ASBG-Colourings of Unicyclic and Cycle-Edge-Disjoint Graphs

Cian O'Brien
Rachel Quinlan and Kevin Jennings

Postgraduate Modelling Research Group
National University of Ireland, Galway

c.obrien40@nuigalway.ie

February 8th, 2019

Alternating Signed Bipartite Graphs

An *Alternating Signed Bipartite Graph (ASBG)* is a graphical representation of Alternating Sign Matrices introduced by Richard Brualdi et al [1].

Alternating Signed Bipartite Graphs

An *Alternating Signed Bipartite Graph (ASBG)* is a graphical representation of Alternating Sign Matrices introduced by Richard Brualdi et al [1].

A bipartite graph G is an ASBG if it is balanced and each edge of G is coloured blue or red such that there is an *allowable ordering* of the vertices in each part of the bipartition.

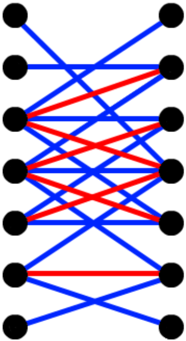
Alternating Signed Bipartite Graphs

An *Alternating Signed Bipartite Graph (ASBG)* is a graphical representation of Alternating Sign Matrices introduced by Richard Brualdi et al [1].

A bipartite graph G is an ASBG if it is balanced and each edge of G is coloured blue or red such that there is an *allowable ordering* of the vertices in each part of the bipartition.

An ordering of the vertices is allowable if the vertices of each part can be embedded in that order on two parallel lines in the plane such that the edges incident with each vertex alternate in colour (beginning and ending with blue) in that embedding.

Alternating Signed Bipartite Graphs



Question: Given a graph G without coloured edges, how can we tell if its edges have a red/blue colouring c such that G^c is an ASBG?

Question: Given a graph G without coloured edges, how can we tell if its edges have a red/blue colouring c such that G^c is an ASBG?

We call such a colouring c an *ASBG-colouring*, and we say that G is *ASBG-colourable*.

ASBG-Colourings

Question: Given a graph G without coloured edges, how can we tell if its edges have a red/blue colouring c such that G^c is an ASBG?

We call such a colouring c an *ASBG-colouring*, and we say that G is *ASBG-colourable*.

Some necessary (but not sufficient) conditions for a given graph G to have an ASBG-colouring c are:

ASBG-Colourings

Question: Given a graph G without coloured edges, how can we tell if its edges have a red/blue colouring c such that G^c is an ASBG?

We call such a colouring c an *ASBG-colouring*, and we say that G is *ASBG-colourable*.

Some necessary (but not sufficient) conditions for a given graph G to have an ASBG-colouring c are:

- G must be bipartite and balanced;

ASBG-Colourings

Question: Given a graph G without coloured edges, how can we tell if its edges have a red/blue colouring c such that G^c is an ASBG?

We call such a colouring c an *ASBG-colouring*, and we say that G is *ASBG-colourable*.

Some necessary (but not sufficient) conditions for a given graph G to have an ASBG-colouring c are:

- G must be bipartite and balanced;
- Each vertex of G must have odd degree. This is because each vertex of G^c must have blue degree one higher than red degree.

Leaves and Twigs

A *leaf* is a vertex of degree 1.

Leaves and Twigs

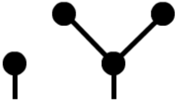
A *leaf* is a vertex of degree 1.

A *twig* is a configuration of three vertices; two leaves which are both attached to a third vertex of degree 3, called the base of the twig.

Leaves and Twigs

A *leaf* is a vertex of degree 1.

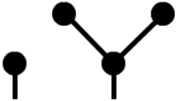
A *twig* is a configuration of three vertices; two leaves which are both attached to a third vertex of degree 3, called the base of the twig.



Leaves and Twigs

A *leaf* is a vertex of degree 1.

A *twig* is a configuration of three vertices; two leaves which are both attached to a third vertex of degree 3, called the base of the twig.

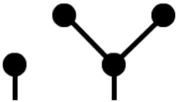


A *leaf-twig configuration* at a vertex v is a leaf and a twig both attached to the same vertex v.

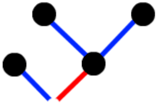
Leaves and Twigs

A *leaf* is a vertex of degree 1.

A *twig* is a configuration of three vertices; two leaves which are both attached to a third vertex of degree 3, called the base of the twig.

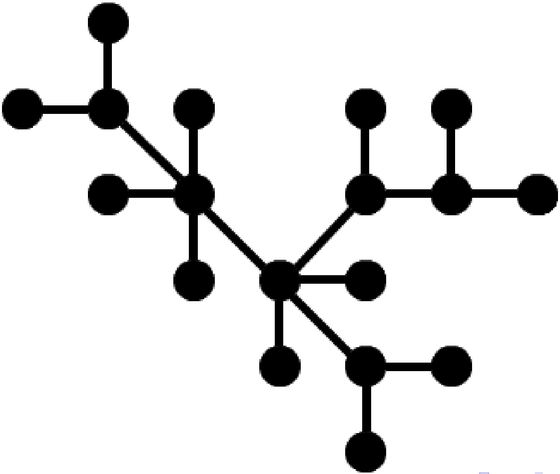


A *leaf-twig configuration* at a vertex v is a leaf and a twig both attached to the same vertex v.



