Contracting Chordal Graphs and Bipartite Graphs to Paths and Trees*

Pinar Heggernes† Pim van 't Hof† Benjamin Lévêque‡
Christophe Paul‡

Abstract

We study the following two graph modification problems: given a graph G and an integer k, decide whether G can be transformed into a tree or into a path, respectively, using at most k edge contractions. These problems, which we call Tree Contraction and Path Contraction, respectively, are known to be NP-complete in general. We show that on chordal graphs these problems can be solved in $O(n+m)$ and $O(nm)$ time, respectively. As a contrast, both problems remain NP-complete when restricted to bipartite input graphs.

1 Introduction

Graph modification problems play a central role in algorithmic graph theory, not in the least because they can be used to model many graph theoretical problems that appear in practical applications [15, 16, 17]. The input of a graph modification problem is an n-vertex graph G and an integer k, and the question is whether G can be modified in such a way that it satisfies some prescribed property, using at most k operations of a given type. Famous examples of graph modification problems where only vertex deletion is allowed include Feedback Vertex Set, Odd Cycle Transversal, and Chordal Deletion. In problems such as Minimum Fill-In and Interval Completion, the only allowed operation is edge addition, while in Cluster Editing both edge additions and edge deletions are allowed.

Many classical problems in graph theory, such as Clique, Independent Set and Longest Induced Path, take as input a graph G and an integer k, and ask whether G contains a vertex set of size at least k that satisfies a

*This work is supported by the Research Council of Norway and by EPSRC UK grant EP/D053633/1. An extended abstract of this paper has appeared in the proceedings of the 6th Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS 2011) [10].

†Department of Informatics, University of Bergen, P.O. Box 7803, N-5020, Bergen, Norway. {pinar.heggernes,pim.vanthof}@ii.uib.no

‡CNRS, LIRMM, Université Montpellier 2, France. {leveque,paul}@lirmm.fr

certain property. Many of these problems can be formulated as graph modification problems: for example, asking whether an n-vertex graph G contains an independent set of size at least k is equivalent to asking whether there exists a set of at most $n-k$ vertices in G whose deletion yields an edgeless graph. Some important and well studied graph modification problems ask whether a graph can be modified into an *acyclic graph* or into a *path*, using at most k operations. If the only allowed operation is vertex deletion, these problems are widely known as Feedback Vertex Set and Longest Induced Path, respectively. The problem Longest Path can be interpreted as the problem of deciding whether a graph G can be turned into a path by deleting edges and isolated vertices, performing at most k deletions in total. All three problems are known to be NP-complete on general graphs [7].

We study two graph modification problems in which the only allowed operation is *edge contraction*. The edge contraction operation plays a key role in graph minor theory, and it also has applications in Hamiltonian graph theory, computer graphics, and cluster analysis [13]. The problem of contracting an input graph G to a fixed target graph H has recently attracted a considerable amount of interest, and several results exist for this problem when G or H belong to special graph classes [2, 3, 4, 11, 12, 13, 14]. The two problems we study in this paper, which we call Tree Contraction and Path Contraction, take as input an n-vertex graph G and an integer k, and the question is whether G can be contracted to a tree or to a path, respectively, using at most k edge contractions. Since the number of connected components of a graph does not change when we contract edges, the answer to both problems is "no" when the input graph is disconnected. Note that contracting a connected graph to a tree is equivalent to contracting it to an acyclic graph. Previous results easily imply that both problems are NP-complete in general [1, 4]. Very recently, it has been shown that Path Contraction and Tree Contraction can be solved in time $2^{k+o(k)} + n^{O(1)}$ and $4.98^k \cdot n^{O(1)}$, respectively [9].

We show that the problems Tree Contraction and Path Contraction can be solved on chordal graphs in $O(n+m)$ and $O(nm)$ time, respectively. It is known that Tree Contraction is NP-complete on bipartite graphs [9], and we show that the same holds for Path Contraction. To relate our results to previous work, we would like to mention that Feedback Vertex Set and Longest Induced Path can be solved in polynomial time on chordal graphs [5, 19]. However, it is easy to find examples that show that the set of trees and paths that can be obtained from a chordal graph G by at most k edge contractions might be completely different from the set of trees and paths that can be obtained from G by at most k vertex deletions. As an interesting contrast, Longest path remains NP-complete on chordal graphs [8].

