Recognizing Graphs of Rank-width at most k

Sang-il Oum

Program in Applied and Computational Mathematics
Princeton University

Dec. 17, 2004
Graph and Hypergraph Decompositions, Vienna

Partially joint work with
Paul Seymour (Princeton University) and
Bruno Courcelle (LaBRI).

Outline

Treewidth vs Clique-width

Treewidth	Clique-width
Robertson and Seymour	Courcelle and Olariu
If twd $\leq k$, every MS_2 formula is decidable in linear time	If cwd $\leq k$, every MS_1 formula is decidable in polynomial time.
H is a minor of G $\Rightarrow \text{twd}(H) \leq \text{twd}(G)$.	H is an induced s.g. of G $\Rightarrow \text{cwd}(H) \leq \text{cwd}(G)$.
Large tree-width $\Leftrightarrow$ large grid minor	?
twd $\leq k$: linear time for fixed k.	cwd $\leq k$ is open for fixed $k > 3$.

Treewidth vs Clique-width

Treewidth	Clique-width
Robertson and Seymour	Courcelle and Olariu
If twd $\leq k$, every MS_2 formula is decidable in linear time	If cwd $\leq k$, every MS_1 formula is decidable in polynomial time.
H is a minor of G $\Rightarrow \text{twd}(H) \leq \text{twd}(G)$.	H is an induced s.g. of G $\Rightarrow \text{cwd}(H) \leq \text{cwd}(G)$.
Large tree-width $\Leftrightarrow$ large grid minor	?
twd $\leq k$: linear time for fixed k.	cwd $\leq k$ is open for fixed $k > 3$.

Treewidth vs Clique-width

Treewidth	Clique-width
Robertson and Seymour	Courcelle and Olariu
If twd $\leq k$, every MS_2 formula is decidable in linear time	If cwd $\leq k$, every MS_1 formula is decidable in polynomial time.
H is a minor of G $\Rightarrow$ twd$(H) \leq$ twd(G).	H is an induced s.g. of G $\Rightarrow$ cwd$(H) \leq$ cwd(G).
Large tree-width $\Leftrightarrow$ large grid minor	?
twd $\leq k$: linear time for fixed k.	cwd $\leq k$ is open for fixed $k > 3$.

Recognizing Tree-width at most k

Approximation Algorithm

Find a tree decomposition of G of width $\leq f(k)$ or confirm that tree-width $> k$.

Decision Algorithm

Using a tree decomposition of width $\leq f(k)$, decide whether tree-width $\leq k$.

- Well-quasi-ordering theorem of graphs of bounded tree-width implies that
  $\exists G_1, G_2, \ldots, G_{h(k)}$ such that
  $\text{twd}(G) \leq k$ iff G_i is not isomorphic to any minor of G.
- For fixed graph H, we can test whether G contains an isomorphic copy of H as a minor in polynomial time if G is given by its tree decomposition.

Recognizing Tree-width at most k

Approximation Algorithm

Find a tree decomposition of G of width $\leq f(k)$ or confirm that tree-width $> k$.

Decision Algorithm

Using a tree decomposition of width $\leq f(k)$, decide whether tree-width $\leq k$.

- Well-quasi-ordering theorem of graphs of bounded tree-width implies that
  $\exists G_1, G_2, \ldots, G_{h(k)}$ such that
  $\mathrm{twd}(G) \leq k$ iff G_i is not isomorphic to any minor of G.
- For fixed graph H, we can test whether G contains an isomorphic copy of H as a minor in polynomial time if G is given by its tree decomposition.

Recognizing Tree-width at most k

Approximation Algorithm

Find a tree decomposition of G of width $\leq f(k)$ or confirm that tree-width $> k$.

Decision Algorithm

Using a tree decomposition of width $\leq f(k)$, decide whether tree-width $\leq k$.

- Well-quasi-ordering theorem of graphs of bounded tree-width implies that
  $\exists G_1, G_2, \ldots, G_{h(k)}$ such that
  $\text{twd}(G) \leq k$ iff G_i is not isomorphic to any minor of G.
- For fixed graph H, we can test whether G contains an isomorphic copy of H as a minor in polynomial time if G is given by its tree decomposition.

Clique-width is difficult?

