The Parameterized Landscape of Labeled Graph Contractions

Manuel Lafond
Department of Computer Science, University of Sherbrooke, Canada

Bertrand Marchand
Department of Computer Science, University of Sherbrooke, Canada

Abstract

In this work, we study the problem of computing a *maximum common contraction* of two vertex-labeled graphs, i.e. how to make them identical by contracting as little edges as possible in the two graphs. We study the problem from a parameterized complexity point of view, using parameters such as the maximum degree, the degeneracy, the clique-width or treewidth of the input graphs as well as the number of allowed contractions. We put this complexity in perspective with that of the *labeled contractibility problem*, i.e determining whether a labeled graph is a contraction of another. Surprisingly, our results indicate very little difference between these problems in terms of parameterized complexity status. We only prove their status to differ when parameterizing by both the degeneracy and the number of allowed contractions, showing W[1]-hardness of the maximum common contraction problem in this case, whereas the contractibility problem is FPT.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized complexity - contractions - labels - widths

Digital Object Identifier 10.4230/LIPIcs.WADS.2025.42

Related Version *Full Version*: https://arxiv.org/abs/2502.16096

1 Introduction

Graphs are used as an abstract model in a wide variety of applicative fields. For instance, in bioinformatics, they can represent structured RNAs [40, 45, 36, 8], evolutionary histories [26], or interaction networks [31]. In several applications, a common computational task is the quantitative *comparison* of graphs, in order to underline a common structure. For example, the comparison of *phylogenetic networks* (graphs representing evolutionary histories) has a rich line of work in evolutionary bioinformatics [13, 37, 26, 11, 12, 22, 33]. Likewise, the *network alignment problem* has been formulated to compare biological interaction networks [17, 19, 28]. As for the comparison of molecules, it has motivated work on the *maximum common subgraph* problem [1, 2, 18]. The *graph edit distance* [23] was also studied for this purpose, especially in a context of image processing and machine learning [43, 44].

The philosophy of the graph edit distance is to compare two input graphs by counting the number of "edit operations" required to transform one graph into the other. This edition process can amount to "reducing" the first graph to a "maximum common reduced graph" before applying "augmenting" operations to get the second graph. For instance, the computation of the graph edit distance, given its use of the vertex deletions and insertions, amounts to the computation of a maximum common subgraph under a special cost function [10]. Following this philosophy, the *edge contraction* operation has been used to define one of the most widely-used distance on evolutionary trees, called the Robinson-Foulds distance [41], whose computation finds a *maximum common contraction* of the two trees. Recently, this approach was generalized to the comparison of phylogenetic networks [37], which are directed graphs representing evolution. However, as far as the authors know, the

19th International Symposium on Algorithms and Data Structures (WADS 2025).
Editors: Pat Morin and Eunjin Oh; Article No. 42; pp. 42:1–42:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

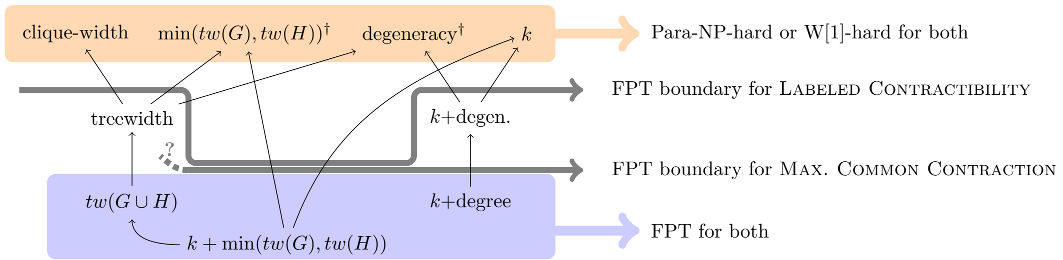

Figure 1 Illustration of our results on the compared parameterized complexity of Labeled Contractibility and Maximum Labeled Common Contraction. Results marked with a † are derived from the litterature [37, 9]. The clique-width, degeneracy and treewidth are to be understood as the maximum value of these parameters on the two input graphs. An arrow from (1) to (2) indicates that bounding (1) implies bounding (2).

problem of computing a maximum common contraction between two undirected graphs has received little to no attention from an algorithmic perspective. There are several deterring factors that may explain this. First, if the graphs are unlabeled, then determining whether no contractions are required is equivalent to checking whether they are isomorphic, a notoriously difficult problem to tackle. To make matters worse, even if the input graphs are two (unlabeled) trees, determining whether one is a contraction of another is NP-hard [38]. On the other hand, this hardness does not apply to *labeled* trees, where each vertex has a unique identifier, since polynomial-time algorithms exist (even just *leaf-labeled* trees as in [41]).

In this article, we study the Maximum Common Labeled Contraction, i.e. the problem of computing a maximum-size common contraction of two fully-labeled graphs, from the perspective of *parameterized complexity*. The input graphs are uniquely labeled (two distinct vertices of a graph have distinct labels), but each graph may have labels not present in the other. We use structural parameterizations, such as the treewidth or clique-width of the input graphs, but also the maximum degree, degeneracy or number of contractions. We also study the *contractibility* problem (given two labeled graphs, is one a contraction of the other ?) and compare the parameterized complexity aspects of both problems. Our results are summarized on Figure 1, and outline little difference in complexity between both problems. We fully establish the location of the barrier between FPT and W[1]-hardness/para-NP-hardness for our chosen parameters – with the notable exception of the maximum common contraction problem when both graphs have bounded treewidth, which remains open. We see that the two problems behave similarly, apart from the parameter $k + \delta$, with δ the degeneracy.

This paper is organized as follows. After an overview of related works, preliminary notions and results are given in Section 2. Then, Section 3 proves our results regarding the contractibility problem, and Section 4 those on the maximum common contraction problem. Due to space constraints, some proofs have been deferred to the Appendix, or to the full online version (`https://arxiv.org/abs/2502.16096`).

Related works. A rich related line of work is the study of the H-Contractibility problem on undirected, unlabeled graphs. It consists in deciding whether an input graph G can be transformed into a graph isomorphic to H using only edge contractions. In a seminal article [9], H-contractibility was proven NP-hard with $H = P_4$, the path on 4 vertices. Follow-up works [34, 35] gave characterizations of the graphs H such that H-Contractibility is NP-hard. On the positive side, H-Contractibility was proven polynomial-time solvable if

G is chordal [7, 5], or if H is planar (but still fixed) [29]. A polynomial algorithm for H of bounded degree and G of bounded treewidth was also given in [38], in addition to the proof that deciding if a tree is a contraction of another is NP-hard (already mentioned above). Variations of the problem, for instance deciding whether k contractions are enough to make a graph fall into a given *class*, have also been studied, including under the parameterized complexity point of view. These include the contractibility into *grid graphs* [42], *bipartite graphs* [24], *paths/trees* [25, 30] or graphs of *bounded degree* [6].

As mentioned in the introduction, maximum common contractions were studied on *phylogenetic networks* [37], which are directed acyclic graphs with a single root and labeled leaves. In [37], the NP-hardness of the problem is proven, and a polynomial-time tractable sub-case is identified, but the parameterized complexity of the problem was not explored. Nonetheless, hardness proofs from [37] as well as [9] can be easily adapted to fully-labeled undirected graphs. It gives the following starting points: contractibility into a P_4 graph is NP-hard on labeled graphs [9], the maximum common contraction problem on labeled graphs is NP-hard if both input graphs have constant degree (and thus also constant degeneracy) [37, Theorem 8], and the labeled contractibility problem is NP-hard on graphs of bounded degeneracy (also in [37, Theorem 8], as although the reduction is to the maximum common contraction problem, it falls back to a contractibility instance of bounded degeneracy).

2 Preliminary notions

For an integer n, we may use the notation $[n] = \{1, 2, \dots, n\}$. We denote the vertex set and edge set of a graph G by $V(G)$ and $E(G)$, respectively. The subgraph of G induced by a subset of vertices $X \subseteq V(G)$ is denoted $G[X]$. We write $G - X$ for the graph $G[V(G) \setminus X]$, and if $X = \{x\}$ has a single element we may write $G - x$. For a vertex u, $N_G(u)$ is the set of neighbors of u in G, and $N_G[u] = N_G(u) \cup \{u\}$. We say that two disjoint subsets $X, Y \subseteq V(G)$ are *adjacent* if there exists an edge between an element of X and an element of Y. The maximum degree of G is denoted $\Delta(G)$. The *degeneracy* of G is $\delta(G)$, which is the smallest integer such that every subgraph of G has a vertex of degree at most $\delta(G)$.

Given two graphs G and H, we write $G = H$ if and only if $V(G) = V(H)$ and $E(G) = E(H)$. This differs from the more common notion of isomorphism: the vertex sets of the two graphs must consist of precisely the same elements, and edges must connect the same pairs of vertices. We can thus view the vertices of G and H as uniquely labeled, and equality requires that vertices with the same labels share the same edges in both graphs. However, we prefer to avoid labeling functions, and instead compare the vertices and edges directly. Note that sets of labels may both intersect ($V(G) \cap V(H) \neq \emptyset$, in fact no common contraction is possible otherwise) and contain elements not present in the other graph ($V(G) \Delta V(H) \neq \emptyset$).

Labeled contractions of graphs. Given a graph G and an edge uv of G, the *labeled contraction* (u, v) is an operation that transforms G as follows: (1) add an edge between u and every vertex of $N_G(v) \setminus N_G[u]$; (2) remove v and all its incident edges. The graph obtained from G after the labeled contraction (u, v) is denoted $G/(u, v)$. Note that because the sets of vertices of our graphs matter, contracting (u, v) is different than contracting (v, u), that is, $G/(u, v) \neq G/(v, u)$ (as these two graphs remove a different vertex). Do note that $G/(u, v)$ and $G/(v, u)$ are isomorphic in the traditional sense.

A *labeled contraction sequence* S on G is a list of vertex pairs $S = ((u_1, v_1), \dots, (u_k, v_k))$ such that, for each $i \in \{0, 1, \dots, k-1\}$, if G_i is the graph obtained after the application of the first i labeled contractions (with $G_0 = G$), then $u_{i+1}v_{i+1}$ is an edge of G_i and G_{i+1} is obtained by applying the labeled contraction (u_{i+1}, v_{i+1}) on G_i. We denote the resulting graph G_k as G/S (if S cannot be applied on G, then G/S is undefined). The number of pairs in S is denoted $|S|$, which here is k.

We say that a graph H is a *labeled contraction of* G if there exists a labeled contraction sequence S such that $G/S = H$. Again, we emphasize that we require equality here. If a graph M is a labeled contraction of both G and H, it is called a *common labeled contraction*. A *maximum common labeled contraction* of G and H is a common labeled contraction with a maximum number of vertices. Equivalently, it is the result of applying a minimum number of labeled contractions on G and H. Formally, we are interested in the following problem.

> MAXIMUM COMMON LABELED CONTRACTION
> **Input:** two graphs G and H, integer k.
> **Question:** Are there labeled contraction sequences S_1 and S_2 such that $G/S_1 = H/S_2$ and $|S_1| + |S_2| \leq k$?

This can be viewed as the computation of a graph distance, since $|S_1| + |S_2|$ give a minimum number of contractions and *expansions* required to transform G into H, where expansions are the reverse of contractions. Note also that the problem is equivalent to asking, given G and H whether there is a common contraction M such that $2|V(M)| \geq |V(G)| + |V(H)| - k$.

An "easier" variant is when the common contraction must be the smaller input graph.