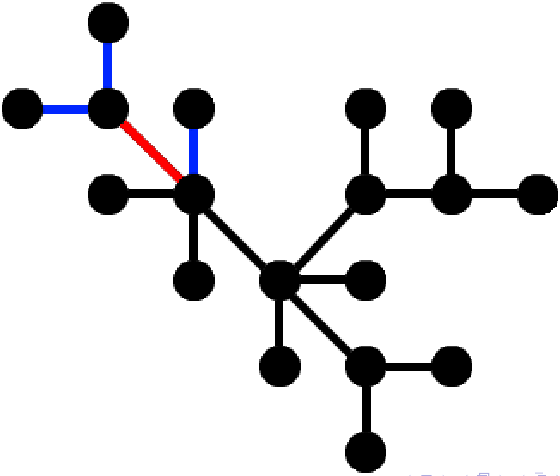
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



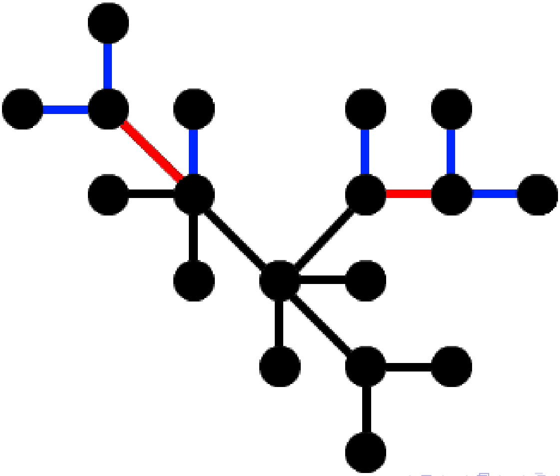
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



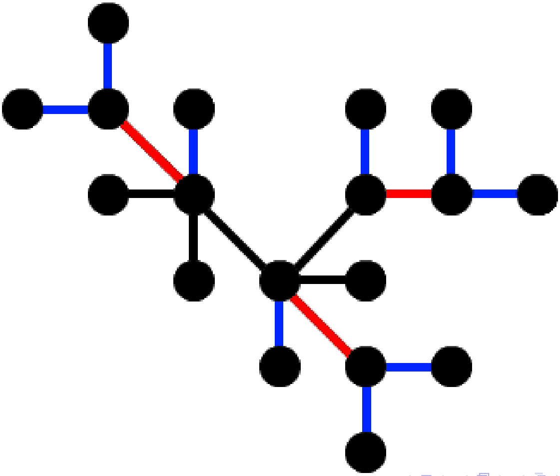
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



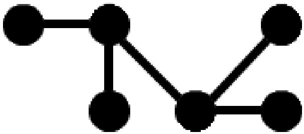
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



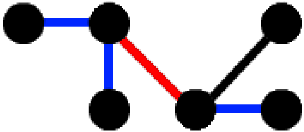
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



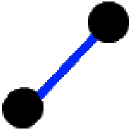
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



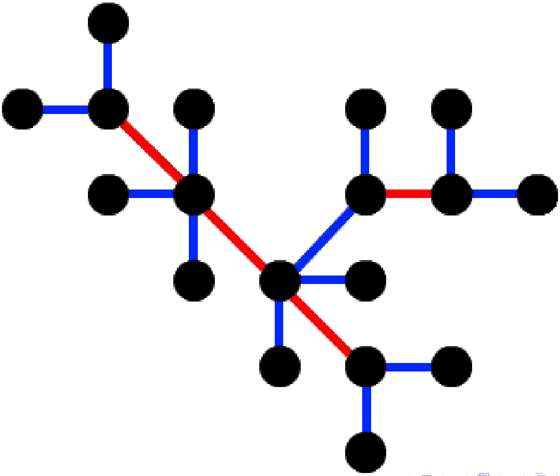
Trees

Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.

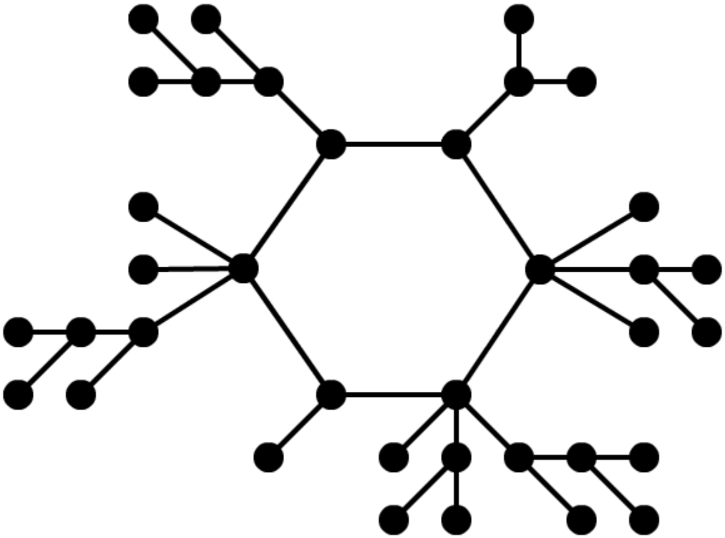


Trees

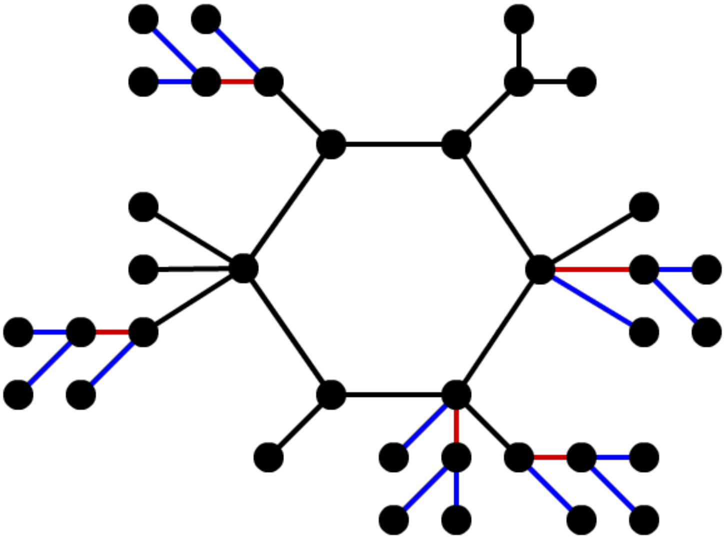
Theorem: A tree T is ASBG-colourable iff leaf-twig configurations can be removed until *the trivial ASBG remains*.



