On Generation of Cut Conjunctions, Minimal k-Connected Spanning Subgraphs, Minimal Connected and Spanning Subsets and Vertices

Konrad Borys

kborys@eden.rutgers.edu

Rutgers University
Outline

- Generating $k$-vertex connected spanning subgraphs
  - Incremental polynomial time algorithm
  - Application: network reliability
- Vertex generation
  - Two representations of polyhedra
  - Proof of NP-hardness
- Monotone generation problem
Generating $k$-Vertex Connected Spanning Subgraphs
$k$-Vertex Connected Spanning Subgraphs

**Input:** $k$-vertex connected graph $G$

\[ G = \begin{array}{c}
\text{\includegraphics{example.png}}
\end{array} \quad k = 1 \]
$k$-Vertex Connected Spanning Subgraphs

**Input:** $k$-vertex connected graph $G$

$$G = \begin{array}{c}
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\bullet \\
\end{array} \quad k = 1$$

**Output:** list of all minimal $k$-vertex connected spanning subgraphs of $G$
$k$-Vertex Connected Spanning Subgraphs

**Input:** $k$-vertex connected graph $G$

\[
G = \begin{array}{c}
\bullet \\
\bullet \\
\bullet
\end{array} \quad k = 1
\]

**Output:** list of all minimal $k$-vertex connected spanning subgraphs of $G$

$\Omega(2^{|V|})$ spanning trees
Complexity of Generation Problems

\[ n = \text{input size}, \quad N = \text{output size} \]

**Polynomial Total Time**

\[ \text{start} \rightarrow \text{poly}(n,N) \rightarrow \text{halt} \]

\[ \text{time} \]
Complexity of Generation Problems

\[ n = \text{input size}, \ N = \text{output size} \]

**Polynomial Total Time**

\[
\text{start} \quad \text{poly}(n,N) \quad \text{halt}
\]

**Incremental Polynomial Time**

\[
\text{start} \quad \text{poly}(n,K) \quad \text{halt}
\]

\[ \text{K elements} \]
Complexity of Generation Problems

\[ n = \text{input size}, \quad N = \text{output size} \]

Polynomial Total Time

Incremental Polynomial Time

Polynomial Delay
Previous Results

Polynomial delay algorithms for generating spanning trees ($k = 1$)

\( n = \# \text{ of vertices}, \ m = \# \text{ of edges}, \ N = \# \text{ of trees} \)

- Read, Tarjan 1975 \( O(Nm + n + m) \) time, \( O(n + m) \) space
- Gabow, Myers 1978 \( O(Nm + n + m) \) time, \( O(n + m) \) space (directed and undirected graphs)
- Kapoor, Ramesh 1992 \( O(N + n + m) \) time, \( O(nm) \) space
- Matsui 1993 \( O(Nn + n + m) \) time, \( O(n + m) \) space
- Shioura, Tamura 1995 \( O(N + n + m) \) time, \( O(nm) \) space
- Shioura, Tamura, Uno 1994 \( O(N + n + m) \) time, \( O(nm) \) space
New Result

We can generate all minimal $k$-vertex connected spanning subgraphs in incremental polynomial time (for any given $k$).
Supergraph Approach

Supergraph = directed graph
Vertices = objects to be generated
Supergraph Approach

Supergraph = directed graph
Vertices = objects to be generated

Supergraph strongly connected $\rightarrow$ breadth-first-search outputs all objects
Supergraph Approach

Supergraph = directed graph
Vertices = objects to be generated

Supergraph strongly connected $\Rightarrow$ breadth-first-search outputs all objects

Generating neighbors in incremental polynomial time $\Rightarrow$ breadth-first-search in incremental polynomial time
Supergraph Approach

Our tasks:

Define neighborhoods that make the supergraph strongly connected

Show that neighbors can be generated in incremental polynomial time
Generating Neighbors

minimal 4-connected

\((V,X)\)
Generating Neighbors

minimal 4-connected    3-connected

(V,X) \rightarrow (V,X-e_1)
Generating Neighbors

minimal 4-connected  3-connected  4-connected

(V,X)

(V,X-e₁)