2 Definitions and Notation

All the graphs considered in this paper are undirected, finite and simple. We use n and m to denote the number of vertices and edges of the *input* graph of the problem or the algorithm under consideration. Given a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. The (open) *neighborhood* of a vertex v in G is the set $N_G(v) = \{w \in V(G) \mid vw \in E(G)\}$ of neighbors of v in G. The *degree* of a vertex v in G, denoted by $d_G(v)$, is $|N_G(v)|$. The *closed neighborhood* of v is the set $N_G[v] = N_G(v) \cup \{v\}$. For any set $S \subseteq V(G)$, we write $N_G(S) = \cup_{v \in S} N_G(v) \setminus S$ and $N_G[S] = N_G(S) \cup S$. A subset $S \subseteq V(G)$ is called a *clique* of G if all the vertices in S are pairwise adjacent. A vertex v is called *simplicial* if the set $N_G[v]$ is a clique. For any set of vertices $S \subseteq V(G)$, we write $G[S]$ to denote the subgraph of G *induced* by S. If the graph $G[S]$ is connected, then the set S is said to be *connected*. We say that two disjoint sets $S, S' \subseteq V(G)$ are *adjacent* if there exist vertices $s \in S$ and $s' \in S'$ such that $ss' \in E(G)$. For any set $S \subseteq V(G)$, we write $G - S$ to denote the graph obtained from G by removing all the vertices in S and their incident edges. If $S = \{s\}$, we simply write $G - s$ instead of $G - \{s\}$.

The *contraction* of edge $e = uv$ in G removes u and v from G, and replaces them by a new vertex, which is made adjacent to precisely those vertices that were adjacent to at least one of the vertices u and v. Instead of speaking of the contraction of edge uv, we sometimes say that a vertex u is *contracted on* v, in which case we use v to denote the new vertex resulting from the contraction. Let $S \subseteq V(G)$ be a connected set. If we repeatedly contract a vertex of $G[S]$ on one of its neighbors in $G[S]$ until only one vertex of $G[S]$ remains, we say that we *contract S into a single vertex*. We say that a graph G can be *k-contracted* to a graph H, with $k \leq n - 1$, if H can be obtained from G by a sequence of k edge contractions. Note that if G can be k-contracted to H, then H has exactly k fewer vertices than G has. We simply say that a graph G can be *contracted* to H if it can be k-contracted to H for some $k \geq 0$. Let H be a graph with vertex set $\{h_1, \ldots, h_{|V(H)|}\}$. Saying that a graph G can be contracted to H is equivalent to saying that G has a so-called *H-witness structure* $\mathcal{W}$, which is a partition of $V(G)$ into *witness sets* $W(h_1), \ldots, W(h_{|V(H)|})$ such that each witness set is connected, and such that for every two $h_i, h_j \in V(H)$, witness sets $W(h_i)$ and $W(h_j)$ are adjacent in G if and only if h_i and h_j are adjacent in H. By contracting each of the witness sets into a single vertex, which can be done due to the connectivity of the witness sets, we obtain the graph H. An H-witness structure of G is, in general, not uniquely defined (see Fig. 1).

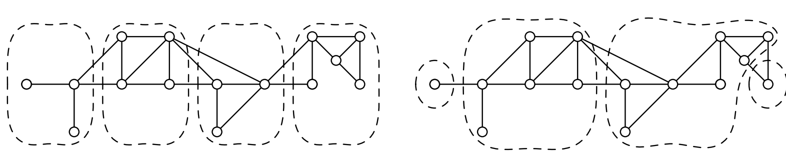

Figure 1: Two P_4-witness structures of a chordal graph.

