On Listing, Sampling, and Counting the Chordal Graphs with Edge Constraints

Shuji Kijima[1] Masashi Kiyomi[2] Yoshio Okamoto[3] Takeaki Uno[4]

[1] Research Institute for Mathematical Sciences,
Kyoto University, Kyoto, 606-8502, Japan.
kijima@kurims.kyoto-u.ac.jp

[2] Department of Information Processing, School of Information Science,
Japan Advanced Institute of Science and Technology, Ishikawa, 923-1292, Japan.
mkiyomi@jaist.ac.jp

[3] Department of Information and Computer Sciences,
Toyohashi University of Technology, Aichi, 441-8580, Japan.
okamotoy@ics.tut.ac.jp

[4] National Institute of Informatics, Tokyo, 101-8430, Japan.
uno@nii.ac.jp

October 15, 2007

Abstract

We discuss the problems to list, sample, and count the chordal graphs with edge constraints. The edge constraints are given as a pair of graphs one of which contains the other and one of which is chordal, and the objects we look at are the chordal graphs contained in one and containing the other. This setting is a natural generalization of chordal completions and deletions. For the listing problem, we give an efficient algorithm running in amortized polynomial time per output with polynomial space. For the sampling problem, we give an instance for which a natural Markov chain suffers from an exponential mixing time. For the counting problem, we show some #P-completeness results. These results provide a unified viewpoint from algorithms theory to problems arising from various areas such as statistics, data mining, and numerical computation.

key words: *graph sandwich, chordal completion/deletion, enumeration, Markov chain Monte Carlo, #P-completeness.*

1 Introduction

A graph is *chordal* if it has no induced cycle of length more than three. The class of chordal graphs often appears as a tractable case of a lot of problems arising from various areas such as statistics, optimization, numerical computation, etc. In those areas, we often approximate a given graph by a chordal graph and then apply efficient algorithms for chordal graphs to the obtained graph. Evaluation criteria for chordal approximations depend on applications. For example, in the context of graphical modeling in statistics, a chordal approximation is desired to minimize AIC (Akaike's Information Criterion), BIC (Bayesian Information Criterion), MDL (Minimum Description Length), etc. [19, 25, 30]; in the context of numerical computation, a chordal approximation is desired to minimize the number of added edges (a.k.a. the minimum fill-in problem) [21, 22, 28, 5]; in the context of discrete optimization, a chordal approximation is desired to minimize the size of a largest clique (a.k.a. the treewidth problem) [20, 13, 14, 4].

Since we are concerned with various sorts of criteria and often these computational problems are NP-hard, listing algorithms and random-sampling algorithms can be useful universal decision-support schemes. An exhaustive list found by an algorithm may provide an exact solution, whereas random samples may provide an approximative solution. Our goal is to provide efficient algorithms for listing problems and random-sampling problems of graphs, or to show the intractability of the problems.

As a chordal approximation, we consider the following two types of changes; either we just insert some edges or we just delete some edges to make a given graph G chordal. A result of the former operation is called a *chordal completion* of G, and a result of the latter operation is called a *chordal deletion* of G. As a computational problem, given a graph G we want to deal with all chordal completions of G or all chordal deletions of G.

However, in fact, we study a generalized problem of this kind. Namely, we are given two graphs $\overline{G}$ and $\underline{G}$ on the same vertex set such that $\underline{G}$ is contained in $\overline{G}$ and one of them is chordal, we want to deal with all chordal graphs that contain $\underline{G}$ and are contained in $\overline{G}$. When $\overline{G}$ is chordal, this problem generalizes the problem on chordal completions (since a complete graph is chordal), and when $\underline{G}$ is chordal, this problem generalizes the problem on chordal deletions (since an empty graph is chordal).

There are (at least) two reasons why we study this generalized problem. The first one is clear: this is more general. The second one comes from a more practical aspect. Since the number of chordal completions of a graph can be quite huge, it would be difficult and even impossible in most of the cases to run a listing algorithm to obtain the exhaustive list of the chordal completions. Also for random sampling, if the size of our sample space is quite large then the probability of needling a desired object will be pretty small. Indeed, as Wormald [31] showed, the number of chordal graphs with n vertices is asymptotically $\sum_{r=0}^{n}\binom{n}{r}2^{r(n-r)}$, which is roughly $2^{n^2/4+O(n\log n)}$. Thus, dealing with all chordal graphs is impractical, and a simple and natural way to narrow down the size of our list is to introduce a way to filter out some undesired candidates from the list, and a way to find a "suitable" chordal approximation in a more flexible manner when combined with several heuristics or local search strategies.

Results We provide an efficient listing algorithm to enumerate all chordal graphs containing a given $\underline{G}$ and contained in a given $\overline{G}$ when $\overline{G}$ or $\underline{G}$ is chordal. The running time is polynomial in the input size per output, and the memory usage is bounded by a polynomial in the input size. The algorithm is based on a binary partition method, which is much simpler than the previous algorithms by Kiyomi and Uno [11] to list all chordal deletions and by Kiyomi, Kijima, and Uno [12] to list all chordal completions. Note also that these previous algorithms are not able to deal with our generalized problems.

As for the random sampling, we give two clues that seem to imply that a random sampling is not easy. The first clue is that counting the chordal graphs containing $\underline{G}$ and contained in $\overline{G}$ is #P-complete, even when $\underline{G}$ is chordal. The proof is done by a parsimonious reduction from the forest counting in a graph. To the best of our knowledge, this is the first result on #P-hardness for the graph sandwich problems. We also show that counting the chordal deletions is #P-complete by a Cook reduction from the forest counting. These results imply that a simple binary partition method does not yield a polynomial-time sampling algorithm. The second clue is the following. We apply the Markov chain Monte Carlo (MCMC) method to our problem. The MCMC is a promising approach for an efficient random sampling from a family of objects that is hard to count. We show that a simple and natural Markov chain suffers from slow mixing time; namely, we give an example for which the mixing time of the Markov chain is exponential.

Related work Our generalized concept is actually a special case of the framework proposed by Golumbic, Kaplan, and Shamir [7] who studied the following *graph sandwich problem*: for a graph property Γ, we are given a pair of graphs $\overline{G}$ and $\underline{G}$ such that $\underline{G}$ is a subgraph of $\overline{G}$, and we are asked to decide if there exists a graph $G \in \Gamma$ that is a supergraph of $\underline{G}$ and at the same time a subgraph of $\overline{G}$. Golumbic, Kaplan, and Shamir [7] proved that graph sandwich problems are NP-complete for many graph properties, e.g., chordal graphs, perfect graphs, interval graphs, etc. As discussed above, for listing problems, Kiyomi and Uno [11] proposed algorithms to list the chordal deletions within constant time delay, and Kiyomi, Kijima, and Uno [12] proposed listing algorithms to list the chordal completions within a polynomial time delay. They are both based on the reverse search technique by Avis and Fukuda [1]. As for the counting problem, we are aware of the paper by Wormald [31] that gives an asymptotic number of the chordal graphs with n vertices. However, as far as graph sandwiches are concerned, neither algorithmic results nor hardness results seem to be known. There has been no result about random sampling of a chordal graph, as far as we see. For the related minimum chordal completion/deletion problems, both of which are well-known to be NP-hard [29, 18], there are some results on polynomial-time approximation, fixed parameter tractability, and exponential-time exact algorithms [17, 16, 4].