(V,X-e₁+Y₁₁)
Generating Neighbors

minimal 4-connected 3-connected 4-connected minimal 4-connected

(V,X) → (V,X-e₁) → (V,X-e₁+Y₁₁) → (V,X₁)
Generating Neighbors

minimal 4-connected  3-connected  4-connected  minimal 4-connected

(V,X)

(V,X-e₁ + Y₁₁)

(V,X-e₁ + Y₁₂)

(V,X-e₁ + Y₁₃)

(V,X₁)

(V,X₂)

(V,X₃)
Generating Neighbors

\[(V, X) \rightarrow (V, X-e_1) \rightarrow (V, X-e_1+Y_{11}) \rightarrow (V, X_1)\]

\[(V, X) \rightarrow (V, X-e_2) \rightarrow (V, X-e_2+Y_{21}) \rightarrow (V, X_2)\]

\[(V, X) \rightarrow (V, X-e_2) \rightarrow (V, X-e_2+Y_{12}) \rightarrow (V, X_3)\]

\[(V, X) \rightarrow (V, X-e_2) \rightarrow (V, X-e_2+Y_{13}) \rightarrow (V, X_4)\]
Generating Neighbors

minimal 4-connected 3-connected 4-connected minimal 4-connected

(V,X)

(V,X-e₁) ➔ (V,X-e₁+Y₁₁) ➔ (V,X₁)

(V,X-e₁) ➔ (V,X-e₁+Y₁₂) ➔ (V,X₂)

(V,X-e₁) ➔ (V,X-e₁+Y₁₃) ➔ (V,X₃)

(V,X-e₂) ➔ (V,X-e₂+Y₂₁) ➔ (V,X₄)

(V,X-e₂) ➔ (V,X-e₂+Y₂₂)

(V,X-e₃) ➔ (V,X-e₃+Y₃₂)
Generating Neighbors

minimal 4-connected  3-connected  4-connected  minimal 4-connected

\( (V, X) \)

\( (V, X-e_1) \) \( (V, X-e_1+Y_{11}) \) \( (V, X_1) \) 

\( (V, X-e_2) \) \( (V, X-e_1+Y_{12}) \) \( (V, X_2) \) 

\( (V, X-e_3) \) \( (V, X-e_1+Y_{13}) \) \( (V, X_3) \) 

\( (V, X-e_2+Y_{21}) \) \( (V, X-e_2+Y_{22}) \) \( (V, X_4) \) 

\( (V, X-e_3+Y_{32}) \) 

\( \approx 2|X| \)
Black Box

\[ G = (V, E) - k\text{-vertex connected graph} \]
\[ (V, X) - \text{minimal } k\text{-vertex connected subgraph of } G \]

**Input:**

\((k - 1)\text{-vertex connected graph } (V, X \setminus e)\)

list of edges \(E \setminus X\)

**Output:**

list of minimal edge sets \(Y \subseteq E \setminus X\) such that
\((V, (X \setminus e) \cup Y)\) is \(k\)-vertex connected
$k$-Separators

minimal 4-vertex connected graph
$k$-Separators

3-vertex connected graph
$k$-Separators

3-vertex connected graph

3-separator
$k$-Separators

3-vertex connected graph

3-separator

3-source
Lattice of $k$-Sources
Black Box

Input:

Available edges: (2,11), (8,t), (s,9), (s,1), (2,12)

3-vertex connected

3-separators: {2,3,4}, {5,3,4}, {10,3,4}, {5,8,9}, {10,8,9}, {5,11,9}, {5,8,12}, {10,11,9},
{10,8,12}, {5,11,12}, {10,11,12}
Input:
Available edges: (2,11), (8,t), (s,9), (s,1), (2,12)  
Add (2,11)

3-vertex connected

3-separators: {2,3,4}, {5,11,9}, {10,11,9}, {5,11,12}, {10,11,12}
**Input:**

Available edges: (2,11), (8,t), (s,9), (s,1), (2,12)  
Add (8,t)

3-vertex connected

3-separators: {2,3,4}
Input:
Available edges: (2,11), (8,t), (s,9), (s,1), (2,12)  Add (s,9)