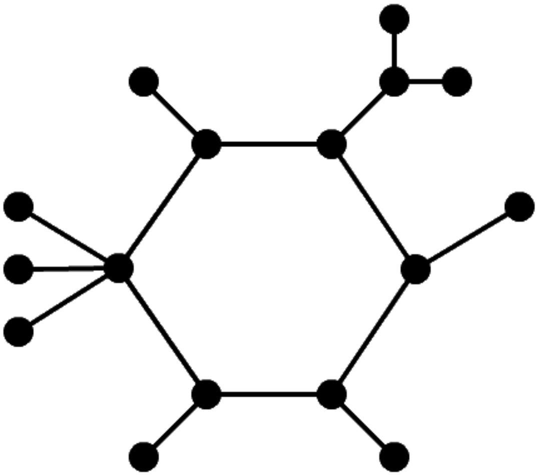
Unicyclic Graphs



Unicyclic Graphs



Unicyclic Graphs

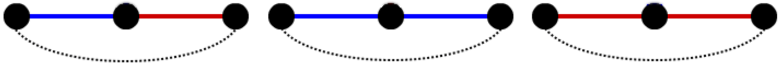


Leaf, Twig, and Triple-Type Vertices

There are three types of vertices that can appear in the cycle of a unicyclic graph: leaf-type, twig-type, and triple-type. This corresponds to the three different pairs of colours that can meet at a vertex in a cycle:

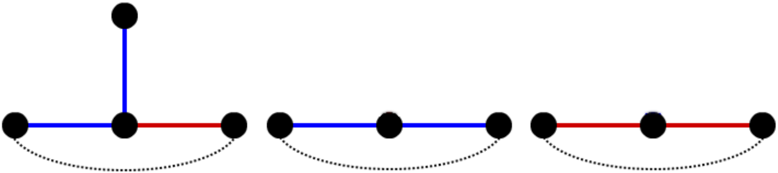
Leaf, Twig, and Triple-Type Vertices

There are three types of vertices that can appear in the cycle of a unicyclic graph: leaf-type, twig-type, and triple-type. This corresponds to the three different pairs of colours that can meet at a vertex in a cycle:



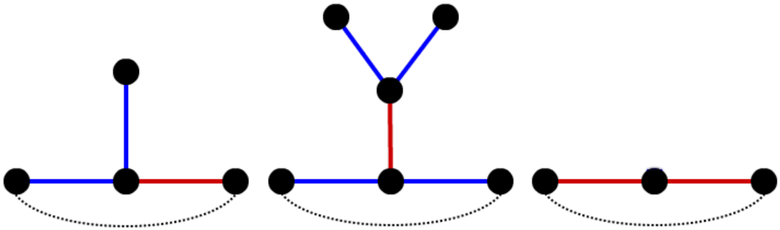
Leaf, Twig, and Triple-Type Vertices

There are three types of vertices that can appear in the cycle of a unicyclic graph: leaf-type, twig-type, and triple-type. This corresponds to the three different pairs of colours that can meet at a vertex in a cycle:



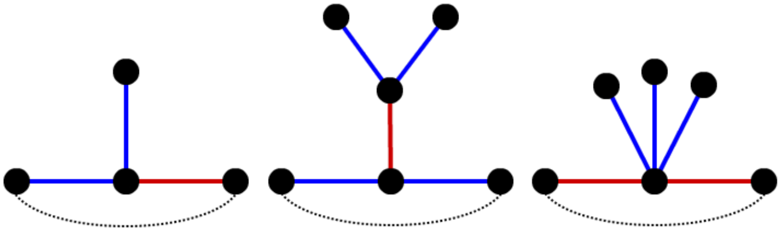
Leaf, Twig, and Triple-Type Vertices

There are three types of vertices that can appear in the cycle of a unicyclic graph: leaf-type, twig-type, and triple-type. This corresponds to the three different pairs of colours that can meet at a vertex in a cycle:

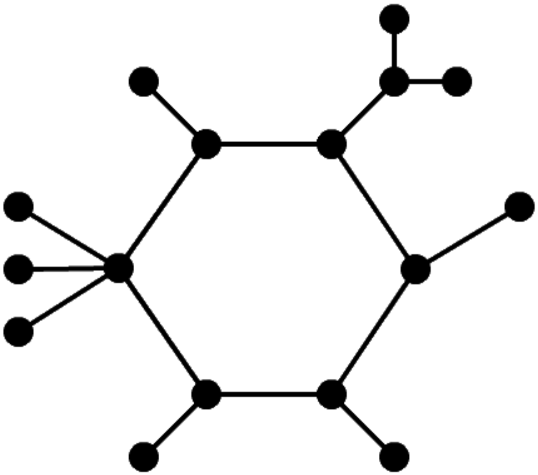


Leaf, Twig, and Triple-Type Vertices

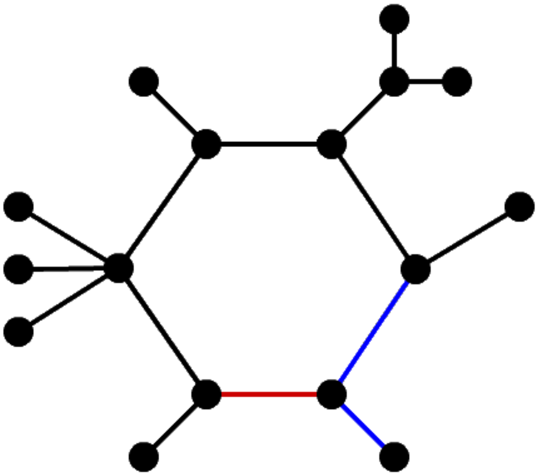
There are three types of vertices that can appear in the cycle of a unicyclic graph: leaf-type, twig-type, and triple-type. This corresponds to the three different pairs of colours that can meet at a vertex in a cycle:



