

Distance-preserving graph contractions

Aaron Bernstein^{1*}, Karl Däubel^{1*}, Yann Disser^{2†},
Max Klimm^{3‡}, Torsten Mütze^{1*} and Frieder Smolny^{1*}

¹Institut für Mathematik, TU Berlin

²Department of Mathematics, Graduate School CE, TU Darmstadt

³Wirtschaftswissenschaftliche Fakultät, HU Berlin

ABSTRACT. Compression and sparsification algorithms are frequently applied in a pre-processing step before analyzing or optimizing large networks/graphs. In this paper we propose and study a new framework contracting edges of a graph (merging vertices into super-vertices), with the goal of preserving pairwise distances as accurately as possible. Formally, given an edge-weighted graph, the contraction should guarantee that for any two vertices at distance d , the corresponding super-vertices remain at distance at least $\varphi(d)$ in the contracted graph, where φ is a tolerance function bounding the permitted distance distortion.

We present a comprehensive picture of the algorithmic complexity of the contraction problem for affine tolerance functions $\varphi(x) = x/\alpha - \beta$, where $\alpha \geq 1$ and $\beta \geq 0$ are arbitrary real-valued parameters. Specifically, we present polynomial-time algorithms for trees as well as hardness and inapproximability results for different graph classes, precisely separating easy and hard cases. Further we prove upper and lower bounds, yielding efficient algorithms to compute (non-optimal) contractions despite our hardness results.

1. INTRODUCTION

When dealing with large networks, it is often beneficial to compress or sparsify the data to manageable size before analyzing or optimizing the network directly. To be useful, a meaningful compression should represent salient features of the original network with good approximation, while being much smaller in size. In this paper, we focus on a compression of undirected edge-weighted graphs that approximately maintains all distances between vertices in the graph.

In this context, an extensively studied concept are *spanners* (e.g. [PS89, ADD⁺93, BKMP05, AB16]). Given an undirected graph $G = (V, E)$ and numbers $\alpha \geq 1$ and $\beta \geq 0$, a subgraph $H = (V, E')$, $E' \subseteq E$, is an (α, β) -*spanner of G* if $\text{dist}_H(u, v) \leq \alpha \cdot \text{dist}_G(u, v) + \beta$ holds for all $u, v \in V$. While the number of edges in a spanner may be much smaller than that of the original graph, the number of vertices is the same for both, leaving further potential for compression untapped. For illustration, consider the road network of Europe with about 50 million vertices [BMSW13], any spanner of which must again have about 50 million vertices and edges. However, to approximately represent distances in Europe's road network one may also merge nearby vertices into super-vertices, thus achieving a much better compression of the network. This is akin to the visual process of zooming out of a graphical representation of the map, where neighbored vertices fade into each other and edges between merged vertices vanish. At a large enough zoom level, the entire network merges into a single vertex.

In this paper we propose and study a new framework for contracting networks that formalizes this intuitive idea and makes it applicable to general graphs (even without

*E-Mail: {bernstein, daeubel, muetze, smolny}@math.tu-berlin.de

†Supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt. E-Mail: disser@mathematik.tu-darmstadt.de

‡E-Mail: max.klimm@hu-berlin.de

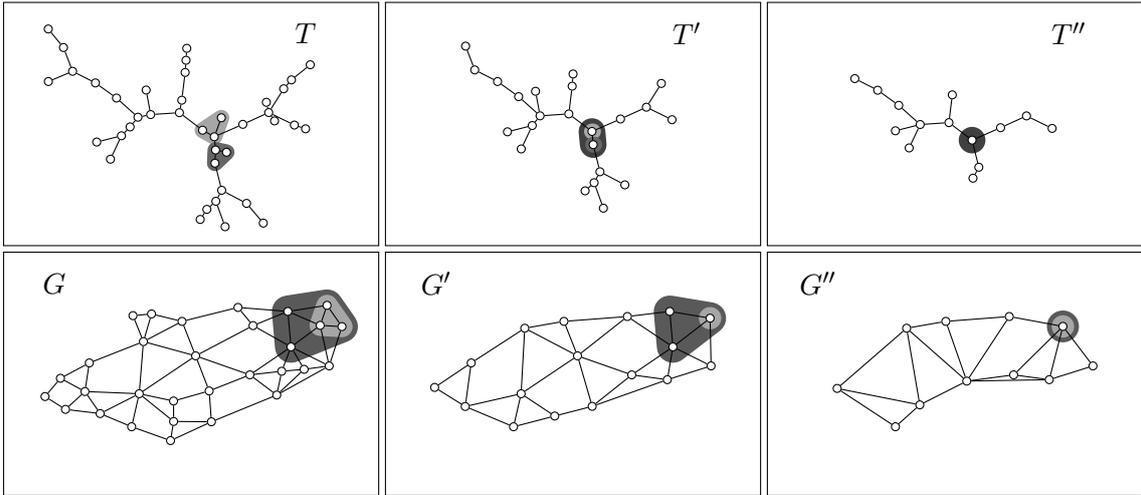


FIGURE 1. The top part of the figure shows an instance of CONTRACTION with tolerance function $\varphi(x) = x/\alpha - \beta$ with $(\alpha, \beta) = (5/4, 3)$ on a tree T , yielding the tree T' , and again on T' yielding T'' . The bottom part shows an instance of CONTRACTION with $(\alpha, \beta) = (4/3, 3)$ on a planar graph G , yielding the graph G' , and again on G' yielding G'' . The edge lengths for the graphs T , T' , T'' and G are given by geometric distances. Some subsets of vertices that are contracted to super-vertices are highlighted in gray.

metric embedding). Specifically, we study a contraction problem on graphs where a subset of edges $C \subseteq E$ is contracted. We denote the resulting simple graph obtained from G by contracting the edges in C and by deleting resulting loops and multiple edges, keeping only the shortest edge between any two vertices, by G/C . For any two vertices in G , we compare their distance in G with the distance of the corresponding super-vertices in G/C .

It is interesting to contrast this concept with graph spanners. When constructing a spanner, the length of the removed edges is implicitly set to ∞ , resulting in an overall increase of distances. On the other hand, a contraction implicitly sets the length of the contracted edges to zero, leading to an overall decrease of distances. For both problems, the ultimate goal is to reduce the complexity of the network while maintaining an approximation guarantee on the distances.

The following example shows that contractions may be better suited than spanners to achieve that goal. In a subgraph with small radius, a spanner can at best result in a spanning tree of the same order, while a contraction can reduce the whole subgraph to a single vertex (both entail essentially the same multiplicative distance distortion). In addition, the contraction may also merge many edges entering the contracted subgraph. Clearly, the objective here is to maximize the total number of contracted and deleted edges, as this minimizes the memory required to represent the resulting network in a computer (using e.g. adjacency lists).

Given the results presented in this paper and the known results for spanners (discussed in detail below), we further believe that the combination of spanners and contractions is very powerful, promising and flexible. As the former only increases and the latter only decreases the distances, the respective distortion guarantees provably also hold for the overall distortion. In fact, both effects may even compensate each other. This is true *regardless* of the order in which both compression operations are applied, even when they are applied repeatedly.

TABLE 1. Overview of algorithmic and hardness results presented in this paper.

| Problem | Graph classes | | | |
|---|----------------------------|--------------------------|------------------------------|---|
| | Path | Tree | Cycle | General |
| CONTRACTION | | | | |
| addit. ($\alpha=1$), unit lg. ($\ell=1$) | $\mathcal{O}(n)$ [Th. 2] | $\mathcal{O}(n)$ [Th. 4] | $\mathcal{O}(n)$ [Th. 3] | $m^{\frac{1}{2}-\varepsilon}$ -inapx. ^a [Th. 10] |
| affine (α, β), unit lg. ($\ell=1$) | | | | |
| addit. ($\alpha=1$) | $\mathcal{O}(n^3)$ [Th. 5] | | NP-hard [Th. 7] | $n^{1-\varepsilon}$ -inapx. [Th. 9] |
| affine (α, β) | | | | |
| WEAK CONTRACTION | | | | |
| additive ($\alpha=1$) | $\mathcal{O}(n^5)$ [Th. 6] | | NP-hard ^b [Th. 7] | |
| affine (α, β) | | | | $n^{1-\varepsilon}$ -inapx. ^c [Th. 12] |

^aeven for bipartite graphs and $\beta = 1$

^balso NP-hard for planar graphs with arb. large girth, $(\alpha, \beta) = (2, 0)$, and unit lg. ($\ell = 1$) [Th. 11].

^ceven if $(\alpha, \beta) = (3/2, 0)$.

In order to measure the distance distortion of the contraction, we assume a non-decreasing tolerance function $\varphi: \mathbb{R} \rightarrow \mathbb{R}$, similar to the corresponding function for spanners, see e.g. [BKMP05]. We are interested in computing contractions that preserve distances in the following sense: For any two vertices u and v at distance d in G , the distance of the corresponding vertices in the contracted graph G/C must be at least $\varphi(d)$. If this condition is satisfied, we call C a φ -distance preserving contraction, or φ -contraction for short. Formally, the algorithmic problem CONTRACTION considered in this paper is to compute for a given graph $G = (V, E)$ with edge lengths $\ell: E \rightarrow \mathbb{R}_{>0}$ and a given tolerance function φ , a φ -contraction $C \subseteq E$ such that the number of contracted and deleted edges is maximized. We are specifically interested in the case where the tolerance function φ is an affine function $\varphi(x) = x/\alpha - \beta$ for real-valued parameters $\alpha \geq 1$ and $\beta \geq 0$. We then simply write (α, β) -contraction instead of φ -contraction. See Figure 1 for some example instances of the problem CONTRACTION.

When considering the case of a purely multiplicative error ($\beta = 0$), a slight subtlety has to be taken into account. Specifically, for a graph with positive edge lengths it is not feasible to contract a single edge. Therefore, we propose a slight modification of our original model: We say that a set $C \subseteq E$ of edges of G is a *weak φ -distance preserving contraction*, or *weak φ -contraction* for short, if it does not contract the entire graph and, for any two vertices u and v at distance d in G , the distance of the corresponding vertices in G/C is either zero or at least $\varphi(d)$. We will refer to the corresponding algorithmic problem as WEAK CONTRACTION. Put differently, in a weak contraction, the distances between different super-vertices satisfy the given distortion guarantee, but for vertices belonging to the same super-vertex, no guarantee is given.

1.1. Our results. In this paper, we present a comprehensive picture of the algorithmic complexity of the described contraction problems. Recall that we are given an input graph with edge lengths and tolerance function φ , and our goal is to compute a (weak) contraction that maximizes the total number of contracted and deleted edges. Our main results concern affine tolerance functions $\varphi(x) = x/\alpha - \beta$ with parameters $\alpha \geq 1$ and $\beta \geq 0$. For the reader's convenience, our results are summarized in Tables 1, and 2. Within the tables and throughout this paper, n and m denote the number of vertices and edges, respectively, of the input graph under consideration.

TABLE 2. Overview of asymptotic bounds presented in this paper.

| CONTRACTION with unit lg. ($\ell=1$) | # of edges in G/C | Time | Reference |
|---|------------------------|------------------|-----------|
| $(\alpha, \beta) = (2k - 1, 1)$ | $n^{1+1/k}$ | $\mathcal{O}(m)$ | [Th. 13] |
| $(\alpha, \beta) = (2\log_2 n - 1, 1)$ | $2n$ | $\mathcal{O}(m)$ | [Cor. 14] |
| $(\alpha, \beta) = (k - 1, 1)$ | $\Omega(n^{1+1/k})$ | — | [Th. 15] |
| $(\alpha, \beta) = (1, 2k)$ | $m - km/n$ | $\mathcal{O}(m)$ | [Th. 16] |
| CONTRACTION with unit lg. ($\ell=1$) and min. degree D | # of vertices in G/C | Time | Reference |
| $(\alpha, \beta) = (5, 1)$ | n/D | $\mathcal{O}(m)$ | [Th. 17] |
| $(\alpha, \beta) = (k, 1)$ | $\Omega(n/(kD))$ | — | [Th. 18] |

Algorithmic results. We develop linear time greedy algorithms for CONTRACTION with unit lengths on paths, cycles, and on trees with $\alpha = 1$ (Theorems 2, 3 and 4). The first two algorithms are inspired by LP rounding techniques, the latter algorithm relies on a structural characterization of optimal solutions.

We present dynamic programming algorithms solving CONTRACTION and WEAK CONTRACTION on trees in time $\mathcal{O}(n^3)$ or $\mathcal{O}(n^5)$, respectively (Theorems 5 and 6). These dynamic programs compute optimal solutions on subtrees, in the latter case combining several Pareto optimal solutions in a two-dimensional parameter space (hence the larger running time).

Note that instead of maximizing the number of contracted and deleted edges, we could optimize for α or β while fixing the other parameters. The resulting problems are polynomially equivalent to our setting, via binary search over one of the parameters.

Hardness results. We complement these algorithms by several hardness results. First we consider the purely additive case where $\alpha = 1$. We show that here both CONTRACTION and WEAK CONTRACTION are NP-hard on cycles for any fixed $\beta > 0$, by a reduction of a variant of PARTITION (Theorem 7). As mentioned before, both problems can be solved efficiently on graphs without cycles, and there is a linear time algorithm for CONTRACTION on cycles with unit lengths. By reductions from CLIQUE we show that both the general as well as the unit lengths case of CONTRACTION with $\alpha = 1$ are hard to approximate within factors of $n^{1-\varepsilon}$ or $m^{1/2-\varepsilon}$, respectively (Theorem 9 and Theorem 10).

Further we consider the purely multiplicative case where $\beta = 0$ (here CONTRACTION is trivial). We show that in this case WEAK CONTRACTION is NP-hard on planar graphs with arbitrarily large girth and unit length edges by a reduction from a special case of PLANAR 3SAT (Theorem 11). Since these graphs are locally tree-like, this result constitutes another rather sharp separation from the polynomially solvable tree case. Furthermore, we show that the problem is hard to approximate within a factor of $n^{1-\varepsilon}$ by a reduction from INDEPENDENT SET (Theorem 12).

Asymptotic bounds. We now discuss our asymptotic bounds for contractions. In this setting, we are interested in (non-optimal) contractions for graphs with unit lengths that can be computed efficiently despite the above-mentioned hardness results. We prove that for any $k \geq 1$ any graph G has a $(2k - 1, 1)$ -contraction C such that G/C has at most $n^{1+1/k}$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$ (Theorem 13) by successively growing clusters around center vertices. Assuming Erdős' girth conjecture, we show a corresponding (not tight) lower bound (Theorem 15). For a purely additive error, we show that for any integer $k \geq 1$ any graph with unit length edges has a $(1, 2k)$ -contraction

C with objective value at least km/n that can be computed in time $\mathcal{O}(m)$ (Theorem 16), which is asymptotically best possible for paths. The idea here is to remove the k vertices of highest degrees.

One possible advantage of contraction compared to spanners is the potentially significant reduction of *vertices* as well as edges. To ground this intuition, we exhibit a contraction that significantly reduces the number of vertices in a graph with minimum degree D to $\mathcal{O}(n/D)$ (Theorem 17). We also present a lower bound (Theorem 18) showing that we cannot guarantee $o(n/D)$ vertices, even if we allow larger approximation error.

1.2. Comparison with previous results. There are several models aiming to compress graphs while preserving distances. They differ by their choice of compression operation, such as replacing the graph by a subgraph or minor, and by whether the aim is to preserve all or only certain distances.

As discussed before, graph spanners are a concept closely related to contractions, where the length of removed edges is set to ∞ rather than to 0. Our results highlight further intrinsic similarities of the two models. As for contractions, computing optimal spanners is NP-hard (see [PS89, LS93]), even to approximate (e.g. [Kor98, EP07]). But it is still possible to obtain powerful asymptotic guarantees in both models. In particular, our $(2k - 1, 1)$ -contraction with $\mathcal{O}(n^{1+1/k})$ edges has a clear analogy to the classic $(2k - 1, 0)$ -spanner with the same number of edges [ADD⁺93] (note that the additive error of 1 in our result is strictly necessary, as discussed above). There is, however, a major difference between the two results: whereas the $(2k - 1, 0)$ -spanner can trivially be shown to be optimal assuming Erdős' girth conjecture, applying this conjecture to the contraction model only yields a lower bound of $n^{1+1/(2k)}$ edges for a $(2k - 1, 1)$ -contraction. Closing this gap thus remains as an interesting open problem in the contraction model, whose solution would likely yield further insight into the relationship to spanners.

It is interesting to note that the clustering yielding our $(2k - 1, 1)$ -contraction was previously used in [PS89] to obtain a $(4r + 1, 0)$ -spanner of the same density. On the other hand, no deterministic linear time algorithm computing a $(2k - 1, 0)$ -spanner is known, though [BS03] achieves randomized linear time. Meanwhile our $(2k - 1, 1)$ -contraction can be constructed deterministically in linear time.

For unweighted graphs, there are also spanner results that significantly sparsify the graph at the cost of a purely additive error: For example, a $(1, 2)$ -spanner with $\mathcal{O}(n^{3/2})$ edges [ACIM99], or a $(1, 6)$ -spanner with $\mathcal{O}(n^{4/3})$ edges [BKMP05]. We do not know if analogous results are possible in the contraction model. Lower bounds would also be interesting; for spanners, it is known that $\mathcal{O}(n^{4/3})$ is the best possible guarantee if we want a constant additive error [AB16]. Finally, for spanners there are results that combine multiplicative and additive error, such as the $(k, k - 1)$ -spanner of [BKMP05].

Gupta [Gup01] considered the problem of approximating a tree metric on a subset of the vertices by another tree, and gave a linear time algorithm computing an 8-approximation. As Chan et al. [CXKR06] observed later, on complete binary trees a solution of minimum distortion is always achieved by a minor (with possibly different edge lengths) of the input tree, so this seems to be the first investigation of contractions that approximate graph distances. Krauthgamer et al. [KNZ14] considered an extension to general graphs, studying the size of minors preserving all distances between a given terminal set of fixed size. Cheung et al. [CGH16] introduced a multiplicative distortion to this model. As here no two terminals may be merged, these approaches cannot compress a graph at all if every vertex is a terminal.

The *pairwise preservers* due to Coppersmith et al. [CE06] combine spanners with the aim of preserving only terminal distances. Given a weighted undirected graph G and a set

of terminal pairs, a pair-wise preserver is a spanning subgraph inducing exactly the same terminal distances as G . Coppersmith et al. proved that for every such graph and every set of $k = \mathcal{O}(n^{1/2})$ terminal pairs there exists a pair-wise preserver of size $\mathcal{O}(kn^{1/2} + n)$.

1.3. Further related work. The preservation of graph properties other than distances has been studied as well. Biedl et al. [BBV00] considered contractions in capacitated networks with the goal of maintaining the maximum flow in the network. Here an edge e is called *useless*, if for every capacity function there is a maximum flow not using e . Biedl et al. showed that finding all useless edges is NP-complete, but solvable in $\mathcal{O}(n^2)$ time on certain planar graphs. For undirected networks, Misiolek et al. [MC05] gave an algorithm finding all useless edges in $\mathcal{O}(n + m)$ time. Toivonen et al. [ZMT10] considered a more general model aiming to maintain the quality of paths with respect to any given function, e.g., distance or capacity. They investigated strategies of removing edges, without decreasing the quality of the best path between any pair of vertices.

Graph simplification problems have also been studied in several other contexts, and we conclude this section by mentioning two such examples: Hübler et al. [HKBG08] studied a problem related to graph mining, examining how to choose an induced subgraph with a given number of vertices and with similar topological properties as the input graph. Dörfler and Bullo [DB13] considered the problem of exactly preserving the effective resistance among a set of terminal vertices of an electrical network via an algebraic network compression known as *Kron reduction*.

1.4. Outline of this paper. In Section 2 we introduce important definitions and notations that will be used throughout this paper. In Section 3 we discuss our three greedy algorithms for solving CONTRACTION with unit lengths on paths, cycles and trees (the latest result requires $\alpha = 1$). In Section 4 we present our two dynamic programming algorithms for CONTRACTION and WEAK CONTRACTION on trees. Sections 5 and 6 are devoted to proving our hardness results, focussing on the case of additive and multiplicative error, respectively. In Section 7 we present our asymptotic results on contractions.

2. PRELIMINARIES

Throughout this paper we consider simple undirected graphs G (without parallel edges or loops). We let $V(G)$ and $E(G)$ denote the vertex and edge set of G , respectively, and we define $n(G) := |V(G)|$ and $m(G) := |E(G)|$. If the context is clear, we simply write V , E , n and m . We also use the notation $[n] := \{1, 2, \dots, n\}$. We assume that G is connected, otherwise the contraction problem can be solved independently for each connected component. Edge lengths are given by a function $\ell: E \rightarrow \mathbb{R}_{>0}$. The *distance* $\text{dist}_\ell(u, v)$ between two vertices u and v is the length of a shortest path between u and v in G with respect to ℓ .

Given a subset of edges $C \subseteq E$, we denote the resulting simple graph obtained from G by contracting the edges in C , deleting resulting loops and keeping only the shortest edge between any two vertices by G/C . We denote the number of deleted loops and multi-edges by $\Delta(C)$ (thus $m(G/C) = m(G) - |C| - \Delta(C)$). Instead of contracting a set $C \subseteq E$ of edges in G , setting their edge lengths to zero has the same effect on the distances in the resulting graph. This is somewhat cleaner conceptually, so we will often adopt this viewpoint. Specifically, we let ℓ_C be the new length function that assigns 0 to every edge in C , and that is equal to the original edges lengths ℓ on the edges $E \setminus C$.