3-vertex connected

4-vertex connected

3-separators:
Black Box

Input:
Available edges: (2,11), (8,t), (s,9), (s,1), (2,12)

Output: blue edges

3-vertex connected

4-vertex connected
Black Box - Key Observation

edge $\leftrightarrow$ sublattice

$\{s, 1, 2, 3, 4, 6, 7\}$ belongs to sublattice corresponding to edge $(2, 11)$
Black Box - Key Observation

edge $(2, 11) \longleftrightarrow$ black sublattice

$\{s, 1, 2, 3, 4, 5, 6, 7, 8\}$

$\{s, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\{s, 1, 2, 3, 4, 5, 6, 7\}$

$\{s, 1, 2, 3, 4, 6, 7, 8\}$

$\{s, 1, 2, 3, 4, 6, 7, 9\}$

$\{s, 1, 2, 3, 4, 5, 6\}$

$\{s, 1, 2, 3, 4, 6\}$

$\{s, 1, 2, 3, 4, 5\}$

$\{s, 1, 2, 3\}$

$\{s, 1\}$

$\{s\}$
edge $(8, t) \leftrightarrow \text{red sublattice}$
Black Box - Key Observation

edge $(s, 9) \leftrightarrow$ blue sublattice
Sublattice Covers

Input: k-conformal hypergraph
Output: list of minimal transversals

Input: collection of sublattices
Output: list of minimal covers

hypergraph whose edges are sublattices is 2-Helly
Sublattice Covers

Input: collection of sublattices
Output: list of minimal covers

Input: k-conformal hypergraph
Output: list of minimal transversals

The hypergraph whose edges are sublattices is 2-Helly.

Incremental polynomial time algorithm [Boros, Elbassioni, Gurvich, Khachyian 2004]
Complexity Results

\[ n = \# \text{ of vertices}, \quad m = \# \text{ of edges}, \quad N = \# \text{ of subgraphs} \]

- \( k \)-vertex connected: \( O(N^3nm^3 + N^2n^4m^5 + Nn^k m^2) \)
Complexity Results

\( n = \# \text{ of vertices}, \ m = \# \text{ of edges}, \ N = \# \text{ of subgraphs} \)

- **\( k \)-vertex connected:** \( O(N^3nm^3 + N^2n^4m^5 + Nn^k m^2) \)

- for \( k = 1 \) (spanning trees): \( O(Nn) \)

- for \( k = 2, 3 \): \( O(N^2 \log(N)m^2 + N^2m^3) \) (improvement due to decomposition theory)

- when \( k \) is part of the input: OPEN
Two-Terminal Reliability

\[ \text{Independent probabilities of edge failure} \]

\[ P \text{rob}(\exists \text{ an operating } s-t \text{ path}) \]
Two-Terminal Reliability

Independent probabilities of edge failure

\[
\text{Prob}(\exists \text{ an operating } s-t \text{ path})
\]

Hard to compute since counting 1-element, 2-element, \ldots, |E|-element s-t cuts is hard
How To Compute $\text{Prob}(\exists \text{ an operating } s-t \text{ path})$

1. Generate all $N$ $s-t$ paths
2. Calculate $\text{Prob}(\exists \text{ an operating } s-t \text{ path})$ using the inclusion-exclusion principle ($A_i = \text{ path } i \text{ is operating}$)

$$\text{Prob}(\exists \text{ an operating } s-t \text{ path}) = \text{Prob}(A_1 \cup \ldots \cup A_N)$$

$$= \sum_{k=1}^{N} (-1)^{k-1} S_k$$

where $S_k = \sum_{l \leq l_1 \leq \ldots \leq l_k \leq N} \text{Prob}(A_{l_1} \cap \ldots \cap A_{l_k})$
How To Compute $\text{Prob}(\exists \text{ an operating } s-t \text{ path})$

1. Generate all $N$ $s-t$ paths

2. Calculate $\text{Prob}(\exists \text{ an operating } s-t \text{ path})$ using the inclusion-exclusion principle ($A_i = \text{path } i \text{ is operating}$)

$$\text{Prob}(\exists \text{ an operating } s-t \text{ path}) = \text{Prob}(A_1 \cup \ldots \cup A_N)$$

$$= \sum_{k=1}^{N} (-1)^{k-1} S_k$$

where $S_k = \sum_{l \leq l_1 \leq \ldots \leq l_k \leq N} \text{Prob}(A_{l_1} \cap \ldots \cap A_{l_k})$

