Edge-connectivity augmentation of graphs over symmetric parity families

Zoltán Szigeti

Laboratoire G-SCOP
INP Grenoble, France

8 April 2009

Outline

1. Edge-connectivity

2. T-cuts

3. Symmetric parity families

Outline

1. Edge-connectivity
   1. Definitions
   2. Cut equivalent trees
   3. Edge-connectivity augmentation
2. T-cuts
3. Symmetric parity families

Outline

1. Edge-connectivity
   1. Definitions
   2. Cut equivalent trees
   3. Edge-connectivity augmentation
2. T-cuts
   1. Definitions
   2. Minimum T-cut
   3. Augmentation of minimum T-cut
3. Symmetric parity families

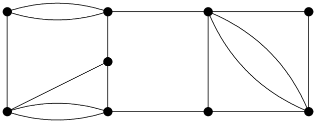
Outline

1. Edge-connectivity
   1. Definitions
   2. Cut equivalent trees
   3. Edge-connectivity augmentation
2. T-cuts
   1. Definitions
   2. Minimum T-cut
   3. Augmentation of minimum T-cut
3. Symmetric parity families
   1. Definition, Examples
   2. Minimum cut over a symmetric parity family
   3. Augmentation of minimum cut over a symmetric parity family

Definitions

Global edge-connectivity

Given a graph $G = (V, E)$ and an integer k, G is called *k-edge-connected* if each cut contains at least k edges.



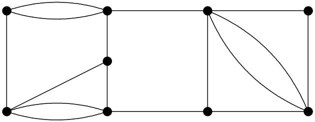
Definitions

Global edge-connectivity

Given a graph $G = (V, E)$ and an integer k, G is called k-edge-connected if each cut contains at least k edges.

Local edge-connectivity

Given a graph $G = (V, E)$ and $u, v \in V$, the local edge-connectivity $\lambda_G(u, v)$ is defined as the minimum cardinality of a cut separating u and v.



Cut equivalent tree

Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

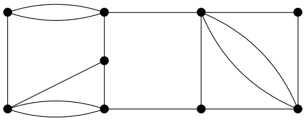
1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*

Cut equivalent tree

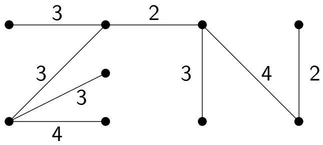
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



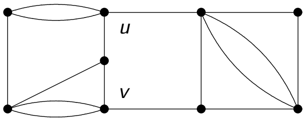
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

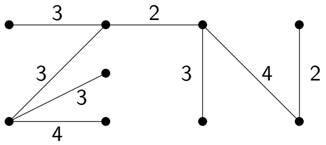
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



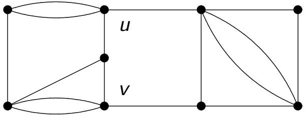
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

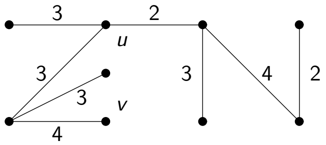
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



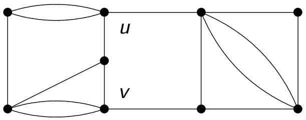
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

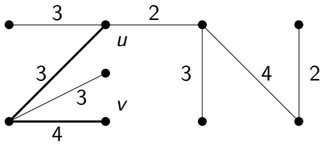
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



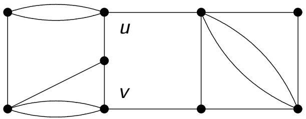
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

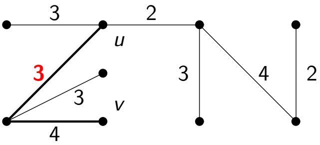
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



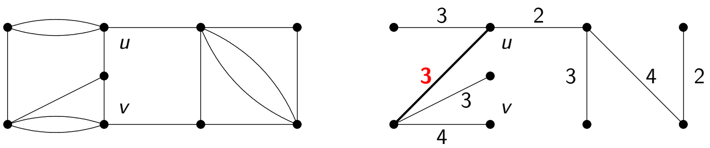
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$

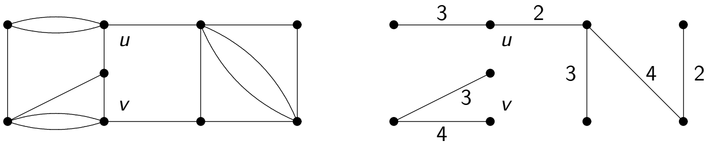
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



