Graphs of Bounded Rank-width

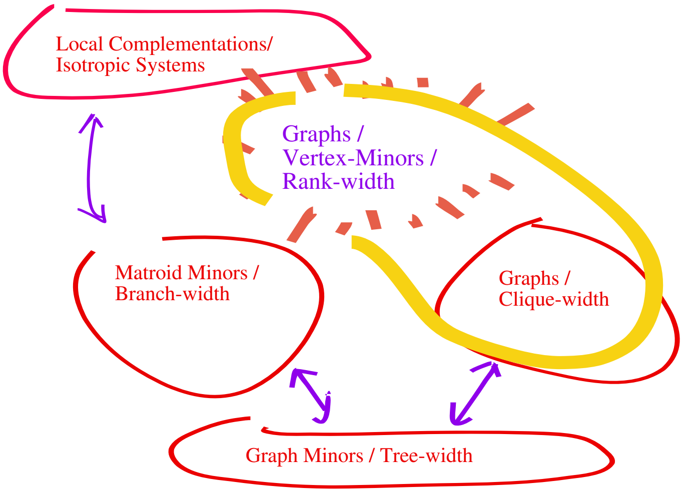
Sang-il Oum

Program in Applied and Computational Mathematics,
Princeton University
School of Mathematics,
Georgia Institute of Technology

July, 2005
Dagstuhl Seminar
Exact Algorithms and Fixed-Parameter Tractability

Partially joint work with Bruno Courcelle, Paul Seymour.

Overview of my contribution



Clique-width: Courcelle and Olariu (2000)

a complexity measure of graphs

k-expression: expression on vertex-labeled graphs with labels $\{1, 2, \cdots, k\}$ using the following 4 operations

$G_1 \oplus G_2$	disjoint union of G_1 and G_2
$\eta_{i,j}(G)$	add edges uv s.t. $lab(u) = i$, $lab(v) = j$ $(i \neq j)$
$\rho_{i \to j}(G)$	relabel all vertices of label i into label j
$\cdot_i$	create a graph with one vertex with label i

Clique-width of G, denoted by $cwd(G)$:
minimum k such that G can be expressed by k-expression (after forgetting the labels)

$$\eta_{1,2}(\rho_{1\to 2}(\eta_{1,2}(\cdot_2 \oplus \cdot_1)) \oplus \cdot_1)$$

Clique-width vs Rank-width

- The definition of clique-width is not so convenient.
- We define another complexity measure of graphs called rank-width,
- If rank-width is small, then clique-width is small and vice versa.

$$\text{Rank-width} \le \text{Clique-width} \le 2^{1+\text{Rank-width}} - 1.$$

- We use cut-rank functions to define rank-width.

cut-rank

G: graph, A: adjacency matrix of G over $\mathrm{GF}(2)$.
Let $A[X, Y]$ denote a submatrix of A with rows$=X$, columns$=Y$.

$$\rho_G(X) = \mathrm{rank}(A[X, V(G) \setminus X])$$

Clique-width vs Rank-width

- The definition of clique-width is not so convenient.
- We define another complexity measure of graphs called rank-width,
- If rank-width is small, then clique-width is small and vice versa.

$$\text{Rank-width} \le \text{Clique-width} \le 2^{1+\text{Rank-width}} - 1.$$

- We use cut-rank functions to define rank-width.

cut-rank

G: graph, A: adjacency matrix of G over $\mathrm{GF}(2)$.
Let $A[X, Y]$ denote a submatrix of A with rows$= X$, columns$= Y$.

$$\rho_G(X) = \mathrm{rank}(A[X, V(G) \setminus X])$$

Clique-width vs Rank-width

- The definition of clique-width is not so convenient.
- We define another complexity measure of graphs called rank-width,
- If rank-width is small, then clique-width is small and vice versa.

$$\text{Rank-width} \le \text{Clique-width} \le 2^{1+\text{Rank-width}} - 1.$$

- We use cut-rank functions to define rank-width.

cut-rank

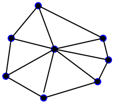
G: graph, A: adjacency matrix of G over $\mathrm{GF}(2)$.
Let $A[X, Y]$ denote a submatrix of A with rows$= X$, columns$= Y$.

$$\rho_G(X) = \text{rank}(A[X, V(G) \setminus X])$$

Rank-width

(with Seymour)

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

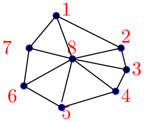


G

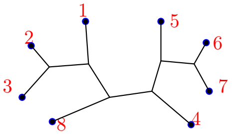
Rank-width

(with Seymour)