> LABELED CONTRACTIBILITY
> **Input:** two graphs G and H with $V(H) \subseteq V(G)$.
> **Question:** is H a labeled contraction of G ?

▶ Remark 1. In the remainder, all contractions are labeled, so we may simply write "contraction" instead of "labeled contraction".

MAXIMUM COMMON LABELED CONTRACTION generalizes LABELED CONTRACTIBILITY. Indeed, H is a contraction of G if and only if G and H have a common contraction of size at least $|V(H)|$, or using at most $|V(G) \setminus V(H)|$ contractions. Therefore, our hardness results that apply to LABELED CONTRACTIBILITY transfer to MAXIMUM COMMON LABELED CONTRACTION. Note that in either problem, we do not require the input graphs to be connected. In LABELED CONTRACTIBILITY, the connected components of G can be matched uniquely to those of H using common vertices. In MAXIMUM COMMON LABELED CONTRACTION, one could proceed as follows: for each connected component C_1 from G and C_2 from H, compute a maximum common contraction between $G[C_1]$ and $G[C_2]$, and assign a weight on $\{C_1, C_2\}$ equal to the size of the common contraction ($-\infty$ if none exists). Then, find a maximum-weight perfect matching in the resulting edge-weighted bipartite graph. Hence, disconnected graphs only affect the complexity by a potential polynomial factor.

Witness structures. When applying a contraction sequence $G/S = H$, each vertex that gets removed from G can be seen as "subsumed" by exactly one of the remaining vertices in H. It is common to replace the result of a contraction sequence with the vertex partition of $V(G)$ that groups subsets of vertices that end up in the same vertex. Such a partition *witnesses* the existence of a contraction sequence from G to H, formalized as follows.

Let G, H be two graphs with $V(H) \subseteq V(G)$. Let $\mathcal{W} = \{W_1, \ldots, W_{|V(H)|}\}$ be a partition of $V(G)$ with $|V(H)|$ non-empty sets. For $u \in V(G)$, we denote by $\mathcal{W}(u) \in \mathcal{W}$ the set of $\mathcal{W}$ that contains u. We say that $\mathcal{W}$ is a *witness structure of* G *into* H if all the following conditions are satisfied (see Figure 2):

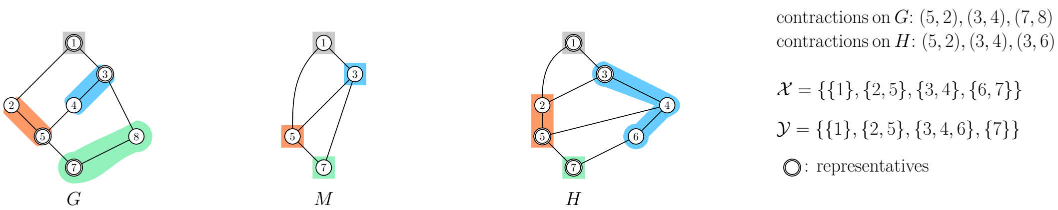

Figure 2 Example of two graphs G, H and a common contraction M. On the graphs, the shaded areas represent the witness sets. The witness structures of respectively G and H into M are called $\mathcal{X}$ and $\mathcal{Y}$, and are given on the right. Note that $V(G) \neq V(H)$, as allowed by our problem statement.

- For every $W_i \in \mathcal{W}$, the induced subgraph $G[W_i]$ is connected.
- Every $W_i \in \mathcal{W}$ contains exactly one vertex of H. This vertex is called the *representative* of W_i (in H).
- For every distinct $u, v \in V(H)$, $uv \in E(H)$ if and only if $\mathcal{W}(u)$ and $\mathcal{W}(v)$ are adjacent.

We note that there is a natural bijection between $\mathcal{W}$ and $V(H)$ formed by the representatives. The following equivalence is well-known in the case of unlabeled graphs [9, 34], and is easily seen to hold in labeled graphs. An example of two graphs, a common contraction, and the corresponding witness structures is given on Figure 2.

▶ **Observation 2.** *Let G, H be graphs. Then H is a contraction of G if and only if there exists a witness structure of G into H.*

Note that each contraction reduces the number of vertices by exactly one, and so if $\mathcal{W}$ is a witness structure of G into H, then the number of contractions needed is $\sum_{W_i \in \mathcal{W}}(|W_i| - 1)$. To finish, we prove that contractions within a witness set can essentially be done in any order, as long as the representative stays. The idea is that if we apply any contraction within a witness set first, we can update the affected witness set and get a witness structure for the modified graph.

▶ **Observation 3.** *Let $\mathcal{W}$ be a witness structure of G and H, and let $uv \in E(G)$ such that u, v are in the same witness set of $\mathcal{W}$. If $v \notin V(H)$, then H is a contraction of $G/(u,v)$.*

3 Labeled Contractibility

We begin by describing where the hardness barrier resides for the contractibility problem. We focus on difficult parameters first, then study those that lead to FPT algorithms.

3.1 W[1]-hardness in parameter k

We first consider the parameter k, which is the number of contractions needed to transform G into H, or equivalently $k = |V(G)| - |V(H)|$. We reduce from the well-known Multicolored Clique problem, a W[1]-hard problem [20] which we recall.

> Multicolored Clique
> **Input:** a graph $G_C = (V, E)$ along with a partition $\{V_1, \ldots, V_k\}$ of V into k sets, with each V_i called a color class.
> **Output:** does G_C contains a *multicolored clique*, i.e., a clique that contains exactly one vertex per color class?

Let $G_C = (V, E)$ be an instance of MultiColored Clique, with $V = V_1 \cup \ldots \cup V_k$ partitioned into k color classes. We assume that for every distinct $i, j \in [k]$, each vertex $v \in V_i$ has at least one non-neighbor in V_j. This is without loss of generality, as we can add an isolated vertex to each V_i without changing the answer to the instance.

Let us construct two graphs G and H from G_C. We start with the simpler H, which is obtained by copying G_C, and making each V_i a clique. More specifically

$$V(H) = V(G_C) \qquad E(H) = E(G_C) \cup \bigcup_{i \in [k]} \{uv : u \in V_i, v \in V_i, u \neq v\}.$$

To obtain G, start with a copy of H, then add k new vertices $t_1, \ldots, t_k$, which are not present in H. Then, make $\{t_1, \ldots, t_k\}$ a clique, and for each $i \in [k]$, add every possible edge between t_i and the vertices V_i in G. This concludes the construction of G and H.

▶ **Theorem 4.** *The Labeled Contractibility problem is W[1]-hard in parameter k, which is the number of contractions needed to transform one input graph into the other.*

Proof sketch. Considering the above construction, if G_C has a multicolored clique $u_1, \ldots, u_k$, then we can apply the set of contractions (u_i, t_i), for $i \in [k]$. This can only add edges between distinct u_i, u_j vertices, but those edges were already present in G (and in H) since the u_i's form a clique. In other words, this just gets rid of the t_i vertices without adding any new extra edge, and thus the resulting graph is identical to H.

Conversely, if G can be contracted into H, then we can argue that each t_i must be contracted with some $u_i \in V_i$ (contracting it with a $u_j \in V_j, j \neq i$ would add all edges between u_j and V_i, which are not in H). For $i \neq j$, if we contract (u_i, t_i) then (u_j, t_j), the edge $u_i u_j$ will be created if not already present. This must be prevented to reach H, so $u_i u_j$ must already be in G, so the u_i's contracted with the t_i's must form a clique. ◀

We observe that this problem is easily seen to be in XP when parameterized by k. Indeed, given graphs G and H, we can just try every sequence of k contractions in G that suppress a vertex of $V(G) \setminus V(H)$. There are $O(n^2)$ choices and we make at most k contractions, resulting in complexity of the form $O(n^{2k})$. Note also that since our reduction is from Multicolored Clique and it preserves the parameter k exactly, it also implies [16, Corollary 14.23] no $f(k) \cdot n^{o(k)}$ algorithm for Labeled Contractibility under the Exponential-Time Hypothesis (ETH). Therefore, the aforementioned XP algorithm is essentially optimal under ETH.

3.2 para-NP-hardness for clique-width

We recall the definition of clique-width. Each vertex of a graph can be assigned a color, and the clique-width is the minimum number of colors required to build a graph using a sequence of the following operations:

- Creation of a new vertex with color i.
- Disjoint union of two (vertex-colored) graphs.
- Adding all possible edges between vertices colored i and vertices colored j, $i \neq j$.
- Recoloring all vertices with color i to color j, where $i \neq j$.

The clique-width of a graph G is denoted $\mathrm{cw}(G)$.

We describe a reduction from Unary Bin Packing, and more specifically the variant in which we ask that every bin is filled exactly up to its capacity, and the input integers are encoded in unary (an integer a takes a bits in the input). We call it Unary Perfect Bin Packing.

> UNARY PERFECT BIN PACKING
> **Input:** Integers $a_0, \ldots, a_{n-1}$ (item sizes) encoded in unary, bin capacity C, and number of bins k, such that $\sum_{x=0}^{n-1} a_x = Ck$.
> **Output:** Is there an assignment $\phi : [0, n-1] \to [0, k-1]$ of each item to a bin such that $\forall i \in [0, k-1]$, $\sum_{j \,:\, \phi(j)=i} a_j = C$?

Note that item size and bin subscripts are indexed at 0 to simplify some calculations later on. We know from [27] that UNARY BIN PACKING is NP-hard and W[1]-hard in the number of bins k. It turns out that UNARY PERFECT BIN PACKING is also W[1]-hard in parameter k, and also NP-hard [32]. We use this problem to show that LABELED CONTRACTIBILITY is NP-hard even when both input graphs have clique-width at most 4.

Reduction description. Consider an input to UNARY PERFECT BIN PACKING, with item sizes $a_0, \ldots, a_{n-1}$, bin capacity C, and number of bins k. An illustration of the reduction is given in Figure 3. From an instance $a_0, \ldots, a_{n-1}, C, k$ of UNARY PERFECT BIN PACKING, we construct graphs G, H as follows. Both G and H contain a set of vertices $B = \{b_0, \ldots, b_{k-1}\}$, each corresponding to a bin, connected as a clique in both graphs. Again in both G and H, there is also a set of Ck vertices $D = \{d_0, \ldots, d_{Ck-1}\}$, connected as a clique in both graphs. In H, each vertex $d_j \in D$ has exactly one neighbor in B, which is b_i such that $\lfloor j/C \rfloor = i$ (thus each b_i has exactly C neighbors in D). This finishes the construction of H. In G, there also vertices $T = \{t_0, \ldots, t_{n-1}\}$ not present in H, which represent the items. The set T forms an independent set. The vertex sets T and B form a bi-clique, so every possible edge between the two sets is present. For each $t_x \in T$, we also include in G a set of vertices $A_x = \{\alpha_x^0, \ldots, \alpha_x^{a_x-1}\}$ containing a_x elements, each having t_x as a neighbor. We denote the set $A_0 \cup \ldots \cup A_{n-1}$ as A (see Figure 3), which forms an independent set and is absent from H. The set A forms a bi-clique with the vertices D in G.

For H to be a contraction of G, there must be a way for each b_i to acquire the neighbors specified by H. In particular, each b_i is connected in H to exactly C vertices of the set D (recall that C is the bin capacity). Specifically, in H there is an edge (b_i, d_j) for any $i \in [0, k-1]$ and $j \in [0, Ck-1]$ such that $\lfloor j/C \rfloor = i$. To do this, each b_i must be contracted with some t_x vertices, making b_i acquire a certain number of neighbors in A. Each such neighbor must then disappear, and to do that there must be a contraction with an element of D. Since $|A| = |D|$, each element of A is contracted with a single element of D, so b_i must have acquired exactly C neighbors into A to start with. We proceed to the proof of this idea.

