Distance-Preserving Graph Contractions*

Aaron Bernstein[1], Karl Däubel[2], Yann Disser[†3], Max Klimm[4], Torsten Mütze[5], and Frieder Smolny[6]

1 **Institut für Mathematik, TU Berlin, Germany**
`bernstein@math.tu-berlin.de`
2 **Institut für Mathematik, TU Berlin, Germany**
`daeubel@math.tu-berlin.de`
3 **Department of Mathematics, Graduate School CE, TU Darmstadt, Germany**
`disser@mathematik.tu-darmstadt.de`
4 **Wirtschaftswissenschaftliche Fakultät, HU Berlin, Germany**
`max.klimm@hu-berlin.de`
5 **Institut für Mathematik, TU Berlin, Germany**
`muetze@math.tu-berlin.de`
6 **Institut für Mathematik, TU Berlin, Germany**
`smolny@math.tu-berlin.de`

Abstract

Compression and sparsification algorithms are frequently applied in a preprocessing 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 analyze the asymptotic behavior of the size of contractions, and find efficient algorithms to compute (non-optimal) contractions despite our hardness results.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases distance oracle, contraction, spanner

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.51

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

* A preprint version of this paper with full proofs is available at `http://arxiv.org/abs/1705.04544`

† Supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.

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. [19, 3, 6, 1]). Given an undirected graph $G = (V, E)$ and real 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 [5], 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 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 this 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, while entailing a multiplicative distance distortion of similar magnitude. 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.

In order to measure the distance distortion of the contraction, we assume a non-decreasing tolerance function $\varphi\colon \mathbb{R} \to \mathbb{R}$, similar to the corresponding function for spanners, see e.g. [6]. 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 $G = (V, E)$

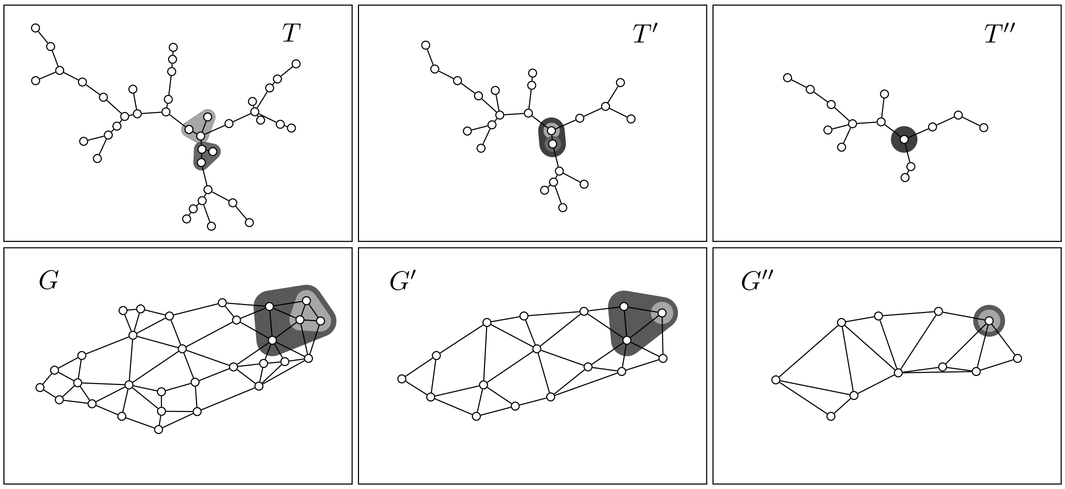

Figure 1 Top: two iterations of Contraction with $\varphi(x) = 4x/5 - 3$ on a tree; bottom: two iterations of Contraction with $\varphi(x) = 3x/4 - 3$ on a planar graph. Distances are geometric and some contracted sets of vertices are highlighted.

with edge lengths $\ell\colon E \to \mathbb{R}_{>0}$ 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 1 Overview of algorithmic and hardness results presented in this paper.

