# Approximating Rank-width and Clique-width Quickly

SANG-IL OUM
KAIST

Rank-width was defined by Oum and Seymour [2006. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B 96,* 4, 514–528] to investigate clique-width. They constructed an algorithm that either outputs a rank-decomposition of width at most $f(k)$ for some function $f$ or confirms that rank-width is larger than $k$ in time $O(|V|^9 \log |V|)$ for an input graph $G = (V, E)$ and a fixed $k$. We develop three separate algorithms of this kind with faster running time. We construct an $O(|V|^4)$-time algorithm with $f(k) = 3k + 1$ by constructing a subroutine for the previous algorithm; we avoid generic algorithms minimizing submodular functions used by Oum and Seymour. Another one is an $O(|V|^3)$-time algorithm with $f(k) = 24k$ by giving a reduction from graphs to binary matroids; then we use an approximation algorithm for matroid branch-width by Hliněný [2005. A parametrized algorithm for matroid branch-width. *SIAM J. Comput. 35,* 2, 259–277]. Finally we construct an $O(|V|^3)$-time algorithm with $f(k) = 3k - 1$ by combining the ideas of above two cited papers.

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Approximation algorithms, branch-width, clique-width, rank-width, , matroids

## 1. INTRODUCTION

Graph complexity measures such as tree-width and branch-width are important for algorithmic purposes and for understanding the structure of families of graphs. One of them is the clique-width, defined by Courcelle and Olariu [2000]. We discuss its definition in the next section. Many NP-hard graph problems are solvable in polynomial time if a tree-like decomposition corresponding to clique-width, called

This research was partially supported by NSF grant 0354742 while the author was at Georgia Institute of Technology and was also partially supported by the SRC Program of Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (No. R11-2007-035-01002-0).

An earlier paper on the first two algorithms appeared in *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science, 2005, Lecture Notes in Comput. Sci. 3787, Springer-Verlag, pp. 49–58.* and is also contained in the author's Ph.D. Thesis: *Graphs of Bounded Rank-width*, Princeton University, 2005.

Author's address: Department of Mathematical Sciences, KAIST, Daejeon, 305-701 Republic of Korea, e-mail: sangil@kaist.edu.

| Paper | Running time | $f(k)$ | Remark |
|---|---|---|---|
| Oum and Seymour [2006] | $O(n^9 \log n)$ | $3k+1$ | Works for any symmetric submodular functions with some conditions. |
| Section 3 | $O(n^4)$ | $3k+1$ | Provides a subroutine for [Oum and Seymour 2006] designed for rank-width. |
| Section 4 | $O(n^3)$ | $24k$ | Uses matroids and Hliněný's [2005] |
| Section 5 | $O(n^3)$ | $3k-1$ | Uses monadic second-order logic. |

Table I. Old and new approximation algorithms for rank-width of an $n$-vertex graph.

a *$k$-expression*, is given as an input in addition to the adjacency list of the input graph. (In fact, the clique-width is the minimum $k$ so that there is a $k$-expression of $G$.) In general, every graph problem expressible in monadic second-order logic with quantifications over vertices and vertex sets ($\mathrm{MS}_1$-logic) can be solved in linear time if the input graph is given with a $k$-expression [Courcelle et al. 2000]. However, problems such as deciding whether the graph is Hamiltonian [Wanke 1994] and finding the chromatic number [Kobler and Rotics 2003] are not expressible in monadic second-order logic but have nevertheless polynomial-time algorithms on graphs of bounded clique-width if the input graph is given with a $k$-expression.

Therefore we hope to have, for each fixed $k$, a polynomial-time algorithm to find a $k$-expression of an input graph if the input graph has clique-width at most $k$. This problem is still open when $k > 3$. Corneil et al. [2000] solved this problem when $k = 3$.

Instead, Oum and Seymour [2006] found an "approximation" algorithm that either outputs a $(2^{3k+2} - 1)$-expression or confirms that the clique-width of $G$ is larger than $k$. This can be combined with algorithms requiring a $k$-expression and therefore those algorithms no longer have to require a $k$-expression as an input to be polynomial-time algorithms. To obtain this approximation algorithm, they defined another graph width parameter, called the rank-width and showed that rank-width is at most clique-width and clique-width is at most $2^{1+\text{rank-width}} - 1$. In addition, they showed a polynomial-time algorithm to find a rank-decomposition of width $3k + 1$ or to confirm that the rank-width is larger than $k$. (Rank-width is defined as the minimum possible width of all rank-decompositions. We will discuss its definition in the next section.)

In this paper we improve their results; we present three separate algorithms with faster running time that, for fixed $k$, output a rank-decomposition of width at most $f(k)$ for some function $f$ or confirm that the rank-width is larger than $k$. We summarize them in Table I.

The following is one of the consequences.

COROLLARY 1.1. *Let $k$ be a fixed positive integer. There is an $O(|V(G)|^3)$-time algorithm that either outputs an $(8^k - 1)$-expression of an input graph $G$ or confirms that the clique-width of $G$ is larger than $k$.*

Courcelle and Oum [2007] showed that, for each $k$, there is a formula of modulo-2 counting monadic second-order logic ($\mathrm{C}_2\mathrm{MS}$ logic) expressing that the rank-width is at most $k$. $\mathrm{C}_2\mathrm{MS}$ logic is an extension of first order logic, which allows set variables denoting sets of objects (in our case, sets of vertices) and the set predicate

Even($X$) expressing that the set denoted by the set variable $X$ is finite and has *even* cardinality. In fact, every graph problem expressible in C$_2$MS logic can also be solved in linear time if an input graph is given with a $k$-expression [Courcelle 1997] (or see [Courcelle and Oum 2007] for the brief explanation). The following theorem is a consequence of our paper combined with [Courcelle and Oum 2007].

THEOREM 1.2. *For fixed $k$, there is an $O(|V(G)|^3)$-time algorithm to test whether the rank-width of a graph $G$ is at most $k$.*

We remark that, if $k$ is not fixed and is given with an input, then testing whether the clique-width at most $k$ is NP-complete, shown by Fellows, Rosamond, Rotics and Szeider [2006]. It is also NP-complete to test whether the rank-width is at most $k$ if $k$ is given with an input. This can be deduced from NP-hardness of finding branch-width of graphs [Seymour and Thomas 1994] by relating the rank-width of bipartite graphs and the branch-width of binary matroids [Oum 2005b] and using the theorem [Fomin et al. 2004; Hicks and McMurray Jr. 2007; Mazoit and Thomassé 2005] stating that the branch-width of the cycle matroid of a graph with at least one cycle is equal to the branch-width of the graph.

## 2. PRELIMINARIES

In this paper, all graphs are simple, undirected, and finite.

### 2.1 Cut-rank functions.

For a matrix $M = (m_{ij} : i \in R, j \in C)$ over a field $F$, if $X \subseteq R$ and $Y \subseteq C$, let $M[X, Y]$ denote the submatrix $(m_{ij} : i \in X, j \in Y)$. For a graph $G$, let $A(G)$ be its adjacency matrix over GF(2).

For a graph $G$ and two disjoint subsets $X, Y \subseteq V(G)$, we define

$$\rho^*_G(X, Y) = \mathrm{rk}(A(G)[X, Y])$$

where rk is the matrix rank function; and we define the *cut-rank* function $\rho_G$ of $G$ by letting $\rho_G(X) = \rho^*_G(X, V(G) \setminus X)$ for $X \subseteq V(G)$.

Both $\rho$ and $\rho^*$ satisfy the following submodular inequalities.

PROPOSITION 2.1 [OUM AND SEYMOUR 2006]. *Let $G$ be a graph. Let $X_1, Y_1, X_2, Y_2$ be subsets of $V(G)$ such that $X_1 \cap Y_1 = X_2 \cap Y_2 = \emptyset$. Then,*

$$\rho^*_G(X_1, Y_1) + \rho^*_G(X_2, Y_2) \geq \rho^*_G(X_1 \cap X_2, Y_1 \cup Y_2) + \rho^*_G(X_1 \cup X_2, Y_1 \cap Y_2).$$

*Moreover, if $X_1, X_2 \subseteq V(G)$ then*

$$\rho_G(X_1) + \rho_G(X_2) \geq \rho_G(X_1 \cap X_2) + \rho_G(X_1 \cup X_2).$$

### 2.2 Rank-width.

A *subcubic tree* is a tree with at least two vertices such that every vertex is incident with at most three edges. A *leaf* of a tree is a vertex incident with exactly one edge. A *rank-decomposition* of a graph $G = (V, E)$ is a pair $(T, \mathcal{L})$ of a subcubic tree $T$ and a bijective function $\mathcal{L} : V \to \{t : t \text{ is a leaf of } T\}$. (If $|V| \leq 1$ then $G$ admits no rank-decomposition.)

For an edge $e$ of $T$, the connected components of $T \setminus e$ induce a partition $(X, Y)$ of the set of leaves of $T$. The *width* of an edge $e$ of a rank-decomposition $(T, \mathcal{L})$

is $\rho_G(\mathcal{L}^{-1}(X))$. The *width* of $(T, \mathcal{L})$ is the maximum width of all edges of $T$. The *rank-width* $\mathrm{rw}(G)$ of $G$ is the minimum of the width of all rank-decompositions of $G$. (If $|V| \leq 1$, we define $\mathrm{rw}(G) = 0$.)

### 2.3 Clique-width.

The notion of clique-width was first introduced by Courcelle and Olariu [2000]. Let $k$ be a positive integer. We call $(G, lab)$ a *$k$-graph* if $G$ is a graph and $lab$ is a mapping from its vertex set to $\{1, 2, \ldots, k\}$. We call $lab(v)$ the *label* of a vertex $v$.

We need the following definitions and operations on $k$-graphs.

(1) For $i \in \{1, \ldots, k\}$, let $\cdot_i$ denote a $k$-graph with a single vertex labeled by $i$.

