderGraph(4)

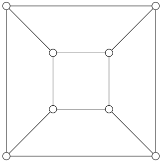


(c) CircularLadderGraph(5)

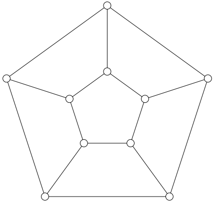
Notable families: Circular Ladder graph



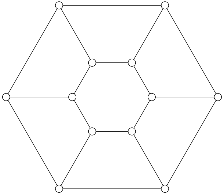
(a) CircularLadderGraph(3)



(b) CirclularLadderGraph(4)

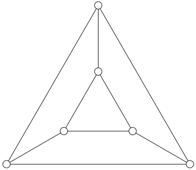


(c) CircularLadderGraph(5)

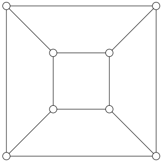


(d) CirclularLadderGraph(6)

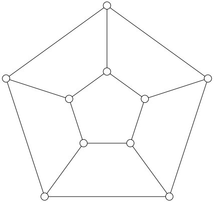
Notable families: Circular Ladder graph



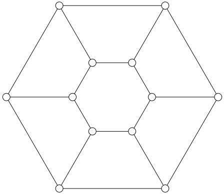
(a) CircularLadderGraph(3)



(b) CirclularLadderGraph(4)



(c) CircularLadderGraph(5)



(d) CirclularLadderGraph(6)

Figure: The family of circular ladder graphs

Notable families: Wheel graph

Notable families: Wheel graph

Wheel graph

Notable families: Wheel graph

Wheel graph

The *Wheel graph* $\mathbb{W}_n$, for $n \geqslant 4$, is the graph obtained from a cycle

$$v_1 v_2 \ldots v_{n-1} v_1$$

of length $n-1$, called the *rim* of $\mathbb{W}_n$, by the addition a universal vertex, called *hub* h.

Wheel graph

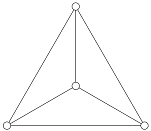
The *Wheel graph* $\mathbb{W}_n$, for $n \geqslant 4$, is the graph obtained from a cycle

$$v_1 v_2 \ldots v_{n-1} v_1$$

of length $n-1$, called the *rim* of $\mathbb{W}_n$, by the addition a universal vertex, called *hub* h. Note that wheels on an even number of vertices, aka W_{2k} for some $k \geqslant 2$, are matching covered — in fact, bricks.

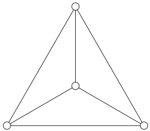
Notable families: Wheel graph

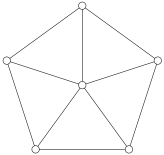
Notable families: Wheel graph



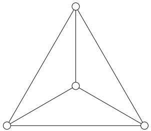
(a) WheelGraph(4)

Notable families: Wheel graph

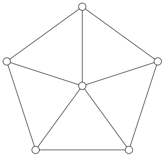

(a) WheelGraph(4)


(b) WheelGraph(6)

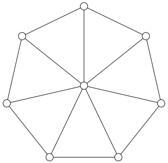
Notable families: Wheel graph



(a) WheelGraph(4)

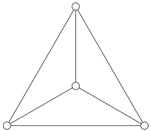


(b) WheelGraph(6)

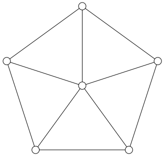


(c) WheelGraph(8)

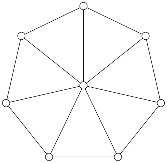
Notable families: Wheel graph



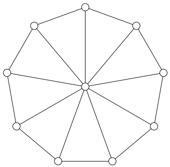
(a) WheelGraph(4)



(b) WheelGraph(6)

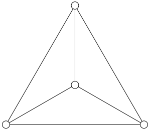


(c) WheelGraph(8)

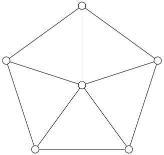


(d) WheelGraph(10)

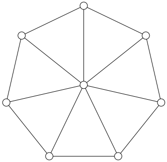
Notable families: Wheel graph



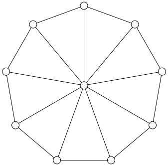
(a) WheelGraph(4)



(b) WheelGraph(6)



(c) WheelGraph(8)



(d) WheelGraph(10)

Figure: The family of wheel graphs, that are matching covered

Notable families: Möbius Ladder graph

Notable families: Möbius Ladder graph

Möbius Ladder graph

Notable families: Möbius Ladder graph

Möbius Ladder graph

The *Möbius ladder graph* $\mathbb{M}_{2n}$, for $n \geqslant 2$, is the graph obtained from a cycle

$$v_1 v_2 \dots v_{2n} v_1$$

of length $2n$ by the addition of the n chords $v_i v_{i+n}$, for $1 \leqslant i \leqslant n$, joining antipodal pairs of vertices of the cycle.