2 Preliminaries

All graphs in this paper are undirected and simple. For a graph G, we denote the set of vertices of G by $V(G)$ and the set of edges of G by $E(G)$. For a pair of graphs G and H on a common vertex set V, we write $G \subset H$ (and $G \subseteq H$) when $E(G) \subset E(H)$ (and $E(G) \subseteq E(H)$, respectively). For a graph $G = (V, E)$ and a pair of vertices $e = \{v_1, v_2\} \in (\binom{V}{2} \setminus E)$, we denote the graph $(V, E \cup \{e\})$ by $G + e$. Similarly, for a graph $G = (V, E)$ and an edge $e \in E$ we denote the graph $(V, E \setminus \{e\})$ by $G - e$. Given a pair of graphs $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subset \overline{G}$, we define the set $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ of chordal graphs sandwiched by $\overline{G}$ and $\underline{G}$ as

$$\Omega_{\mathrm{C}}(\overline{G}, \underline{G}) \stackrel{\text{def.}}{=} \{G \mid G \text{ is chordal}, \ \underline{G} \subseteq G \subseteq \overline{G}\}. \tag{1}$$

A graph in $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ is called a *chordal sandwich* for the pair $\overline{G}$ and $\underline{G}$ while $\overline{G}$ and $\underline{G}$ are called the *ceiling graph* and the *floor graph* of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$, respectively. If $\overline{G}$ is a complete graph, then a chordal sandwich is called a *chordal completion* of $\underline{G}$. If $\underline{G}$ is an empty graph (i.e. has no edge), then a chordal sandwich is called a *chordal deletion* of $\overline{G}$.

Note that the graphs are "labeled" in $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$, meaning that we distinguish $G \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ from $G' \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ when their edge sets are different even if they are isomorphic graphs.

We study the following three types of problems: given a pair of graphs $\overline{G}$ and $\underline{G}$ with $\underline{G} \subset \overline{G}$

- output all graphs in $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ (listing);
- output the number $|\Omega_{\mathrm{C}}(\overline{G}, \underline{G})|$ (counting);
- output one graph in $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ uniformly at random (sampling).

Golumbic, Kaplan, and Shamir [7] showed that, given a pair of graphs $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subset \overline{G}$, deciding whether $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ has an element is NP-complete. Therefore, three problems above are all intractable without any restriction. In this paper, we always assume that at least one of $\overline{G}$ and $\underline{G}$ is chordal. For later reference, we write this assumption as a condition.

Condition 1 *A pair of graphs $\overline{G}$ and $\underline{G}$ satisfies $\underline{G} \subset \overline{G}$, and at least one of $\overline{G}$ and $\underline{G}$ is chordal.*

Procedure $A(\overline{G},\underline{G})$ (when $\overline{G}$ is chordal)

```
begin
  find an edge e ∈ Ē \ E̲
        such that Ḡ − e is chordal
  If such e exists do
    output Ḡ − e
    call A(Ḡ, G̲ + e)
    call A(Ḡ − e, G̲)
  otherwise halt
end.
```

Procedure $B(\overline{G},\underline{G})$ (when $\underline{G}$ is chordal)

```
begin
  find an edge e ∈ Ē \ E̲
        such that G̲ + e is chordal
  If such e exists do
    output G̲ + e
    call B(Ḡ, G̲ + e)
    call B(Ḡ − e, G̲)
  otherwise halt
end.
```

Figure 1: Procedures in the listing algorithms.

The following proposition is a key to some of our results.

Proposition 2.1 *Suppose a pair of chordal graphs $\overline{G} = (V, \overline{E})$ and $\underline{G} = (V, \underline{E})$ satisfies $\underline{G} \subset \overline{G}$, and let $k = |\overline{E} \backslash \underline{E}|$. Then there exists a sequence of chordal graphs $G_0, G_1, \ldots, G_k$ that satisfies $G_0 = \underline{G}$, $G_k = \overline{G}$, and $G_{i+1} = G_i + e_i$ with an appropriate edge $e_i \in \overline{E} \setminus \underline{E}$ for each $i \in \{0, \ldots, k-1\}$.*

Proof. We use the following result by Rose, Tarjan, and Lueker [22]: for a graph $G = (V, E)$ and a chordal graph $G' = (V, E \cup F)$ with $E \cap F = \emptyset$, the graph G' is a minimal chordal completion of G (i.e., $\Omega_{\mathrm{C}}(G', G) = \{G'\}$) if and only if $G' - f$ is not chordal for each $f \in F$.

The proof is done by induction on k. If $k = 0$, then $\underline{G} = \overline{G}$ and we are done. Now assume that $k \geq 1$, and the proposition holds for all $k' < k$. In this case, $\overline{G}$ is not a minimal chordal completion of $\underline{G}$ since $\underline{G} \neq \overline{G}$ and $\underline{G}$ is actually a minimal chordal completion of itself. By the result of Rose, Tarjan, and Leuker above, there must exist an edge $f \in \overline{E} \setminus \underline{E}$ such that $\overline{G} - f$ is chordal. Then, letting $G_{k-1} = \overline{G} - f$ and $e_{k-1} = f$, we have $\overline{G} = G_k = G_{k-1} + e_{k-1}$. Further, by the induction hypothesis, there exists a sequence of chordal graphs $\underline{G} = G_0, G_1, \ldots, G_{k-1}$ such that $G_{i+1} = G_i + e_i$ for some $e_i \in (\overline{E} \setminus \{e_{k-1}\}) \setminus \underline{E}$. □

Note that Proposition 2.1 implies that the set of chordal sandwiches forms a graded poset with respect to the inclusion relation of edge sets.

3 Listing All Chordal Sandwiches

We give algorithms to list all chordal sandwiches in $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ for given $\overline{G}$ and $\underline{G}$ satisfying Condition 1.

First consider the case in which the ceiling graph $\overline{G}$ is chordal. Then, there exists an edge $e \in \overline{E} \setminus \underline{E}$ such that $\overline{G} - e$ is chordal if $\Omega_{\mathrm{C}}(\overline{G}, \underline{G}) \setminus \{\overline{G}\} \neq \emptyset$, from Proposition 2.1. For the edge e, we consider a pair of sets $\Omega_{\mathrm{C}}(\overline{G} - e, \underline{G})$ and $\Omega_{\mathrm{C}}(\overline{G}, \underline{G} + e)$. Then, each graph of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ without e is a member of $\Omega_{\mathrm{C}}(\overline{G} - e, \underline{G})$, and each graph of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ with e is a member of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G} + e)$, from the definition of a chordal sandwich. Thus, we obtain a binary partition of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ as follows:

$$\Omega_{\mathrm{C}}(\overline{G}, \underline{G}) = \Omega_{\mathrm{C}}(\overline{G} - e, \underline{G}) \cup \Omega_{\mathrm{C}}(\overline{G}, \underline{G} + e), \text{ and } \Omega_{\mathrm{C}}(\overline{G} - e, \underline{G}) \cap \Omega_{\mathrm{C}}(\overline{G}, \underline{G} + e) = \emptyset.$$

Then, the ceiling graph $\overline{G}$ of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G} + e)$ is chordal, and the ceiling graph $\overline{G} - e$ of $\Omega_{\mathrm{C}}(\overline{G} - e, \underline{G})$ is chordal again from the choice of e. We can repeat the binary partition recursively, until every set becomes empty. More concretely, in our algorithm we first output $\overline{G}$ and call Procedure $A(\overline{G}, \underline{G})$ in Figure 1.

Now we estimate the time complexity of our algorithm. Let $n = |V|$, $m = |\overline{E}|$ and $k = |\overline{E} \setminus \underline{E}|$. We can find an edge e in Step 2 in $\mathrm{O}(k(n+m))$ time by a simple try-and-error approach with a linear-time algorithm to recognize a chordal graph by Rose, Tarjan, and Lueker [22] or by Tarjan and Yannakakis [24]. The try-and-error algorithm can be improved to $\mathrm{O}(kn + n\log n)$ time by a dynamic data structure proposed by Ibarra [10].