▶ **Theorem 5.** *The LABELED CONTRACTIBILITY problem is NP-hard, even if the input graphs have clique-width at most 4.*

Proof sketch. The graph G actually has clique-width 3. We can first create the $G[T \cup A]$ subgraph, a forest of star trees, using two colors. We can create the B clique and then connect it to all of T using a third color, recoloring B to the color of T, then adding D and connecting it to A reusing that third color. As for H, we can incorporate each subgraph $H[\{b_i\} \cup (N_H(b_i) \cap D)]$ one at a time. We assume we already have the $b_1, \ldots, b_{i-1}$ vertices using one color, and their neighbors in D using another, and we add b_i using a new color and its neighbors in D using another new color, which allows connecting them to the previous B and D vertices (and then we recolor b_i and its neighbors in D).

To see that the generated instance is equivalent, suppose that ϕ assigns items to bins perfectly. If bin i has items $a_{i_1}, \ldots, a_{i_q}$, in G we contract $(b_i, t_{i_1}), \ldots, (b_i, t_{i_q})$. Since each t_{i_j} has a_{i_j} neighbors in A, this gives b_i exactly C neighbors from A, say $v_0, \ldots, v_{C-1}$. Then to ensure that b_i has the correct neighbors from D, we contract $(v_0, d_{iC}), (v_1, d_{iC+1}), \ldots, (v_{C-1}, d_{(i+1)C-1})$. This results in the graph H.

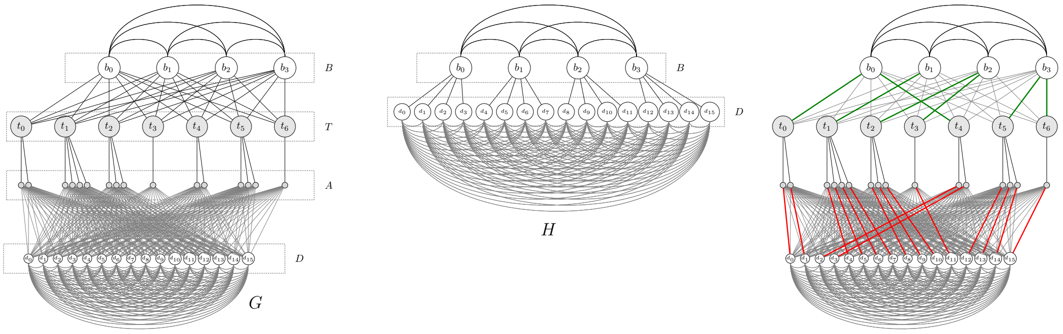

Figure 3 Illustration of the reduction from UNARY PERFECT BIN PACKING to fully-labeled contractibility, on an instance. This instance of UNARY PERFECT BIN PACKING has 4 bins (represented by the vertices labelled $b_0, \dots, b_3$), 7 items (represented by the vertices $t_0, \dots t_6$), the items have size respectively $2, 4, 3, 1, 2, 3$ and 1 as represented by their number of neighbors (without counting b_i), and the size of each bin is 4 (as represented by the number of neighbors each b_i vertex has in H). On the right, the contraction of the highlighted edges yields H. As we discuss in the proof, the green edges then encode a solution to UNARY PERFECT BIN PACKING.

Conversely, if we can contract G into H, then we can argue in terms of witness sets: $b_i \in B$ cannot be in the same witness set as $v \in A$, since b_i would become adjacent to all of D; a $t_i \in T$ cannot be with a $d_j \in D$, as d_j would become adjacent to all of B. This implies that each $t_i \in T$ is in the witness set of an element of B, and each $v \in A$ with an element of D. Moreover, each $d_j \in D$ *must* have some $v \in A$ in its witness set, as otherwise d_j becomes impossible to connect with any $b_i \in B$. Hence the witness sets containing vertices of A and D form a perfect matching. This then makes it easy to show that if some b_i has $t_{i_1}, \dots, t_{i_q}$ in its witness set, then these members of T must have had exactly $|N_H(b_i)| = C$ neighbors in D, establishing the correspondence with perfect bin assignments. ◀

3.3 An FPT algorithm in k and degeneracy

Let us start with a definition of the degeneracy. Given a total order σ of the vertices of a graph G, we denote by $N^G_{\geq,\sigma}(u)$ the neighbors of u that are after it according to σ. We define the degeneracy $\delta(G)$ of a graph as: $\delta(G) = \min_\sigma \max_{u \in V(G)} |N^G_{\geq,\sigma}(u)|$.

We call $N^G_{\geq,\sigma}(u)$ the *remaining neighborhood* of u according to σ. This denomination is to be understood in an interpretation of degeneracy as an elimination process.

▶ **Lemma 6.** *If H is obtained from G by $\leq k$ contractions, then $\delta(H) \leq \delta(G) + k$*

Proof sketch. We prove that a single contraction of an edge u, v in a graph only increases the degeneracy by at most one. Given an elimination ordering σ optimal for $\delta(G)$, we modify it to place the merged vertex (whose degree might have doubled compared to u, v) as late into σ as possible. By placing it after all of its neighbors, we make its remaining neighborhood empty, while augmenting it by at most 1 vertex for other vertices. ◀

We describe a branching algorithm for LABELED CONTRACTIBILITY that tries a bounded number of contractions at each recursion. On a recursive call that receives graphs G, H and integer k, the enumeration proceeds as described below. This algorithm is also given in pseudo-code in the appendix.

- If $G = H$ we return true. If $k < 0$, or $|V(G) \setminus V(H)| > k$, we return false.
- Otherwise, we take a vertex u of G of minimum degree. Assuming that G is the result of at most k contractions, by Lemma 6, the degree of u is at most $\delta(G) + k$.
- If u is in G but not in H, we know that u must be removed by a contraction at some point. So we branch into at most $\delta(G) + k$ ways of contracting u with one of its neighbors (which is safe by Observation 3 on the order of contractions), and decrease k by 1.
- Otherwise, u is in both G and H and must be kept. If $N_G(u) = N_H(u)$, then we may simply ignore u from now on, as further contractions on G that remove vertices outside of H will not change the neighborhood of u. We thus remove u from both graphs (note that this step may fail on the maximum common contraction problem).
- If $N_G(u) \subsetneq N_H(u)$, i.e. u has a neighbor v in H, but $v \notin N_G(u)$, then we know that some contraction must affect u or v, as otherwise the edge uv will never be created. We branch into all ways of contracting u with one of its neighbors not in H, or v with one of its neighbors not in H. (if there is no such neighbor for either of them, we return false). This branches into at most $2k$ cases, because $|V(G) \setminus V(H)| \leq k$.
- If u has a neighbor $v \in V(H)$ in G, but $v \notin N_H(u)$ (i.e. both u, v must be kept, but they share an unwanted edge) then we return false.
- Finally, if none of the above holds, then $N_H(u) \subseteq N_G(u)$, and u has a neighbor v in G with $v \notin V(H)$. This vertex v must be contracted into one of its neighbors. It cannot be contracted into a vertex of $V(H) \setminus N_H(u)$ (as $N_H(u)$ is already complete), it may therefore only be contracted into a vertex in $V(G) \setminus V(H)$(which contains $\leq k - 1$ elements when not counting v) or $N_G(u) \cup \{u\}$ (which contains $\leq \delta(G) + k + 1$ elements) We branch over these $\delta(G) + 2k$ possibilities.

▶ **Theorem 7.** *Labeled Contractibility can be solved in time $O\left((\delta(G) + 2k)^k \cdot (n + m)\right)$, where k is the number of contractions, $n = |V(G)|$, and $m = |E(G)|$.*

Proof sketch. The algorithm is a bounded search tree with two kinds of recursive calls: non-branching calls (when $N_H(u) = N_G(u)$) and branching calls involving a contraction. In the latter, we make at most $\delta(G) + 2k$ recursive calls that each decrease k by 1. ◀

4 Maximum Common Labeled Contractions

We now turn to maximum common contractions. We start with our hardness result on graphs of small degeneracy, contrasting with the positive result for the contractibility problem.

4.1 W[1]-hardness in 4-degenerate graphs

We show that Maximum Common Labeled Contraction is W[1]-hard in parameter k even on graphs of degeneracy 4. Therefore, there is little hope for an FPT algorithm in parameter $k + \delta(G)$. To give credit where is due, the reduction was initially inspired by [39] for the hardness of domination problems on graphs of bounded degeneracy, although the adaptations required for our proof makes it largely different. We will need the following lemma, which is a generalization of Observation 3.

▶ **Lemma 8.** *Let H be a contraction of a graph G, and let $\mathcal{W}$ be the witness structure of G into H. Let $R \subseteq E(G)$ be a matching of G such that, for every edge $xy \in R$, the vertices x and y are in the same witness set of $\mathcal{W}$.*

Let G' be the graph obtained after contracting every edge of R, making sure not to remove a representative vertex of $\mathcal{W}$. Then H is a contraction of G'.

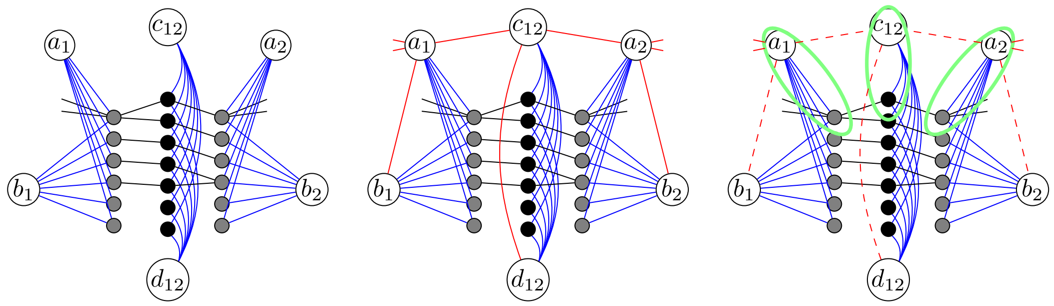

Figure 4 Left: an illustration of the vertices of and edges of G relating V_1 and V_2. Gray vertices represent V_1' and V_2', and black vertices represent Z_{12}' (the bottom two vertices in each group represent extra vertices). Some edges are colored blue only for clarity. Middle: the same subgraph but in H. The red edges are those that are unique to H. Note that a_1 has other neighbors $c_{13}, \dots, c_{1k}$, similar with a_2, but all the neighbors of b_1, b_2, c_{12}, d_{12} are shown. Right: witness sets corresponding to contractions in a solution for the forward direction (assume witness sets $\{a_1, u_1\}, \{a_2, u_2\}, \{c_{12}, z_e\}$ such that $e = u_i u_j$). Whether the red edges are present or not, this will result in the same graph.

We can now describe our reduction. Let $G_C = (V, E)$ be an instance of Multicolored Clique with $V = V_1 \cup V_2 \cup \ldots \cup V_k$ partitioned into k color classes. Construct an instance G, H of Maximum Common Labeled Contraction, starting with the description of G. The reader may refer to Figure 4. First, for each $i \in [k]$, define V_i' as a set of vertices obtained by starting with V_i, then adding $4k^3$ new vertices. We refer to elements of $V_i' \setminus V_i$ as *extra vertices*. Add to G the set of vertices $V_1' \cup \ldots \cup V_k'$. Then for each distinct $i, j \in [k]$, and for each edge edge $e = uv$ of G_C with $u \in V_i$ and $v \in V_j$, create a new vertex z_e and add the path $u - z_e - v$ (or said differently, add the edge uv to G and subdivide it, which creates z_e). We may write $z_e = z_{uv} = z_{vu}$. Denote by Z_{ij} the set of z_e vertices created at this step (we consider that $Z_{ij} = Z_{ji}$). Next, obtain the set of vertices $Z_{ij}' = Z_{ji}'$ by adding $4k^3$ new vertices to Z_{ij} (these new vertices are also called extra vertices).