Problems

- Is cwd $\leq k$ co-NP for fixed k?
- (Courcelle and Olariu) How different can be cwd(G) and cwd(G') if G and G' differ by exactly one edge?
- Nice characterization of graphs of cwd(G) $\leq k$ by induced subgraph relation?
- When is clique-width large?

How can we make our research easier?

Clique-width is difficult?

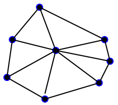
Problems

- Is cwd $\leq k$ co-NP for fixed k?
- (Courcelle and Olariu) How different can be cwd(G) and cwd(G') if G and G' differ by exactly one edge?
- Nice characterization of graphs of cwd$(G) \leq k$ by induced subgraph relation?
- When is clique-width large?

How can we make our research easier?

Definition of Rank-width
O. and Seymour

- Rank-decomposition of G: a pair (T, L)
  - T: cubic tree,
  - L: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \text{cutrk}_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of (T, L) = maximum width of e over $e \in E(T)$.
- Rank-width of G
  - Minimum width of Rank-decompositions of G.

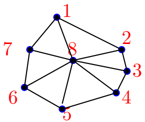


G

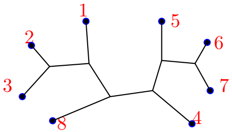
Definition of Rank-width

O. and Seymour

- Rank-decomposition of G: a pair (T, L)
  - T: cubic tree,
  - L: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \text{cutrk}_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of (T, L) = maximum width of e over $e \in E(T)$.
- Rank-width of G
  - Minimum width of Rank-decompositions of G.



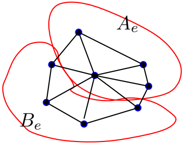
G



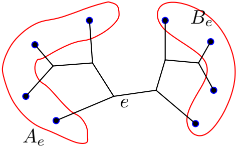
Rank-decomposition of G

Definition of Rank-width
O. and Seymour

- Rank-decomposition of G: a pair (T, L)
  - T: cubic tree,
  - L: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \text{cutrk}_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of (T, L) = maximum width of e over $e \in E(T)$.
- Rank-width of G
  - Minimum width of Rank-decompositions of G.



G



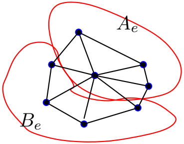
Rank-decomposition of G

$$\text{rank}\begin{pmatrix}1&0&0&0\\0&0&0&0\\1&1&1&1\\0&0&0&1\end{pmatrix}$$

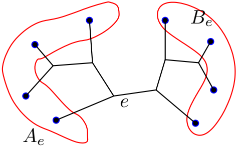
Definition of Rank-width

O. and Seymour

- Rank-decomposition of G: a pair (T, L)
  - T: cubic tree,
  - L: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \text{cutrk}_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of (T, L) = maximum width of e over $e \in E(T)$.
- Rank-width of G
  - Minimum width of Rank-decompositions of G.



G



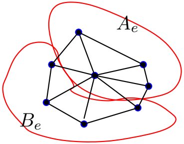
Rank-decomposition of G

$$\text{rank}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 1\end{pmatrix}$$

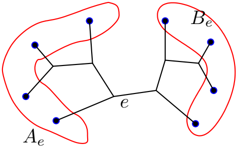
Definition of Rank-width

O. and Seymour

- Rank-decomposition of G: a pair (T, L)
  - T: cubic tree,
  - L: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \text{cutrk}_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of (T, L) = maximum width of e over $e \in E(T)$.
- Rank-width of G
  - Minimum width of Rank-decompositions of G.



G



Rank-decomposition of G

$$\text{rank}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Cut-rank

Definition of Cut-rank function cutrk : $2^{V(G)} \to \mathbb{Z}$

$\text{cutrk}_G(A) = \text{rank}(M)$,
M is a $A \times (V(G) \setminus A)$ matrix over $\mathbb{Z}_2$ such that

$$M_{xy} = \begin{cases} 1 & \text{if } xy \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

1. If M has at most k distict rows, then $\text{rank}(M) \le k$. Conversely, if $\text{rank}(M) = k$, then there are at most 2^k distinct rows.
2. Submodular inequality of a rank function

$$\text{rank}\begin{pmatrix} A & B \\ C & D \end{pmatrix} + \text{rank}(C) \ge \text{rank}\begin{pmatrix} A \\ C \end{pmatrix} + \text{rank}\begin{pmatrix} C & D \end{pmatrix}.$$

$$\Rightarrow \text{cutrk}_G(X) + \text{cutrk}_G(Y) \ge \text{cutrk}_G(X \cap Y) + \text{cutrk}_G(X \cup Y).$$