(2) For distinct $i, j \in \{1, 2, \ldots, k\}$, we define a unary operator $\eta_{i,j}$ such that
$$\eta_{i,j}(G, lab) = (G', lab)$$
where $V(G') = V(G)$ and $E(G') = E(G) \cup \{vw : v, w \in V(G), lab(v) = i, lab(w) = j\}$. This adds edges between vertices labeled by $i$ and vertices labeled by $j$.

(3) We let $\rho_{i\to j}$ be the unary operator such that
$$\rho_{i\to j}(G, lab) = (G, lab')$$
where
$$lab'(v) = \begin{cases} j & \text{if } lab(v) = i, \\ lab(v) & \text{otherwise.} \end{cases}$$
This mapping relabels every vertex labeled by $i$ into $j$.

(4) Finally, $\oplus$ is a binary operation that makes the disjoint union. Note that $G \oplus G \neq G$.

A well-formed expression $t$ written with these symbols is called a *$k$-expression.* The $k$-graph produced by performing these operations in order therefore has vertex set the set of occurrences of the constant symbols in $t$; and this $k$-graph (and any $k$-graph isomorphic to it) is called the *value* $val(t)$ of $t$. If a $k$-expression $t$ has value $(G, lab)$, we say that $t$ is a *$k$-expression* of $G$. The *clique-width* of a graph $G$, denoted by $\mathrm{cw}(G)$, is the minimum $k$ such that there is a $k$-expression of $G$.

For instance, $K_4$ (the complete graph with four vertices) can be constructed by
$$\eta_{1,2}(\cdot_2 \oplus \rho_{2\to1}(\eta_{1,2}(\cdot_2 \oplus \rho_{2\to1}(\eta_{1,2}(\cdot_1 \oplus \cdot_2))))).$$
Therefore, $K_4$ has a 2-expression, and $\mathrm{cw}(K_4) \leq 2$. It is easy to see that $\mathrm{cw}(K_4) > 1$, and therefore $\mathrm{cw}(K_4) = 2$. We remark that an $n$-vertex graph of clique-width at most $k$ can have arbitrary long $k$-expressions, but such long $k$-expressions can be shortened to $k$-expressions of length $O(n)$ by removing useless operations.

Oum and Seymour [2006] showed that having small clique-width is equivalent to having small rank-width as follows.

Proposition 2.2 [Oum and Seymour 2006]. *For every graph $G$,*
$$\mathrm{rw}(G) \leq \mathrm{cw}(G) \leq 2^{\mathrm{rw}(G)+1} - 1.$$
*Moreover for fixed $k$, there is an $O(|V(G)|^2)$-time algorithm that transforms a rank-decomposition of width $k$ into a $(2^{k+1} - 1)$-expression of the input graph $G$.*

### 2.4 Local complementation.

For two sets $A$ and $B$, let $A\Delta B = (A \setminus B) \cup (B \setminus A)$.

*Definition* 2.3. Let $G = (V, E)$ be a graph and $v \in V$. The graph obtained by applying *local complementation* at $v$ to $G$ is

$$G * v = (V, E\Delta\{xy : xv, yv \in E, x \neq y\}).$$

The graph obtained by *pivoting* an edge $uv$ is defined by $G \wedge uv = G * u * v * u$. We say that $H$ is *locally equivalent* to $G$ if $H$ can be obtained by applying a sequence of local complementations to $G$.

A pivoting is well-defined because $G * u * v * u = G * v * u * v$ if $u$ and $v$ are adjacent [Oum 2005b]. Both local complementation and pivoting have appeared in several papers, notably in Bouchet's papers [1988; 1989; 1990; 1994].

It is easy to show the following Proposition.

Proposition 2.4 [Bouchet 1989]; see [Oum 2005b]. *Let $G' = G * v$. Then for every $X \subseteq V(G)$, we have*

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

The following lemma will be used in Section 3.

Lemma 2.5 [Oum 2005b]. *Let $G$ be a graph and $v \in V(G)$. Suppose that $(X_1, X_2)$ and $(Y_1, Y_2)$ are partitions of $V(G) \setminus \{v\}$. If $w$ is a neighbor of $v$ then*

$$\rho_{G\setminus v}(X_1) + \rho_{G\wedge vw\setminus v}(Y_1) \geq \rho_G(X_1 \cap Y_1) + \rho_G(X_2 \cap Y_2) - 1.$$

## 3. FIRST ALGORITHM

We show that, for fixed $k$, there is an $O(n^4)$-time algorithm that, with a $n$-vertex graph, outputs a rank-decomposition of width at most $3k + 1$ or confirms that the input graph has rank-width larger than $k$. Oum and Seymour [2006] used generic algorithms minimizing submodular functions [Iwata et al. 2001] to find a set $Z$ minimizing the cut-rank function $\rho_G(Z)$ such that $X \subseteq Z \subseteq V(G) \setminus Y$ for given disjoint subsets $X$, $Y$ of $V(G)$ satisfying $|X|, |Y| \leq 3k$. If this can be done in time $\gamma$ then we obtain an $O(n(n^2 + \gamma))$-time algorithm to output a rank-decomposition of width at most $3k+1$ or confirm that the input graph has rank-width larger than $k$. In [Oum and Seymour 2006], $\gamma$ is $O(n^8 \log n)$, and therefore the $O(n^9 \log n)$-time algorithm is obtained.

To obtain an $O(n^4)$-time algorithm, we construct a direct combinatorial algorithm that minimizes the cut-rank function. We first define *blocking sequences*, introduced by Geelen [1995]. Let $G$ be a graph and $A, B$ be two disjoint subsets of $V(G)$. A sequence $v_1, v_2, \ldots, v_m$ of distinct vertices in $V(G) \setminus (A \cup B)$ is called a *blocking sequence* for $(A, B)$ in $G$ if it satisfies the following:

(i) $\rho^*_G(A, B \cup \{v_1\}) > \rho^*_G(A, B)$.

(ii) $\rho^*_G(A \cup \{v_i\}, B \cup \{v_{i+1}\}) > \rho^*_G(A, B)$ for all $i \in \{1, 2, \ldots, m-1\}$.

(iii) $\rho^*_G(A \cup \{v_m\}, B) > \rho^*_G(A, B)$.

(iv) No proper subsequence satisfies (i)—(iii).

The following proposition appears frequently in applications of blocking sequences.

Proposition 3.1. *Let $G$ be a graph and $A, B$ be two disjoint subsets of $V(G)$. The following are equivalent:*

*(i) There is no blocking sequence for $(A, B)$ in $G$.*

*(ii) There exists $Z$ such that $A \subseteq Z \subseteq V(G) \setminus B$ and $\rho_G(Z) = \rho^*_G(A, B)$.*

Proof. (i)→(ii): Let $k = \rho^*_G(A, B)$. We construct the *auxiliary digraph* $D = ((V(G) \setminus (A \cup B)) \cup \{A^\circ, B^\circ\}, E)$ from $G$ such that

i) $(A^\circ, x) \in E$ if $\rho^*_G(A, B \cup \{x\}) > k$ for $x \in V(G) \setminus (A \cup B)$,

ii) $(x, B^\circ) \in E$ if $\rho^*_G(A \cup \{x\}, B) > k$ for $x \in V(G) \setminus (A \cup B)$,

iii) $(x, y) \in E$ if $\rho^*_G(A \cup \{x\}, B \cup \{y\}) > k$ for distinct $x, y \in V(G) \setminus (A \cup B)$.

Since there is no blocking sequence for $(A, B)$ in $G$, there is no directed path from $A^\circ$ to $B^\circ$ in $D$. Let $J$ be the set of ends of directed paths in $D$ from $A^\circ$ with at least one edge. We claim that $\rho_G(A \cup J) = k$.

To prove this, we would like to show that $\rho^*_G(A \cup X, B \cup Y) = k$ for all $X \subseteq J$, $Y \subseteq V(G) \setminus (A \cup B \cup J)$. First notice that $\rho^*_G(A \cup X, B \cup Y) \geq \rho^*_G(A, B) = k$ and therefore it is enough to prove that $\rho^*_G(A \cup X, B \cup Y) \leq k$.

We proceed by induction on $|X| + |Y|$. If $|X| \leq 1$ and $|Y| \leq 1$, then we have $\rho^*_G(A, B) \leq \rho^*_G(A \cup X, B \cup Y) \leq k$ because there is no directed path from $A^\circ$ to $B^\circ$ in $D$.

If $|X| > 1$ then from Proposition 2.1, for every $x \in X$ we have

$$\rho^*_G(A \cup X, B \cup Y) + \rho^*_G(A, B \cup Y) \leq \rho^*_G(A \cup (X \setminus \{x\}), B \cup Y) + \rho_G(A \cup \{x\}, B \cup Y) = 2k,$$

because $\rho^*_G(A \cup \{x\}, B \cup Y) = \rho^*_G(A \cup (X \setminus \{x\}), B \cup Y) = \rho^*_G(A, B \cup Y) = k$ by induction. So, $\rho^*_G(A \cup X, B \cup Y) = k$.

Similarly if $|Y| > 1$ then for every $y \in Y$ we have $\rho^*_G(A \cup X, B \cup Y) + \rho^*_G(A \cup X, B) \leq \rho^*_G(A \cup X, B \cup (Y \setminus \{y\})) + \rho_G(A \cup X, B \cup \{y\}) = 2k$, and therefore $\rho^*_G(A \cup X, B \cup Y) = k$.

(ii)→(i): Suppose that there is a blocking sequence $v_1, v_2, \ldots, v_m$. Then, $v_m \notin Z$ because $\rho^*_G(A \cup \{v_m\}, B) > \rho_G(Z)$. Similarly $v_1 \in Z$ because $\rho^*_G(A, B \cup \{v_1\}) > \rho_G(Z)$. Therefore there exists $i \in \{1, 2, \ldots, m-1\}$ such that $v_i \in Z$ but $v_{i+1} \notin Z$. But this is a contradiction, because $\rho_G(Z) < \rho^*_G(A \cup \{v_i\}, B \cup \{v_{i+1}\}) \leq \rho^*_G(Z, V(G) \setminus Z) = \rho_G(Z)$. □

The proof of Proposition 3.1 is algorithmic. We summarize the algorithm as follows.

