Sampling and Counting Acyclic Orientations in Chordal Graphs
(Student Abstract)

Wenbo Sun
Rochester Institute of Technology
Rochester, NY, U.S.A.
ws3109@rit.edu

Abstract
Sampling and counting of different types of acyclic orientations over chordal graphs are central tasks in several AI applications such as causal structure learning. For a given undirected graph, an acyclic orientation is an assignment of directions to all of its edges which makes the resulting directed graph cycle-free. Sampling is often closely related to the corresponding counting problem. Counting of acyclic orientations of a given chordal graph can be done in polynomial time, but the previously known techniques do not seem to lead to a corresponding (efficient) sampler. In this work, we propose a dynamic programming framework which yields a counter and a uniform sampler, both of which run in (essentially) linear time. An interesting feature of our sampler is that it is a stand-alone algorithm that, unlike other DP-based samplers, does not need any preprocessing which determines the corresponding counts.

Preliminaries: Chordal Graphs, Clique Trees
For a graph $G$, we use $G[U]$ to denote the graph induced in $G$ on the vertex set $U \subseteq V(G)$. Every chordal graph $G$ can be represented by a clique tree $T_G$ where $V(T_G)$ is the set of maximal cliques of $G$ and the tree satisfies the induced subtree property: For every vertex $v \in V(G)$, the induced subgraph $T_G[A_v]$ is connected, where $A_v$ is the set of maximal cliques of $G$ containing $v$. Let $T_{G,C}$ be the clique tree $T_G$ rooted at a maximal clique $C$. If $G$ is clear from the context, we will simply write $T_C$. We denote by $T_{C,C} \subseteq T_{C,C}$ the subtree of $T_{C,C}$ containing $C$ and its descendants; we write $T_{C}$ if $C$ is clear from the context.

Each clique $C$ in $T_C$ can be partitioned into a separator set $\text{Sep}(C) = C \cap \text{Parent}(C)$ and a residual set $\text{Res}(C) = C \setminus \text{Sep}(C)$, where $\text{Parent}(C)$ is the parent clique of $C$ in $T_C$. If $C = C_r$, then $\text{Parent}(C) = \emptyset$. The following properties hold:

- For each vertex $v$ in $G$, there is a unique clique $C_v$ that contains $v$ in its residual set. This implies that $|V(T_G)| \leq |V(G)|$ and that $C_v$ is the root of $T_{C_v}[A_v]$; we denote this rooted subtree by $T_{C_v}$. All other cliques in $T_{C_v}$ that contain $v$ have it in their separator set.
- For a clique $C$ let $D(C)$ be the set of vertices in the descendant cliques of $C$ in $T_{C_v}$ except $\text{Sep}(C)$, i.e., $D(C) := \cup_{C' \in V(T_C)} C' - \text{Sep}(C)$. Let $A(C)$ be the vertices in the cliques in $T_C$ except $\text{Sep}(C)$, i.e., $A(C) := \cup_{C \in V(T_C)} C - \text{Sep}(C)$. The separator $\text{Sep}(C)$ separates $A(C)$ and $D(C)$ in $G$; there is no edge with one endpoint in $A(C)$ and the other in $D(C)$.
- Construction of a clique tree for a connected chordal graph can be done in time $O(|E(G)|)$.

We use $G[T_C]$ for the subgraph induced by the vertices that belong to cliques in $T_C$, i.e., $G[T_C] := G[\cup_{C \in V(T_C)} C']$.
We also define the following subgraph of $G [T_C ]$: Let $\hat{G} [T_C ]$ be $G [T_C ]$ with the edges within the separator set $\text{Sep}(C)$ removed, i.e., $\hat{G} [T_C ] := G [T_C ] - E(G[\text{Sep}(C)])$.

**Our Contribution**

The main contribution of our work is summarized in Theorem 3 and in Algorithm 1, which generates a uniformly random acyclic orientation for the given chordal graph. In other words, each orientation is generated with probability $\frac{1}{|\Omega|}$, where $\Omega$ is the set of all acyclic orientations of the graph. The proof of Theorem 3 relies on the following two lemmas (their proofs, as well as the full proof of the theorem, are available upon request, and will be included in the author’s thesis).

**Lemma 1.** Let $C$ be a clique in the rooted clique tree $T$ and let $C_1, C_2, \ldots, C_d$ be its children cliques. The edge sets of the graphs $\hat{G} [T_{C_i}], i = 1, \ldots, d$, are mutually disjoint.

