Approximation algorithm for the Clique-width

Sang-il Oum
Applied & Computational Math.
Princeton Univ.
sangil@princeton.edu

Dec, 2003

Abstract

$O(n^9 \log n)$-time algorithm to output either clique-width $> k$ or $\leq f(k)$, where $f(k)$ is independent of n.

Cowork with Paul Seymour.

Clique-width

Definition 1. *[Courcelle and Olariu, 2000]*

k-expression: expression on vertex-labelled 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 $\mathrm{cwd}(G)$: minimum k such that G can be expressed by k-expression (after forgetting the labels)

Clique-width and Algorithms

For graphs of clique-width $\leq k$, if an input is given by its k-expression, then many NP-complete problems can be solved in polynomial time, assuming k is a constant.

- All graph properties, expressible in monadic second order logic with quantifications over vertices and vertex sets [Courcelle et al., 2000] (a logic formula with $\neg$, $\vee$, $\wedge$, (,), $x = y$, $x \sim y$, $x \in X$, $\forall x$, $\exists y$, $\forall X$, $\exists Y$)
- Hamiltoian path/circuit [Espelage et al., 2001], [Wanke, 1994]
- Finding the chromatic number [Kobler and Rotics, 2003]

If we don't have a k-expression,

Suppose our input graphs have clique-width ≤ 10, but inputs are given by its adjacency list. How to constuct a 10-expression of an input graph?

It's open for $k > 3$ whether there exists a poly-time algorithm to find a k-expression assuming $cwd(G) \leq k$.
$k = 3$: [Corneil et al., 2000]
$k = 2$: [Corneil et al., 1985]

Any algorithms that guarantee to find a $f(k)$-expression also make algorithms based on k-expressions run in poly time, because $f(k)$ is independent of n.

Overview

Instead of clique-width, we develoved the techniques for branch-width of a symmetric submodular functions, and apply it to some function on graphs to get the 'rank-width'.

- Rank-width and clique-width are compatible: if one is bounded, another is also bounded. rank-width $\leq$ clique-width $\leq 2^{\text{rank-width}+1}$

- For fixed k, $\exists$ $O(n^9 \log n)$-time algorithm, which confirms rank-width$> k$ or outputs a rank-decomposition of width $\leq 3k+1$.

- We have a $O(n)$-time algorithm to convert the rank-decomposition of width $\leq 3k+1$ into 2^{3k+2}-expression.

Branch-width of a symmetric submodular function

Let $f : V \to \mathbb{Z}$ be s.t. $f(X) = f(V - X)$, $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$, $f(\{v\}) - f(\emptyset) \leq 1 \ \forall v$. Assume $f(\emptyset) = 0$.

Definition 2. *[Geelen et al., 2002]*

Branch-decompositon of f: cubic tree T with a bijection between leaf nodes of T and V

Width of T: $\max_{e \in T} f(A_e)$ where (A_e, B_e) is a partition of V induced by $e \in T$

Branch-width of f, denoted by $bw(f)$: minimum width over all possible branch-decomposition of f

Well-Linkedness and Branch-Width

Definition 3. *$A \subseteq V$ is called well-linked iff for any partition (X,Y) of A,*

$$X \subseteq Z \subseteq V \setminus Y \quad \Rightarrow \quad f(Z) \geq \min(|X|,|Y|).$$

Theorem 1. *1. If f has a well-linked set A of size k, then $bw(f) \geq k/3$.*

2. If f has no well-linked set of size k, then $bw(f) \leq k$; $\exists$ a poly-time algorithm that constructs the branch-decomp. of width $\leq k$ or finds a well-linked set of size k.

$\Rightarrow$ poly-time algorithm to confirm $bw(f) > k$ or $bw(f) \leq 3k+1$ and output its branch-decomposition of width $\leq 3k+1$.

Rank-width

Definition 4. *Let G be a simple graph.* $f^*_G(A,B) = \mathrm{rank}(M^B_A)$, *where M^B_A is a 0-1 A-by-B matrix $(m_{ij})_{i\in A, j\in B}$ such that*

$$m_{ij} = \begin{cases} 1 & \text{if } i \text{ is adjacent to } j \text{ in } G \\ 0 & \text{otherwise.} \end{cases}$$

Let $f_G(X) = f^*_G(X, V-X)$ "rank of a cut".