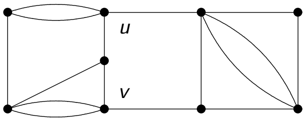
Graph $G = (V, E)$ Cut equivalent tree $H = (V, E')$

Cut equivalent tree

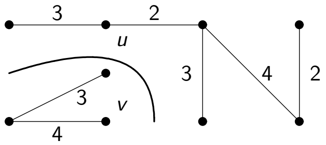
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



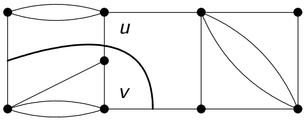
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

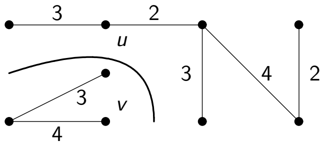
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



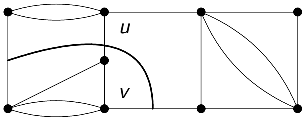
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

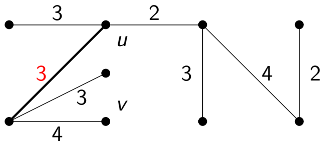
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



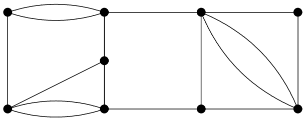
Cut equivalent tree $H = (V, E')$

Cut equivalent tree

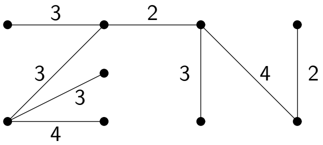
Theorem (Gomory-Hu)

For every graph $G = (V, E)$, we can find, in polynomial time, a tree $H = (V, E')$ and $c : E' \to \mathbb{Z}$ such that for all $u, v \in V$

1. *the local edge-connectivity $\lambda_G(u, v)$ is equal to the minimum value $c(e)$ of the edges e of the unique (u, v)-path in H,*
2. *if e achives this minimum, then a minimum cut of G separating u and v is given by the two connected components of $H - e$.*



Graph $G = (V, E)$



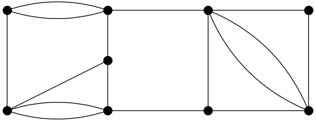
Cut equivalent tree $H = (V, E')$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



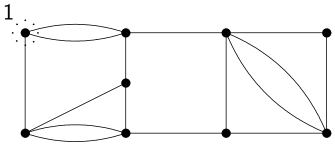
Graph $G, k = 4$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)

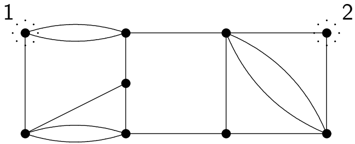


Graph $G, k = 4$

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)

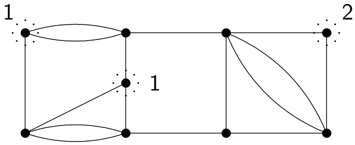


Graph $G, k = 4$

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



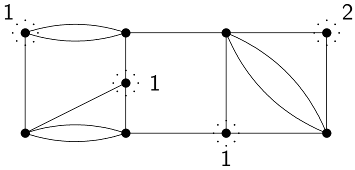
Graph $G, k = 4$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)

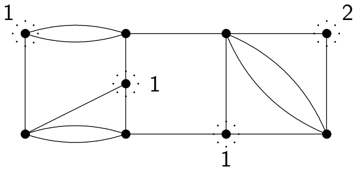


Graph $G, k = 4$

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



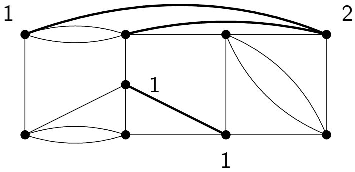
$$\text{Opt} \geq \lceil \tfrac{5}{2} \rceil = 3$$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



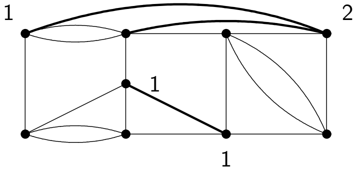
Graph $G + F$ is 4-edge-connected and $|F| = 3$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



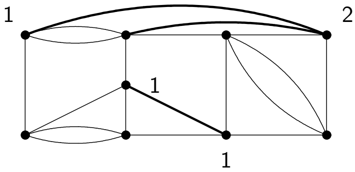
Opt$= \lceil \frac{1}{2}$maximum deficiency of a subpartition of $V\rceil$

Edge-Connectivity Augmentation

Global edge-connectivity augmentation of a graph

Given a graph $G = (V, E)$ and an integer $k \geq 2$, what is the minimum number of new edges whose addition results in a k-edge-connected graph ?

