Finding Branch-decompositions & Rank-decompositions

Sang-il Oum

Dept. of Combinatorics and Optimization
Faculty of Mathematics,
Univ. of Waterloo.

July, 2007

Joint work with Petr Hliněný

Dagstuhl workshop 2007.

A function $f : 2^V \to \mathbb{Z}$ is a connectivity function if

(i) $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, (submodular)

(ii) $f(X) = f(V \setminus X)$, (symmetric)

(iii) $f(\emptyset) = 0$.

$v(X)$ =number of vertices meeting both X and $E \setminus X$.

$e(X)$ =number of edges meeting both X and $V \setminus X$.

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.

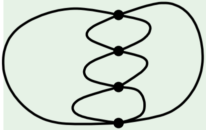
For a graph G, let A = adj matrix. $\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.

A function $f : 2^V \to \mathbb{Z}$ is a connectivity function if

(i) $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, (submodular)

(ii) $f(X) = f(V \setminus X)$, (symmetric)

(iii) $f(\emptyset) = 0$.



$v(X)$ =number of vertices meeting both X and $E \setminus X$.

$e(X)$ =number of edges meeting both X and $V \setminus X$.

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.

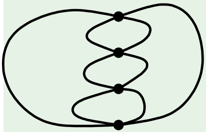
For a graph G, let A = adj matrix. $\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.

A function $f : 2^V \to \mathbb{Z}$ is a connectivity function if

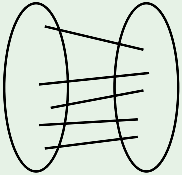
(i) $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, (submodular)

(ii) $f(X) = f(V \setminus X)$, (symmetric)

(iii) $f(\emptyset) = 0$.



$v(X)$ =number of vertices meeting both X and $E \setminus X$.



$e(X)$ =number of edges meeting both X and $V \setminus X$.

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.

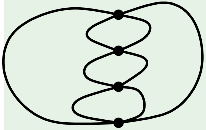
For a graph G, let A = adj matrix. $\rho_G(X) = \text{rank } A[X, V \setminus X]$.

A function $f : 2^V \to \mathbb{Z}$ is a connectivity function if

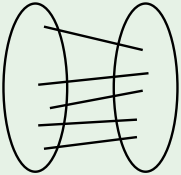
(i) $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, (submodular)

(ii) $f(X) = f(V \setminus X)$, (symmetric)

(iii) $f(\emptyset) = 0$.



$v(X)$ =number of vertices meeting both X and $E \setminus X$.



$e(X)$ =number of edges meeting both X and $V \setminus X$.

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.

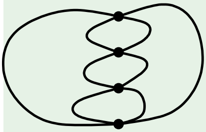
For a graph G, let A = adj matrix. $\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.

A function $f : 2^V \to \mathbb{Z}$ is a connectivity function if

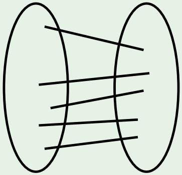
(i) $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, (submodular)

(ii) $f(X) = f(V \setminus X)$, (symmetric)

(iii) $f(\emptyset) = 0$.



$v(X)$ =number of vertices meeting both X and $E \setminus X$.

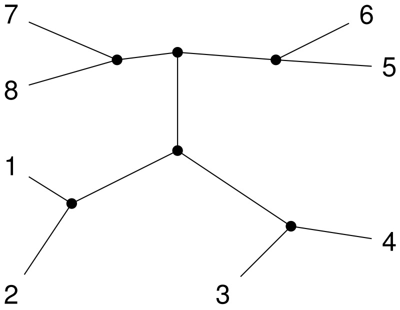


$e(X)$ =number of edges meeting both X and $V \setminus X$.

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.

For a graph G, let $A =$ adj matrix. $\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.

Branch-decomposition of f: a pair (T, L) of
a *subcubic tree* T and a *bijection* $L : V \to \{\text{leaves of T}\}$.