Problem	Graph classes: Path	Tree	Cycle	General
CONTRACTION				
addit. (α=1), unit lg.	$\mathcal{O}(n)$ [Th. 1 1]	$\mathcal{O}(n)$ [Th. 1 3]	$\mathcal{O}(n)$ [Th. 1 2]	$m^{\frac{1}{2}-\varepsilon}$-inapx.[a] [Th. 8]
affine (α, β), unit lg.				
addit. (α=1)	$\mathcal{O}(n^3)$ [Th. 2]		NP-hard [Th. 6]	$n^{1-\varepsilon}$-inapx. [Th. 7]
affine (α, β)				
WEAK CONTRACTION				
additive (α=1)	$\mathcal{O}(n^5)$ [Th. 4]		NP-hard[b] [Th. 6]	
affine (α, β)				$n^{1-\varepsilon}$-inapx.[c] [Th. 10]

[a] even for bipartite graphs and $\beta = 1$
[b] also NP-hard for planar graphs with arb. large girth, $(\alpha, \beta) = (2, 0)$, and unit lg. ($\ell = 1$) [Th. 9].
[c] even if $(\alpha, \beta) = (3/2, 0)$.

Algorithmic results

We develop linear time greedy algorithms for CONTRACTION with unit lengths on paths, cycles, and on trees with $\alpha = 1$ (Theorem 1). 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 2 and 4). 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 6). 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 7 and Theorem 8).

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 9). 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 10).

Table 2 Overview of asympotic bounds presented in this paper.

CONTRACTION with unit lg. (ℓ=1)	# of edges in G/C	Time	Reference
$(\alpha,\beta)=(2k-1,1)$	$n^{1+1/k}$	$\mathcal{O}(m)$	[Th. 11]
$(\alpha,\beta)=(2\log_2 n-1,1)$	$2n$	$\mathcal{O}(m)$	[Cor. 12]
$(\alpha,\beta)=(k-1,1)$	$\Omega(n^{1+1/k})$	—	[Th. 14]
$(\alpha,\beta)=(1,k)$	$m-km/(2n)$	$\mathcal{O}(m)$	[Th. 15 1]
$(\alpha,\beta)=(1,k)$	$\mathcal{O}(n^2/k)$	$\mathcal{O}(m)$	[Th. 15 2]
$(\alpha,\beta)=(1,\mathcal{O}(1))$	$\Omega(n^{4/3-o(1)})$	—	[1]
CONTRACTION with unit lg. (ℓ=1) and min. degree D	# of vertices in G/C	Time	Reference
$(\alpha,\beta)=(5,1)$	n/D	$\mathcal{O}(m)$	[Th. 16]
$(\alpha,\beta)=(k,1)$	$\Omega(n/(kD))$	—	[Th. 17]

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 11) by successively growing clusters around center vertices. Assuming Erdős' girth conjecture, we show a corresponding (not tight) lower bound (Theorem 14).

For a purely additive error, we observe two simple $(1,k)$-contractions that can be computed in $\mathcal{O}(m)$ time (Theorem 15). We show that for any even integer $0 \leq k \leq n$, the edges incident to the $k/2$ vertices of highest degrees form a $(1,k)$-contraction with objective value at least $km/(2n)$, which is asymptotically best possible for paths. Another $(1,k)$-contraction C is implicitly used by Bernstein and Chechik in their faster deterministic algorithm for dynamic shortest paths in dense graphs [8]. For any number $0 < k \leq n$, it consists of the edges incident to two vertices of degree at least n/k, and G/C has $\mathcal{O}(n^2/k)$ edges. Both of these contractions can be computed in $\mathcal{O}(m)$ time. Further we note that the main result in [1] implies that for all $\varepsilon > 0$, any contraction C such that G/C has $\mathcal{O}(n^{4/3-\varepsilon})$ edges does not admit a constant additive error.

One possible advantage of contraction compared to spanners is the potentially significant reduction of *vertices* as well as edges, e.g. reducing the complexity of performing algorithmic tasks in the smaller graph. To ground this intuition, we exhibit a contraction that significantly reduces the number of vertices in any graph with minimum degree D to $\mathcal{O}(n/D)$ (Theorem 16). We also present a lower bound (Theorem 17) 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. Like contractions, spanners are NP-hard to compute optimally (see [19, 18]). While the spanner literature considers the problem of minimizing the number of remaining edges, we analyze the objective of maximizing the number of contracted edges, prohibiting a direct comparison of the respective inapproximability results. We note however that approximation algorithms for spanner problems have been studied extensively, even though strong lower bounds are known. For instance, computing (2, 0)-spanners in unweighted graphs is $\Theta(\log n)$-hard to approximate ([16, 15]), for further references see e.g. [11].

Despite these negative results, 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 for unweighted graphs has a clear analogy to the classic $(2k-1,0)$-spanner with the same number of edges [3] (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 [19] 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 [7] achieves randomized linear time. Meanwhile our $(2k-1,1)$-contraction can be constructed deterministically in linear time.