$\approx 2^{|E|}$ requires $2^N$ operations
How To Approximate $\text{Prob}(\exists \text{ an operating } s\text{-}t \text{ path})$

Generate only $K << N$ $s\text{-}t$ paths

Approximate $\text{Prob}(\exists \text{ an operating } s\text{-}t \text{ path})$ using only first $L << N$ moments $S_1, \ldots, S_L$
### Other Generation Problems

<table>
<thead>
<tr>
<th>Reliability Measures</th>
<th>Generation Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Prob}(\exists \text{ an operating } s\text{-}t \text{ path}) )</td>
<td>( s\text{-}t \text{ paths} ) [Read, Tarjan]</td>
</tr>
<tr>
<td></td>
<td>( \text{minimal } s\text{-}t \text{ cuts} ) [Tsukiyama et al]</td>
</tr>
</tbody>
</table>
# Other Generation Problems

<table>
<thead>
<tr>
<th>Reliability Measures</th>
<th>Generation Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Prob(\exists$ an operating $s-t$ path)</td>
<td>$s-t$ paths [Read, Tarjan]</td>
</tr>
<tr>
<td>$Prob(\text{ all nodes can communicate})$</td>
<td>minimal $s-t$ cuts [Tsukiyama et al]</td>
</tr>
<tr>
<td></td>
<td>spanning trees [Read, Tarjan]</td>
</tr>
<tr>
<td></td>
<td>minimal cuts</td>
</tr>
<tr>
<td></td>
<td>$k$-connected spanning subgraphs</td>
</tr>
</tbody>
</table>
### Other Generation Problems

<table>
<thead>
<tr>
<th>Reliability Measures</th>
<th>Generation Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Prob}(\exists \text{an operating } s\text{-}t \text{ path}) )</td>
<td>( s\text{-}t \text{ paths} ) [Read, Tarjan]</td>
</tr>
<tr>
<td>( \text{Prob}(\text{all nodes can communicate}) )</td>
<td>minimal ( s\text{-}t \text{ cuts} ) [Tsukiyama et al]</td>
</tr>
<tr>
<td>( \text{Prob}(\exists s_{1}\text{-}t_{1} \text{ path } \land \ldots \land s_{k}\text{-}t_{k} \text{ path}) )</td>
<td>spanning trees [Read, Tarjan]</td>
</tr>
<tr>
<td></td>
<td>minimal cuts</td>
</tr>
<tr>
<td></td>
<td>( k)-connected spanning subgraphs</td>
</tr>
<tr>
<td></td>
<td>path conjunctions [Boros et al]</td>
</tr>
<tr>
<td></td>
<td>cut conjunctions [ISAAC 05]</td>
</tr>
</tbody>
</table>
Other Generation Problems

Reliability Measures

\[ \text{Prob}(\exists \text{ an operating } s-t \text{ path}) \]

\[ \text{Prob}(\text{ all nodes can communicate}) \]

\[ \text{Prob}(\exists s_1-t_1 \text{ path } \land \ldots \land s_k-t_k \text{ path}) \]

Generation Problems

\[ s-t \text{ paths} \quad \text{[Read, Tarjan]} \]

\[ \text{minimal } s-t \text{ cuts} \quad \text{[Tsukiyama et al]} \]

\[ \text{spanning trees} \quad \text{[Read, Tarjan]} \]

\[ \text{minimal cuts} \]

\[ k\text{-connected spanning subgraphs} \]

\[ \text{path conjunctions} \quad \text{[Boros et al]} \]

\[ \text{cut conjunctions} \quad \text{[ISAAC 05]} \]

Network consists of subnetworks whose edges fail simultaneously
# Other Generation Problems

