ON THE BOOLEAN WIDTH OF A GRAPH: STRUCTURE AND APPLICATIONS

I. Adler, B.-M. Bui-Xuan, Y. Rabinovich, G. Renault, J.A. Telle, M. Vatshelle
WG, Ζαρός, Κρήτη, 29 June 2010
Overview of results

- Boolean width of $G(n, p)$ is $\Theta\left(\frac{(\log n)^2}{p}\right)$
- Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c \cdot k^2})$, such as
  - **Max Independent Set**,  
  - **Min Dominating Set**,  
  - **Max Induced $k$-Regular Subgraph**,  
  - **$H$-Covering**,  
  - **$H$-Coloring**,  
  - ...

  Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).

- Relations to other parameters
Overview of results

- Boolean width of $G(n, p)$ is $\Theta(\frac{(\log n)^2}{p})$
- Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c \cdot k^2})$, such as
  - MAX INDEPENDENT SET,
  - MIN DOMINATING SET,
  - MAX INDUCED $k$-REGULAR SUBGRAPH,
  - $H$-COVERING,
  - $H$-COLORING,
  - ...

Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).
- Relations to other parameters
Overview of results

- Boolean width of $G(n, p)$ is $\Theta(\frac{(\log n)^2}{p})$
- Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c\cdot k^2})$, such as
  - **MAX INDEPENDENT SET**,
  - **MIN DOMINATING SET**,
  - **MAX INDUCED $k$-REGULAR SUBGRAPH**,
  - **$H$-COVERING**,
  - **$H$-COLORING**,
  - ...  

  Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).
- Relations to other parameters
Overview of results

- Boolean width of $G(n, p)$ is $\Theta\left(\frac{(\log n)^2}{p}\right)$
- Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c\cdot k^2})$, such as
  - Max Independent Set,
  - Min Dominating Set,
  - Max Induced $k$-Regular Subgraph,
  - $H$-Covering,
  - $H$-Coloring,
  - ...

  Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).
- Relations to other parameters
Overview of results

- Boolean width of $G(n, p)$ is $\Theta\left(\frac{(\log n)^2}{p}\right)$
- Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c \cdot k^2})$, such as
  - MAX INDEPENDENT SET,
  - MIN DOMINATING SET,
  - MAX INDUCED $k$-REGULAR SUBGRAPH,
  - $H$-COVERING,
  - $H$-COLORING,
  - ...

Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).

- Relations to other parameters
Overview of results

• Boolean width of $G(n, p)$ is $\Theta\left(\frac{(\log n)^2}{p}\right)$

• Solve a large class of problems on $k$-bounded Boolean width in time $O^*(2^{c \cdot k^2})$, such as
  - Max Independent Set,
  - Min Dominating Set,
  - Max Induced $k$-Regular Subgraph,
  - $H$-Covering,
  - $H$-Coloring,
  - ...

Improving previous results by Gerber and Kobler for this class of problems (on bounded clique-width).

• Relations to other parameters
Contents

1. Boolean width
2. Boolean width of random graphs
3. Algorithms on bounded Boolean width (if time permits)
The Boolean cost for a cut

Given a graph $G = (V, E)$, a cut is a partition $(X, \overline{X})$ of $V$ such that $X \cup \overline{X} = V$. A cut $(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Define the neighbor set function $N_X$ by

$$N_X : \text{Pow}(X) \to \text{Pow}((\overline{X})$$

$$A \mapsto N(A) \cap \overline{X}$$

Note: In general, $N_X$ is neither injective nor surjective.

**Definition**

$$\text{Boolcost}(X) := \log_2 |\mathcal{N}_X(\text{Pow}(X))|.$$ 

**Lemma**

$$\text{Boolcost}(X) = \text{Boolcost}(\overline{X}).$$
The Boolean cost for a cut

$G$ graph.
$(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Define the neighbor set function $\mathcal{N}_X$ by

$$\mathcal{N}_X : \text{Pow}(X) \rightarrow \text{Pow}(\overline{X})$$
$$A \mapsto N(A) \cap \overline{X}$$

Note: In general, $\mathcal{N}_X$ is neither injective nor surjective.

**Definition**

$\text{Boolcost}(X) := \log_2 |\mathcal{N}_X(\text{Pow}(X))|$.

**Lemma**

$\text{Boolcost}(X) = \text{Boolcost}(\overline{X})$. 
The Boolean cost for a cut

$G$ graph.
$(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Define the neighbor set function $\mathcal{N}_X$ by

$$\mathcal{N}_X : \text{Pow}(X) \rightarrow \text{Pow}(\overline{X})$$

$$A \mapsto N(A) \cap \overline{X}$$

Note: In general, $\mathcal{N}_X$ is neither injective nor surjective.

Definition

$\text{Boolcost}(X) := \log_2 |\mathcal{N}_X (\text{Pow}(X))| .$

Lemma

$\text{Boolcost}(X) = \text{Boolcost}(\overline{X})$. 
**The Boolean cost for a cut**

Let $G$ be a graph. $(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Define the neighbor set function $\mathcal{N}_X$ by

$$\mathcal{N}_X : \text{Pow}(X) \rightarrow \text{Pow}(\overline{X})$$

$$A \mapsto N(A) \cap \overline{X}$$

Note: In general, $\mathcal{N}_X$ is neither injective nor surjective.

**Definition**

$$\text{Boolcost}(X) := \log_2 |\mathcal{N}_X (\text{Pow}(X))| .$$

**Lemma**

$$\text{Boolcost}(X) = \text{Boolcost}(\overline{X}).$$
The Boolean cost for a cut

\( G \) graph. 
\((X, \overline{X})\) with \( V(G) = X \cup \overline{X} \) is a cut of \( G \).