Algorithm 3.2. *Find a blocking sequence for $(A, B)$ in $G$.*

*Input. A graph $G = (V, E)$ and two disjoint subsets $A$ and $B$ of $V$.*

*Output. One of the following:*

*(1) a blocking sequence $v_1, v_2, \ldots, v_m$ for $(A, B)$ in $G$, or*

*(2) a set $Z$ such that $A \subseteq Z \subseteq V \setminus B$, $\rho_G(Z) = \rho^*_G(A, B)$.*

*(B1) Construct the auxiliary digraph $D = ((V \setminus \{A, B\}) \cup \{A^\circ, B^\circ\}, E')$ such that*

*i) $(A^\circ, x) \in E'$ if $\rho^*_G(A, B \cup \{x\}) > \rho^*_G(A, B)$ for all $x \in V(G) \setminus (A \cup B)$,*

*ii) $(x, B^\circ) \in E'$ if $\rho^*_G(A \cup \{x\}, B) > \rho^*_G(A, B)$ for all $x \in V(G) \setminus (A \cup B)$,*

*iii) $(x, y) \in E'$ if $\rho^*_G(A \cup \{x\}, B \cup \{y\}) > \rho^*_G(A, B)$ for all distinct $x, y \in V(G) \setminus (A \cup B)$.*

*(B2) Find a shortest directed path from $A^\circ$ to $B^\circ$ in $D$.*
*If there is no such path then let $J$ be the set of ends of all directed paths from $A^\circ$ in $D$ with at least one edge. Then $\rho_G(A \cup J) = \rho^*_G(A,B)$ by the proof of Proposition 3.1. Output $Z = A \cup J$ and stop.*
*Otherwise let $A^\circ, v_1, v_2, \ldots, v_m, B^\circ$ be the shortest path in $D$. Then output $v_1, v_2, \ldots, v_m$ as a blocking sequence for $(A,B)$ in $G$.*

Proposition 3.3. *Let $G = (V,E)$ be a graph and $A, B$ be disjoint subsets of $V$. If $v_1, v_2, \ldots, v_m$ is a blocking sequence for $(A,B)$ in $G$ then there is a vertex $w \in B$ adjacent to $v_m$. Furthermore,*

*(1) if $m > 1$ then $v_1, v_2, \ldots, v_{m-1}$ is a blocking sequence for $(A,B)$ in $G \wedge wv_m$,*
*(2) if $m = 1$ then $\rho^*_{G\wedge wv_m}(A,B) = \rho^*_G(A,B) + 1$.*

Proof. Let $k = \rho^*_G(A,B)$. Since $\rho^*_G(A \cup \{v_m\}, B) > k$, there is a vertex $w \in B$ adjacent to $v_m$.

We prove one short lemma to be used:
Lemma: *If $X$ and $Y$ are disjoint subsets of $V$ such that $A \subseteq X$, $B \subseteq Y$, $v_m \notin X \cup Y$ and $\rho^*_G(X,Y) = k$ then $\rho^*_{G\wedge wv_m}(X,Y) = \rho^*_G(X, Y \cup \{v_m\})$.*

Let us prove the lemma. Let $H = G[X \cup Y \cup \{v_m\}]$ (the induced subgraph of $G$ on $X \cup Y \cup \{v_m\}$). We observe that $\rho^*_{G\wedge wv_m}(X \cup \{v_m\}, Y) = \rho_{H\wedge wv_m}(X \cup \{v_m\}) = \rho_H(X \cup \{v_m\}) = \rho^*_G(X \cup \{v_m\}, Y)$ and $\rho^*_{G\wedge wv_m}(X, Y \cup \{v_m\}) = \rho_{H\wedge wv_m}(X) = \rho_H(X) = \rho^*_G(X, Y \cup \{v_m\})$. Then,

$$\begin{aligned}\rho^*_{G\wedge wv_m}(X,Y) + \rho^*_G(X,Y) &\geq \rho^*_G(X, Y \cup \{v_m\}) + \rho^*_G(X \cup \{v_m\}, Y) - 1\\ &\geq \rho^*_G(X, Y \cup \{v_m\}) + k\\ &= \rho^*_{G\wedge wv_m}(X, Y \cup \{v_m\}) + \rho^*_G(X,Y).\end{aligned}$$

It follows that $\rho^*_{G\wedge wv_m}(X,Y) = \rho^*_{G\wedge wv_m}(X, Y \cup \{v_m\})$ and the lemma is proved.

Now let us prove the Proposition by using the lemma.

(1) We claim that if $m > 1$ then $v_1, v_2, \ldots, v_{m-1}$ is a blocking sequence for $(A,B)$ in $G \wedge wv_m$.

Let us first check the axiom (i) of blocking sequences. By applying Lemma 2.5 for $G[A \cup B \cup \{v_1, v_m\}]$ (the subgraph of $G$ induced on $A \cup B \cup \{v_1, v_m\}$), we have

$$\rho^*_{G\wedge wv_m}(A, B \cup \{v_1\}) + \rho^*_G(A \cup \{v_1\}, B) \geq \rho^*_G(A, B \cup \{v_1, v_m\}) + \rho^*_G(A \cup \{v_1, v_m\}, B) - 1.$$

Since $\rho^*_G(A, B \cup \{v_1, v_m\} \geq \rho^*_G(A, B \cup \{v_1\}) \geq k+1$, $\rho^*_G(A \cup \{v_1, v_m\}, B) \geq \rho^*_G(A \cup \{v_m\}, B) \geq k+1$, and $\rho^*_G(A \cup \{v_1\}, B) = k$, we obtain that $\rho^*_{G\wedge wv_m}(A, B \cup \{v_1\}) \geq k+1$.

To check (ii), we apply the same inequality we obtain that

$$\begin{aligned}&\rho^*_{G\wedge wv_m}(A \cup \{v_i\}, B \cup \{v_{i+1}\}) + \rho^*_G(A \cup \{v_i, v_{i+1}\}, B)\\ &\qquad \geq \rho^*_G(A \cup \{v_i\}, B \cup \{v_{i+1}, v_m\}) + \rho^*_G(A \cup \{v_i, v_{i+1}, v_m\}, B) - 1 \geq 2k+1\end{aligned}$$

for each $i \in \{1, 2, 3, \ldots, m-2\}$ and therefore $\rho^*_{G\wedge wv_m}(A \cup \{v_i\}, B \cup \{v_{i+1}\}) \geq k+1$.

To obtain (iii), we apply the above lemma. By letting $X = A \cup \{v_{m-1}\}$ and $Y = B$, we obtain that $\rho^*_{G\wedge wv_m}(A \cup \{v_{m-1}\}, B) = \rho^*_G(A \cup \{v_{m-1}\}, B \cup \{v_m\}) \geq k+1$.

To show (iv), we claim that no subsequence of $v_1, v_2, \ldots, v_{m-1}$ satisfies the axiom (i)—(iii) for blocking sequences. By the lemma, $\rho^*_{G\wedge wv_m}(A, B \cup \{v_i\}) = k$ for each $i > 1$ by letting $X = A$, $Y = B \cup \{v_i\}$. Similarly we have $\rho^*_{G \wedge wv_m}(A \cup \{v_i\}, B \cup \{v_j\}) = k$ for $i, j$ such that $1 \le i < i+1 < j \le m-1$. Finally $\rho^*_{G \wedge wv_m}(A \cup \{v_j\}, B) = k$ for each $j < m-1$ by letting $X = A \cup \{v_j\}$, $Y = B$. Thus, no subsequence of $v_1, v_2, \ldots, v_{m-1}$ is a blocking sequence of $G \wedge wv_m$.

Therefore, $v_1, v_2, \ldots, v_{m-1}$ is a blocking sequence for $(A, B)$ in $G \wedge wv_m$.

(2) If $m = 1$ then we obtain $\rho^*_{G \wedge wv_1}(A, B) = \rho^*_G(A, B \cup \{v_1\})$, by the previous lemma with letting $X = A$ and $Y = B$. Since $v_1$ is a blocking sequence, we have $\rho^*_G(A, B \cup \{v_1\}) \ge k+1$. Since $\rho^*_G(A, B \cup \{v_1\}) \le \rho^*_G(A, B) + 1$, we conclude that $\rho^*_{G \wedge wv_1}(A, B) = k+1$. □

Proposition 3.3 is now used to find a cut with minimum cut-rank while separating $A$ and $B$, as follows.

Algorithm 3.4. *Find a set $Z$ minimizing $\rho_G(Z)$ while separating $A$ and $B$.*

*Input. A graph $G = (V, E)$ and two disjoint subsets $A$ and $B$ of $V$.*

*Output. A subset $Z$ of vertices such that $A \subseteq Z \subseteq V \setminus B$ and*

$$\rho_G(Z) = \min_{A \subseteq X \subseteq V \setminus B} \rho_G(X).$$

```
(M1) Let $G_{0,0} = G$.
(M2) For each integer $i \ge 0$, we run Algorithm 3.2 to find a blocking sequence
     $v^i_1, v^i_2, \ldots, v^i_m$ for $(A, B)$ in $G_{i,0}$ or a set $Z$ such that $\rho_{G_{i,0}}(Z) = \rho^*_{G_{i,0}}(A, B)$.
     If the set $Z$ is obtained, then output $Z$ and stop.
     Otherwise, for each $j = 0, 1, \ldots, m-1$, find a vertex $w \in B$ adjacent to $v_{m-j}$
     in $G_{i,j}$ and let $G_{i,j+1} = G_{i,j} \wedge wv_{m-j}$.
     We let $G_{i+1,0} = G_{i,m}$.
```

Proposition 3.5. *Algorithm 3.4 is correct and its running time is $O(|V|^5)$.*

Proof. Clearly, $G_{i,0}$ is obtained from $G$ by a sequence of pivotings and therefore if we obtain $Z$ at $G_{i,0}$, then $\rho^*_{G_{i,0}}(A, B) = \rho_{G_{i,0}}(Z) = \rho_G(Z)$ and $\rho_G(X) = \rho_{G_{i,j}}(X) \ge \rho^*_{G_{i,j}}(A, B)$ for all $A \subseteq X \subseteq V \setminus B$.