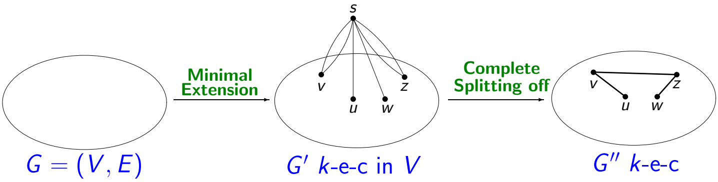
1. Minimax theorem (Watanabe, Nakamura)
2. Polynomially solvable (Cai, Sun)



General method

Frank's algorithm

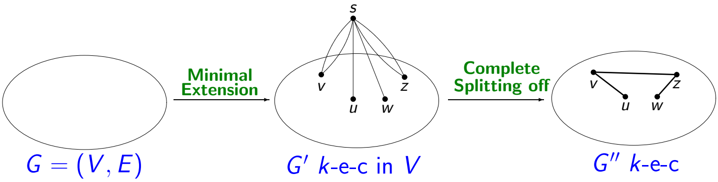
1. Minimal extension,
   - (i) Add a new vertex s,
   - (ii) Add a minimum number of new edges incident to s to satisfy the edge-connectivity requirements,
   - (iii) If the degree of s is odd, then add an arbitrary edge incident to s.
2. Complete splitting off.



General method

Frank's algorithm

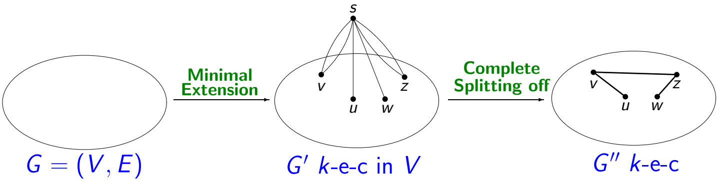
1. Minimal extension,
   (i) Add a new vertex s,
   (ii) Add a minimum number of new edges incident to s to satisfy the edge-connectivity requirements,
   (iii) If the degree of s is odd, then add an arbitrary edge incident to s.
2. Complete splitting off.



General method

Frank's algorithm

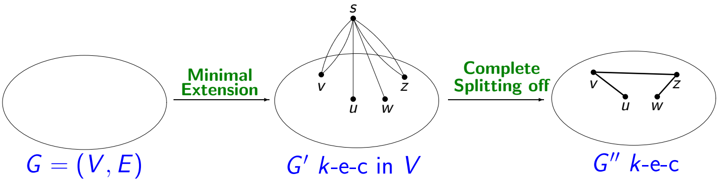
1. Minimal extension,
   (i) Add a new vertex s,
   (ii) Add a minimum number of new edges incident to s to satisfy the edge-connectivity requirements,
   (iii) If the degree of s is odd, then add an arbitrary edge incident to s.
2. Complete splitting off.



General method

Frank's algorithm

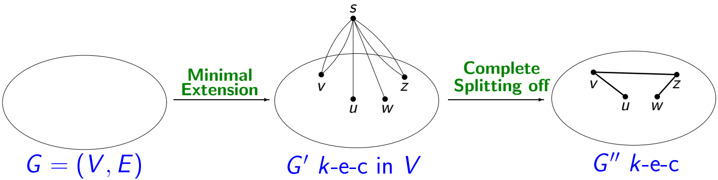
1. Minimal extension,
   (i) Add a new vertex s,
   (ii) Add a minimum number of new edges incident to s to satisfy the edge-connectivity requirements,
   (iii) If the degree of s is odd, then add an arbitrary edge incident to s.
2. Complete splitting off.



General method

Frank's algorithm

1. Minimal extension,
   (i) Add a new vertex s,
   (ii) Add a minimum number of new edges incident to s to satisfy the edge-connectivity requirements,
   (iii) If the degree of s is odd, then add an arbitrary edge incident to s.
2. Complete splitting off.



Minimal extension

Definition

A function p on 2^V is called skew-supermodular if at least one of following inequalities hold for all $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y),$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X).$$

Minimal extension

Definition

A function p on 2^V is called skew-supermodular if at least one of following inequalities hold for all $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y),$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X).$$

Theorem (Frank)

Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric skew-supermodular function.