Define the neighbor set function \( \mathcal{N}_X \) by

\[
\mathcal{N}_X : \text{Pow}(X) \to \text{Pow}(\overline{X}) \\
A \mapsto N(A) \cap \overline{X}
\]

Note: In general, \( \mathcal{N}_X \) is neither injective nor surjective.

**Definition**

\( \text{Boolcost}(X) := \log_2 |\mathcal{N}_X(\text{Pow}(X))| \).

**Lemma**

\( \text{Boolcost}(X) = \text{Boolcost}(\overline{X}) \).
Boolean cost – another perspective

Let $(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ be a cut of $G$.

Let $M_X =$

\[
\begin{array}{c|cc}
& \overline{X} & X \\
\hline
X & \text{zeros} & \text{ones} \\
\end{array}
\]

be the adjacency matrix of the cut.

**Definition**

Let $\Sigma M_X := \{ \text{Boolean sums of all subsets of row vectors} \}$. Here: $1 + 1 = 1$.

**Observation**

$\text{Boolcost}(X) = \log_2 |\Sigma M_X|$.
Boolean cost – another perspective

$G$ graph.
$(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Let $M_X = \begin{array}{c|c}
X & \overline{X} \\
\hline
zeros & \text{zeros} \\
and & \text{ones}
\end{array}$
be the adjacency matrix of the cut.

**Definition**

Let $\Sigma M_X := \{ \text{Boolean sums of all subsets of row vectors} \}$. Here: $1 + 1 = 1$.

**Observation**

$\text{Boolcost}(X) = \log_2 |\Sigma M_X|$.
Boolean cost – another perspective

\( G \) graph. 
\((X, \overline{X}) \) with \( V(G) = X \cup \overline{X} \) is a cut of \( G \).

Let \( M_X = \begin{array}{c|c}
X & \overline{X} \\
\hline
zeros & and \\
ones & 
\end{array} \)
be the adjacency matrix of the cut.

**Definition**

Let \( \Sigma M_X := \{ \text{Boolean sums of all subsets of row vectors} \} \).
Here: \( 1 + 1 = 1 \).

**Observation**

\( \text{Boolcost}(X) = \log_2 |\Sigma M_X| \).
Boolean cost – another perspective

$G$ graph. 
$(X, \overline{X})$ with $V(G) = X \cup \overline{X}$ is a cut of $G$.

Let $M_X = \begin{pmatrix} \overline{X} \\ X \end{pmatrix}$ be the adjacency matrix of the cut.

**Definition**

Let $\Sigma M_X := \{ Boolean \ sums \ of \ all \ subsets \ of \ row \ vectors \}$. Here: $1 + 1 = 1$.

**Observation**

$\text{Boolcost}(X) = \log_2 |\Sigma M_X|$.
Boolean width

A graph $G$. A decomposition tree for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \to V(G)$ is a bijection

Example

$V(G) = \{r, s, u, v, w\}$

Definition

For $e \in E(T)$:

$\text{Boolcost}(e) := \text{Boolcost}(X)$,

$\text{Boolcost}(T, \delta) := \max \{ \text{Boolcost}(e) \mid e \in E(T) \}$,

$\text{Boolw}(G) := \min \{ \text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G \}$
**Boolean width**

A graph $G$. A decomposition tree for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$:

- $\text{Boolcost}(e) := \text{Boolcost}(X)$,
- $\text{Boolcost}(T, \delta) := \max\{\text{Boolcost}(e) \mid e \in E(T)\}$,
- $\text{Boolw}(G) := \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}$
Boolean width

$G$ graph. A **decomposition tree** for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$: $\text{Boolcost}(e) := \text{Boolcost}(X)$,

$\text{Boolcost}(T, \delta) := \max\{\text{Boolcost}(e) \mid e \in E(T)\}$,

$\text{Boolw}(G) := \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}$
**Boolean width**

$G$ graph. A *decomposition tree* for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$: $
\begin{align*}
\text{Boolcost}(e) &:= \text{Boolcost}(X), \\
\text{Boolcost}(T, \delta) &:= \max\{\text{Boolcost}(e) \mid e \in E(T)\}, \\
\text{Boolw}(G) &:= \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}
\end{align*}$
**Boolean width**

A graph $G$ has a decomposition tree for $G$ is a pair $(T, \delta)$, where
- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$: \( \text{Boolcost}(e) := \text{Boolcost}(X) \),
\( \text{Boolcost}(T, \delta) := \max\{ \text{Boolcost}(e) \mid e \in E(T) \} \),
\( \text{Boolw}(G) := \min\{ \text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G \} \)
**Boolean width**

$G$ graph. A decomposition tree for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

\[\begin{align*}
\text{Defnition} & \\
\text{For } e \in E(T): \text{Boolcost}(e) := \text{Boolcost}(X), \\
\text{Boolcost}(T, \delta) := \max \{ \text{Boolcost}(e) \mid e \in E(T) \}, \\
\text{Boolw}(G) := \min \{ \text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G \}
\end{align*}\]
**Boolean width**

G graph. A **decomposition tree** for G is a pair \((T, \delta)\), where
- \(T\) is a cubic tree
- \(\delta : \text{Leaves}(T) \rightarrow V(G)\) is a bijection

**Example**

\[V(G) = \{r, s, u, v, w\}\]

**Definition**

For \(e \in E(T)\): \(\text{Boolcost}(e) := \text{Boolcost}(X)\),

\[
\text{Boolcost}(T, \delta) := \max\{\text{Boolcost}(e) \mid e \in E(T)\},
\]