A *tolerance function* is a non-decreasing function $\varphi: \mathbb{R} \rightarrow \mathbb{R}$. Roughly speaking, this function describes by how much the distance between two vertices may drop when contracting edges (i.e., setting edge lengths to zero). Formally, given a graph G with edge

lengths ℓ and a tolerance function φ , we say that a subset of edges $C \subseteq E$ is a φ -distance preserving contraction or φ -contraction for short, if

$$\text{dist}_{\ell_C}(u, v) \geq \varphi(\text{dist}_{\ell}(u, v)) \quad (1)$$

holds for any two vertices u and v in G . Similarly, we say that C is a *weak* φ -distance preserving contraction or *weak* φ -contraction for short, if (1) or $\text{dist}_{\ell_C}(u, v) = 0$ holds for any two vertices u and v , and if the graph (V, C) is disconnected (equivalently, if G/C is not a single vertex). The last condition prevents solutions $C \subseteq E$ for which the graph is contracted to a single vertex. If $\varphi(x) = x/\alpha - \beta$, then we simply write (weak) (α, β) -contraction instead of (weak) φ -contraction.

An *instance* of the problem CONTRACTION or WEAK CONTRACTION is a triple (G, ℓ, φ) , where G is the underlying graph, ℓ the length function and φ the tolerance function, and the objective is to find a (weak) φ -distance preserving contraction $C \subseteq E$, such that

$$\Phi(C) := |C| + \Delta(C) = m(G) - m(G/C) \quad (2)$$

is maximized. This quantity equals the number of edges we save when going from G to G/C . Note that for instance on trees we have $\Phi(C) = |C|$ for any (weak) contraction C .

(WEAK) CONTRACTION

Input: A graph $G = (V, E)$ with edge lengths $\ell: E \rightarrow \mathbb{R}_{>0}$ and a non-decreasing function $\varphi: \mathbb{R} \rightarrow \mathbb{R}$.

Output: A (weak) φ -distance preserving contraction $C \subseteq E$ maximizing $\Phi(C)$.

In this context we sometimes refer to a set of edges that forms a (weak) contraction as a *feasible* solution, and to a (weak) contraction of maximum size as an *optimal* solution.

We begin by proving that our contraction model behaves nicely when contracting edges in phases, i.e., the total error is simply the error accumulated over the contraction phases (but not more). To state this result we denote the composition of tolerance functions φ and ψ as $(\psi \circ \varphi)(x) := \psi(\varphi(x))$.

Theorem 1. *Let C be a (weak) φ -contraction for G , and let C' be a (weak) ψ -contraction for G/C . Then $C \cup C'$ is a (weak) $(\psi \circ \varphi)$ -contraction for G .*

Proof. We only prove the statement for contractions φ and ψ . The proof for weak contractions works analogously. Let ℓ denote the edge lengths of G and consider a pair of vertices $u, v \in V(G)$. Then we have $\text{dist}_{\ell_{C \cup C'}}(u, v) \geq \psi(\text{dist}_{\ell_C}(u, v))$ by the definition of C' and $\text{dist}_{\ell_C}(u, v) \geq \varphi(\text{dist}_{\ell}(u, v))$ by the definition of C . Combining these inequalities and using that ψ is non-decreasing we obtain $\text{dist}_{\ell_{C \cup C'}}(u, v) \geq \psi(\varphi(\text{dist}_{\ell}(u, v)))$, as desired. ■

Note that Theorem 1 only talks about the *feasibility* of repeated contractions, but not about their *optimality* when searching for contractions of maximum cardinality. With respect to solution quality, contracting in phases may be arbitrarily bad: Consider a star with k unit length edges and additive tolerance functions $\varphi(x) = \psi(x) = x - 1$. An optimum $(\psi \circ \varphi)$ -contraction contains all k edges, whereas finding an optimal φ -contraction C and then an optimal ψ -contraction of G/C allows contracting only one edge in each phase, leading to a $(\psi \circ \varphi)$ -contraction of value 2.

3. GREEDY ALGORITHMS

In this section we consider three special cases of the problem CONTRACTION with affine tolerance function $\varphi(x) = x/\alpha - \beta$. We obtain simple greedy algorithms computing maximum size φ -contractions in $\mathcal{O}(n)$ time on paths and cycles with unit lengths, and on trees with unit lengths and $\alpha = 1$.

3.1. Paths with unit length edges. In this section we consider the special case of contracting a path P_n with $n - 1$ unit length edges $\ell = 1$ and the tolerance function $\varphi(x) = x/\alpha - \beta$. In this case optimal solutions have a very special structure, which leads to a straightforward greedy algorithm running in linear time. Recall that as a path is a tree, our objective function satisfies $\Phi(C) = |C|$ for any contraction C .

Observe that a solution $C \subseteq E(P_n)$ for the instance (P_n, ℓ, φ) of the problem CONTRACTION is feasible, if and only if every subpath $P' \subseteq P_n$ satisfies the condition

$$|P' \cap C| \leq (1 - 1/\alpha)|E(P')| + \beta. \quad (3)$$

This observation leads to the following natural greedy algorithm $\text{GREEDY}(P_n, \alpha, \beta)$: The algorithm considers the edges e_1, e_2, \dots, e_{n-1} of P_n as they are encountered when starting from one of the two end vertices of P_n . It iteratively constructs a solution C for the subpath on the first i edges e_1, e_2, \dots, e_i for $i = 1, 2, \dots, n - 1$, by initializing $C := \emptyset$, and by adding the edge e_i to C if and only if the condition $|C| + 1 \leq (1 - 1/\alpha)i + \beta$ is satisfied (so after adding e_i to C , (3) is still satisfied).

Theorem 2. *Let P_n be a path with unit length edges $\ell = 1$ and consider the tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha, \beta \geq 1$. The set of edges computed by the algorithm $\text{GREEDY}(P_n, \alpha, \beta)$ is an optimal solution for the instance (P_n, ℓ, φ) of the problem CONTRACTION, and it is computed in time $\mathcal{O}(n)$.*

Proof. Let $C \subseteq E(P_n)$ be the set of edges computed by the algorithm $\text{GREEDY}(P_n, \alpha, \beta)$. Clearly, we have $|C| = \lfloor (1 - 1/\alpha)|E(P_n)| + \beta \rfloor$, and this is optimal according to (3). However, it remains to show that C is feasible. For $1 \leq i \leq j \leq n - 1 = |E(P_n)|$ we let $P_{i,j}$ denote the subpath of P_n formed by the edges e_i, e_{i+1}, \dots, e_j . By the definition of our algorithm we know that $|P_{1,i} \cap C| = \lfloor (1 - 1/\alpha)i + \beta \rfloor$, from which we obtain that

$$\begin{aligned} |P_{i,j} \cap C| &= |P_{1,j} \cap C| - |P_{1,i-1} \cap C| = \lfloor (1 - 1/\alpha)j + \beta \rfloor - \lfloor (1 - 1/\alpha)(i - 1) + \beta \rfloor \\ &\leq (1 - 1/\alpha)j + \beta - ((1 - 1/\alpha)(i - 1) + \beta - 1) \\ &= (1 - 1/\alpha)(j - i + 1) + 1 \\ &\leq (1 - 1/\alpha)|E(P_{i,j})| + \beta, \end{aligned}$$

where we used the assumption $\beta \geq 1$ in the last step. Using (3) it thus follows that C is feasible. \blacksquare

3.2. Cycles with unit length edges. In this section we consider the special case of contracting a cycle C_n with n vertices and unit length edges $\ell = 1$ and the tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1, \beta \geq 0$. For this case we present a greedy algorithm running in linear time. The main purpose of this result is to clearly separate the polynomially solvable cases of CONTRACTION from the NP-hard cases, and the case of a cycle with unit length edges precisely forms this boundary on the polynomially solvable side. Recall in this context that we can solve CONTRACTION in polynomial time on any tree (this will be proved in Section 4.1 below), and that CONTRACTION is NP-hard already on a cycle for $\alpha = 1$ (with arbitrary edge lengths; we will show this in Section 5.1 below).

We first argue that on a cycle it is equivalent to maximize the number of contracted edges $|C|$ or to maximize our objective function $\Phi(C)$ defined in (2). This is because the set of pairs $(|C|, \Phi(C))$ for all feasible contractions C in a cycle $G = C_n$ is given by $\{(1, 1), (2, 2), \dots, (n - 3, n - 3), (n - 2, n - 1), (n - 1, n), (n, n)\}$, so it forms a monotone function, implying that maximizing either one of the two quantities is equivalent. Based on this argument, for the rest of this section we consider maximizing the number $|C|$ of contracted edges.

Observe that a solution $C \subseteq E(C_n)$ (C_n is the cycle we want to contract, and C is the set of edges to be contracted) for the instance (C_n, ℓ, φ) of the problem CONTRACTION is

feasible, if and only if every subpath $P \subseteq C_n$ of length $d := |E(P)| \in \{1, 2, \dots, n-1\}$ satisfies the condition

$$|P \cap C| \leq \lfloor d - \min\{d, n-d\}/\alpha + \beta \rfloor. \quad (4)$$

Rounding down on the right-hand side of (4) is justified because $|P \cap C|$ is always an integer.

Defining

$$\lambda' := \min_{d \in \{1, 2, \dots, n-1\}} \frac{\lfloor d - \min\{d, n-d\}/\alpha + \beta \rfloor}{d}, \quad (5a)$$

$$\lambda := \min\{1, \lambda'\}, \quad (5b)$$

we obtain from (4) that $\lambda \in [0, 1]$ is the maximal amount by which we can contract each edge in a uniform *fractional* solution. Inspired by the rounding technique from [BOR80], we turn this fractional solution into an integer optimal solution, yielding the following greedy algorithm $\text{GREEDY}(C_n, \alpha, \beta)$: The algorithm considers the edges e_1, e_2, \dots, e_n of C_n as they are encountered when walking around the cycle. It iteratively constructs a solution C by initializing $C := \emptyset$ and by adding the edge e_i to C if and only if $\lfloor \lambda i \rfloor - \lfloor \lambda(i-1) \rfloor = 1$ for all $i = 1, 2, \dots, n$ (since $\lambda \in [0, 1]$, this difference is always either 0 or 1). Note that we contract all edges of C_n if and only if $\lambda = 1$.

Theorem 3. *Let C_n be a cycle with unit length edges $\ell = 1$ and consider the tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$, $\beta \geq 0$. The set of edges computed by the algorithm $\text{GREEDY}(C_n, \alpha, \beta)$ is an optimal solution for the instance (C_n, ℓ, φ) of the problem CONTRACTION , and it is computed in time $\mathcal{O}(n)$.*

The next lemma shows that the contraction computed by our algorithm has the maximum size.

Lemma 3.1. *For any feasible solution $C \subseteq E(C_n)$ we have $|C| \leq \lfloor \lambda n \rfloor$ with λ defined in (5).*

Proof. If $\lambda = 1$ this inequality is trivial. So let us assume that $\lambda = \lambda' < 1$ and that the minimum in (5a) is attained for some $d \in \{1, 2, \dots, n-1\}$. Starting at some vertex u of the cycle, we walk along the cycle and cover it with n consecutive paths P_1, P_2, \dots, P_n of length d each (P_{i+1} starts where P_i ends). The sum of the lengths of the paths is nd , so this process ends at the starting vertex u , and each edge of the cycle and each edge of C is covered exactly d times. We therefore obtain

$$|C| = \frac{1}{d} \sum_{i=1}^n |P_i \cap C| \stackrel{(4)}{\leq} \sum_{i=1}^n \frac{\lfloor d - \min\{d, n-d\}/\alpha + \beta \rfloor}{d} \stackrel{(5)}{=} \lambda n.$$

As $|C|$ must be integral this inequality yields the desired bound $|C| \leq \lfloor \lambda n \rfloor$. ■

With Lemma 3.1 in hand, we are now ready to prove Theorem 3.

Proof of Theorem 3. In this proof we will use that for any two real numbers x and y we have

$$\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor, \quad (6a)$$

$$\lfloor x \rfloor - \lfloor y \rfloor \leq \lfloor x - y \rfloor. \quad (6b)$$

Let $C \subseteq E(C_n)$ be the set of edges computed by the algorithm $\text{GREEDY}(C_n, \alpha, \beta)$. Clearly, we have $|C| = \sum_{i=1}^n (\lfloor \lambda i \rfloor - \lfloor \lambda(i-1) \rfloor) = \lfloor \lambda n \rfloor$, which is optimal by Lemma 3.1. However, it remains to show that C is feasible. We consider a path P of length $d :=$

$|E(P)| \in \{1, 2, \dots, n-1\}$ on the edges $e_k, e_{k+1}, \dots, e_{k+d-1}$ (indices are considered cyclically modulo n , so $e_{n+i} = e_i$). We distinguish two cases: If $k + d - 1 \leq n$, we have

$$|P \cap C| = \sum_{i=k}^{k+d-1} (\lfloor \lambda i \rfloor - \lfloor \lambda(i-1) \rfloor) = \lfloor \lambda(k+d-1) \rfloor - \lfloor \lambda(k-1) \rfloor \stackrel{(6b)}{\leq} \lceil \lambda d \rceil. \quad (7)$$

If $k + d - 1 > n$, we obtain

$$\begin{aligned} |P \cap C| &= \sum_{i=k}^n (\lfloor \lambda i \rfloor - \lfloor \lambda(i-1) \rfloor) + \sum_{i=1}^{d-n+k-1} (\lfloor \lambda i \rfloor - \lfloor \lambda(i-1) \rfloor) \\ &= \lfloor \lambda n \rfloor - \lfloor \lambda(k-1) \rfloor + \lfloor \lambda(d-n+k-1) \rfloor \\ &\stackrel{(6a)}{\leq} \lfloor \lambda(d+k-1) \rfloor - \lfloor \lambda(k-1) \rfloor \stackrel{(6b)}{\leq} \lceil \lambda d \rceil. \end{aligned} \quad (8)$$

Applying (5) and using that $\lceil \lfloor x \rfloor \rceil = \lfloor x \rfloor$ shows that the right-hand sides of (7) and (8) can both be bounded from above by $\lfloor d - \min\{d, n-d\} / \alpha + \beta \rfloor$, proving that C is indeed feasible by (4). \blacksquare

3.3. Trees with unit length edges and additive error. In this section we consider the special case of contracting a tree T with unit length edges $\ell = 1$ and the tolerance function $\varphi(x) = x - \beta$ (purely additive error; we can assume w.l.o.g. that β is an integer). Note that in this setting the objective function defined in (2) satisfies $\Phi(C) = |C|$ for any contraction C . It turns out that in this case, optimal solutions have a very special structure that can be exploited to compute them in linear time. Specifically, an optimal solution is obtained by taking all edges of T which have the property that only short paths start from one of its end vertices. Formally, for the tree T and $d \in \mathbb{N}_{\geq 0}$, we let $L(T, d)$ denote the set of all edges e of T which have one end vertex v such that all paths that start at v and do not contain e have length at most $d-1$ (together with e these paths have length at most d). E.g., we have $L(T, 0) = \emptyset$, and the set $L(T, 1)$ are all the edges incident to a leaf (see Figure 2).

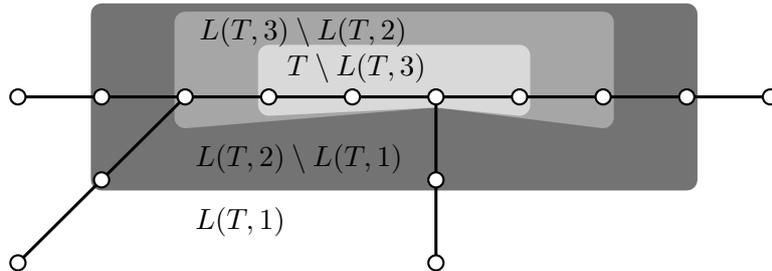


FIGURE 2. Illustration of the sets of edges $L(T, d)$, $d = 1, 2, 3$. The set $L(T, 3)$ is an optimal solution of the problem CONTRACTION with $\beta = 6$.

Clearly, the set $L(T, d)$ can be computed in linear time by repeatedly removing all leaves of T in d rounds. This is a variant of the well-known linear time algorithm to compute the so-called center of a tree (see [Ski08, Section 15.11]).

Theorem 4. *Let T be a tree with unit length edges $\ell = 1$ and consider the tolerance function $\varphi(x) = x - \beta$, $\beta \in \mathbb{N}_{\geq 0}$. If β is even, the set of edges $L(T, d)$ with $d := \lfloor \beta/2 \rfloor$ is an optimal solution for the instance (T, ℓ, φ) of the problem CONTRACTION. If β is odd, $L(T, d) \cup \{e\}$, $e \in E \setminus L(T, d)$, is an optimal solution. These solutions can be computed in time $\mathcal{O}(n)$.*

we obtain that the number of edges from D' on P' is at most $\beta - 1 + 1 = \beta$, as desired. This completes the proof. \blacksquare

4. DYNAMIC PROGRAMS FOR GENERAL TREES

In this section we describe dynamic programming algorithms for the problems CONTRACTION and WEAK CONTRACTION on trees with general edge lengths and affine tolerance functions. Recall that on trees our objective function satisfies $\Phi(C) = |C|$ for any contraction C .

4.1. Contraction on trees. In this section we describe a dynamic programming algorithm for the problem of computing an optimal contraction of a tree T with arbitrary edge lengths $\ell: E \rightarrow \mathbb{R}_{>0}$ and an affine tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$, $\beta \geq 0$, generalizing the solution for the special case presented at the beginning of the previous section. The goal is to prove the following result.

Theorem 5. *Let T be a tree with edge lengths $\ell: E \rightarrow \mathbb{R}_{>0}$ and consider the tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$, $\beta \geq 0$. An optimal solution for the instance (T, ℓ, φ) of the problem CONTRACTION can be computed by dynamic programming in time $\mathcal{O}(n^3)$.*

Observe that a solution $C \subseteq E$ is feasible if and only if for any two vertices u and v of T we have $\text{load}_{C,\alpha}(u, v) \leq \beta$, where the *load between u and v* is defined as

$$\text{load}_{C,\alpha}(u, v) := \text{dist}_\ell(u, v)/\alpha - \text{dist}_{\ell_C}(u, v) \quad (9a)$$

(recall (1)). For any vertex v of T we also define the *load of T at v* as

$$\text{load}_{C,\alpha}(T, v) := \max\{\text{load}_{C,\alpha}(u, v) : u \in V\}. \quad (9b)$$

Note that $\text{load}_{C,\alpha}(T, v) \geq 0$, as we have $\text{load}_{C,\alpha}(v, v) = 0$. The next lemma states a criterion when feasible solutions of subtrees can be combined to a feasible solution of the entire tree. The definitions (9a), (9b) and the lemma are illustrated in Figure 4.

Lemma 4.1. *Consider a partition of T into two subtrees T_1 and T_2 that only have a vertex $v \in V$ in common. Then $C \subseteq E$ is a feasible solution for the instance (T, ℓ, φ) of the problem CONTRACTION if and only if the following two conditions hold: $C \cap E(T_1)$ and $C \cap E(T_2)$ are feasible solutions for the instances (T_1, ℓ, φ) and (T_2, ℓ, φ) respectively; and we have $\text{load}_{C,\alpha}(T_1, v) + \text{load}_{C,\alpha}(T_2, v) \leq \beta$.*

Proof. Observe that the path between two vertices $u \in T_1$ and $w \in T_2$ contains the vertex v , so we obtain $\text{load}_{C,\alpha}(u, w) = \text{load}_{C,\alpha}(u, v) + \text{load}_{C,\alpha}(v, w)$ from (9a). Using (9b) it follows that the condition $\text{load}_{C,\alpha}(u, w) \leq \beta$ holding for all such pairs of vertices u, w is equivalent to $\text{load}_{C,\alpha}(T_1, v) + \text{load}_{C,\alpha}(T_2, v) \leq \beta$. \blacksquare

We will use this lemma to formulate our dynamic programming algorithm. The idea is to compute optimal solution for subtrees and combining them to an optimal solution for the entire tree.

To describe the algorithm we introduce a few definitions. An *ordered rooted tree* is a rooted tree with a specified left-to-right ordering for the children of each vertex. Given the tree T , we can pick an arbitrary vertex as the root, and for each descendant of the root an arbitrary left-to-right ordering of its children, yielding an ordered rooted tree (different roots and orderings yield different ordered rooted trees, but any one of them is good for our purposes). We slightly abuse notation in the following and use T to denote this ordered rooted tree. All trees considered in the rest of this section are ordered and rooted. For any vertex v of T , we let T_v denote the subtree of T rooted at v , and we use $c(v)$ to denote the number of children of v . If $u_1, u_2, \dots, u_{c(v)}$ are the children of v (in the specified ordering), we write $T_{v,i}$, $i \in \{1, \dots, c(v)\}$, for the subtree of T that

contains v , u_i and all the descendants of u_i . We also define $T_{v,0} := \{v\}$. Furthermore, we define $T_{v,i}^+ := \bigcup_{0 \leq j \leq i} T_{v,j}$, so we have $T_v = T_{v,c(v)}^+$. These definitions are illustrated in Figure 4.