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



G

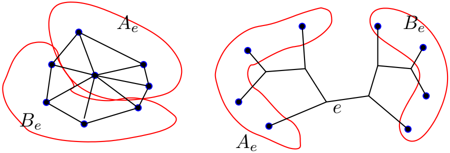


Rank-decomposition of G

Rank-width

(with Seymour)

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

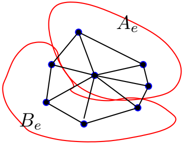


G Rank-decomposition of G

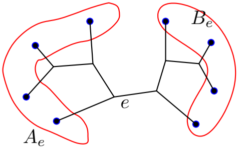
$$\text{rank}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rank-width
(with Seymour)

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



G



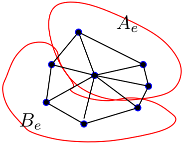
Rank-decomposition of G

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

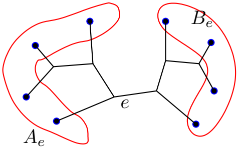
Rank-width

(with Seymour)

- Rank-decomposition of G: a pair $(T, \mathcal{L})$
  - T: subcubic tree,
  - $\mathcal{L}$: bijection from $V(G)$ to leaves of T.
- For each edge $e \in E(T)$, width of e
  - $= \rho_G(A_e)$
    where (A_e, B_e) is a partition of $V(G)$ given by $T \setminus e$.
- width of $(T, \mathcal{L})$ = 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

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

Rank-width: Why interesting?

Classical excuse: (cartoon from Garey and Johnson's book)

"I can't find an efficient algorithm for this graph problem, but neither can all these famous people."

Rank-width: Why interesting?

"How about restricting inputs?
I can find an efficient algorithm
if input graphs are restricted to graphs of small rank-width."

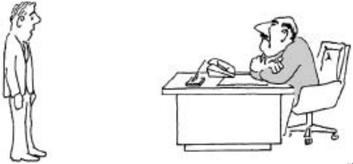
Algorithms based on Small Clique-width

- Hamiltoian path/circuit Wanke (1994), Espelage, Gurski, and Wanke (2001), Finding the chromatic number Kobler and Rotics (2003), Maximum weight stable set, ...
- All graph problems, expressible in monadic second-order logic formulas with quantifications over vertices and vertex sets. Courcelle, Makowsky, and Rotics (2000)
  (logic formulas with $\neg$, $\vee$, $\wedge$, (,), $x = y$, $x \sim y$, $x \in X$, $\forall x$, $\exists y$, $\forall X$, $\exists Y$)

Algorithms based on Small Clique-width

- Hamiltoian path/circuit Wanke (1994), Espelage, Gurski, and Wanke (2001), Finding the chromatic number Kobler and Rotics (2003), Maximum weight stable set, ...
- All graph problems, expressible in monadic second-order logic formulas with quantifications over vertices and vertex sets. Courcelle, Makowsky, and Rotics (2000)
  (logic formulas with $\neg$, $\vee$, $\wedge$, (,), $x = y$, $x \sim y$, $x \in X$, $\forall x$, $\exists y$, $\forall X$, $\exists Y$)

"Frankly, ... most of them require its k-expression as an input.
I don't know how to find it even if it is known to exist."



Problems for Clique-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of clique-width at most k?
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the k-expression of the input graph if there is one?
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the $f(k)$-expression if the input graph has clique-width at most k?

Problems for Clique-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of clique-width at most k?

- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the k-expression of the input graph if there is one?

- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the $f(k)$-expression if the input graph has clique-width at most k?

Problems for Clique-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of clique-width at most k?

- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the k-expression of the input graph if there is one?

- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the $f(k)$-expression if the input graph has clique-width at most k?

Problems for Clique-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of clique-width at most k?
  Open for $k > 3$
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the k-expression of the input graph if there is one?
  Open for $k > 3$
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the $f(k)$-expression if the input graph has clique-width at most k?
  Solved by means of rank-width

Problems for Clique-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of clique-width at most k?
  Open for $k > 3$
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the k-expression of the input graph if there is one?
  Open for $k > 3$
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the $f(k)$-expression if the input graph has clique-width at most k?
  Solved by means of rank-width

We don't know yet whether recognition of clique-width $\leq k$ is in coNP.

Problems for Rank-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of rank-width at most k?
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most k if there is one?
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most $f(k)$ if the input graph has rank-width at most k?

Problems for Rank-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of rank-width at most k?

- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most k if there is one?

- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most $f(k)$ if the input graph has rank-width at most k?