\(\text{Boolw}(G) := \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}\)
**Boolean width**

A graph $G$. A **decomposition tree** for $G$ is a pair $(T, \delta)$, where

- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$: $\text{Boolcost}(e) := \text{Boolcost}(X)$,

$\text{Boolcost}(T, \delta) := \max\{\text{Boolcost}(e) \mid e \in E(T)\}$,

$\text{Boolw}(G) := \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}$
**Boolean width**

$G$ graph. A **decomposition tree** for $G$ is a pair $(T, \delta)$, where
- $T$ is a cubic tree
- $\delta : \text{Leaves}(T) \rightarrow V(G)$ is a bijection

**Example**

$V(G) = \{r, s, u, v, w\}$

**Definition**

For $e \in E(T)$: \(\text{Boolcost}(e) := \text{Boolcost}(X)\),

\(\text{Boolcost}(T, \delta) := \max\{\text{Boolcost}(e) \mid e \in E(T)\}\),

\(\text{Boolw}(G) := \min\{\text{Boolcost}(T, \delta) \mid (T, \delta) \text{ dec. tree for } G\}\)
Boolean width: relation to other graph invariants

Fact (Bui-Xuan, Telle and Vatshelle)
Boolean width $\sim$ clique-width $\sim$ rank-width.

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- $H$ an incidence graph, then:
  $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

Rank-width
With our proof method: Simple proof for a similar theorem on rank-width.
A sightly stronger result was shown by Oum in Rank-width is less than or equal to branch-width using matroids.
Boolean width: relation to other graph invariants

Fact (Bui-Xuan, Telle and Vatshelle)

Boolean width $\sim$ clique-width $\sim$ rank-width.

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- If $H$ an incidence graph, then:
  $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

Rank-width

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in Rank-width is less than or equal to branch-width using matroids.
**Boolean width: relation to other graph invariants**

**Fact (Bui-Xuan, Telle and Vatshelle)**

*Boolean width* $\sim$ *clique-width* $\sim$ *rank-width*.

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- $H$ an incidence graph, then: $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

**Rank-width**

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in *Rank-width is less than or equal to branch-width* using matroids.
**Boolean width: relation to other graph invariants**

**Fact (Bui-Xuan, Telle and Vatshelle)**

Boolean width $\sim$ clique-width $\sim$ rank-width.

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- If $H$ an incidence graph, then:
  $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

**Rank-width**

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in *Rank-width is less than or equal to branch-width* using matroids.
**Boolean width: relation to other graph invariants**

**Fact (Bui-Xuan, Telle and Vatshelle)**

Boolean width $\sim$ clique-width $\sim$ rank-width.

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- If $H$ an incidence graph, then:
  $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

**Rank-width**

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in Rank-width is less than or equal to branch-width using matroids.
Boolean width: relation to other graph invariants

Fact (Bui-Xuan, Telle and Vatshelle)

Boolean width $\sim$ clique-width $\sim$ rank-width.

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- $H$ an incidence graph, then:
  $\text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H)$.

Rank-width

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in Rank-width is less than or equal to branch-width using matroids.
Boolean width: relation to other graph invariants

Fact (Bui-Xuan, Telle and Vatshelle)

Boolean width $\sim$ clique-width $\sim$ rank-width.

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

- $\text{Boolw}(G) \leq \text{branch-width}(G) \leq 2 \cdot \text{Boolw}(I(G))$.
- If $H$ is an incidence graph, then:
  \[ \text{Boolw}(H) \leq \text{branch-width}(H) \leq 2 \cdot \text{Boolw}(H). \]

Rank-width

With our proof method: Simple proof for a similar theorem on rank-width.

A sightly stronger result was shown by Oum in *Rank-width is less than or equal to branch-width* using matroids.
Contents

1. Boolean width
2. Boolean width of random graphs
3. Algorithms on bounded Boolean width
Random Graphs

Definition
\( G(n, p) := \text{random graph on } n \text{ vertices where every edge is chosen independently at random with probability } p. \)

Fact
Asymptotically almost surely, \( G(n, p) \) has
- tree-width / branch-width \( \Theta(n) \) (Bodlaender, Kloks 1992).
- rank-width \( \Theta(n) \) (Lee, Lee, Oum 2009).
- clique-width \( \Theta(n) \) (Johansson 1998).

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)
Asymptotically almost surely, \( \text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right) \).
Random Graphs

Definition
$G(n, p) := \text{random graph on } n \text{ vertices where every edge is chosen independently at random with probability } p.$

Fact
Asymptotically almost surely, $G(n, p)$ has
- tree-width / branch-width $\Theta(n)$ (Bodlaender, Kloks 1992).
- rank-width $\Theta(n)$ (Lee, Lee, Oum 2009).
- clique-width $\Theta(n)$ (Johansson 1998).

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)
Asymptotically almost surely, $\text{Boolw}(G(n, p)) = \Theta(\frac{(\log n)^2}{p})$. 
#### Random Graphs

**Definition**

\[ G(n, p) := \text{random graph on } n \text{ vertices where every edge is chosen independently at random with probability } p. \]

**Fact**

Asymptotically almost surely, \( G(n, p) \) has

- tree-width / branch-width \( \Theta(n) \) (Bodlaender, Kloks 1992).
- rank-width \( \Theta(n) \) (Lee, Lee, Oum 2009).
- clique-width \( \Theta(n) \) (Johansson 1998).