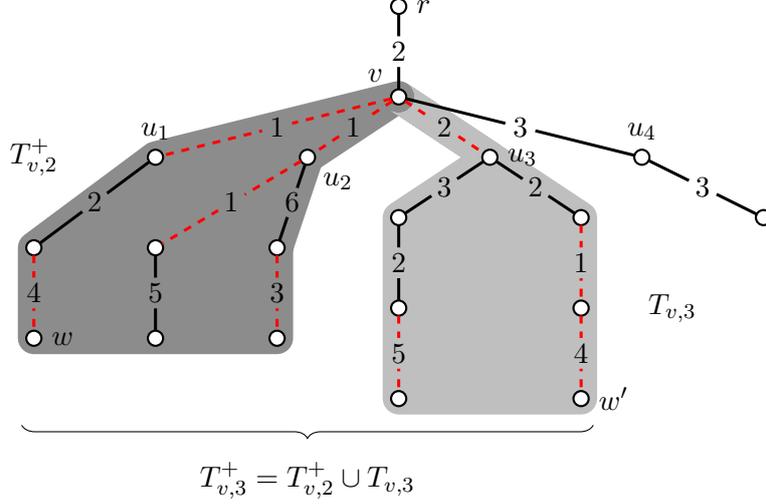


FIGURE 4. Illustration of the definitions of Section 4.1 for $(\alpha, \beta) = (2, 4)$. For the set C of dashed edges we obtain $\text{load}_{C,\alpha}(T_{v,2}^+, v) = \text{load}_{C,\alpha}(v, w) = 7/2 - 2 = 3/2$ and $\text{load}_{C,\alpha}(T_{v,3}, v) = \text{load}_{C,2}(v, w') = 9/2 - 2 = 5/2$. We therefore have $\text{load}_{C,\alpha}(T_{v,2}^+, v) + \text{load}_{C,\alpha}(T_{v,3}, v) = 4 \leq \beta$, so by Lemma 4.1 the set C is a feasible (α, β) -contraction in $T_{v,3}^+$.

Using these definitions it follows straightforwardly from (9a) and (9b) that for any set of edges $C \subseteq E(T_{u_i})$ we have

$$\text{load}_{C \cup \{v, u_i\}, \alpha}(T_{v,i}, v) = \text{load}_{C,\alpha}(T_{u_i}, u_i) + \ell(v, u_i)/\alpha, \quad (10a)$$

$$\text{load}_{C,\alpha}(T_{v,i}, v) = \max\{\text{load}_{C,\alpha}(T_{u_i}, u_i) - (1 - 1/\alpha)\ell(v, u_i), 0\}. \quad (10b)$$

Note that the load increases if the edge $\{v, u_i\}$ is added (see (10a)), and it decreases otherwise (see (10b)). Moreover, for any set of edges $C \subseteq T_{v,i}^+$ and any $i = 1, 2, \dots, c(v)$ we obtain from those definitions that

$$\text{load}_{C,\alpha}(T_{v,i}^+, v) = \max\{\text{load}_{C,\alpha}(T_{v,i-1}^+, v), \text{load}_{C,\alpha}(T_{v,i}, v)\}. \quad (11)$$

These rules allow us to compute the load of all subtrees of T in a bottom-up fashion. Our dynamic program maintains the minimum load of all subtrees of T in three-dimensional matrices L and L^+ . We begin defining these matrices in an abstract way, and then establish several recursive relations which directly translate into a dynamic program. Specifically, for $v \in V$, $i \in \{0, 1, \dots, c(v)\}$ and $s \in \{0, 1, \dots, m\}$ (recall that $m = |E|$) we define

$$L(v, i, s) := \min\{\text{load}_{C,\alpha}(T_{v,i}, v) : C \text{ is a feasible solution of } (T_{v,i}, \ell, \varphi) \text{ of size } |C| = s\}. \quad (12)$$

If there is no feasible solution of the required size, we have $L(v, i, s) = \infty$. The entries of $L^+(v, i, s)$ are defined analogously to (12) by considering the load of $T_{v,i}^+$ instead of $T_{v,i}$. In words, the entries $L(v, i, s)$ and $L^+(v, i, s)$ describe feasible solutions C of size s of the instances $(T_{v,i}, \ell, \varphi)$ or $(T_{v,i}^+, \ell, \varphi)$, respectively, of the problem CONTRACTION for which the load at the vertex v is as small as possible (the matrices contain the minimum achievable load, not the corresponding set of edges).

Lemma 4.2. *Let v be a vertex of T and let $u_1, u_2, \dots, u_{c(v)}$ be the children of v . Then the matrices L and L^+ defined in and directly after (12) satisfy the relations*

$$L(v, i, 0) = L^+(v, i, 0) = 0 \quad \text{for all } i \in \{0, 1, \dots, c(v)\}, \quad (13a)$$

$$L(v, 0, s) = L^+(v, 0, s) = \infty \quad \text{for all } s \in \{1, 2, \dots, m\}, \quad (13b)$$

$$L(v, i, s) = \begin{cases} \mu & \text{if } \mu \leq \beta, \\ \infty & \text{otherwise,} \end{cases} \quad (13c)$$

where $\mu := \min \{L^+(u_i, c(u_i), s-1) + \ell(v, u_i)/\alpha, \max\{L^+(u_i, c(u_i), s) - (1-1/\alpha)\ell(v, u_i), 0\}\}$ for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$.

Moreover, we have

$$L^+(v, i, s) = \min \{ \max\{L^+(v, i-1, t), L(v, i, s-t)\} : t \in \{0, 1, \dots, s\} \text{ and } L^+(v, i-1, t) + L(v, i, s-t) \leq \beta \} \quad (13d)$$

for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$.

The most interesting of these recursive relations are of course (13c) and (13d). The relation (13c) captures the two possibilities of either adding the edge $\{v, u_i\}$ or not adding it to a partial solution in the tree $T_{u_i, c(u_i)}^+ = T_{u_i}$ to obtain a solution for the tree $T_{v, i}$ (recall (10)). The relation (13d), on the other hand, describes how to distribute s contraction edges in $T_{v, i}^+$ among the two subtrees $T_{v, i-1}^+$ and $T_{v, i}$ (t is the number of edges contracted in the first tree, and $s-t$ the number of edges in the second tree, respectively).

Proof. The relations (13a) and (13b) follow immediately from the definitions of the trees $T_{v, i}$ and $T_{v, i}^+$ and from (12). The relation (13c) follows from (10) and (12). The relation (13d) follows from (11) and (12) with the help of Lemma 4.1. ■

We are now ready to prove Theorem 5.

Proof of Theorem 5. Given the instance (T, ℓ, φ) , we fix an arbitrary root r of T and an arbitrary ordering of the children of each vertex, making T an ordered rooted tree. We then compute the entries of the matrices L and L^+ using Lemma 4.2. We first initialize various entries using (13a) and (13b), and compute the remaining entries in a bottom-up fashion moving upwards from the leaves to the root. Specifically, at a vertex v with children $u_1, u_2, \dots, u_{c(v)}$ for which all the entries of L and L^+ have already been computed, we first compute $L(v, i, s)$ for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$ using (13c), and then $L^+(v, i, s)$ for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$ using (13d).

Let s^* be the largest s such that $L^+(r, c(r), s) \leq \beta$. From (12) we obtain that s^* is the size of an optimal solution of the instance (T, ℓ, φ) . The corresponding set of edges $C \subseteq E$ can be obtained by keeping track of the arguments for which the minima and maxima in (13c) and (13d) are attained in each step.

Clearly, L and L^+ both have $\mathcal{O}(n^2)$ entries, and computing each entry takes time $\mathcal{O}(n)$, so the running time of our dynamic program is $\mathcal{O}(n^3)$. ■

4.2. Weak Contraction on trees. In this section we consider the problem of computing weak contractions for a tree T with affine tolerance function $\varphi(x) = x/\alpha - \beta$. Here, our main result is a dynamic programming algorithm that builds on the algorithmic ideas presented in Section 4.1.

Theorem 6. *Let T be a tree with edge lengths $\ell: E \rightarrow \mathbb{R}_{>0}$ and consider the tolerance function $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$, $\beta \geq 0$. An optimal solution for the instance (T, ℓ, φ) of the problem WEAK CONTRACTION can be computed by dynamic programming in time $\mathcal{O}(n^5)$.*

In this setting we need to specifically keep track of pairs of vertices whose distance remains positive when contracting a set of edges $C \subseteq E$ (i.e., not all edges in between these vertices are contracted). To this end we extend the definitions (9) as follows: For any vertex v of T we define the *weak load of T at v* as

$$\text{wload}_{C,\alpha}(T, v) := \max\{\text{load}_{C,\alpha}(u, v) : u \in V \text{ and } \text{dist}_{\ell_C}(u, v) > 0\}. \quad (14)$$

Note that in the maximization we have to consider all vertices u such that at least one edge on the path from u to v is not in C . This definition together with (9b) yields $\text{wload}_{C,\alpha}(T, v) \leq \text{load}_{C,\alpha}(T, v)$. In contrast to the load, the weak load may be negative. In particular, $\text{wload}_{C,\alpha}(T, v) = -\infty$ if and only if $C = E$.

The following lemma is the counterpart to Lemma 4.1 for weak contractions. It describes how to combine feasible solutions on subtrees to a feasible solution of the entire tree. There is one important subtlety here: While the notion of a weak contraction forbids contracting all edges of T , we clearly have to allow this for partial solutions on subtrees of T (as long as some other edge not in the subtree is not contracted, this might still yield a feasible solution).

Lemma 4.3. *Consider a partition of T into two subtrees T_1 and T_2 that only have a vertex $v \in V$ in common. Then $C \subsetneq E$ is a feasible solution for the instance (T, ℓ, φ) of the problem WEAK CONTRACTION if and only if the following two conditions hold: For $i = 1, 2$, either C contains every edge of T_i or $C \cap E(T_i)$ is a feasible solution for the instance (T_i, ℓ, φ) of WEAK CONTRACTION; and we have*

$$\text{load}_{C,\alpha}(T_1, v) + \text{wload}_{C,\alpha}(T_2, v) \leq \beta \quad \text{and} \quad \text{wload}_{C,\alpha}(T_1, v) + \text{load}_{C,\alpha}(T_2, v) \leq \beta. \quad (15)$$

Proof. Let $C \subsetneq E$. For the rest of the proof we omit the subscripts C and α and simply write $\text{load}_{C,\alpha} = \text{load}$ and $\text{wload}_{C,\alpha} = \text{wload}$.

We first assume that C is a feasible solution for the instance (T, ℓ, φ) of the problem WEAK CONTRACTION. I.e., any two vertices u, w of T with $\text{dist}_{\ell_C}(u, w) > 0$ satisfy the condition $\text{load}(u, w) \leq \beta$. This is true in particular for all pairs of vertices $u, w \in T_i$, $i = 1, 2$, implying that either $C \supseteq T_i$ or $C \cap T_i \subsetneq T_i$ is a feasible solution for the instance (T_i, ℓ, φ) . If $\text{wload}(T_2, v) = -\infty$, the claimed inequality $\text{load}(T_1, v) + \text{wload}(T_2, v) \leq \beta$ is trivially satisfied. So suppose that $\text{wload}(T_2, v)$ is a finite number, and let $u \in T_1$ and $w \in T_2$ be such that $\text{load}(u, v) = \text{load}(T_1, v)$, and $\text{dist}_{\ell_C}(v, w) > 0$ as well as $\text{load}(v, w) = \text{wload}(T_2, v)$. Then we also have $\text{dist}_{\ell_C}(u, w) > 0$, so we know that $\text{load}(u, w) \leq \beta$ by the assumption that C is feasible for (T, ℓ, φ) . Combining this last inequality with the relation $\text{load}(u, w) = \text{load}(u, v) + \text{load}(v, w) = \text{load}(T_1, v) + \text{wload}(T_2, v)$ proves that the right hand side of the equation is at most β , as claimed. The proof of the second inequality $\text{wload}(T_1, v) + \text{load}(T_2, v) \leq \beta$ works symmetrically. This proves one direction of the equivalence.

To prove the reverse direction, we now assume that either $C \supseteq T_i$ or $C \cap T_i \subsetneq T_i$ is a feasible solution for the instance (T_i, ℓ, φ) for $i = 1, 2$, and that $\text{load}(T_1, v) + \text{wload}(T_2, v) \leq \beta$ and $\text{wload}(T_1, v) + \text{load}(T_2, v) \leq \beta$. To show that C is a feasible solution for the instance (T, ℓ, φ) , let $u \in T_1$ and $w \in T_2$ be such that $\text{dist}_{\ell_C}(u, w) > 0$. It follows that $\text{dist}_{\ell_C}(u, v) > 0$ or $\text{dist}_{\ell_C}(v, w) > 0$. We first consider the case that $\text{dist}_{\ell_C}(u, v) > 0$. By the definitions (9) and (14) we have $\text{load}(u, v) \leq \text{wload}(T_1, v)$, and also $\text{load}(v, w) \leq \text{load}(T_2, v)$, yielding $\text{load}(u, w) = \text{load}(u, v) + \text{load}(v, w) \leq \text{wload}(T_1, v) + \text{load}(T_2, v) \leq \beta$ (the last inequality holds by assumption). This proves that $\text{load}(u, w) \leq \beta$, as desired. The proof of the other case $\text{dist}_{\ell_C}(v, w) > 0$ works symmetrically. This completes the proof of the lemma. ■

As in Section 4.1, we view T as an ordered rooted tree, and consider its subtrees T_v , $T_{v,i}$ and $T_{v,i}^+$ for all $v \in V$ and $i \in \{0, 1, \dots, c(v)\}$ (recall the definitions given after Lemma 4.1). Let us briefly highlight the differences between Lemmas 4.1 and 4.3. The

dynamic programming algorithm presented in Section 4.1 exploits the fact that the optimal way to contract exactly $|C| = s$ edges in a subtree T_v of T rooted at a particular vertex v is to contract a set of edges that minimizes $\text{load}_{C,\alpha}(T_v, v)$. This is possible as the optimality condition in Lemma 4.1 only depends on this parameter. Here the situation is more complicated, as Lemma 4.3 also considers $\text{wload}_{C,\alpha}(T_v, v)$. Figure 5 illustrates that it is not sufficient to minimize only one of these parameters.

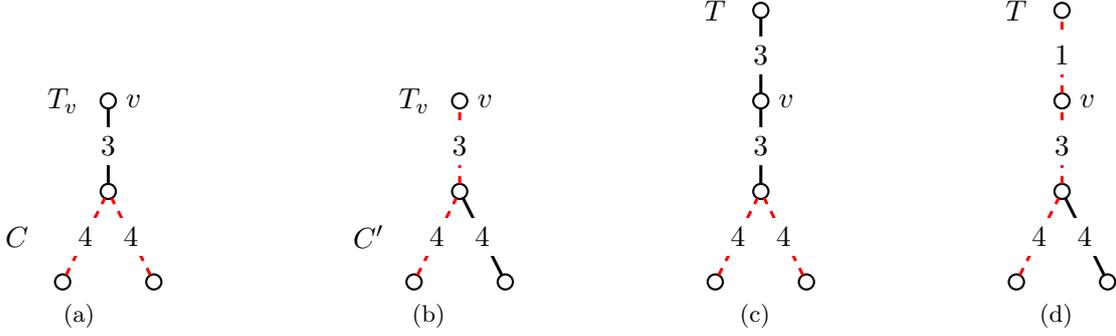


FIGURE 5. Example of the behavior of the parameters $\text{load}()$ and $\text{wload}()$ for $(\alpha, \beta) = (2, 1/2)$. Consider the tree T in (c) and (d) with length functions that differ only in the value they assign to the topmost edge. Parts (a) and (b) of the figure show the subtree T_v of T and two different subsets of edges C and C' of T_v , respectively, drawn by dashed edges. We have $\text{load}_{C,\alpha}(T_v, v) = \text{wload}_{C,\alpha}(T_v, v) = 7/2 - 3 = 1/2$, while $\text{load}_{C',\alpha}(T_v, v) = 7/2$ and $\text{wload}_{C',\alpha}(T_v, v) = 7/2 - 4 = -1/2$, so no solution dominates the other. For the length function in (c), C is feasible and optimal, but C' is not feasible. For the length function in (d), on the other hand, C' is feasible and can be extended to an optimal solution of size 3, but C can not be extended.

Consequently, we keep track of an entire Pareto front of non-dominated partial solutions (see Figure 6). Formally, we define the *set* $F(T_v, s)$ of *feasible partial solutions of size* s as the family of all sets $C \subseteq E(T_v)$ with $|C| = s$ such that either $C = E(T_v)$ or C is a feasible solution for the instance (T_v, ℓ, φ) of WEAK CONTRACTION. For two sets $C, C' \in F(T_v, s)$ we say that C *dominates* C' at v if $\text{load}_{C,\alpha}(T_v, v) \leq \text{load}_{C',\alpha}(T_v, v)$ and $\text{wload}_{C,\alpha}(T_v, v) \leq \text{wload}_{C',\alpha}(T_v, v)$, and we define the *Pareto front* $P(T_v, v, s)$ as a minimal family of sets $C \in F(T_v, s)$ such that no set $C' \in F(T_v, s)$ dominates C at v . Note that the domination relation is reflexive, so there may be several different such minimal families, all with the same pairs of load and weak load values, and any choice among them is equally good for us. This definition is illustrated in Figure 6.

The following crucial lemma asserts that the number of points on the Pareto front, i.e., the size of the family $P(T_v, v, s)$ is at most $n + 1$. This property is essential for our dynamic programming approach, and it does not follow immediately from the definition of $P(T_v, v, s)$, as the set of feasible solutions $F(T_v, s)$ is typically of exponential size.

Lemma 4.4. *For any $C \subseteq E(T_v)$, we have $\text{load}_{C,\alpha}(T_v, v) \in \Lambda(T_v, v) := \{\text{dist}_\ell(u, v)/\alpha : u \in V(T_v)\}$ or $\text{load}_{C,\alpha}(T_v, v) = \text{wload}_{C,\alpha}(T_v, v)$. Consequently, the Pareto front $P(T_v, v, s)$ has size at most $n + 1$.*

Proof. By the definitions (9) and (14) we have $\text{wload}_{C,\alpha}(T_v, v) \leq \text{load}_{C,\alpha}(T_v, v)$ for all $C \subseteq E(T_v)$. Now let $C \subseteq E(T_v)$ be such that $\text{wload}_{C,\alpha}(T_v, v) < \text{load}_{C,\alpha}(T_v, v)$. Again

by the previously mentioned definitions this implies that $\text{load}_{C,\alpha}(T_v, v) = \text{load}_{C,\alpha}(u, v) = \text{dist}_\ell(u, v)/\alpha$ for some $u \in V(T_v)$, which is indeed an element of the set $\Lambda(T_v, v)$. Consequently, the Pareto front $P(T_v, v, s)$ consists of at most one set $C \in F(T_v, s)$ with $\text{wload}_{C,\alpha}(T_v, v) = \text{load}_{C,\alpha}(T_v, v)$ and at most one set $C \in F(T_v, s)$ with $\text{wload}_{C,\alpha}(T_v, v) < \text{load}_{C,\alpha}(T_v, v)$ for each number in $\Lambda(T_v, v)$. Using that $|\Lambda(T_v, v)| \leq |V(T_v)| \leq n$ it follows that $|P(T_v, v, s)| \leq n + 1$. \blacksquare

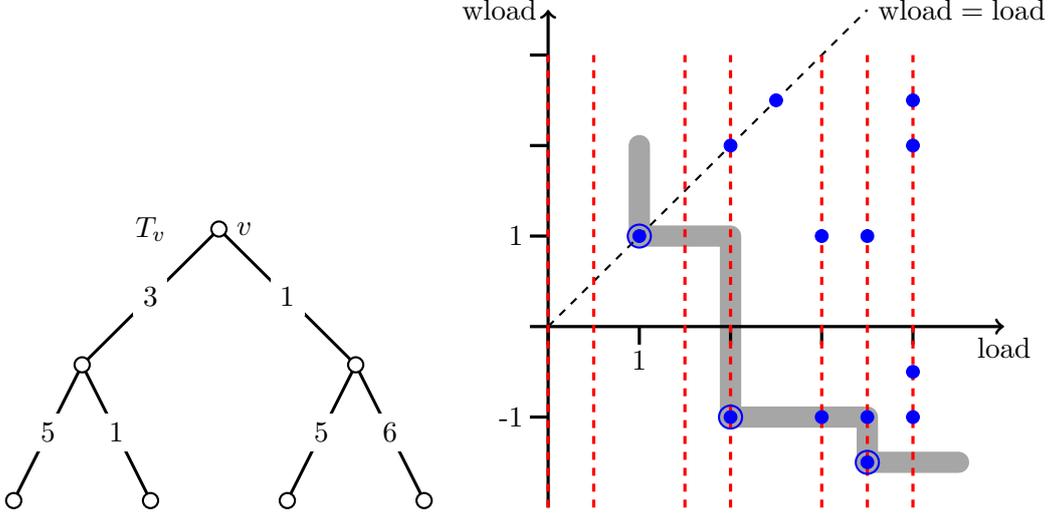


FIGURE 6. The right hand side of the figure shows the set of all pairs of (load, wload) values for all feasible solutions of size $s = 3$ for the rooted tree (T_v, v) shown on the left for the parameter values $(\alpha, \beta) = (2, 13/2)$. The points in $P(T_v, v, s)$ are highlighted with small circles. The vertical dashed lines represent all values in the set $\Lambda(T_v, v) = \{0, 1/2, 3/2, 2, 3, 7/2, 4\}$. Moreover, we have $\lambda^*(T_v, v, s) = 1$.

By Lemma 4.4 the load values of all points on the Pareto front with $\text{wload}() < \text{load}()$ are in the set $\Lambda(T_v, v)$. There might also be one point with $\text{wload}() = \text{load}()$ on the Pareto front (as in the example shown in Figure 6), and this load value might not be an element of $\Lambda(T_v, v)$. We extend the set $\Lambda(T_v, v)$ accordingly by defining for $s \in \{0, 1, \dots, m\}$ (recall that $m = |E|$)

$$\lambda^*(T_v, v, s) := \min\{\text{load}_{C,\alpha}(T_v, v) : C \in F(T_v, s)\}, \quad (16a)$$

$$\Lambda^*(T_v, v, s) := \Lambda(T_v, v) \cup \{\lambda^*(T_v, v, s)\}. \quad (16b)$$

If the set $F(T_v, s)$ is empty, we have $\lambda^*(T_v, v, s) = \infty$.