If H is a subgraph of G and $v \in N_G(V(H))$, then we refer to the vertices in $N_G(v) \cap V(H)$ as the *H-neighbors* of v. The *distance* $d_G(u, v)$ between two vertices u and v in G is the number of edges in a shortest path between u and

v, and $diam(G) = \max_{u,v \in V(G)} d_G(u,v)$. For any two vertices u and v of a path P in G, we write uPv to denote the subpath of P from u to v in G. We use P_ℓ to denote the graph isomorphic to a path on ℓ vertices, i.e., P_ℓ is the graph with ordered vertex set $\{p_1, p_2, p_3, \ldots, p_\ell\}$ and edge set $\{p_1p_2, p_2p_3, \ldots, p_{\ell-1}p_\ell\}$. Similarly, C_ℓ denotes the graph that is isomorphic to a cycle on ℓ vertices, i.e., C_ℓ is the graph with ordered vertex set $\{c_1, c_2, c_3, \ldots, c_\ell\}$ and edge set $\{c_1c_2, c_2c_3, \ldots, c_{\ell-1}c_\ell, c_\ell c_1\}$. A graph is *chordal* if it does not contain a chordless cycle on at least four vertices as an induced subgraph.

3 Contracting Chordal Graphs

In this section we show that TREE CONTRACTION and PATH CONTRACTION can be solved in polynomial time on chordal graphs. It is easy to see that the class of chordal graphs is closed under edge contractions, and we use this observation throughout this section.

We first consider TREE CONTRACTION. For any connected graph G, we say that a tree T is *optimal* for G if G can be contracted to T, but cannot be contracted to any tree with strictly more vertices than T. A *leaf* of a tree T is a vertex that has degree 1 in T.

Lemma 1 *Let G be a connected graph on at least 2 vertices. If G has a simplicial vertex v, then G has a T-witness structure $\mathcal{W}$ for some optimal tree T, such that $W(x) = \{v\}$ for some leaf x of T.*

Proof. Suppose G has a simplicial vertex v. Let T be an optimal tree for G, and let $\mathcal{W}$ be a T-witness structure of G. Since $|V(G)| \geq 2$ and every connected graph on at least two vertices can be contracted to P_2, we know that T contains at least two vertices; note that T also has at least two leaves. Let x be the vertex of T such that $v \in W(x)$.

First suppose that $W(x) = \{v\}$. Since T is a tree on at least two vertices and $N_G[v]$ is a clique in G, all vertices of $N_G(v)$ must be contained in a single witness set $W(y)$ that is adjacent to $W(x)$. This means that y is the unique neighbor of x in T, implying that x is a leaf of T.

Now suppose v is not the only vertex in $W(x)$. Then $W(x)$ must contain at least one neighbor of v, since every witness set induces a connected subgraph of G. Since the set $N_G[v]$ is a clique of G, the vertices of $N_G[v]$ either all belong to $W(x)$, or belong to two witness sets $W(x)$ and $W(y)$, where y is a neighbor of x in T. In the first case, G can also be contracted to the tree T', obtained from T by adding a new vertex x' and an edge $x'x$ to T; we can define a T'-witness structure $\mathcal{W}'$ of G by setting $W'(x') = \{v\}$, $W'(x) = W(x) \setminus \{v\}$, and $W'(w) = W(w)$ for every $w \in V(T') \setminus \{x, x'\}$. Since T' has one more vertex than T, this contradicts the assumption that T is an optimal tree. Hence we must have the second case, i.e., the vertices of $N_G[v]$ belong to two witness sets $W(x)$ and $W(y)$ for two adjacent vertices x and y of T. Let T'' be the tree obtained from T by contracting x on y and by adding a vertex y' and an edge $y'y$. We can define a T''-witness structure $\mathcal{W}''$ of G by setting $W''(y') = \{v\}$,

$W''(y) = (W(x) \cup W(y)) \setminus \{v\}$, and $W''(w) = W(w)$ for every $w \in V(T'') \setminus \{y, y'\}$. Since $|V(T'')| = |V(T)|$, we conclude that T'' is an optimal tree of G. ∎