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

Asymptotically almost surely, 
\[ \text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right). \]
Random Graphs

**Definition**

\( G(n, p) := \text{random graph on } n \text{ vertices where every edge is chosen independently at random with probability } p. \)

**Fact**

Asymptotically almost surely, \( G(n, p) \) has

- tree-width / branch-width \( \Theta(n) \) (Bodlaender, Kloks 1992).
- rank-width \( \Theta(n) \) (Lee, Lee, Oum 2009).
- clique-width \( \Theta(n) \) (Johansson 1998).

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

Asymptotically almost surely, \( \text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right) \).
Boolean width of random graphs

**Theorem**

Asymptotically almost surely, \( \text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right) \).

Proof outline:

Claim (upper bound): a.a.s. any dec. tree for \( G := G(n, p) \) has \( \text{Boolcost} \mathcal{O}\left(\frac{(\log n)^2}{p}\right) \).

Claim (lower bound): a.a.s. any dec. tree for \( G := G(n, p) \) has a cut with \( \text{Boolcost} \Omega\left(\frac{(\log n)^2}{p}\right) \).
**Boolean width of random graphs**

**Theorem**
Asymptotically almost surely, $\text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right)$.

Proof outline:

Claim (upper bound): a.a.s. any dec. tree for $G := G(n, p)$ has $\text{Boolcost} = O\left(\frac{(\log n)^2}{p}\right)$.

Claim (lower bound): a.a.s. any dec. tree for $G := G(n, p)$ has a cut with $\text{Boolcost} = \Omega\left(\frac{(\log n)^2}{p}\right)$.
**Boolean width of random graphs**

**Theorem**

Asymptotically almost surely, $\text{Boolw}(G(n, p)) = \Theta\left(\frac{(\log n)^2}{p}\right)$.

Proof outline:

Claim (upper bound): a.a.s. any dec. tree for $G := G(n, p)$ has $\text{Boolcost} \ O\left(\frac{(\log n)^2}{p}\right)$.

Claim (lower bound): a.a.s. any dec. tree for $G := G(n, p)$ has a cut with $\text{Boolcost} \ \Omega\left(\frac{(\log n)^2}{p}\right)$.
Boolean width of random graphs: Upper bound

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} = O\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have $\text{Boolcost}(X) = O\left(\frac{(\log n)^2}{p}\right)$.

Let $k := \lceil \log n \rceil$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s. $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = O((\log n)^2)$.
Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have
$\text{Boolcost}(X) = \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s $\left| N(A) \cap \overline{X} \right| \geq \left| \overline{X} \right| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = \mathcal{O}\left((\log n)^2\right)$. 

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \in O\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have $\text{Boolcost}(X) = O\left(\frac{(\log n)^2}{p}\right)$.
Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s. $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = O\left((\log n)^2\right)$. 

\[ \square \]
Boolean width of random graphs: Upper bound

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} = O\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have
$\text{Boolcost}(X) = O\left(\frac{(\log n)^2}{p}\right)$.
Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s. $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$.
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is
  $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^\log n = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = O\left((\log n)^2\right)$.
Boolean width of random graphs: Upper bound

**Claim (upper bound)**

A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} = \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

**Proof.**

Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have

\[
\text{Boolcost}(X) = \mathcal{O}\left(\frac{(\log n)^2}{p}\right).
\]

Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,

  hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.

  Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^\log n = 2 \cdot 2^{(\log n)^2}$.

- Taking log yields: $\text{Boolcost}(X) = \mathcal{O}\left((\log n)^2\right)$. 

\[\square\]
Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} = \mathcal{O}(\frac{(\log n)^2}{p})$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have
\[\text{Boolcost}(X) = \mathcal{O}(\frac{(\log n)^2}{p}).\]
Let $k := \lceil \log n \rceil$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = \mathcal{O}\left((\log n)^2\right)$. 

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \ O\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have $\text{Boolcost}(X) = O\left(\frac{(\log n)^2}{p}\right)$.

Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \Sigma_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s. $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \Sigma_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is $\leq 2 \cdot \Sigma_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = O\left(\left(\log n\right)^2\right)$.
Boolean width of random graphs: Upper bound

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \in \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have

$\text{Boolcost}(X) = \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Let $k := \lfloor \log n \rfloor$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is
  $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.
**Boolean width of random graphs: Upper bound**

**Claim (upper bound)**
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \, O\left(\frac{(\log n)^2}{p}\right)$.

**Proof.**
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have 

$$\text{Boolcost}(X) = O\left(\frac{(\log n)^2}{p}\right).$$

Let $k := \lceil \log n \rceil$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.

- Lemma: If $|A| > k$, a.a.s $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,

- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.

- Hence a.a.s. the number of subsets of $\overline{X}$ is 
  $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.

- Taking log yields: $\text{Boolcost}(X) = O\left((\log n)^2\right)$. 

\qed
Boolean width of random graphs: Upper bound

Claim (upper bound)
A.a.s. any dec. tree for $G(n, p)$ has $\text{Boolcost} \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Proof.
Show: all cuts $(X, \overline{X})$ of $G(n, p)$ a.a.s. have

$\text{Boolcost}(X) = \mathcal{O}\left(\frac{(\log n)^2}{p}\right)$.

Let $k := \lceil \log n \rceil$. Let $p := 1/2$ for simplicity.

- The sets $A \subseteq X$ with $|A| \leq k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Lemma: If $|A| > k$, a.a.s $|N(A) \cap \overline{X}| \geq |\overline{X}| - k$,
- hence the sets $A \subseteq X$ with $|A| > k$ contribute $\leq \sum_{i=0}^{k} \binom{n}{i}$ subsets of $\overline{X}$.
- Hence a.a.s. the number of subsets of $\overline{X}$ is
  $\leq 2 \cdot \sum_{i=0}^{k} \binom{n}{i} \leq 2 \cdot n^k = 2 \cdot n^{\log n} = 2 \cdot 2^{(\log n)^2}$.
- Taking log yields: $\text{Boolcost}(X) = \mathcal{O}\left((\log n)^2\right)$.
Claim (Lower bound)
A.a.s. any dec. tree for $G(n, p)$ has a cut with $\text{Boolcost} \Omega\left(\frac{(\log n)^2}{p}\right)$.