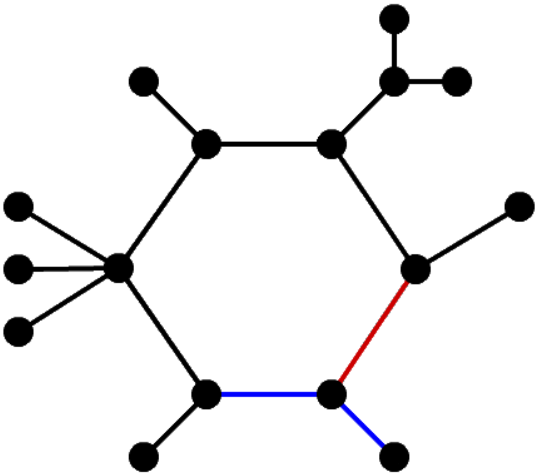
Unicyclic Graphs



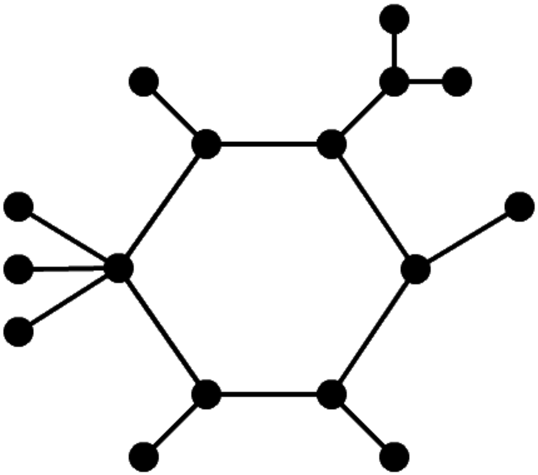
Unicyclic Graphs



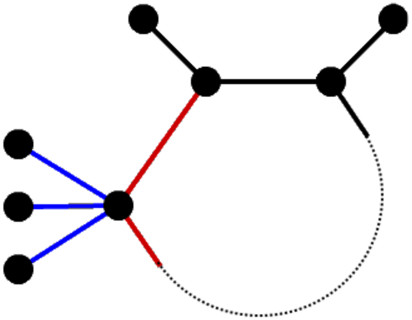
Unicyclic Graphs



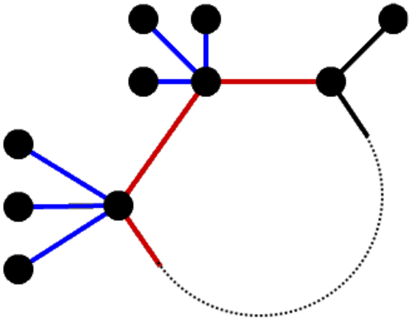
Unicyclic Graphs



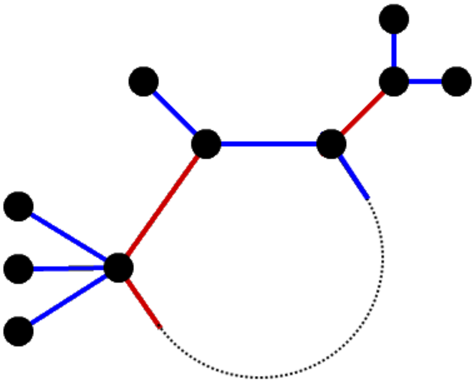
Unicyclic Graphs



Unicyclic Graphs



Unicyclic Graphs



Unicyclic Graphs

Theorem: A unicyclic graph G is ASBG-colourable if and only if:

- The cycle of G has even length;

Unicyclic Graphs

Theorem: A unicyclic graph G is ASBG-colourable if and only if:

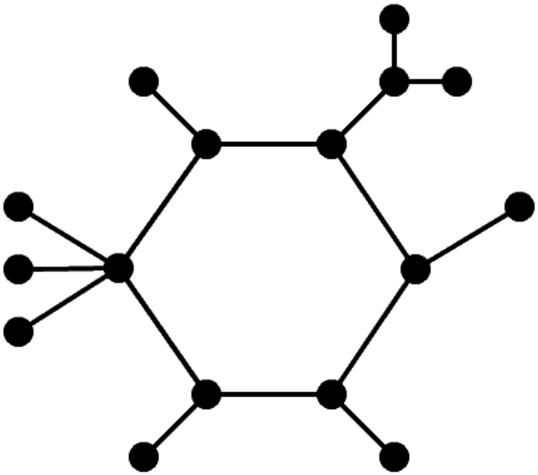
- The cycle of G has even length;
- Every vertex of the cycle of G is either leaf-type, twig-type, or triple-type;

Unicyclic Graphs

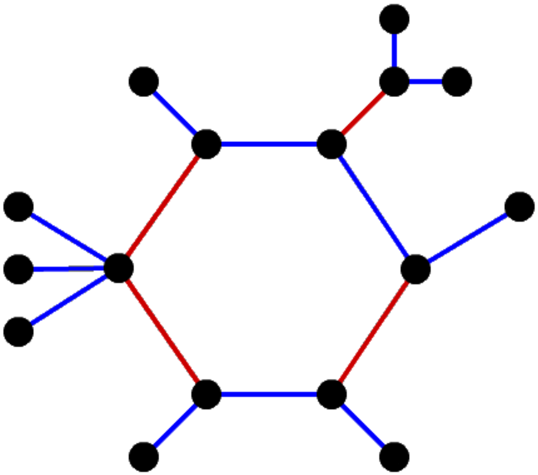
Theorem: A unicyclic graph G is ASBG-colourable if and only if:

- The cycle of G has even length;
- Every vertex of the cycle of G is either leaf-type, twig-type, or triple-type;
- Any twig/triple-type vertex must be an odd (even) distance away from the next twig/triple-type vertex if it is of the same (opposite) type.