The binary partition is valid in the sense that we always obtain a pair of non-empty sets in recursive calls. Thus, the accumulated number of recursive calls made by a call to $A(\overline{G}, \underline{G})$ is proportional to the number of outputs $|\Omega_{\mathrm{C}}(\overline{G}, \underline{G})|$. Therefore, the total time complexity is $\mathrm{O}(k(n+m) \cdot |\Omega_{\mathrm{C}}(\overline{G}, \underline{G})|)$ or $\mathrm{O}((kn + n\log n) \cdot |\Omega_{\mathrm{C}}(\overline{G}, \underline{G})|)$, depending on the algorithm to find the edge e.

Consider next the case in which the floor graph $\underline{G}$ is chordal. In this case, we may obtain a similar algorithm. Procedure $B(\overline{G}, \underline{G})$ in Figure 1 shows the concrete algorithm. The time complexity can also be estimated similarly, but in this case we can find an appropriate e faster, namely in $\mathrm{O}(k\log^2 n + n)$ [10].

Acceleration of finding an appropriated edge We can construct $\mathrm{O}(n+m)$ time algorithms to find a chordal graph $\overline{G} - e \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ and $\underline{G} + e \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$, respectively, when both $\overline{G}$ and $\underline{G}$ are chordal. These algorithms are not based on try-and-error; each of them essentially executes perfect eliminations twice (see Section A in Appendix). They are faster than Ibarra's dynamic algorithm [10] when k is sufficiently large, namely $k = \Omega(m/n)$ if $\overline{G}$ is chordal and $k = \Omega(m/\log^2 n)$ if $\underline{G}$ is chordal, respectively. Note that this modification may not improve the theoretical time complexity of our listing algorithms, since even when both of $\overline{G}$ and $\underline{G}$ are chordal, graphs appearing in recursive calls are usually not chordal.

4 Hardness of Counting the Chordal Sandwiches

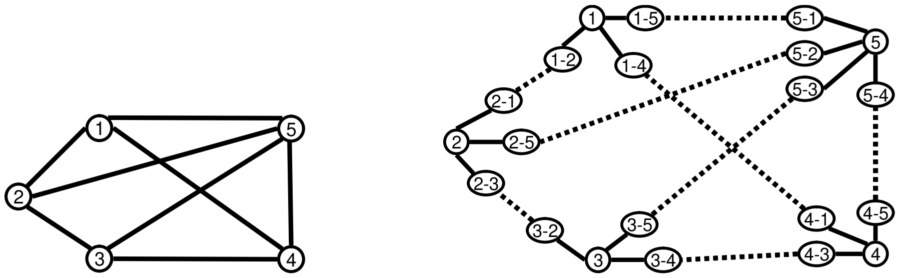
Here, we show the #P-completeness of counting the chordal sandwiches by a parsimonious reduction. We also show the #P-completeness of counting the chordal deletions by a Cook reduction. These results imply that random sampling of chordal graphs is not easy, as indicated by many previous results about the relationship between the (approximate) counting and the random sampling (see e.g., [23]).

4.1 Counting the chordal sandwiches is at least as hard as counting the forests

First we show the following Theorem.

Theorem 4.1 *The computation of $|\Omega_{\mathrm{C}}(\overline{G}, \underline{G})|$ is #P-complete, even when $\underline{G}$ is a connected chordal graph.*

We give a reduction from the problem to count the forests in a graph, which is known to be #P-complete [27]. Note that the reduction is parsimonious. Thus, if we have an approximation algorithm for the chordal sandwich counting, then we obtain an approximation algorithm for the forest counting with the same approximation ratio. Note that it is still a widely open problem whether or not a fully polynomial-time randomized approximation scheme exists for the forest counting problem.



(a) An original graph H. (b) The reduced graphs $\overline{G}$ and $\underline{G}$.

Figure 2: An example of the transformation.

Proof. The problem is clearly in #P. It is enough to show the #P-hardness.

First, we give a transformation of an instance (i.e., a graph) H of the forest counting problem into an instance (i.e., a pair of graphs) $\overline{G}$ and $\underline{G}$ of the chordal sandwich counting problem. To construct $\overline{G}$, we just replace every edge $\{u, v\} \in E(H)$ with a path of length three. Let $w_{u,v}$ and $w_{v,u}$ be new vertices of $\overline{G}$, which subdivide an edge $\{u, v\} \in E(H)$. Then, $|V(\overline{G})| = |V(H)| + 2|E(H)|$ and $|E(\overline{G})| = 3|E(H)|$ hold. To construct $\underline{G}$, we just remove every edge of the form $\{w_{u,v}, w_{v,u}\} \in E(\overline{G})$ from $\overline{G}$. Figure 2 shows an example of the transformation. In Figure 2 (b), the edges of $\underline{G}$ are drawn by solid lines, and the edges of $\overline{G}$ are drawn by solid lines and dashed lines. Note that $\underline{G}$ is a chordal graph consisting of n disjoint stars. Moreover, the girth (i.e., the length of a shortest cycle) of $\overline{G}$ is at least 9.

Next, we show that there exists one-to-one correspondence between the set of forests in H and $\Omega_{\rm C}(\overline{G}, \underline{G})$. For a forest $F = (V(H), E(F))$ in H, we define the corresponding graph $G \in \Omega_{\rm C}(\overline{G}, \underline{G})$ as $E(G) = E(\underline{G}) \cup \{\{w_{u,v}, w_{v,u}\} \in E(\overline{G}) \setminus E(\underline{G}) \mid \{u, v\} \in E(F)\}$. Then, G does not have any cycle, and G is chordal. Conversely, every graph in $\Omega_{\rm C}(\overline{G}, \underline{G})$ does not contain any cycle and is chordal since the girth of $\overline{G}$ is at least 9. Thus, for any $G \in \Omega_{\rm C}(\overline{G}, \underline{G})$, there exists a corresponding forest in H as the inverse of the above map. Hence, we obtain a bijection. Thus, we showed that the computation of $|\Omega_{\rm C}(\overline{G}, \underline{G})|$ is #P-hard even when $\underline{G}$ is chordal.

To obtain the full theorem we transform $\Omega_{\rm C}(\overline{G}, \underline{G})$ into $\Omega_{\rm C}(\overline{G}', \underline{G}')$ in which $\underline{G}'$ is connected and chordal. Let G be a graph. We transform G into $\Phi(G)$ defined as $V(\Phi(G)) \stackrel{\text{def.}}{=} V(G) \cup \{v_0\}$ and $E(\Phi(G)) \stackrel{\text{def.}}{=} E(G) \cup \{\{v_0, v\} \mid v \in V(G)\}$. Clearly $\Phi(G)$ is connected. Furthermore, G is chordal if and only if $\Phi(G)$ is chordal. Now, we define a pair of graphs $\overline{G}' \stackrel{\text{def.}}{=} \Phi(\overline{G})$ and $\underline{G}' \stackrel{\text{def.}}{=} \Phi(\underline{G})$ from the pair graphs $\overline{G}$ and $\underline{G}$. Then, $\underline{G}'$ is chordal when $\underline{G}$ is chordal, and $\Omega_{\rm C}(\overline{G}, \underline{G})$ and $\Omega_{\rm C}(\overline{G}', \underline{G}')$ are in one-to-one correspondence via Φ. Thus, we obtain the theorem. □

4.2 Hardness of counting the chordal deletions

The set of chordal deletions of G is described as the set of chordal sandwiches in $\Omega_{\rm C}(G, I_n)$, where I_n is an empty graph with n vertices and no edge.

Theorem 4.2 *The computation of* $|\Omega_{\rm C}(G, I_n)|$ *is #P-complete.*

We give a Cook-reduction from the forest counting problem in a graph. Note that the reduction does not preserve the approximation ratio.