1. *The minimum number of edges in an extension ($d(X) \geq p(X)$ for all $X \subseteq V$) is equal to the maximum p-value of a subpartition of V.*
2. *An optimal extension can be found in polynomial time in the special cases mentioned in this talk.*

Minimal extension

Definition

A function p on 2^V is called skew-supermodular if at least one of following inequalities hold for all $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y),$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X).$$

Theorem (Frank)

Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric skew-supermodular function.

1. *The minimum number of edges in an extension ($d(X) \geq p(X)$ for all $X \subseteq V$) is equal to the maximum p-value of a subpartition of V.*
2. *An optimal extension can be found in polynomial time in the special cases mentioned in this talk.*

Minimal extension

Definition

A function p on 2^V is called skew-supermodular if at least one of following inequalities hold for all $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y),$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X).$$

Theorem (Frank)

Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric skew-supermodular function.

1. *The minimum number of edges in an extension ($d(X) \geq p(X)$ for all $X \subseteq V$) is equal to the maximum p-value of a subpartition of V.*
2. *An optimal extension can be found in polynomial time in the special cases mentioned in this talk.*

Minimal extension

Definition

A function p on 2^V is called skew-supermodular if at least one of following inequalities hold for all $X, Y \subseteq V$:

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y),$$
$$p(X) + p(Y) \leq p(X - Y) + p(Y - X).$$

Theorem (Frank)

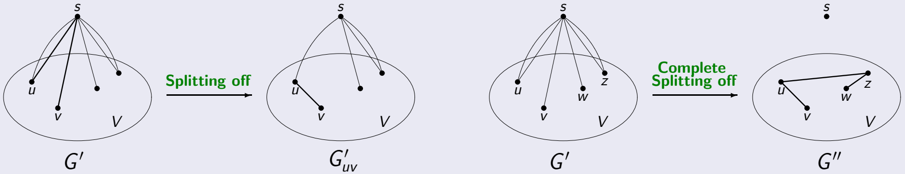
Let $p : 2^V \to \mathbb{Z} \cup \{-\infty\}$ be a symmetric skew-supermodular function.

1. *The minimum number of edges in an extension ($d(X) \geq p(X)$ for all $X \subseteq V$) is equal to the maximum p-value of a subpartition of V.*
2. *An optimal extension can be found in polynomial time in the special cases mentioned in this talk.*

For global edge-connectivity augmentation $p(X) := k - d_G(X)$.

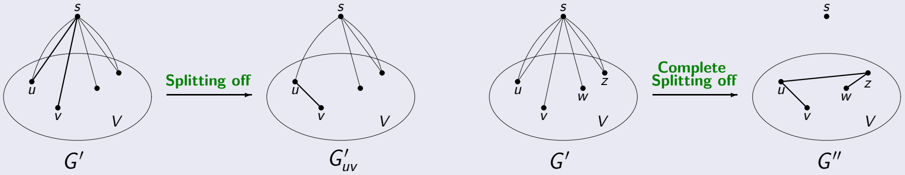
Complete splitting off

Definitions



Complete splitting off

Definitions



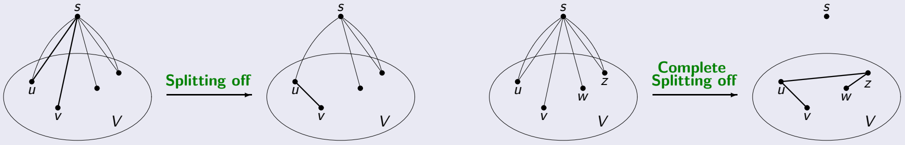
Theorem (Mader)

Let $G' = (V + s, E)$ be a graph so that $d(s)$ is even and no cut edge is incident to s.

1. *Then there exists a complete splitting off at s that preserves the local edge-connectivity between all pairs of vertices in V.*
2. *Such a complete splitting off can be found in polynomial time.*

Complete splitting off

Definitions



G' G'_{uv} G' G''

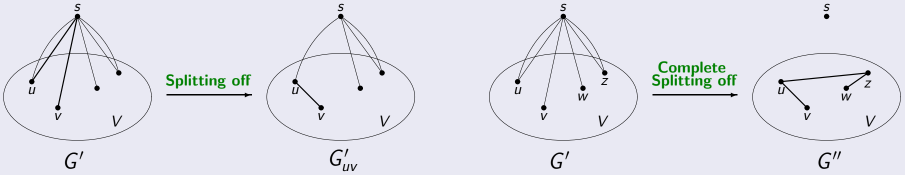
Theorem (Mader)

Let $G' = (V + s, E)$ be a graph so that $d(s)$ is even and no cut edge is incident to s.