Problems for Rank-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of rank-width at most k?
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most k if there is one?
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most $f(k)$ if the input graph has rank-width at most k?

Problems for Rank-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of rank-width at most k?
  Solved
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most k if there is one?
  Open
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most $f(k)$ if the input graph has rank-width at most k?
  Solved

Problems for Rank-width

- **Recognition**
  For a fixed k, is there a polynomial-time algorithm that recognizes graphs of rank-width at most k?
  Solved
- **Construction**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most k if there is one?
  Open
- **Approximation**
  For a fixed k, is there a polynomial-time algorithm that outputs the rank-decomposition of width at most $f(k)$ if the input graph has rank-width at most k?
  Solved

It is in coNP to recognize graphs of rank-width $\leq k$ (proved with Seymour).

Seese's conjecture

$\mathcal{C}$: set of graphs
If there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for an input φ (monadic second-order logic formula, MS logic formula),
then $\mathcal{C}$ has bounded rank-width.

We prove a slight weakening (with Courcelle).

If φ is allowed to use Even(X) (true if $|X|$ is even), (modulo-2 counting monadic second-order logic formula, C_2MS logic formula),
then there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ only if $\mathcal{C}$ has bounded rank-width.

Seese's conjecture

$\mathcal{C}$: set of graphs
If there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for an input φ (monadic second-order logic formula, MS logic formula),
then $\mathcal{C}$ has bounded rank-width.

We prove a slight weakening (with Courcelle).

If φ is allowed to use Even(X) (true if $|X|$ is even),
(modulo-2 counting monadic second-order logic formula, C_2MS logic formula),
then there is an algorithm that answers
whether all graphs in $\mathcal{C}$ satisfy φ only if $\mathcal{C}$ has bounded rank-width.

Vertex-minors

We introduce vertex-minors for rank-width.

- Robertson and Seymour
  If H is a minor of G, then the tree-width of H is at most the tree-width of G.
- Courcelle and Olariu
  If H is an induced subgraph of G, then the clique-width of H is at most the clique-width of G.
- If H is a vertex-minor of G, then the rank-width of H is at most the rank-width of G.

Binary Matroids

$M = (E, I)$ is a matroid if

- E is a finite set,
- I is a collection of independent subsets of E, satisfying
  - (I1) $\emptyset \in I$
  - (I2) If $A \in I$ and $B \subseteq A$, then $B \in I$.
  - (I3) For every $Z \subseteq E$, the maximal subsets of Z in I have the same size $= r(Z)$, the rank of Z.

We call M a binary matroid
if $\exists$ matrix N over $\mathrm{GF}(2)$ such that

- E: set of column vectors of N,
- $I = \{X \subseteq E : X \text{ is independent as vectors}\}$.

Matroid connectivity: $\lambda(X) = r(X) + r(E \setminus X) - r(E) + 1$.

Standard Representation of Binary Matroids

- Elementary row operations don't change the binary matroid.
- We may delete dependent rows.

Therefore, WMA binary matroids are represented by matrices of the following form $\left(\begin{array}{c|c} \text{identity matrix} & B \end{array}\right)$.

Fundamental graph: A bipartite graph whose adjacency matrix is $\begin{pmatrix} 0 & B \\ B^t & 0 \end{pmatrix}$.

Pivoting: By elementary row operations, we may swap one column in the identity matrix with another column in B.

Matroid minor operations on standard representation

- Pivoting.
- Delete a column not in the identity matrix.
- Delete a row.

Standard Representation of Binary Matroids

- Elementary row operations don't change the binary matroid.
- We may delete dependent rows.

Therefore, WMA binary matroids are represented by matrices of the following form $\left(\begin{array}{c|c} \text{identity matrix} & B \end{array} \right)$.

Fundamental graph: A bipartite graph whose adjacency matrix is $\begin{pmatrix} 0 & B \\ B^t & 0 \end{pmatrix}$.

Pivoting: By elementary row operations, we may swap one column in the identity matrix with another column in B.

Matroid minor operations on standard representation

- Pivoting.
- Delete a column not in the identity matrix.
- Delete a row.

Bipartite Graphs and Binary Matroids

Connectivity of Binary Matroids
= Cut-rank of its Fundamental (Bipartite) Graph +1

Branch-width of binary matroids can be defined as rank-width of their fundamental graphs $+1$.

Observation

Pivoting on binary matroids changes fundamental graphs, but it preserves their cut-rank functions.
$\Rightarrow$ Pivoting on bipartite graphs preserves cut-rank functions.