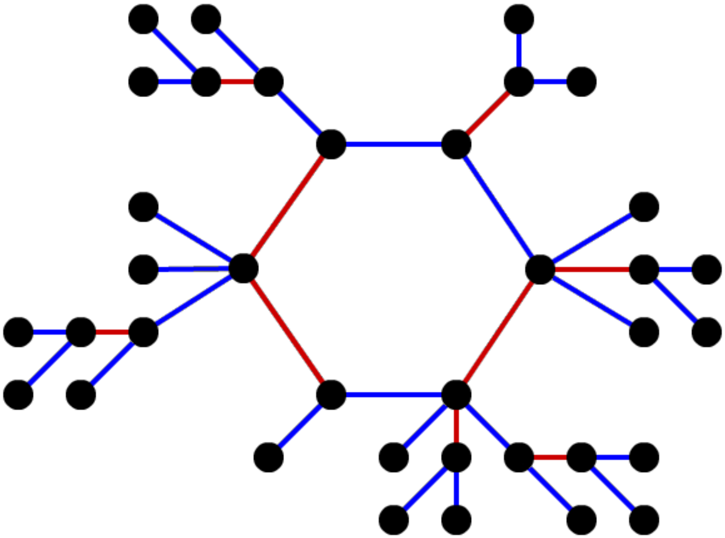
Unicyclic Graphs



Unicyclic Graphs

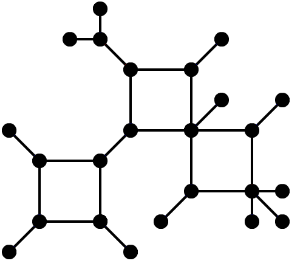


Unicyclic Graphs



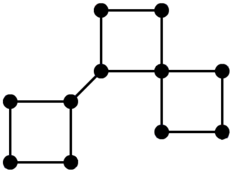
Cycle-Edge-Disjoint Graphs

A *cycle-edge-disjoint graph* is a graph in which no pair of cycles share an edge.



Cycle-Edge-Disjoint Graphs

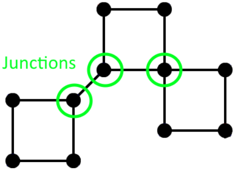
A *cycle-edge-disjoint graph* is a graph in which no pair of cycles share an edge.



We will define the *skeleton of a graph* $Sk(G)$ to be the graph that remains after all leaves have been repeatedly removed.

Cycle-Edge-Disjoint Graphs

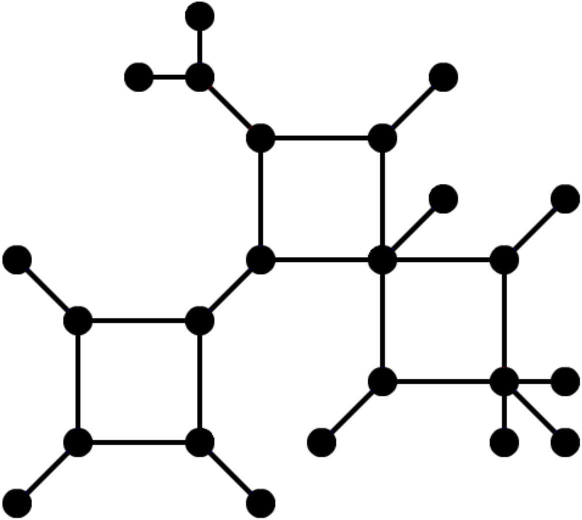
A *cycle-edge-disjoint graph* is a graph in which no pair of cycles share an edge.



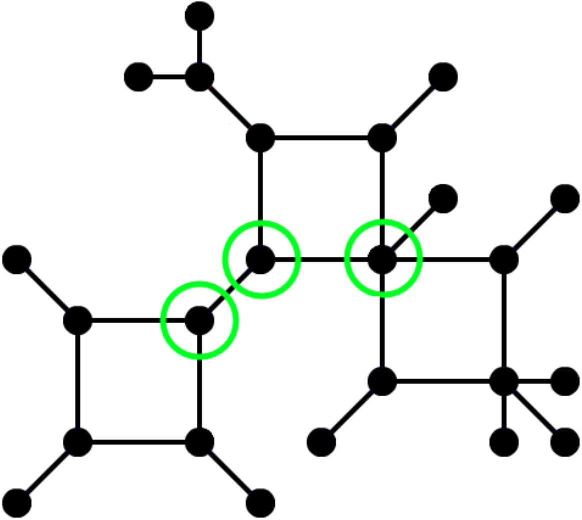
We will define the *skeleton of a graph* $Sk(G)$ to be the graph that remains after all leaves have been repeatedly removed.

As well as leaf, twig, and triple-type vertices, cycle-edge-disjoint graphs (and all graphs in general) have one other type of vertex: a junction.