Proof.

- Show: A.a.s. any cut $(X, \overline{X})$ of $G := G(n, p)$ has $\text{Boolcost}(X) = \Omega\left(\frac{\log^2 n}{p}\right)$.

- Note: Every dec. tree has a cut $(X, \overline{X})$ with $|X| \approx \frac{1}{3}|V(G)|$ and $|\overline{X}| \approx \frac{2}{3}|V(G)|$.

- Suffices to show: a.a.s. all such $(\frac{1}{3}, \frac{2}{3})$-cuts $(X, \overline{X})$ satisfy $\text{Boolcost}(X) = \Omega\left((\log n)^2\right)$. 
Claim (Lower bound)
A.a.s. any dec. tree for $G(n, p)$ has a cut with $\text{Boolcost} \Omega\left(\frac{(\log n)^2}{p}\right)$.

Proof.

- Show: A.a.s. any cut $(X, \overline{X})$ of $G := G(n, p)$ has $\text{Boolcost}(X) = \Omega\left(\frac{\log^2 n}{p}\right)$.

- Note: Every dec. tree has a cut $(X, \overline{X})$ with $|X| \approx \frac{1}{3}|V(G)|$ and $|\overline{X}| \approx \frac{2}{3}|V(G)|$.

- Suffices to show: a.a.s. all such $(\frac{1}{3}, \frac{2}{3})$-cuts $(X, \overline{X})$ satisfy $\text{Boolcost}(X) = \Omega\left((\log n)^2\right)$. 
Claim (Lower bound)

A.a.s. any dec. tree for $G(n, p)$ has a cut with $\text{Boolcost} \Omega\left(\frac{(\log n)^2}{p}\right)$.

Proof.

• Show: A.a.s. any cut $(X, \overline{X})$ of $G := G(n, p)$ has $\text{Boolcost}(X) = \Omega\left(\frac{\log^2 n}{p}\right)$.

• Note: Every dec. tree has a cut $(X, \overline{X})$ with $|X| \approx \frac{1}{3}|V(G)|$ and $|\overline{X}| \approx \frac{2}{3}|V(G)|$.

• Suffices to show: a.a.s. all such $(\frac{1}{3}, \frac{2}{3})$-cuts $(X, \overline{X})$ satisfy $\text{Boolcost}(X) = \Omega\left((\log n)^2\right)$. 
Claim (Lower bound)
A.a.s. any dec. tree for $G(n, p)$ has a cut with $\text{Boolcost} \Omega\left(\frac{(\log n)^2}{p}\right)$.

Proof.

- Show: A.a.s. any cut $(X, \overline{X})$ of $G := G(n, p)$ has
  $\text{Boolcost}(X) = \Omega\left(\frac{\log^2 n}{p}\right)$.

- Note: Every dec. tree has a cut $(X, \overline{X})$ with $|X| \approx \frac{1}{3}|V(G)|$ and $|\overline{X}| \approx \frac{2}{3}|V(G)|$.

- Suffices to show: a.a.s. all such $(\frac{1}{3}, \frac{2}{3})$-cuts $(X, \overline{X})$ satisfy $\text{Boolcost}(X) = \Omega\left( (\log n)^2 \right)$. 
**Boolean width of random graphs: Lower bound**

**Lemma**
A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

**Proof.**
- Show: Lemma holds for a cut with ‘good’ probability. (\(\sim\) for all cuts.)
- Show: a.a.s., \(N_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking \(\log\).) For this:
  - Define a digraph on \(N_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\sim\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
- Hence a.a.s., \(N_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
Lemma
A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[ \text{Boolcost}(X) = \Omega((\log n)^2). \]

Proof.

- Show: Lemma holds for a cut with ‘good’ probability. (\(\sim\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking \(\log\)). For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\sim\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
- Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
Lemma

A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

Proof.

- Show: Lemma holds for a cut with ‘good’ probability. (\(\sim\) for all cuts.)
  - Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
    - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
    - Remove some carefully selected vertices ("blemishes") from the digraph \(\sim\) bound on the degree.
    - A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
    - Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
**Boolean width of random graphs: Lower bound**

**Lemma**
A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

**Proof.**

- Show: Lemma holds for a cut with ‘good’ probability. (\(\rightsquigarrow\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\rightsquigarrow\) bound on the degree.
  - A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
  - Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
**Boolean width of random graphs: Lower bound**

**Lemma**
A.a.s. all $\left(\frac{1}{3}, \frac{2}{3}\right)$-cuts $(X, \overline{X})$ of $G(n, p)$ satisfy

$$
\text{Boolcost}(X) = \Omega((\log n)^2).
$$

**Proof.**

- Show: Lemma holds for a cut with ‘good’ probability. ($\sim$ for all cuts.)
- Show: a.a.s., $\mathcal{N}_X(Pow(X))$ contains an $\subseteq$-antichain of size $2^{(\log n)^2}$ (the Lemma then follows by taking log.) For this:
  - Define a digraph on $\mathcal{N}_X(Pow(X))$ by interpreting $\subseteq$ as $\leftarrow$.
  - Remove some carefully selected vertices ("blemishes") from the digraph $\sim$ bound on the degree.
  - A.a.s. the remaining digraph contains an independent set of size $2^{(\log n)^2}$.
  - Hence a.a.s., $\mathcal{N}_X(Pow(X))$ has an $\subseteq$-antichain of size $2^{(\log n)^2}$.
Boolean width of random graphs: Lower bound

Lemma
A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[ \text{Boolcost}(X) = \Omega((\log n)^2). \]

Proof.