Cut-rank

Definition of Cut-rank function cutrk : $2^{V(G)} \to \mathbb{Z}$

$\text{cutrk}_G(A) = \text{rank}(M)$,
M is a $A \times (V(G) \setminus A)$ matrix over $\mathbb{Z}_2$ such that

$$M_{xy} = \begin{cases} 1 & \text{if } xy \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

1. If M has at most k distict rows, then $\text{rank}(M) \le k$. Conversely, if $\text{rank}(M) = k$, then there are at most 2^k distinct rows.
2. Submodular inequality of a rank function

$$\text{rank}\begin{pmatrix} A & B \\ C & D \end{pmatrix} + \text{rank}(C) \ge \text{rank}\begin{pmatrix} A \\ C \end{pmatrix} + \text{rank}\begin{pmatrix} C & D \end{pmatrix}.$$

$$\Rightarrow \text{cutrk}_G(X) + \text{cutrk}_G(Y) \ge \text{cutrk}_G(X \cap Y) + \text{cutrk}_G(X \cup Y).$$

Properties of Rank-width

- $\text{rwd}(G) \leq \text{cwd}(G) \leq 2^{\text{rwd}(G)+1} - 1$.
- $\text{rwd}(G) \leq 1$ iff G is distance-hereditary i.e. in every induced subgraph H and $u, v \in V(H)$, $d_H(u, v) = d_G(u, v)$.
- $\text{rwd}(G \setminus v) = \text{rwd}(G) - 1$ or $\text{rwd}(G)$.
  $\text{rwd}(G \setminus e) - \text{rwd}(G) = 0, 1$, or -1.
  $\text{rwd}(\overline{G}) - \text{rwd}(G) = 0, 1$, or -1.
- $\text{rwd}(G \oplus H) = \max(\text{rwd}(G), \text{rwd}(H))$.
- Robertson and Seymour (Graph Minors. X. '91)

Tangle Lemma

$\exists$ tangle of order $k \iff \text{rwd} \geq k$.

This can be used to show that $\text{rwd} \leq k$ is co-NP.

Properties of Rank-width

- $\mathrm{rwd}(G) \leq \mathrm{cwd}(G) \leq 2^{\mathrm{rwd}(G)+1} - 1$.
- $\mathrm{rwd}(G) \leq 1$ iff G is distance-hereditary i.e. in every induced subgraph H and $u, v \in V(H)$, $d_H(u,v) = d_G(u,v)$.
- $\mathrm{rwd}(G \setminus v) = \mathrm{rwd}(G) - 1$ or $\mathrm{rwd}(G)$.
  $\mathrm{rwd}(G \setminus e) - \mathrm{rwd}(G) = 0, 1$, or -1.
  $\mathrm{rwd}(\overline{G}) - \mathrm{rwd}(G) = 0, 1$, or -1.
- $\mathrm{rwd}(G \oplus H) = \max(\mathrm{rwd}(G), \mathrm{rwd}(H))$.
- Robertson and Seymour (Graph Minors. X. '91)

Tangle Lemma

$\exists$ tangle of order $k \iff \mathrm{rwd} \geq k$.

This can be used to show that $\mathrm{rwd} \leq k$ is co-NP.

Properties of Rank-width

- $\text{rwd}(G) \leq \text{cwd}(G) \leq 2^{\text{rwd}(G)+1} - 1$.
- $\text{rwd}(G) \leq 1$ iff G is distance-hereditary i.e. in every induced subgraph H and $u, v \in V(H)$, $d_H(u, v) = d_G(u, v)$.
- $\text{rwd}(G \setminus v) = \text{rwd}(G) - 1$ or $\text{rwd}(G)$.
  $\text{rwd}(G \setminus e) - \text{rwd}(G) = 0, 1$, or -1.
  $\text{rwd}(\overline{G}) - \text{rwd}(G) = 0, 1$, or -1.
- $\text{rwd}(G \oplus H) = \max(\text{rwd}(G), \text{rwd}(H))$.
- Robertson and Seymour (Graph Minors. X. '91)

Tangle Lemma

$\exists$ tangle of order $k \iff \text{rwd} \geq k$.