We now describe recursive relations for the weak load that are analogous to (10) and (11) for the load. It follows straightforwardly from (9) and (14) that for any vertex v of T and its children u_i , $i = 1, 2, \dots, c(v)$, and for any set of edges $C \subseteq E(T_{u_i})$ we have

$$\text{wload}_{C \cup \{v, u_i\}, \alpha}(T_{v,i}, v) = \text{wload}_{C, \alpha}(T_{u_i}, u_i) + \ell(v, u_i)/\alpha, \quad (17a)$$

$$\text{wload}_{C, \alpha}(T_{v,i}, v) = \text{load}_{C, \alpha}(T_{u_i}, u_i) - (1 - 1/\alpha)\ell(v, u_i). \quad (17b)$$

Note that the weak load increases if the edge $\{v, u_i\}$ is added (see (17a)). On the other hand, if the edge $\{v, u_i\}$ is not added, it may decrease or increase (the right hand side of (17b) refers to the load, *not* to the weak load). Moreover, for any set of edges $C \subseteq T_{v,i}^+$ and any $i = 1, 2, \dots, c(v)$ the definition (14) readily implies

$$\text{wload}_{C, \alpha}(T_{v,i}^+, v) = \max\{\text{wload}_{C, \alpha}(T_{v,i-1}^+, v), \text{wload}_{C, \alpha}(T_{v,i}, v)\}. \quad (18)$$

These rules together with the corresponding relations (10) and (11) allow us to compute the weak load and the load of all Pareto optimal partial solutions in a bottom-up fashion, similar to the approach taken in Section 4.1. Before it was sufficient to compute one optimal partial solution for every subtree $T_{v,i}$ and $T_{v,i}^+$, $i \in \{1, 2, \dots, c(v)\}$, and every possible size s of the contracted set of edges, but now our dynamic program keeps track of the entire Pareto fronts $P(T_{v,i}, v, s)$ and $P(T_{v,i}^+, v, s)$. We store the corresponding pairs of load and weak load values on the Pareto front in separate four-dimensional matrices W , W^+ , L and L^+ (the entries of W and W^+ are certain weak load values, and the entries of L and L^+ are the corresponding load values). We begin defining these matrices in an abstract way, and then establish several recursive relations which directly translate into a dynamic programming algorithm. Specifically, for $v \in V$, $i \in \{0, 1, \dots, c(v)\}$, $s \in \{0, 1, \dots, m\}$ and $\lambda \in \Lambda(T_{v,i}, v)$ with $\Lambda(T_{v,i}, v)$ as in Lemma 4.4 we define

$$W(v, i, s, \lambda) := \min\{\text{wload}_{C,\alpha}(T_{v,i}, v) : C \in F(T_{v,i}, s) \text{ with } \text{load}_{C,\alpha}(T_{v,i}, v) \leq \lambda\}, \quad (19a)$$

$$L(v, i, s, \lambda) := \min\{\text{load}_{C,\alpha}(T_{v,i}, v) : C \in F(T_{v,i}, s), \text{wload}_{C,\alpha}(T_{v,i}, v) = W(v, i, s, \lambda)\}. \quad (19b)$$

If there is no set C satisfying these requirements, we have $W(v, i, s, \lambda) = L(v, i, s, \lambda) = \infty$. The entries of $W^+(v, i, s, \lambda)$ and $L^+(v, i, s, \lambda)$ are defined analogously to (19) by considering the tree $T_{v,i}^+$ instead of $T_{v,i}$ (in particular, in this case we have $\lambda \in \Lambda(T_{v,i}^+, v)$).

The definitions of $W(v, i, s, \lambda)$ and $L(v, i, s, \lambda)$ given in (19) extend straightforwardly to the value $\lambda = \lambda^*(T_{v,i}, v, s)$ defined in (16a). Similarly, the definitions of $W^+(v, i, s, \lambda)$ and $L^+(v, i, s, \lambda)$ from before extend to the value $\lambda = \lambda^*(T_{v,i}^+, v, s)$. It is easy to see that we have in fact

$$L(v, i, s, \lambda^*(T_{v,i}, v, s)) = \lambda^*(T_{v,i}, v, s) \quad (20)$$

(an analogous relation holds for the entries of L^+).

The recursive relations satisfied by the matrices W , L , W^+ and L^+ defined before are captured by the following two lemmas. The initialization steps and the recursive computation of W and L are treated in Lemma 4.5. The recursive computation of W^+ and L^+ is somewhat more technical, and is treated separately in Lemma 4.6.

Lemma 4.5. *Let v be a vertex of T and let $u_1, u_2, \dots, u_{c(v)}$ be the children of v . Then the matrices W , W^+ , L and L^+ defined in and directly after (19) satisfy the relations*

$$W(v, 0, 0, 0) = W^+(v, 0, 0, 0) = -\infty, \quad (21a)$$

$$W(v, i, 0, \lambda) = -(1 - 1/\alpha)\ell(v, u_i) \quad \text{for all } i \in \{1, 2, \dots, c(v)\} \text{ and } \lambda \in \Lambda^*(T_{v,i}, v, 0), \quad (21b)$$

$$L(v, i, 0, \lambda) = 0 \quad \text{for all } i \in \{0, 1, \dots, c(v)\} \text{ and } \lambda \in \Lambda^*(T_{v,i}, v, 0), \quad (21c)$$

$$L^+(v, i, 0, \lambda) = 0 \quad \text{for all } i \in \{0, 1, \dots, c(v)\} \text{ and } \lambda \in \Lambda^*(T_{v,i}^+, v, 0), \quad (21d)$$

$$W(v, 0, s, \lambda) = L(v, 0, s, \lambda) = \infty \quad \text{for all } s \in \{1, 2, \dots, m\} \text{ and } \lambda \in \Lambda^*(T_{v,0}, v, s), \quad (21e)$$

$$W^+(v, 0, s, \lambda) = L^+(v, 0, s, \lambda) = \infty \quad \text{for all } s \in \{1, 2, \dots, m\} \text{ and } \lambda \in \Lambda^*(T_{v,0}^+, v, s). \quad (21f)$$

Furthermore, we have

$$W(v, i, s, \lambda) = \begin{cases} \mu & \text{if } \lambda = 0, \\ \min\{\mu, \nu\} & \text{if } \lambda > 0 \text{ and } \mu \leq \min\{\beta, \lambda\}, \\ \nu & \text{if } \lambda > 0 \text{ and } \mu > \min\{\beta, \lambda\} \text{ and } \nu \leq \beta, \\ \infty & \text{otherwise,} \end{cases} \quad (21g)$$

$$L(v, i, s, \lambda) = \begin{cases} \max\{\mu, 0\} & \text{if } W(v, i, s, \lambda) = \mu, \\ L^+(u_i, c(u_i), s-1, \lambda - \ell(v, u_i)/\alpha) + \ell(v, u_i)/\alpha & \text{otherwise,} \end{cases} \quad (21h)$$

with $\mu := \lambda^*(T_{u_i}, u_i, s) - (1 - 1/\alpha)\ell(v, u_i)$ and $\nu := W^+(u_i, c(u_i), s-1, \lambda - \ell(v, u_i)/\alpha) + \ell(v, u_i)/\alpha$ for all $i \in \{1, 2, \dots, c(v)\}$, $s \in \{1, 2, \dots, m\}$ and $\lambda \in \Lambda(T_{v,i}, v)$.

Finally, we have

$$L(v, i, s, \lambda^*) = \lambda^* = \begin{cases} \min\{\lambda + \ell(v, u_i)/\alpha, \max\{0, \mu\}\} & \text{if } \mu \leq \beta, \\ \lambda + \ell(v, u_i)/\alpha & \text{otherwise,} \end{cases} \quad (21i)$$

$$W(v, i, s, \lambda^*) = \begin{cases} \rho & \text{if } L(v, i, s, \lambda^*) = \lambda + \ell(v, u_i)/\alpha, \\ \mu & \text{otherwise,} \end{cases} \quad (21j)$$

where $\lambda \in \Lambda^*(T_{u_i}, u_i, s-1)$ is minimal such that $\rho := W^+(u_i, c(u_i), s-1, \lambda) + \ell(v, u_i)/\alpha \leq \beta$, if such a value λ exists, and $\lambda := \rho := \infty$ otherwise, for all $i \in \{1, 2, \dots, c(v)\}$, $s \in \{1, 2, \dots, m\}$ and $\lambda^* = \lambda^*(T_{v,i}, v, s)$.

Note that the relations (21a)–(21f) are the initialization steps, and the relations (21g)–(21j) capture the two possibilities of either adding or not adding the edge $\{v, u_i\}$ to a partial solution in the tree $T_{u_i, c(u_i)}^+ = T_{u_i}$ to obtain a solution for the tree $T_{v,i}$ (recall (10) and (17)).

We only refer to well-defined entries of W^+ and L^+ in (21h) and in the definition of ν , as $\lambda - \ell(v, u_i)/\alpha \in \Lambda(T_{u_i}, u_i)$ holds for every $\lambda \in \Lambda(T_{v,i}, v) \setminus \{0\}$. Note that we either have $\nu \leq \lambda$ or $\nu = \infty$, while μ may also take a value in the open interval (λ, ∞) .

Proof. The relations (21a)–(21f) follow immediately from the definitions of the trees $T_{v,i}$ and $T_{v,i}^+$ and the definitions of the respective matrices given in (19) and afterwards. The relations (21g) and (21i) follow from (17) and the definitions of W and L , respectively: Consider a partial solution $C \in F(T_{v,i}, s)$. If $\text{load}_{C, \alpha}(T_{v,i}, v) = 0$, then C does not contain the edge $\{v, u_i\}$, so we have $W(v, i, s, 0) = \mu$. The other cases of (21g) as well as (21i) are implied by the following observation: If $\text{wload}_{C, \alpha}(T_{v,i}, v) \leq \lambda$ and $\{v, u_i\} \notin C$, then by (17) we have $\mu \leq \beta$ and $\mu \leq \lambda$.

The relation (21h) is closely related to (21g). If $\mu \neq \nu$, then (21h) follows immediately from (21g) and the definitions of W and L . If $\mu = \nu \leq \min\{\beta, \lambda\}$, then both a partial solution containing the edge $\{v, u_i\}$ as well as one missing this edge minimize the weak load. As the weak load is bounded from above by the load, we get $L(v, i, s, \lambda) = W(v, i, s, \lambda) = \mu$ in this case. This implies (21h). An analogous argument yields (21j). ■

The following lemma describes the recursive relations satisfied by the entries of W^+ and L^+ . Specifically, the lemma describes how to distribute s contraction edges in $T_{v,i}^+$ among the two subtrees $T_{v,i-1}^+$ and $T_{v,i}$ (t is the number of edges contracted in the first tree, and $s-t$ the number of edges in the second tree, respectively). To compute a single point on the Pareto front $P(T_{v,i}^+, v, s)$, we need to consider all points on the Pareto fronts $P(T_{v,i-1}^+, v, t)$ and $P(T_{v,i}, v, s-t)$.

Lemma 4.6. *Let v be a vertex of T , and let $s \in \{0, 1, \dots, m\}$ and $i \in \{1, 2, \dots, c(v)\}$ be fixed throughout this lemma. For $t \in \{0, 1, \dots, s\}$ we let $\Pi(t)$ denote the set of all pairs (λ_1, λ_2) with $\lambda_1 \in \Lambda^*(T_{v,i-1}^+, v, t)$ and $\lambda_2 \in \Lambda^*(T_{v,i}, v, s-t)$ such that $W^+(v, i-1, t, \lambda_1) + L(v, i, s-t, \lambda_2) \leq \beta$ and $L^+(v, i-1, t, \lambda_1) + W(v, i, s-t, \lambda_2) \leq \beta$. For $t \in \{0, 1, \dots, s\}$ and $\lambda \in \Lambda(T_{v,i}^+, v)$ we let $\Pi(t, \lambda) \subseteq \Pi(t)$ denote the set of all pairs $(\lambda_1, \lambda_2) \in \Pi(t)$ satisfying $\max\{L^+(v, i-1, t, \lambda_1), L(v, i, s-t, \lambda_2)\} \leq \lambda$. For all $\lambda \in \Lambda(T_{v,i}^+, v)$, defining*

$$W(t) := \min \{ \max\{W^+(v, i-1, t, \lambda_1), W(v, i, s-t, \lambda_2)\} : (\lambda_1, \lambda_2) \in \Pi(t, \lambda) \}, \quad (22a)$$

$$L(t) := \min \{ \max\{L^+(v, i-1, t, \lambda_1), L(v, i, s-t, \lambda_2)\} : (\lambda_1, \lambda_2) \text{ minimizes (22a)} \}, \quad (22b)$$

we have

$$W^+(v, i, s, \lambda) = \min\{W(t) : t \in \{0, 1, \dots, s\}\}, \quad (22c)$$

$$L^+(v, i, s, \lambda) = \min\{L(t) : t \text{ minimizes (22c)}\}. \quad (22d)$$

For $\lambda^* = \lambda^*(T_{v,i}^+, v, s)$, defining

$$L^*(t) := \min \{ \max\{L^+(v, i-1, t, \lambda_1), L(v, i, s-t, \lambda_2)\} : (\lambda_1, \lambda_2) \in \Pi(t) \}, \quad (22e)$$

$$W^*(t) := \min \{ \max\{W^+(v, i-1, t, \lambda_1), W(v, i, s-t, \lambda_2)\} : (\lambda_1, \lambda_2) \text{ minimizes (22e)} \}, \quad (22f)$$

we have

$$L^+(v, i, s, \lambda^*) = \lambda^* = \min\{L^*(t) : t \in \{0, 1, \dots, s\}\}, \quad (22g)$$

$$W^+(v, i, s, \lambda^*) = \min\{W^*(t) : t \text{ minimizes (22g)}\}. \quad (22h)$$

Proof. The relation (22c) follows by combining the definitions (19a) and (22a) with the relations (11), (18) and the condition (15) from Lemma 4.3. The argument for (22d) is analogous, using the definitions (19b) and (22b) instead of (19a) and (22a).

The relation (22g) follows by combining the definitions (16a) and (22e) (recall also (20)) with the relations (11), (18) and the condition (15) from Lemma 4.3. The argument for (22h) is analogous, using the definitions (19a) and (22f) instead of (16a) and (22e). \blacksquare

We can trivially compute the quantities $W(t)$, $L(t)$, $W^*(t)$ and $L^*(t)$ as defined in Lemma 4.6 in time $\mathcal{O}(n^2)$ (using that $|\Pi(t)| = \mathcal{O}(n^2)$ and $|\Pi(t, \lambda)| = \mathcal{O}(n^2)$ by Lemma 4.4). The following lemma shows how to do the same computation in time $\mathcal{O}(n)$, so that the entries $W^+(v, i, s, \lambda)$ and $L^+(v, i, s, \lambda)$ can be computed via (22c), (22d), (22g) and (22h) in time $\mathcal{O}(n^2)$ (instead of the trivial bound $\mathcal{O}(n^3)$).

Lemma 4.7. *If the numbers in the sets $\Lambda^*(T_{v,i-1}^+, v, t)$ and $\Lambda^*(T_{v,i}, v, s-t)$ are sorted increasingly, the quantities $W(t)$, $L(t)$, $W^*(t)$ and $L^*(t)$ defined in Lemma 4.6 can be computed in time $\mathcal{O}(n)$. Consequently, $W^+(v, i, s, \lambda)$ and $L^+(v, i, s, \lambda)$ can be computed for all $s \in \{0, 1, \dots, m\}$ and all $\lambda \in \Lambda^*(T_{v,i}^+, v, s)$ in time $\mathcal{O}(n^2)$.*

Proof. We define the sequence P_1 of all pairs of finite numbers $(L^+(v, i-1, t, \lambda), W^+(v, i-1, t, \lambda))$ for all $\lambda \in \Lambda^*(T_{v,i-1}^+, v, t)$ in increasing order of λ -values. Similarly, we define the sequence P_2 of all pairs of finite numbers $(L(v, i, s-t, \lambda), W(v, i, s-t, \lambda))$ for all $\lambda \in \Lambda^*(T_{v,i}, v, s-t)$ in increasing order of λ -values. By Lemma 4.4 each of these lists has size $\mathcal{O}(n)$. Note that these sequences correspond to the Pareto fronts $P(T_{v,i-1}^+, v, t)$ and $P(T_{v,i}, v, s-t)$, respectively. Some pairs of points may appear multiple times consecutively in P_1 and P_2 , and in a preprocessing step we eliminate these duplicates in time $\mathcal{O}(n)$. We

know that after this preprocessing step, the first entries in the simplified lists P_1 and P_2 are strictly increasing, and the second entries are strictly decreasing (recall Figure 6).

We first argue how to compute $W(t)$ and $L(t)$. We begin discarding all pairs from each list whose first entry (L^+ or L , respectively) is strictly greater than λ in time $\mathcal{O}(n)$. We then process the remaining lists P_1 and P_2 beginning at the last entries (L_j^+, W_j^+) and (L_k, W_k) (with smallest W^+ or W -values, respectively) in two phases.

In the first phase we compute $W(t)$ as follows: If $L_j^+ + W_k > \beta$, we discard the last element of P_1 by decreasing j by 1 (by our sorting of the lists we know that $L_j^+ + W_{k'} > \beta$ for all $k' \leq k$). If $W_j^+ + L_k > \beta$, we discard the last element of P_2 by decreasing k by 1 (by our sorting of the lists we know that $W_{j'}^+ + L_k > \beta$ for all $j' \leq j$). Once $L_j^+ + W_k \leq \beta$ and $W_j^+ + L_k \leq \beta$ for the first time, we have found $W(t) = \max\{W_j^+, W_k\}$. If this never happens we know that $W(t) = \infty$. This computation is correct by the definition of $\Pi(t, \lambda)$ in Lemma 4.6 and by (22a), and it takes time $\mathcal{O}(n)$.

In the second phase we compute $L(t)$ as follows: If $W(t) = \infty$, we know that $L(t) = \infty$, too. Otherwise we distinguish two cases: If $W_j^+ \geq W_k$, we decrease k further as long as both inequalities $W_j^+ \geq W_k$ and $L_j^+ + W_k \leq \beta$ are still satisfied (so that they still hold for the final k). If $W_j^+ \leq W_k$, we decrease j further as long as both inequalities $W_j^+ \leq W_k$ and $W_j^+ + L_k \leq \beta$ are still satisfied (so that they still hold for the final j). In the end we set $L(t) = \max\{L_j^+, L_k\}$. Note that in the first case, the third constraint $W_j^+ + L_k \leq \beta$ remains valid by the monotonicity $L_{k'} \leq L_k$ for all $k' \leq k$, and in the second case, the third constraint $L_j^+ + W_k \leq \beta$ remains valid by the monotonicity $L_{j'}^+ \leq L_j^+$ for all $j' \leq j$. Therefore, the correctness of the computation of $L(t)$ follows from (22b).

The procedure to compute $W^*(t)$ and $L^*(t)$ processes P_1 and P_2 (as obtained from the preprocessing step explained in the beginning) starting at the first entries (L_j^+, W_j^+) , $j = 1$, and (L_k, W_k) , $k = 1$, in two phases very similarly to before. We omit the details here. ■

We are now ready to prove Theorem 6.

Proof of Theorem 6. Given the instance (T, ℓ, φ) , we fix an arbitrary root r of T and an arbitrary ordering of the children of each vertex, making T an ordered rooted tree.

We begin precomputing and sorting all of the sets $\Lambda(T_{v,i}, v)$ and $\Lambda(T_{v,i}^+, v)$, $v \in V$, $i \in \{0, 1, \dots, c(v)\}$, and we maintain them as sorted lists throughout the algorithm. This takes time $\mathcal{O}(n^2 \log n)$ in total (recall Lemma 4.4).

We then compute the entries of the matrices W , L , W^+ and L^+ using Lemmas 4.5 and 4.7. We first initialize various entries using (21a)–(21f), and compute the remaining entries in a bottom-up fashion moving upwards from the leaves to the root. Specifically, at a vertex v with children $u_1, u_2, \dots, u_{c(v)}$ for which all the entries of W , L , W^+ and L^+ have already been computed, we first compute $W(v, i, s, \lambda)$ and then $L(v, i, s, \lambda)$ for all $i \in \{1, 2, \dots, c(v)\}$, $s \in \{1, 2, \dots, m\}$ and $\lambda \in \Lambda(T_{v,i}, v)$ using (21g) and (21h), then we compute $L(v, i, s, \lambda^*(T_{v,i}, v, s)) = \lambda^*(T_{v,i}, v, s)$ and $W(v, i, s, \lambda^*(T_{v,i}, v, s))$ for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$ using (21i) and (21j). We obtain sorted lists containing the numbers in $\Lambda^*(T_{v,i}, v, s)$ by inserting $\lambda^*(T_{v,i}, v, s)$ at the correct position into the precomputed list $\Lambda(T_{v,i}, v)$. Next, we compute $W^+(v, i, s, \lambda)$ and then $L^+(v, i, s, \lambda)$ for all $i \in \{1, 2, \dots, c(v)\}$, $s \in \{1, 2, \dots, m\}$ and $\lambda \in \Lambda(T_{v,i}^+, v)$ using (22c) and (22d), and then we compute $L^+(v, i, s, \lambda^*(T_{v,i}^+, v, s)) = \lambda^*(T_{v,i}^+, v, s)$ and $W^+(v, i, s, \lambda^*(T_{v,i}^+, v, s))$ for all $i \in \{1, 2, \dots, c(v)\}$ and $s \in \{1, 2, \dots, m\}$ using (22g) and (22h). We obtain sorted lists containing the numbers in $\Lambda^*(T_{v,i}^+, v, s)$ by inserting $\lambda^*(T_{v,i}^+, v, s)$ at the correct position into the precomputed list $\Lambda(T_{v,i}^+, v)$.