Before we present our algorithm for Tree Contraction on chordal graphs in Theorem 1 below, we recall a useful characterization of chordal graphs via vertex orderings. For a given graph G and an ordering $\sigma = \langle v_1, v_2, \ldots, v_n \rangle$ of its vertices, we denote by G_i the graph $G[\{v_i, v_{i+1}, \ldots, v_n\}]$. Such an ordering σ of the vertices in $V(G)$ is called a *perfect elimination ordering (peo)* of G if v_i is simplicial in G_i, for $1 \leq i \leq n$. A graph is chordal if and only if it has a peo [6]. Chordal graphs can be recognized in linear time and a peo can be computed in linear time as well [18]. We denote by $\sigma - v_i$ the ordering which is obtained by simply removing v_i from σ and keeping the ordering of all other vertices, and we define $\sigma - S$ analogously for a vertex set S. The following lemma will allow us to implement our algorithm for Tree Contraction on chordal graphs in linear time.

Lemma 2 *Let G be a chordal graph with peo $\sigma = \langle v_1, v_2, \ldots, v_n \rangle$. Let v_iv_j be an edge of G such that $i < j$. Let G' be the graph obtained from G by contracting v_i on v_j. Then $\sigma - v_i$ is a peo of G'.*

Proof. It suffices to show that v_p is simplicial in G'_p, for $1 \leq p \leq n$ and $p \neq i$. Observe first that, since v_i is simplicial in G_i and is adjacent to v_j, every neighbor of v_i in G_i is also a neighbor of v_j in G_i. Consequently, for every vertex v_p such that $p > i$, $N_{G_p}(v_p) = N_{G'_p}(v_p)$, and v_p is simplicial in G'_p, since it is simplicial in G_p. Let us consider $p < i$. Since v_p is simplicial in G_p, its neighborhood in G_p is a clique. If v_p is not adjacent to v_i, or it is adjacent to both v_i and v_j, then it clearly remains simplicial in G'_p. If v_p is adjacent to v_i and not to v_j in G, then its neighborhood in G'_p is the same as in G_p, with the exception that v_j replaces v_i. Since v_j inherits all neighbors of v_i, the neighborhood of v_p is a clique in G'_p, and hence v_p is simplicial in G'_p. ∎

Theorem 1 Tree Contraction *can be solved in $O(n + m)$ time on chordal graphs.*

Proof. Before presenting our algorithm for Tree Contraction on chordal graphs, we first make some observations. Let v be a simplicial vertex of a connected chordal graph G on at least two vertices. Let G' denote the graph obtained from G by first contracting $N_G(v)$ into a single vertex w, and then removing v from the graph. Note that G' is a connected chordal graph. Let T' be an optimal tree for G', and let $\mathcal{W}'$ be a T'-witness structure of G'. Let $W'(x) \in \mathcal{W}'$ be the witness set containing vertex w for some vertex $x \in V(T')$. Let T be the tree obtained from T' by adding a new vertex y and an edge xy to T'. We claim that T is an optimal tree for G.

By Lemma 1, G has an optimal tree T^* and a T^*-witness structure $\mathcal{W}^*$ such that v is the only vertex in some witness set $W^*(a)$ of $\mathcal{W}^*$ for some leaf a of T^*. It is clear that all the neighbors of v must belong to the witness set $W^*(b)$

of $\mathcal{W}^*$, where b is the unique neighbor of a in T^*. This means that G' can be contracted to the tree $T^* - a$. Since T' is an optimal tree of G', this implies that $|V(T')| \leq |V(T^* - a)| = |V(T^*)| - 1$, or equivalently $|V(T^*)| \geq |V(T')| + 1$. Since $|V(T)| = |V(T')| + 1$, we have that $|V(T^*)| \geq |V(T)|$. On the other hand, since G can be contracted to T as well as to T^*, and T^* is an optimal tree for G, we have $|V(T^*)| \leq |V(T)|$. Hence $|V(T^*)| = |V(T)|$, which implies that T is an optimal tree for G.