## Reliability Measures

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Prob}(\exists \text{ an operating } s-t \text{ path}) )</td>
<td>( s-t ) paths [Read, Tarjan]</td>
</tr>
<tr>
<td>( \text{Prob}(\text{ all nodes can communicate}) )</td>
<td>( s-t ) cuts [Tsukiyama et al]</td>
</tr>
<tr>
<td>( \text{Prob}(\exists s_1-t_1 \text{ path} \land \ldots \land s_k-t_k \text{ path}) )</td>
<td>spanning trees [Read, Tarjan], minimal cuts, ( k )-connected spanning subgraphs path conjunctions [Boros et al], cut conjunctions [ISAAC 05]</td>
</tr>
</tbody>
</table>

## Generation Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Prob}(\text{ all nodes can communicate}) )</td>
<td>connected spanning sets in matroids [ESA 06]</td>
</tr>
</tbody>
</table>

Network consists of subnetworks whose edges fail simultaneously
Vertex Generation
Minkowski, Weyl: Two representations of a polyhedron $P$
Minkowski, Weyl: Two representations of a polyhedron $P$

halfspace representation

$P = \{ x : Ax \leq b \}$

\[-x_1 + x_2 \leq 3\]
\[-4x_1 + x_2 \leq -3\]
\[-x_1 - x_2 \leq -2\]
\[x_1 - x_2 \leq 4\]
Minkowski, Weyl: Two representations of a polyhedron $P$

**Halfspace representation**

$$P = \{ x : Ax \leq b \}$$

**Vertex representation**

$$P = \text{conv}\{v_1, \ldots, v_k\} + \text{cone}\{d_1, \ldots, d_l\}$$

Constraints:

- $-x_1 + x_2 \leq 3$
- $-4x_1 + x_2 \leq -3$
- $-x_1 - x_2 \leq -2$
- $x_1 - x_2 \leq 4$

Vertex:

$$(x_1, x_2) = \lambda_1 (2, 5) + \lambda_2 (1, 1) + \lambda_3 (3, -1) + \mu (1, 1)$$

with:

- $\lambda_1 + \lambda_2 + \lambda_3 = 1$
- $\lambda_1, \lambda_2, \lambda_3, \mu \geq 0$
Two problems

Vertex Generation

Input: halfspace representation
Output: list of vertices and extreme directions

Facet Generation

Input: vertex representation
Output: list of halfspaces

Vertex Generation and Facet Generation are equivalent

\[ P^* = \{ y : x^T y \leq 1 \text{ for all } x \in P \} \]

vertices of \( P \) \( \leftrightarrow \) facets of \( P^* \)
facets of \( P \) \( \leftrightarrow \) vertices of \( P^* \)
$2n$ inequalities

$0 \leq x_1 \leq 1$

$0 \leq x_2 \leq 1$

$\vdots$

$0 \leq x_n \leq 1$

$2^n$ vertices

$\{0, \ldots, 0, 0\}, \{0, \ldots, 0, 1\},$

$\{0, \ldots, 1, 0\}, \{0, \ldots, 1, 1\},$

$\ldots, \{1, \ldots, 1, 1\}$
Irredundant Infeasible Subsystems

IIS Generation

**Input:** infeasible system $Ax \leq b$

**Output:** list of minimal infeasible subsystems

IIS Generation and Vertex Generation are equivalent

vertices of $\{y : y^T A = 0, y \geq 0, b^T y = 1\} \leftrightarrow$ minimal infeasible subsystems of $Ax \geq b$ [Gleeson, Ryan]
Previous Results

Efficient algorithms for special classes:

- simple polyhedra (every vertex incident with $n$ facets) [Avis, Fukuda 1992]
- simplicial polyhedra (the dual of simple polyhedra) [Bremner, Fukuda, Marzetta 1998]
- network polytopes [Provan 1994]
- polytopes with zero-one vertices [Bussieck, Lubbecke 1998]
- polyhedra in which every facet defining inequality involves at most two nonzero coefficients [Abdullahi 2003]

All known algorithms perform poorly in general case [Fukuda, Bremner, Seidel 1995]
New Results

Generating vertices of an unbounded polyhedron is NP-hard [SODA 06, DCG].