Möbius Ladder graph

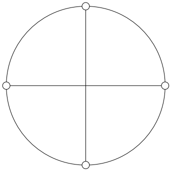
The *Möbius ladder graph* $\mathbb{M}_{2n}$, for $n \geqslant 2$, is the graph obtained from a cycle

$$v_1 v_2 \dots v_{2n} v_1$$

of length $2n$ by the addition of the n chords $v_i v_{i+n}$, for $1 \leqslant i \leqslant n$, joining antipodal pairs of vertices of the cycle.

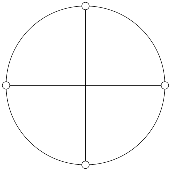
Notable families: Möbius Ladder graph

Notable families: Möbius Ladder graph

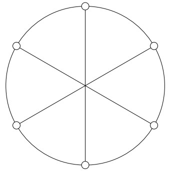


(a) MöbiusLadderGraph(2)

Notable families: Möbius Ladder graph

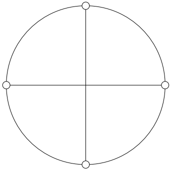


(a) MöbiusLadderGraph(2)

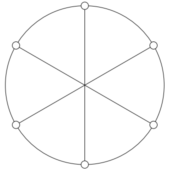


(b) MöbiusLadderGraph(3)

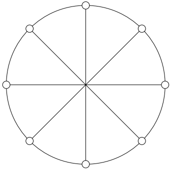
Notable families: Möbius Ladder graph



(a) MöbiusLadderGraph(2)

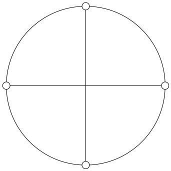


(b) MöbiusLadderGraph(3)

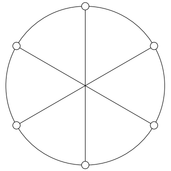


(c) MöbiusLadderGraph(4)

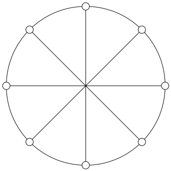
Notable families: Möbius Ladder graph



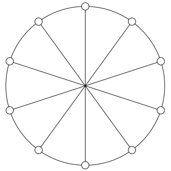
(a) MöbiusLadderGraph(2)



(b) MöbiusLadderGraph(3)

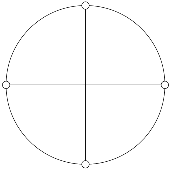


(c) MöbiusLadderGraph(4)

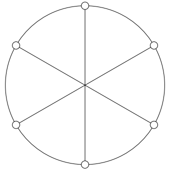


(d) MöbiusLadderGraph(5)

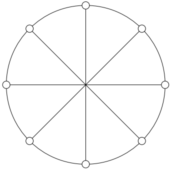
Notable families: Möbius Ladder graph



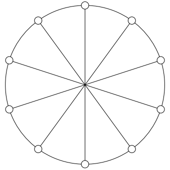
(a) MöbiusLadderGraph(2)



(b) MöbiusLadderGraph(3)



(c) MöbiusLadderGraph(4)



(d) MöbiusLadderGraph(5)

Figure: The family of möbius ladder graphs

Notable families: Biwheel graph

Notable families: Biwheel graph

Biwheel graph

Notable families: Biwheel graph

Biwheel graph

The *Biwheel graph* $\mathbb{B}_{2n}$, for $n \geqslant 4$, is the bipartite graph obtained from a cycle

$$v_1 v_2 \dots v_{2n-2} v_1$$

of length $2n-2$, called the *rim* of $\mathbb{B}_{2n}$, by the addition of two vertices, h_1 and h_2, called the *hubs* of $\mathbb{B}_{2n}$, and by the addition of edges $h_1 v_1, h_1 v_3, \dots, h_1 v_{2n-3}$ and edges $h_2 v_2, h_2 v_4, \dots, h_2 v_{2n-2}$.

Notable families: Biwheel graph

Biwheel graph

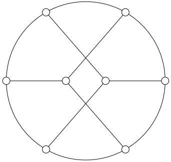
The *Biwheel graph* $\mathbb{B}_{2n}$, for $n \geqslant 4$, is the bipartite graph obtained from a cycle

$$v_1 v_2 \dots v_{2n-2} v_1$$

of length $2n-2$, called the *rim* of $\mathbb{B}_{2n}$, by the addition of two vertices, h_1 and h_2, called the *hubs* of $\mathbb{B}_{2n}$, and by the addition of edges $h_1 v_1, h_1 v_3, \dots, h_1 v_{2n-3}$ and edges $h_2 v_2, h_2 v_4, \dots, h_2 v_{2n-2}$.