Can we find a graph operation that preserves the cut-rank functions (of non-bipartite graphs)?

Bipartite Graphs and Binary Matroids

Connectivity of Binary Matroids
= Cut-rank of its Fundamental (Bipartite) Graph +1

Branch-width of binary matroids can be defined as rank-width of their fundamental graphs $+1$.

Observation

Pivoting on binary matroids changes fundamental graphs, but it preserves their cut-rank functions.
$\Rightarrow$ Pivoting on bipartite graphs preserves cut-rank functions.

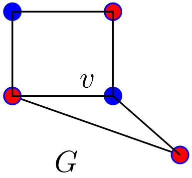
Can we find a graph operation that preserves the cut-rank functions (of non-bipartite graphs)?

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.

$G * v$: graph obtained by local complementation at v.

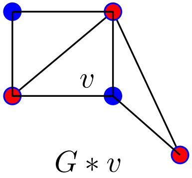


Local Complementation

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.

$G * v$: graph obtained by local complementation at v.

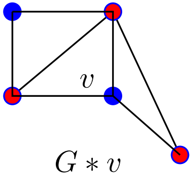


Local Complementation

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.

$G * v$: graph obtained by local complementation at v.



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

$\rho_G(X) = \rho_{G*v}(X)$.

Moreover, a pivoting on bipartite graph is equal to $G * v * w * v$.

Vertex-minors

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.

If H is a vertex-minor of G, then $\text{rwd}(H) \leq \text{rwd}(G)$.
Why?

Relation to Minors of Binary Matroids?

Vertex-minors

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.

If H is a vertex-minor of G, then $\text{rwd}(H) \le \text{rwd}(G)$.
Why?

$\text{rwd}(G \setminus v) \le \text{rwd}(G)$, $\text{rwd}(G * v) = \text{rwd}(G)$.

Relation to Minors of Binary Matroids?

Vertex-minors

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.

If H is a vertex-minor of G, then $\text{rwd}(H) \le \text{rwd}(G)$.
Why?

$$\text{rwd}(G \setminus v) \le \text{rwd}(G), \text{rwd}(G * v) = \text{rwd}(G).$$

Relation to Minors of Binary Matroids?

If N is a minor of M, then
every fundamental graph of N is a vertex-minor of a fundamental graph of M.

Recognition Algorithm for Rank-width

We combine three main theorems to obtain the recognition algorithm for rank-width (when k is fixed).

Approximating Rank-width

Poly-time algorithm that either outputs a rank-decomp. of width $\leq f(k)$ or confirms that rank-width $> k$.

Excluded Vertex-minors

$\exists$ a finite list of graphs such that no one in the list is a vertex-minor of G iff G has rank-width $\leq k$.

Testing Vertex-minors

Linear-time algorithm that tests whether a fixed graph is a vertex-minor of G when rank-decomp. of width $\leq f(k)$ is given.

Approximating Rank-width: 3 Algorithms

Approximating Rank-width

Poly-time algorithm that either outputs a rank-decomp. of width $\le f(k)$
or confirms that rank-width $> k$.

1. (with Seymour) $f(k) = 3k + 1$, $O(n^9 \log n)$
   Uses the submodular function minimization alg.
2. $f(k) = 3k + 1$, $O(n^4)$
   We find a faster minimization algorithm for cut-rank functions.
3. $f(k) = 24k$, $O(n^3)$
   Reduce the problem to branch-width of binary matroids.
   Then use Hliněný's algorithm.

Approximating Rank-width: 3 Algorithms

Approximating Rank-width

Poly-time algorithm that either outputs a rank-decomp. of width $\leq f(k)$
or confirms that rank-width $> k$.

1. (with Seymour) $f(k) = 3k + 1$, $O(n^9 \log n)$
   Uses the submodular function minimization alg.
2. $f(k) = 3k + 1$, $O(n^4)$
   We find a faster minimization algorithm for cut-rank functions.
3. $f(k) = 24k$, $O(n^3)$
   Reduce the problem to branch-width of binary matroids. Then use Hliněný's algorithm.

Approximating Rank-width: 3 Algorithms

Approximating Rank-width

Poly-time algorithm that either outputs a rank-decomp. of width $\leq f(k)$
or confirms that rank-width $> k$.

1. (with Seymour) $f(k) = 3k + 1$, $O(n^9 \log n)$
   Uses the submodular function minimization alg.
2. $f(k) = 3k + 1$, $O(n^4)$
   We find a faster minimization algorithm for cut-rank functions.
3. $f(k) = 24k$, $O(n^3)$
   Reduce the problem to branch-width of binary matroids.
   Then use Hliněný's algorithm.