There are also spanner results that significantly sparsify unweighted graphs at the cost of a purely additive error, as a (1,2)-spanner with $\mathcal{O}(n^{3/2})$ edges [2], or a (1,6)-spanner with $\mathcal{O}(n^{4/3})$ edges [6]. We do not know if analogous results are possible in the contraction model. The incompressibility result in [1] mentioned above implies the same lower bound for spanners as for contractions and every other distance oracle with additive error: For every $\varepsilon > 0$ any spanner of size $\mathcal{O}(n^{4/3-\varepsilon})$ does not admit a constant additive error. Finally, for spanners there are results that combine multiplicative and additive error, such as the $(k, k-1)$-spanner of [6].

Gupta [14] 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. [9] 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. [17] 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. [10] 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.

1.3 Outline of this paper

In Section 2 we introduce important definitions and notations that will be used throughout this paper. In Sections 3–6 we formally state our results, in exactly the same order as they were discussed in Section 1.1 before. Due to the limited space in this extended abstract, we will only mention the main steps and ideas needed to prove a few selected theorems. Full proofs can be found in the preprint [12].

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 $\ell\colon E \to \mathbb{R}_{>0}$. The *distance* $\mathrm{dist}_\ell(u, v)$ 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 $\varphi\colon \mathbb{R} \to \mathbb{R}$. 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

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

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 $\mathrm{dist}_{\ell_C}(u, v) = 0$ 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

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

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.

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.

Note that our contraction model is well-behaved in the sense that successively solving (Weak) Contraction with general tolerance functions φ and ψ yields a feasible solution with respect to the composition $\psi \circ \varphi$ (see for a proof [12]).

3 Greedy algorithms

In this section we summarize our results on greedy algorithms that allow solving several special cases of the problem Contraction with affine tolerance function $\varphi(x) = x/\alpha - \beta$ in linear time. The proof of the three cases in Theorem 1 can be found in [12].

▶ **Theorem 1.** *We can solve* Contraction *in time $\mathcal{O}(n)$ in the following three cases:*
(i) *Paths with $\ell = 1$ and $\varphi(x) = x/\alpha - \beta$, $\alpha, \beta \geq 1$.*
(ii) *Cycles with $\ell = 1$ and $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1, \beta \geq 0$.*
(iii) *Trees with $\ell = 1$ and $\varphi(x) = x - \beta$, $\beta \geq 0$.*

Generalizing cases 1 and 3, in the next section we will present polynomial time algorithms for the general case on trees (with somewhat larger running times). In contrast to the algorithmic result for unit length cycles in case 2, we will see in Section 5 that Contraction is NP-hard on cycles with general edge lengths, even with $\alpha = 1$.

4 Dynamic programs for general trees

In this section we consider the problem of computing (weak) contractions for trees $T = (V, E)$ with affine tolerance function $\varphi(x) = x/\alpha - \beta$. Recall that on trees we have $f(C) = |C|$ for every (weak) contraction C. In the following we present the main steps of our dynamic programming approach for solving these problems, first for the problem Contraction and then for Weak Contraction. Full proofs are deferred to [12].

▶ **Theorem 2.** *We can solve* Contraction *on trees with $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$ and $\beta \geq 0$ in time $\mathcal{O}(n^3)$.*

The idea is to root the tree T at an arbitrary vertex, and to decompose the problem by splitting T into rooted subtrees at every vertex. Specifically, let v be a vertex of T, and T_1 and T_2 subtrees of T rooted at v that only have the vertex v in common. Now consider an optimal contraction C on T, and let C_1 and C_2 be the subsets of C on T_1 or T_2, respectively. Clearly, C_1 and C_2 are feasible contractions on their subtrees. Furthermore, the set C_2 has maximum size under the condition that its union with C_1 forms a feasible contraction (and vice versa).

We thus identified two quality parameters of solutions on rooted subtrees that we need to consider as possible parts of optimal contractions in T: One is their size, the other is whether they can be combined with other partial solutions in the rest of T, when growing subtrees towards the root. To quantify this seemingly unwieldy second parameter, we observe that a solution $C \subseteq E$ is feasible if and only if for any two vertices u and v of T we have $\mathrm{load}_{C,\alpha}(u, v) \leq \beta$, where the *load between u and v* is defined as

$$\mathrm{load}_{C,\alpha}(u, v) := \mathrm{dist}_\ell(u, v)/\alpha - \mathrm{dist}_{\ell_C}(u, v).$$

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

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

Note that $\mathrm{load}_{C,\alpha}(T, v) \geq 0$, as we have $\mathrm{load}_{C,\alpha}(v, v) = 0$. The following lemma justifies that this definition is the correct second quality parameter.