1. *Then there exists a complete splitting off at s that preserves the local edge-connectivity between all pairs of vertices in V.*
2. *Such a complete splitting off can be found in polynomial time.*

Complete splitting off

Definitions



Theorem (Mader)

Let $G' = (V + s, E)$ be a graph so that $d(s)$ is even and no cut edge is incident to s.

1. *Then there exists a complete splitting off at s that preserves the local edge-connectivity between all pairs of vertices in V.*
2. *Such a complete splitting off can be found in polynomial time.*

Negative Result

Negative Result

Definition

A graph H covers a function $p : 2^{V(H)} \to \mathbb{Z} \cup \{-\infty\}$ if each cut $\delta_H(X)$ contains at least $p(X)$ edges.

Negative Result

Definition

A graph H covers a function $p : 2^{V(H)} \to \mathbb{Z} \cup \{-\infty\}$ if each cut $\delta_H(X)$ contains at least $p(X)$ edges.

Minimum Cover of a Symmetric Skew-Supermodular Function by a Graph

Instance : $p : 2^V \to \mathbb{Z}$ symmetric skew-supermodular, $\gamma \in \mathbb{Z}^+$.
Question : Does there exist a graph on V with at most γ edges that covers p?

Negative Result

Definition

A graph H covers a function $p : 2^{V(H)} \rightarrow \mathbb{Z} \cup \{-\infty\}$ if each cut $\delta_H(X)$ contains at least $p(X)$ edges.

Minimum Cover of a Symmetric Skew-Supermodular Function by a Graph

Instance : $p : 2^V \rightarrow \mathbb{Z}$ symmetric skew-supermodular, $\gamma \in \mathbb{Z}^+$.
Question : Does there exist a graph on V with at most γ edges that covers p?

Theorem (Z. Király, Z. Nutov)

The above problem is NP-complete.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.

   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.

   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.

   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.

   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.
   Examples :
   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.
   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.

   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.
   (b) $T = V$: a perfect matching is a T-join.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.
   (b) $T = V$: a perfect matching is a T-join.

Properties

1. If X, Y are T-odd, then either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. A T-join and a T-cut always have an edge in common.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.
   (b) $T = V$: a perfect matching is a T-join.

Properties

1. If X, Y are T-odd, then either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. A T-join and a T-cut always have an edge in common.

T-cut, T-join

Definitions

Given a connected graph $G = (V, E)$ and $T \subseteq V$ with $|T|$ even.

1. A subset X of V is called T-odd if $|X \cap T|$ is odd.
2. A cut $\delta(X)$ is called T-cut if X is T-odd.
3. A subset F of E is called T-join if $T = \{v \in V : d_F(v) \text{ is odd}\}$.

   Examples :

   (a) $T = \{u, v\}$: a (u, v)-path is a T-join.
   (b) $T = V$: a perfect matching is a T-join.

Properties

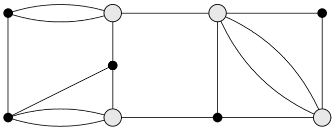
1. If X, Y are T-odd, then either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. A T-join and a T-cut always have an edge in common.

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



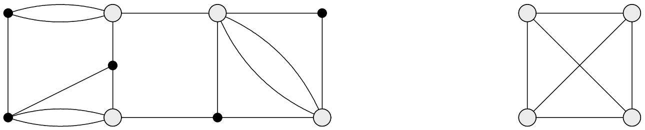
Graph G and vertex set T

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



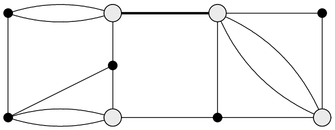
Graph G and vertex set T

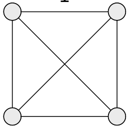
How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*





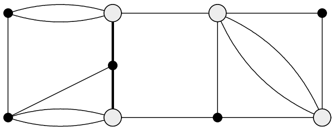
Graph G and vertex set T

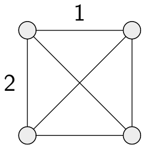
How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*