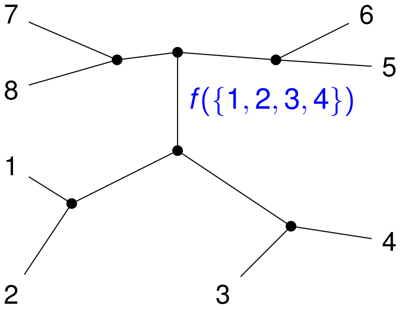
Branch-width

Carving-width

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let A = adj matrix.
$\rho_G(X) = \text{rank } A[X, V \setminus X]$.
Rank-width of graphs

Branch-decomposition of f: a pair (T, L) of
a *subcubic tree* T and a *bijection* $L : V \to \{\text{leaves of T}\}$.



Width of an edge e of T: $f(A_e)$
(A_e, B_e) is a partition of V given by deleting e.

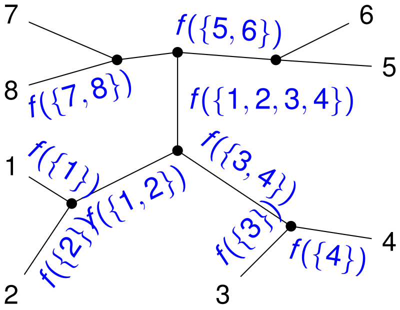
Branch-width

Carving-width

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let A = adj matrix.
$\rho_G(X) = \text{rank } A[X, V \setminus X]$.
Rank-width of graphs

Branch-decomposition of f: a pair (T, L) of
a *subcubic tree* T and a *bijection* $L : V \to \{\text{leaves of T}\}$.



Width of an edge e of T: $f(A_e)$
(A_e, B_e) is a partition of V given by deleting e.

Width of (T, L): $\max_e \text{width}(e)$

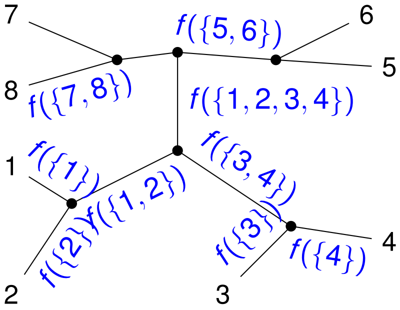
Branch-width

Carving-width

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let A = adj matrix.
$\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.
Rank-width of graphs

Branch-decomposition of f: a pair (T, L) of a *subcubic tree* T and a *bijection* $L : V \to \{\text{leaves of T}\}$.



Width of an edge e of T: $f(A_e)$
(A_e, B_e) is a partition of V given by deleting e.

Width of (T, L): $\max_e \text{width}(e)$

Branch-width: $min_{(T,L)}\, \text{width}(T, L)$.
(If $|V| \leq 1$, then branch-width=0)

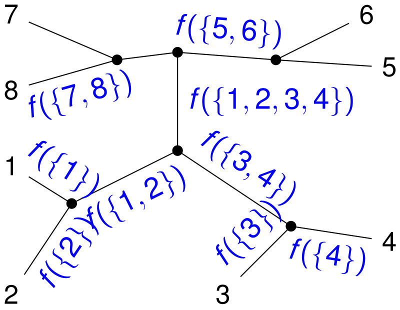
Branch-width

Carving-width

$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let A = adj matrix.
$\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.
Rank-width of graphs

Branch-decomposition of f: a pair (T, L) of a *subcubic tree* T and a *bijection* $L : V \to \{\text{leaves of T}\}$.

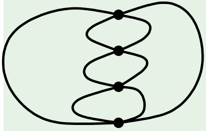


Width of an edge e of T: $f(A_e)$
(A_e, B_e) is a partition of V given by deleting e.