If there is a blocking sequence with $m$ vertices in $G_{i,0}$, then we apply $m$ pivotings. By Proposition 3.3, we reduce the length of the blocking sequence by each pivoting, and after all $m$ pivotings, we have

$$\rho^*_{G_{i+1,0}}(A, B) = \rho^*_{G_{i,m}}(A, B) = \rho^*_{G_{i,0}}(A, B) + 1.$$

Since $\rho^*_{G_{i+1,0}}(A, B) \le \min_{A \subseteq X \subseteq V \setminus B} \rho_G(X) \le |V|$, the loop (M2) runs at most $|V|$ times and it will eventually find no blocking sequences at $G_{i,0}$ for some $i$.

Now let us consider the running time. We store the graph $G$ in the adjacency matrix $A(G)$ by preprocessing the input adjacency list in time $O(|V|^2)$.

To perform (B1) of Algorithm 3.2 on $G_{i,0}$, we first diagonalize $A(G_{i,0})[A, B]$, which is the matrix for $\rho^*_{G_{i,0}}(A, B)$. This can be done in time $O(|V|^3)$. Then each of i) and ii) of Algorithm 3.2 can be done in time $O(|V|^2)$ for each vertex $x$. For each pair $x, y$, iii) can be done in time $O(|V|^2)$. Overall, it takes time $O(|V|^4)$ to construct the auxiliary digraph $D$.

Finding a shortest directed path in (B2) of Algorithm 3.2 is done in time $O(|V|^2)$. If there is no shortest directed path, we can find $J$ in time $O(|V|^2)$ by the depth-first search.

Since $m \leq |V|$, we need apply at most $|V|$ pivotings to get $G_{i+1,0}$ from $G_{i,0}$ in (M2) and this can be done in time $O(|V|^4)$. Notice that we also need time to find $w$ in $B$ and $|B| \leq |V|$.

Since $\min_{A\subseteq X\subseteq V\setminus B} \rho_G(X) \leq \rho_G(A) \leq |V|$, the number of iterations in (M2) is at most $|V|$. Therefore, the running time of Algorithm 3.4 is $O(|V|^5)$. □

Proposition 3.5 is enough to be used for our purposes, but if we limit the size of $|A|$ and $|B|$, then we can achieve a better running time as follows.

PROPOSITION 3.6. *Let $l$ be a fixed constant. If we require the input of Algorithm 3.4 to satisfy $|A|, |B| \leq l$, then the running time of Algorithm 3.4 is $O(|V|^3)$.*

PROOF. We assume that the graph $G$ is stored in the adjacency matrix by preprocessing the adjacency list in time $O(|V|^2)$.

Since the rank of $(l+1)\times(l+1)$ or $(l+1)\times l$ matrices can be evaluated in time $O(l^3) = O(1)$, (B1) of Algorithm 3.2 is done in time $O(|V|^2)$.

Finding a shortest directed path in (B2) of Algorithm 3.2 is done in time $O(|V|^2)$. If there is no shortest directed path, we can find $J$ in time $O(|V|^2)$ by the depth-first search.

In order to get $G_{i+1,0}$ from $G_{i,0}$, we need at most $|V|$ pivotings because $m \leq |V|$. Therefore it can be done in time $O(|V|^3)$; notice that since $|B| \leq l$, it takes a constant time to find $w$.

Since $\min_{A\subseteq X\subseteq V\setminus B} \rho_{G_{i,0}}(X) \leq \rho_{G_{i,0}}(A) \leq l$, the number of iterations in (M2) is at most $l$. Therefore the running time of Algorithm 3.4 for fixed $l$ is $O(|V|^3)$. □

As we discussed in the beginning of this section, we obtain the following theorem.

THEOREM 3.7. *For fixed $k$, there is an algorithm, for the input graph $G = (V,E)$, that either concludes that $\mathrm{rw}(G) > k$ or outputs a rank-decomposition of $G$ of width at most $3k+1$; and its running time is $O(|V|^4)$.*

PROOF. The running time of Oum and Seymour's algorithm [2006] is $O(|V|(|V|^2+\gamma))$ where $\gamma$ is the time to find a set $Z$ minimizing $\rho_G$. Their algorithm used generic algorithms minimizing submodular functions to find a set $Z$ containing $A$ not meeting $B$ and minimizing $\rho_G(Z)$ for some $|A|, |B| \leq 3k$. Instead we use Algorithm 3.4. In particular both $A$ and $B$ has size at most $3k$ and therefore we use the running time proved in Proposition 3.6 instead of Proposition 3.5. So $\gamma = O(|V|^3)$. So the new running time is $O(|V|^4)$. □

Since we can convert the rank-decomposition of width $k$ to a $(2^{k+1}-1)$-expression (a decomposition related to clique-width) in time $O(|V|^2)$ [Oum and Seymour 2006], we obtain the following corollary.

COROLLARY 3.8. *For fixed $k$, there is an algorithm, for the input graph $G = (V,E)$, that either concludes that $\mathrm{cw}(G) > k$ or outputs a $(2^{3k+2}-1)$-expression of $G$; and its running time is $O(|V|^4)$.*

## 4. SECOND ALGORITHM

In the second algorithm, we will transform our problem into a matroid problem and then use Hliněný's algorithm [2005] on matroid branch-width. It turns out that the rank-width of bipartite graphs is essentially equivalent to the branch-width of binary matroids [Oum 2005b]. We will discuss this after showing that we can transform our problem into a problem on bipartite graphs.

### 4.1 Graphs to Bipartite Graphs.

Courcelle [2006] showed that Seese's conjecture [Seese 1991] is true if and only if it is true for bipartite graphs by using a graph transformation $B$ from graphs to bipartite graphs described as follows. For a graph $G = (V, E)$, $B(G)$ is defined as a bipartite graph on the vertex set $V \times \{1, 2, 3, 4\}$ such that

(i) if $v \in V$ and $i \in \{1, 2, 3\}$ then $(v, i)$ is adjacent to $(v, i + 1)$ in $B(G)$,

(ii) if $vw \in E$ then $(v, 1)$ is adjacent to $(w, 4)$ in $B(G)$.

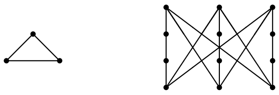

Fig. 1. $K_3$ and $B(K_3)$

A theorem by Courcelle [2006] combined with Proposition 2.2 implies that there exist two functions $f_1$ and $f_2$ such that $f_1(\mathrm{rw}(G)) \leq \mathrm{rw}(B(G)) \leq f_2(\mathrm{rw}(G))$, but does not give explicit constructions of $f_1$ and $f_2$. We will prove this inequality when $f_1(k) = k/4$ and $f_2(k) = \max(2k, 1)$. This result will be used in the next section.

Proposition 4.1. *For every graph $G$, we have* $\mathrm{rw}(B(G)) \leq \max(2\,\mathrm{rw}(G), 1)$.

Proof. If the rank-width of $G$ is zero then $G$ has no edges at all and it is easy to see that the rank-width of $B(G)$ is 1.

Now we assume that the rank-width of $G$ is non-zero. We claim that $\mathrm{rw}(B(G)) \leq 2\,\mathrm{rw}(G)$. Let $k$ be the rank-width of $G$. Then there is a rank-decomposition $(T, \mathcal{L})$ of $G$ of width $k$. Let $N$ be the set of leaves of $T$.

We aim to construct a rank-decomposition $(T', \mathcal{L}')$ of $B(G)$ so that its width is at most $2k$. Let $T'$ be a tree such that $V(T') = (V(T) \times \{0\}) \cup (N \times \{1, 2, 3, 4, 12, 34\})$ and

(i) if $vw \in E(T)$ then $(v, 0)$ is adjacent to $(w, 0)$ in $T'$,

(ii) for all $v \in N$, $(v, 12)$ is adjacent to both $(v, 1)$ and $(v, 2)$ in $T'$,

(iii) for all $v \in N$, $(v, 34)$ is adjacent to both $(v, 3)$ and $(v, 4)$ in $T'$,

(iv) for all $v \in N$, $(v, 0)$ is adjacent to both $(v, 12)$ and $(v, 34)$ in $T'$.

Informally speaking, we obtain $T'$ from $T$ by replacing each leaf with a rooted binary tree having four leaves. For each vertex $(v, i)$ of $B(G)$, we define $\mathcal{L}'((v, i)) = (\mathcal{L}(v), i) \in V(T')$. Then $(T', \mathcal{L}')$ is a rank-decomposition of $B(G)$.

We claim that the width of $(T', \mathcal{L}')$ is at most $2k$.

For each edge $e = vw \in E(T)$, let $(X, Y)$ be a partition of $N$ induced by the connected components of $T \setminus e$. Then, the edge $(v, 0)(w, 0)$ of $E(T')$ induces a partition $(X \times \{1, 2, 3, 4\}, Y \times \{1, 2, 3, 4\})$ of $N \times \{1, 2, 3, 4\}$. We observe that $\mathcal{L}'^{-1}(X \times \{1, 2, 3, 4\}) = \mathcal{L}^{-1}(X) \times \{1, 2, 3, 4\}$. It is straightforward to see that

$$\rho_{B(G)}(\mathcal{L}'^{-1}(X \times \{1, 2, 3, 4\}) = 2\rho_G(\mathcal{L}^{-1}(X)) \leq 2k.$$

We now consider remaining edges of $T'$. Each of them induces a partition $(X, Y)$ of leaves of $T'$ such that $|X| \leq 2$ or $|Y| \leq 2$. So, $\rho_{B(G)}(\mathcal{L}'^{-1}(X)) \leq 2$. Therefore we obtain that the width of $(T', \mathcal{L}')$ is at most $2k$. □

Proposition 4.2. *Let $G = (V, E)$ be a graph and $(T', \mathcal{L}')$ be a rank-decomposition of $B(G)$ of width $k$. If $T$ is a minimum subtree of $T'$ containing all leaves in $\mathcal{L}'(V \times \{1\})$ and $\mathcal{L} : V \to \{t : t \text{ is a leaf of } T\}$ is a bijective function defined as $\mathcal{L}(v) = \mathcal{L}'((v, 1))$, then $(T, \mathcal{L})$ is a rank-decomposition of $G$ of width at most $4k$.*

*In particular this implies that* $\mathrm{rw}(G) \leq 4\,\mathrm{rw}(B(G))$.