Reduction to Bipartite Graphs

Graph $G = (V, E) \implies$ Bipartite graph $B(G)$ (Courcelle (2004)

- $(v, 1)$, $(v, 2)$, $(v, 3)$, $(v, 4)$ are vertices of $B(G)$ corresponding to $v \in V$.
- $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$ iff $vw \in E$.
- $(v, 1)(v, 2)(v, 3)(v, 4)$ is a 3-edge path for each v.

G

$B(G)$

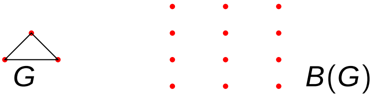
Theorem

If $E(G) \neq \emptyset$, then rank-width of $B(G) = 2$ rank-width of G.

Reduction to Bipartite Graphs

Graph $G = (V, E) \Longrightarrow$ Bipartite graph $B(G)$ (Courcelle (2004)

- $(v, 1)$, $(v, 2)$, $(v, 3)$, $(v, 4)$ are vertices of $B(G)$ corresponding to $v \in V$.
- $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$ iff $vw \in E$.
- $(v, 1)(v, 2)(v, 3)(v, 4)$ is a 3-edge path for each v.



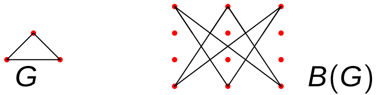
Theorem

If $E(G) \neq \emptyset$, then
rank-width of $B(G) = 2$ rank-width of G.

Reduction to Bipartite Graphs

Graph $G = (V, E) \implies$ Bipartite graph $B(G)$ (Courcelle (2004))

- $(v, 1)$, $(v, 2)$, $(v, 3)$, $(v, 4)$ are vertices of $B(G)$ corresponding to $v \in V$.
- $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$ iff $vw \in E$.
- $(v, 1)(v, 2)(v, 3)(v, 4)$ is a 3-edge path for each v.



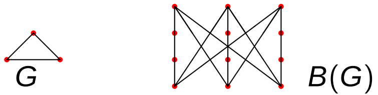
Theorem

If $E(G) \neq \emptyset$, then
rank-width of $B(G) = 2$ rank-width of G.

Reduction to Bipartite Graphs

Graph $G = (V, E) \Longrightarrow$ Bipartite graph $B(G)$ (Courcelle (2004))

- $(v, 1)$, $(v, 2)$, $(v, 3)$, $(v, 4)$ are vertices of $B(G)$ corresponding to $v \in V$.
- $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$ iff $vw \in E$.
- $(v, 1)(v, 2)(v, 3)(v, 4)$ is a 3-edge path for each v.



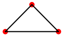
Theorem

If $E(G) \neq \emptyset$, then rank-width of $B(G) = 2$ rank-width of G.

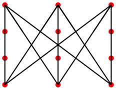
Reduction to Bipartite Graphs

Graph $G = (V, E) \Longrightarrow$ Bipartite graph $B(G)$ (Courcelle (2004))

- $(v, 1)$, $(v, 2)$, $(v, 3)$, $(v, 4)$ are vertices of $B(G)$ corresponding to $v \in V$.
- $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$ iff $vw \in E$.
- $(v, 1)(v, 2)(v, 3)(v, 4)$ is a 3-edge path for each v.



G



$B(G)$

Theorem

If $E(G) \neq \emptyset$, then
rank-width of $B(G) = 2$ rank-width of G.

Implication to Algorithms based on Small Clique-width

1. We have a $O(n^2)$-time algorithm transforming a rank-decomposition of width k into the $(2^{k+1}-1)$-expression.

   width $24k \to$ $(2 \cdot 16777216^k - 1)$-expession.
2. This can be given as an input to algorithms based on small clique-width.
3. Since k is constant, we no longer require k-expressions as an input to obtain the polynomial-time algorithms.

Approximating Rank-width

$O(n^3)$-time algorithm that either outputs a rank-decomp. of width $\leq 24k$ or confirms that rank-width $> k$.

$n = |V(G)|$.

Implication to Algorithms based on Small Clique-width

1. We have a $O(n^2)$-time algorithm transforming a rank-decomposition of width k into the $(2^{k+1}-1)$-expression.

   width $24k \to$ $(2 \cdot 16777216^k - 1)$-expession.
2. This can be given as an input to algorithms based on small clique-width.
3. Since k is constant, we no longer require k-expressions as an input to obtain the polynomial-time algorithms.