Now let (G, k) be an instance of Tree Contraction, where G is a chordal graph. If G is disconnected, then (G, k) is a trivial "no"-instance. If $|V(G)| = 1$, then G is already a tree, so we output "yes" if $k \geq 0$ and "no" otherwise. Now suppose G is connected and $|V(G)| \geq 2$. The arguments in the first two paragraphs of the proof show that we can contract G to an optimal tree for G as follows. We repeatedly find a simplicial vertex v, contract its neighborhood into a single vertex, and remove v from the graph. We continue this process until we have removed all vertices. By applying all the edge contractions that have been performed during this procedure to the original graph G, we find an optimal tree for G. Let $\sigma = \langle v_1, v_2, \ldots, v_n \rangle$ be a peo of G, and let v_j be the neighbor of v_1 with the largest index. We pick v_1 to be the simplicial vertex we start with, and we choose v_j to be the vertex on which every vertex of $N_G(v_1)$ is contracted. Let G' be the resulting graph after this operation. By repeatedly applying Lemma 2, we find that $\sigma' = \sigma - (N_G(v_1) \setminus \{v_j\})$ is a peo of G'. Hence the first vertex in $\sigma' - v_1$ is a simplicial vertex of $G' - v_1$. Consequently, we can pick this vertex as our next simplicial vertex, and repeat the process.

For the running time of the algorithm, consider the following implementation. First we compute a peo $\sigma = \langle v_1, v_2, \ldots, v_n \rangle$ of G in $O(n + m)$ time. Then we mark all vertices in $N_G(v_1) \setminus \{v_j\}$, where v_j is the vertex of $N_G(v_1)$ with the largest index. We now iterate over all values of i from 2 to $n - 1$, and proceed at each iteration i as follows. If v_i is marked, then we continue with the next iteration $i + 1$. If v_i is not marked, then we mark all neighbors of v_i in G_i, except the neighbor with the highest index. If the neighbor with the highest index is already marked, then all neighbors of v_i in G_i will become marked. In the end, the unmarked vertices are the ones that become the vertices of the optimal tree resulting from the algorithm above. Hence it suffices to check whether we have at least $n - k$ unmarked vertices. Since there are $O(n)$ iterations and each iteration i requires $O(d_G(v_i))$ steps, the running time follows. ∎

Note that the problem of contracting a connected chordal graph to a tree is equivalent to the problem of contracting a connected chordal graph to a bipartite graph. The "reverse" problem of contracting a bipartite graph to a connected chordal graph is equivalent to the problem of contracting a bipartite graph to a tree. It turns out that this problem is NP-complete, as we will see in the next section.

We now turn our attention to Path Contraction on chordal graphs. The following observation is due to Levin, Paulusma and Woeginger [13].

Observation 1 ([13]) *Let $\mathcal{W}$ be an H-witness structure of a graph G. Let u and v be two vertices of G and let x and y be two vertices of H such that $u \in W(x)$ and $v \in W(y)$. Then $d_G(u,v) \geq d_H(x,y)$.*

Observation 1 immediately implies that a graph G cannot be contracted to a path of length more than $diam(G)$. We now show that if G is a connected chordal graph, then G can be contracted to a path of length $diam(G)$. Note that this is not the case for every connected graph: for example, the graph C_ℓ has diameter $\lfloor \ell/2 \rfloor$, but cannot be contracted to a path of length more than 1.

Theorem 2 *Every connected chordal graph G can be contracted to a path of length $diam(G)$.*

Proof. Let u and v be two vertices of a connected chordal graph G such that $d_G(u,v) = diam(G)$, and let P be a shortest path from u to v. We show that G can be contracted to P. Since G is chordal, it has a simplicial vertex w. If $w \notin V(P)$, then we contract w on one of its neighbors. Since the neighborhood of w is a clique, this is equivalent to deleting w. Observe that a simplicial vertex cannot belong to a shortest path between two other vertices. Hence no shortest path between u and v contains w, and thus all shortest paths between u and v are unchanged after this operation. If $w \in V(P)$, then w is either u or v, as all other vertices on P have two non-adjacent neighbors on P, and are therefore not simplicial. Since $N_G(w)$ is a clique in G, any shortest path between u and v contains exactly one vertex of $N_G(w)$. Let x be the only vertex in $N_G(w) \cap V(P)$. We contract every vertex of $N_G(w)$ on x. After this operation, all shortest paths between w and the other endpoint of P are preserved. After the contraction of $N_G(w)$, we delete w from G. In each of the described two cases, the resulting graph G' is chordal and has at most $n-1$ vertices. We can thus repeat this procedure until the graph is empty. Applying the edge contractions that are defined by this procedure on the original graph G will result in P, as no vertex of P is ever contracted on another vertex, and no chords are formed between non-consecutive vertices of P. ∎