Proof. Let $e$ be an edge of $T$. Since $T$ is a subtree of $T'$, $e$ is also an edge of $T'$. Let $(X, Y)$ be a partition of leaves of $T'$ induced by connected components of $T' \setminus e$.

For subsets $A_1, A_2, A_3, A_4$ of $V$, we denote the set $(A_1 \times \{1\}) \cup (A_2 \times \{2\}) \cup (A_3 \times \{3\}) \cup (A_4 \times \{4\})$ by $A_1|A_2|A_3|A_4$ to simplify our notation. Let $\mathcal{L}'^{-1}(X) = A_1|A_2|A_3|A_4$. Let $\overline{A}_i = V \setminus A_i$ for $i \in \{1, 2, 3, 4\}$. Because the width of $(T', \mathcal{L}')$ is $k$, we have

$$\rho_{B(G)}(A_1|A_2|A_3|A_4) = \rho^*_{B(G)}(A_1|A_2|A_3|A_4, \overline{A}_1|\overline{A}_2|\overline{A}_3|\overline{A}_4) \leq k.$$

We now claim that for each $i \in \{1, 2, 3\}$

$$\rho^*_{B(G)}((A_i \times \{i\}) \cup (A_{i+1} \times \{i+1\}), (\overline{A}_i \times \{i\}) \cup (\overline{A}_{i+1} \times \{i+1\}) = |A_i \Delta A_{i+1}|. \quad (1)$$

To see this, we look at the matrix defining the cut-rank functions.

$$\begin{aligned}
&\rho^*_{B(G)}((A_i \times \{i\}) \cup (A_{i+1} \times \{i+1\}), (\overline{A}_i \times \{i\}) \cup (\overline{A}_{i+1} \times \{i+1\}) \\
&\quad = \mathrm{rk}\left[\begin{array}{ccc} & \overline{A}_i \times \{i\} & \overline{A}_{i+1} \times \{i+1\} \\ A_i \times \{i\} & 0 & \text{(0-1 submatrix)} \\ A_{i+1} \times \{i+1\} & \text{(0-1 submatrix)} & 0 \end{array}\right] \\
&\quad = \rho^*_{B(G)}(A_i \times \{i\}, \overline{A}_{i+1} \times \{i+1\}) + \rho^*_{B(G)}(A_{i+1} \times \{i+1\}, \overline{A}_i \times \{i+1\}).
\end{aligned}$$

Both cut-rank functions can be evaluated.

$$\begin{aligned}
\rho^*_{B(G)}(A_i \times \{i\}, \overline{A}_{i+1} \times \{i+1\}) &= \mathrm{rk}\left[\begin{array}{ccc} & A_i \setminus A_{i+1} & \overline{A}_{i+1} \setminus A_i \\ A_i \setminus A_{i+1} & \text{identity matrix} & 0 \\ A_i \cap A_{i+1} & 0 & 0 \end{array}\right] \\
&= |A_i \setminus A_{i+1}|.
\end{aligned}$$

By symmetry, we also obtain that $\rho^*_{B(G)}(A_{i+1} \times \{i+1\}, \overline{A}_i \times \{i+1\}) = |A_{i+1} \setminus A_i|$. Since $|A_i \setminus A_{i+1}| + |A_{i+1} \setminus A_i| = |A_i \Delta A_{i+1}|$, the equation (1) is proved.

Since $\rho^*_{B(G)}((A_i \times \{i\}) \cup (A_{i+1} \times \{i+1\}),\ (\overline{A}_i \times \{i\}) \cup (\overline{A}_{i+1} \times \{i+1\})) \leq \rho_{B(G)}(A_1|A_2|A_3|A_4)$, we have $|A_i \Delta A_{i+1}| \leq k$ for each $i \in \{1, 2, 3\}$. By adding these inequalities for $i \in \{1, 2, 3\}$, we obtain that $|A_1 \Delta A_4| \leq 3k$.

Let $M$ be the adjacency matrix of $G$. We observe that $\mathrm{rk}(M[A_4, \overline{A}_1]) = \rho^*_{B(G)}(A_4 \times \{4\}, \overline{A}_1 \times \{1\}) \leq \rho_{B(G)}(A_1|A_2|A_3|A_4) \leq k$. Then we have the following bound of

$\rho_G(A_1)$:

$$\begin{aligned}\rho_G(A_1) &= \operatorname{rk}(M[A_1,\overline{A}_1])\\ &\le \operatorname{rk}(M[A_4\cup(A_1\Delta A_4),\overline{A}_1])\\ &\le \operatorname{rk}(M[A_4,\overline{A}_1])+\operatorname{rk}(M[A_1\Delta A_4,\overline{A}_1])\le 4k.\end{aligned}$$

Therefore the width of $(T,\mathcal{L})$ is at most $4k$. □

### 4.2 Binary Matroids and Branch-width.

Let us review matroid theory. For general matroid theory, we refer to Oxley's book [1992]. We call $\mathcal{M}=(E,\mathcal{I})$ a *matroid* if $E$ is a finite set and $\mathcal{I}$ is a collection of subsets of $E$, satisfying

(i) $\emptyset\in\mathcal{I}$

(ii) If $A\in\mathcal{I}$ and $B\subseteq A$ then $B\in\mathcal{I}$.

(iii) For every $Z\subseteq E$, the maximal subsets of $Z$ in $\mathcal{I}$ have the same size $r(Z)$. We call $r(Z)$ the *rank* of $Z$.

An element of $\mathcal{I}$ is called *independent* in $\mathcal{M}$. We let $E(\mathcal{M})=E$. A matroid $\mathcal{M}=(E,\mathcal{I})$ is *binary* if there exists a matrix $N$ over GF(2) such that $E$ is a set of column vectors of $N$ and $\mathcal{I}=\{X\subseteq E : X \text{ is linearly independent}\}$. The *connectivity* function $\lambda_{\mathcal{M}}$ of $\mathcal{M}$ is $\lambda_{\mathcal{M}}(X)=r(X)+r(E\setminus X)-r(E)+1$.

Let $G=(V,E)$ be a bipartite graph with a bipartition $V=A\cup B$. Let $Bin(G,A,B)$ be the binary matroid on $V$, represented by the $A\times V$ matrix

$$\begin{array}{cc} & \begin{array}{cc} A & B \end{array}\\ A & \begin{pmatrix} I_A & A(G)[A,B]\end{pmatrix},\end{array}$$

where $I_A$ is the $A\times A$ identity matrix. If $\mathcal{M}=Bin(G,A,B)$ then $G$ is called a *fundamental graph* of $\mathcal{M}$.

A *branch-decomposition* of a matroid $\mathcal{M}$ is a pair $(T,\mathcal{L})$ of a subcubic tree $T$ and a bijective function $\mathcal{L}: E(\mathcal{M})\to\{t : t \text{ is a leaf of } T\}$. (If $|E(\mathcal{M})|\le 1$ then $\mathcal{M}$ admits no branch-decomposition.)

For an edge $e$ of $T$, the connected components of $T\setminus e$ induce a partition $(X,Y)$ of the set of leaves of $T$. The *width* of an edge $e$ of a branch-decomposition $(T,\mathcal{L})$ is $\lambda_{\mathcal{M}}(\mathcal{L}^{-1}(X))$. The *width* of $(T,\mathcal{L})$ is the maximum width of all edges of $T$. The *branch-width* $\operatorname{bw}(\mathcal{M})$ of $\mathcal{M}$ is the minimum of the width of all branch-decompositions of $\mathcal{M}$. (If $|E(\mathcal{M})|\le 1$, we define $\operatorname{bw}(\mathcal{M})=1$.) Branch-width has been defined by Robertson and Seymour [1991].

The following proposition links branch-width of binary matroids with rank-width of bipartite graphs.

Proposition 4.3 [Oum 2005b]. *Let $G=(V,E)$ be a bipartite graph with a bipartition $V=A\cup B$ and let $\mathcal{M}=Bin(G,A,B)$. Then for every $X\subseteq V$, $\lambda_{\mathcal{M}}(X)=\rho_G(X)+1$.*

Corollary 4.4 [Oum 2005b]. *Let $G=(V,E)$ be a bipartite graph with a bipartition $V=A\cup B$ and let $\mathcal{M}=Bin(G,A,B)$. Then the branch-width of $\mathcal{M}$ is one more than the rank-width of $G$.*

### 4.3 Using Binary Matroids.

We will now show another algorithm that approximate rank-width as in Section 3, but in time $O(n^3)$ with a worse approximation ratio. We take a different approach based on Propositions 4.1 and 4.2. We use the following algorithm for binary matroids developed by Hliněný [2005].

Theorem 4.5 [Hliněný 2005, Theorem 4.12]. *For fixed $k$, there is an $O(n^3)$-time algorithm that, for a binary matroid with $n$ elements, either obtains a branch-decomposition of width at most $3k+1$ or confirms that the given matroid has branch-width larger than $k+1$. We assume that binary matroids are given by their matrix representations.*

This algorithm can be used to approximate rank-width of a bipartite graph $G$ because we can run this algorithm for binary matroids having $G$ as a fundamental graph. By Propositions 4.1 and 4.2, we obtain a bipartite graph $B(G)$ for each graph $G$ such that $\frac{1}{4}\operatorname{rw}(G) \leq \operatorname{rw}(B(G)) \leq \max(2\operatorname{rw}(G), 1)$. Moreover we can construct $B(G)$ in time $O(n^2)$ when $n = |V(G)|$ and transform the rank-decomposition of $B(G)$ of width $m$ into a rank-decomposition of $G$ of width at most $4m$ in linear time by Proposition 4.2. Therefore we obtain the following algorithm.