Then for each $i \in [k]$, add two new vertices a_i and b_i, and make a_i and b_i adjacent to every vertex of V_i' in G. Note that a_i and b_i are not adjacent, and that the extra vertices of V_i' are only adjacent to a_i and b_i. Finally, for each distinct $i, j \in [k]$, add two vertices c_{ij} and d_{ij}, non-adjacent. We define $c_{ji} = c_{ij}$ and $d_{ji} = d_{ij}$. Make c_{ij} and d_{ij} adjacent to every vertex in Z_{ij}'. In particular, extra vertices of Z_{ij}' are only adjacent to c_{ij}, d_{ij}. This completes the construction of G. To construct H, start with a copy of G, then add the following edges: $a_i b_i$ for each $i \in [k]$; $a_i c_{ij}$, $a_j c_{ij}$ and $c_{ij} d_{ij}$ for each distinct $i, j \in [k]$; Finally, define $K := 2(k + \binom{k}{2})$. This concludes the construction.

▶ **Theorem 9.** *The* Maximum Common Labeled Contraction *is W[1]-hard in parameter k, the total number of contractions needed to achieve a common contraction, even if both input graphs have degeneracy* 4.

Proof sketch. The graphs G, H have degeneracy at most 4 since we can have an elimination order that removes every Z_{ij}' first (vertices of degree 4), then the V_i' vertices (remaining degree 2), followed by the c_{ij}, d_{ij} vertices and then the a_i and b_i vertices.

Next, if G_C has a multicolored clique $u_1, \dots, u_k$, in either graph G or H, we contract each (a_i, u_i) pair, and each $(c_{ij}, z_{u_i u_j})$ pair, as in Figure 4. Since, in G and H, the edges $u_i z_{u_i u_j}$ and $u_j z_{u_i u_j}$ exist, the effect is "collapsing" these red edges in H. Since the rest of the graphs are the same, this results in identical graphs. The other direction assumes that K contraction suffice on G and H to reach a common contraction, and produces a multicolored clique of G_C. We can argue that in G, one of a_i or b_i is incident to a contraction because of the red edge a_ib_i in H, and one of c_{ij} or d_{ij} is incident to a contraction because of $c_{ij}d_{ij} \in E(H)$. These require $K/2$ contractions in G and there must be $K/2$ contractions in H. These are all of the contractions, which are therefore a matching. Lemma 8 is used in a more involved argument to show that it is actually the a_i and c_{ij} vertices that are incident to a contraction (using b_i or d_{ij} vertices leads us to add too many new neighbors to extra vertices). Once this is established, we can argue that the contraction partners u_i of a_i and u_j of a_j must form an edge u_iu_j, because the situation from Figure 4 on the right is enforced. ◀

4.2 Parameterization by $k + \Delta$

Let G be a graph of maximum degree Δ. It is not hard to see that applying at most k contractions to G results in a graph of maximum degree $O(k\Delta)$, which we make precise in the next lemma. It then becomes relatively easy to get an FPT algorithm in $k + \Delta$.

▶ **Lemma 10.** *Let G be a graph of maximum degree $\Delta \geq 2$. Then applying $t \geq 0$ contractions to G results in a graph of maximum degree at most $\Delta + t(\Delta - 2)$.*

Moreover, this bound is tight, that is, for any t, there exists a graph G of maximum degree Δ and a sequence of t contractions that results in a graph of maximum degree $\Delta + t(\Delta - 2)$.

Proof sketch. For the upper bound, consider a witness set W from a witness structure of G into a graph H obtained after t contractions. Note that $G[W]$ has at least $|W| - 1$ edges since it is connected. After contracting W to a single vertex, its degree is at most $|W|\Delta - 2(|W| - 1)$, since the inside edges are double-counted. Using $|W| \leq t + 1$ gives the bound. The lower bound is achieved from a graph that starts as a path with $t + 1$ vertices. We add leaves adjacent to every path vertex until they have degree Δ. Contracting the whole path gives the lower bound. ◀

▶ **Proposition 11.** *The Maximum Common Labeled Contraction problem can be solved in time $O((8(k+1)\Delta)^{k+1} \cdot n)$, where k is the number of contractions, $\Delta = \max(\Delta(G), \Delta(H))$, $n = |V(G)| + |V(H)|$ and $m = |E(G)| + |E(H)|$.*

Proof sketch. Consider a recursive branching algorithm as follows. If there exists a vertex u present in one graph but no the other, we know that u must be incident to a contraction and must disappear. By Observation 3, we may do this contraction first. By Lemma 10, u has degree at most $\Delta + k(\Delta - 2)$, which we bound by $(k + 1)\Delta$ for simplicity. We branch on all the ways of contracting u with a neighbor. If no such u exists, then $V(G) = V(H)$. If $E(G) = E(H)$, we are done, so assume there is $uv \in E(G) \setminus E(H)$. We observe that u or v must be incident to some contraction, although it could be in either graph, and u/v could be the kept vertex or not. There are $8(k + 1)\Delta$ ways to branch on such contractions. Since each branch reduces the parameter k by 1, we get a recursive search tree with $(8(k + 1)\Delta)^k$ calls. We add a factor proportional to $k\Delta n$ to check all the edges at each recursion. ◀

4.3 Tractability when parameterizing by the treewidth of $G \cup H$

We now prove that Maximum Common Contraction is FPT in $tw(G \cup H)$, i.e. the treewidth of the graph with vertices $V(G) \cup V(H)$, and edges $E(G) \cup E(H)$. We remark that $tw(G \cup H)$ could be unbounded even if both G, H have bounded treewidth [3], but nonetheless we derive consequences from our result at the end of the section. We use Courcelle's theorem [15] by formulating the problem in the monadic second-order logic on graphs. More precisely, we use an extended version of Courcelle's theorem [4] allowing for *labeled edges* and *labeled vertices* if labels are taken from a finite set. Indeed, in $G \cup H$, we still need to distinguish vertices in $V(G)$ and $V(H)$, and edges in $E(G)$ and $E(H)$. Alternatively, following the terminology of [21], we are applying Courcelle's theorem on a *relational structure* (Theorem 11.37 in [21]) on $V(G) \cup V(H)$, containing two unary relations labeling $V(G)$ and $V(H)$, and two binary relations for $E(G)$ and $E(H)$. One can verify that the treewith of this relational structure (in the sense of [21, Definition 11.23]) is $tw(G \cup H)$.

▶ **Theorem 12.** *Maximum Common Conraction on fully-labeled graphs is fixed-parameter tractable in* $tw(G \cup H)$*.*

Proof. In this proof, we denote $V(G)$ by V_1, $V(H)$ by V_2, $E(G)$ by E_1 and $E(H)$ by E_2 for simplicity. We use the following formulation of the problem: we are looking for sub-sets $S_1 \subseteq E_1$ and $S_2 \subseteq E_2$ of edges, and a subset $R \subseteq V_1 \cap V_2$ of vertices ("representatives") such that S_1 and S_2 induce, in G and H respectively, connected components (the witness sets) and such that each connected component contains exactly one vertex in R.

This can be expressed in MSOL in the following way:

$$\begin{aligned}&\exists S_1 \subseteq E_1, \exists S_2 \subseteq E_2, \exists R \subseteq V_1 \cap V_2 \text{ s.t}\\ &\qquad \textbf{contraction}_1(S_1, R) \text{ and } \textbf{contraction}_2(S_2, R) \text{ and}\\ &\qquad \forall r, r' \in R\ \textbf{edge}_1(S_1, r, r') \text{ iff } \textbf{edge}_2(S_2, r, r')\end{aligned}$$

Where in this formula, for $i = 1, 2$ $\textbf{contraction}_i(S_i, R)$ checks that that there is exactly one element of R per connected component of V_i induced by S_i, and $\textbf{edge}_i(S_i, r, r')$ is true if and only there is an edge in E_i between the connected components induced by S_i on V_i containing r and r', respectively. We express $\textbf{contraction}_i(S_i, R)$ as follows:

$$\begin{aligned}\textbf{contraction}_i(S_i, R) = \forall r, r' \in R, r \neq r', \neg\textbf{path}(r, r', S_i) \text{ and } \forall u \in V_i \setminus R\ \exists r \in R \text{ s.t.}\\ \textbf{path}(u, r, S_i) \text{ and } \forall r' \neq r\ \neg\textbf{path}(u, r', S_i)\end{aligned}$$

Where in this formula, $\textbf{path}(x, y, U)$ checks whether $x = y$ or there exists a path between x and y using only edges in U (which can be checked in MSOL [14]) if $x \neq y$. Indeed, two vertices are in the same connected component induced by a sub-set of edges if and only there is a path between them using only edges from the sub-set.

As for $\textbf{edge}_i(S_i, r, r')$, it can be expressed as:

$$\textbf{edge}_i(S_i, r, r') = \exists x, y \in V_i \text{ s.t. } \textbf{path}(x, r, S_i) \text{ and } \textbf{path}(y, r, S_i) \text{ and } \{x, y\} \in E_i$$

Overall, this expression only uses quantification over edge subsets (S_1, S_2) and vertex sub-sets (R). By the extended version of Courcelle's theorem allowing to optimize for the size of monadic variables ([4] or [21, Exercise 11.44]), one can find S_1, S_2, R verifying this formula while maximizing $|R|$ (or equivalently minimizing $|S_1| + |S_2|$) in a complexity FPT in the treewidth of $(V_1 \cup V_2, E_1 \cup E_2) = G \cup H$. ◀

We now derive consequences of Theorem 12 by establishing parameterizations that lead $G \cup H$ to have bounded treewidth. In the following, we have graphs G and H and $G \cup H$ is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. For space reasons, the definition of a tree decomposition can be found in the appendix.

▶ **Lemma 13.** *If H is a contraction of G, then $tw(G \cup H) \leq 2 \cdot tw(G)$.*

Proof sketch. Start with a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of G. Then take a witness structure $\mathcal{W}$ of G into T. For each bag X_t of $\mathcal{T}$, and for each $x \in X_t$, add to X_t the representative of $\mathcal{W}(x)$ in H, if not already there. This at most doubles the treewidth.

The bags that contain some $u \in V(G) \cup V(H)$ are connected in the modified decomposition, as otherwise a bag that separates u would also separate $\mathcal{W}(u)$, which should be connected. Also, for each $uv \in E(H)$ some bag has both u, v because $\mathcal{W}(u), \mathcal{W}(v)$ are adjacent. ◀

▶ **Lemma 14.** *If G and H are such that $\exists S_1, S_2$ labeled contraction sequences of size $|S_1| + |S_2| \leq k$ and $G/S_1 = H/S_2$, then $tw(G \cup H) \leq \min(tw(G), tw(H)) + k$*

Proof sketch. Let M be a maximum common contraction of G and H. Since contractions do not increase the treewidth, $tw(M) \leq \min(tw(G), tw(H))$. Then, take a tree decomposition $\mathcal{T}$ of M, then add $(V(G) \setminus V(H)) \cup (V(H) \setminus V(G))$ to every bag. This adds at most k vertices to each bag, and one can argue that this is a valid tree decomposition of $G \cup H$. ◀

▶ **Corollary 15.** *The following results holds:*

- *Labeled Contractibility is FPT in parameter* $\max(tw(G), tw(H))$.
- *Maximum Common Labeled Contraction is FPT in parameter* $k + \min(tw(G), tw(H))$.