Generating vertices of a bounded polyhedron which belong to an open half-space $H$ is NP-hard [SODA 06, DCG].
Proof - Negative Circuits in Digraphs

Negative Circuits Generation Problem
**Input**: directed graph with edge weights
**Output**: list of all negative circuits

Negative Circuits Extension Problem
**Input**: directed graph with edge weights, \( \mathcal{X} \) - a collection of negative circuits
**Output**: Yes, if there is negative circuit which does not belong to \( \mathcal{X} \) No, if \( \mathcal{X} \) contains all negative circuits
Proof

Strategy

3-SAT \leq_p \text{Negative Circuits Extension Problem} \leq_p \text{Negative Circuits Generation Problem}

our result

simulation (standard technique)
Proof

Strategy

3-SAT $\leq_P$ Negative Circuits Extension Problem $\leq_P$ Negative Circuits Generation Problem

our result

simulation (standard technique)

Thus generating negative circuits is NP-hard
Simulation

\( n \) = input size, \( N \) = total number of negative circuits
\( \mathcal{X} \) a collection of negative circuits

Suppose: algorithm \( \textbf{A} \) generates all \( N \) negative circuits in time \( \phi(n, N) \), where \( \phi(n, N) \) is \( \text{poly}(n, N) \)

Run \( \textbf{A} \) and interrupt it after \( \phi(n, |\mathcal{X}|) + 1 \) time (if it did not stop before)
Simulation

\( n = \) input size, \( N = \) total number of negative circuits
\( \mathcal{X} = \) a collection of negative circuits

Suppose: algorithm \( A \) generates all \( N \) negative circuits in time \( \phi(n, N) \), where \( \phi(n, N) \) is \( \text{poly}(n, N) \)

Run \( A \) and interrupt it after \( \phi(n, |\mathcal{X}|) + 1 \) time (if it did not stop before)

\( A \) outputs a negative circuit not in \( \mathcal{X} \) \( \Rightarrow \) Yes
Simulation

\( n = \text{input size}, \ N = \text{total number of negative circuits} \)
\( \mathcal{X} \ ) a collection of negative circuits

Suppose: algorithm \( \textbf{A} \) generates all \( N \) negative circuits in time \( \phi(n, N) \), where \( \phi(n, N) \) is \( \text{poly}(n, N) \)

Run \( \textbf{A} \) and interrupt it after \( \phi(n, |\mathcal{X}|) + 1 \) time (if it did not stop before)

\( \textbf{A} \) outputs a negative circuit not in \( \mathcal{X} \) \( \Rightarrow \) Yes

\( \textbf{A} \) does not output a negative circuit not in \( \mathcal{X} \) and it does not halt after \( \phi(n, |\mathcal{X}|) + 1 \) time (interrupted) \( \Rightarrow \) Yes
Simulation

\[ n = \text{input size}, \quad N = \text{total number of negative circuits} \]
\[ \mathcal{X} = \text{a collection of negative circuits} \]

Suppose: algorithm \( A \) generates all \( N \) negative circuits in time \( \phi(n, N) \), where \( \phi(n, N) \) is \( \text{poly}(n, N) \)

Run \( A \) and interrupt it after \( \phi(n, |\mathcal{X}|) + 1 \) time (if it did not stop before)

- \( A \) outputs a negative circuit not in \( \mathcal{X} \) \( \Rightarrow \) Yes
- \( A \) does not output a negative circuit not in \( \mathcal{X} \) and it does not halt after \( \phi(n, |\mathcal{X}|) + 1 \) time (interrupted) \( \Rightarrow \) Yes
- \( A \) outputs all negative circuits of \( \mathcal{X} \) and halts \( \Rightarrow \) No
Our Result - Proof
Proof - Circulation Cone

\[ A - \text{incidence matrix of a digraph} \]
\[ \{y : y^\top A = 0, y \geq 0\} \]

extreme directions ↔ circuits

extreme direction \( y \) ↔ circuit \( \{(u, v) \in E : y_{uv} \neq 0\} \)
circuit \( C \) ↔ extreme direction: characteristic vector of \( C \)
Proof - Circulation Polyhedron

\[ A - \text{incidence matrix of a digraph, } b - \text{vector of edge weights} \]

\[ P = \{ y : y^T A = 0, y \geq 0, b^T y = -1 \} \]

vertices $\leftrightarrow$ negative circuits
Proof - Circulation Polyhedron

$A$ - incidence matrix of a digraph, $b$ - vector of edge weights

$P = \{ y : y^T A = 0, y \geq 0, b^T y = -1 \}$

$P$ is unbounded

$y_1$ negative circuit ($b^T y_1 = -1$)