Corollary 1 Path Contraction *can be solved in $O(nm)$ time on chordal graphs.*

Proof. Let (G,k) be an instance of Path Contraction, where G is a chordal graph. If G is disconnected, then (G,k) is a trivial "no"-instance. Suppose G is connected. By Theorem 2, G can be k-contracted to a path if and only if $k \geq n - diam(G)$. Hence, in order to solve Path Contraction, we only need to determine the diameter of G. Unfortunately, no faster algorithm for computing the diameter is known for chordal graphs compared to arbitrary graphs. Hence we resort to the straightforward algorithm of running a breadth first search n times, each time from a different vertex of G. As breadth first search has a running time of $O(n+m)$, we get a total $O(nm)$ running time. ∎

4 Contracting Bipartite Graphs

In this section we show that Path Contraction is NP-complete when restricted to the class of bipartite graphs. We first show how previous work implies that the same holds for Tree Contraction.

The Red-Blue Domination problem takes as input a bipartite graph $G = (A, B, E)$ and an integer t, and asks whether there exists a subset of at most t vertices in B that dominates A. This problem is equivalent to Set Cover and Hitting Set, and is therefore NP-complete [7]. Heggernes et al. [9] give a polynomial-time reduction from Red-Blue Domination to Tree Contraction. Since the graph G' in the constructed instance of Tree Contraction is bipartite, they implicitly proved the following result.

Theorem 3 Tree Contraction *is NP-complete on bipartite graphs.*

We now show that Path Contraction also remains NP-complete when restricted to bipartite graphs.

Theorem 4 Path Contraction *is NP-complete on bipartite graphs.*

Proof. We first introduce some additional terminology. A *hypergraph* H is a pair $(Q, \mathcal{S})$ consisting of a set $Q = \{q_1, \dots, q_n\}$, called the *vertices* of H, and a set $\mathcal{S} = \{S_1, \dots, S_m\}$ of nonempty subsets of Q, called the *hyperedges* of H. A 2-*coloring* of a hypergraph $H = (Q, \mathcal{S})$ is a partition (Q_1, Q_2) of Q such that $Q_1 \cap S_j \neq \emptyset$ and $Q_2 \cap S_j \neq \emptyset$ for $j = 1, \dots, m$. The Hypergraph 2-Colorability problem is to decide whether a given hypergraph has a 2-coloring. This problem, also known as Set Splitting, is NP-complete, and it remains NP-complete when we assume that H has at least two hyperedges and $Q \in \mathcal{S}$ (see for example [4]).

We now prove that the problem of contracting a bipartite graph to P_6 is NP-complete, using a reduction from Hypergraph 2-Colorability. Let $H = (Q, \mathcal{S})$ be a hypergraph with $Q = \{q_1, \dots, q_n\}$ and $\mathcal{S} = \{S_1, \dots, S_m\}$, and assume that $|\mathcal{S}| \geq 2$ and $S_m = Q$. The *incidence graph* of H is the bipartite graph with vertex set $Q \cup \mathcal{S}$ and an edge between a vertex $q \in Q$ and $S \in \mathcal{S}$ if and only if $q \in \mathcal{S}$; note that every vertex of the incidence graph is labeled with the name of the vertex or hyperedge of H it corresponds to. We create a graph G from the incidence graph of H as follows. First we add four new vertices s_1, s_2, s_1', s_2' and a copy $\mathcal{S}' = \{S_1', \dots, S_m'\}$ of $\mathcal{S}$, such that $S_i' = S_i$ for every $1 \leq i \leq m$. Then we add the following edges:

- $S_i' q_j$ if and only if $q_j \in S_i$;
- $S_i S_j'$ for every $1 \leq i, j \leq m$;
- $s_2 S_i$ for every $1 \leq i \leq m$;
- $s_2' S_i'$ for every $1 \leq i \leq m$;
- $s_1 s_2$ and $s_1' s_2'$.