Conclusion. We have explored the parameterized complexity of computing a maximum common contraction of two fully-labeld graphs, in which each label is used at most once in each graph. Natural follow-ups could look at the case where labels can be used more than once. Note that it is the hardest possible case, as it contains both unlabelled and uniquely labeled graphs. A first step could be to bound the number of times each label may be present. As for the unlabeled case, it consists in computing edge-contraction distances between graph isomorphism classes, and has not been explored yet. We finish with some open problems: (1) given Theorem 5, is Labeled Contractibility still NP-hard if both input graphs have clique-width three, or even if they are both cographs (clique-width two)? (2) is there a $2^{O(\delta k)} n^c$ time algorithm for Labeled Contractibility, and/or a $2^{O(\Delta k)} n^c$ time algorithm for Maximum Common Labeled Contraction? Or better, could there be subexponential algorithms parameterized by δk or Δk? (3) is Maximum Common Labeled Contraction FPT in $\max(tw(G), tw(H))$?

References

1 Faisal N Abu-Khzam, Édouard Bonnet, and Florian Sikora. On the complexity of various parameterizations of common induced subgraph isomorphism. *Theor. Comp. Sci.*, 2017.

2 Tatsuya Akutsu, Avraham A Melkman, and Takeyuki Tamura. Improved hardness of maximum common subgraph problems on labeled graphs of bounded treewidth and bounded degree. *International Journal of Foundations of Computer Science*, 31(02):253–273, 2020. doi:10.1142/S0129054120500069.

3 Bogdan Alecu, Vadim V Lozin, Daniel A Quiroz, Roman Rabinovich, Igor Razgon, and Viktor Zamaraev. The treewidth and pathwidth of graph unions. *SIAM J on Discrete Mathematics*, 2024.

4 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991. doi:10.1016/0196-6774(91)90006-K.

5 Rémy Belmonte, Petr A Golovach, Pinar Heggernes, Pim van't Hof, Marcin Kamiński, and Daniël Paulusma. Detecting fixed patterns in chordal graphs in polynomial time. *Algorithmica*, 69(3):501–521, 2014. doi:10.1007/S00453-013-9748-5.

6 Rémy Belmonte, Petr A Golovach, Pim van't Hof, and Daniël Paulusma. Parameterized complexity of three edge contraction problems with degree constraints. *Acta Informatica*, 51(7):473–497, 2014. doi:10.1007/S00236-014-0204-Z.

7 Rémy Belmonte, Pinar Heggernes, and Pim van't Hof. Edge contractions in subclasses of chordal graphs. *Discrete Applied Mathematics*, 160(7-8):999–1010, 2012. doi:10.1016/J.DAM.2011.12.012.

8 Théo Boury, Yann Ponty, and Vladimir Reinharz. Automatic exploration of the natural variability of rna non-canonical geometric patterns with a parameterized sampling technique. In *WABI 2023 – 23rd Workshop on Algorithms in Bioinformatics*, 2023. doi:10.4230/LIPIcs.WABI.2023.20.

9 Andries Evert Brouwer and Henk Jan Veldman. Contractibility and np-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987. doi:10.1002/JGT.3190110111.

10 Horst Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern recognition letters*, 18(8):689–694, 1997. doi:10.1016/S0167-8655(97)00060-3.

11 Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. Metrics for phylogenetic networks i: Generalizations of the robinson-foulds metric. *TCBB*, 6(1):46–61, 2008. doi:10.1109/TCBB.2008.70.

12 Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. Metrics for phylogenetic networks ii: Nodal and triplets metrics. *TCBB*, 6(3):454–469, 2008. doi:10.1109/TCBB.2008.127.

13 Gabriel Cardona, Joan Carles Pons, Gerard Ribas, and Tomás Martínez Coronado. Comparison of orchard networks using their extended μ-representation. *TCBB*, 21(3):501–507, 2024. doi:10.1109/TCBB.2024.3361390.

14 Bruno Courcelle. Graph rewriting: An algebraic and logic approach. In *Formal models and semantics*, pages 193–242. Elsevier, 1990. doi:10.1016/B978-0-444-88074-1.50010-X.

15 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

16 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Yves-Pol Deniélou, Frédéric Boyer, Alain Viari, and Marie-France Sagot. Multiple alignment of biological networks: A flexible approach. In *CPM*, pages 263–273. Springer, 2009. doi:10.1007/978-3-642-02441-2_23.

18 Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(1):68–79, 2011.

19 Fazle E Faisal, Lei Meng, Joseph Crawford, and Tijana Milenković. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology*, 2015.

20 Michael R Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comp. Sci.*, 410(1):53–61, 2009. doi:10.1016/J.TCS.2008.09.065.

21 J Flum and M Grohe. Parameterized complexity theory. 2006. texts in theoretical computer science. an eatcs series, 2006.

22 Philippe Gambette, Leo Van Iersel, Mark Jones, Manuel Lafond, Fabio Pardi, and Celine Scornavacca. Rearrangement moves on rooted phylogenetic networks. *PLoS Comp. Bio.*, 2017.

23 Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13:113–129, 2010. doi:10.1007/S10044-008-0141-Y.

24 Pinar Heggernes, Pim Van'T Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM J on Discrete Mathematics*, 2013.

25 Pinar Heggernes, Pim Van't Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul. Contracting graphs to paths and trees. *Algorithmica*, 68:109–132, 2014. doi:10.1007/S00453-012-9670-2.

26 Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications.* Cambridge University Press, 2010.

27 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013. doi:10.1016/J.JCSS.2012.04.004.

28 Maxim Kalaev, Vineet Bafna, and Roded Sharan. Fast and accurate alignment of multiple protein networks. In *RECOMB*, pages 246–256. Springer, 2008. doi:10.1007/978-3-540-78839-3_21.

29 Marcin Kamiński, Daniël Paulusma, and Dimitrios M Thilikos. Contractions of planar graphs in polynomial time. In *ESA*, pages 122–133. Springer, 2010. doi:10.1007/978-3-642-15775-2_11.

30 Ramaswamy Krithika, Pranabendu Misra, Ashutosh Rai, and Prafullkumar Tale. Lossy kernels for graph contraction problems. In *36th IARCS Annual Conference on Foundations of Software Technology and Theor. Comp. Sci.* Schloss Dagstuhl – Leibniz Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.23.

31 Vincent Lacroix, Ludovic Cottret, Patricia Thébault, and Marie-France Sagot. An introduction to metabolic networks and their structural analysis. *TCBB*, 5(4):594–617, 2008. doi:10.1109/TCBB.2008.79.

32 Manuel Lafond, Alitzel López Sánchez, and Weidong Luo. Cluster Editing on Cographs and Related Classes. In Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyen Kim Thang, editors, *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 64:1–64:21, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2025.64.

33 Kaari Landry, Aivee Teodocio, Manuel Lafond, and Olivier Tremblay-Savard. Defining phylogenetic network distances using cherry operations. *TCBB*, 20(3):1654–1666, 2022. doi:10.1109/TCBB.2022.3162991.

34 Asaf Levin, Daniël Paulusma, and Gerhard J Woeginger. The computational complexity of graph contractions i: Polynomially solvable and np-complete cases. *Networks: An International Journal*, 51(3):178–189, 2008. doi:10.1002/NET.20214.

35 Asaf Levin, Daniël Paulusma, and Gerhard J Woeginger. The computational complexity of graph contractions ii: two tough polynomially solvable cases. *Networks: An International Journal*, 52(1):32–56, 2008. doi:10.1002/NET.20249.

36 Bertrand Marchand, Yann Ponty, and Laurent Bulteau. Tree diet: reducing the treewidth to unlock fpt algorithms in rna bioinformatics. *Algorithms for Molecular Biology*, 17(1):8, 2022. doi:10.1186/S13015-022-00213-Z.

37 Bertrand Marchand, Nadia Tahiri, Olivier Tremblay-Savard, and Manuel Lafond. Finding Maximum Common Contractions Between Phylogenetic Networks. In *WABI*, pages 16:1–16:24, 2024. doi:10.4230/LIPIcs.WABI.2024.16.

38 Jiří Matoušek and Robin Thomas. On the complexity of finding iso-and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992. doi:10.1016/0012-365X(92)90687-B.

39 Mohsen Alambardar Meybodi, Fedor V Fomin, Amer E Mouawad, and Fahad Panolan. On the parameterized complexity of [1, j]-domination problems. *Theor. Comp. Sci.*, 804:207–218, 2020. doi:10.1016/J.TCS.2019.11.032.

40 Philippe Rinaudo, Yann Ponty, Dominique Barth, and Alain Denise. Tree decomposition and parameterized algorithms for rna structure-sequence alignment including tertiary interactions and pseudoknots. In *WABI*, pages 149–164. Springer, 2012.

41 David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

42 Saket Saurabh, Uéverton dos Santos Souza, and Prafullkumar Tale. On the parameterized complexity of grid contraction. *Journal of Computer and System Sciences*, 129:22–38, 2022. doi:10.1016/J.JCSS.2022.04.004.

43 Francesc Serratosa. Redefining the graph edit distance. *SN Computer Science*, 2(6):438, 2021. doi:10.1007/S42979-021-00792-5.

44 Zhiping Zeng, Anthony KH Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars: On approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1):25–36, 2009. doi:10.14778/1687627.1687631.

45 Jizhen Zhao, Russell L Malmberg, and Liming Cai. Rapid ab initio prediction of rna pseudoknots via graph tree decomposition. *J of Mathematical Biology*, 56:145–159, 2008.

A Proofs for Section 2 (Preliminary notions)

▶ **Observation 2.** *Let G, H be graphs. Then H is a contraction of G if and only if there exists a witness structure of G into H.*

Proof. The forward direction can be obtained by starting with G and a partition of $V(G)$ in which every vertex is by itself, then after each contraction (u, v), we update the set associated with u by adding all the elements associated with v. When we are done, only $V(G)$ remains and the resulting vertex-set association forms a witness structure. For the reverse direction, we note that the connected subgraphs $G[W_i]$ can all be contracted to a single vertex independently, and under the conditions of witness structures, doing so must result in H. ◀

▶ **Observation 3.** *Let $\mathcal{W}$ be a witness structure of G and H, and let $uv \in E(G)$ such that u, v are in the same witness set of $\mathcal{W}$. If $v \notin V(H)$, then H is a contraction of $G/(u, v)$.*

Proof. Denote $G' = G/(u, v)$. It suffices to consider the partition $\mathcal{W}'$ of $V(G')$ obtained from $\mathcal{W}$ by just removing v from $\mathcal{W}(u)$. It is easy to see that $|\mathcal{W}'| = |V(H)|$, that $\mathcal{W}(u) \setminus \{v\}$ still contains the same representative of $V(H)$ since $v \notin V(H)$, and that $G'[\mathcal{W}(u) \setminus \{v\}]$ is still connected. Moreover, the edges with one endpoint in $\mathcal{W}(u) \setminus \{v\}$ and the other outside are the same as $\mathcal{W}(u)$, and so $\mathcal{W}'$ is a witness structure of G' into H. By Observation 2, H is a contraction of G. ◀

B Proofs for Section 3 (Labeled Contractibility)

▶ **Theorem 4.** *The LABELED CONTRACTIBILITY problem is W[1]-hard in parameter k, which is the number of contractions needed to transform one input graph into the other.*

Proof. Let G_C be an instance of MULTICOLORED CLIQUE and let G, H be constructed as above. We show that G_C has a multicolored clique if and only if H is a contraction of G.