Proposition 1. *f^*_G is symmetric, uniform and submodular. f_G is symmetric and submodular.*

Definition 5. *Rank-decomposition of $G \equiv$ branch-decomposition of f_G.*
Rank-width of $G \equiv$ branch-width of f_G.

Rank-width and Clique-width

Proposition 2.

$$\textit{rank-width} \leq \textit{clique-width} \leq 2^{\textit{rank-width}+1}$$

Proof. (Idea) If M has at most k distinct rows, then $\mathrm{rank}(M) \leq k$. Conversely, $\mathrm{rank}(M) \leq k$ implies M has at most 2^k distinct rows/columns, if M is a 0-1 matrix. $\square$

Time Complexities

- Calculating f_G^*: $O(n^3)$ time
- Converting rank-decomposition of width $\leq k$ into 2^{k+1}-expression: $O(n)$ time.

What is f^* in general?

We need a general f^* to apply our algorithm to general f other than f_G. Let $3^V = \{(A,B) : A \cap B = \emptyset, A, B \subseteq V\}$.

Definition 6. *$f^* : 3^V \to \mathbb{Z}$ is an extension of a submodular function $f : V \to \mathbb{Z}$ iff*

1. $f^*(X, V - X) = f(X)$ *for all* $X \subseteq V$,
2. *(uniform) if* $A \subseteq C$, $B \subseteq D$, *then* $f(A,B) \leq f(C,D)$,
3. *(submodular)* $f^*(A,B) + f^*(C,D) \geq f^*(A \cap C, B \cup D) + f^*(A \cup C, B \cap D)$.

If we fix B, then $f^*(X,B)$ is a rank function on a matroid over $V - B$.

What is f^*? — continued

There is at least one extension of f.

Proposition 3. $f_{\min}(A,B) = \min_{A \subseteq Z \subseteq (V-B)} f(Z)$ *is an extension of* f*.*

Fact: f_G^* is an extension of f_G.

For each problem, we can choose the most convenient f^* to reduce the running time. For instance, calculating $(f_G)_{\min}$ is much slower than calculating f_G^*.

Time Complexity when f^* is given

Suppose we have a function f^*, whose running time is $O(\gamma)$.

We use the submodular function minimization algorithm by [Iwata et al., 2001], whose runnning time is $O(n^5 \delta \log M)$. M is the maximum value of the submodular function and δ is the running time of the submodular function.

Job	Time
Find a basis	$O(n\gamma)$
Find Z	$O(2^{k-1}(n^5 \gamma \log n))$

$$O(n(n\gamma + 2^{k-1} n^5 \gamma \log n)) = O(n^6 \gamma \log n).$$

For rank-width: $\gamma = O(n^3) \Rightarrow O(n^9 \log n)$.

Time Complexity when f is given

Suppose we have a function f, whose running time is $O(\gamma)$. Let's use $f_{\min}$ as an extension of f. We can calculate $f_{\min}$ by the submodular function minimization algorithm.

Job	Time
Find a basis	$O(n \cdot n^5 \gamma \log n)$
Find Z	$O(2^{k-1}(n^5 \gamma \log n))$

$$O(n(n^6\gamma\log n + 2^{k-1}n^5\gamma\log n)) = O(n^7\gamma\log n).$$

Branch-width of a matroid

Let M be a matroid with the rank function r. $\lambda(X) = r(X) + r(E - X) - r(M) + 1$ is a connectivity function.

Definition 7. *Branch-width of $M \equiv$ branch-width of λ.*

Note that $\lambda(\emptyset) = 1$. So there's a small adjustment.

Corollary 1. *For given k, there is an algorithm using the rank oracle to output $bw(M) > k$ or output a branch-decomposition of order $\leq 3k - 1$, and its running time and number of oracle calls is at most $O(n^7 \log n)$.*

Other aspects

Proposition 4. *For fixed k, deciding $bw(f) \leq k$ is in NP$\cap$co-NP.*

Proof. To achieve co-NP, use tangles [Robertson and Seymour, 1991], [Geelen et al., 2003] $\square$

Let $W(G)$ be size of the largest well-linked set w.r.t. f_G. By the theorem 1, $W(G)$ is compatible with clique-width and rank-width. Assume rank is calculated over $\mathbb{Z}_2$.