Let s^* be the largest s such that $W^+(r, c(r), s, \lambda^*(T, r, s))$ is finite. From (19) we obtain that s^* is the size of an optimal solution of the instance (T, ℓ, φ) . The corresponding set of edges $C \subseteq E$ can be obtained by keeping track of the arguments for which the minima and maxima in (21g)–(21j) and (22a)–(22h) are attained in each step.

Each of the matrices W , L , W^+ and L^+ has $\mathcal{O}(n^3)$ entries (recall Lemma 4.4). Computing an entry of W or L takes $\mathcal{O}(n)$ time by Lemma 4.5, while computing an entry of W^+ or L^+ can be achieved in time $\mathcal{O}(n^2)$ by Lemma 4.7, so the running time of our dynamic program is $\mathcal{O}(n^5)$. ■

5. HARDNESS FOR ADDITIVE TOLERANCE FUNCTIONS

In this section we prove that the problems CONTRACTION and WEAK CONTRACTION for the tolerance function $\varphi(x) = x - \beta$ (purely additive error) are hard already on cycles (Section 5.1 below). We then prove that CONTRACTION with the same tolerance function is hard to approximate for general graphs and for bipartite graphs (Section 5.2).

5.1. Hardness of Contraction and Weak Contraction. Recall that we can compute optimal (weak) (α, β) -contractions in polynomial time on trees (this was shown in Section 4.1), and have a linear time algorithm for CONTRACTION on cycles with unit length edges (this was shown in Section 3.2). We now show that the problem with $\alpha = 1$ is NP-hard on cycles with arbitrary edge lengths.

Theorem 7. *For any fixed $\beta > 0$, the problems CONTRACTION and WEAK CONTRACTION with tolerance function $\varphi(x) = x - \beta$, $\beta \geq 0$, are NP-hard on cycles.*

Theorem 7 (where β is not part of the input) follows immediately from Theorem 8 below (where β is part of the input). The reason is that an instance with $\alpha = 1$ does not change when multiplying all edge lengths and β by some constant.

Theorem 8. *The problems CONTRACTION and WEAK CONTRACTION with tolerance function $\varphi(x) = x - \beta$, $\beta \geq 0$, are NP-hard on cycles.*

The rest of this section is devoted to proving Theorem 8.

For our proof we will use the following variant of the well-known problem PARTITION, referred to as CLOSE-TO-1 PARTITION. To state the problem we say that a set of positive real numbers $\{a_1, a_2, \dots, a_n\}$ is *close to 1*, if $\sum_{i=1}^n a_i = n$ and $\varepsilon := \sum_{i=1}^n |a_i - 1| < 1/5$.

CLOSE-TO-1 PARTITION

Input: A set of positive real numbers $\{a_1, a_2, \dots, a_n\}$ that is close to 1.

Output: ‘Yes’ if there is a subset $I \subseteq [n]$ such that $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i$, ‘No’ otherwise.

Note that for a ‘Yes’-instance of this problem, the solution $I \subseteq [n]$ must have size $n/2$, so $|I| = |[n] \setminus I| = \sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = n/2$. In particular, this implies that n is even.

In the classical problem PARTITION, the input set is not constrained to be close to 1. PARTITION was shown to be NP-complete already in Karp’s seminal paper [Kar72]. The fact that CLOSE-TO-1 PARTITION is also NP-complete follows from a straightforward rescaling argument.

Lemma 5.1. *CLOSE-TO-1 PARTITION is NP-complete.*

Proof. Given an instance $\{a_1, a_2, \dots, a_n\}$ of PARTITION, we first add n additional zeroes $a_{n+1} = a_{n+2} = \dots = a_{2n} = 0$ to the instance (by this we ensure that a partition with equal sums is transformed into one where both partition classes have the same number n of

summands). We then linearly transform all the a_i according to $a'_i := (a_i + C)/D$, where C and D are sufficiently large constants so that the transformed values a'_i are close to 1. The transformed set of numbers has even cardinality $2n$, is close to 1, and it admits a partition into two sets of size n with equal sum if and only if the original instance allows a partition into two sets with equal sum. ■

Proof of Theorem 8. We first focus on the problem CONTRACTION. We reduce CLOSE-TO-1 PARTITION, which is NP-complete by Lemma 5.1, to the problem CONTRACTION on a cycle with tolerance function $\varphi(x) = x - \beta$, $\beta \geq 0$.

Let $\mathcal{I} = \{a_1, a_2, \dots, a_n\}$ be an instance of CLOSE-TO-1 PARTITION such that $a_1 \geq a_2 \geq \dots \geq a_n$. This ensures that all a_i that are bigger than 1 appear before all a_i that are smaller than 1, which is the only property of the ordering that we exploit in the proof later on. The instance of CONTRACTION we construct is on the cycle C_{2n+4} with $2n + 4$ edges. We label the vertices of the cycle by walking around the cycle as follows: The first $n + 1$ vertices are labelled u_0, u_1, \dots, u_n , then there are two special vertices v_1, v_2 , and the remaining $n + 1$ vertices are labelled w_0, w_1, \dots, w_n , see Figure 7. We denote the subpaths of the cycle given by all edges $\{u_i, u_{i+1}\}$ as P_u , and the subpath given by all edges $\{w_i, w_{i+1}\}$ by P_w .

We now define $\varepsilon := \sum_{i=1}^n |1 - a_i| < 1/5$, $\beta := n/2 + 2\varepsilon$ and $\beta' := \beta + 1 > \beta$, and the length function ℓ on the cycle edges by setting $\ell(u_{i-1}, u_i) := a_i$ and $\ell(w_{i-1}, w_i) = 2 - a_i$ for all $i \in [n]$, and by $\ell(u_n, v_1) = \ell(v_2, w_1) := \varepsilon$, $\ell(v_1, v_2) := \beta'$, and $\ell(w_n, u_0) := \beta' + 2\varepsilon$ (see Figure 7).

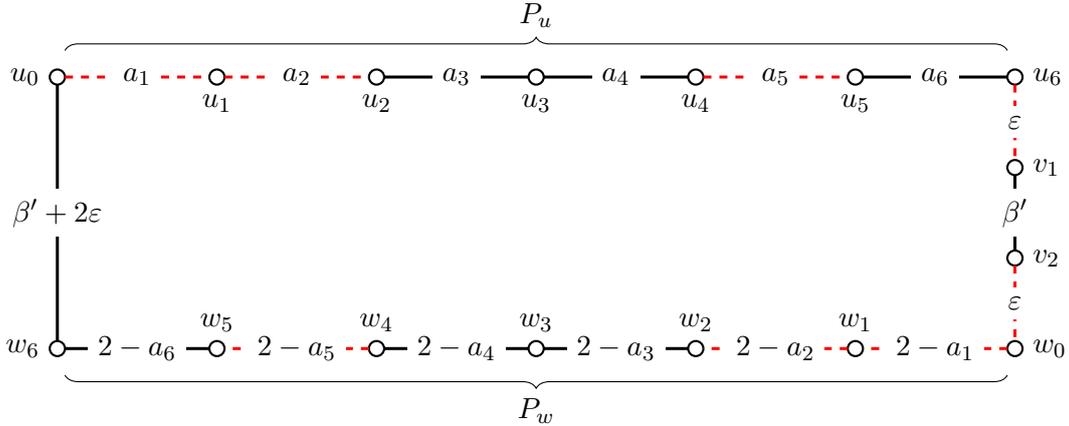


FIGURE 7. Reduction from CLOSE-TO-1 PARTITION to CONTRACTION for $n = 6$ discussed in the proof of Theorem 8. The dashed edges in the figure represent the set $C(I)$ for $I = \{1, 2, 5\}$.

Now consider the instance $\mathcal{J} := (C_{2n+4}, \ell, \varphi)$ with $\varphi(x) = x - \beta$ of the problem CONTRACTION. Observe that no φ -contraction may contain an edge $\{u, v\}$ of length greater than β (in particular, no feasible solution may contain one of the edges of length β' or $\beta' + 2\varepsilon$). Furthermore any (weak) φ -contraction C on this graph satisfies $f(C) = |C|$.

We will show that \mathcal{J} has an optimal solution of cardinality (and thus of value) $n + 2$ if and only if \mathcal{I} is a ‘Yes’-instance. In particular, we will see that any feasible solution of \mathcal{J} of size $n + 2$ contains the two edges of length ε and exactly $n/2$ edges with length a_i , $i \in I$, from P_u and the corresponding edges with length $2 - a_i$, $i \in I$, from P_w . Such solutions correspond to subsets of $[n]$ in the following natural way: For any subset $I \subseteq [n]$ of size $n/2$ we let $C(I)$ be the subset of edges of the cycle C_{2n+4} consisting of the two edges

of length ε and of all edges $\{u_{i-1}, u_i\}$ and $\{w_{i-1}, w_i\}$ (of length a_i or $2 - a_i$, respectively) for all $i \in I$. Thus we will show that $C(I)$ is an optimal solution of the instance \mathcal{J} of CONTRACTION if and only if $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = n/2$, i.e., \mathcal{I} is a ‘Yes’-instance of CLOSE-TO-1 PARTITION.

Both directions of this equivalence are captured and proved as Claim 2 and 4 below. Claims 1 and 3 are auxiliary statements used in the proof of these two main claims.

For any path P on the cycle we let $\ell(P)$ denote the sum of $\ell(e)$ over all edges e of P . For all $i \in [n]$ we denote by P_i^\sqsupset and P_i^\sqsubset the path on the cycle between the vertices u_i and w_i that contains and that does not contain the edge $\{v_1, v_2\}$, respectively (in Figure 7, these are the right and left segment of the cycle).

Claim 1: For all $i \in [n]$, the number $\ell(P_i^\sqsupset)$ lies in the interval $[n + \beta' + \varepsilon, n + \beta' + 2\varepsilon]$ and the number $\ell(P_i^\sqsubset)$ lies in the interval $[n + \beta' + 2\varepsilon, n + \beta' + 3\varepsilon]$. In particular, we have $\text{dist}_\ell(u_i, w_i) = \min\{\ell(P_i^\sqsupset), \ell(P_i^\sqsubset)\} = \ell(P_i^\sqsupset)$ and the difference $\ell(P_i^\sqsubset) - \ell(P_i^\sqsupset)$ lies in the interval $[0, 2\varepsilon]$.

Proof of Claim 1: Note that the condition $\sum_{i=1}^n a_i = n$ implies that

$$\varepsilon = 2 \sum_{a_i: a_i \geq 1} (a_i - 1) = 2 \sum_{a_i: a_i < 1} (1 - a_i). \quad (23)$$

By our assumption $a_1 \geq a_2 \geq \dots \geq a_n$, the numbers $\ell(P_i^\sqsupset)$ form a unimodular sequence for $i = 0, 1, \dots, n$ that is maximized for $i = 0$ and $i = n$, proving that $\ell(P_i^\sqsupset) \leq n + \beta' + 2\varepsilon$ (note that $\ell(P_u) = \ell(P_w) = n$). By (23) the minimum of this unimodular sequence is at most ε smaller than the maximum. This proves the first part of the claim. As $\ell(P_i^\sqsupset) + \ell(P_i^\sqsubset) = 2(n + \beta' + 2\varepsilon)$, we obtain the second part of the claim. The last part of the claim is an immediate consequence of the first two. \square

Claim 2: If $I \subseteq [n]$ is a solution of the instance \mathcal{I} of CLOSE-TO-1 PARTITION such that $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = n/2$, then $C(I)$ is a $(1, \beta)$ -contraction.

Proof of Claim 2: It suffices to prove that there is no pair of vertices whose distance decreases by more than β when contracting the edges in $C(I)$.

We start by verifying this for the pairs u_i, w_i for $i \in [n]$. We first consider the path P_i^\sqsupset between u_i and w_i . Observe that $\sum_{e \in C(I) \cap P_i^\sqsupset} \ell(e)$ lies in the interval $[n/2 + \varepsilon, n/2 + 2\varepsilon] = [\beta - \varepsilon, \beta]$. Similarly to before, this follows from the observation that by the assumption $a_1 \geq a_2 \geq \dots \geq a_n$ those sums form a unimodular sequence for $i = 0, 1, \dots, n$ that is maximized for $i = 0$ and $i = n$, and by using (23) (recall also that $|I| = n/2$). Consequently, we have

$$\ell_{C(I)}(P_i^\sqsupset) \geq \ell(P_i^\sqsupset) - \beta. \quad (24)$$

Since $\sum_{e \in C(I)} \ell(e) = n + 2\varepsilon = 2\beta - 2\varepsilon$, we obtain that $\sum_{e \in C(I) \cap P_i^\sqsubset} \ell(e)$ lies in the interval $[\beta - 2\varepsilon, \beta - \varepsilon]$, yielding

$$\ell_{C(I)}(P_i^\sqsubset) \geq \ell(P_i^\sqsubset) - (\beta - \varepsilon) \geq \ell(P_i^\sqsubset) - \beta. \quad (25)$$

Combining (24) and (25) proves that

$$\text{dist}_{\ell_{C(I)}}(u_i, w_i) \geq \text{dist}_\ell(u_i, w_i) - \beta. \quad (26)$$

Now consider two vertices u_i and w_j , $j < i$ (the case $j > i$ can be treated analogously). Let $P_{i,j}^\sqsupset$ and $P_{i,j}^\sqsubset$ be the path on the cycle between the vertices u_i and w_j that contains and that does not contain the edge $\{v_1, v_2\}$, respectively. Using that $P_{i,j}^\sqsupset \subseteq P_i^\sqsupset$ we obtain

$$\ell_{C(I)}(P_{i,j}^\sqsupset) \geq \ell(P_{i,j}^\sqsupset) - \beta \quad (27)$$

from (24).

We know that $a_i \leq 1 + 1/5 \leq 8/5$ and consequently

$$2 - a_i \geq 2/5 \geq 2\varepsilon \quad (28)$$

by the assumption that the input $\{a_1, a_2, \dots, a_n\}$ of the instance \mathcal{I} is close to 1 (there is plenty of leeway in all those inequalities). Furthermore, we have

$$\text{dist}_\ell(u_i, w_j) \leq \ell(P_i^\sqsupset) - (2 - a_i) \stackrel{(28)}{\leq} \ell(P_i^\sqsupset) - 2\varepsilon \leq \min\{\ell(P_i^\sqsupset), \ell(P_i^\sqsubset)\} = \text{dist}_\ell(u_i, w_i), \quad (29)$$

where the second-to-last inequality follows from Claim 1.

Combining those observations yields

$$\ell_{C(I)}(P_{i,j}^\sqsubset) \geq \text{dist}_{\ell_{C(I)}}(u_i, w_i) \stackrel{(26)}{\geq} \text{dist}_\ell(u_i, w_i) - \beta \stackrel{(29)}{\geq} \text{dist}_\ell(u_i, w_j) - \beta. \quad (30)$$

Combining (27) and (30) proves that

$$\text{dist}_{\ell_{C(I)}}(u_i, w_j) \geq \text{dist}_\ell(u_i, w_j) - \beta. \quad (31)$$

From (27) and (30) we can derive analogous relations for the remaining cases where we need to consider the distance between a vertex u_i , $i \in [n]$, and a vertex $w \in \{v_1, v_2, u_0, u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_n\}$, between a vertex w_i , $i \in [n]$, and a vertex $u \in \{v_1, v_2, w_0, w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_n\}$, and between the vertices v_1 and v_2 . This completes the proof of Claim 2. \square

Claim 3: Every $(1, \beta)$ -contraction C contains at most $n/2$ edges in $(P_u \cup P_w) \cap P_i^\sqsupset$ and at most $n/2$ edges in $(P_u \cup P_w) \cap P_i^\sqsubset$ for all $i \in [n]$.

Proof of Claim 3: Note that for any $I \subseteq [n]$ and $k \in \{0, 1, \dots, n\}$ we have $\sum_{i \in I: i > k} a_i + \sum_{i \in I: i \leq k} (2 - a_i) \geq |I| - \varepsilon$ by the definition of ε . Consequently, assuming for the sake of contradiction that C contains strictly more than $n/2$ edges in $(P_u \cup P_w) \cap P_i^\sqsupset$, we have $\ell_C(P_i^\sqsupset) - \ell(P_i^\sqsupset) \geq n/2 + 1 - \varepsilon$. Similarly, assuming that C contains strictly more than $n/2$ edges in $(P_u \cup P_w) \cap P_i^\sqsubset$ yields $\ell_C(P_i^\sqsubset) - \ell(P_i^\sqsubset) \geq n/2 + 1 - \varepsilon$. By Claim 1 the difference $\ell(P_i^\sqsubset) - \ell(P_i^\sqsupset)$ lies in the interval $[0, 2\varepsilon]$, so in both cases we obtain

$$\text{dist}_\ell(u_i, w_i) - \text{dist}_{\ell_C}(u_i, w_i) \geq \frac{n}{2} + 1 - \varepsilon - 2\varepsilon > \frac{n}{2} + 2\varepsilon = \beta,$$

where we used that $\varepsilon < 1/5$ in the second-to-last step. This contradicts the fact that C is a $(1, \beta)$ -contraction, proving Claim 3. \square

Claim 4: Let C be a feasible solution of the instance \mathcal{J} of CONTRACTION. Then we have $|C| \leq n + 2$, and if $|C| = n + 2$, we have $C = C(I)$ for some set $I \subseteq [n]$ with $\sum_{i \in I} a_i = \sum_{i \in [n] \setminus I} a_i = n/2$.

Proof of Claim 4: As C does not contain any of the edges of length β' or $\beta' + 2\varepsilon$, we have $|C| \leq n + 2$ by Claim 3 (the $+2$ comes from the two edges of length ε that may be contained in C). Suppose now that $|C| = n + 2$. Applying Claim 3 again shows that C must contain both edges of length ε , and that it contains the edge $\{u_{i-1}, u_i\}$ if and only if it contains the edge $\{w_{i-1}, w_i\}$, for all $i \in [n]$. Defining $I := \{i \in [n] : \{u_{i-1}, u_i\} \in C\}$ we have $|I| = n/2$ and $C = C(I)$.

By Claim 1 we have $\text{dist}_\ell(u_0, w_0) = \ell(P_0^\sqsupset)$ and $\text{dist}_\ell(u_n, w_n) = \ell(P_n^\sqsupset)$. As C is a $(1, \beta)$ -contraction containing the two edges of length ε we thus obtain $\sum_{i \in I} a_i = \sum_{e \in C \cap P_u} \ell(e) \leq \beta - 2\varepsilon = n/2$. Similarly, we have $\sum_{i \in [n] \setminus I} a_i = \sum_{i \in I} (2 - a_i) = \sum_{e \in C \cap P_w} \ell(e) \leq \beta - 2\varepsilon = n/2$. As $\sum_{i \in [n]} a_i = n$, these two inequalities must be tight, yielding $\sum_{i \in A} a_i = \sum_{i \in [n] \setminus I} a_i = n/2$. \square

Combining Claims 2 and 4 proves the statement of the theorem for the problem CONTRACTION.

The hardness proof for the problem WEAK CONTRACTION is analogous, only the argument that no feasible solution C of the instance \mathcal{J} contains one of the edges of length β' or $\beta' + 2\varepsilon$ (this is part of the proof of Claim 4) has to be adjusted. \blacksquare

The reader might be tempted to ‘simplify’ the previous reduction proof by omitting the four special edges of length ε , β' and $\beta' + 2\varepsilon$ and by setting $\beta := n/2$ instead. However, this would invalidate Claim 2 (specifically, the estimate (25) would not always hold).

5.2. Inapproximability of Contraction. We are able to extend the before-mentioned hardness result for CONTRACTION as follows:

Theorem 9. *For any fixed $\beta > 0$ and $\varepsilon > 0$, it is NP-hard to approximate the problem CONTRACTION with tolerance function $\varphi(x) = x - \beta$, $\beta \geq 0$, to within a factor of $n^{1-\varepsilon}$.*

For the following theorem the additive error is fixed to $\beta = 1$.

Theorem 10. *For any $\varepsilon > 0$, it is NP-hard to approximate the problem CONTRACTION with tolerance function $\varphi(x) = x - 1$ on bipartite graphs with unit length edges $\ell = 1$ to within a factor of $m^{1/2-\varepsilon}$.*

Our reductions are based on the inapproximability of the well-known CLIQUE problem. Recall that a *clique* in a graph G is a complete subgraph of G .

| |
|--|
| <p>CLIQUE Input: A graph G. Output: A clique in G of maximum size.</p> |
|--|

It was shown in [Zuc07] that for any $\varepsilon > 0$, it is NP-hard to approximate CLIQUE to within a factor of $n^{1-\varepsilon}$.

The following lemma will be used in our proofs. It shows that for $(1, \beta)$ -contractions the feasibility condition (1) needs not be checked for all pairs of vertices u and v , but only for those satisfying certain extra conditions.

Lemma 5.2. *A set of edges $C \subseteq E$ is a $(1, \beta)$ -contraction if and only if all pairs of vertices $u, v \in V$ with the property that every shortest path with respect to ℓ_C between u and v starts and ends with an edge from C satisfy condition (1).*

Proof. Suppose for the sake of contradiction that all pairs of vertices $u, v \in V$ as in the lemma satisfy condition (1) and that C is *not* a $(1, \beta)$ -contraction. Then there is a pair of vertices $u, v \in V$ violating (1) and a shortest path P with respect to ℓ_C between u and v that does not start or end with an edge from C . We choose u and v such that $\text{dist}_{\ell_C}(u, v)$ is minimal, and we may assume that the first edge $\{u, w\}$ of P is not contained in C , so $\text{dist}_{\ell}(u, v) - \text{dist}_{\ell_C}(u, v) = \text{dist}_{\ell}(w, v) - \text{dist}_{\ell_C}(w, v)$. By our choice of u and v , the vertices w and v satisfy (1), i.e., the right-hand side of this equation is bounded by β , a contradiction. ■

Proof of Theorem 9. Let $\beta, \varepsilon > 0$ be fixed and let $G = (V, E)$ be an instance of CLIQUE.