Proof. Let H be a graph with n vertices and m edges. First, we describe the construction of a graph G^i for each $i \in \{1, \ldots, n\}$ from H as follows; we just replace every edge $\{u, v\}$ in H with a path $\mathcal{P}^i(\{u, v\})$ of length $i+1$. Then, $|V(G^i)| = |V(H)| + i|E(H)|$ and $|E(G^i)| = (i+1)|E(H)|$ hold. Moreover, the girth of G^i is at least $3i$. Note that G^2 is the same graph as $\overline{G}$ appearing in the proof of Theorem 4.1.

We denote the set of forests with k edges in H as $\mathcal{F}(k)$. Now, we show that

$$|\Omega_{\mathrm{C}}(G^i, I_n)| = \sum_{k=0}^{n-1} (2^{i+1} - 1)^{m-k} |\mathcal{F}(k)| \qquad \text{for } i \in \{1, \ldots, n\}.$$

For a forest $F = (V(H), E(F))$ in H, we define a set $\Delta^i(F) \subseteq \Omega_{\mathrm{C}}(G^i, I_n)$ as $G \in \Delta^i(F)$ if and only if G satisfies the following two conditions;

1. If $\{u, v\} \in E(F)$, then G contains all $i+1$ edges in the path $\mathcal{P}^i(\{u, v\})$.
2. If $\{u, v\} \notin E(F)$, then G contains at most i edges in the path $\mathcal{P}^i(\{u, v\})$.

Then, we obtain $|\Delta^i(F)| = (2^{i+1} - 1)^{m-k}$ for a forest $F \in \mathcal{F}(k)$. If a pair of forests F_1 and F_2 in H satisfies $F_1 \neq F_2$, then $\Delta^i(F_1) \cap \Delta^i(F_2) = \emptyset$ holds from the definition. Every graph in $\Omega_{\mathrm{C}}(G^i, I_n)$ must be a forest, namely chordal, since the girth of G^i is at least 4. Thus, for any $G \in \Omega_{\mathrm{C}}(G^i, I_n)$, there exists a forest F in H such that $G \in \Delta^i(F)$. It implies $\Omega_{\mathrm{C}}(G^i, I_n) = \bigcup\{\Delta^i(F) \mid F \text{ is a forest of } H\}$, and hence we obtain

$$|\Omega_{\mathrm{C}}(G^i, I_n)| = \sum_{F} |\Delta^i(F)| = \sum_{k=0}^{n-1} \sum_{F \in \mathcal{F}(k)} (2^{i+1} - 1)^{m-k} = \sum_{k=0}^{n-1} (2^{i+1} - 1)^{m-k} |\mathcal{F}(k)|.$$

Then we obtain a linear equation system

$$\begin{pmatrix} |\Omega_{\mathrm{C}}(G^1, I_n)| \\ |\Omega_{\mathrm{C}}(G^2, I_n)| \\ \vdots \\ |\Omega_{\mathrm{C}}(G^n, I_n)| \end{pmatrix} = A \begin{pmatrix} |\mathcal{F}(0)| \\ |\mathcal{F}(1)| \\ \vdots \\ |\mathcal{F}(n-1)| \end{pmatrix},$$

where A is an $n \times n$ matrix defined by

$$A \stackrel{\text{def.}}{=} \begin{pmatrix} 3^m & 3^{m-1} & \cdots & 3^{m-n+1} \\ 7^m & 7^{m-1} & \cdots & 7^{m-n+1} \\ \vdots & \vdots & \ddots & \vdots \\ (2^{n+1}-1)^m & (2^{n+1}-1)^{m-1} & \cdots & (2^{n+1}-1)^{m-n+1} \end{pmatrix}.$$

Here, A is a Vandermonde matrix, and is non-singular. Thus we have reduced the forest counting into the chordal deletion counting. $\square$

5 A Simple Markov Chain and Its Slow-Mixing

Here, we consider a uniform sampling on $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ satisfying Condition 1. We give a simple and natural Markov chain. Note that the following Markov chain can be easily modified into ones for non-uniform distributions by a Metropolis-Hastings method.

Let $\mathcal{M}$ be a Markov chain with a state space $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ with Condition 1. A transition of $\mathcal{M}$ from a current state $G \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ to a next state G' is defined as follows; Choose an edge $e \in (\overline{E} \setminus \underline{E})$ uniformly at random. We consider the following three cases.

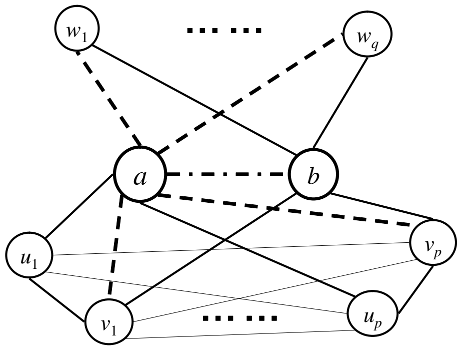


Figure 3: Example of an input pair on which the simple Markov chain $\mathcal{M}$ mixes slowly.

1. If $e \notin E(G)$ and $G + e$ is chordal, then set $H = G + e$.
2. If $e \in E(G)$ and $G - e$ is chordal, then set $H = G - e$.
3. Otherwise set $H = G$.

Let $G' = H$ with the probability 1/2, otherwise let $G' = G$. Clearly $G' \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$.

The chain $\mathcal{M}$ is irreducible from the fact that $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ on Condition 1 forms a graded poset. The chain $\mathcal{M}$ is clearly aperiodic, and hence $\mathcal{M}$ is ergodic. The unique stationary distribution of $\mathcal{M}$ is the uniform distribution on $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$, since the detailed balanced equation $P(G, G') = P(G', G)$ holds for any pair of $G \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ and $G' \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$. From Proposition 2.1, the diameter of $\mathcal{M}$ is at most $2k$, where $k = |\overline{E} \setminus \underline{E}|$.

Now, we discuss the mixing time of the Markov chain. Let μ and ν be a pair of distributions on a common finite set Ω. The *total variation distance* $d_{\mathrm{TV}}(\mu, \nu)$ between μ and ν is defined by $d_{\mathrm{TV}}(\mu, \nu) \stackrel{\text{def.}}{=} \frac{1}{2} \sum_{x \in \Omega} |\mu(x) - \nu(x)|$. For an arbitrary positive ε, the *mixing time* $\tau(\varepsilon)$ of an ergodic Markov chain MC with a state space Ω is defined by $\tau(\varepsilon) \stackrel{\text{def.}}{=} \max_{x \in \Omega} \min\{t \mid \forall s \geq t,\ d_{\mathrm{TV}}(P_x^s, \pi) \leq \varepsilon\}$ where π is the unique stationary distribution of $\mathcal{M}$, and P_x^s denotes a distribution of $\mathcal{M}$ at time s starting from a state x.

In the following, we show that the Markov chain $\mathcal{M}$ is not rapidly mixing for some inputs.