Corollary 4.6. *For fixed $k$, there is an $O(n^3)$-time algorithm that, for an $n$-vertex graph, either obtains a rank-decomposition of width at most $24k$ or confirms that the rank-width of the input graph is larger than $k$.*

Proof. Let $G = (V, E)$ be the input graph. We may assume that $E(G) \neq \emptyset$. First we construct $B(G)$ in time $O(n^2)$. We run the algorithm of Theorem 4.5 with an input consisting of $\mathcal{M} = Bin(B(G), V \times \{1, 3\}, V \times \{2, 4\})$ and the constant $2k$.

If it confirms that the branch-width of $\mathcal{M}$ is larger than $2k + 1$, then the rank-width of $B(G)$ is larger than $2k$, and therefore the rank-width of $G$ is larger than $k$.

If it outputs a branch-decomposition of $\mathcal{M}$ of width at most $6k+1$, then the output is a rank-decomposition of $B(G)$ of width at most $6k$. This can be transformed into a rank-decomposition of $G$ of width at most $24k$ in linear time by Proposition 4.2. □

## 5. THIRD ALGORITHM

In this section we aim to combine the idea of Oum and Seymour [2006] with that of Hliněný [2005]. The main bottleneck of the algorithm by Oum and Seymour was the subroutine minimizing the cut-rank functions. In the first algorithm, we improve the running time by constructing such an algorithm faster than generic algorithms minimizing submodular functions. Here we use monadic second-order logic to achieve a better running time. Given a monadic second-order formula with free variables representing sets of vertices, in order to produce a satisfying assignment to the free variables, we need a $k$-expression of the input graph. We will construct a $k$-expression from subgraphs iteratively. Hliněný [2005] used a similar idea on matroids of bounded branch-width and formulas of an appropriate version of monadic second-order logic. Since we are able to find such assignments in linear time in this case, we can use a lemma that is refined from the paper by Oum and Seymour [2006] to reduce $3k + 1$ to $3k - 1$ in the approximation algorithm.

### 5.1 Monadic second-order logic.

We consider a graph $G$ without parallel edges as a relational structure $\langle V(G), \mathrm{edg}\rangle$ where $\mathrm{edg}(x,y)$ is true if and only if two vertices $x$ and $y$ are adjacent in $G$. *Monadic second-order logic* is the extension of first-order logic allowing variables denoting subsets of the domains of the considered relational structure. On graphs, the domain is the set of vertices. Thus monadic second-order formulas on graphs are logic formulas which can use $\exists$, $\forall$, $\vee$, $\wedge$, $\neg$, $\in$, true, and edg with first-order variables each representing a vertex and set variables each representing a set of vertices. This logic is often called monadic second-order logic of the first kind ($\mathrm{MS}_1$ logic), to distinguish from that of the second kind ($\mathrm{MS}_2$ logic). Monadic second-order logic of the second kind is more general; it allows set variables, each representing a set of edges or a set of vertices.

### 5.2 Refining the Greedy Algorithm for Rank-width.

There is a notion called a *tangle*, that is strongly related to branch-width and rank-width. Here we define it for rank-width. For a graph $G = (V, E)$, a $\rho_G$-tangle of order $k+1$ is a set $\mathcal{T}$ of subsets of $V$ satisfying the following three axioms.

(T1) If $\rho_G(X) \le k$ then either $X \in \mathcal{T}$ or $V \setminus X \in \mathcal{T}$.
(T2) For all $X_1, X_2, X_3 \in \mathcal{T}$, we have $X_1 \cup X_2 \cup X_3 \neq V$.
(T3) $V \setminus \{v\} \notin \mathcal{T}$ for all $v \in V$.

Robertson and Seymour [1991] showed that the minimum order of a $\rho_G$-tangle is equal to the rank-width. Their theorem (3.1) in [Robertson and Seymour 1991] is for arbitrary symmetric submodular functions and in this paper we state its corollary for rank-width.

THEOREM 5.1. *For a graph $G$, no $\rho_G$-tangle of order $k+1$ exists if and only if the rank-width of $G$ is at most $k$.*

We use Theorem 5.1 to prove the following lemma.

LEMMA 5.2. *Let $k$ be a fixed positive integer. Let $G = (V, E)$ be a graph and $(A, B)$ be a partition of $V$ such that $\rho_G(B) = 3k - 1$. If the rank-width of $G$ is at most $k$, then there exists a partition $(S, T)$ of $V$ such that*

*(i)* $\rho_G(S) \le k$,
*(ii)* $\rho^*_G(S \cap A, B) \ge k$ *and* $\rho^*_G(T \cap A, B) \ge k$,
*(iii)* $S \cap B \neq \emptyset$ *and* $T \cap B \neq \emptyset$.

PROOF. Suppose that there is no such partition $(S, T)$ of $V$. We will seek a contradiction by showing a $\rho_G$-tangle of order $k+1$. Let

$$\mathcal{T} = \{X \subseteq V : \rho_G(X) \le k \text{ and } (X \subseteq A \text{ or } \rho^*_G(X \cap A, B) \le k-1)\}.$$

We claim that $\mathcal{T}$ is a $\rho_G$-tangle of order $k+1$,

(T3) is trivial. Let us show (T1). Suppose that $\rho_G(X) \le k$. If $X \subseteq A$ then $X \in \mathcal{T}$. If $V \setminus X \subseteq A$ then $V \setminus X \in \mathcal{T}$. Now we may assume that $X \cap B \neq \emptyset$ and $(V \setminus X) \cap B \neq \emptyset$. By our assumption, either $\rho^*_G(X \cap A, B) \le k-1$ or $\rho^*_G((V \setminus X) \cap A, B) \le k-1$. So either $X \in \mathcal{T}$ or $V \setminus X \in \mathcal{T}$.

We will now show (T2). Suppose that $X_1, X_2, X_3 \in \mathcal{T}$ such that $X_1 \cup X_2 \cup X_3 = V$. Thus we know that for each $i = 1, 2, 3$, we have either $X_i \subseteq A$ or $\rho^*_G(X_i \cap A, B) \leq k - 1$. However, if $X_i \subseteq A$ then $\rho^*_G(X_i \cap A, B) = \rho^*_G(X_i, B) \leq \rho_G(X_i) \leq k$. Thus we have $\rho^*_G(X_i, B) \leq k$ for all $i = 1, 2, 3$. Since $\sum_{i=1}^{3} \rho^*_G(X_i \cap A, B) \geq \rho^*_G(A, B) = 3k - 1$, we may assume that $X_1, X_2 \subseteq A$.

Since $X_1 \cup X_2 \cup X_3 = V$, we deduce that $X_3 \not\subseteq A$ and therefore $\rho^*_G(X_3 \cap A, B) \leq k - 1$. Since $V \setminus X_3 \subseteq A$, we have $\rho^*_G(A \setminus X_3, B) \leq \rho_G(V \setminus X_3) \leq k$ and therefore

$$\rho(A) = 3k - 1 \leq \rho^*_G(A \setminus X_3, B) + \rho^*_G(X_3 \cap A, B) \leq 2k - 1,$$

a contradiction. We just showed that $\mathcal{T}$ satisfies (T2). We conclude that $\mathcal{T}$ is a $\rho_G$-tangle of order $k + 1$, contradictory to Theorem 5.1. □

An important consequence of Lemma 5.2 is the following lemma, which will allow us to use a greedy algorithm to find a rank-decomposition.

Lemma 5.3. *Let $k$ be a fixed positive integer. Let $G = (V, E)$ be a graph and $(A, B)$ be a partition of $V$ such that $\rho_G(B) \leq 3k - 1$ and $|B| \geq 2$. If the rank-width of $G$ is at most $k$, then there exists a partition $(X, Y)$ of $B$ such that $X \neq \emptyset$, $Y \neq \emptyset$, and $\rho_G(X) \leq 3k - 1$ and $\rho_G(Y) \leq 3k - 1$.*

Proof. If $\rho_G(B) \leq 3k - 2$ then let $X$ be a subset of $B$ with $|X| = 1$ and let $Y = B \setminus X$. Then

$$\rho_G(Y) = \rho_G((V \setminus B) \cup X) \leq \rho_G(V \setminus B) + \rho_G(X) \leq 3k - 2 + 1$$

and $\rho_G(X) \leq 1 \leq 3k - 1$.

Therefore we may assume that $\rho_G(B) = 3k - 1$. By Lemma 5.2, there exists a partition $(S, T)$ of $V$ such that $\rho_G(S) \leq k$, $\rho^*_G(S \cap A, B) \geq k$, $\rho^*_G(T \cap A, B) \geq k$, and $S \cap B, T \cap B \neq \emptyset$. By the submodular inequality,

$$\rho_G(S) + \rho_G(A) \geq \rho_G(S \cap A) + \rho_G(S \cup A).$$

Since $\rho_G(S \cap A) \geq \rho^*_G(S \cap A, B) \geq k$ and $\rho_G(S) \leq k$, we deduce that $\rho_G(T \cap B) = \rho_G(S \cup A) \leq \rho_G(A) = 3k - 1$.

Similarly, from the inequality

$$\rho_G(T) + \rho_G(A) \geq \rho_G(T \cap A) + \rho_G(T \cup A),$$

we deduce that $\rho_G(S \cap B) \leq 3k - 1$. Then $(S \cap B, T \cap B)$ is a wanted partition of $B$. □

We note that in fact $\operatorname{rw}(B(G)) = \max(2 \operatorname{rw}(G), 1)$ for all graphs $G$, shown in [Oum 2005a]. But we are not aware of any methods to transform any rank-decomposition of $B(G)$ of width $\max(2k, 1)$ into that of $G$ of width $k$ quickly.