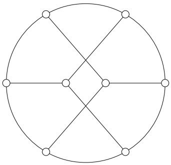
Notable families: Biwheel graph

Notable families: Biwheel graph

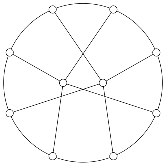


(a) BiwheelGraph(8)

Notable families: Biwheel graph

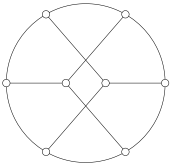


(a) BiwheelGraph(8)

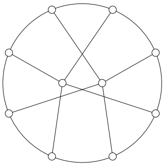


(b) BiwheelGraph(10)

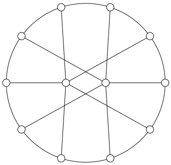
Notable families: Biwheel graph



(a) BiwheelGraph(8)

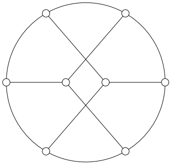


(b) BiwheelGraph(10)

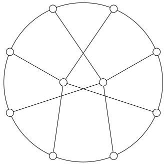


(c) BiwheelGraph(12)

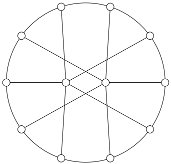
Notable families: Biwheel graph



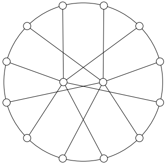
(a) BiwheelGraph(8)



(b) BiwheelGraph(10)

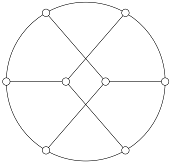


(c) BiwheelGraph(12)

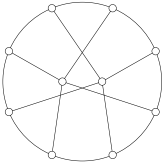


(d) BiwheelGraph(14)

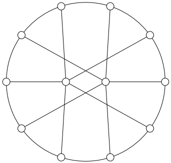
Notable families: Biwheel graph



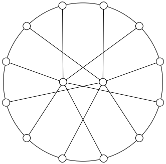
(a) BiwheelGraph(8)



(b) BiwheelGraph(10)



(c) BiwheelGraph(12)



(d) BiwheelGraph(14)

Figure: The family of biwheel graphs

Notable families: Truncated Biwheel graph

Notable families: Truncated Biwheel graph

Truncated Biwheel graph

Notable families: Truncated Biwheel graph

Truncated Biwheel graph

The *Truncated biwheel graph* $\mathbb{T}_{2n}$, for $n \geqslant 3$, is the graph obtained from a path $v_1 v_2 \ldots v_{2n-2}$ of length $2n-3$, by the addition of two vertices, h_1 and h_2, and by the addition of edges $h_1 v_1, h_1 v_3, \ldots, h_1 v_{2n-3}$, edges $h_2 v_2, h_2 v_4, \ldots, h_2 v_{2n-2}$ and edges $h_1 v_{2n-2}$ and $h_2 v_1$.

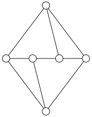
Notable families: Truncated Biwheel graph

Truncated Biwheel graph

The *Truncated biwheel graph* $\mathbb{T}_{2n}$, for $n \geqslant 3$, is the graph obtained from a path $v_1 v_2 \ldots v_{2n-2}$ of length $2n-3$, by the addition of two vertices, h_1 and h_2, and by the addition of edges $h_1 v_1, h_1 v_3, \ldots, h_1 v_{2n-3}$, edges $h_2 v_2, h_2 v_4, \ldots, h_2 v_{2n-2}$ and edges $h_1 v_{2n-2}$ and $h_2 v_1$.

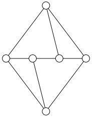
Notable families: Truncated Biwheel graph

Notable families: Truncated Biwheel graph

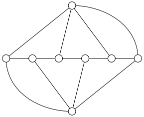


(a) TruncatedBiwheelGraph(3)

Notable families: Truncated Biwheel graph

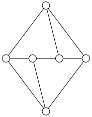


(a) TruncatedBiwheelGraph(3)

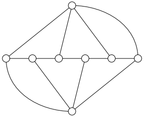


(b) TruncatedBiwheelGraph(4)

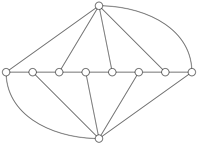
Notable families: Truncated Biwheel graph



(a) TruncatedBiwheelGraph(3)