Suppose that G_C has a multicolored clique $u_1, \ldots, u_k$, where $u_i \in V_i$ for each $i \in [k]$. We present a witness structure $\mathcal{W}$ of G into H. In $\mathcal{W}$, add the sets $\{u_i, t_i\}$ for each $i \in [k]$, and the singleton sets $\{\{v\} : v \in V(G_C) \setminus \{u_1, \ldots, u_k\}\}$. This corresponds to the sequence of contractions $(u_1, t_1), \ldots, (u_k, t_k)$. It is clear that $|\mathcal{W}| = |V(H)|$, that each element of $\mathcal{W}$ has a unique element of $V(H)$, and that each $G[W_i]$ is connected. We must argue that $vw \in E(H)$ if and only if there is some edge between $\mathcal{W}(v)$ and $\mathcal{W}(w)$ in G.

Let $v, w \in V(H)$ be two distinct vertices of H, which are also vertices of G and G_C. If $vw \in E(H)$, then because G started as a copy of H, $vw \in E(G)$. This edge still exists between $\mathcal{W}(v)$ and $\mathcal{W}(w)$ in G, because $\mathcal{W}(v) \neq \mathcal{W}(w)$. So instead assume that $vw \notin E(H)$. Let $v \in V_i, w \in V_j$, and notice that $i \neq j$ since we made V_i and V_j cliques in H. Suppose for contradiction that there is some edge between $\mathcal{W}(u)$ and $\mathcal{W}(v)$ in G. By construction, this edge cannot be vw, and it cannot be $t_i w$ nor $t_j v$, which do not exist. So $\mathcal{W}(v)$ must contain

t_i and $\mathcal{W}(w)$ must contain t_j. By the construction of $\mathcal{W}$, this means that $v = u_i$ and $w = u_j$. Since u_i, u_j are in a clique, $vw \in E(G)$ and $vw \in E(H)$, a contradiction. Thus $\mathcal{W}(u)$ and $\mathcal{W}(v)$ are not adjacent in G, and it follows that $\mathcal{W}$ is a witness structure of G into H. By Observation 2, H is a contraction of G.

Conversely, suppose that one may apply at most k contractions to G to obtain H. Let $\mathcal{W}$ be a witness structure of G into H. For $i \in [k]$, consider the vertex t_i of G, and denote by u the representative of $\mathcal{W}(t_i)$, where $u \in V(H)$. We claim that $u \in V_i$, so assume instead that $u \in V_j$, with $i \neq j$. Recall that t_i is adjacent to every vertex in V_i in G, so if $u \in V_j$, then $\mathcal{W}(u) = \mathcal{W}(t_i)$ is adjacent to every witness set that contains a vertex of V_i. But $u \in V_j$, a contradiction since we assume that u has at least one non-neighbor in V_i in G_C, and thus in H as well. It follows that the unique vertex u of G in the same witness set as t_i belongs to V_i.

Since this holds for every $i \in [k]$, this also implies that no two t_i vertices belong to the same witness set. We may then define the set of k distinct vertices $u_1, \ldots, u_k$ where, for $i \in [k]$, u_i is the vertex of V_i in the same witness set as t_i. We argue that this set forms a clique of G_C. Let $i, j \in [k]$ with $i \neq j$. Because of the edge t_it_j, $\mathcal{W}(t_i)$ and $\mathcal{W}(t_j)$ are adjacent in G. Thus there is an edge between $\mathcal{W}(u_i)$ and $\mathcal{W}(u_j)$, and by the conditions of witness sets the edge u_iu_j must exist in H. Thus $u_1, \ldots, u_k$ forms a multicolored clique of G_C. ◀

▶ **Theorem 5.** *The* Labeled Contractibility *problem is NP-hard, even if the input graphs have clique-width at most* 4.

Proof. We first prove the correctness of the reduction, and then argue about the clique-width of the constructed graphs.

Let $a_0, \ldots, a_{n-1}, C, k$ be an instance of Unary Perfect Bin Packing, and G, H constructed as described above. Suppose first that Unary Perfect Bin Packing is a yes-instance, i.e., that there exists an assignmnent ϕ of the items to the bins such that the sum of the sizes of the items in a bin is exactly C for each of the k bins. In G, we first contract each edge (b_i, t_x) such that $\phi(x) = i$ (green edges on the example in Figure 3), keeping the vertex b_i. Since each item is assigned to a bin, each t_x is contracted into some vertex in B, therefore the set of vertices T disappears. Since for each bin i, $\sum_{x\,:\,\phi(x)=i} a_x = C$, and each t_x has exactly $|A_x| = a_x$ neighbors in A, each b_i now has exactly C neighbors in A. Let i be an integer between 0 and $k-1$, and let us denote $v_i^0, \ldots, v_i^{C-1}$ the C neighbors of b_i in A in this partially contracted graph. We next contract each v_i^y (for $0 \leq y \leq C-1$) with the vertex d_{Ci+y} of D (we can, as A and D form a bi-clique in G), keeping the d_{Ci+y} vertices. Doing this for all i, we delete the set A, which leaves only B and D. As required, B D each form a clique, and there is an edge b_id_j if and only if $\lfloor j/C \rfloor = i$. We have therefore described a contraction sequence from G to H.

In the other direction, suppose that H is a contraction of G. Let $\mathcal{W}$ be a witness structure of G into H. To ease notation a bit, we will denote by $W_i^b = \mathcal{W}(b_i)$ the witness set of each b_i and by $W_j^d = \mathcal{W}(d_j)$ the witness set of each d_j. We argue that each element of T must be part of some W_i^b. Indeed, as B and D are the only vertex sets left in H, the sets W_i^b and W_j^d must form a partition of $V(G)$, and in particular of the elements of A and T are in one of those witness sets. However, if a vertex t_x from T is in some W_j^d, then the after the contractions the vertex d_j becomes connected to all of B, which is not the case in H. Each vertex t_x is therefore in one of the sets W_i^b.

We can then define ϕ as $\phi(x) = i$ if and only if $t_x \in W_i^b$ (such an i is unique as the W_i^b do not intersect).

Let us now argue that ϕ assigns items to bins perfectly. Let u be some vertex of A. If u is in a witness set W_i^b, then after contracting the vertex b_i becomes connected to all of D, which is not the case in H. Therefore, u is in some witness set W_j^d. We also argue the opposite, that for any $d_j \in D$, there is some $u \in A$ such that $u \in W_j^d$. Indeed, if some W_j^d contains no element of A, then note that in G the neighborhood of d_j is composed of $D \setminus \{d_j\} \cup A$, and other elements of D cannot be in W_j^d. Therefore, we would have $W_j^d = \{d_j\}$. In G, $\{d_j\}$ is only adjacent to witness sets that contain some vertex of A or D. As we just argued, each vertex of A is in the witness set of a D vertex. Thus the witness set W_j^d has no neighbor belonging to a W_i^b set, which is a contradiction as d_j shares an edge with some vertex of B in H.

Therefore, we have both that all vertices in A are in some W_j^d, and that each W_j^d contains some vertex in A. Since $|A| = \sum_{x=0}^{n-1} a_x = Ck = |D|$, this implies that each W_j^d is of the form $\{d_j, u\}$, for some $u \in A$ uniquely associated to j. In particular, this implies that each vertex t_x, which has a_x neighbors in A, is therefore adjacent to exactly a_x distinct witness sets W_j^d. We write $S_x = \{d_j \in D : N_G(t_x) \cap W_j^d \neq \emptyset\}$, i.e., the set of vertices of D whose witness set is adjacent to that of t_x. We have $|S_x| = a_x$, and for $x \neq y$, we also have $S_x \cap S_y = \emptyset$. Now, $\forall i \in [0, k-1]$, $\cup_{x:\phi(x)=i} S_x$ denotes the representatives of the witness sets of D neighbor to W_i^b, and therefore the neighborhood of b_i outside of B in H. As each b_i has exactly C neighbors outside of B in H, we must have $\sum_{x\,:\,\phi(x)=i} |S_x| = \sum_{x\,:\,\phi(x)=i} a_x = C$. Therefore, $a_0, \ldots, a_{n-1}, C, k$ is a yes-instance.

Clique-width analysis. We finish by arguing that both G and H have clique-width at most 4. Let us start with H, which we construct inductively on $k = 0, 1, \ldots, k-1$. For $k = 0$, we add vertices $d_0, \ldots, d_{C-1}$ with color 1, b_0 with color 2, and add all edges between colors 1 and 2. This finishes the construction for $k = 0$, with vertices of D added so far having color 1 and those of B having color 2. Then, given the graph H constructed with $k-1$ vertices in B, with all vertices of D having color 1 and those of B having color 2, we introduce the vertices $d_{(k-1)C}, \ldots, d_{kC-1}$ under color 3, vertex b_{k-1} under color 4. We add all edges between color 3 and 4 (which creates exactly the required edges between b_{k-1} and $d_{(k-1)C}, \ldots, d_{kC-1}$), then connect color 1 and 3 as a bi-clique (making D a clique) and likewise for colors 2 and 4 (making B a clique). We finish this step by recoloring 3 with 1 and 4 with 2, in case we may pursue the construction.

As for G, it actually has clique-width 3. We can first construct the subgraph induced by A and T using two colors. Indeed, $G[A \cup T]$ is a forest of star trees, so we can just create each star tree independently with two colors, and then take the disjoint union of all those star trees. This can be done in a way that $G[A \cup T]$ is built with vertices of A having color 1 and those of T having color 2. We then construct the B clique independently using two colors and recolor every vertex of B to 3. We then add B to the construction and add all edges between B and T (colors 1 and 3). Then recolor 3 to 1, so that B and T have the same colors. In the same manner, we construct the D clique and give it color 3, and add edges between colors 2 and 3 (A and D). This results in G. ◀

▶ **Lemma 6.** *If H is obtained from G by $\leq k$ contractions, then $\delta(H) \leq \delta(G) + k$*

Proof. We prove the result by induction. The base case $k = 0$ is trivial. Let us therefore consider now G' obtained from G by a single contraction, of u into v. Let also σ be an elimination order such that $\max_{x \in V(G)} |N^G_{\geq,\sigma}(x)| = \delta(G)$. We distinguish below two cases. In both, we find an order σ' of the vertices of G' such that $\max_{x \in V(G')} |N^{G'}_{\geq,\sigma'}(x)| \leq \delta(G) + 1$, which allows to conclude.

- If $N(u) \cup N(v)$ (neighborhood of v in G') contains no element ranked higher than v in σ, we simply get σ' by removing u from σ. We have $N^G_{\sigma,\geq}(v) = N^{G'}_{\geq,\sigma'}(v) = \emptyset$. As for $x \in V(G')$ and $x \neq v$, $|N^{G'}_{\geq,\sigma'}(x)| \leq |N^G_{\geq,\sigma}(x)|$ if x is before u in σ (the size decreases by 1 if x is a neighbor of both u and v and stays the same otherwise), $|N^{G'}_{\geq,\sigma'}(x)| = |N^G_{\geq,\sigma}(x)|$ if x is after v in σ and $|N^{G'}_{\geq,\sigma'}(x)| \leq |N^G_{\geq,\sigma}(x)| + 1$ if x is between u and v in σ (with an increase in the case where x is a neighbor of u but not v). Overall, we do have $\max_{x \in V(G')} |N^{G'}_{\geq,\sigma'}(x)| \leq \delta(G) + 1$.
- Otherwise, let t be the highest-ranked element of $N(u) \cup N(v)$ by σ (it is after v, as otherwise we would be in the previous case). In this case, we get σ' by removing u an moving v right after t in σ. By doing so, we have $|N_{\geq,\sigma'}(v)| = 0$, while for $x \in V(G')$ such that $x \neq v$, $|N^{G'}_{\geq,\sigma'}(x)| \leq |N^G_{\geq,\sigma}(x)| + 1$ if x is a neighbor of u after u in σ, or if x is a neighbor of v after v in σ. If x is not a neighbor of u and v, or if x is before u in σ, $|N^{G'}_{\geq,\sigma'}(x)| \leq |N^G_{\geq,\sigma}(x)|$. Again, we obtain $\max_{x \in V(G')} |N^{G'}_{\geq,\sigma'}(x)| \leq \delta(G) + 1$.