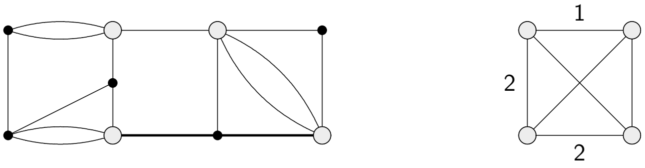
Graph G and vertex set T

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



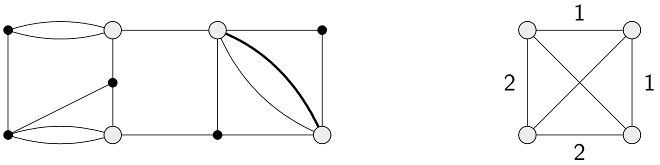
Graph G and vertex set T

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



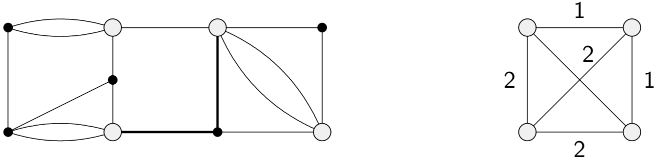
Graph G and vertex set T

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



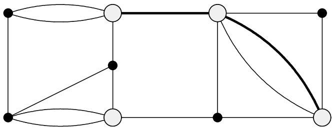
Graph G and vertex set T

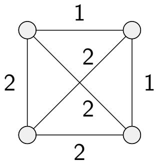
How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*





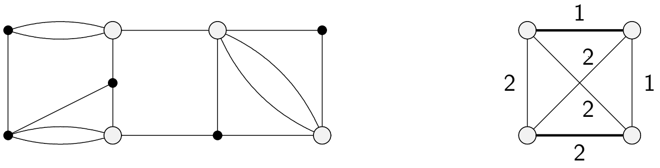
Graph G and vertex set T

How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*



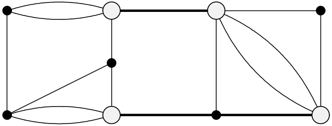
Graph G and vertex set T

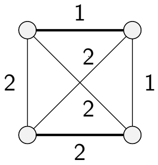
How to find a minimum T-join ?

Theorem (Edmonds-Johnson)

A minimum T-join can be found in polynomial time using

1. *shortest paths algorithm (Dijkstra) and*
2. *minimum weight perfect matching algorithm (Edmonds).*





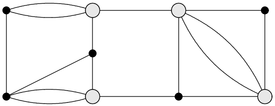
Graph G and minimum T-join

How to find a minimum T-cut ?

Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



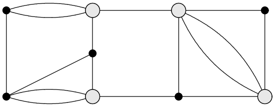
Graph G and vertex set T

How to find a minimum T-cut ?

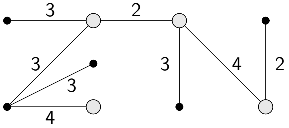
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



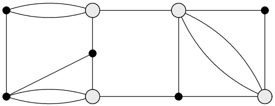
Cut equivalent tree H

How to find a minimum T-cut ?

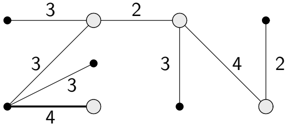
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



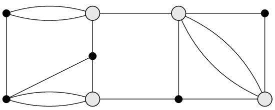
Cut equivalent tree H

How to find a minimum T-cut ?

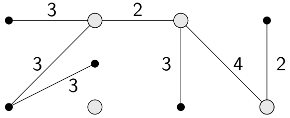
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



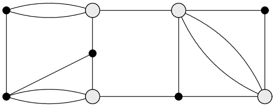
Cut equivalent tree H

How to find a minimum T-cut ?

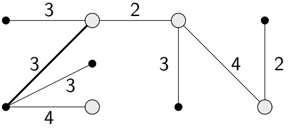
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



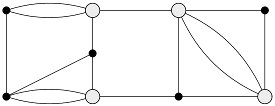
Cut equivalent tree H

How to find a minimum T-cut ?

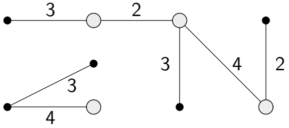
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



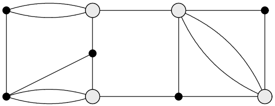
Cut equivalent tree H

How to find a minimum T-cut ?

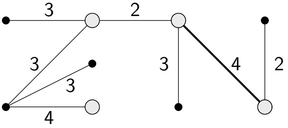
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



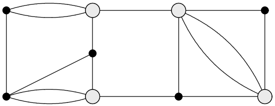
Cut equivalent tree H

How to find a minimum T-cut ?

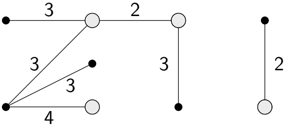
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



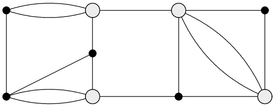
Cut equivalent tree H

How to find a minimum T-cut ?