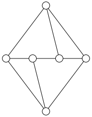


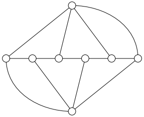
(b) TruncatedBiwheelGraph(4)

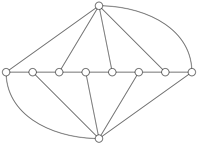


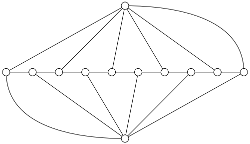
(c) TruncatedBiwheelGraph(5)

Notable families: Truncated Biwheel graph

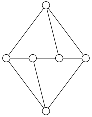

(a) TruncatedBiwheelGraph(3)


(b) TruncatedBiwheelGraph(4)

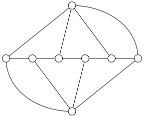

(c) TruncatedBiwheelGraph(5)


(d) TruncatedBiwheelGraph(6)

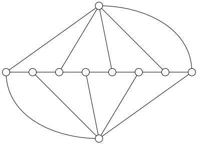
Notable families: Truncated Biwheel graph



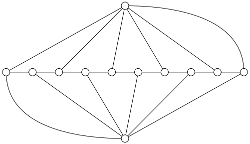
(a) TruncatedBiwheelGraph(3)



(b) TruncatedBiwheelGraph(4)



(c) TruncatedBiwheelGraph(5)



(d) TruncatedBiwheelGraph(6)

Figure: The family of truncated biwheel graphs

Notable families: Staircase graph

Notable families: Staircase graph

Staircase graph

Notable families: Staircase graph

Staircase graph

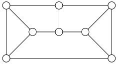
Consider the Ladder graph $\mathbb{L}_{2n-2}$ obtained from two disjoint paths $u_1 u_2 \ldots u_{n-1}$ and $v_1 v_2 \ldots v_{n-1}$ by adding, for $1 \leqslant i \leqslant n-1$, an edge joining u_i and v_i. For $n \geqslant 3$, the *Staircase graph* $\mathbb{S}_{2n}$ is the graph obtained from $\mathbb{L}_{2n-2}$ by adding two new vertices x and y, and then joining x to u_1 and v_1, the vertex y to u_{n-1} and v_{n-1}, and x and y to each other.

Staircase graph

Consider the Ladder graph $\mathbb{L}_{2n-2}$ obtained from two disjoint paths $u_1 u_2 \ldots u_{n-1}$ and $v_1 v_2 \ldots v_{n-1}$ by adding, for $1 \leqslant i \leqslant n-1$, an edge joining u_i and v_i. For $n \geqslant 3$, the *Staircase graph* $\mathbb{S}_{2n}$ is the graph obtained from $\mathbb{L}_{2n-2}$ by adding two new vertices x and y, and then joining x to u_1 and v_1, the vertex y to u_{n-1} and v_{n-1}, and x and y to each other.

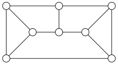
Notable families: Staircase graph

Notable families: Staircase graph

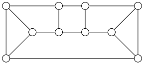


(a) StaircaseGraph(4)

Notable families: Staircase graph

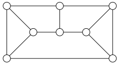


(a) StaircaseGraph(4)

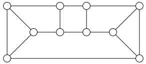


(b) StaricaseGraph(5)

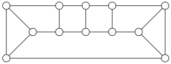
Notable families: Staircase graph



(a) StaircaseGraph(4)

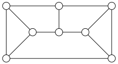


(b) StaricaseGraph(5)

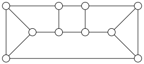


(c) StaircaseGraph(6)

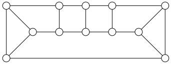
Notable families: Staircase graph



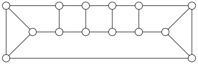
(a) StaircaseGraph(4)



(b) StaricaseGraph(5)

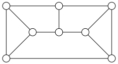


(c) StaircaseGraph(6)

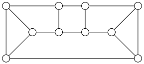


(d) StaircaseGraph(7)

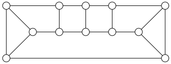
Notable families: Staircase graph



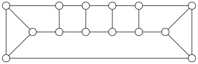
(a) StaircaseGraph(4)



(b) StaricaseGraph(5)



(c) StaircaseGraph(6)



(d) StaircaseGraph(7)

Figure: The family of staircase graphs

An overview

An overview

Module	Method	Status

An overview

	Module	Method	Status
A	Fundamentals	M_alternating_tree_mark()	Implemented
		is_matching_covered()	Implemented
		is_bicritical()	Implemented