Proposition 5.1 *There exist infinitely many pairs of chordal graphs $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subseteq \overline{G}$ for which the mixing time of $\mathcal{M}$ on $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ is exponential in n, where n is the number of vertices of $\overline{G}$ (and $\underline{G}$).*

Proof. Figure 3 shows an example. Let V be a set of vertices $\{a, b, v_1, \ldots, v_p, u_1, \ldots, u_p, w_1 \ldots, w_q\}$. Let $\underline{G} = (V, E(\underline{G}))$ be a graph defined by

$$
\begin{aligned}
E(\underline{G}) \stackrel{\text{def.}}{=} & \{\{a, u_i\} \mid i \in \{1, \ldots, p\}\} \cup \{\{b, v_i\} \mid i \in \{1, \ldots, p\}\} \\
& \cup \{\{b, w_i\} \mid i \in \{1, \ldots, q\}\} \cup \{\{u_i, v_j\} \mid (i, j) \in \{1, \ldots, p\}^2\}.
\end{aligned}
$$

Let $\overline{G} = (V, E(\overline{G}))$ be a graph defined by

$$
E(\overline{G}) \stackrel{\text{def.}}{=} E(\underline{G}) \cup \{\{a, v_i\} \mid i \in \{1, \ldots, p\}\} \cup \{\{a, w_i\} \mid i \in \{1, \ldots, q\}\} \cup \{\{a, b\}\}.
$$

In Figure 3, $\underline{G}$ is described by solid lines, and $\overline{G}$ is described by solid lines and dashed lines. Note that both $\underline{G}$ and $\overline{G}$ are chordal.

Now, let $G = (V, E(G))$ be a graph defined by $E(G) \stackrel{\text{def.}}{=} E(\underline{G}) \cup \{\{a, v_i\} \mid i \in \{1, \dots, p\}\}$, and Let G' be a graph defined by $G' \stackrel{\text{def.}}{=} G + \{a, b\}$. Then, $G \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ and $G' \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$. We show that a bottle-neck lies between G and G'.

If a graph $H \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ contains the edge $\{a, b\}$, then H contains all edges of $\{\{a, v_i\} \mid i \in \{1, \dots, p\}\} \subset E(\overline{G}) \setminus E(\underline{G})$. Otherwise H has a chordless cycle a-b-v_i-u_i-a with $\{a, v_i\} \notin E(H)$. If a graph $H \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ does not contain the edge $\{a, b\}$, then H does not contain any edge of $\{\{a, w_i\} \mid i \in \{1, \dots, q\}\} \subset E(\overline{G}) \setminus E(\underline{G})$. Otherwise H has a chordless cycle w_i-a-u_1-v_1-b-w_i when $\{a, v_1\} \notin E(H)$, or w_i-a-v_1-b-w_i when $\{a, v_1\} \in E(H)$, with $\{a, w_i\} \in E(H)$. From above, the set $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ can be partitioned into $\Omega_{\mathrm{C}}(\overline{G}, G')$ and $\Omega_{\mathrm{C}}(G, \underline{G})$, where $\Omega_{\mathrm{C}}(\overline{G}, G') \cap \Omega_{\mathrm{C}}(G, \underline{G}) = \emptyset$. Thus, the pair of G and G' is the bottleneck of $\mathcal{M}$ on $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ from the definition of $\mathcal{M}$.

If a graph H satisfies $G' \subseteq H \subseteq \overline{G}$ then H is chordal. It implies $|\Omega_{\mathrm{C}}(\overline{G}, G')| = 2^q$. In the same way, we obtain $|\Omega_{\mathrm{C}}(G, \underline{G})| = 2^p$. Let $p = q = (n-2)/3$, and by computing the conductance between G and G', we can show that the mixing time of $\mathcal{M}$ starting from a worst state is exponential of n, based on the *conductance method* [23]. $\square$

6 Concluding Remarks

We gave a simple and natural Markov chain for uniform sampling of $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$, and showed an example for which the mixing time of the chain is exponential, even when both of $\overline{G}$ and $\underline{G}$ are chordal. It is open if there is a rapidly mixing Markov chain. Our Markov chain uses the fact that $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ for given $\overline{G}$ and $\underline{G}$ with Condition 1 forms a graded poset. However, it is known that the set of chordal sandwiches generally does not form a lattice even when both $\overline{G}$ and $\underline{G}$ are chordal. A future work would include a characterization of the pairs $\overline{G}$ and $\underline{G}$ such that $\Omega_{\mathrm{C}}(\overline{G}, \underline{G})$ forms a lattice.

It is open that counting the chordal sandwiches is #P-hard when a given ceiling graph is restricted to be chordal. We conjecture that counting the chordal completions (i.e., when a given ceiling graph is complete) is #P-complete. We can consider listing, counting, and sampling the graph sandwiches for other graph classes, such as interval, proper interval, or perfect graphs. We can show that counting the interval sandwiches is #P-complete even when a given floor graph is connected and interval (see Section B in Appendix).

References

[1] D. Avis and K. Fukuda, Reverse search for enumeration, Discrete Applied Mathematics, **65** (1996), 21–46.

[2] G. A. Dirac, On rigid circuit graphs, Abhandl. Math. Seminar Univ. Hamburg, **25** (1961), 71–76.

[3] D. R. Fulkerson, O. A. Gross, Incidence matrices and interval graphs, Pacific Journal of Mathematics, **15** (1965), 835–855.

[4] F. V. Fomin, D. Kratsch, and I. Todinca, Exact (exponential) algorithms for treewidth and minimum fill-in. Lecture Notes in Computer Science, **3142** (2004), 568–580.

[5] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, Exploiting sparsity in semidefinite programming via matrix completion I: General framework, SIAM Journal on Optimization, **11** (2000), 647–674.

[6] M. C. Golumbic, Algorithmic graph theory and perfect graphs, Academic Press, New York, 1980.

[7] M. C. Golumbic, H. Kaplan, and R. Shamir, Graph sandwich problems, Journal of Algorithms, **19** (1995), 449–473.

[8] P. Heggernes, Minimal triangulations of graphs: A survey, Discrete Mathematics, **306** (2006), 297–317.

[9] P. Heggernes, K. Suchan, I. Todinca, and Y. Villanger, Characterizing minimal interval completions: towards better understanding of profile and pathwidth, Lecture Notes in Computer Science, **4393** (2007), 236–247.

[10] L. Ibarra, Fully dynamic algorithms for chordal graphs, Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1999), 923–924.

[11] M. Kiyomi and T. Uno, Generating chordal graphs included in given graphs, IEICE Transactions on Information and Systems, **E89-D** (2006), 763–770.

[12] M. Kiyomi, S. Kijima, and T. Uno, Listing chordal graphs and interval graphs, Lecture Notes in Computer Science, **4271** (2006), 68–77.

[13] T. Kloks, H. L. Bodlaender, H. Müller, and D. Kratsch, Computing treewidth and minimum fill-in: All you need are the minimal separators, Lecture Notes in Computer Science, **726** (1993), 260-271.

[14] T. Kloks, H. L. Bodlaender, H. Müller, and D. Kratsch, Erratum to the ESA' 93 proceedings, Lecture Notes in Computer Science, **855** (1994), 508.

[15] C. G. Leckerkerker and J. C. Boland, Representation of a finite graph by a set of intervals on the real line, Fund Math, **51** (1962), 45–64.

[16] D. Marx, Chordal deletion is fixed-parameter tractable, Lecture Notes in Computer Science, **4271** (2006), 37–48.

[17] A. Natanzon, R. Shamir, and R. Sharan, A polynomial approximation algorithm for the minimum fill-in problem, SIAM Journal on Computing, **30** (2000), 1067–1079.

[18] A. Natanzon, R. Shamir, and R. Sharan, Complexity classification of some edge modification problems, Discrete Applied Mathematics, **113** (2001), 109–128.

[19] T. Pedersen, R. F. Bruce, J. Wiebe, Sequential Model Selection for Word Sense Disambiguation, Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP 1997), 388–395.

[20] N. Robertson and P. Seymour, Graph minors II. Algorithmic aspects of tree-width, Journal of Algorithms, **7** (1986), 309–322.

[21] D. J. Rose, A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, in: R.C. Read (Ed.), Graph Theory and Computing, Academic Press, New York, 1972, pp. 183–217.

[22] D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic aspects of vertex elimination on graphs, SIAM Journal on Computing, **5** (1976), 266–283.