Finally, for every $S_i \in \mathcal{S}$ and $q_j \in Q$ we subdivide the edge $S_i q_j$ by replacing it with a path $S_i t_{i,j} q_j$. Let $T = \{t_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$.

The constructed graph G is bipartite, as assigning color 1 to the vertices in $\{s_1, s_2'\} \cup \mathcal{S} \cup Q$ and color 2 to the vertices in $\{s_2, s_1'\} \cup T \cup \mathcal{S}'$ yields a proper 2-coloring of G.

We claim that H has a 2-coloring if and only if G can be contracted to P_6. Suppose H has a 2-coloring, and let (Q_1, Q_2) be a 2-coloring of H. We define a P_6-witness structure $\mathcal{W}$ of G as follows. Let $W(p_1) = \{s_1\}$, $W(p_2) = \{s_2\}$, $W(p_3) = \mathcal{S} \cup T \cup Q_1$, $W(p_4) = \mathcal{S}' \cup Q_2$, $W(p_5) = s_2'$, and $W(p_6) = s_1'$. Since (Q_1, Q_2) is a 2-coloring of H, every vertex $S_i \in \mathcal{S}$ has at least one neighbor $t_{i,k}$ which is adjacent to some $q_k \in Q_1$, and at least one neighbor $t_{i,\ell}$ adjacent to some $q_\ell \in Q_2$. Since $\mathcal{S}'$ is a copy of $\mathcal{S}$, every vertex in $\mathcal{S}'$ has at least one neighbor in Q_1 and at least one neighbor in Q_2. This, together with the observation that the sets $S_m \cup \{t_{m,j} \mid 1 \leq j \leq n\} \cup Q_1 \subset W(p_3)$ and $S_m' \cup Q_2 \subset W(p_4)$ are both connected, implies that the witness sets $W(p_3)$ and $W(p_4)$ are connected. It is clear that contracting each of the witness sets $W(p_i)$ into a single vertex yields the graph P_6.

To prove the converse statement, assume that G can be contracted to P_6, and let $\mathcal{W}$ be a P_6-witness structure of G. The vertices s_1 and s_1' are the only two vertices of G that have distance at least 5. Hence, as a result of Observation 1, we must have $W(p_1) \cup W(p_6) = \{s_1, s_1'\}$. Without loss of generality, let $W(p_1) = \{s_1\}$ and $W(p_6) = \{s_1'\}$. Again by Observation 1 and by the definition of a witness structure, we also know that $W(p_2) = \{s_2\}$, $W(p_5) = \{s_2'\}$, and $\mathcal{S} \subseteq W(p_3)$ and $\mathcal{S}' \subseteq W(p_4)$. Let $Q_1 = W(p_3) \cap Q$ and $Q_2 = W(p_4) \cap Q$. Since the witness set $W(p_4)$ is connected by definition, every vertex in $\mathcal{S}'$ must be adjacent to at least one vertex in Q_2. Similarly, the fact that $W(p_3)$ is connected implies that, for every vertex $S_i \in \mathcal{S}$, there must be a vertex $q_j \in Q_1$ such that both $t_{i,j}$ and q_j are in $W(p_3)$. As $\mathcal{S}'$ is a copy of $\mathcal{S}$, this implies that (Q_1, Q_2) is a 2-coloring of H.

Recall that G is bipartite. Hence we have proved that the problem of deciding whether a bipartite graph can be contracted to P_6 is NP-complete. For any fixed $\ell > 6$, we can prove that the problem of contracting a bipartite graph to P_ℓ is NP-complete by adding a path of length $\ell - 6$ to the graph G, making exactly one of its end vertices adjacent to the vertex s_1 in G, and slightly modifying the arguments accordingly. This, together with the observation that a graph G can be k-contracted to a path if and only if G can be contracted to P_{n-k}, proves the theorem. ∎

5 Concluding Remarks

In the introduction, we mentioned the relationship between the problems Tree Contraction and Path Contraction and their vertex-deletion variants Feedback Vertex Set and Longest Induced Path. We would like to point out that the minimum number of edges that needs to be contracted to contract a graph G to a tree or a path might differ considerably from the minimum number of vertices or edges that needs to be deleted to obtain this goal.