An overview

	Module	Method	Status
A	Fundamentals	M_alternating_tree_mark()	Implemented
		is_matching_covered()	Implemented
		is_bicritical()	Implemented
B	Canonical Partition	maximal_barrier()	Implemented
		canonical_partition()	Implemented

An overview

	Module	Method	Status
A	Fundamentals	M_alternating_tree_mark()	Implemented
		is_matching_covered()	Implemented
		is_bicritical()	Implemented
B	Canonical Partition	maximal_barrier()	Implemented
		canonical_partition()	Implemented
C	Tight cut decomposition	is_brick()	Implemented
		is_brace()	Implemented
		tight_cut_decomposition()	Implemented
		bricks_and_braces()	Implemented
		number_of_bricks()	Implemented
		number_of_braces()	Implemented
		number_of_petersen_bricks()	Implemented

An overview (continued)

An overview (continued)

Module	Method	Status

An overview (continued)

	Module	Method	Status
D	Notable Families	MöbiusLadderGraph()	Implemented
		StaircaseGraph()	Implemented
		BiwheelGraph()	Ongoing
		TruncatedBiwheelGraph()	Implemented

An overview (continued)

	Module	Method	Status
D	Notable Families	MöbiusLadderGraph()	Implemented
		StaircaseGraph()	Implemented
		BiwheelGraph()	Ongoing
		TruncatedBiwheelGraph()	Implemented
E	Dependency relation	is_removable_edge()	Implemented
		removable_edges()	Implemented
		is_removable_doubleton()	Implemented
		removable_doubletons()	Implemented

An overview (continued)

An overview (continued)

Module	Method	Status

An overview (continued)

	Module	Method	Status
F	Ear decomposition	retract()	Ongoing
		matching_covered_ear_decomposition()	To be done
			To be done
		optimal_ear()	To be done
		optimal_ear_decomposition()	To be done

An overview (continued)

	Module	Method	Status
F	Ear decomposition	retract()	Ongoing
		matching_covered_ear_decomposition()	To be done
		optimal_ear()	To be done
		optimal_ear_decomposition()	To be done
G	Brick and brace generation	is_strictly_thin_edges()	To be done
		mccuaig_brace_decomposition()	To be done
		norine_thomas_brick_decomposition()	To be done

What's next?

What's next?

- One major goal is to make these implementations accessible for everyone.

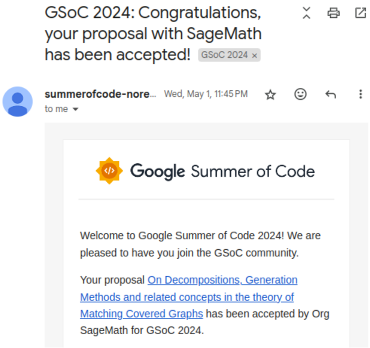
What's next?

- One major goal is to make these implementations accessible for everyone.

- Proposal to SageMath (via Google Summer of Code 2024).

What's next?

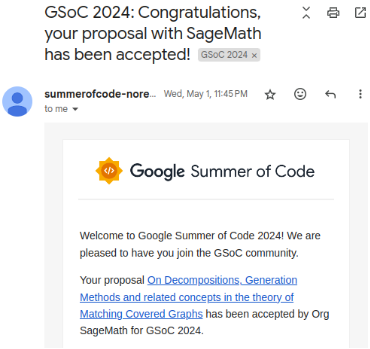
- One major goal is to make these implementations accessible for everyone.



- Proposal to SageMath (via Google Summer of Code 2024).

What’s next?

- One major goal is to make these implementations accessible for everyone.


GSoC 2024: Congratulations, your proposal with SageMath has been accepted! GSoC 2024 ×

summerofcode-nore... Wed, May 1, 11:45 PM
to me

Google Summer of Code

Welcome to Google Summer of Code 2024! We are pleased to have you join the GSoC community.

Your proposal On Decompositions, Generation Methods and related concepts in the theory of Matching Covered Graphs has been accepted by Org SageMath for GSoC 2024.

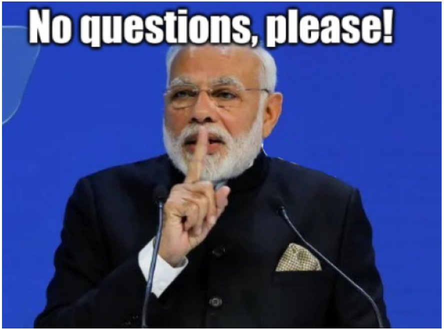
- Should be accessible by everyone from 26th of August 2024.

Yes!

Yes! you folks made my day.

Yes! you folks made my day.

Yes! you folks made my day.



Yes! you folks made my day.

Yes! you folks made my day.