**Lemma 2.** Let $G$ be a connected chordal graph and let $T$ be a rooted clique tree of $G$. For a clique $C$ in $T$ and an acyclic orientation $\sigma$ over $C$, let $\text{AO} (T_C, \sigma)$ be the set of acyclic orientations of $G [T_C ]$ that are consistent with $\sigma$. For any $C$ and any two acyclic orientations $\sigma_1$ and $\sigma_2$ over $C$, we have
\[
|\text{AO} (T_C, \sigma_1)| = |\text{AO} (T_C, \sigma_2)|.
\]

In order to make the running times of our algorithms more readable, we assume that each arithmetic operation takes a constant time. This is, of course, a bit optimistic, since the ultimate number of orientations can be as high as $2^m$ for a graph with $m$ edges, and, therefore, the true running time of each arithmetic operation adds a factor of about $m \log(m)$. We use $\mathcal{O}(\cdot)$ notation to indicate that this factor is omitted from our running time estimate.

**Theorem 3.** Let $G$ be a connected chordal graph. The number of its acyclic orientations can be calculated in $\mathcal{O}(|V(G)| + \mathcal{O}(|E(G)|))$ time.

**Proof sketch.** Let $T$ be a clique tree of $G$ rooted at a clique $C_r$. For a clique $C$ in $T$, we define $\text{AO} (T_C)$ as the number of acyclic orientations of $G [T_C ]$ under the assumption that the orientation of the edges of $G[\text{Sep}(C)]$ has been fixed. Then, $\text{AO} (T_{C_r})$ computes the overall number of acyclic orientations of $G$, since $\text{Sep}(C_r) = \emptyset$. We show how to compute $\text{AO} (T_C)$ by dynamic programming over the clique tree:
\[
\text{AO} (T_C) = \frac{|C|!}{|\text{Sep}(C)|!} \prod_{C_i \in \text{Sep}(C)} \text{AO} (T_{C_i}),
\]
where $C_1, \ldots, C_d$ are the children cliques of $C$ in $T$ (and $d = 0$ if $C$ is a leaf of $T$). Let $\sigma_{\text{Sep}(C)}$ be the given orientation of $G [\text{Sep}(C)]$, we extend it to an acyclic orientation $\sigma_C$ over $C$ by picking an orientation from the $\frac{|C|!}{|\text{Sep}(C)|!}$ candidates. The calculation is obviously correct when $C$ is a leaf. When $C$ is a non-leaf, we fix an arbitrary $\sigma_C$ and let $\sigma_{\text{Sep}(C_i)}$ be the orientation restricted to $G [\text{Sep}(C_i)]$, and we use $A_i$ to denote the set of acyclic orientations of $G [T_{C_i}]$ consistent with $\sigma_{\text{Sep}(C_i)}$. By Lemma 1, we can prove that there is a bijection between $\text{AO} (T_C, \sigma_C)$ and $A_1 \times \cdots \times A_d$. Then, by the inductive hypothesis, the correct calculation of $|\text{AO} (T_{C_i})|$ yields that $\prod_{C_i \in \text{Sep}(C)} \text{AO} (T_{C_i}) = |\text{AO} (T_C, \sigma_C)|$, and by Lemma 2 we know that $|\text{AO} (T_C, \sigma_C)|$ does not depend on the specific orientation of $\sigma_C$, and there are $|C|!$ possible $\sigma_C$’s, yielding the expression 1.

The construction of $T$ costs $O(|V(G)|)$, and each clique $C$ of $T$ is processed exactly once and it costs $O(\deg_{T}(C) + 1)$. Hence the running time is $\mathcal{O}(|E(G)|)$.

The proof of Theorem 3 naturally yields a uniform sampler of acyclic orientations (see Algorithm 1), which runs in time $\mathcal{O}(|E(G)|)$ for any input chordal graph $G$.

**Future Plans**

We view this work as a first step in the direction of counting and sampling different types of graph orientations on chordal graphs, such as bipolar orientations, sink-free orientations, and strong orientations.

**References**

Ganian, R.; Hamm, T.; and Talvitie, T. 2020. An Efficient Algorithm for Counting Markov Equivalent DAGs. In AAAI, 10136–10143.

Ghassami, A.; Salehkaleybar, S.; Kiyavash, N.; and Zhang, K. 2019. Counting and sampling from Markov equivalent DAGs using clique trees. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 3664–3671.

Naor, J.; Naor, M.; and Schäffer, A. A. 1989. Fast parallel algorithms for chordal graphs. SIAM Journal on Computing, 18(2): 327–349.

Stanley, R. P. 1973. Acyclic orientations of graphs. Discrete Mathematics, 5(2): 171–178.

Talvitie, T.; and Koivisto, M. 2019. Counting and sampling Markov equivalent directed acyclic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 7984–7991.

Wienöbst, M.; Bannach, M.; and Liskiewicz, M. 2020. Polynomial-Time Algorithms for Counting and Sampling Markov Equivalent DAGs. arXiv preprint arXiv:2012.09679.