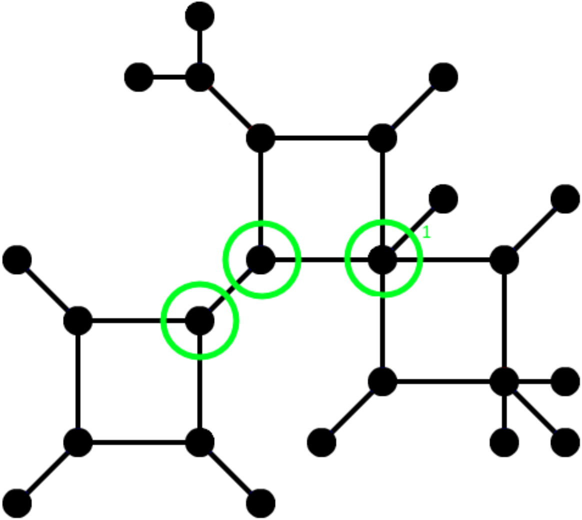
Junction Weight Algorithm



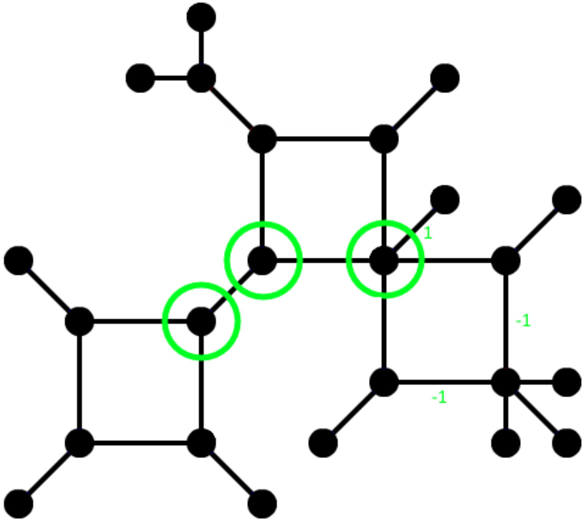
Junction Weight Algorithm



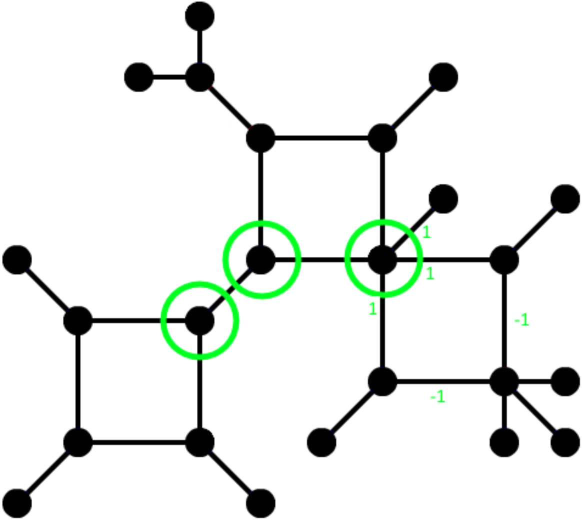
Junction Weight Algorithm



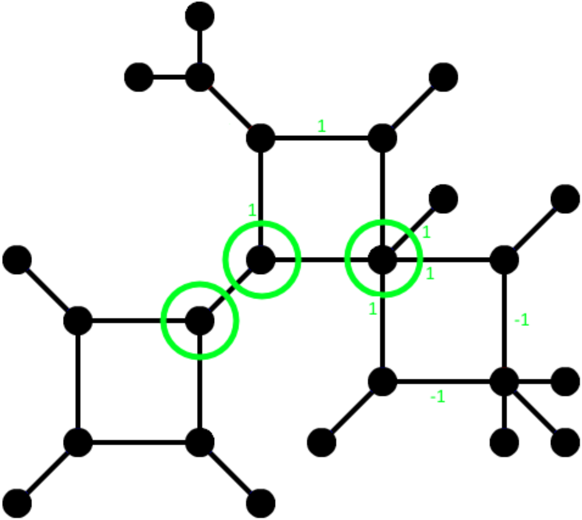
Junction Weight Algorithm



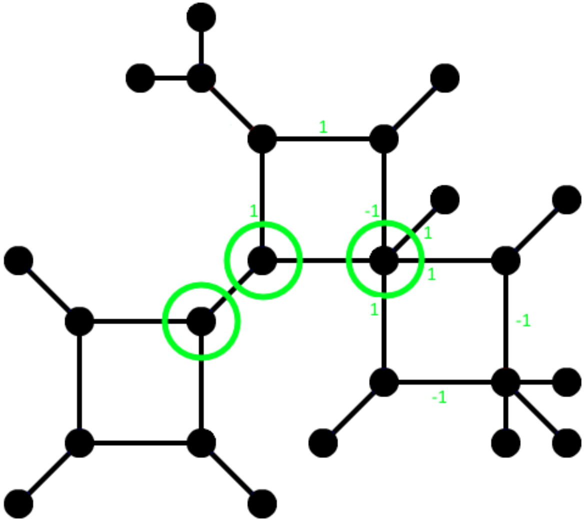
Junction Weight Algorithm



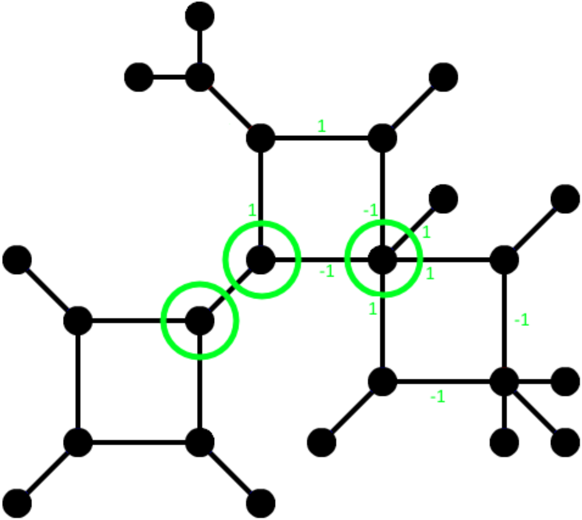
Junction Weight Algorithm



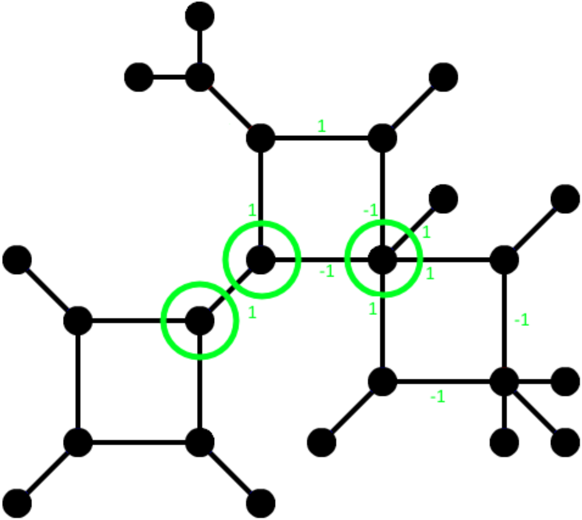
Junction Weight Algorithm



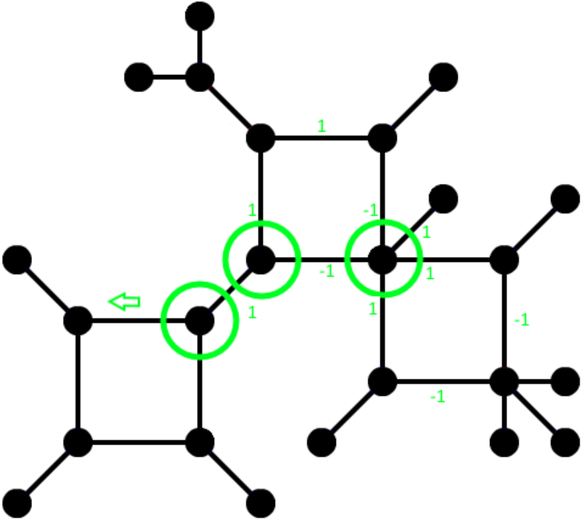
Junction Weight Algorithm



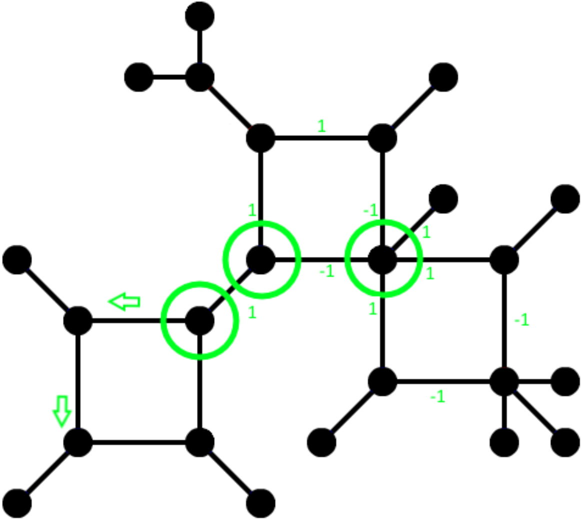
Junction Weight Algorithm



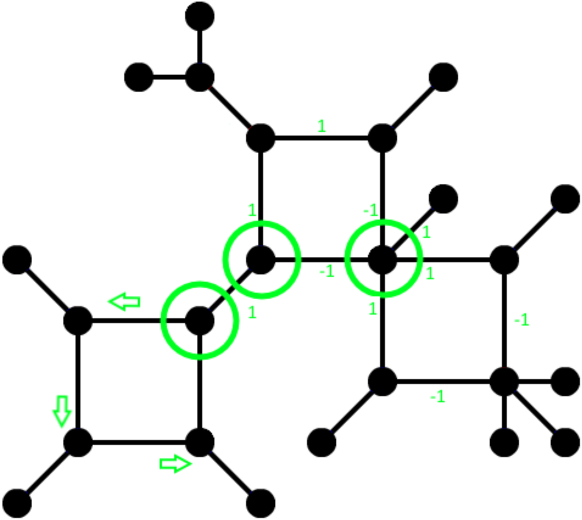
Junction Weight Algorithm



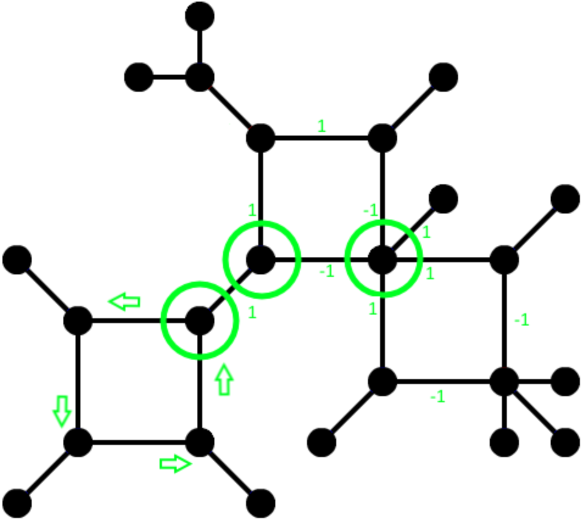
Junction Weight Algorithm



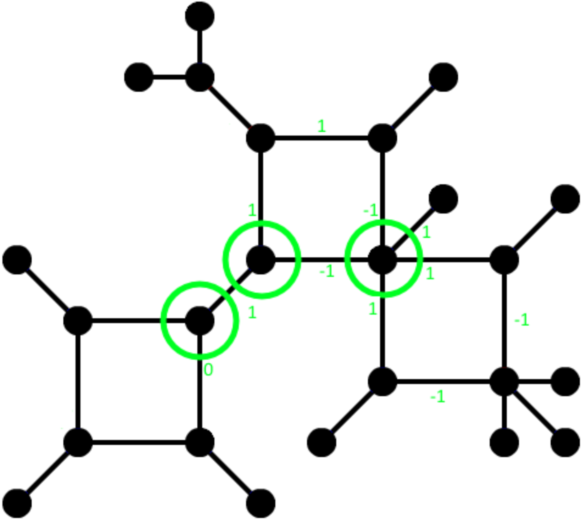
Junction Weight Algorithm



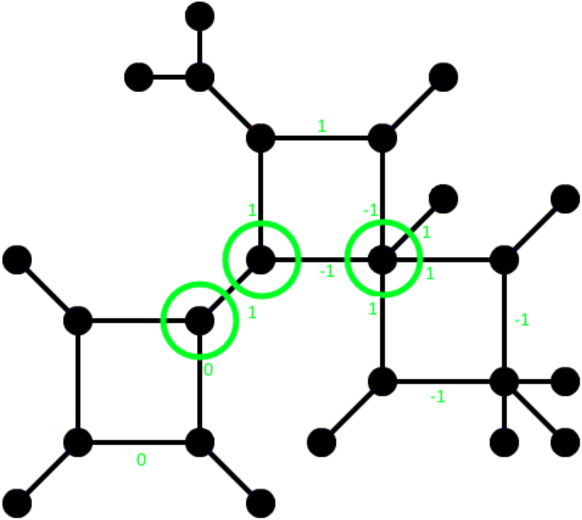