This can be used to show that $\text{rwd} \leq k$ is co-NP.

Approximating Rank-width

O. and Seymour

Our objective

For fixed k, we find an fixed-parameter-tractable algorithm that

- confirms that rank-width$> k$, or
- outputs the rank-decomposition of width $\leq 3k + 1$.

Well-Linkedness

For tree decomposition, (B. Reed)

$X \subseteq V(G)$ is well-linked if for $A, B \subseteq X$, if $|A| = |B|$, then there are $|A|$ vertex disjoint paths between A and B.

- $\exists$ well-linked set of size $k \Rightarrow \text{twd} \geq k/4 - 1$.
- $\nexists$ well-linked set of size $k \Rightarrow \text{twd} \leq k - 2$.

For rank-decomposition,

$X \subseteq V(G)$ is called well-linked,
if for every partition (A, B) of X,
$A \subseteq Z \subseteq V(G) \setminus B \quad \Rightarrow \quad \text{cutrk}_G(Z) \geq \min(|A|, |B|)$.

- $\exists$ well-linked set of size $k \Rightarrow \text{rwd} \geq k/3$.
- $\nexists$ well-linked set of size $k \Rightarrow \text{rwd} \leq k$.

Well-Linkedness

For tree decomposition, (B. Reed)

$X \subseteq V(G)$ is well-linked if for $A, B \subseteq X$, if $|A| = |B|$, then there are $|A|$ vertex disjoint paths between A and B.

- $\exists$ well-linked set of size $k \Rightarrow$ twd $\geq k/4 - 1$.
- $\nexists$ well-linked set of size $k \Rightarrow$ twd $\leq k - 2$.

For rank-decomposition,

$X \subseteq V(G)$ is called well-linked,
if for every partition (A, B) of X,
$A \subseteq Z \subseteq V(G) \setminus B \quad \Rightarrow \quad \text{cutrk}_G(Z) \geq \min(|A|, |B|)$.

- $\exists$ well-linked set of size $k \Rightarrow$ rwd $\geq k/3$.
- $\nexists$ well-linked set of size $k \Rightarrow$ rwd $\leq k$.

Well-Linkedness

For tree decomposition, (B. Reed)

$X \subseteq V(G)$ is well-linked if for $A, B \subseteq X$, if $|A| = |B|$, then there are $|A|$ vertex disjoint paths between A and B.

- $\exists$ well-linked set of size $k \Rightarrow$ twd $\geq k/4 - 1$.
- $\nexists$ well-linked set of size $k \Rightarrow$ twd $\leq k - 2$.

For rank-decomposition,

$X \subseteq V(G)$ is called well-linked,
if for every partition (A, B) of X,
$A \subseteq Z \subseteq V(G) \setminus B \quad \Rightarrow \quad \text{cutrk}_G(Z) \geq \min(|A|, |B|)$.

- $\exists$ well-linked set of size $k \Rightarrow$ rwd $\geq k/3$.
- $\nexists$ well-linked set of size $k \Rightarrow$ rwd $\leq k$.

Well-Linkedness

For tree decomposition, (B. Reed)

$X \subseteq V(G)$ is well-linked if for $A, B \subseteq X$, if $|A| = |B|$, then there are $|A|$ vertex disjoint paths between A and B.

- $\exists$ well-linked set of size $k \Rightarrow$ twd $\geq k/4 - 1$.
- $\nexists$ well-linked set of size $k \Rightarrow$ twd $\leq k - 2$.

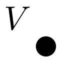
For rank-decomposition,

$X \subseteq V(G)$ is called well-linked,
if for every partition (A, B) of X,
$A \subseteq Z \subseteq V(G) \setminus B \quad \Rightarrow \quad \text{cutrk}_G(Z) \geq \min(|A|, |B|)$.

- $\exists$ well-linked set of size $k \Rightarrow$ rwd $\geq k/3$.
- $\nexists$ well-linked set of size $k \Rightarrow$ rwd $\leq k$.

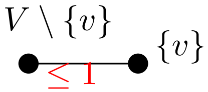
Approximation Algorithm — Trivial case

We try to grow a cubic tree T with a labeling function
$L : V(G) \to \{\text{leaves of } T\}$
to get a rank-decomposition of width $\leq k$.