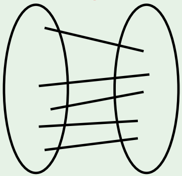
Width of (T, L): $\max_e \text{width}(e)$

Branch-width: $min_{(T,L)}$ width(T, L).
(If $|V| \leq 1$, then branch-width=0)

Branch-width

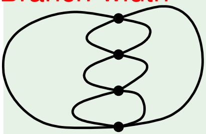


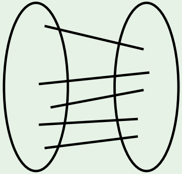
Carving-width



$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let $A =$ adj matrix.
$\rho_G(X) = \text{rank}\, A[X, V \setminus X]$.
Rank-width of graphs





$\mathcal{M}$: matroid, $\lambda(X) = r(X) + r(E(\mathcal{M}) - X) - r(E(\mathcal{M}))$.
Branch-width of matroids.

For a graph G, let $A =$ adj matrix.
$\rho_G(X) = \operatorname{rank} A[X, V \setminus X]$.
Rank-width of graphs

Testing Branch-width $\leq k$ for fixed k

- Branch-width of graphs: Linear (Bodlaender, Thilikos '97)
- Carving-width of graphs: Linear (Thilikos, Serna, Bodlaendar '00)
- Branch-width of matroids represented over a fixed finite field: $O(|E(\mathcal{M})|^3)$ (Hliněný '05)
- Rank-width of graphs: $O(|V(G)|^3)$ (Oum '05)
- Any connectivity function: $O(\gamma n^{8k+6} \log n)$ (Oum and Seymour '07)

Constructing Branch-decomposition of width $\leq k$

Suppose that branch-width $\leq k$ (for a connectivity function).

How can we construct a branch-decomposition of width $\leq k$?

Jim Geelen (2005, in OS'07)

- We can test branch-width of connectivity functions induced by partitions of V (by treating each part as one element).
- Recursively find a pair $a, b \in V$ such that merging them does not increase branch-width. Merge them in one part.

We can construct, in time $O(\gamma n^{8k+9} \log n)$,

- rank-decomposition of width $\leq k$ (if rwd $\leq k$)
- branch-decomposition of width$\leq k$ (if bwd $\leq k$) for matroids.

Constructing Branch-decomposition of width $\leq k$

Suppose that branch-width $\leq k$ (for a connectivity function).

How can we construct a branch-decomposition of width $\leq k$?

Jim Geelen (2005, in OS'07)

- We can test branch-width of connectivity functions induced by partitions of V (by treating each part as one element).
- Recursively find a pair $a, b \in V$ such that merging them does not increase branch-width. Merge them in one part.

We can construct, in time $O(\gamma n^{8k+9} \log n)$,

- rank-decomposition of width $\leq k$ (if rwd $\leq k$)
- branch-decomposition of width$\leq k$ (if bwd $\leq k$) for matroids.

We present:

Fixed-parameter-tractable algorithm to construct

- rank-decomposition of width $\leq k$ (if rwd $\leq k$)
- branch-decomposition of width$\leq k$ (if bwd $\leq k$)
  for matroids represented over a fixed finite field.

An essential step is:

Can we test branch-width of a partitioned matroid $\leq k$?

- Partition= disjoint nonempty subsets of V whose union is V.
- Partitioned matroid:
  a matroid with a partition of the element set.
- Branch-width of a partitioned matroid:
  treat each part as a single element.

Then recursively find a pair a, b such that merging them does not increase branch-width. Merge them in one part and repeat.

We present:

Fixed-parameter-tractable algorithm to construct

- rank-decomposition of width $\leq k$ (if rwd $\leq k$)
- branch-decomposition of width$\leq k$ (if bwd $\leq k$)
  for matroids represented over a fixed finite field.

An essential step is:

Can we test branch-width of a partitioned matroid $\leq k$?

- Partition= disjoint nonempty subsets of V whose union is V.
- Partitioned matroid:
  a matroid with a partition of the element set.
- Branch-width of a partitioned matroid:
  treat each part as a single element.