Approximating Rank-width

$O(n^3)$-time algorithm that either outputs a rank-decomp. of width $\leq 24k$ or confirms that rank-width $> k$.

$n = |V(G)|$.

Implication to Algorithms based on Small Clique-width

1. We have a $O(n^2)$-time algorithm transforming a rank-decomposition of width k into the $(2^{k+1}-1)$-expression.

   width $24k \to$ $(2 \cdot 16777216^k - 1)$-expession.
2. This can be given as an input to algorithms based on small clique-width.
3. Since k is constant, we no longer require k-expressions as an input to obtain the polynomial-time algorithms.

Approximating Rank-width

$O(n^3)$-time algorithm that either outputs a rank-decomp. of width $\leq 24k$ or confirms that rank-width $> k$.

$n = |V(G)|$.

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

Theorem

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.

The proof uses isotropic systems by Bouchet (80's).

Corollary

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

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

Testing Vertex-minors
(with Courcelle)

Testing Vertex-minors

Linear-time algorithm that tests whether a fixed graph is a vertex-minor of G
when rank-decomp. of width $\leq f(k)$ is given.

- Let H be a fixed graph.
- We construct a C_2MS 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 formula φ_k that decides whether $\mathrm{rwd}(G) \leq k$.
- (Courcelle) Every C_2MS formula on G is decidable in polynomial time if $\mathrm{cwd}(G) \leq k$ for a fixed k.

Recognition Algorithm (Summary)

For each k, we obtain a $O(n^3)$-time algorithm by combining three theorems.

Approximating Rank-width

Poly-time algorithm that either outputs a rank-decomp. of width $\leq f(k)$ or confirms that rank-width $> k$.

Excluded Vertex-minors

$\exists$ a finite list of graphs such that no one in the list is a vertex-minor of G iff G has rank-width $\leq k$.

Testing Vertex-minors

Linear-time algorithm that tests whether a fixed graph is a vertex-minor of G when rank-decomp. of width $\leq f(k)$ is given.

Seese's conjecture
Seese (1991)

The MS_2 logic formulas (monadic second-order logic formulas of second kind) allow quantifications over edges and edge-sets in addition to MS logic formulas.

Seese's MS_2 theorem

If there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for every MS_2 logic formula φ given as an input, then $\mathcal{C}$ has bounded tree-width.

- Robertson and Seymour's grid theorem, explaining the structure of graphs having large tree-width.

Seese's conjecture

If there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for every MS logic formula φ given as an input, then $\mathcal{C}$ has bounded rank-width.

Weakening of Seese's conjecture

We say that $\mathcal{C}$ has a decidable C_2MS theory if there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for every C_2MS logic formula φ given as an input.

We prove the following (with Courcelle).

If $\mathcal{C}$ has a decidable C_2MS theory, then $\mathcal{C}$ has bounded rank-width.

We need the following.

- Structure theorem for graphs having large rank-width.
- Logic formulas describing this structure.

Weakening of Seese's conjecture

We say that $\mathcal{C}$ has a decidable C_2MS theory if there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for every C_2MS logic formula φ given as an input.

We prove the following (with Courcelle).

If $\mathcal{C}$ has a decidable C_2MS theory, then $\mathcal{C}$ has bounded rank-width.

We need the following.

- Structure theorem for graphs having large rank-width.
- Logic formulas describing this structure.

Weakening of Seese's conjecture

We say that $\mathcal{C}$ has a decidable C_2MS theory if there is an algorithm that answers whether all graphs in $\mathcal{C}$ satisfy φ for every C_2MS logic formula φ given as an input.

We prove the following (with Courcelle).

If $\mathcal{C}$ has a decidable C_2MS theory, then $\mathcal{C}$ has bounded rank-width.

We need the following.

- Structure theorem for graphs having large rank-width.
- Logic formulas describing this structure.

Enough to consider Bipartite Graphs

Courcelle (2004)

graphs G $\Longrightarrow$ bipartite graphs $B(G)$

Enough to consider Bipartite Graphs
Courcelle (2004)

C_2MS logic formula $\varphi^{\#}$ on $G \in \mathcal{C}$	$\Longleftarrow$	C_2MS logic formulas φ on $B(G) \in B(\mathcal{C})$

Enough to consider Bipartite Graphs

Courcelle (2004)

C$_2$MS logic formula $\varphi^{\#}$ on $G \in \mathcal{C}$	$\Longleftarrow$	C$_2$MS logic formulas φ on $B(G) \in B(\mathcal{C})$

It is enough to consider bipartite graphs. Why?