Proposition 5. *For fixed k, $W(G) \leq k$ can be decided in $O(n^9 \log n)$.*

Proof. $W(G) \leq k$ is expressible by monadic second order logic. $\square$

Summary

- $\exists$ well-linked set of size $k \Rightarrow bw(f) \geq \frac{k}{3} + b$, $\not\exists$ well-linked set of size $k \Rightarrow bw(f) \leq k + b$, if $b = f(\emptyset)$.

- Fixed-parameter-tractable algorithm that confirms $bw(f) > k$ or outputs a branch-decomposition of width $\leq 3k + 1 - 2f(\emptyset)$, if f is symmetric submodular and $f(\{v\}) - f(\emptyset) \leq 1$. $\Longrightarrow$ can be applied to branch-width of a matroid and rank-width

- Rank-width $bw(f_G)$ is compatible with clique-width. Futhermore, there is a $O(n)$ algorithm to convert the branch-decomposition of width $\leq k$ into a 2^{k+1}-expression.

Proof of Theorem 1

Proof. 1. Suppose T is a branch decomposition of f. Then, there exists $e \in E(T)$ such that $|A_e \cap A| \geq k/3$ and $|B_e \cap A| \geq k/3$. Therefore, $f(A_e) \geq \min(|A_e \cap A|, |B_e \cap A|) \geq k/3$. $bw(f) \geq k/3$.

2. Greedy algorithm works. Let $B \subseteq V$ be such that we want a 'partial' branch-decomp. of B of width $\leq k$, which is a rooted binary tree.

If $f(B) < k$, move one vertex of B into $V - B$, and run this algorithm. Join the return with v. $f(B) \leq f(B - \{v\}) + f(\{v\}) \leq k$.

Say $f(B) = k$. Let $A = V - B$. Find a basis $X \subseteq A$ s.t. $|X| = f^*(X, B) = k$. X is not well-linked, so find Z such that

$$f(Z) < \min(|Z \cap X|, |(V-Z) \cap X|).$$

Want to split B into $Z \cap B$ and $(V-Z) \cap B$.

$Z \cap B \neq \emptyset$ unless $f(Z) \geq f^*(Z \cap X, B) = |Z \cap X|$. Similarly $(V-Z) \cap B \neq \emptyset$.

$$\begin{aligned}
&|(V-Z) \cap X| + f(B) \\
&> f(Z) + f(B) \geq f(Z \cup B) + f(Z \cap B) \\
&\geq f^*((V-Z) \cap X, B) + f(Z \cap B) \\
&= |(V-Z) \cap X| + f(Z \cap B)
\end{aligned}$$

$f(Z \cap B) < f(B)$ and similarly $f((V-Z) \cap B) < f(B)$.

Run for $B \leftarrow Z \cap B$ and $B \leftarrow (V-Z) \cap B$, and join two returns. $\square$

References

[Corneil et al., 2000] Corneil, D. G., Habib, M., Lanlignel, J.-M., Reed, B., and Rotics, U. (2000). Polynomial time recognition of clique-width ≤ 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. Proceedings. Berlin: Springer. Lect. Notes Comput. Sci. 1776, 126-134* .

[Corneil et al., 1985] Corneil, D. G., Perl, Y., and Stewart, L. K. (1985). A linear recognition algorithm for cographs. *SIAM J. Comput.*, 14(4):926–934.

[Courcelle et al., 2000] 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 and Olariu, 2000] Courcelle, B. and Olariu, S. (2000). Upper bounds to the clique width of graphs. *Discrete Appl. Math.*, 101(1-3):77–114.

[Espelage et al., 2001] 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)*, volume 2204 of *Lecture Notes in Comput. Sci.*, pages 117–128. Springer, Berlin.

[Geelen et al., 2003] Geelen, J. F., Gerards, A. M. H., Robertson, N., and Whittle, G. (2003). Obstructions to branch-decomposition of matroids. manuscript.

[Geelen et al., 2002] Geelen, J. F., Gerards, A. M. H., and Whittle, G. (2002). Branch-width and well-quasi-ordering in matroids and graphs. *J. Combin. Theory Ser. B*, 84(2):270–290.

[Iwata et al., 2001] 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 and Rotics, 2003] 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.

[Robertson and Seymour, 1991] Robertson, N. and Seymour, P. D. (1991). Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190.

[Wanke, 1994] Wanke, E. (1994). k-NLC graphs and polynomial algorithms. *Discrete Appl. Math.*, 54(2-3):251–266. Efficient algorithms and partial k-trees.