Then recursively find a pair a, b such that merging them does not increase branch-width. Merge them in one part and repeat.

We present:

Fixed-parameter-tractable algorithm to construct

- rank-decomposition of width $\leq k$ (if rwd $\leq k$)
- branch-decomposition of width$\leq k$ (if bwd $\leq k$)
  for matroids represented over a fixed finite field.

An essential step is:

Can we test branch-width of a partitioned matroid $\leq k$?

- Partition= disjoint nonempty subsets of V whose union is V.
- Partitioned matroid:
  a matroid with a partition of the element set.
- Branch-width of a partitioned matroid:
  treat each part as a single element.

Then recursively find a pair a, b such that merging them does not increase branch-width. Merge them in one part and repeat.

Essence of the algorithm

From a given partitioned matroid $(M, \mathcal{P})$ represented over a finite field F,

- find a 'normalized matroid' N such that $\mathrm{bwd}(M, \mathcal{P}) = \mathrm{bwd}(N)$.
- Try to apply Hliněný's algorithm to decide whether branch-width of $N \leq k$.

- Attach a gadget to each part to create N.
- Make sure that N is representable over a finite filed F', where $|F'| <$ some function$(|F|, k)$.

Essence of the algorithm

From a given partitioned matroid $(M, \mathcal{P})$
represented over a finite field F,

- find a 'normalized matroid' N such that $\text{bwd}(M, \mathcal{P}) = \text{bwd}(N)$.
- Try to apply Hliněný's algorithm to
  decide whether branch-width of $N \leq k$.

- Attach a gadget to each part to create N.
- Make sure that N is representable over a finite filed F',
  where $|F'| <$ some function$(|F|, k)$.

Gadget: titanic set

Definition

- A set A is titanic if
  for every partition (X_1, X_2, X_3) of A,
  $\exists i$, $f(X_i) \geq f(A)$.
- A partition $\{P_1, P_2, \ldots, P_m\}$ is titanic
  if P_i is titanic for all i.
- Width of a partition: $\max f(P_i)$.

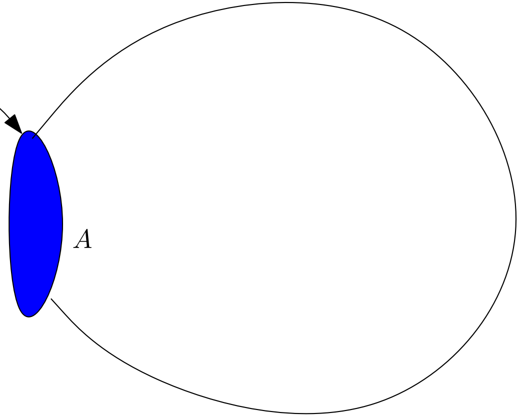
RS1991, Graph Minors X: if $\text{bwd}(f) \leq k$, $f(A) \leq k$, and A is titanic,
then $V \setminus A$ is k-branched.

Theorem

If $\mathcal{P}$: titanic partition of width $\leq k$, and $\text{bwd}(f) \leq k$,
then $\text{bwd}(f, \mathcal{P}) \leq k$.

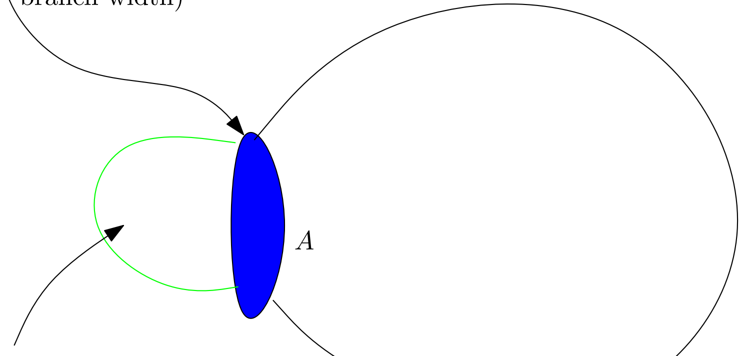
Gadget for matroids: Amalgam with uniform matroids