1. Suppose that $\mathcal{C}$ has a decidable C$_2$MS theory. There is an algorithm A on $\mathcal{C}$ that answers whether an input C$_2$MS logic formula is satisfied by all graphs in $\mathcal{C}$.
2. If a C$_2$MS logic formula φ on $B(\mathcal{C})$ is given, we can answer whether all graphs in $\mathcal{B}(\mathcal{C})$ satisfy φ by using $\varphi^{\#}$ on $\mathcal{C}$.
3. So $B(\mathcal{C})$ has a decidable C$_2$MS theory.
4. Moreover, if $B(\mathcal{C})$ has bounded rank-width, then so does $\mathcal{C}$.

Enough to consider Bipartite Graphs
Courcelle (2004)

C_2MS logic formula $\varphi^{\#}$ on $G \in \mathcal{C}$ $\Longleftarrow$ C_2MS logic formulas φ on $B(G) \in B(\mathcal{C})$

How to construct $\varphi^{\#}$?

1. Replace $\forall Y(\mu(Y))$ with $\forall Y_1, Y_2, Y_3, Y_4 \mu'(Y_1, Y_2, Y_3, Y_4)$.
2. Replace $\forall y \mu(y)$ with $\forall y_1 \mu_{y_1}(y_1) \wedge \forall y_2 \mu_{y_2}(y_2) \wedge \forall y_3 \mu_{y_3}(y_3) \wedge \forall y_4 \mu_{y_4}(y_4)$.
3. Replace $xy \in E(B(G))$ with one of the following depending on the context:
   $x_1 = y_2, x_2 = y_3, x_3 = y_4, x_1 y_4 \in E(G)$,
   $y_1 = x_2, y_2 = x_3, y_3 = x_4, y_1 x_4 \in E(G)$.
4. Replace $x \in Y$ with one of the following depending on the context:
   $x_1 \in Y_1, x_2 \in Y_2, x_3 \in Y_3, x_4 \in Y_4$.
5. etc.

Enough to consider Bipartite Graphs

Courcelle (2004)

graphs G $\Longrightarrow$ bipartite graphs $B(G)$

This kind of mapping is called a C_2MS transduction. (Courcelle) It describes $B(G)$ from G by C_2MS logic formulas.

C_2MS logic formula $\varphi^{\#}$ on $G \in \mathcal{C}$ $\Longleftarrow$ C_2MS logic formulas φ on $B(G) \in B(\mathcal{C})$

This is called a backward translation of the above C_2MS transduction.

In general, if $\mathcal{C}$ has a decidable C_2MS theory and μ is a C_2MS transduction, then $\mu(\mathcal{C})$ also has a decidable C_2MS theory.

Bipartite graphs with Large Rank-width

Geelen, Gerards, Whittle (2003)

For each k, there exists $N(k)$ such that if a binary matroid M does not contain the cycle matroid of $k \times k$ grid as a minor, then the branch-width of $M \leq N(k)$.

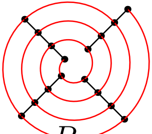
For each k, there exists $N(k)$ such that if a bipartite graph G does not contain the fundamental graph of the cycle matroid of $k \times k$ grid as a vertex-minor, then the rank-width of $G \leq N(k)$.

Bipartite graphs with Large Rank-width

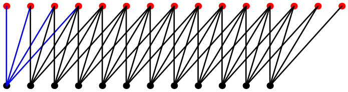
Geelen, Gerards, Whittle (2003)

For each k, there exists $N(k)$ such that if a binary matroid M does not contain the cycle matroid of $k \times k$ grid as a minor, then the branch-width of $M \leq N(k)$.

For each k, there exists $N(k)$ such that if a bipartite graph G does not contain S_k as a vertex-minor, then the rank-width of $G \leq N(k)$.



R_4



S_4

C_2MS transduction to Grids

(With Courcelle)

There exist C_2MS transductions

$$\tau_1 : \text{graphs } G \quad \Longrightarrow \quad \text{all vertex-minors of } G,$$

and

$$\tau_2 : \text{graphs } S_k \quad \Longrightarrow \quad k \times (2k-2) \text{ grids for all } k.$$

We combine with this.

Seese (1991)

If $\mathcal{C}$ is a set of graphs such that every planar graph is a minor of a planar graph contained in $\mathcal{C}$, then $\mathcal{C}$ does not have a decidable MS(or C_2MS) theory.