- Show: Lemma holds for a cut with ‘good’ probability. (\(\sim\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\sim\) bound on the degree.
  - A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
  - Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
**Boolean width of random graphs: Lower bound**

**Lemma**
A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

**Proof.**

- Show: Lemma holds for a cut with ‘good’ probability. (\(\sim\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
- Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
- Remove some carefully selected vertices (”blemishes”) from the digraph \(\sim\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
- Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
**Boolean width of random graphs: Lower bound**

**Lemma**

A.a.s. all \((\frac{1}{3}, \frac{2}{3})\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

**Proof.**

- Show: Lemma holds for a cut with ‘good’ probability. (\(\leadsto\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\leadsto\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
- Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
**Boolean width of random graphs: Lower bound**

**Lemma**
A.a.s. all \((1/3, 2/3)\)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega((\log n)^2).
\]

**Proof.**

- Show: Lemma holds for a cut with ‘good’ probability. \((\sim)\) for all cuts.
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\sim\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
  - Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
Boolean width of random graphs: Lower bound

Lemma
A.a.s. all \( \left( \frac{1}{3}, \frac{2}{3} \right) \)-cuts \((X, \overline{X})\) of \(G(n, p)\) satisfy

\[
\text{Boolcost}(X) = \Omega\left((\log n)^2\right).
\]

Proof.

- Show: Lemma holds for a cut with ‘good’ probability. (\(\leadsto\) for all cuts.)
- Show: a.a.s., \(\mathcal{N}_X(Pow(X))\) contains an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\) (the Lemma then follows by taking log.) For this:
  - Define a digraph on \(\mathcal{N}_X(Pow(X))\) by interpreting \(\subseteq\) as \(\leftarrow\).
  - Remove some carefully selected vertices ("blemishes") from the digraph \(\leadsto\) bound on the degree.
- A.a.s. the remaining digraph contains an independent set of size \(2^{(\log n)^2}\).
- Hence a.a.s., \(\mathcal{N}_X(Pow(X))\) has an \(\subseteq\)-antichain of size \(2^{(\log n)^2}\).
Contents

1. Boolean width
2. Boolean width of random graphs
3. Algorithms on bounded Boolean width (if time permits)
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix. Let $D$ be $q \times q$. A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that for every $x \in V_i$: $|N(x) \cap V_j| \in D[i, j]$.

Example

$$D_{IS} = \begin{pmatrix} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{pmatrix}$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G$ $\iff$ $X$ is an independent set in $G$. Why?

MAX INDEPENDENT SET: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G.$
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that f.a.

$i, j \leq q$: for every $x \in V_i$: $|N(x) \cap V_j| \in D[i, j]$.

Example

$$D_{IS} = \left( \begin{array}{cc} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{array} \right)$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G \iff X$ is an independent set in $G$.

Why?

MAX INDEPENDENT SET: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.  

Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that f.a. $i, j \leq q$:

$$\text{for every } x \in V_i: |N(x) \cap V_j| \in D[i, j].$$

Example

$$D_{IS} = \begin{pmatrix} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{pmatrix}$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G$ $\iff$ $X$ is an independent set in $G$.

Why?

Max Independent Set: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. 
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that f.a. $i, j \leq q$:

$$\text{for every } x \in V_i: |N(x) \cap V_j| \in D[i,j].$$

Example

$$D_{IS} = \left( \begin{array}{cc} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{array} \right)$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G \iff X$ is an independent set in $G$.

Why?

<table>
<thead>
<tr>
<th></th>
<th>$X$</th>
<th>$\overline{X}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>${0}$</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$\overline{X}$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
</tbody>
</table>

Max Independent Set: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. 
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that f.a. $i, j \leq q$:

for every $x \in V_i$: $|N(x) \cap V_j| \in D[i, j]$.

Example

$$D_{IS} = \begin{pmatrix}
\{0\} & \mathbb{N} \\
\mathbb{N} & \mathbb{N}
\end{pmatrix}$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G$ $\iff$ $X$ is an independent set in $G$.

Why?

Max Independent Set: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. 
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that for every $x \in V_i$:

$$|N(x) \cap V_j| \in D[i, j].$$

Example

$$D_{IS} = \begin{pmatrix}
\{0\} & \mathbb{N} \\
\mathbb{N} & \mathbb{N}
\end{pmatrix}$$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G \iff X$ is an independent set in $G$.

Why?

<table>
<thead>
<tr>
<th></th>
<th>$X$</th>
<th>$\overline{X}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>${0}$</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$\overline{X}$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
</tbody>
</table>

MAX INDEPENDENT SET: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. 
Degree Constraint Matrix

A square matrix $D$ with entries being finite or cofinite subsets of $\mathbb{N}$ is called a degree constraint matrix.

Let $D$ be $q \times q$.

A $D$-partition of $G$ is a partition $(V_1, \ldots, V_q)$ of $V(G)$ such that f.a. $i, j \leq q$:

for every $x \in V_i$: $\left| N(x) \cap V_j \right| \in D[i, j]$.

Example

$D_{IS} = \left( \begin{array}{cc} \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} \end{array} \right)$

Then $(X, \overline{X})$ is a $D_{IS}$-partition of $G \iff X$ is an independent set in $G$.

Why?