Junction Weight Algorithm



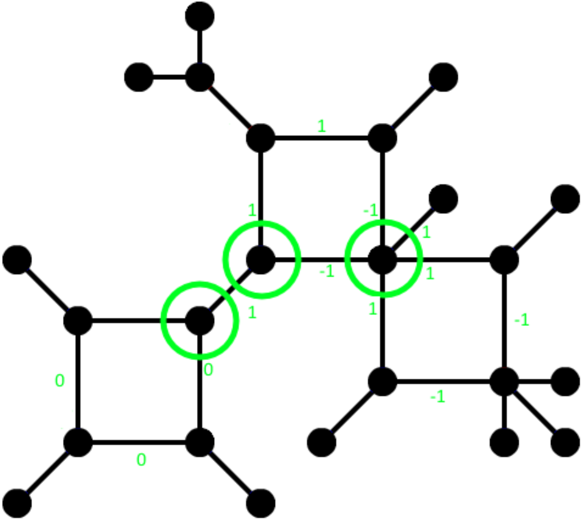
Junction Weight Algorithm



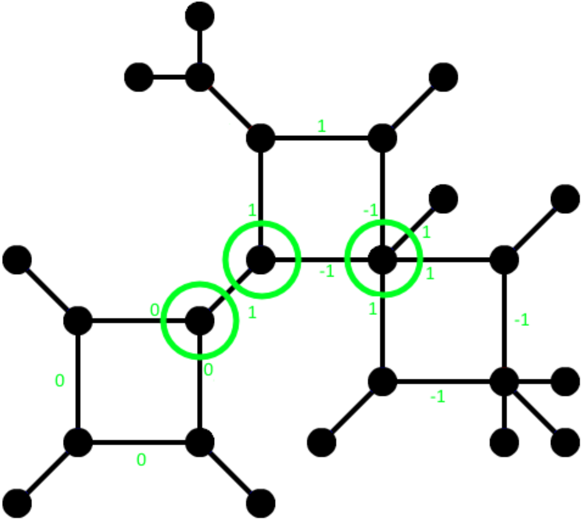
Junction Weight Algorithm



Junction Weight Algorithm



Junction Weight Algorithm



Cycle Edge Disjoint Graphs

Theorem: A cycle-edge-disjoint graph G is ASBG-colourable if and only if:

- All cycles of G are of even length;

Cycle Edge Disjoint Graphs

Theorem: A cycle-edge-disjoint graph G is ASBG-colourable if and only if:

- All cycles of G are of even length;
- Every vertex of the $Sk(G)$ is either leaf-type, twig-type, or triple-type**, or a junction of weight 1**;

Cycle Edge Disjoint Graphs

Theorem: A cycle-edge-disjoint graph G is ASBG-colourable if and only if:

- All cycles of G are of even length;
- Every vertex of the $Sk(G)$ is either leaf-type, twig-type, or triple-type**, or a junction of weight 1**;
- Any twig/triple-type vertex must be an odd (even) distance away from the next twig/triple-type vertex if it is of the same (opposite) type.

Cycle Edge Disjoint Graphs

Theorem: A cycle-edge-disjoint graph G is ASBG-colourable if and only if:

- All cycles of G are of even length;
- Every vertex of the $Sk(G)$ is either leaf-type, twig-type, or triple-type**, or a junction of weight 1**;
- Any twig/triple-type vertex must be an odd (even) distance away from the next twig/triple-type vertex if it is of the same (opposite) type. **(Also junctions only have surplus weights of ± 2 and are an odd (even) length away from the next junction if they require the same (opposite) surplus weight.)**

Richard A. Brualdi, Kathleen P. Kiernan, Seth A. Meyer, Michael W. Schroeder , *Patterns of Alternating Sign Matrices*, Department of Mathematics University of Wisconsin, 2011