$\lambda(A) = |A| \leq k$
(otherwise, contract or delete some $\in A$, maintaining the same partitioned branch-width)



Gadget for matroids: Amalgam with uniform matroids

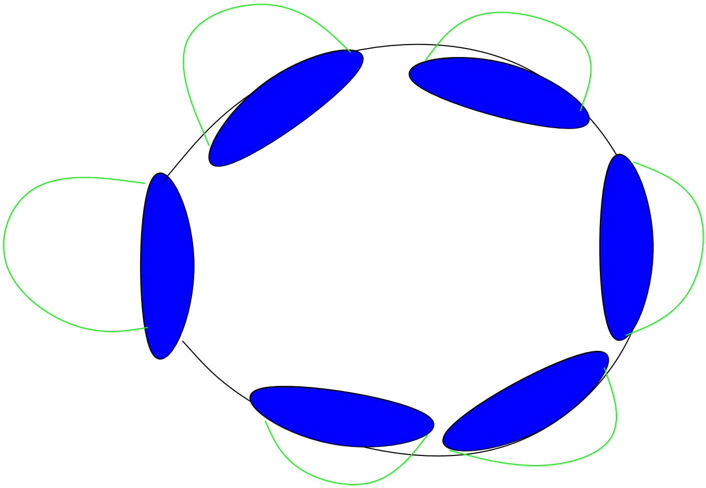
$\lambda(A) = |A| \leq k$
(otherwise, contract or delete some $\in A$, maintaining the same partitioned branch-width)



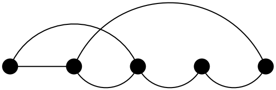
uniform matroid
of $3|A| + 1$ elements, rank $|A|$

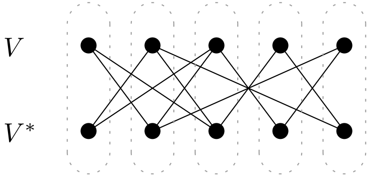
Gadget for matroids: Amalgam with uniform matroids

“Normalized matroid”



Graphs to Binary matroids





$$M = \text{matroid represented by } \begin{array}{cc} & \begin{array}{cc} \quad V \quad & \qquad V^* \qquad \end{array} \\ V & \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & \begin{array}{c}\text{Adjacency} \\ \text{Matrix of } G\end{array} \\ & & 1 & \end{array}\right) \end{array}.$$

Partition $\mathcal{P} = \{v, v^* : v \in V(G)\}$.

Rank-width of G = (Branch-width of $(M, \mathcal{P})$)/2

Running time

We can output

- branch-decomposition of matroids (represented over a fixed finite field) of width $\leq k$
- rank-decomposition of graphs of width $\leq k$

in time

- $O(n^6)$ with the naive implementation.
- $O(n^3)$ if combined Hliněný's algorithm more *seriously*.

(n: number of elements in a matroid, or number of vertices in a graph)

Can you do this for arbitrary connectivity functions?

Thanks for the attention!

Running time

We can output

- branch-decomposition of matroids (represented over a fixed finite field) of width $\leq k$
- rank-decomposition of graphs of width $\leq k$

in time

- $O(n^6)$ with the naive implementation.
- $O(n^3)$ if combined Hliněný's algorithm more *seriously*.

(n: number of elements in a matroid, or number of vertices in a graph)

Can you do this for arbitrary connectivity functions?

Thanks for the attention!

Running time

We can output

- branch-decomposition of matroids (represented over a fixed finite field) of width $\leq k$
- rank-decomposition of graphs of width $\leq k$

in time

- $O(n^6)$ with the naive implementation.
- $O(n^3)$ if combined Hliněný's algorithm more *seriously*.

(n: number of elements in a matroid, or number of vertices in a graph)

Can you do this for arbitrary connectivity functions?

Thanks for the attention!