### 5.3 Improving Rank-decompositions Iteratively.

By Lemma 5.3, if we can find such a partition $(X, Y)$ in polynomial time, we can construct a rank-decomposition of width at most $3k - 1$ by a greedy algorithm. If we are given an $l$-expression of $G$ for fixed $l$, then we can use dynamic programming to find such a partition. We use the fact that given a monadic second-order formula with free variables representing sets of vertices, we can find an assignment of the free variables to satisfy the formula in linear time if the $k$-expression is given as an input [Courcelle et al. 2000]. The next algorithm will require an $l$-expression of the input graph $G$. We need a lemma.

Lemma 5.4. *For every $k$, there is a monadic second-order formula $\mu_k(X)$ with a free set variable $X$ such that $\mu_k(X)$ is true in a graph $G$ if and only if $\rho_G(X) \leq k$.*

Proof. Let $Y = V(G) \setminus X$ and let $M = A(G)[X,Y]$, the submatrix of the adjacency matrix of $G$ with rows indexed by $X$ and columns indexed by $Y$. Then $\rho_G(X) = \mathrm{rk}(M) \leq k$ if and only if no $k+1$ distinct row vectors of $M$ are linearly independent. Then we test whether every set of $k+1$ distinct vertices of $X$ contains a nonempty subset $Z$ such that each vertex of $Y$ has even number of neighbors in $Z$. Hence, we express $\mu_k(X)$ as follows: For every choice of $x_1, \ldots, x_{k+1}$ in $V(G)$, one of the following is true:

—$x_i \notin X$ for some $i = 1, 2, \ldots, k+1$.

—$x_i = x_j$ for some $1 \leq i < j \leq k+1$.

—There exists a nonempty subset $Z$ of $\{x_1, x_2, \ldots, x_{k+1}\}$ such that every vertex not in $X$ has an even number of neighbors in $Z$.

Since $Z$ is a subset of $\{x_1, \ldots, x_{k+1}\}$, this can be expressed by first-order variables. Thus $\mu_k$ can be written with a single set variable $X$. □

To illustrate Lemma 5.4, we show $\mu_2(X)$. In the following formula, $x \sim y$ means that $x$ is adjacent to $y$.

$$\begin{aligned}
\mu_2(X) = \forall x_1 \forall x_2 \forall x_3 (x_1 \notin X) &\vee (x_2 \notin X) \vee (x_3 \notin X) \\
&\vee (x_1 = x_2) \vee (x_2 = x_3) \vee (x_3 = x_1) \\
&\vee (\forall y, y \notin X \Rightarrow y \not\sim x_1) \\
&\vee (\forall y, y \notin X \Rightarrow y \not\sim x_2) \\
&\vee (\forall y, y \notin X \Rightarrow y \not\sim x_3) \\
&\vee (\forall y, y \notin X \Rightarrow (y \not\sim x_1 \wedge y \not\sim x_2) \vee (y \sim x_1 \wedge y \sim x_2)) \\
&\vee (\forall y, y \notin X \Rightarrow (y \not\sim x_2 \wedge y \not\sim x_3) \vee (y \sim x_2 \wedge y \sim x_3)) \\
&\vee (\forall y, y \notin X \Rightarrow (y \not\sim x_3 \wedge y \not\sim x_1) \vee (y \sim x_3 \wedge y \sim x_1)) \\
&\vee (\forall y, y \notin X \Rightarrow \\
&\qquad ((y \not\sim x_1 \wedge y \not\sim x_2 \wedge y \not\sim x_3) \\
&\qquad \vee (y \sim x_1 \wedge y \sim x_2 \wedge y \not\sim x_3) \\
&\qquad \vee (y \sim x_1 \wedge y \not\sim x_2 \wedge y \sim x_3) \\
&\qquad \vee (y \not\sim x_1 \wedge y \sim x_2 \wedge y \sim x_3))).
\end{aligned}$$

Algorithm 5.5. *Let $k$, $l$ be fixed constants. We assume that $k \geq 1$.*

*Input.* *A graph $G$, an $l$-expression $t$ of $G$, and a subset $B$ of $V(G)$ such that $\rho_G(B) \leq 3k-1$.*

*Output.* *If $|B| = 1$ then it outputs $(T, \mathcal{L})$ where $T$ is a graph with a single vertex, called a root, and $\mathcal{L} : B \to V(T)$ is a function.*

*If $|B| > 1$ then it either confirms that the rank-width of $G$ is larger than $k$ or outputs a pair $(T, \mathcal{L})$ of a rooted binary tree $T$ and a bijection $\mathcal{L}$ from $B$ to leaves* *of $T$ such that for each edge $e$ of $T$, if $\mathcal{L}(X)$ is the set of all leaves of $T$ which are descendants of $e$, then $\rho_G(X) \leq 3k-1$.*

*(B1) If $|B| = 1$ then we know what to do. Output $(T, \mathcal{L})$.*

*(B2) Now we assume that $|B| \geq 2$.*
*Find a nonempty partition $(X, Y)$ of $B$ such that $\rho_G(X) \leq 3k-1$ and $\rho_G(Y) \leq 3k-1$ by searching an assignment satisfying a monadic second-order formula*

$$\varphi(X, B) = ((X \neq \emptyset) \wedge (X \neq B) \wedge (X \subseteq B) \wedge \mu_{3k-1}(X) \wedge \mu_{3k-1}(B \setminus X)).$$

*(The above expression is not exactly a monadic second-order formula. But we can easily express it in monadic second-order logic. For instance, $\mu_{3k-1}(B \setminus X)$ can be translated into $\forall Y((\forall x, x \in Y \Leftrightarrow (x \notin X \wedge x \in B)) \Rightarrow \mu_{3k-1}(Y)$.)*
*If there is no such partition of $B$, then the rank-width of $G$ is larger than $k$. Stop.*

*(B3) Run this algorithm recursively with $B = X$ and $B = Y$. If the algorithm confirms that the rank-width of $G$ is larger than $k$, then stop.*
*Otherwise, we obtain a pair $(T_X, \mathcal{L}_X)$ from $B = X$ and a pair $(T_Y, \mathcal{L}_Y)$ from $B = Y$.*

*(B4) Let $T$ be a rooted tree obtained by joining $T_X$ and $T_Y$ (creating a new root and make it adjacent to roots of $T_X$ and $T_Y$). Let $\mathcal{L}(v) = \mathcal{L}_X(v)$ if $v \in X$ and $\mathcal{L}(v) = \mathcal{L}_Y(v)$ if $v \in Y$. Output $(T, \mathcal{L})$.*

By Lemma 5.3, it is obvious that Algorithm 5.5 is correct. For (B2), we need a subroutine to search an assignment to satisfy a monadic second-order formula. The following lemma is based on [Courcelle et al. 2000, Theorem 4].

LEMMA 5.6. *Let $k$ be a constant and let $\varphi$ be a monadic second-order formula with free set variables $X$ and $B$. Let $G$ be a graph given by a $k$-expression $t$ and let $B$ be a set of vertices of $G$. Then in time $O(|t|)$, we can check whether there exists a set $X$ such that $(X, B)$ satisfies $\varphi$ in $G$ and furthermore we can output such a set $X$ if it exists.*
*(We write $|t|$ to denote the size of the $k$-expression $t$.)*

PROOF. We consider $G$ as a $k$-graph defined by $t$. (A *$k$-graph* is a vertex-labeled graph with labels in $\{1, 2, \ldots, n\}$.) First, we transform a $k$-expression $t$ into a $2k$-expression $t'$ such that in every subexpression of $t'$, labels $k+1, k+2, \ldots, 2k$ specify vertices in $B$. To find such a $2k$-expression $t'$, we transform $\cdot_i$ in $t$ to $\cdot_{i+k}$ if the vertex represented by $\cdot_i$ is in $B$ and transform $\rho_{i\to j}$ in $t$ into $\rho_{i\to j} \circ \rho_{i+k\to j+k}$ and $\eta_{i,j}$ in $t$ into $\eta_{i,j} \circ \eta_{i,j+k} \circ \eta_{i+k,j} \circ \eta_{i+k,j+k}$. Let $G'$ be a 2k-graph defined by $t'$. Thus $G$ is identical to $G'$ if we ignore labels of vertices.

By translating $x \in B$ into the condition that the label of $x$ is $k+1$, $k+2$, …, or $2k$, we transform $\varphi$ on $G$ into $\varphi'$ on $G'$ with one free variable $X$. Then $\varphi(X, B)$ holds on $G$ if and only if $\varphi'(X)$ holds on $G'$.

Now we use [Courcelle et al. 2000, Theorem 4]. Their theorem provides an $O(|t'|)$-time algorithm to find the maximum $|X|$ such that $\varphi'(X)$ is true on $G'$. In addition, their algorithm actually finds one set $X$ with the maximum size which satisfies $\varphi'$ on $G'$. (This is not clearly stated in their theorem but in their proof [Courcelle et al. 2000, line 13–14 of page 146], they store one assignment to satisfy each formula at every node during dynamic programming. We can simply read the true assignment for the top node.) We remark that taking the maximum size is not relevant in our situation. □

Let us calculate the running time of Algorithm 5.5. Let $n = |V(G)|$. We first remark that the size of the $k$-expression defining $G$ can be big but we can transform any $k$-expression into another $k$-expression of size $O(n)$. Furthermore, the $(2^{k+1}-1)$-expression obtained from a rank-decomposition of width at most $k$ always has size $O(n)$, and this is the only construction that we will use later. Thus we assume that $t$ has size $O(n)$. Then (B2) can be done in time $O(n)$ by Lemma 5.6. (B4) can be done in time $O(1)$. At (B3) we call this algorithm recursively. The total number of recursive calls is at most $n$, and therefore the running time of this algorithm is $O(n^2)$.

We note that if $B = V$ then Algorithm 5.5 will either output a rank-decomposition of $G$ of width at most $3k-1$ or confirm that the rank-width is larger than $k$. But Algorithm 5.5 requires an $l$-expression as an input. How can we provide it? One of the obvious method is to use our second algorithm in Section 4 to provide an $l$-expression for Algorithm 5.5. Then the overall running time is $O(n^3)$. This method is perfectly valid and we can deduce Theorem 5.8 and Corollary 5.9.