<table>
<thead>
<tr>
<th></th>
<th>$X$</th>
<th>$\overline{X}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>$X$</td>
<td>${0}$</td>
</tr>
<tr>
<td>$\overline{X}$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
</tbody>
</table>

Max Independent Set: Given $G$, find maximal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. 
**Vertex subset problems**

Framework introduced by Proskurowski and Telle in 1997. Let $D$ be a $2 \times 2$ degree constraint matrix.

**OPT-$D$-PARTITIONING**

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task:</td>
<td>Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \bar{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- ...
**Vertex subset problems**

Framework introduced by Proskurowski and Telle in 1997. Let $D$ be a $2 \times 2$ degree constraint matrix.

### OPT-D-PARTITIONING

| Input:   | $G$  |
| Task:    | Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. |

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- ...
Vertex subset problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $2 \times 2$ degree constraint matrix.

<table>
<thead>
<tr>
<th>OPT-\textit{D-PARTITIONING}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G$</td>
</tr>
<tr>
<td><strong>Task:</strong> Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework
- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- $\ldots$
Vertex subset problems

Framework introduced by Proskurowski and Telle in 1997. Let $D$ be a $2 \times 2$ degree constraint matrix.

<table>
<thead>
<tr>
<th>OPT-$D$-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G$</td>
</tr>
<tr>
<td><strong>Task:</strong> Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- ...
Vertex subset problems

Framework introduced by Proskurowski and Telle in 1997. Let $D$ be a $2 \times 2$ degree constraint matrix.

<table>
<thead>
<tr>
<th>OPT-D-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G$</td>
</tr>
<tr>
<td><strong>Task:</strong> Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE** and **MAX PERFECT CODE**
- ...
**Vertex subset problems**

Framework introduced by Proskurowski and Telle in 1997.

Let $D$ be a $2 \times 2$ degree constraint matrix.

<table>
<thead>
<tr>
<th>OPT-$D$-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G$</td>
</tr>
<tr>
<td><strong>Task:</strong> Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- ...
Vertex subset problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $2 \times 2$ degree constraint matrix.

<table>
<thead>
<tr>
<th>OPT-$D$-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G$</td>
</tr>
<tr>
<td><strong>Task:</strong> Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Problems within this framework

- **MIN DOMINATING SET**
- **MAX INDUCED $k$-REGULAR SUBGRAPH**
- **MIN PERFECT CODE and MAX PERFECT CODE**
- ...
Boolean width and vertex subset problems

Let $D$ be a $2 \times 2$ degree constraint matrix.
Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

\begin{center}
\begin{tabular}{|l|}
\hline
\textsc{Opt-D-Partitioning($\mathcal{C}$)} \\
\hline
Input: & $G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$ \\
Task: & Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$. \\
\hline
\end{tabular}
\end{center}

\textbf{Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)}

\textsc{Opt-D-Partitioning($\mathcal{C}$)} can be solved in time

$$O(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),$$

where $n = |V(G)|$, $m = |E(G)|$, and $c$ a constant.
Boolean width and vertex subset problems

Let $D$ be a $2 \times 2$ degree constraint matrix. Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

**OPT-$D$-PARTITIONING($\mathcal{C}$)**

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task:</td>
<td>Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

OPT-$D$-PARTITIONING($\mathcal{C}$) can be solved in time

$$O(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),$$

where $n = |V(G)|$, $m = |E(G)|$, and $c$ a constant.
**Boolean width and vertex subset problems**

Let $D$ be a $2 \times 2$ degree constraint matrix.
Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

**OPT-$D$-PARTITIONING($\mathcal{C}$)**

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task:</td>
<td>Find optimal size of an $X \subseteq V(G)$, s.t. $(X, \overline{X})$ is a $D_{IS}$-partition of $G$.</td>
</tr>
</tbody>
</table>

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

**OPT-$D$-PARTITIONING($\mathcal{C}$) can be solved in time**

$$O(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),$$

**where** $n = |V(G)|$, $m = |E(G)|$, **and** $c$ **a constant.**
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) :=$ incidence matrix of $H$, but: replace all 1s with $\mathbb{N}$s and 0s with $\{0\}$s. Then:

$\left( V_1, \ldots, V_{|V(H)|} \right)$ is an $M(H)$-partition of $G \iff$ there is a homomorphism from $G$ to $H$.

$$M(K_3) = \begin{pmatrix} \{0\} & \mathbb{N} & \mathbb{N} \\ \mathbb{N} & \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} & \{0\} \end{pmatrix}$$

<table>
<thead>
<tr>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>${0}$</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>${0}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
</tbody>
</table>
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) :=$ incidence matrix of $H$, but: replace all 1s with $\mathbb{N}$s and 0s with $\{0\}$s. Then:

$(V_1, \ldots, V_{|V(H)|})$ is an $M(H)$-partition of $G$ $\iff$ there is a homomorphism from $G$ to $H$.

\[
M(K_3) = \begin{pmatrix}
\{0\} & N & N \\
N & \{0\} & N \\
N & N & \{0\}
\end{pmatrix}
\quad
\begin{array}{c|ccc}
V_1 & V_2 & V_3 \\
V_1 & \{0\} & N & N \\
V_2 & N & \{0\} & N \\
V_2 & N & N & \{0\}
\end{array}
\]
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) :=$ incidence matrix of $H$, but:
replace all 1s with $N$s and 0s with $\{0\}$s. Then:

$$(V_1, \ldots, V_{|V(H)|})$$ is an $M(H)$-partition of $G \iff$ there is a
homomorphism from $G$ to $H$.

$$M(K_3) = \left( \begin{array}{ccc}
\{0\} & N & N \\
N & \{0\} & N \\
N & N & \{0\}
\end{array} \right)$$