We define a graph $H = H(G)$ as follows, see Figure 8: The vertex set of H is given by $(V \times \{1, 2\}) \cup \{s\}$, i.e., we create two copies of each original vertex and add a special vertex s . The edge set of H is given by $\{\{(u, 1), (v, 1)\} : \{u, v\} \in E\}$ plus the edges $\{(v, 1), (v, 2)\}$ and $\{s, (v, 2)\}$ for all $v \in V$. The first set of edges are simply the original edges of G on the first copies of the vertices, the second set is a perfect matching between the two copies of the vertex set, and the third set of edges connects the special vertex s to all vertices of the second copy of the vertex set. The length function ℓ on the edges of H is set to $2\beta + 2$, β or $\beta + 1$ for those three sets of edges, respectively.

Now consider the instance $\mathcal{I} := (H, \ell, \varphi)$ of the problem CONTRACTION with the tolerance function $\varphi(x) = x - \beta$. Clearly, any $(1, \beta)$ -contraction C in H can contain only edges of the form $\{(u, 1), (u, 2)\}$ for some $u \in V$. As H does not contain two edges between two

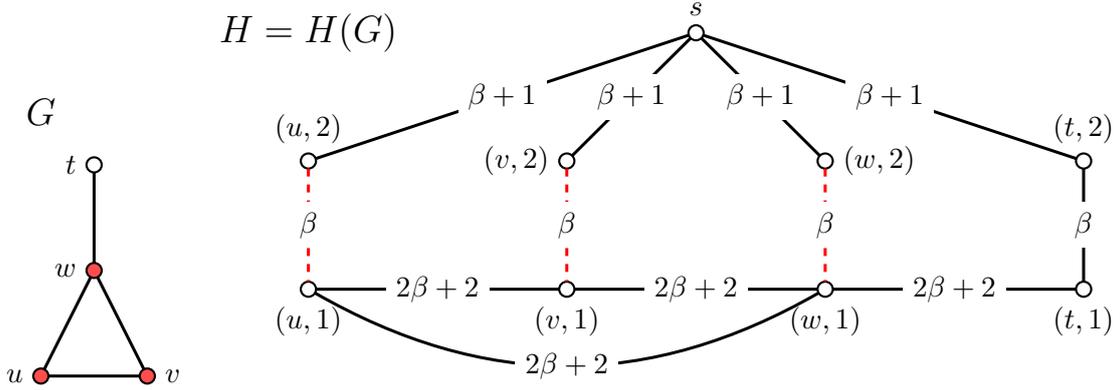


FIGURE 8. An instance G of CLIQUE (left) and the corresponding instance $\mathcal{I} = (H, \ell, \varphi)$ (right) of CONTRACTION constructed in the proof of Theorem 9. The dashed edges form the set $C(U)$ for $U = \{u, v, w\}$.

different connected components of (V, C) , our objective function defined in (2) satisfies $\Phi(C) = |C|$ for any feasible solution C of \mathcal{I} . We will show that it allows a feasible solution with k edges (and thus of value k) if and only if G has a clique with k vertices. Formally, for $U \subseteq V$ we define $C(U) := \{(u, 1), (u, 2)\} : u \in U\}$ (see Figure 8). We proceed to show that U induces a clique in G if and only if $C(U)$ is a $(1, \beta)$ -contraction in $H = H(G)$.

Note that for any two vertices $u, v \in U$ we have

$$\begin{aligned} \text{dist}_{\ell_{C(U)}}((u, 1), (v, 1)) &= 2\beta + 2 = \begin{cases} \text{dist}_{\ell}((u, 1), (v, 1)) & \text{if } \{u, v\} \in E, \\ \text{dist}_{\ell}((u, 1), (v, 1)) - 2\beta & \text{otherwise,} \end{cases} \\ \text{dist}_{\ell_{C(U)}}((u, 1), (v, 2)) &= 2\beta + 2 = \text{dist}_{\ell}((u, 1), (v, 2)) - \beta, \\ \text{dist}_{\ell_{C(U)}}((u, 2), (v, 2)) &= 2\beta + 2 = \text{dist}_{\ell}((u, 2), (v, 2)), \\ \text{dist}_{\ell_{C(U)}}((u, 1), (u, 2)) &= 0 = \text{dist}_{\ell}((u, 1), (u, 2)) - \beta. \end{aligned}$$

These relations together with Lemma 5.2 show that $C(U)$ is a $(1, \beta)$ -contraction in H if and only if U is a clique in G .

As $n(H)$ differs from $n(G)$ only by a constant factor, an $n^{1-\varepsilon}$ -approximation algorithm for CONTRACTION would yield an $n^{1-\varepsilon'}$ -approximation algorithm for CLIQUE via this reduction. Together with the before-mentioned inapproximability of CLIQUE [Zuc07] this proves the theorem. \blacksquare

The rest of this section is devoted to proving Theorem 10, so we now focus on $(1, 1)$ -contractions in bipartite graphs with unit length edges $\ell = 1$. The next lemma characterizes the structure of contractions in this setting.

Lemma 5.3. *Let $G = (V, E)$ be a bipartite graph with unit edge lengths $\ell = 1$ and let $C \subseteq E$ be a set of edges.*

- (i) *If C is a $(1, 1)$ -contraction, then C is a matching.*
- (ii) *If $C = \{e, f\}$ with edges $e = \{u_1, u_2\}, f = \{v_1, v_2\} \in E$, then C is a $(1, 1)$ -contraction if and only if $\text{dist}_{\ell}(u_1, v_1) = \text{dist}_{\ell}(u_2, v_2)$ and $\text{dist}_{\ell}(u_1, v_2) = \text{dist}_{\ell}(u_2, v_1)$.*
- (iii) *C is a $(1, 1)$ -contraction if and only if all two-element subsets of C are.*

Proof. (i) Suppose for the sake of contradiction that C contains a path (u, v, w) on two edges. As G is bipartite, it has no triangles, so $\text{dist}_{\ell}(u, w) = 2$ and $\text{dist}_{\ell_C}(u, w) = 0$, a contradiction to the assumption that C is a $(1, 1)$ -contraction.

- (ii) For the edges $e = \{u_1, u_2\}$ and $f = \{v_1, v_2\}$ we define $d_{i,j} := \text{dist}_\ell(u_i, v_j)$ for $i, j \in \{1, 2\}$.

Let $C = \{e, f\}$ be a $(1, 1)$ -contraction. Both $d_{1,1}$ and $d_{2,2}$ must have the same parity (as G is bipartite), so if $d_{1,1} < d_{2,2}$, the difference between them is exactly 2. However, this would mean that $\text{dist}_{\ell_C}(u_2, v_2) = d_{1,1} = d_{2,2} - 2 = \text{dist}_\ell(u_2, v_2) - 2$, a contradiction to the assumption that C is a $(1, 1)$ -contraction. Repeating the same argument with $d_{1,1}$ and $d_{2,2}$ interchanged shows that $d_{1,1} = d_{2,2}$. An analogous argument shows that $d_{1,2} = d_{2,1}$.

Now suppose that $d_{1,1} = d_{2,2}$ and $d_{1,2} = d_{2,1}$. From these conditions it follows that for all $i, j \in \{1, 2\}$ every path between u_i and v_j that contains both edges e and f has length at least $d_{i,j} + 2$ with respect to ℓ . Consequently, we have $\text{dist}_{\ell_C}(u_i, v_j) \geq \text{dist}_\ell(u_i, v_j) - 1$ for $C = \{e, f\}$. By Lemma 5.2, C is a $(1, 1)$ -contraction.

- (iii) One direction of the equivalence is obvious, so we only need to prove the other direction. So we assume that all two-element subsets of C are $(1, 1)$ -contractions, and we need to prove that C is a $(1, 1)$ -contraction. The argument is a straightforward generalization of the argument for (ii) from before. Let P be a path that contains exactly k edges from C , and that starts and ends with an edge from C . Let e_1, e_2, \dots, e_k be those edges and $u_{1,1}, u_{1,2}, u_{2,1}, u_{2,2}, \dots, u_{k,1}, u_{k,2}$ their end vertices as they are encountered when traversing P (so $u_{1,1}$ and $u_{k,2}$ are the end vertices of P). For all $i = 1, 2, \dots, \lfloor k/2 \rfloor$ the pair of edges e_{2i-1} and e_{2i} and their end vertices satisfy the distance conditions from (ii). From these conditions it follows that the subpath of P between $u_{2i-1,1}$ and $u_{2i,2}$ has length at least $\text{dist}_\ell(u_{2i-1,1}, u_{2i,2}) + 2$. So overall the length of P is at least $\text{dist}_\ell(u_{1,1}, u_{k,2}) + 2\lfloor k/2 \rfloor \geq \text{dist}_\ell(u_{1,1}, u_{k,2}) + (k - 1)$. Consequently, we have $\text{dist}_{\ell_C}(u_{1,1}, u_{k,2}) \geq \text{dist}_\ell(u_{1,1}, u_{k,2}) - 1$. By Lemma 5.2, C is a $(1, 1)$ -contraction. ■

With Lemma 5.3 in hand, we are now ready to prove Theorem 10.

Proof of Theorem 10. Let $\varepsilon > 0$ be fixed and let $G = (V, E)$ be an instance of CLIQUE. We construct a bipartite graph $H = H(G)$ as follows, see Figure 9: For every vertex $v \in V$, the graph H contains two vertices $(v, 1)$ and $(v, 2)$ and the edge $f_v := \{(v, 1), (v, 2)\}$. For every edge $e = \{u, v\} \in E$, we add a vertex x_e and the edges $f_{e,u} := \{x_e, (u, 1)\}$ and $f_{e,v} := \{x_e, (v, 1)\}$ to H . Furthermore, we add a new special vertex s to H and all the edges $\{s, (v, 2)\}$, $v \in V$, and $\{s, x_e\}$, $e \in E$. It is easy to check that the graph H defined in this way is bipartite.

All edges of H receive unit lengths ($\ell = 1$) and we consider the instance $\mathcal{I} = (H, \ell, \varphi)$ of the problem CONTRACTION with the tolerance function $\varphi(x) = x - 1$.

For any set of vertices $U \subseteq V$ we define $C(U) := \{f_u : u \in U\}$ (see Figure 9).

Claim 1: If $U \subseteq V$ is a clique in G , then $C(U)$ is a $(1, 1)$ -contraction in H and $\Phi(C(U)) = |U|$.

Proof of Claim 1: Let U be a set of vertices in G that form a clique, and let $u, v \in U$ be two vertices from this clique. Then we have $\text{dist}_\ell((u, 1), (v, 1)) = \text{dist}_\ell((u, 2), (v, 2)) = 2$ and $\text{dist}_\ell((u, 1), (v, 2)) = \text{dist}_\ell((u, 2), (v, 1)) = 3$, so Lemma 5.3 (ii) implies that $C(\{u, v\})$ is a $(1, 1)$ -contraction in H . Repeating this argument for every pair of vertices from U and applying Lemma 5.3 (iii) yields that $C(U)$ is a $(1, 1)$ -contraction in H . As there are never two edges in H between any two connected components of the graph $(V, C(U))$, we have $\Phi(C(U)) = |C(U)| = |U|$. □

For any set of edges $C \subseteq E(H)$, we let $U(C)$ be the set of vertices $v \in V$ for which $(v, 1)$ is incident to an edge in C .

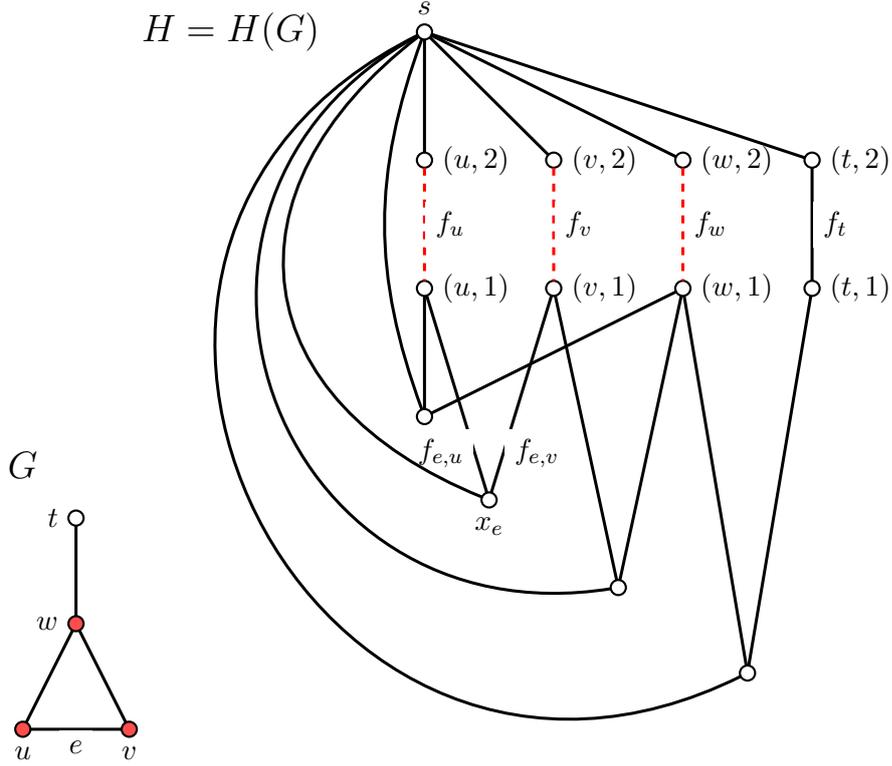


FIGURE 9. An instance G of CLIQUE (left) and the corresponding instance $\mathcal{I} = (H, \ell, \varphi)$ (right) of CONTRACTION constructed in the proof of Theorem 10. The dashed edges form the set $C(U)$ for $U = \{u, v, w\}$.

Claim 2: If $C \subseteq E(H)$ is a $(1, 1)$ -contraction, then C is a matching in H and $U(C)$ is a clique in G of size at least $\Phi(C) - 3$.

Proof of Claim 2: C is a matching by Lemma 5.3 (i).

Let $u, v \in U(C)$. We will show that $e = \{u, v\} \in E$ by applying Lemma 5.3 (ii) to the two edges in C incident to $(u, 1)$ and $(v, 1)$. To prove that $e \in E$ it suffices to show that $\text{dist}_\ell((u, 1), (v, 1)) = 2$.

Let us first consider the case that $f_u, f_v \in C$. As $\text{dist}_\ell((u, 2), (v, 2)) = 2$ (the shortest path between those vertices goes via s), Lemma 5.3 (ii) implies that $\text{dist}_\ell((u, 1), (v, 1)) = 2$. We now consider the case that there is an edge $e' \in E \setminus \{e\}$ with $f_u, f_{e',v} \in C$. We then have $\text{dist}_\ell((u, 2), x_{e'}) = 2$ (via s), so Lemma 5.3 (ii) yields $\text{dist}_\ell((u, 1), (v, 1)) = 2$. Finally, we consider the case that there are two edges $e', e'' \in E \setminus \{e\}$ with $f_{e',u}, f_{e'',v} \in C$. We then have $\text{dist}_\ell(x_{e'}, x_{e''}) = 2$ (via s), again implying that $\text{dist}_\ell((u, 1), (v, 1)) = 2$. This proves that indeed $e \in E$, so $U(C)$ forms a clique in G .

Every edge in H is either incident to s or to a vertex of the form $(v, 1)$, $v \in V$. Since at most one of the edges incident to s can be in C , the definition of $U(C)$ shows that the size of $U(C)$ is either $|C| - 1$ or $|C|$. Therefore, to finish the proof of Claim 2, it suffices to show that $\Phi(C) \leq |C| + 2$. If C contains no two edges that are connected by more than one edge in H , then we have $\Phi(C) = |C|$. Otherwise we consider two such edges f and g from C . It is easy to check that either f or g must be incident to s , so suppose that the edge f contains s . We first consider the case that $f = \{s, x_e\}$ for some edge $e = \{u, v\} \in E$. In this case it follows that $g = \{(u, 1), (u, 2)\}$ or $g = \{(v, 1), (v, 2)\}$, so we have $\Phi(C) = |C| + 2$. Now consider the case that $f = \{s, (u, 2)\}$ for some vertex $u \in V$.

In this case it follows that $g = \{(u, 1), x_e\}$ for exactly one edge $e \in E$ incident to u in G , showing that $\Phi(C) = |C| + 2$. In all three cases we have $\Phi(C) \leq |C| + 2$, as claimed. \square

Combining Claims 1 and 2 will allow us to prove the following claim:

Claim 3: If there is an $n^{1/2-\varepsilon}$ -approximation algorithm for CONTRACTION, then there is an $n^{1-\varepsilon/2}$ -approximation algorithm for CLIQUE.

Proof of Claim 3: Suppose for the sake of contradiction that such an approximation algorithm for CONTRACTION exists. We use it to compute a clique in a given instance G of CLIQUE as follows: We construct $\mathcal{I} = (H(G), \ell, \varphi)$ and compute a solution C of CONTRACTION for this instance, and we define the clique $U(C)$ as before (recall Claim 2). If $U(C) \neq \emptyset$, we return $U(C)$, otherwise we return any vertex from G . We denote the clique computed in this fashion by U .

We may assume that $n(G) \geq 16^{1/\varepsilon}$, in particular $n(H) \geq 16^{1/\varepsilon}$. It follows that

$$n(H) = 1 + 2n(G) + m(G) \leq 1 + 2n(G) + \binom{n(G)}{2} \leq n(G)^2. \quad (32)$$

By assumption we know that

$$\Phi(C) \cdot n(H)^{1/2-\varepsilon} \geq \Phi(C^*), \quad (33)$$

where C^* is an optimal solution of \mathcal{I} . In particular, $\Phi(C)$ is positive.

Combining these observations we get

$$\begin{aligned} |U| \cdot n(G)^{1-\varepsilon/2} &\stackrel{(32)}{\geq} |U| \cdot n(H)^{1/2-\varepsilon/2} \\ &\geq \max\{\Phi(C) - 3, 1\} \cdot n(H)^{1/2-\varepsilon/2} \\ &= (\max\{\Phi(C) - 3, 1\} \cdot n(H)^{\varepsilon/2}) \cdot n(H)^{1/2-\varepsilon} \\ &\geq \Phi(C) \cdot n(H)^{1/2-\varepsilon} \\ &\stackrel{(33)}{\geq} \Phi(C^*) \\ &\geq \omega(G), \end{aligned}$$

where the second inequality holds because of Claim 2, and the last inequality involving the clique number $\omega(G)$ holds because of Claim 1. \square

As $m(H) = \Theta(n(H))$, Claim 3 implies the theorem (using the inapproximability of CLIQUE proved in [Zuc07]). \blacksquare

6. HARDNESS FOR MULTIPLICATIVE TOLERANCE FUNCTION

By Theorem 7, the problem WEAK CONTRACTION with purely additive tolerance function $\varphi(x) = x - \beta$ is NP-hard on cycles. In this section we prove the hardness and inapproximability of this problem also in the case of a purely multiplicative tolerance function $\varphi(x) = x/\alpha$, $\alpha \geq 1$. Recall that the problem CONTRACTION is trivial for this tolerance function (we may not contract any edges).

6.1. Hardness of planar Weak Contraction. To state the main result of this section recall that the *girth* of a graph G is defined as the minimum length of a cycle in G .

Theorem 11. *For any $g \geq 2$, the problem WEAK CONTRACTION with tolerance function $\varphi(x) = x/2$, is NP-hard for planar graphs with girth at least $3g$ and unit length edges $\ell = 1$.*

Theorem 11 implies that WEAK CONTRACTION is hard for a general multiplicative tolerance function $\varphi(x) = x/\alpha$, $\alpha \geq 1$, but it leaves open the question whether this is true also for other fixed values of α other than 2 (when α is not part of the input). The

arguments given in this section for $\alpha = 2$ carry over straightforwardly to any fixed value $2 \leq \alpha < 3$, but not to 3 or larger values (for $\alpha < 2$ and unit length edges the problem is trivial).

We first characterize the set of feasible solutions in this special case.

Lemma 6.1. *Let $G = (V, E)$ be a graph with girth at least 6 and unit length edges $\ell = 1$, and consider the tolerance function $\varphi(x) = x/2$. Furthermore, let $C \subseteq E$ be a set of edges such that (V, C) is disconnected. Then C is a weak $(2, 0)$ -contraction if and only if for any two edges $e, f \in C$ either e and f are incident and both contain a degree-1 vertex, or any path containing e and f also contains at least two edges not in C .*

Recall that the assumption that (V, C) is disconnected prevents solutions $C \subseteq E$ for which the contracted graph G/C is a single vertex. Note that Lemma 6.1 does not require G to be planar.

Proof. To prove the equivalence, we need the following auxiliary claim:

Claim: If C is a weak $(2, 0)$ -contraction, then every component of (V, C) that is not a single edge is a star with the property that each of its vertices except the center of the star has degree 1 in G .