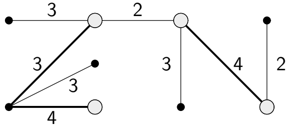
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



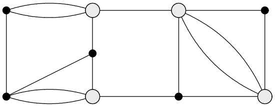
Cut equivalent tree H and edge set $J(H)$

How to find a minimum T-cut ?

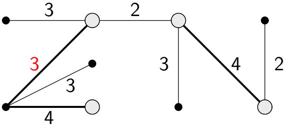
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



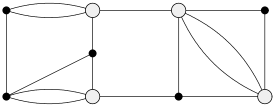
Cut equivalent tree H

How to find a minimum T-cut ?

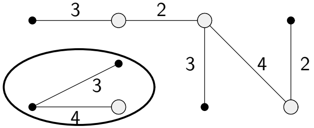
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Graph G and vertex set T



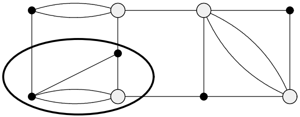
Cut equivalent tree H

How to find a minimum T-cut ?

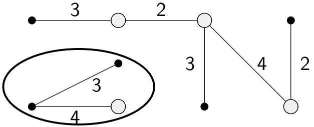
Theorem (Padberg-Rao)

A minimum T-cut can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are T-odd,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*



Minimum T-cut in G



Cut equivalent tree H

Proof

Lemma

For any T-cut $\delta(X)$ there exist $x \in X, y \notin X$ such that $\lambda_G(x, y) \geq c(e^*)$.

Proof

Lemma

For any T-cut $\delta(X)$ there exist $x \in X, y \notin X$ such that $\lambda_G(x, y) \geq c(e^*)$.

Proof : $J(H)$ is a T-join so there exists $xy \in J(H) \cap \delta_H(X)$ and $\lambda_G(x, y) = c(xy) \geq c(e^*)$.

Proof

Lemma

For any T-cut $\delta(X)$ there exist $x \in X, y \notin X$ such that $\lambda_G(x,y) \geq c(e^*)$.

Proof : $J(H)$ is a T-join so there exists $xy \in J(H) \cap \delta_H(X)$ and $\lambda_G(x,y) = c(xy) \geq c(e^*)$.

Correctness of Padberg-Rao's algorithm

Let $\delta(X)$ be a minimum T-cut and $\delta(Y)$ the T-cut defined by e^*.
By the lemma, there exist $x \in X, y \notin X$ such that

$$c(e^*) = d(Y) \geq d(X) \geq \lambda_G(x,y) \geq c(e^*).$$

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   (i) $k - d_G(X)$ satisfies both inequalities,
   (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   (i) $k - d_G(X)$ satisfies both inequalities,
   (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

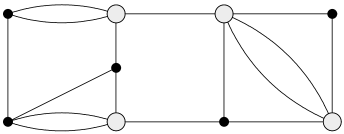
1. works because $p(X) := k - d_G(X)$ if X is T-odd and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) X, Y are T-odd $\Longrightarrow$ either $X \cap Y, X \cup Y$ or $X - Y, Y - X$ are T-odd.
2. works because for all T-odd sets, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*



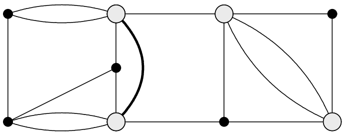
Graph G, vertex set T and $k = 4$

How to augment a minimum T-cut ?

Theorem (Z.Sz.)

Given a connected graph $G = (V, E)$, $T \subseteq V$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each T-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*



Minimum T-cut in $G + F$ is 4

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Examples

The most important examples are :

1. $\mathcal{F} := 2^V - \{\emptyset, V\}$
2. $\mathcal{F} := \{X \subset V : X \text{ is } T\text{-odd}\}$ where $T \subseteq V$ with $|T|$ even.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Examples

The most important examples are :

1. $\mathcal{F} := 2^V - \{\emptyset, V\}$
2. $\mathcal{F} := \{X \subset V : X \text{ is } T\text{-odd}\}$ where $T \subseteq V$ with $|T|$ even.

Definition : symmetric parity family

Definition

A family $\mathcal{F}$ of subsets of V is called symmetric parity family if

1. $\emptyset, V \notin \mathcal{F}$,
2. if $A \in \mathcal{F}$, then $V - A \in \mathcal{F}$,
3. if $A, B \notin \mathcal{F}$ and $A \cap B = \emptyset$, then $A \cup B \notin \mathcal{F}$.