[23] A. Sinclair, Algorithms for Random Generation and Counting: A Markov Chain Approach, Birkhäuser, Boston, 1993.

[24] R. E. Tarjan and M. Yannakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM Journal on Computing, **13** (1984), 566–579.

[25] A. Takemura and Y. Endo, Evaluation of per-record identification risk and swappability of records in a microdata set via decomposable models, arXiv:math.ST/0603609.

[26] V. G. Valiant, The complexity of computing the permanent, Theoretical Computer Science, **8** (1979), 189–201.

[27] V. G. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing, **8** (1979), 410–421.

[28] N. Yamashita, Sparse quasi-Newton updates with positive definite matrix completion, Mathematical Programming, 2007, published online at
`http://www.springerlink.com/content/28271224t1nt3580/`

[29] M. Yannakakis, Computing the minimum fill-in is NP-complete, SIAM Journal on Algebraic and Discrete Methods, **2** (1981), 77–79.

[30] J. Whittaker, Graphical models in applied multivariate statistics, Wiley, New York, 1990.

[31] N. C. Wormald, Counting labeled chordal graphs, Graphs and Combinatorics, **1** (1985), 193–200.

A Acceleration of Finding an Appropriate Edge in Our Listing Algorithms

Here, we give a linear time algorithm to find an appropriate edge which we use in our listing algorithms.

A.1 Notations and Properties of Chordal Graphs

As a preliminary step, we explain a well-known characterization of chordal graphs by perfect elimination orderings. For a graph $G = (V, E)$ and a vertex $v \in V$, let $\delta(v; G)$ be the set of edges incident to $v \in V$ on G, and let $N(v; G)$ be the set of vertices adjacent to $v \in V$ on G, i.e., $\delta(v; G) = \{\{v, u\} \in E \mid u \in V\}$ and $N(v; G) = \{u \in V \mid \{v, u\} \in E\}$. For a graph $G = (V, E)$ and a subset $V' \subseteq V$ of vertices, $G[V']$ denotes the graph induced from G by V', i.e., $G[V'] = (V', \{e = \{v, v'\} \in E \mid v \in V',\ v' \in V'\})$.

For a graph G, a vertex $v \in V(G)$ is *simplicial* if the set $N(v; G)$ of vertices induces a clique. For a graph $G = (V, E)$ with n vertices, a sequence $\boldsymbol{v} = (v_1, \ldots, v_n)$ of all vertices in V is a *perfect elimination ordering* of G if the vertex v_i is a simplicial vertex of the graph $G[v_i, \ldots, v_n]$ for each $i \in \{1, \ldots, n\}$. It is known that a graph G is chordal if and only if the graph G has a perfect elimination ordering [2, 3, 21]. Moreover, if G is chordal, then there exists a perfect elimination ordering $(v_1, \ldots, v_n)$ satisfying $v_n = u$ for any vertex $u \in V$. Rose, Tarjan, and Lueker [22] proposed an $\mathrm{O}(n + m)$ algorithm to check the chordality of a graph. The algorithm also provides a perfect elimination ordering $(v_1, \ldots, v_n)$ satisfying $v_n = u$ for any vertex $u \in V$ when the graph is chordal. The next lemma is used in the proof of Propositions A.2 and A.3.

Lemma A.1 *For a pair of graphs $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subseteq \overline{G}$, if $v \in V$ is a simplicial vertex of $\underline{G}$ and $\delta(v; \overline{G}) = \delta(v; \underline{G})$ holds, then v is also a simplicial vertex of $\overline{G}$.*

Proof. Since $\delta(v; \overline{G}) = \delta(v; \underline{G})$, $N(v; \overline{G})$ is identical to $N(v; \underline{G})$. Since $\underline{G}[N(v; \underline{G})]$ is a clique of $\underline{G}$ and $\underline{G}[N(v; \underline{G})] \subseteq \overline{G}[N(v; \overline{G})]$ holds, $\overline{G}[N(v; \overline{G})]$ is a clique of $\overline{G}$. Hence v is a simplicial vertex of $\overline{G}$. □

A.2 Linear Time Algorithm to Find a Chordal Graph $\underline{G} + e \in \Omega_{\mathrm{C}}(\overline{G}, \underline{G})$

Here we describe our algorithm.

Proposition A.2 *Given a pair of chordal graphs $\overline{G} = (V, \overline{E})$ and $\underline{G} = (V, \underline{E})$ satisfying $\underline{G} \subset \overline{G}$, we can find an edge $e \in \overline{E} \setminus \underline{E}$ such that $\underline{G} + e$ is chordal in $\mathrm{O}(n + m)$ time.*

Proof. Consider the following Procedure 1, given the pair of chordal graphs $\overline{G}$ and $\underline{G}$.

<u>Procedure 1</u>

Step 1. Find a perfect elimination ordering $(p_1, \ldots, p_n)$ of $\underline{G}$.

Step 2. Find an index $s = \max\{i \in \{1, \ldots, n\} \mid D_i \neq \emptyset\}$,
where $D_i \overset{\text{def.}}{=} E(\overline{G}[p_i, \ldots, p_n]) \cap (\overline{E} \setminus \underline{E})$.

Step 3. Output $G' \overset{\text{def.}}{=} (V, \underline{E} \cup D_s)$.

Note first that we can always choose an index in Step 2 since $D_1 = E(\overline{G}[p_1, \ldots, p_n]) \cap (\overline{E} \setminus \underline{E}) = \overline{E} \cap (\overline{E} \setminus \underline{E}) = \overline{E} \setminus \underline{E} \neq \emptyset$. Secondly, we note that the output graph G' of Procedure 1 satisfies $\underline{G} \subset G' \subseteq \overline{G}$.

Now we show that G' is chordal. To this end we construct a perfect elimination ordering of G'. Let $s \in \{1,\ldots,n\}$ be the index obtained in Step 2. Fix an index $i \in \{1,\ldots,s-1\}$ arbitrary. We may observe that p_i is a simplicial vertex of $G'[p_i,\ldots,p_n]$ from Lemma A.1 since $\underline{G}[p_i,\ldots,p_n] \subseteq G'[p_i,\ldots,p_n]$, p_i is a simplicial vertex of $\underline{G}[p_i,\ldots,p_n]$, and $\delta(p_i;G') = \delta(p_i;\underline{G})$. Furthermore $G'[p_s,\ldots,p_n]$ is chordal since $G'[p_s,\ldots,p_n] = \overline{G}[p_s,\ldots,p_n]$ and $\overline{G}[p_s,\ldots,p_n]$ is chordal. It implies that $G'[p_s,\ldots,p_n]$ has a perfect elimination ordering $(p'_s,\ldots,,p'_n)$. Therefore an order $(p_1,\ldots,p_{s-1},p'_s,\ldots,p'_n)$ is a perfect elimination ordering of G', and hence G' is chordal.

If $|D_s| = 1$, we readily obtain the claim. If $|D_s| \geq 2$, we need an extra procedure. Namely, since G' and $\underline{G}$ satisfy $\underline{G} \subset G'$, we execute Procedure 1 again but for the pair of G' and $\underline{G}$ and find a perfect elimination ordering with a special property in Step 1. Below is more detail.