Proof of Claim: Let M be a component of (V, C) with more than one edge. Clearly, there must be an edge $\{u, v\}$ with vertices $u \notin V(M)$ and $v \in V(M)$. If M contains a path P on two edges starting at v and ending at some vertex w , then $\text{dist}_\ell(u, w) = 3$ and $\text{dist}_{\ell_C}(u, w) = 1$, a contradiction to the assumption that C is a weak $(2, 0)$ -contraction (note that $P \cup \{u, v\}$ is the shortest path between u and w , as the girth of G is at least 6). Thus the edges of M must form a star centered at v . By the same argument, no vertex outside M can be connected to any vertex of M other than v . This proves the claim. \square

We first assume that C is a weak $(2, 0)$ -contraction, and we need to show that any two edges $e, f \in C$ satisfy the conditions of the lemma. If e and f are incident, the statement follows from the auxiliary claim from before. If e and f are not incident, we consider an inclusion-minimal path P containing both e and f . We let u and v be the end vertices of P , u' the other end vertex of e , and v' the other end vertex of f (u' and v' are the vertices at distance 1 from the ends of the path). If the distance between u' and v' was only 1, we have $\text{dist}_\ell(u, v) = 3$ and $\text{dist}_{\ell_C}(u, v) = 1$ (here we need again the assumption that the girth is at least 6), a contradiction to the assumption that C is a weak $(2, 0)$ -contraction. Therefore at least two edges lie between u' and v' . The auxiliary claim from before implies that no two incident edges on P between u and v are contained in C , therefore P must contain at least two edges not in C . This proves one direction of the equivalence.

To prove the other direction, we now assume that any two edges e, f satisfy the conditions of the lemma, and we need to show that C is a weak $(2, 0)$ -contraction. Consider any two vertices u and v with $\text{dist}_{\ell_C}(u, v) > 0$, and any path between u and v . As no inner vertex of P is a leaf, we know that between any two consecutive edges from C on P there are at least 2 edges not in C . This proves that $\text{dist}_{\ell_C}(u, v) \geq \text{dist}_\ell(u, v)/2$, as desired.

This completes the proof of the lemma. \blacksquare

For a given propositional formula F in conjunctive normal form (CNF) the bipartite *variable-clause graph* $\Gamma(F)$ is defined as follows: The two partition classes of $\Gamma(F)$ are given by the sets of variables and clauses of F , and there is an edge between a variable x and a clause c if x appears in c . If c contains x as a positive or negative literal, we call the corresponding edge of $\Gamma(F)$ a *positive or negative edge*, respectively. A planar drawing of $\Gamma(F)$, where positive and negative edges appear in cyclically contiguous intervals around every variable vertex, is called *contiguous*.

We call a k -CNF formula *regular*, if every clause contains exactly k literals, no clause contains a literal twice, every variable appears at least once as a positive literal and at least once as a negative literal in the formula.

Consider now the following variant of 3SAT.

CONTIGUOUS PLANAR 3SAT

Input: A regular 3-CNF formula F and a contiguous planar drawing of $\Gamma(F)$.

Output: ‘Yes’, if F has a satisfying assignment, ‘No’ otherwise.

Lemma 6.2. CONTIGUOUS PLANAR 3SAT is NP-complete.

Proof. The more general variant of CONTIGUOUS PLANAR 3SAT not requiring F to be regular was shown to be NP-complete in [dBK12]. We now show how to reduce this generalization to CONTIGUOUS PLANAR 3SAT, which will prove the lemma. Given a (not necessarily regular) 3-CNF formula F we first eliminate all variables appearing only as negative or only as positive literals and all clauses containing exactly one literal, as well as multiple appearances of literals in the same clause. This yields a formula F' in which all clauses have two or three literals, no clause contains a literal twice, and every variable appears at least once as a positive literal and at least once as a negative literal in F' . Moreover, since $\Gamma(F')$ is a subgraph of $\Gamma(F)$, we also obtain a contiguous planar drawing of $\Gamma(F')$. As a last step we eliminate clauses c with two literals by introducing a new variable x for each of them and replacing c by the equivalent formula $(c \vee x) \wedge (c \vee \bar{x})$. It is easy to check that the resulting formula F'' is regular and equisatisfiable to F , and to obtain a contiguous planar drawing of $\Gamma(F'')$, see Figure 10.

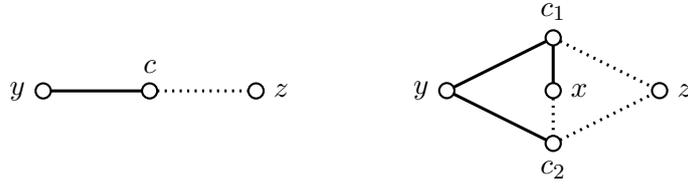


FIGURE 10. The clause $c = (y \vee \bar{z})$ is replaced by $c_1 \wedge c_2 = (c \vee x) \wedge (c \vee \bar{x})$, preserving the contiguous planar drawing of the variable-clause graph. The positive and negative edges are drawn as solid and dotted edges, respectively.

■

Proof of Theorem 11. We first present the proof for the case $g = 2$, and then sketch how to generalize it for larger values of g .

We reduce CONTIGUOUS PLANAR 3SAT to WEAK CONTRACTION. Consider an instance F of CONTIGUOUS PLANAR 3SAT with variables x_1, x_2, \dots, x_n and clauses c_1, c_2, \dots, c_m .

Given the formula F , we construct from it a graph $G = G(F)$ as follows, see Figures 11 and 12. For every variable x_i , $i \in [n]$, we add a variable gadget $H(x_i)$ as shown on the left hand side of Figure 11 to the graph G . The vertices u_i and \bar{u}_i will be used later to connect this gadget to other parts of the graph. The idea of the variable gadget is that an optimal solution of our instance of WEAK CONTRACTION should contain either the four edges $T_i := \{t_i, \bar{t}_i, t'_i, t''_i\}$ or the four edges $F_i := \{f_i, \bar{f}_i, f'_i, f''_i\}$, corresponding to setting x_i to **true** or **false**, respectively.

For every clause c_j , $j \in [m]$, we add a clause gadget $H(c_j)$ (a star with three edges) as shown on the right hand side of Figure 11 to the graph G . The vertices v_j^1 , v_j^2 and v_j^3 will be used later to connect this gadget to other parts of the graph. The idea of the clause gadget is that a feasible solution contains at most one of these three edges, and if it does contain one of them, this restricts the choice we have inside the respective neighbouring variable gadget.

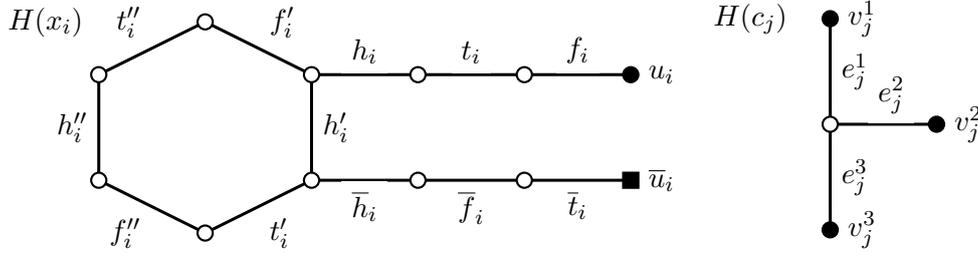


FIGURE 11. Variable gadget (left) and clause gadget (right) used in the proof of Theorem 11 for $g = 2$. The vertices used for connecting those gadgets to the rest of the graph are marked in black.

We connect the variable and clause gadgets in G as follows (see Figure 12): For every $j \in [m]$ and $k \in [3]$, if the k -th literal in the clause c_j is x_i , we add an edge connecting u_i to v_j^k , and if the k -th literal in the clause c_j is \bar{x}_i , we add an edge connecting \bar{u}_i to v_j^k . We refer to the edges added to G in this step as *connection edges*.

This completes the definition of the graph $G = G(F)$. It is easy to see that this graph is planar. Specifically, a planar embedding can be obtained from the given planar embedding of $\Gamma(F)$ by replacing variable vertices x_i in $\Gamma(F)$ by the variable gadgets $H(x_i)$ in G , and by replacing clause vertices c_j by the clause gadgets $H(c_j)$. Using that for each variable vertex x_i in $\Gamma(F)$ the positive and negative edges appear in cyclically contiguous intervals around x_i , the connection edges in G (that connect the variable and clause gadgets) can also be drawn in a planar fashion.

Moreover, it is easy to check that G has girth 6 and no degree-1 vertices.

Now consider the instance $\mathcal{I} := (G, \ell, \varphi)$ of the problem WEAK CONTRACTION with $\ell = 1$ (unit length edges) and the tolerance function $\varphi(x) = x/2$.

Lemma 6.1 implies that any feasible solution of \mathcal{I} is a matching, as G has no vertices of degree 1. As G contains no cycles of length 3 or 4, it cannot contain two edges between vertex sets of two different components of (V, C) for any such feasible solution C . This implies that our objective function satisfies $\Phi(C) = |C|$.

We proceed to show that F is satisfiable if and only if \mathcal{I} has an optimal solution of cardinality (and thus of value) $4n + m$. Specifically, a satisfying assignment of F corresponds to a solution that contains exactly all edges of either T_i or F_i in $H(x_i)$ for every variable $i \in [n]$ (corresponding to the value **true** or **false** assigned to this variable, respectively) and exactly one edge in $H(c_j)$ for each clause $j \in [m]$ (corresponding to a literal that satisfies this clause).

Formally, for any variable assignment $\tau : \{x_1, x_2, \dots, x_n\} \rightarrow \{\mathbf{true}, \mathbf{false}\}$, we define the set of edges $C(\tau) \subseteq E(G)$ as follows: $C(\tau)$ contains all edges of T_i for any variable x_i , $i \in [n]$, that τ sets to **true**, and it contains all edges of F_i for any variable x_i that τ sets to **false**. Moreover, for every clause c_j , $j \in [m]$, that is satisfied by τ , we choose an index $k \in [3]$ of a literal in c_j that is satisfied by τ and add the edge e_j^k to $C(\tau)$.

The following claim is an immediate consequence of Lemma 6.1.

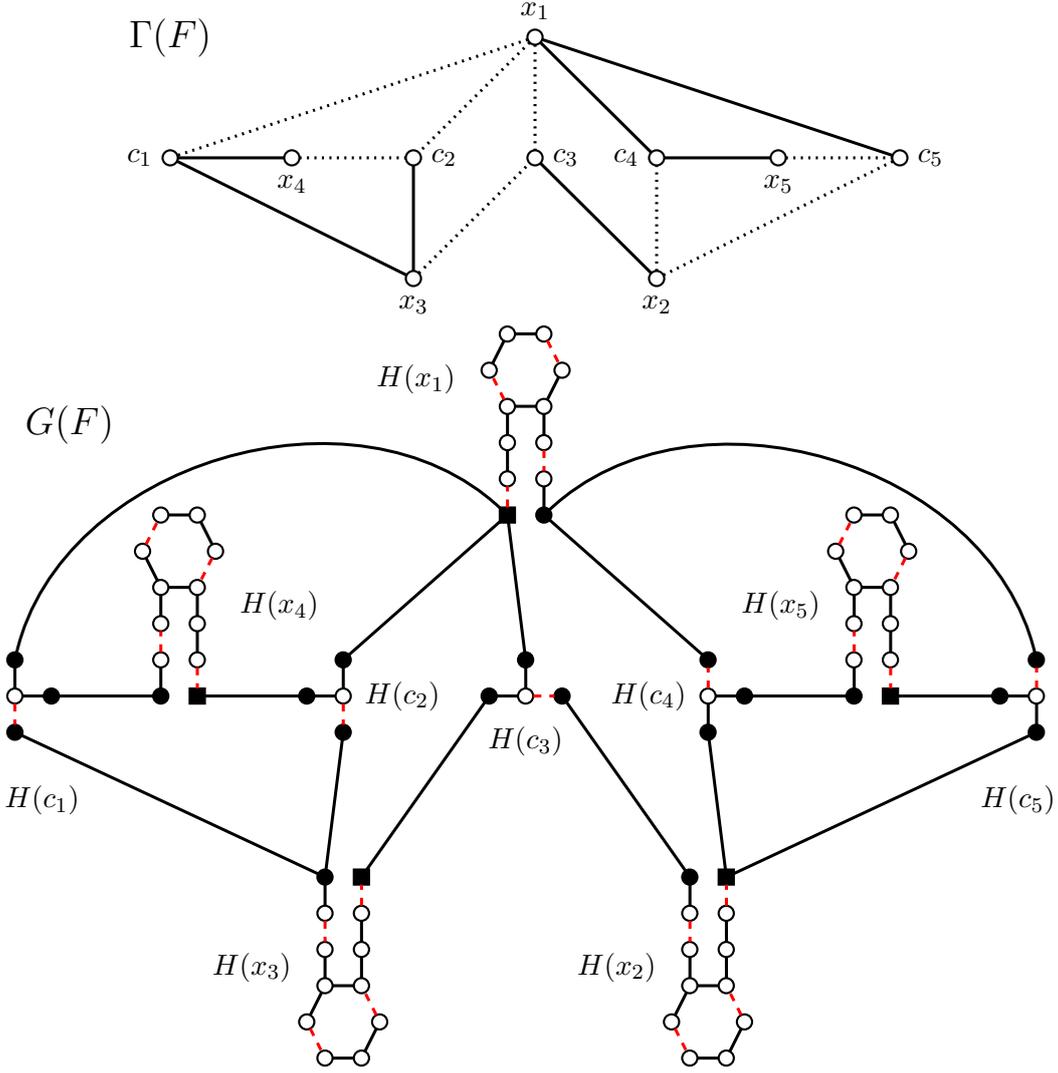


FIGURE 12. The graphs $\Gamma(F)$ (top) and $G = G(F)$ (bottom) constructed in the proof of Theorem 11 for $g = 2$ and the formula $F = c_1 \wedge c_2 \wedge c_3 \wedge c_4 \wedge c_5$ with clauses $c_1 := (\bar{x}_1 \vee x_3 \vee x_4)$, $c_2 := (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$, $c_3 := (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$, $c_4 := (x_1 \vee \bar{x}_2 \vee x_5)$ and $c_5 := (x_1 \vee \bar{x}_2 \vee \bar{x}_5)$. The positive and negative edges of $\Gamma(F)$ are drawn as solid and dotted edges, respectively. The set of dashed edges in $G(F)$ is the set $C(\tau)$ for the variable assignment τ that sets all variables to **true**.

Claim 1: Any subset $C \subseteq E(G)$ is a feasible solution if and only if every path containing two edges from C also contains at least two edges not in C .

By Claim 1, for every variable assignment τ of F , the set $C(\tau)$ is a feasible solution of \mathcal{I} . In particular, if τ satisfies F , then $C(\tau)$ is a feasible solution of size $4n + m$. The remainder of the proof is devoted to showing the converse, i.e., if $C \subseteq E(G)$ is a feasible solution of size $4n + m$, then F is satisfiable.

For all $i \in [n]$ we let $H(x_i)^+$ denote the subgraph of G induced by all edges of $H(x_i)$ and all connection edges incident to either u_i or \bar{u}_i .

Claim 2: For any $j \in [m]$, C contains at most one edge from $H(c_j)$. For any $i \in [n]$, C contains at most four edges from $H(x_i)^+$. Moreover, if C contains one of the connection

edges incident to u_i or \bar{u}_i for some $i \in [n]$, it does not contain any edges from the gadget $H(c_j)$ that is connected to $H(x_i)$ via this edge.

Proof of Claim 2: The first and last statement are immediate consequences of Claim 1. The argument for the second statement is as follows: For all $i \in [n]$ we let E_i denote the set of edges $\{h_i, t_i, f_i\}$ plus the connection edges incident to u_i , and we let \bar{E}_i denote the set of edges $\{\bar{h}_i, \bar{t}_i, \bar{f}_i\}$ plus the connection edges incident to \bar{u}_i . By Claim 1, C contains at most two edges from E_i , and if the intersection size is two, then C must contain the edge h_i . Similarly, C contains at most two edges from \bar{E}_i , and if the intersection size is two, then C must contain the edge \bar{h}_i . As C cannot contain h_i and \bar{h}_i simultaneously, C contains at most three edges from $E_i \cup \bar{E}_i$, and if the intersection size is three, then C must contain either h_i or \bar{h}_i . Again by Claim 1, C contains at most two edges from the 6-cycle $\{f'_i, h'_i, t'_i, f''_i, h''_i, t''_i\}$. However, if C contains one of the edges h_i or \bar{h}_i , it contains at most one edge from this 6-cycle. This proves that C indeed contains at most four edges from $H(x_i)^+$. \square

Note that every edge of G belongs to exactly one subgraph $H(x_i)^+$ or $H(c_j)$. So if $|C| = 4n + m$, we know by Claim 2 that C contains exactly four edges from $H(x_i)$ for all $i \in [n]$ and exactly one edge from $H(c_j)$ for all $j \in [m]$, and none of the connection edges in G .

Claim 3: For any $i \in [n]$, if C contains four edges from $H(x_i)$ and if f_i is not among them, then those edges must be T_i . On the other hand, if \bar{t}_i is not among them, those edges must be F_i . In particular, these two cases cannot occur simultaneously.

Proof of Claim 3: If C contains four edges from $H(x_i)$ and f_i is not among them, Claim 1 enforces taking first the edge t_i , then t'_i and t''_i , and eventually \bar{t}_i . This proves the first part of the statement. The argument for the second part is symmetric. The third part of the statement is a consequence of the first two. \square

So given a solution C of \mathcal{I} of size $4n + m$, we can derive from it a satisfying assignment τ of F as follows: For every clause c_j , $j \in [m]$, we consider the unique edge e_j^k from $H(c_j)$ that belongs to C . We follow the attachment edge incident to e_j^k , leading to the corresponding variable gadget $H(x_i)$, and connecting to either u_i or \bar{u}_i . If the attachment edge connects to u_i , then by Claim 1, $f_i \notin C$, so by Claim 3, the four edges of $H(x_i)$ contained in C must be T_i , so we define $\tau(x_i) := \text{true}$. If the attachment edge connects to \bar{u}_i , then by Claim 1, $\bar{t}_i \notin C$, so by Claim 3, the four edges of $H(x_i)$ contained in C must be F_i , so we define $\tau(x_i) := \text{false}$. This process does not lead to any contradicting variable assignments by the last statement of Claim 3. However, this process may leave some variables x_i undefined, and we can set them arbitrarily, e.g., $\tau(x_i) := \text{true}$. By construction, each clause receives a satisfying literal, so the assignment τ is indeed a satisfying assignment of F .

This proves that F is satisfiable if and only if \mathcal{I} has a feasible solution of size $4n + m$ (which must be optimal by Claim 2), completing the proof of the theorem in the case $g = 2$.

For values $g \geq 2$, the construction of the gadgets $H(x_i)$ and $H(c_j)$ can be generalized as follows: We subdivide each of the edges h_i, \bar{h}_i and h''_i , and each of the edges e_j^1, e_j^2 and e_j^3 into $1 + 3(g - 2)$ edges. Then the resulting graph $G = G(F)$ clearly has girth $3g$, and the above arguments can be easily modified to show that any solution C of \mathcal{I} contains at most $1 + 3(g - 2) = 3g - 5$ edges from $H(c_j)$ for all $j \in [m]$, and at most $4 + 3(g - 2) = 3g - 2$ edges from $H(x_i)^+$ for all $i \in [n]$, and that F is satisfiable if and only if \mathcal{I} has an optimal solution of size $(3g - 2)n + (3g - 5)m$. This completes the proof. \blacksquare

6.2. Inapproximability of Weak Contraction. We are able to further extend our hardness results for WEAK CONTRACTION as follows:

Theorem 12. For any $\varepsilon > 0$, it is NP-hard to approximate the problem WEAK CONTRACTION with tolerance function $\varphi(x) = 2x/3$ to within a factor of $n^{1-\varepsilon}$.

Theorem 12 implies that WEAK CONTRACTION is hard to approximate for general multiplicative tolerance functions $\varphi(x) = x/\alpha$, $\alpha \geq 1$, but it leaves open the question whether this is true also for other fixed values of α other than $3/2$ (when α is not part of the input). The arguments given in this section for $\alpha = 3/2$ carry over straightforwardly to any fixed value $1 < \alpha < 2$, but not to 2 or larger values (for $\alpha = 1$ the problem is trivial).

This time we reduce from the well-known INDEPENDENT SET problem (which is equivalent to CLIQUE by considering the complement graph). Recall that an *independent set* in a graph G is a subset of vertices of G such that no two vertices in the subset are adjacent.

INDEPENDENT SET
Input: A graph G .
Output: An independent set in G of maximum size.

We use again the fact that for any $\varepsilon > 0$, INDEPENDENT SET is NP-hard to approximate to within a factor of $n^{1-\varepsilon}$ [Zuc07].

Proof of Theorem 12. Let $G = (V, E)$ be an instance of INDEPENDENT SET. We construct a graph $H = H(G)$ and a length function ℓ on the edges of H as follows, see Figure 13: We start with a copy of G , and all edges of this copy receive length 2. The vertices of this copy are also denoted by V . We then add additional vertices and edges to H as follows: To every vertex $v \in V$ we attach two pending edges $\{v, (v, 1)\}$ and $\{v, (v, 2)\}$ of length 1 or 2, respectively. We may assume G , and thus also H , to be connected.