$y_2$ positive circuit ($b^T y_2 = 1$)

$y_1 + t(y_1 + y_2) \in P \ \forall t \geq 0$
Proof - Circulation Polyhedron

\[ A - \text{incidence matrix of a digraph, } b - \text{vector of edge weights} \]
\[ P = \{ y : y^T A = 0, \ y \geq 0, \ b^T y = -1 \} \]

**Corollary 1.** *Generating vertices of an unbounded polyhedron is NP-hard.*
Proof - Circulation Polyhedron

\( A \) - incidence matrix of a digraph, \( b \) - vector of edge weights

\[ P = \{ y : y^T A = 0, y \geq 0, b^T y = -1 \} \]

vertices of \( \{ y : y^T A = 0, y \geq 0, b^T y = -1 \} \) \( \leftrightarrow \) minimal infeasible subsystems of \( Ax \geq -b \)

**Corollary 2.** Generating minimal infeasible subsystems of a system of linear inequalities is NP-hard.
Proof - Circulation Polytope

\[ P = \{ y : y^T A = 0, \ y \geq 0, \ 1^T y = 1 \} \] is bounded

vertices ↔ circuits
Proof - Circulation Polytope

\[ P = \{ y : y^T A = 0, \ y \geq 0, \ 1^T y = 1 \} \text{ is bounded} \]

\[ H = \{ y : b^T y < 0 \} \]

- vertices $\leftrightarrow$ circuits
- vertices which belong to $H$ $\leftrightarrow$ negative circuits
Proof - Circulation Polytope

\[ P = \{ y : y^T A = 0, \ y \geq 0, \ 1^T y = 1 \} \text{ is bounded} \]

\[ H = \{ y : b^T y < 0 \} \]

Corollary 5. \textit{Generating vertices of } \( P \) \textit{ not in } \( H \) \textit{ is NP-hard.}
Vertex Generation For Bounded Polyhedra

NP-hard

open half-space H

NP-hard

open
Monotone Generation Problem
Monotone Generation Problem

**Input**: set $E$, monotone Boolean function $\pi : 2^E \rightarrow \{0, 1\}$

**Output**: list of all minimal subsets of $E$ satisfying $\pi$
Monotone Generation Problem

**Input:** set $E$, monotone Boolean function $\pi : 2^E \rightarrow \{0, 1\}$

**Output:** list of all minimal subsets of $E$ satisfying $\pi$

$E = \{1, 2, 3, 4\}$

red $\pi = 1$

blue $\pi = 0$
Monotone Generation Problem

**Input:** set $E$, monotone Boolean function $\pi : 2^E \rightarrow \{0, 1\}$

**Output:** list of all minimal subsets of $E$ satisfying $\pi$

$E = \{1, 2, 3, 4\}$

red $\pi = 1$

blue $\pi = 0$
Monotone Generation Problem

Let $G = (V, E)$ be a graph, and consider a subset $X$ of edges.

Define

$$
\pi(X) = \begin{cases} 
1, & \text{if } (V, X) \text{ is } k\text{-vertex connected;} \\
0, & \text{otherwise.}
\end{cases}
$$
Monotone Generation Problem

- graph $G = (V, E)$, subset $X$ of edges

$$\pi(X) = \begin{cases} 1, & \text{if } (V, X) \text{ is } k\text{-vertex connected}; \\ 0, & \text{otherwise}. \end{cases}$$

- infeasible system $Ax \geq b$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$

matrix $A(X)$ of rows of $A$ whose indices belong to $X \subseteq \{1, \ldots, m\}$

$$\pi(X) = \begin{cases} 1, & \text{if } A(X)x \geq b \text{ is an infeasible subsystem}; \\ 0, & \text{otherwise}. \end{cases}$$