Let $(q_1,\ldots,q_n)$ be a perfect elimination ordering of $\underline{G}$ satisfying $q_n = p_s$. We define $D'_i \stackrel{\text{def.}}{=} E\left(G'[q_i\ldots,q_n]\right) \cap D_s$ for $i \in \{1,\ldots,n\}$. Since D_s consists of edges of $G'[p_s,\ldots,p_n]$ only and satisfies $E(G'[p_{s+1},\ldots,p_n]) \cap D_s = \emptyset$ from the choice of s, every edge of D_s is incident to the vertex p_s on the graph G'. Therefore, the cardinality of the set $D'_i \setminus D'_{i+1}$ is at most one for each $i \in \{1,\ldots,n-1\}$. Let $t \in \{1,\ldots,n\}$ be an index satisfying

$$t = \max\{i \in \{1,\ldots,n\} \mid D'_i \neq \emptyset\}.$$

Then $|D'_t| = 1$. Let e be a unique element of D'_t. We may observe that $\underline{G} + e$ is chordal by the same argument as G' is chordal. $\square$

The following Algorithm 1 is naturally derived from the proof.

Algorithm 1

Input: a pair of chordal graphs $\overline{G} = (V,\overline{E})$, $\underline{G} = (V,\underline{E})$ satisfying $\underline{G} \subset \overline{G}$.
Output: a chordal graph $G := \underline{G} + e$ satisfying $\underline{G} \subset G \subseteq \overline{G}$ and $e \in \overline{E} \setminus \underline{E}$.

Phase I (Procedure 1)

I-1. Find a perfect elimination ordering $(p_1,\ldots,p_n)$ of $\underline{G}$.
I-2. Find an index $s = \max\{i \in \{1,\ldots,n\} \mid D_i \neq \emptyset\}$,
where $D_i \stackrel{\text{def.}}{=} E(\overline{G}[p_i,\ldots,p_n]) \cap (\overline{E} \setminus \underline{E})$.
I-3. If $|D_s| = 1$, then output $G + e$ for a unique $e \in D_s$, and halt.
Otherwise, set $G' \stackrel{\text{def.}}{=} (V, \underline{E} \cup D_s)$, and go to Phase II.

Phase II (simple modification of Procedure 1)

II-1. Find a perfect elimination ordering $(q_1,\ldots,q_n)$ of $\underline{G}$ satisfying $q_n = p_s$.
II-2. Find an index $t = \max\{i \in \{1,\ldots,n\} \mid D'_i \neq \emptyset\}$,
where $D'_i \stackrel{\text{def.}}{=} E(G'[q_i,\ldots,q_n]) \cap D_s$.
II-3. Output $\underline{G} + e$ for a unique $e \in D'_t$, and halt.

A.3 Linear Time Algorithm to Find a Chordal Graph $\overline{G} - e \in \Omega_{\mathrm{C}}(\overline{G},\underline{G})$

We next consider the case that $\underline{G}$ is chordal. Similarly to the previous case, we propose the following algorithm.

Algorithm 2

Input: A pair of chordal graphs $\overline{G} = (V, \overline{E})$, $\underline{G} = (V, \underline{E})$ satisfying $\underline{G} \subset \overline{G}$.
Output: A chordal graph $G := \overline{G} - e$ satisfying $\underline{G} \subseteq G \subset \overline{G}$ and $e \in \overline{E} \setminus \underline{E}$.

Phase I

I-1. Find a perfect elimination ordering $(p_1, \ldots, p_n)$ of $\underline{G}$.
I-2. Find an index $s = \min\{i \in \{1, \ldots, n\} \mid A_i \neq \emptyset\}$,
where $A_i \stackrel{\text{def.}}{=} \delta(p_i; \overline{G}[p_i, \ldots, p_n]) \cap (\overline{E} \setminus \underline{E})$.
I-3. If $|A_s| = 1$, then output $\overline{G} - e$ for a unique $e \in A_s$, and halt.
Otherwise, set $G' \stackrel{\text{def.}}{=} (V, \overline{E} \setminus A_s)$ and go to Phase II.

Phase II (simple modification of Phase I)

II-1. Find a perfect elimination ordering $(q_1, \ldots, q_n)$ of G' satisfying $q_n = p_s$.
II-2. Find an index $t = \min\{i \in \{1, \ldots, n\} \mid A'_i \neq \emptyset\}$,
where $A'_i \stackrel{\text{def.}}{=} \delta(p_i; \overline{G}[q_i, \ldots, q_n]) \cap A_s$.
II-3. Output $\overline{G} - e$ for a unique $e \in A'_t$, and halt.

It is easy to see that the time complexity of Algorithm 2 is $\mathrm{O}(n+m)$, since we can execute each step of each phase in $\mathrm{O}(n+m)$ time.

Proposition A.3 *For a pair of chordal graphs $\overline{G} = (V, \overline{E})$ and $\underline{G} = (V, \underline{E})$ satisfying $\underline{G} \subset \overline{G}$, the output $\overline{G} - e$ of Algorithm 2 is chordal.*

Proof. Consider the following procedure for the pair of chordal graphs $\overline{G}$ and $\underline{G}$, which corresponds to Phase I.

Procedure 2

Step 1. Find a perfect elimination ordering $(p_1, \ldots, p_n)$ of $\underline{G}$.
Step 2. Find an index $s = \min\{i \in \{1, \ldots, n\} \mid A_i \neq \emptyset\}$,
where $A_i \stackrel{\text{def.}}{=} \delta(p_i; \overline{G}[p_i, \ldots, p_n]) \cap (\overline{E} \setminus \underline{E})$.
Step 3. Output $G' \stackrel{\text{def.}}{=} (V, \overline{E} \setminus A_s)$.

Note first that we can always find an index s in Step 2 since $\overline{G}$ contains at least one edge, say $e = \{p_i, p_j\}$ with $i < j$, and $e \in A_i$. Secondly we note that the output graph G' of Procedure 2 satisfies $\underline{G} \subset G' \subseteq \overline{G}$.

Now we show that G' is chordal. To this end we construct a perfect elimination ordering of G'. Let $s \in \{1, \ldots, n\}$ be the index obtained in Step 2. For each index $i \in \{1, \ldots, s-1\}$, we may observe that p_i is a simplicial vertex of $G'[p_i, \ldots, p_n]$ from Lemma A.1 since $\underline{G}[p_i, \ldots, p_n] \subseteq G'[p_i, \ldots, p_n]$, p_i is a simplicial vertex of $\underline{G}[p_i, \ldots, p_n]$, and $\delta(p_i; G') = \delta(p_i; \underline{G})$. Furthermore, $G'[p_s, \ldots, p_n]$ is chordal since $G'[p_s, \ldots, p_n] = \overline{G}[p_s, \ldots, p_n]$ and $\overline{G}[p_s, \ldots, p_n]$ is chordal. It implies that $G'[p_s, \ldots, p_n]$ has a perfect elimination ordering $(p'_s, \ldots, p'_n)$. Therefore an order $(p_1, \ldots, p_{s-1}, p'_s, \ldots, p'_n)$ is a perfect elimination ordering of G', and hence G' is chordal.

If $|A_s| = 1$, then Procedure 2, i.e., Phase I of Algorithm 2, immediately gives an appropriate chordal graph. Thus we obtain the claim in this case. On the other hand, if $|A_s| \geq 2$, then we execute Phase II of the algorithm, which is essentially the same as Procedure 2 except for having $\overline{G}$ and G' as an input and finding a perfect elimination ordering that ends with p_s. In this case, every edge of A_s is incident to the vertex p_s on the graph G' by the choice of s. Therefore, the cardinality of A'_t, which corresponds to A_s of Procedure 2, is one. Thus, we obtain the claim also in this case. $\square$

B Hardness of Counting the Interval Sandwiches

Here, we show that counting the interval sandwiches is #P-complete. We begin with definitions.

A graph is an *interval graph* if it has an interval representation. It is easy to see and well-known that an interval graph is chordal. An *asteroidal triple* (or *AT* in short) in a graph is an (unordered) triple of independent vertices of the graph such that every two of them are connected by a path avoiding the neighborhood of the third. A graph is *asteroidal-triple free* (or *AT-free*) if it does not contain any asteroidal triples. Leckerkerker and Boland showed that a graph is interval if and only if it is chordal and AT-free [15]. More information on interval graphs can be found in [6, 7].