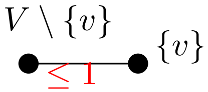
Approximation Algorithm — Trivial case

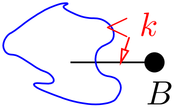
We try to grow a cubic tree T with a labeling function
$L : V(G) \to \{\text{leaves of } T\}$
to get a rank-decomposition of width $\leq k$.



Approximation Algorithm — Trivial case

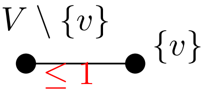
We try to grow a cubic tree T with a labeling function
$L : V(G) \to \{\text{leaves of } T\}$
to get a rank-decomposition of width $\leq k$.

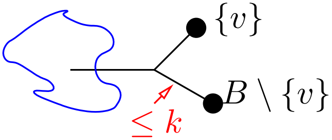




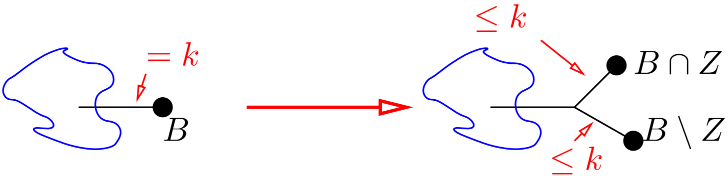
Approximation Algorithm — Trivial case

We try to grow a cubic tree T with a labeling function
$L : V(G) \to \{\text{leaves of } T\}$
to get a rank-decomposition of width $\leq k$.





Approximation Algorithm — Crutial part



Pick $X \in V \setminus B$ such that $|X| = \text{cutrk}^*(X, B) = k$. Since X is not well-linked, $\exists Z$ such that

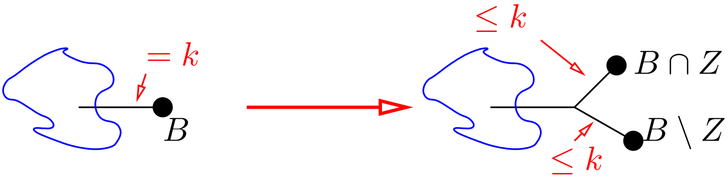
$$\text{cutrk}(Z) < \min(|Z \cap X|, |(V \setminus Z) \cap X|) \neq 0$$

Divide B into $B \cap Z$ and $B \cap (V \setminus Z)$.

$$\begin{aligned} \text{cutrk}(B) + \text{cutrk}(Z) &\geq \text{cutrk}(B \cap Z) + \text{cutrk}(B \cup Z) \\ &\geq \text{cutrk}(B \cap Z) + |(V \setminus Z) \cap X| \\ &> \text{cutrk}(B \cap Z) + \text{cutrk}(Z) \end{aligned}$$

So, $\text{cutrk}(B \cap Z) < \text{cutrk}(B) = k$.

Approximation Algorithm — Crutial part



Pick $X \in V \setminus B$ such that $|X| = \text{cutrk}^*(X, B) = k$. Since X is not well-linked, $\exists Z$ such that

$$\text{cutrk}(Z) < \min(|Z \cap X|, |(V \setminus Z) \cap X|) \neq 0$$

Divide B into $B \cap Z$ and $B \cap (V \setminus Z)$.

$$\begin{aligned}
\text{cutrk}(B) + \text{cutrk}(Z) &\geq \text{cutrk}(B \cap Z) + \text{cutrk}(B \cup Z) \\
&\geq \text{cutrk}(B \cap Z) + |(V \setminus Z) \cap X| \\
&> \text{cutrk}(B \cap Z) + \text{cutrk}(Z)
\end{aligned}$$

So, $\text{cutrk}(B \cap Z) < \text{cutrk}(B) = k$.

How to find Z?

- For each subset S of X, we need to find $Z' \subseteq V(G) \setminus X$ minimizing $\text{cutrk}(Z' \cup S)$ and look whether $\text{cutrk}(Z' \cup S) < \min(|S|, |X \setminus S|)$.
- If no such Z' is found, then $rwd \geq k/3$.
- Use "submodular function minimization" algorithms.

Iwata, Fleischer, and Fujishige '01

$O(n^5 \gamma \log M)$ time algorithm to minimize submodular functions.

- γ: time to compute the submodular function f.
- M: maximum value of f.

If $f(Z') = \text{cutrk}(Z' \cup S)$, then $O(n^8 \log n)$.