We have therefore found an ordering σ' of G' such that $\max_{x \in V(G')} |N^{G'}_{\geq,\sigma'}(x)| \leq \delta(G) + 1$. By definition of the degeneracy, $\delta(G') \leq \delta(G) + 1$. Therefore, by induction over the number k of contractions, $\delta(H) \leq \delta(G) + k$ ◀

C Proofs for Section 4 (Maximum Common Labeled Contractions)

▶ **Lemma 8.** *Let H be a contraction of a graph G, and let $\mathcal{W}$ be the witness structure of G into H. Let $R \subseteq E(G)$ be a matching of G such that, for every edge $xy \in R$, the vertices x and y are in the same witness set of $\mathcal{W}$.*

Let G' be the graph obtained after contracting every edge of R, making sure not to remove a representative vertex of $\mathcal{W}$. Then H is a contraction of G'.

Proof. As in the proof of Observation 3, it suffices to observe that if we start from $\mathcal{W}$ and contract (x, y), where $xy \in R$, then after removing y from $\mathcal{W}(y)$, the resulting collection $\mathcal{W}'$ is a witness structure of $G/(x, y)$ into H (this of course requires y to not be a representative). Moreover, if $R \setminus \{xy\}$ is non-empty, then any edge that remains still has its ends in one witness set, because R is a matching. Thus we may continue applying such contractions inductively. ◀

▶ **Theorem 9.** *The* Maximum Common Labeled Contraction *is W[1]-hard in parameter k, the total number of contractions needed to achieve a common contraction, even if both input graphs have degeneracy 4.*

Proof of Theorem 9. Consider an instance $G_C = (V, E)$ of Multicolored Clique and graphs G, H constructed as above, along with parameter $K = 2(k + \binom{k}{2})$, the total number of contractions.

We first argue that the degeneracy of G and H is at most 4. Since G is a subgraph of H, it suffices to consider H only. We provide an elimination sequence in which every vertex has at most four neighbors before being deleted. First consider any extra vertex in a V_i' or Z_{ij}' set. These have degree two, so we may delete them first. Next consider any z_e vertex of H from some Z_{ij} set, where $e = uv$. It has four neighbors $\{u, v, c_{ij}, d_{ij}\}$. Delete all the Z_{ij} vertices, for every i, j. After that, consider any u vertex in some V_i set. It has two remaining neighbors a_i and b_i. Delete all of those next. Then delete the d_{ij} vertices (one remaining neighbor c_{ij}) and the c_{ij} vertices (two remaining neighbors a_i, a_j). The resulting graph has maximum degree 1 as the $a_i b_i$'s form a matching, and it follows that H and G have degeneracy 4.

We next show that G_C admits a multicolored clique if and only if G, H have a common contraction that can be achieved with a total of at most K contractions.

Suppose that G_C admits a multicolored clique $u_1, u_2, \ldots, u_k$, where each u_i is in V_i. We perform the same contractions in G and H. In either graph:

- for each $i \in [k]$, contract (a_i, u_i);
- for each distinct $i, j \in [k]$, contract $(c_{ij}, z_{u_i u_j})$.

Note that the kept vertex is a_i and c_{ij}, respectively. This clearly requires $K = 2(k + \binom{k}{2})$ contractions in total. We must argue that applying them in G or H results in the same graph. Let M_G and M_H be the graphs obtained by applying the above contractions on G and H, respectively. We use the witnesses for our arguments.

Let $\mathcal{W}$ be the witness structure of G into M_G. Note that $\mathcal{W}$ contains $\{a_i, u_i\}$ for $i \in [k]$, $\{c_{ij}, z_{u_i u_j}\}$ for distinct $i, j \in [k]$, and every other vertex is in a witness set of size 1. Also note that $\mathcal{W}$ is also a witness structure of H into M_H, since we apply the same contractions on both graphs. We argue that two witness sets of $\mathcal{W}$ are adjacent in G if and only if the same two witness sets are adjacent in H, from which we can deduce that $M_G = M_H$.

- First consider $i \in [k]$ and some vertex $v \in V_i' \setminus \{u_i\}$. Then $\{v\}$ is an element of $\mathcal{W}$. From the construction, we see that H adds no neighbor to v. Thus $N_G(v) = N_H(v)$, and it follows that $\{v\}$ is adjacent to some $W \in \mathcal{W}$ in G if and only if $\{v\}$ is adjacent to W in H.
- Consider distinct $i, j \in [k]$ and a $z \in Z_{ij}' \setminus \{z_{u_i u_j}\}$ vertex. Then $\{z\} \in \mathcal{W}$, and again from $N_G(z) = N_H(z)$ we make the same conclusion as in the previous case.
- From now on we do not need to consider the adjacencies of $\{v\}$ and $\{z\}$ witness sets. Only witness sets of the form $\{b_i\}, \{d_{ij}\}, \{a_i, u_i\}, \{c_{ij}, z_{u_i u_j}\}$ remain. Consider vertex b_i, where $i \in [k]$. Then $\{b_i\} \in \mathcal{W}$. In either graph, $\{b_i\}$ is adjacent to $\{a_i, u_i\}$ and no other remaining witness set.
- Consider vertex d_{ij}, with distinct $i, j \in [k]$. In either graph, $\{d_{ij}\}$ is only adjacent to $\{c_{ij}, z_{u_i u_j}\}$ among the witness sets that remain to consider.
- The remaining witness sets have the form $\{a_i, u_i\}$ or $\{c_{ij}, z_{u_i u_j}\}$. Consider $i \in [k]$ and witness set $\{a_i, u_i\}$. Among the witness sets that remain to consider, in H, it is adjacent to all witness sets in $\{\{c_{ij} z_{u_i u_j}\} : j \in [k] \setminus \{i\}\}$, but not to any other $\{a_j u_j\}$ set nor to any other $\{c_{jh} z_e\}$ witness set with $j, h \neq i$. The same is true in G, because u_i has every $z_{u_i u_j}, j \in [k] \setminus \{i\}$ in its neighborhood.
- Finally, for distinct $i, j \in [k]$, we must consider $\{c_{ij}, z_{u_i u_j}\}$. In either graph, all their adjacencies were handled previously, as these are not adjacent to any other $\{c_{xy}, z_{u_x u_y}\}$ set.

We have thus shown that if we partition $V(G)$ into $\mathcal{W}$ and apply the corresponding contractions, and do the same on H, we obtain the same graph.

In the converse direction, let M be a common contraction of G and H achievable in a total of K contractions. We need two witness structures, one for G into M and one for H into M. So let $\mathcal{X}$ be a witness structure of G into M, and $\mathcal{Y}$ a witness structure of H into M. Note that for a vertex w, we write $\mathcal{X}(w)$ for the witness set of $\mathcal{X}$ that contains w, and $\mathcal{Y}(w)$ for the witness set of $\mathcal{Y}$ that contains w.

We must consider multiple ways that G and H could be turned into M, so we split the proof into claims.

▷ Claim 16. For each $i \in [k]$, at least one of $\mathcal{X}(a_i)$ or $\mathcal{X}(b_i)$ has two elements or more. Likewise, for each distinct $i, j \in [k]$, at least one of $\mathcal{X}(c_{ij})$ or $\mathcal{X}(d_{ij})$ has two elements or more.

Proof. Suppose for contradiction that $\mathcal{X}(a_i) = \{a_i\}$ and $\mathcal{X}(b_i) = \{b_i\}$. Then a_i, b_i are representatives in $\mathcal{X}$ and thus vertices of M. Since $a_ib_i \notin E(G)$, they do not share an edge in M. Now considering H, we have $\mathcal{Y}(a_i) \neq \mathcal{Y}(b_i)$ since a_i, b_i are both in M. However, $\mathcal{Y}(a_i), \mathcal{Y}(b_i)$ are adjacent in H, and thus according to $\mathcal{Y}$ we should have the edge a_ib_i in M, a contradiction.

The argument for $\mathcal{X}(c_{ij}), \mathcal{X}(d_{ij})$ is identical. Assume that $\mathcal{X}(c_{ij}) = \{c_{ij}\}$ and $\mathcal{X}(d_{ij}) = \{d_{ij}\}$. Then c_{ij}, d_{ij} are in M and do not share an edge since $c_{ij}d_{ij} \notin E(G)$. However, $c_{ij}d_{ij} \in E(H)$, leading to the same contradiction as above. ◁

Claim 16 lets us construct a matching $R \subseteq E(G)$ as follows. For each $i \in [k]$, by Claim 16 there is either an edge a_iu or b_iu whose ends are in the same set of $\mathcal{X}$, where u is a neighbor of a_i or b_i in G and thus $u \in V_i'$. Choose any such edge and add it to R (make one choice per $i \in [k]$, which clearly gives a matching so far). Likewise for distinct $i, j \in [k]$, there is either an edge $c_{ij}z$ or $d_{ij}z$ whose ends are in the same set of $\mathcal{X}$, where $z \in Z_{ij}'$. Add any such edge to R, for each i, j, and notice that this does produce a matching R such that the ends of each edge are in the same witness set.

By Lemma 8, R corresponds to a set of contractions we can perform first in a sequence from G to M. We have $|R| = k + \binom{k}{2} = K/2$, and applying the contractions removes $K/2$ vertices from G. Since $V(G) = V(H)$, at least $K/2$ vertices must also disappear from H to M, and it follows that H also requires at least $K/2$ contractions. Because M can be reached with a total of K contractions, we deduce from Claim 16 that after applying the contractions corresponding to R, there is no room for further contractions. It follows that *exactly* one of a_i or b_i has a partner in its witness set of $\mathcal{X}$, and the same holds for c_{ij} versus d_{ij} (we note that a_i, b_i cannot be in the same witness set of $\mathcal{X}$ since they are not neighbors, same for c_{ij}, d_{ij}). Furthermore, the edges contracted from G to M must consist of precisely R, although we do not know which end of each edge is kept or removed. However we can argue that it is the a_i's and the c_{ij}'s that are part of the matching, not the b_i's or d_{ij}'s.

▷ Claim 17. For each $i \in [k]$, $\mathcal{X}(a_i) = \{a_i, u\}$ for some $u \in V_i'$. Likewise for each distinct $i, j \in [k]$, $\mathcal{X}(c_{ij}) = \{c_{ij}, z\}$ for some $z \in Z_{ij}'$.

Proof. Let $i \in [k]$ and note that by Claim 16 and the above discussion that either $|\mathcal{X}(a_i)| = 2$ or $|\mathcal{X}(b_i)| = 2$, but not both. Suppose for contradiction that $\mathcal{X}(b_i) = \{b_i, u\}$ for some $u \in V_i'$. Then $\mathcal{X}(a_i) = \{a_i\}$. Since at most one vertex of V_i' can be part of a contraction, in M every vertex of $V_i' \setminus \{u\}$ is present. So by inspecting $N_G(a_i)$, we infer that in M, the vertex a_i is adjacent to $V_i' \setminus \{u\}$, and to u or b_i, whichever is the representative of $\mathcal{X}(b_i)$, and a_i has no other neighbor in M.