In order to see this, let G_ℓ be the graph obtained from P_ℓ by adding a vertex x and making this vertex adjacent to all the vertices of the path, for any $\ell \geq 2$. Observe that G_ℓ can be transformed into a tree or a path by deleting just one vertex, namely x. The minimum number of edges that needs to be deleted to transform G_ℓ into a tree or a path is $\ell - 1$. The longest path G_ℓ can be contracted to is P_2, and it takes $\ell - 1$ edge contractions to contract G_ℓ into P_2. On the other hand, G_ℓ can be transformed into a star (with centre x) by contracting no more than $\lfloor \ell/2 \rfloor$ edges.

The class of *interval* graphs is a well known and intensively studied subclass of chordal graphs, with numerous applications in different fields. What is the computational complexity of the problem of deciding whether or not a chordal graph can be contracted to an interval graph using at most k edge contractions?

Acknowledgements

The authors would like to thank Daniel Lokshtanov, Daniël Paulusma, and Yngve Villanger for fruitful discussions.

References

[1] T. Asano and T. Hirata. Edge-contraction problems. Journal of Computer and System Sciences 26: 197–208, 1983.

[2] R. Belmonte, P.A. Golovach, P. Heggernes, P. van 't Hof, D. Paulusma, and M. Kamiński. Finding contractions and induced minors in chordal graphs via disjoint paths. Algorithmica, to appear. doi:10.1007/s00453-013-9748-5.

[3] R. Belmonte, P. Heggernes, and P. van 't Hof. Edge contractions in subclasses of chordal graphs. Discrete Applied Mathematics 160(7-8): 999–1010, 2012.

[4] A. E. Brouwer and H. J. Veldman. Contractibility and NP-completeness. Journal of Graph Theory 11: 71–79, 1987.

[5] F. V. Fomin and Y. Villanger. Finding induced subgraphs via minimal triangulations. *Proceedings of STACS 2010*, Dagstuhl, 2010, 383–394.

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

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman, 1979.

[8] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Second edition. Annals of Discrete Mathematics 57, Elsevier, 2004.

[9] P. Heggernes, P. van 't Hof, B. Lévêque, D. Lokshtanov, and C. Paul. Contracting graphs to paths and trees. Algorithmica, to appear. doi:10.1007/s00453-012-9670-2

[10] P. Heggernes, P. van 't Hof, B. Lévêque, and C. Paul. Contracting chordal graphs and bipartite graphs to paths and trees. *Proceedings of LAGOS 2011*, Electronic Notes in Discrete Mathematics 37: 87–92, 2011.

[11] P. van 't Hof, M. Kamiński, D. Paulusma, S. Szeider, and D. M. Thilikos. On graph contractions and induced minors. Discrete Applied Mathematics 160(6): 799–809, 2012.

[12] M. Kamiński, D. Paulusma, and D. M. Thilikos. Contractions of planar graphs in polynomial time. *Proceedings of ESA 2010*, LNCS 6346: 122–133, Springer, 2010.

[13] A. Levin, D. Paulusma, and G. J. Woeginger. The computational complexity of graph contractions I: Polynomially solvable and NP-complete cases. Networks 51: 178–189, 2008.

[14] A. Levin, D. Paulusma, and G. J. Woeginger. The computational complexity of graph contractions II: Two tough polynomially solvable cases. Networks, 52: 32–56, 2008.

[15] F. Mancini. *Graph modification problems related to graph classes*. Ph.D. thesis, University of Bergen, Norway, 2008.

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

[17] R. Sharan. *Graph Modification Problems and their Applications to Genomic Research*. Ph.D. thesis, Tel-Aviv University, Israel, 2002.

[18] 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: 566–579, 1984.

[19] M. Yannakakis and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. Information Processing Letters 24: 133–137, 1987.