However, we present another method so that three main algorithms in this paper are independent of one another.

ALGORITHM 5.7. *Let $G = (V, E)$ be a graph with $V = \{v_1, v_2, \dots, v_n\}$. Let $G_i$ be the subgraph of $G$ induced by $\{v_1, v_2, \dots, v_i\}$. We may assume that $|V| \geq 2$. Let $k$ be a fixed positive integer.*

*Input. A graph $G$.*

*Output. Either outputs a rank-decomposition of width at most $3k-1$ or certifies that the rank-width of $G$ is larger than $k$.*

*(A1) Let $i = 2$ and let $(T_2, \mathcal{L}_2)$ be a rank-decomposition of $G_2$, so that $T_2$ is a tree with two vertices and $\mathcal{L}_2$ is a bijection from $\{v_1, v_2\}$ to leaves of $T_2$.*

*(A2) Increase $i$ by 1.*
*We assume that we have a rank-decomposition $(T_{i-1}, \mathcal{L}_{i-1})$ of $G_{i-1}$ of width at most $3k-1$.*

*(A3) Let $T_i'$ be a tree obtained from $T_{i-1}$ by subdividing an arbitrary edge of $T_{i-1}$ and attaching a leaf. We let $\mathcal{L}_i'(v_i)$ be the new leaf and $\mathcal{L}_i'(v_j) = \mathcal{L}_{i-1}(v_j)$ for all $j < i$.*
*Then $(T_i', \mathcal{L}_i')$ is a rank-decomposition of $G_i$ of width at most $3k$.*

*(A4) By using Proposition 2.2, we obtain a $(2^{3k+1}-1)$-expression $t_i$ of $G_i$.*

*(A5) We now apply Algorithm 5.5 to $G_i$ with $l = 2^{3k+1}-1$, $t_i$ and $B = \{v_1, v_2, \dots, v_i\}$. It will either output a rank-decomposition $(T_i, \mathcal{L}_i)$ (of $G_i$) of width at most $3k-1$ or confirm that the rank-width of $G_i$ is larger than $k$.*
*If it confirms that $\mathrm{rw}(G_i) > k$, which implies $rwd(G) > k$, then stop.*

*(A6) If $i < n$ then go to (A2).*

*(A7) $(T_n, \mathcal{L}_n)$ is a rank-decomposition of $G$ of width at most $3k-1$.*

In short, we obtain the rank-decomposition of $G_i$ by using dynamic programming based on a rank-decomposition of $G_{i-1}$. It is clear that Algorithm 5.7 is correct. Let us analyze the running time. Let $n = |V(G)|$. (A3) takes time $O(1)$. (A4) takes time $O(n^2)$ by Proposition 2.2. (A5) takes time $O(n^2)$ because the running time of Algorithm 5.5 is $O(n^2)$. Since the number of iterations is $O(n)$, we conclude that the running time of Algorithm 5.7 is $O(n^3)$. Therefore we deduce the following theorem.

THEOREM 5.8. *Let $k$ be a fixed positive integer. There is an $O(|V(G)|^3)$-time algorithm that either outputs a rank-decomposition (of an input graph $G$) of width at most $3k-1$ or confirms that the rank-width of $G$ is larger than $k$.*

By Proposition 2.2, we obtain the following corollary.

COROLLARY 5.9. *Let $k$ be a fixed positive integer. There is an $O(|V(G)|^3)$-time algorithm that either outputs an $(8^k-1)$-expression of an input graph $G$ or confirms that the clique-width of $G$ is larger than $k$.*

## 6. DISCUSSIONS

Many applications of clique-width are polynomial-time algorithms to solve graph problems when inputs are restricted to graphs of bounded clique-width. Most of them ([Wanke 1994; Courcelle et al. 2000; Espelage et al. 2001; Gerber and Kobler 2003; Kobler and Rotics 2003]) require $k$-expression of the input graph as an input to take an advantage of tree-structures. But by using [Oum and Seymour 2006], we do not need $k$-expressions as an explicit input, because we can generate a $(2^{1+f(k)}-1)$-expression in polynomial time and provide it as an input. The result of this paper will make this preprocessing much faster.

Courcelle and Oum [2007] show that there is an $O(|V|^9 \log |V|)$-time algorithm that recognizes graphs of rank-width at most $k$ for an input graph $G = (V, E)$ and a fixed $k$; they use an approximation algorithm by Oum and Seymour [2006] as a first step, and it is the slowest part of their algorithm. By the result of this paper, we obtain the following.

THEOREM 6.1. *For fixed $k$, there is an $O(n^3)$-time algorithm to decide whether an $n$-vertex graph has rank-width at most $k$.*

However, this algorithm does not produce a rank-decomposition of width at most $k$ even if it says such a decomposition exists. Recently, Hliněný and Oum [2008] solved this problem; they found a cubic-time algorithm to, for fixed $k$, provide a rank-decomposition of width at most $k$ if it exists.

### ACKNOWLEDGMENTS

The author would like to thank Paul Seymour, Jim Geelen, Petr Hliněný, and Bruno Courcelle for valuable discussions. In particular, Jim Geelen suggested the use of blocking sequences and Bruno Courcelle kindly provided many suggestions, especially on Section 5.

### REFERENCES

BOUCHET, A. 1988. Graphic presentations of isotropic systems. *J. Combin. Theory Ser. B 45,* 1, 58–76.

Bouchet, A. 1989. Connectivity of isotropic systems. In *Combinatorial Mathematics: Proceedings of the Third International Conference (New York, 1985)*. Ann. New York Acad. Sci., vol. 555. New York Acad. Sci., New York, 81–93.

Bouchet, A. 1990. $\kappa$-transformations, local complementations and switching. In *Cycles and rays (Montreal, PQ, 1987)*. NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., vol. 301. Kluwer Acad. Publ., Dordrecht, 41–50.

Bouchet, A. 1994. Circle graph obstructions. *J. Combin. Theory Ser. B 60,* 1, 107–144.

Corneil, D. G., Habib, M., Lanlignel, J.-M., Reed, B., and Rotics, U. 2000. Polynomial time recognition of clique-width $\leq 3$ graphs (extended abstract). In *Gonnet, Gastón H. (ed.) et al., LATIN 2000: Theoretical informatics. 4th Latin American symposium, Punta del Este, Uruguay, April 10-14, 2000.* Lecture Notes in Comput. Sci., vol. 1776. Springer, Berlin, 126–134.

Courcelle, B. 1997. The expression of graph properties and graph transformations in monadic second-order logic. In *Handbook of graph grammars and computing by graph transformation, Vol. 1.* World Sci. Publishing, River Edge, NJ, 313–400.

Courcelle, B. 2006. The monadic second-order logic of graphs XV: On a conjecture by D. Seese. *J. Appl. Log. 4,* 1, 79–114.

Courcelle, B., Makowsky, J. A., and Rotics, U. 2000. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst. 33,* 2, 125–150.

Courcelle, B. and Olariu, S. 2000. Upper bounds to the clique width of graphs. *Discrete Appl. Math. 101,* 1-3, 77–114.

Courcelle, B. and Oum, S. 2007. Vertex-minors, monadic second-order logic, and a conjecture by Seese. *J. Combin. Theory Ser. B 97,* 1, 91–126.

Espelage, W., Gurski, F., and Wanke, E. 2001. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Graph-theoretic concepts in computer science (Boltenhagen, 2001)*. Lecture Notes in Comput. Sci., vol. 2204. Springer, Berlin, 117–128.

Fellows, M. R., Rosamond, F. A., Rotics, U., and Szeider, S. 2006. Clique-width minimization is NP-hard. In *Proceedings of the 38th annual ACM Symposium on Theory of Computing.* ACM Press New York, NY, USA, 354–362.

Fomin, F. V., Fraigniaud, P., and Thilikos, D. M. 2004. The price of connectedness in expansions. Tech. Rep. 273, Department of Informatics, University of Bergen, Bergen, Norway. May. Presented at SIAM Conference on Discrete Mathematics, Victoria, British Columbia, Canada, June 2006.

Geelen, J. F. 1995. Matchings, matroids and unimodular matrices. Ph.D. thesis, University of Waterloo.

Gerber, M. U. and Kobler, D. 2003. Algorithms for vertex-partitioning problems on graphs with fixed clique-width. *Theoret. Comput. Sci. 299,* 1-3, 719–734.

Hicks, I. V. and McMurray Jr., N. B. 2007. The branchwidth of graphs and their cycle matroids. *J. Combin. Theory Ser. B 97,* 5, 681–692.

Hliněný, P. 2005. A parametrized algorithm for matroid branch-width. *SIAM J. Comput. 35,* 2, 259–277, loose erratum (electronic).

Hliněný, P. and Oum, S. 2008. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*. Accepted.

Iwata, S., Fleischer, L., and Fujishige, S. 2001. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM) 48,* 4, 761–777.

Kobler, D. and Rotics, U. 2003. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Appl. Math. 126,* 2-3, 197–221.

Mazoit, F. and Thomassé, S. 2005. Branchwidth of graphic matroids. Manuscript.

Oum, S. 2005a. Graphs of bounded rank-width. Ph.D. thesis, Princeton University.

Oum, S. 2005b. Rank-width and vertex-minors. *J. Combin. Theory Ser. B 95,* 1, 79–100.

Oum, S. and Seymour, P. 2006. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B 96,* 4, 514–528.

OXLEY, J. G. 1992. *Matroid theory.* Oxford University Press, New York.

ROBERTSON, N. AND SEYMOUR, P. 1991. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B 52,* 2, 153–190.

SEESE, D. 1991. The structure of the models of decidable monadic theories of graphs. *Ann. Pure Appl. Logic 53,* 2, 169–195.

SEYMOUR, P. AND THOMAS, R. 1994. Call routing and the ratcatcher. *Combinatorica 14,* 2, 217–241.

WANKE, E. 1994. $k$-NLC graphs and polynomial algorithms. *Discrete Appl. Math. 54,* 2-3, 251–266.