Consider vertex c_{ij} of H and $\mathcal{Y}(c_{ij})$. We cannot have $\mathcal{Y}(a_i) = \mathcal{Y}(c_{ij})$, as this would make $\mathcal{Y}(a_i)$ adjacent to all extra vertices of Z_{ij}' that are in H (and such vertices exist in M since there are $4k^3 > K$ of them). The established adjacencies of a_i in M do not allow this. Thus $\mathcal{Y}(c_{ij}) \neq \mathcal{Y}(a_i)$ and the two witness sets are adjacent in H because $a_ic_{ij} \in E(H)$. The representative x of $\mathcal{Y}(c_{ij})$ must be a neighbor of a_i in M, and so x is in $V_i' \cup \{b_i\}$. However, this makes x adjacent to all extra vertices of Z_{ij}' that remain in M, which is clearly not possible by applying the contractions on G of the matching R discussed above. Therefore, $|\mathcal{X}(a_i)| = 2$.

Let us move on to the second part, that each $\mathcal{X}(c_{ij})$ has two elements. The proof goes along the same lines. Suppose instead that $\mathcal{X}(d_{ij}) = \{d_{ij}, z\}$ for some $z \in Z_{ij}'$ and $\mathcal{X}(c_{ij}) = \{c_{ij}\}$. From $N_G(c_{ij})$ we infer that in M, c_{ij} has only elements of $Z_{ij}' \cup \{d_{ij}\}$ in its neighborhood. Now consider $\mathcal{Y}(a_i)$ and note that $\mathcal{Y}(a_i) \neq \mathcal{Y}(c_{ij})$ as this would make c_{ij} adjacent to extra vertices of V_i'. The two witness sets are adjacent, and so the representative x of $\mathcal{Y}(a_i)$ is in $Z_{ij}' \cup \{d_{ij}\}$. This makes x adjacent to extra vertices of V_i', which is not possible from the contractions of R on G. Thus $|\mathcal{X}(c_{ij})| = 2$. ◁

By Claim 17, each a_i is contracted with exactly one $u_i \in V_i'$. We claim that if we take all u_i's in the same witness set as an a_i, we get a multicolored clique of G_C.

▷ Claim 18. Consider distinct $i, j \in [k]$ and u_i, u_j be such that $\mathcal{X}(a_i) = \{a_i, u_i\}$ and $\mathcal{X}(a_j) = \{a_j, u_j\}$. Then $u_i \in V_i$, $u_j \in V_j$, and $u_i u_j \in E(G_C)$.

Proof. We know that $u_i \in V_i'$ and $u_j \in V_j'$. Let us assume that either u_i or u_j is an extra vertex, or that $u_i u_j \notin E(G_C)$. Either way, in G and H the vertex $z_{u_i u_j}$ does not exist. Now let $z \in Z_{ij}'$ such that $\mathcal{X}(c_{ij}) = \{c_{ij}, z\}$, which exists by Claim 17. In G and H, at least one of u_i or u_j is not adjacent to z, as otherwise z would be $z_{u_i u_j}$. Suppose without loss of generality that u_i is not adjacent to z, and let $x \in \{a_i, u_i\}$ be the representative of $\mathcal{X}(a_i)$, and $y \in \{c_{ij}, z\}$ the representative of $\mathcal{X}(c_{ij})$. Then in G, the witness sets $\mathcal{X}(x) = \{a_i, u_i\}$ and $\mathcal{X}(y) = \{c_{ij}, z\}$ are not adjacent.

Observe that in M, the neighborhood of x includes all extra vertices $V_i' \setminus \{x\}$. Likewise, y has among its neighbors in M all extra vertices of $Z_{ij}' \setminus \{y\}$. One implication is that in H, $\mathcal{Y}(x)$ must be adjacent to all witness sets of $\mathcal{Y}$ that contain an extra vertex of V_i' (except possibly $\mathcal{Y}(x)$ itself). Since there are $4k^3 > 2K$ extra vertices, this is only possible if $\mathcal{Y}(x)$ has a vertex that is a neighbor of all extra vertices, which is either b_i or a_i. Since b_i is in M, $b_i \notin \mathcal{Y}(x)$, and thus a_i is in $\mathcal{Y}(x)$. Using the same reasoning on y and c_{ij}, we get that c_{ij} is in $\mathcal{Y}(y)$. Hence $\mathcal{Y}(x)$ and $\mathcal{Y}(y)$ are adjacent in H, a contradiction since $\mathcal{X}(x)$ and $\mathcal{X}(y)$ are not adjacent in G. ◁

By this last claim, we can take the set $\{u_i : u_i \in \mathcal{X}(a_i), i \in [k]\}$, and all these vertices are in G_C and every pair shares an edge. Thus G_C has a multicolored clique. ◀

▶ **Proposition 11.** *The* Maximum Common Labeled Contraction *problem can be solved in time $O((8(k+1)\Delta)^{k+1} \cdot n)$, where k is the number of contractions, $\Delta = \max(\Delta(G), \Delta(H))$, $n = |V(G)| + |V(H)|$ and $m = |E(G)| + |E(H)|$.*

Proof. Consider an instance of Maximum Common Labeled Contraction consisting of labeled graphs G, H of maximum degree Δ or less, and parameter k for the allowed number of contractions. We describe a recursive branching algorithm in which each branching path makes at most k contractions (or otherwise returns that no solution is possible in this path). By Lemma 10, any instance encountered in a recursion has maximum degree at most $\Delta + k(\Delta - 2)$, which we upper bound by $(k+1)\Delta$ for simplicity.

So we assume that we are in a recursive call which receives G and H and integer k. Suppose that there is some $v \in V(G)$ such that $v \notin V(H)$. Then to obtain any common contraction, v must be in the same witness set as one of its neighbors. We branch on every possible contraction (w, v), for every neighbor w of v. We decrease the parameter k by 1 in each branch. This branches into at most $(k+1)\Delta$ cases, and by Observation 3 one of these branches will lead to a maximum common contraction with k operations if one exists. The same branching can be applied if H has some v not in G.

So assume that $V(G) = V(H)$. If $E(G) = E(H)$ then we are done, so assume without loss of generality that G has an edge uv that is not in H. Observe that to obtain a common contraction, u or v must be incident to at least one contraction, in at least one of G or H, as otherwise the edge uv stays present in G and it remains a non-edge in H. That is, one of u or v has a neighbor in the same witness set, in one of the graphs. By Observation 3, we can branch into all the contractions that u or v could be part of, in either graph, i.e., we branch into every contraction (x, y) where $\{u, v\} \cap \{x, y\} \neq \emptyset$, and where $xy \in E(G) \cup E(H)$. The number of such possible contractions incident to u is at most $2(|N_G(u)| + |N_H(u)|) \leq 2((k+1)\Delta + (k+1)\Delta) = 4(k+1)\Delta$ (we multiply by 2 because contractions are ordered pairs). The same bound holds for v, so in total we branch into at most $8(k+1)\Delta$ possibilities, still decreasing k by 1 in each case.

It is easy to see that the algorithm finds a common contraction achievable with at most k contractions if and only if one exists. For the complexity, the algorithm creates a recursion tree in which each vertex has at most $8(k+1)\Delta$ children, and whose depth is at most k. In each recursive call, there are at most $2k\Delta \cdot n$ edges, so we can check in time $O(k\Delta n)$ whether G and H are the same graph, and if not find a vertex or a vertex pair to branch on. This results in a running time of $O((8(k+1)\Delta)^{k+1} \cdot n)$. ◀

Let G be a graph. A *tree decomposition* of G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ where T is a tree and, for each $t \in V(T)$, $X_t \subseteq V(G)$. The sets X_t are called *bags*. Moreover, $\mathcal{T}$ satisfies the following:

1. for each $x \in V(G)$, some bag X_t of $\mathcal{T}$ contains x.
2. for each $x \in V(G)$, the vertices of T corresponding to bags that contain x form a connected subtree of T.
3. for each edge $uv \in E(G)$, some bag contains both u and v.

The *width* of $\mathcal{T}$ is $\max_{t \in V(T)}(|X_t| - 1)$, and the *treewidth* of G is the minimum width of a tree decomposition of G.

▶ **Lemma 13.** *If H is a contraction of G, then $tw(G \cup H) \leq 2 \cdot tw(G)$.*

Proof. Let $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ be a tree decomposition of G, and $\mathcal{W}$ a witness structure of G into H. We build a new decomposition $\mathcal{T}' = (T, \{Z_t\}_{t \in T})$ and argue it is a tree decomposition of $G \cup H$. The underlying tree T is the same for $\mathcal{T}$ and $\mathcal{T}'$. As for the bags, we set:

$$Z_t = X_t \cup \{u \in V(H) : \exists x \in X_t \text{ s.t } x \in \mathcal{W}(u)\}$$

In other words, we take each bag X_t, and for each $x \in X_t$ we add the representative in H of the witness set of x, which results in Z_t. Let us check that $\mathcal{T}'$ is a valid tree decomposition of $G \cup H$.

- (connectivity of vertex representation) First note that a vertex $x \in V(G) \setminus V(H)$ belongs to Z_t if and only if x belongs to X_t, since only vertices of H are added to bags. Since the bags containing x are connected in $\mathcal{T}$, they are still connected in $\mathcal{T}'$. Now Let $u \in V(H)$ and $t \in V(T)$. First, note that since $u \in V(G)$, there is at least one bag of $\mathcal{T}$ and thus $\mathcal{T}'$ containing it. Then, let t_1, t_2, t_3 be three vertices of T such that t_3 is on the path from t_1 to t_2. Suppose that $u \in Z_{t_1}$, $u \in Z_{t_2}$ but $u \notin Z_{t_3}$. By definition of $\mathcal{T}'$, $\exists a \in X_{t_1}$ such that $a \in \mathcal{W}(u)$ and $b \in X_{t_2}$ such that $b \in \mathcal{W}(u)$. Note that $a = u$ is possible, in which case $u \in X_{t_1}$ (likewise, $b = u$ is possible). We also have $\mathcal{W}(u) \cap X_{t_3} = \emptyset$, as otherwise u would be added to Z_{t_3}, and in particular, $a, b \notin X_{t_3}$.
  If $a = b$, we have a contradiction with $\mathcal{T}$ being a valid tree decomposition of G, since $a \in V(G)$ and the bags of $\mathcal{T}$ containing it would be disconnected by t_3. If $a \neq b$, since $\mathcal{T}$ is a tree decomposition of G, X_{t_3} separates a from b, i.e. intersects all paths from a to b in G. However, by the connectivity of $\mathcal{W}(u)$, there is a path from a to b lying entirely in $\mathcal{W}(u)$, which yields a contradiction.
- (edge representation) For $uv \in E(G)$, some bag of $\mathcal{T}$ contains u and v, and the same holds in $\mathcal{T}'$. Then, for each edge uv of H, there must be at least one edge xy of G such that $x \in \mathcal{W}(u)$, $y \in \mathcal{W}(v)$. There must be a bag in $\mathcal{T}$ containing both x and y. By definition, this bag contains u and v in $\mathcal{T}'$, therefore representing the edge.

By definition of $\mathcal{T}'$, each of its bag Z_t is at most twice as big as X_t, because each $x \in X_t$ enforces the addition of at most one new vertex in Z_t, namely the $u \in V(H)$ that is the representative of its witness set. Therefore the width of $\mathcal{T}'$ is at most $2tw(G)$. ◀