- Running time of our approximation algorithm: $O(n(n^3 + n^8 \log n)) = O(n^9 \log n)$.

How to find Z?

- For each subset S of X, we need to find $Z' \subseteq V(G) \setminus X$ minimizing $\text{cutrk}(Z' \cup S)$ and look whether $\text{cutrk}(Z' \cup S) < \min(|S|, |X \setminus S|)$.
- If no such Z' is found, then $rwd \geq k/3$.
- Use "submodular function minimization" algorithms.

Iwata, Fleischer, and Fujishige '01

$O(n^5 \gamma \log M)$ time algorithm to minimize submodular functions.

- γ: time to compute the submodular function f.
- M: maximum value of f.

If $f(Z') = \text{cutrk}(Z' \cup S)$, then $O(n^8 \log n)$.

- Running time of our approximation algorithm: $O(n(n^3 + n^8 \log n)) = O(n^9 \log n)$.

How to find Z?

- For each subset S of X, we need to find $Z' \subseteq V(G) \setminus X$ minimizing $\text{cutrk}(Z' \cup S)$ and look whether $\text{cutrk}(Z' \cup S) < \min(|S|, |X \setminus S|)$.
- If no such Z' is found, then $rwd \geq k/3$.
- Use “submodular function minimization” algorithms.

Iwata, Fleischer, and Fujishige ’01

$O(n^5 \gamma \log M)$ time algorithm to minimize submodular functions.

- γ: time to compute the submodular function f.
- M: maximum value of f.

If $f(Z') = \text{cutrk}(Z' \cup S)$, then $O(n^8 \log n)$.

- Running time of our approximation algorithm: $O(n(n^3 + n^8 \log n)) = O(n^9 \log n)$.

Well-quasi-ordering of Graphs of Bounded Treewidth

Theorem (Robertson and Seymour)

If $\{G_1, G_2, \ldots\}$ is an infinite sequence of graphs of twd $\leq k$, then there exist $i < j$ such that G_i is isomorphic to a minor of G_j.

Corollary

For each k, $\exists$ list of graphs G_1, G_2, ..., $G_{h(k)}$ such that twd$(G) \leq k$ iff G_i is not isomorphic to a minor of G for all i.

We prove a similar statement for rank-width.

Local Complementation

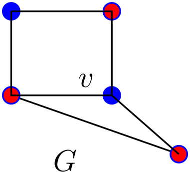
Local complementation at v

For all distinct neighbors x, y of v,
if $xy \in E(G)$, then remove the edge xy otherwise add an edge xy.

Let $G * v$ be a graph obtained by local complementation at v.

Cut-rank and Local Complementation

$\text{cutrk}_G(X) = \text{cutrk}_{G*v}(X)$.



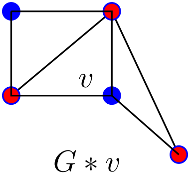
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ & A & & & & B & & \\ & C & & & & D & & \end{pmatrix}$$

Local Complementation

Local complementation at v

For all distinct neighbors x, y of v,
if $xy \in E(G)$, then remove the edge xy otherwise add an edge xy.

Let $G * v$ be a graph obtained by local complementation at v.



Cut-rank and Local Complementation

$\mathrm{cutrk}_G(X) = \mathrm{cutrk}_{G*v}(X)$.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ & A & & & & B & & \\ & C & & & & D & & \end{pmatrix}$$

Local Complementation

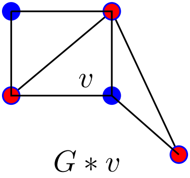
Local complementation at v

For all distinct neighbors x, y of v,
if $xy \in E(G)$, then remove the edge xy otherwise add an edge xy.

Let $G * v$ be a graph obtained by local complementation at v.

Cut-rank and Local Complementation

$\text{cutrk}_G(X) = \text{cutrk}_{G*v}(X)$.



$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ & A & & & & B & & \\ & C & & & & D & & \end{pmatrix}$$

Vertex-minor

Definition

H is a vertex-minor of G if H can be obtained from G by applying a sequence of

- vertex deletions and
- local complementations.

Then, if H is a vertex-minor of G, then

$$rwd(H) \leq \text{rwd}(G).$$

Well-quasi-ordering of Graphs of Bounded Rank-width

Theorem (O.)

If $\{G_1, G_2, \ldots\}$ is an infinite sequence of graphs of rwd $\leq k$, then there exist $i < j$ such that G_i is isomorphic to a vertex-minor of G_j.