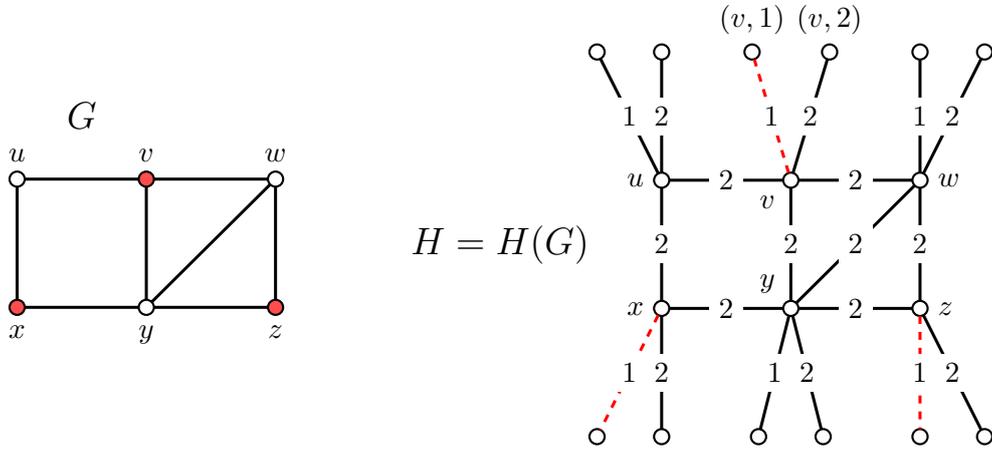


FIGURE 13. An instance G of INDEPENDENT SET (left) and the corresponding instance $H = H(G)$ of WEAK CONTRACTION (right) constructed in the proof of Theorem 12. The dashed edges are the set $C(U)$ for the independent set $U = \{v, x, z\}$.

Now consider the instance $\mathcal{I} = (H, \ell, \varphi)$ of the problem WEAK CONTRACTION with the tolerance function $\varphi(x) = 2x/3$.

We proceed to show that \mathcal{I} has a feasible solution of value k if and only if G has an independent set of size k . This is an immediate consequence of Claim 3 below. To prove Claim 3 we need the following two auxiliary claims.

Claim 1: For any induced subgraph of H that is a path on two edges, a feasible solution C of \mathcal{I} does not contain only the longer of the two edges (either it contains none of the two, the shorter of the two if there is one, or both).

Proof of Claim 1: Consider a path on two edges $\{u, v\}, \{v, w\}$ of length 2 in H such that $\{u, w\} \notin E(H)$, and suppose for the sake of contradiction that $\{u, v\} \in C$, but $\{v, w\} \notin C$. Then we have $\text{dist}_\ell(u, w) = 4$ and $\text{dist}_{\ell_C}(u, w) = 2$, violating the condition (1) for the given tolerance function. A similar contradiction arises if one of the edges has length 1 and the other length 2, and only the edge of length 2 is contracted. This proves the claim. \square

Claim 2: No feasible solution of \mathcal{I} contains an edge of length 2.

Proof of Claim 2: Assume for the sake of contradiction that a feasible solution C contains an edge e of length 2. Note that any edge f of H may be reached from e via a walk $e_1 \dots e_k$ where $e_1 = e$ and $e_k = f$, and for all $i < k - 1$ we have $\ell(e_i) = 2$ and the edges e_i and e_{i+1} induce a path in H . Now successively applying Claim 1 to the subgraphs induced by e_i and e_{i+1} for $i < k$ shows that C contracts f . Thus C violates the condition that a weak contraction must not contract every edge. \square

Claim 2 implies that our objective function satisfies $\Phi(C) = |C|$ for every feasible solution C of \mathcal{I} , because H never contains two edges between two different connected components of (V, C) .

For any set of vertices $U \subseteq V(G)$ we define $C(U) := \{\{u, (u, 1)\} : u \in U\}$.

Claim 3: A set of edges $C \subseteq E(H)$ is a feasible solution of \mathcal{I} if and only if $C = C(U)$ for an independent set U in G .

Proof of Claim 3: Let C be a feasible solution of \mathcal{I} . By Claim 2, C contains only edges of length 1, so we have $C = C(U)$ for some set of vertices U in G . Suppose that two such vertices $u, v \in U$ are connected by an edge, then we would have $\text{dist}_\ell(u, v) = 4$ and $\text{dist}_{\ell_C}(u, v) = 2$, violating the condition (1) for the given tolerance function. It follows that U is an independent set.

To prove the other direction of the equivalence, let U be an independent set in G and consider the set of edges $C(U)$ in H . To verify that C is a weak $(3/2, 0)$ -contraction, it suffices to check condition (1) between the end vertices of paths on two edges, one of length 1 from C and the other of length 2, and for paths on k edges that start and end with an edge of length 1 from C . In the first case the contraction $C(U)$ changes the distance from 3 to 2, which is compatible with (1). In the second case the contraction $C(U)$ changes the distance from $2k - 2$ to $2k - 4$, which is also compatible with (1), where we use that $k \geq 4$ because of the assumption that U is an independent set. \square

Claim 3 implies that \mathcal{I} has a feasible solution with k edges if and only if G has an independent set of size k . As $n(H) = 3n(G) = \mathcal{O}(n(G))$, the theorem follows from the [Zuc07] result. \blacksquare

7. ASYMPTOTIC BOUNDS

In this section we show how to compute contractions for graphs that are not optimal (i.e., they do not maximize the objective function (2)), but that can be computed efficiently despite our hardness results presented in the previous sections. The main results of this section are Theorem 13 and the corresponding (not tight) lower bound (Theorem 15) that is conditional on the validity of Erdős' girth conjecture. We also prove that a contraction significantly reduces the number of vertices in dense input graphs (Theorem 17), and provide a corresponding matching lower bound (Theorem 18).

Throughout this section, we assume all graphs to have unit length edges $\ell = 1$.

Theorem 13. *Let $k \geq 1$ be a real number. Any graph G has a $(2k - 1, 1)$ -contraction C such that the contracted graph G/C has at most $n^{1+1/k}$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$.*

In Theorem 13 and all following statements and proofs, n and m refer to the number of vertices and edges of the input graph G , not of the contracted graph G/C .

Setting $k := \log_2 n$ in Theorem 13 yields the following corollary.

Corollary 14. *Any graph G has a $(2 \log_2 n - 1, 1)$ -contraction C such that the contracted graph G/C has at most $2n$ edges, and such a contraction can be computed in time $\mathcal{O}(m)$.*

To prove Theorem 13, we use a clustering approach as presented in [Awe85]. Specifically, the following crucial lemma appears in a slightly weaker form in that paper. To state the lemma we introduce a few definitions. For any real number $r \geq 1$, we define an r -partition of a graph $G = (V, E)$ as a set of clusters $P_i \subseteq V$, $i \in [l]$, with corresponding cluster centers $p_i \in P_i$, where the P_i are required to form a partition of the vertex set V and where $\text{dist}_\ell(p_i, u) \leq r - 1$ for all $u \in P_i$ and $i \in [l]$. We denote the resulting r -partition by $P := \{(p_i, P_i) : i \in [l]\}$. We write $\rho(P)$ for the number of pairs $1 \leq i < j \leq l$ for which P_i and P_j are connected by at least one edge, and we refer this quantity as the *density* of P .

Lemma 7.1. *Let $r \geq 1$ be a real number. Any graph G has an r -partition P with density $\rho(P) \leq n^{1+1/r}$, and such a partition can be computed in time $\mathcal{O}(m)$.*

Proof. The idea of the algorithm is to build an r -partition P of G iteratively in rounds. In each round, we build a new cluster and remove all vertices from that cluster from the graph, processing the subgraph on the remaining vertices in the next round. The algorithm proceeds until all vertices are assigned to a cluster. In round i , we choose an arbitrary vertex p_i as a cluster center, and define layers $L_{i,0}, L_{i,1}, \dots$ around the vertex p_i , where the layer $L_{i,j}$ consists of all vertices at distance exactly j from p_i (this distance is measured in the subgraph of G under consideration in this round). We continue computing these layers as long as the number of vertices in the new layer is at least the number of vertices in all previous layers times the factor $n^{1/r}$. The cluster P_i is defined as the union of all layers around p_i satisfying this expansion condition. We refer to the first layer violating this condition (which is *not* added to P_i anymore) as the *rejected layer*. We let P denote the partition of the vertices of G computed in this fashion.

To verify that P is indeed an r -partition, we proceed to show that each vertex within a cluster has distance at most $r - 1$ from the center vertex of that cluster, and that the density $\rho(P)$ of the partition is at most $n^{1+1/r}$. Intuitively, the expansion condition in the definition of the layers ensures that a cluster has few layers and that the number of edges that go to unclustered vertices is small.

Consider a cluster P_i with center vertex p_i and the layers $L_{i,0}, L_{i,1}, \dots, L_{i,d}$. Suppose for the sake of contradiction that $d \geq r$. By the definition of the layers in the algorithm we know that $|L_{i,j}| \geq n^{1/r} \sum_{k=0}^{j-1} |L_{i,k}|$ holds for all $j \in [d]$, implying that $|L_{i,j}| \geq n^{j/r}$. Consequently, the size of the cluster satisfies $|P_i| = \sum_{j=0}^d |L_{i,j}| \geq 1 + n^{r/r} = n + 1$, a contradiction.

We now show that $\rho(P) \leq n^{1+1/r}$. The key idea is that the number of vertices in the rejected layer of a cluster P_i is at most $n^{1/r} |P_i|$. Thus the number of edges from P_i to clusters that are created later is at most $n^{1/r} |P_i|$. For every edge between two clusters we let the cluster that is created first account for that edge. Summing over all these edges between clusters yields the desired upper bound of $\rho(P) \leq n \cdot n^{1/r} = n^{1+1/r}$.

Using breadth-first search, the partitioning algorithm described above runs in time $\mathcal{O}(m)$ (recall that G is assumed to be connected). This completes the proof of the lemma. ■

With Lemma 7.1 in hand, we are now ready to prove Theorem 13.

Proof of Theorem 13. Given $G = (V, E)$, we first compute a k -partition P into l clusters as described by Lemma 7.1. We define the set C of contracted edges as the union of all

edges within the clusters, $C := \{\{u, v\} \in E : u, v \in P_i \text{ for some } i \in [l]\}$. We thus contract each cluster into a single vertex and remove from every set of resulting parallel edges all but a single edge.

We proceed to show that C is a $(2k-1, 1)$ -contraction, i.e., we show that $\text{dist}_{\ell_C}(u, v) \geq \text{dist}_{\ell}(u, v)/(2k-1) - 1$ for all $u, v \in V$. Consider two vertices $u \in P_i$ and $v \in P_j$, where i and j might be equal. Let $Q_{u,v}$ be the shortest path from u to v in G with edge lengths ℓ_C (all edges from C receive length zero). The length d of $Q_{u,v}$ is the number of edges on that path that connect different clusters. Note that $Q_{u,v}$ enters and leaves each of the $d+1$ visited clusters at most once, using at most $2k-2$ edges in every cluster, so in G (where all edges have unit lengths) we get $\text{dist}_{\ell}(u, v) \leq d + (d+1)(2k-2)$.

Combining these observations we obtain

$$\text{dist}_{\ell_C}(u, v) = d \geq d - \frac{1}{2k-1} = \frac{d + (d+1)(2k-2)}{2k-1} - 1 \geq \frac{\text{dist}_{\ell}(u, v)}{2k-1} - 1,$$

proving the claim.

It remains to show that the contracted graph G/C has at most $n^{1+1/k}$ edges, which is an immediate consequence of the upper bound $m(G/C) = \rho(P) \leq n^{1+1/k}$ given by Lemma 7.1. This completes the proof of the theorem. \blacksquare

Using Erdős' girth conjecture we can deduce a nearly matching lower bound on the size of a contraction. Erdős' girth conjecture [Erd64] asserts that there exist graphs with $\Omega(n^{1+1/k})$ edges and girth $2k+1$, and this conjecture has been verified for $k=1, 2, 3, 5$ [Wen91].

Theorem 15. *Assuming Erdős' girth conjecture, there exists for any integer $k \geq 2$ a graph G such that any $(k-1, 1)$ -contraction C results in a graph G/C with $\Omega(n^{1+1/k})$ edges.*

Proof. For a given integer $k \geq 2$ let G be a graph that is guaranteed by Erdős' girth conjecture, i.e., G has girth $2k+1$ and $\Omega(n^{1+1/k})$ edges. Consider any $(k-1, 1)$ -contraction C on G , and consider a connected component of the graph (V, C) . Applying (1) shows that $\text{dist}_{\ell}(u, v) \leq k-1$ holds for any two vertices u and v in that component. Using that the girth of G is $2k+1$, it follows that for any cycle in G , the connected component of (V, C) does not contain a contiguous segment of cycle edges of length at least half of the cycle. This implies that all connected components of the graph (V, C) are trees with diameter at most $k-1$. Therefore, the total number of edges within all connected components of (V, C) is at most n . We will further argue that there is at most one edge between any two connected components. Suppose for the sake of contradiction that there are two components of (V, C) with two different edges connecting them, say $\{u, v\}$ and $\{u', v'\}$, where u and u' lie in the same connected component and v and v' in the other. As the diameter of each component is at most $k-1$, it follows that in G there is a path from u to u' of length at most $k-1$, and a path from v to v' of length at most $k-1$. Together with the two edges connecting the components we obtain a cycle of length at most $2(k-1) + 2 = 2k$, contradicting the assumption that G has girth $2k+1$.

Therefore, the resulting graph after the contraction has $\Omega(m) = \Omega(n^{1+1/k})$ edges. \blacksquare

We proceed by proving our asymptotic bounds for contractions with a purely additive error.

Theorem 16. *Any graph G has a $(1, 2k)$ -contraction C for any integer $0 \leq k \leq n/2$ with objective value $\Phi(C) \geq km/n$ that can be computed in time $\mathcal{O}(m)$.*

This result implies that the number of edges in G/C is at most $m - km/n$. If G is a path, no $(1, 2k)$ -contraction has an objective value greater than $2k$, and $km/n = k(1 - 1/n)$,

showing that the objective value in Theorem 16 can be improved by at most a factor of two.

Proof. Let U be the set of k vertices in G of highest degree. Then we have

$$\sum_{u \in U} \deg(u) \geq k/n \sum_{v \in V} \deg(v) = k/n \cdot 2m = 2km/n.$$

Let C be the set of edges incident to any vertex in U . As each edge is incident to at most two vertices in U , we get $|C| \geq 1/2 \sum_{u \in U} \deg(u) \geq km/n$ from the previous inequality. As no shortest path visits a vertex in U twice, C is indeed a $(1, 2k)$ -contraction. The set C can be computed as follows: We first compute the degrees of all vertices in time $\mathcal{O}(m)$, then find the k -th largest element in this list in time $\mathcal{O}(n)$, and by another linear time sweep over this list we select k vertices of highest degree. Overall, the required time is $\mathcal{O}(m)$. ■

All of the results above show that contractions can be effectively used to reduce the number of edges in a dense graph. But one possible advantage of using contraction instead of spanners is that it also has the potential to reduce the number of *vertices* in the graph. Unfortunately, for constant approximation errors, it is not possible to guarantee more than a constant-factor reduction in general graphs: it is not hard to see that given a path on n vertices, any $(k, 1)$ -contraction will still result in at least n/k vertices. The same problem applies to general dense graphs, since they could still contain a long path within them. That being said, it seems likely that in practice contraction can lead to significant vertex reduction in many dense graphs. We ground this practical intuition with the following theoretical result for the special case of graphs with large minimum degree.

Theorem 17. *Let D be an integer. Any graph G with minimum degree at least D has a $(5, 1)$ -contraction C such that the contracted graph G/C has at most n/D vertices, and such a contraction can be computed in time $\mathcal{O}(m)$.*

Proof. Recall the definition of an r -partition. For a cluster P_i with center vertex p_i we refer to r as the *radius* of that cluster. This is the maximum distance of all cluster vertices from p_i .

We will show how to construct a 3-partition in which the number of clusters P_i is at most n/D . Using the exact same argument as in the proof of Theorem 13, such a 3-partition yields the desired $(5, 1)$ -contraction. Our construction first builds clusters of radius 1, and then extends them to clusters of radius 2. The clustering with radius 1 proceeds very similarly as in the proof of Lemma 7.1 before with $r = 1$. The crucial difference is that we choose as center vertices only vertices with degree at least D . If no such vertices are left, the clustering process terminates, and the remaining unclustered vertices have degree strictly less than D . It is easy to see that since those vertices have degree at least D in the original graph, they must be adjacent to a vertex in a radius 1 cluster. We can thus assign each of those vertices to such a cluster arbitrarily, yielding a clustering of all vertices of G with radius 2.

The number of clusters is at most n/D because by construction every cluster contains at least D vertices. This shows that the number of vertices in the contracted graph is at most n/D .

This algorithm can be implemented in time $\mathcal{O}(m)$ by using an adjacency list representation where we keep track of degree information after removing an edge from the graph. ■

To see that we cannot guarantee less than n/D vertices, even with larger approximation error, consider the graph G that consists of n/D isolated D -cliques. We now show that

even if G is connected, we cannot guarantee $o(n/D)$ vertices in the contracted graph, even if we allow a larger (constant) approximation error.

Theorem 18. *Let D and k be integers. There exists a graph G with minimum degree D such that any $(k, 1)$ -contraction C results in a graph G/C with $\Omega(n/(kD))$ vertices.*

Proof. Assume for simplicity that n is divisible by D . We construct the graph G as follows. We partition the n vertices into n/D layers, with each layer containing exactly D vertices. For $1 \leq i < n/D$, all vertices in layer i receive an edge to all vertices in layer $i + 1$. Clearly all vertices in the resulting graph have degree at least D . Let u and v be two vertices in layers i and j , respectively. Then clearly we have $\text{dist}_\ell(u, v) \geq |j - i|$. Now let C be any $(k, 1)$ -contraction on G , and consider the connected components of the graph (V, C) . Applying (1) shows that $\text{dist}_\ell(u, v) \leq k$ holds for any two vertices u and v in the same component. Combining these two inequalities shows that every connected component contains vertices from at most k layers. As there are n/D layers, the contracted graph has at least $n/(kD)$ vertices. ■

ACKNOWLEDGEMENTS

We thank Martin Skutella for stimulating discussions about the problems treated in this paper.

REFERENCES

- [AB16] A. Abboud and G. Bodwin. The $4/3$ additive spanner exponent is tight. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 351–361. ACM, New York, 2016.
- [ACIM99] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.
- [ADD⁺93] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9(1):81–100, 1993.
- [Awe85] B. Awerbuch. Complexity of network synchronization. *J. Assoc. Comput. Mach.*, 32(4):804–823, 1985.
- [BBV00] T. C. Biedl, B. Brejová, and T. Vinař. Simplifying flow networks. In *Mathematical foundations of computer science 2000 (Bratislava)*, volume 1893 of *Lecture Notes in Comput. Sci.*, pages 192–201. Springer, Berlin, 2000.
- [BKMP05] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie. New constructions of (α, β) -spanners and purely additive spanners. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 672–681. ACM, New York, 2005.
- [BMSW13] D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner, editors. *Graph partitioning and graph clustering*, volume 588 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI; Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), Piscataway, NJ, 2013. Papers from the 10th DIMACS Implementation Challenge Workshop held at the Georgia Institute of Technology, Atlanta, GA, February 13–14, 2012.
- [BOR80] J. J. Bartholdi, III, J. B. Orlin, and H. D. Ratliff. Cyclic scheduling via integer programs with circular ones. *Oper. Res.*, 28(5):1074–1085, 1980.
- [BS03] S. Baswana and S. Sen. A simple linear time algorithm for computing a $(2k - 1)$ -spanner of $O(n^{1+1/k})$ size in weighted graphs. In *Automata, languages and programming*, volume 2719 of *Lecture Notes in Comput. Sci.*, pages 384–396. Springer, Berlin, 2003.
- [CE06] D. Coppersmith and M. Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discrete Math.*, 20(2):463–501, 2006.
- [CGH16] Y. K. Cheung, G. Goranci, and M. Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 131:1–131:14, 2016.
- [CXKR06] T.-H. H. Chan, D. Xia, G. Konjevod, and A. Richa. A tight lower bound for the Steiner point removal problem on trees. In *Approximation, randomization and combinatorial optimization*, volume 4110 of *Lecture Notes in Comput. Sci.*, pages 70–81. Springer, Berlin, 2006.

- [DB13] F. Dörfler and F. Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Trans. Circuits Syst. I. Regul. Pap.*, 60(1):150–163, 2013.
- [dBK12] M. de Berg and A. Khosravi. Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geometry Appl.*, 22(3):187–206, 2012.
- [EP07] Michael Elkin and David Peleg. The hardness of approximating spanner problems. *Theor. Comp. Sys.*, 41(4):691–729, December 2007.
- [Erd64] P. Erdős. Extremal problems in graph theory. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36. Publ. House Czechoslovak Acad. Sci., Prague, 1964.
- [Gup01] A. Gupta. Steiner points in tree metrics don’t (really) help. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 220–227. SIAM, Philadelphia, PA, 2001.
- [HKBG08] C. Hübler, H.-P. Kriegel, K. M. Borgwardt, and Z. Ghahramani. Metropolis algorithms for representative subgraph sampling. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 283–292, 2008.
- [Kar72] R. M. Karp. Reducibility among combinatorial problems. pages 85–103, 1972.
- [KNZ14] R. Krauthgamer, H. L. Nguyen, and T. Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014.
- [Kor98] Guy Kortsarz. On the hardness of approximation spanners. In *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX ’98*, pages 135–146, London, UK, UK, 1998. Springer-Verlag.
- [LS93] A. L. Liestman and T. C. Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993.
- [MC05] E. Misiołek and D. Z. Chen. Efficient algorithms for simplifying flow networks. In *Computing and combinatorics*, volume 3595 of *Lecture Notes in Comput. Sci.*, pages 737–746. Springer, Berlin, 2005.
- [PS89] D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.
- [Ski08] S. Skiena. *The Algorithm Design Manual (2. ed.)*. Springer, 2008.
- [Wen91] R. Wenger. Extremal graphs with no C^4 ’s, C^6 ’s, or C^{10} ’s. *J. Combin. Theory Ser. B*, 52(1):113–116, 1991.
- [ZMT10] F. Zhou, S. Mahler, and H. Toivonen. Network simplification with minimal loss of connectivity. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 659–668, 2010.
- [Zuc07] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3:103–128, 2007.