Examples

The most important examples are :

1. $\mathcal{F} := 2^V - \{\emptyset, V\}$
2. $\mathcal{F} := \{X \subset V : X \text{ is } T\text{-odd}\}$ where $T \subseteq V$ with $|T|$ even.

How to find a minimum $\mathcal{F}$-cut ?

Theorem (Goemans-Ramakrishnan)

Given a connected graph G and a symmetric parity family $\mathcal{F}$, a minimum $\mathcal{F}$-cut, that is a minimum cut over $\mathcal{F}$, can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are in $\mathcal{F}$,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*

How to find a minimum $\mathcal{F}$-cut ?

Theorem (Goemans-Ramakrishnan)

Given a connected graph G and a symmetric parity family $\mathcal{F}$, a minimum $\mathcal{F}$-cut, that is a minimum cut over $\mathcal{F}$, can be found in polynomial time

1. *using a cut equivalent tree H and*
2. *taking the set $J(H)$ edges e of H for which the two connected components of $H - e$ are in $\mathcal{F}$,*
3. *taking the minimum value $c(e^*)$ of an edge of $J(H)$,*
4. *taking the cut defined by the two connected components of $H - e^*$.*

Proof

Lemma

For any $X \in \mathcal{F}$ there exist $x \in X, y \notin X$ such that $\lambda_G(x, y) \geq c(e^*)$.

Proof

Lemma

For any $X \in \mathcal{F}$ there exist $x \in X, y \notin X$ such that $\lambda_G(x, y) \geq c(e^*)$.

Proof : Exercise : there exists an edge $xy \in \delta_{J(H)}(X)$.

Proof

Lemma

For any $X \in \mathcal{F}$ there exist $x \in X, y \notin X$ such that $\lambda_G(x,y) \geq c(e^*)$.

Proof : Exercise : there exists an edge $xy \in \delta_{J(H)}(X)$.

Correctness of Goemans-Ramakrishnan's algorithm

The same proof works as for Padberg-Rao's algorithm.

How to augment a minimum $\mathcal{F}$-cut ?

Theorem (Z.Sz.)

Given a connected graph G, a symmetric parity family $\mathcal{F}$ and $k \in \mathbb{Z}$, the minimum number of edges whose addition results in a graph so that each $\mathcal{F}$-cut is of size at least k is equal to $\lceil \frac{1}{2}$ maximum p-value of a subpartition of $V \rceil$. An optimal augmentation can be found in polynomial time using

1. *Frank's minimal extension and*
2. *Mader's complete splitting off.*

Proof

1. works because $p(X) := k - d_G(X)$ if $X \in \mathcal{F}$ and $-\infty$ otherwise is symmetric skew-supermodular
   - (i) $k - d_G(X)$ satisfies both inequalities,
   - (ii) If $X, Y \in \mathcal{F}$, then either $X \cap Y, X \cup Y \in \mathcal{F}$ or $X - Y, Y - X \in \mathcal{F}$.
2. works because for all $X \in \mathcal{F}$, $d_{G'}(X) \geq k$ and, by the above lemma, $k \leq \lambda_{G'}(x, y) = \lambda_{G''}(x, y) \leq d_{G''}(X)$.

Conclusion

1. Special cases :
    1. Global edge-connectivity augmentation (Watanabe, Nakamura)
    2. Minimum T-cut augmentation
2. A new polynomial special case of the NP-complete problem MINIMUM COVER OF A SYMMETRIC SKEW-SUPERMODULAR FUNCTION BY A GRAPH

Conclusion

1. Special cases :
   1. Global edge-connectivity augmentation (Watanabe, Nakamura)
   2. Minimum T-cut augmentation
2. A new polynomial special case of the NP-complete problem MINIMUM COVER OF A SYMMETRIC SKEW-SUPERMODULAR FUNCTION BY A GRAPH

Conclusion

1. Special cases :
   1. Global edge-connectivity augmentation (Watanabe, Nakamura)
   2. Minimum T-cut augmentation
2. A new polynomial special case of the NP-complete problem MINIMUM COVER OF A SYMMETRIC SKEW-SUPERMODULAR FUNCTION BY A GRAPH

Conclusion

1. Special cases :
    1. Global edge-connectivity augmentation (Watanabe, Nakamura)
    2. Minimum T-cut augmentation
2. A new polynomial special case of the NP-complete problem MINIMUM COVER OF A SYMMETRIC SKEW-SUPERMODULAR FUNCTION BY A GRAPH