Given a pair of $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subset \overline{G}$, we define the set $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ of interval graphs sandwiched by $\overline{G}$ and $\underline{G}$ as

$$\Omega_{\mathrm{I}}(\overline{G}, \underline{G}) \stackrel{\text{def.}}{=} \{G \mid G \text{ is interval, } \underline{G} \subseteq G \subseteq \overline{G}\}. \tag{2}$$

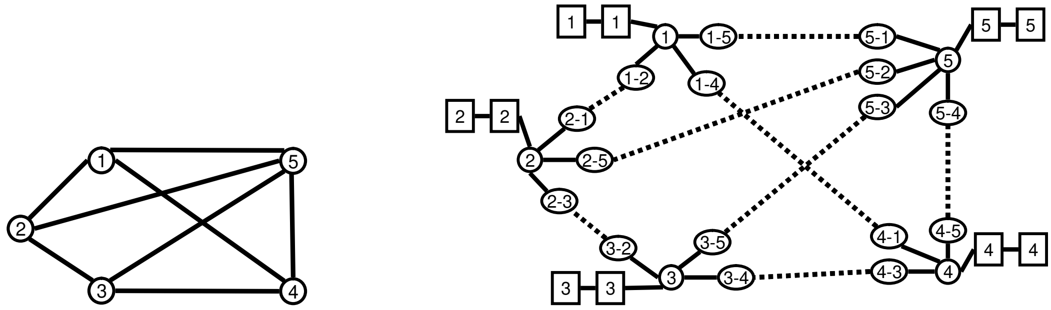
A graph in $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ is called an *interval sandwich* for the pair of $\overline{G}$ and $\underline{G}$ while $\overline{G}$ and $\underline{G}$ are called the *ceiling graph* and the *floor graph* of $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$, respectively. Note that the graphs are "labeled" in $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ in an analogous fashion to the set of chordal graph sandwiches. Golumbic, Kaplan, and Shamir [7] showed that given a pair of graphs $\overline{G}$ and $\underline{G}$ satisfying $\underline{G} \subset \overline{G}$, deciding whether $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ has an element is NP-complete. The rest of the section is devoted to the following theorem.

Theorem B.1 *The computation of $|\Omega_{\mathrm{I}}(\overline{G}, \underline{G})|$ is #P-complete, even when $\underline{G}$ is a connected interval graph.*

Proof. The problem is clearly in #P. It is enough to show #P-hardness. We give a reduction from the problem to count the matchings in a graph, which is known to be #P-complete [27].

First, we give a transformation of an instance (i.e., a graph) H of the matching counting problem into an instance (i.e., a pair of graphs) $\overline{G}$ and $\underline{G}$ of the interval sandwich counting problem. The construction of $\overline{G}$ is done in a similar way to the proof of Theorem 4.1; We replace every edge $\{u, v\} \in E(H)$ with a path of length three. Let $w_{u,v}$ and $w_{v,u}$ be new vertices of $\overline{G}$ which subdivide an edge $\{u, v\} \in E(H)$. Furthermore, we add an extra path a_v-b_v-v with new vertices a_v and b_v to every vertex $v \in V(H)$. This completes the construction of $\overline{G}$. Note that $|V(\overline{G})| = 3|V(H)| + 2|E(H)|$, $|E(\overline{G})| = 3|E(H)| + 2|V(H)|$, and the girth of $\overline{G}$ is at least 9. To construct $\underline{G}$, we just remove every edge of the form $\{w_{u,v}, w_{v,u}\} \in E(\overline{G})$. Figure 4 shows an example of the transformation. In Figure 4 (b), the edges of $\underline{G}$ are drawn by solid lines, and the edges of $\overline{G}$ are drawn by solid lines and dashed lines. The graph $\underline{G}$ consists of n disjoint trees. Let T_v be a connected component (i.e., a tree) of $\underline{G}$ including $v \in V(H)$. Then T_v is AT-free. Thus $\underline{G}$ is interval.

Next, we show that there exists one-to-one correspondence between the set of matchings in H and $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$. For a matching $M \subset E(H)$ of H, we define the corresponding graph $G \in \Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ as $E(G) = E(\underline{G}) \cup \{\{w_{u,v}, w_{v,u}\} \in E(\overline{G}) \setminus E(\underline{G}) \mid \{u, v\} \in M\}$. Then, G does not have any cycle. Moreover, we can observe that G is AT-free. Thus, G is interval. Conversely, every graph in $\Omega_{\mathrm{I}}(\overline{G}, \underline{G})$ does not simulataneously contain two edges $\{w_{v,u}, w_{u,v}\} \in E(\overline{G})$ and $\{w_{v,u'}, w_{u',v}\} \in E(\overline{G})$ which correspond to edges $\{v, u\} \in E(H)$ and $\{v, u'\} \in E(H)$ since it would contain an AT otherwise. Thus, for any $G \in \Omega_{\mathrm{I}}(\overline{G}, \underline{G})$, there exists a corresponding matching in H as the inverse



(a) An original graph H. (b) The reduced graphs $\overline{G}$ and $\underline{G}$.

Figure 4: An example of the transformation.

of the map above. Hence, we obtain a bijection. Thus, we showed that the computation of $|\Omega_{\mathrm{I}}(\overline{G},\underline{G})|$ is #P-hard even when $\underline{G}$ is interval.

To obtain the full theorem we transform $\overline{G}$ and $\underline{G}$ obtained from H as described above into $\overline{G}'$ and $\underline{G}'$ such that $\underline{G}'$ is connected and interval. Let G be a graph satisfying $\underline{G} \subseteq G \subseteq \overline{G}$, and we transform G into $\Phi'(G)$ defined as

$$\begin{aligned}
V(\Phi'(G)) &\stackrel{\text{def.}}{=} (V(G) \setminus \{a_v, b_v \mid v \in V(H)\}) \cup \{a\} \cup \{b\}, \text{ and} \\
E(\Phi'(G)) &\stackrel{\text{def.}}{=} E(G[V(G) \setminus \{a_v, b_v \mid v \in V(H)\}]) \cup \{\{b, v\} \mid v \in V(\Phi'(G)) \setminus \{b\}\}.
\end{aligned}$$

Note that $\Phi'(G)[V(\Phi'(G)) \setminus \{a\}] = \Phi(G)$, where Φ is the map defined in the proof of Theorem 4.1. Clearly $\Phi'(G)$ is connected and chordal. Furthermore, if G is AT-free, then $\Phi'(G)$ is AT-free, since b is a cut and dominating vertex. Inversely, if G has an AT, then w.l.o.g. let $\{a_v, u, u'\} \subset V(G)$ be an AT where $\{w_{v,u}, w_{u,v}\} \in E(G)$ and $\{w_{v,u'}, w_{u',v}\} \in E(G)$, and $\Phi'(G)$ has an AT $\{a, u, u'\} \subset V(\Phi'(G))$. Now, we define a pair of graphs $\overline{G}' \stackrel{\text{def.}}{=} \Phi'(\overline{G})$ and $\underline{G}' \stackrel{\text{def.}}{=} \Phi'(\underline{G})$ from the pair graphs $\overline{G}$ and $\underline{G}$. Then, $\underline{G}'$ is interval when $\underline{G}$ is interval, and $\Omega_{\mathrm{I}}(\overline{G},\underline{G})$ and $\Omega_{\mathrm{I}}(\overline{G}',\underline{G}')$ are in one-to-one correspondence via Φ'. Thus, we obtain the theorem. $\square$