Proof sketch

(With Courcelle)

1. Let $\mathcal{C}$ be a set of bipartite graphs having a decidable C_2MS theory.
   Suppose that $\mathcal{C}$ has unbounded rank-width.
2. By Geelen et al., $\tau_1(\mathcal{C})$ contains S_k for infinitely many k.
3. $\tau_2(\tau_1(\mathcal{C}))$ contains $k \times (2k-2)$ grids for infinitely many k.
4. Since τ_1 and τ_2 are C_2MS transductions, $\tau_2(\tau_1(\mathcal{C}))$ also has a decidable C_2MS theory.
   A contradiction to Seese's theorem!

Proof sketch
(With Courcelle)

1. Let $\mathcal{C}$ be a set of bipartite graphs having a decidable C_2MS theory.
   Suppose that $\mathcal{C}$ has unbounded rank-width.
2. By Geelen et al., $\tau_1(\mathcal{C})$ contains S_k for infinitely many k.
3. $\tau_2(\tau_1(\mathcal{C}))$ contains $k \times (2k-2)$ grids for infinitely many k.
4. Since τ_1 and τ_2 are C_2MS transductions, $\tau_2(\tau_1(\mathcal{C}))$ also has a decidable C_2MS theory.
   A contradiction to Seese's theorem!

Proof sketch
(With Courcelle)

1. Let $\mathcal{C}$ be a set of bipartite graphs having a decidable C_2MS theory.
   Suppose that $\mathcal{C}$ has unbounded rank-width.
2. By Geelen et al., $\tau_1(\mathcal{C})$ contains S_k for infinitely many k.
3. $\tau_2(\tau_1(\mathcal{C}))$ contains $k \times (2k-2)$ grids for infinitely many k.
4. Since τ_1 and τ_2 are C_2MS transductions, $\tau_2(\tau_1(\mathcal{C}))$ also has a decidable C_2MS theory.
   A contradiction to Seese's theorem!

Proof sketch

(With Courcelle)

1. Let $\mathcal{C}$ be a set of bipartite graphs having a decidable C_2MS theory.
   Suppose that $\mathcal{C}$ has unbounded rank-width.
2. By Geelen et al., $\tau_1(\mathcal{C})$ contains S_k for infinitely many k.
3. $\tau_2(\tau_1(\mathcal{C}))$ contains $k \times (2k-2)$ grids for infinitely many k.
4. Since τ_1 and τ_2 are C_2MS transductions, $\tau_2(\tau_1(\mathcal{C}))$ also has a decidable C_2MS theory.
   A contradiction to Seese's theorem!

Summary

1. Approximation algorithm. (with Seymour)
   We no longer require k-expressions to algorithms on small clique-width.
2. Recognition algorithm.
   For fixed k, we can recognize n-vertex graphs of rank-width $\leq k$ in $O(n^3)$ time.
3. Structure theorems.
   Vertex-minors and well-quasi-ordering. Relation to binary matroids and isotropic systems.
4. Characterize a set of graphs having a decidable C_2MS theory. (with Courcelle)
   Weakened Seese's conjecture.

Summary

1. Approximation algorithm. (with Seymour)
   We no longer require k-expressions to algorithms on small clique-width.
2. Recognition algorithm.
   For fixed k, we can recognize n-vertex graphs of rank-width $\leq k$ in $O(n^3)$ time.
3. Structure theorems.
   Vertex-minors and well-quasi-ordering. Relation to binary matroids and isotropic systems.
4. Characterize a set of graphs having a decidable C_2MS theory. (with Courcelle)
   Weakened Seese's conjecture.

Summary

1. Approximation algorithm. (with Seymour)
   We no longer require k-expressions to algorithms on small clique-width.
2. Recognition algorithm.
   For fixed k, we can recognize n-vertex graphs of rank-width $\leq k$ in $O(n^3)$ time.
3. Structure theorems.
   Vertex-minors and well-quasi-ordering. Relation to binary matroids and isotropic systems.
4. Characterize a set of graphs having a decidable C_2MS theory. (with Courcelle)
   Weakened Seese's conjecture.

Summary

1. Approximation algorithm. (with Seymour)
   We no longer require k-expressions to algorithms on small clique-width.
2. Recognition algorithm.
   For fixed k, we can recognize n-vertex graphs of rank-width $\leq k$ in $O(n^3)$ time.
3. Structure theorems.
   Vertex-minors and well-quasi-ordering. Relation to binary matroids and isotropic systems.
4. Characterize a set of graphs having a decidable C_2MS theory. (with Courcelle)
   Weakened Seese's conjecture.

Finally

Further research

1. Generalize the grid theorem to general graphs.
2. Develop a direct algorithm to test vertex-minors.
3. Construction problem.

Thank you for your attention!