Corollary

For each k, $\exists$ list of graphs G_1, G_2, ..., $G_{h(k)}$ such that rwd$(G) \leq k$ iff G_i is not isomorphic to a vertex-minor of G for all i.

This corollary has an elementary proof saying that $|V(G_i)| \leq (6^{k+1} - 1)/5$.

Checking a Fixed Vertex-minor

Courcelle and O.

- Let H be a fixed graph.
- We construct a C_2MS_1 formula φ_H that describes whether H is isomorphic to a vertex-minor of G.
  Main idea
  - (A. Bouchet)
    vertex-minor of graphs $\Leftrightarrow$ minor of isotropic systems.
  - Logical formulation of isotropic systems.
- By the previous corollary, we obtain a C_2MS_1 formula φ_k that decides whether $\mathrm{rwd}(G) \leq k$.
- (Courcelle) Every C_2MS_1 formula on G is decidable in polynomial time if $\mathrm{cwd}(G) \leq k$ for a fixed k.

Combining Everything

Recognizing rwd $\leq k$

Run the approximation algorithm. $O(n^9 \log n)$ time.

- If it finds a well-linked set of size $3k + 1$, then we confirm that rwd $> k$ and stop.
- Otherwise, we obtain the rank-decomposition of width at most $3k + 1$.
  - Convert it into the $(2^{3k+2} - 1)$-*expression* related to clique-width. $O(n^2)$ time.
  - Use it to test a C_2MS_1 formula describing that rwd $\leq k$. $O(n)$ time.

Combining Everything

Recognizing rwd $\leq k$

Run the approximation algorithm. $O(n^9 \log n)$ time.

- If it finds a well-linked set of size $3k + 1$, then we confirm that rwd $> k$ and stop.
- Otherwise, we obtain the rank-decomposition of width at most $3k + 1$.
  - Convert it into the $(2^{3k+2} - 1)$-*expression* related to clique-width. $O(n^2)$ time.
  - Use it to test a C_2MS_1 formula describing that rwd $\leq k$. $O(n)$ time.

Combining Everything

Recognizing rwd $\leq k$

Run the approximation algorithm. $O(n^9 \log n)$ time.

- If it finds a well-linked set of size $3k+1$, then we confirm that rwd $> k$ and stop.
- Otherwise, we obtain the rank-decomposition of width at most $3k+1$.
  - Convert it into the $(2^{3k+2}-1)$-*expression* related to clique-width. $O(n^2)$ time.
  - Use it to test a C_2MS_1 formula describing that rwd $\leq k$. $O(n)$ time.

Combining Everything

Recognizing rwd $\leq k$

Run the approximation algorithm. $O(n^9 \log n)$ time.

- If it finds a well-linked set of size $3k + 1$, then we confirm that rwd $> k$ and stop.
- Otherwise, we obtain the rank-decomposition of width at most $3k + 1$.
  - Convert it into the $(2^{3k+2} - 1)$-*expression* related to clique-width. $O(n^2)$ time.
  - Use it to test a C_2MS_1 formula describing that rwd $\leq k$. $O(n)$ time.

Combining Everything

Recognizing rwd $\leq k$

Run the approximation algorithm. $O(n^9 \log n)$ time.

- If it finds a well-linked set of size $3k + 1$, then we confirm that rwd $> k$ and stop.
- Otherwise, we obtain the rank-decomposition of width at most $3k + 1$.
  - Convert it into the $(2^{3k+2} - 1)$-*expression* related to clique-width. $O(n^2)$ time.
  - Use it to test a C_2MS_1 formula describing that rwd $\leq k$. $O(n)$ time.

Open Problems

1. For a fixed k, is it possible to output the rank-decomposition of width at most k, *if there is one*, in polynomial time?
2. Can we avoid using the general submodular minimization algorithm?
   - Let A, B be disjoint subsets of $V(G)$. Can we find a polynomial-time algorithm to find Z minimizing $\text{cutrk}_G(Z)$ such that $A \subseteq Z \subseteq V(G) \setminus B$?
     If G is bipartite, this can be done in $O(n^3)$ time. (Matroid intersection Theorem)
3. When does a graph have large rank-width (or clique-width)?
4. Is the rank-width of $n \times n$ grid $n - 1$?