<table>
<thead>
<tr>
<th></th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>${0}$</td>
<td>$N$</td>
<td>$N$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$N$</td>
<td>${0}$</td>
<td>$N$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$N$</td>
<td>$N$</td>
<td>${0}$</td>
</tr>
</tbody>
</table>
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) :=$ incidence matrix of $H$, but:
replace all 1s with $\mathbb{N}s$ and 0s with $\{0\}s$. Then:

$(V_1, \ldots, V_{|V(H)|})$ is an $M(H)$-partition of $G \iff$ there is a homomorphism from $G$ to $H$.

$$M(K_3) = \begin{pmatrix}
\{0\} & \mathbb{N} & \mathbb{N} \\
\mathbb{N} & \{0\} & \mathbb{N} \\
\mathbb{N} & \mathbb{N} & \{0\}
\end{pmatrix}$$

<table>
<thead>
<tr>
<th></th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>{0}</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>{0}</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
<td>{0}</td>
</tr>
</tbody>
</table>
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997.
Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) :=$ incidence matrix of $H$, but: replace all 1s with $\mathbb{N}$s and 0s with $\{0\}$s. Then:

$(V_1, \ldots, V_{|V(H)|})$ is an $M(H)$-partition of $G$ $\iff$ there is a homomorphism from $G$ to $H$.

$$M(K_3) = \left( \begin{array}{cccc}
\{0\} & \mathbb{N} & \mathbb{N} \\
\mathbb{N} & \{0\} & \mathbb{N} \\
\mathbb{N} & \mathbb{N} & \{0\}
\end{array} \right)$$

<table>
<thead>
<tr>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>${0}$</td>
<td>$\mathbb{N}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>${0}$</td>
</tr>
<tr>
<td>$V_2$</td>
<td>$\mathbb{N}$</td>
<td>$\mathbb{N}$</td>
</tr>
</tbody>
</table>
Vertex partitioning problems

Framework introduced by Proskurowski and Telle in 1997. Let $D$ be a $q \times q$ degree constraint matrix.

Example

$H$-COLOURING $M(H) := \text{incidence matrix of } H$, but: replace all 1s with $\mathbb{N}$s and 0s with $\{0\}$s. Then:

$(V_1, \ldots, V_{|V(H)|})$ is an $M(H)$-partition of $G \iff$ there is a homomorphism from $G$ to $H$.

$$M(K_3) = \begin{pmatrix} \{0\} & \mathbb{N} & \mathbb{N} \\ \mathbb{N} & \{0\} & \mathbb{N} \\ \mathbb{N} & \mathbb{N} & \{0\} \end{pmatrix}$$

$$\begin{array}{c|ccc} & V_1 & V_2 & V_3 \\ \hline V_1 & \{0\} & \mathbb{N} & \mathbb{N} \\ V_2 & \mathbb{N} & \{0\} & \mathbb{N} \\ V_2 & \mathbb{N} & \mathbb{N} & \{0\} \end{array}$$
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

<table>
<thead>
<tr>
<th>$D$-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: $G$</td>
</tr>
<tr>
<td>Question: Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

**$D$-PARTITIONING**

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question:</td>
<td>Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

\[
\begin{array}{|c|c|}
\hline
\text{D-PARTITIONING} & \\
\hline
\text{Input:} & G \\
\text{Question:} & \text{Does } G \text{ have a } D\text{-partition?} \\
\hline
\end{array}
\]

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

<table>
<thead>
<tr>
<th>D-PARTITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
</tr>
<tr>
<td>Question:</td>
</tr>
</tbody>
</table>

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

$D$-PARTITIONING

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question:</td>
<td>Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix.

**D-PARTITIONING**

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question:</td>
<td>Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

Problems within this framework

- $k$-COLORING
- $H$-COLORING
- $H$-COVERING
- ...
Boolean width and vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix. Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

<table>
<thead>
<tr>
<th>$D$-PARTITIONING($\mathcal{C}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$</td>
</tr>
<tr>
<td><strong>Question:</strong> Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)

$D$-PARTITIONING($\mathcal{C}$) can be solved in time

$$O(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),$$

where $n = |V(G)|$, $m = |E(G)|$, and $c$ a constant.
Boolean width and vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix. Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

<table>
<thead>
<tr>
<th>$D$-PARTITIONING($\mathcal{C}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$</td>
</tr>
<tr>
<td><strong>Question:</strong> Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

**Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)**

$D$-PARTITIONING($\mathcal{C}$) can be solved in time

$$\mathcal{O}(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),$$

where $n = |V(G)|$, $m = |E(G)|$, and $c$ a constant.
Boolean width and vertex partitioning problems

Let $D$ be a $q \times q$ degree constraint matrix. Let $\mathcal{C}$ be a class of graphs of $b$-bounded Boolean width.

\[
\text{\textsc{D-Partitioning}(C)}
\]

<table>
<thead>
<tr>
<th>Input:</th>
<th>$G \in \mathcal{C}$ and a decomp. witnessing $\text{boolw}(G) \leq b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question:</td>
<td>Does $G$ have a $D$-partition?</td>
</tr>
</tbody>
</table>

\textbf{Theorem (A., Bui-Xuan, Rabinovich, Renault, Telle, Vatshelle)}

\textsc{D-Partitioning}($\mathcal{C}$) can be solved in time

\[
\mathcal{O}(n \cdot m + n \cdot b \cdot 2^{c \cdot b^2}),
\]

where $n = |V(G)|$, $m = |E(G)|$, and $c$ a constant.
Σας ευχαριστώ πολύ!