▶ **Lemma 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 $C \subseteq E$ 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 $\mathrm{load}_{C,\alpha}(T_1, v) + \mathrm{load}_{C,\alpha}(T_2, v) \leq \beta$.*

Our strategy is to recursively find all contractions on subtrees, that for some fixed size between 1 and n minimize the load. To this end, we choose an arbitrary root vertex r of T, and starts by considering rooted subtrees consisting of single leaves. We then grows these subtrees towards the root r using two operations: Either two subtrees T_1 and T_2 with the same root v as before are joined (keeping the root v), or a subtree T' containing all successors of its root v in T is extended by adding the edge that leads from v to its parent vertex u in T (in this case, u becomes the new root). Let T^* be the resulting joined or extended subtree arising from the respective operation, and let C be any contraction on T^*. In case of a join-operation we have

$$\mathrm{load}_{C,\alpha}(T^*, v) = \max\{\mathrm{load}_{C,\alpha}(T_1, v), \mathrm{load}_{C,\alpha}(T_2, v)\}, \tag{4a}$$

and the size of C is simply the sum of the sizes of the subsets of C on T_1 and T_2. In case of an extend-operation we have

$$\mathrm{load}_{C,\alpha}(T^*, v) = \begin{cases} \max\{\mathrm{load}_{C,\alpha}(T', u) + \ell(v, u)/\alpha, & \text{if } \{u, v\} \in C, \\ \max\{\mathrm{load}_{C,\alpha}(T', u) - (1 - 1/\alpha)\ell(v, u), 0\}, & \text{otherwise,} \end{cases} \tag{4b}$$

and the size of C is either equal to the size of the subset of C on T' in the second case, or one more in the first case.

These formulas indicate a monotone behavior of our two parameters, which allows us to compute the necessary partial solutions on T^* by combining the previously computed partial solutions of its subtrees. Furthermore they allow us to compute our parameters for the combined sets.

This yields the dynamic programming algorithm for Contraction referred to in Theorem 2. A similar approach also works for the problem Weak Contraction.

▶ **Theorem 4.** *We can solve* Weak Contraction *on trees with $\varphi(x) = x/\alpha - \beta$, $\alpha \geq 1$ and $\beta \geq 0$ in time $\mathcal{O}(n^5)$.*

Here, our task is complicated by the fact that the combinability of solutions on subtrees cannot be captured by one single parameter. As we need to keep track of pairs of vertices whose distances remain positive when contracting a set of edges $C \subseteq E$, we define the *weak load* of a rooted tree T at one of its vertices v by

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

allowing us to formulate the following combinability criterion analogous to Lemma 3 from before.

▶ **Lemma 5.** *Let T, T_1, T_2 and v be as in Lemma 3. 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*

$$\mathrm{load}_{C,\alpha}(T_1, v) + \mathrm{wload}_{C,\alpha}(T_2, v) \leq \beta \quad \text{and} \quad \mathrm{wload}_{C,\alpha}(T_1, v) + \mathrm{load}_{C,\alpha}(T_2, v) \leq \beta. \tag{5}$$

We now proceed similarly by computing sets of solutions on rooted subtrees of T that are optimal with respect to the three parameters size, load and weak load. In particular, for any fixed size we compute a Pareto front of subsolutions of that size, minimizing both load and weak load. The key step for getting an efficient algorithm is to prove that these Pareto fronts have polynomial, in fact even linear, size (this is not clear a priori, as the number of feasible solutions on subtrees can be exponential). Using that the weak load has similar monotonicity properties and recursive formulas as stated in (4) for the load, we thus arrive at an efficient dynamic program. As our algorithm computes $\mathcal{O}(n^2)$ Pareto fronts of size $\mathcal{O}(n)$ at every vertex, and we can combine optimal solutions from two such fronts in time $\mathcal{O}(n)$, we get an additional factor of n^2 in the running time compared to our first dynamic program, giving an overall running time of $\mathcal{O}(n^5)$.

5 Hardness and inapproximability

In this section we state our NP-hardness and inapproximability results for the problems Contraction and Weak Contraction. All proofs throughout this section can be found in [12].

We start by considering the purely additive case, where $\alpha = 1$. Recall that we can compute maximum size (weak) contractions in polynomial time on trees with arbitrary edge lengths (Theorem 2), and on cycles with unit length edges (Theorem 12). In contrast to that, our next result asserts that both problems are NP-hard on cycles with arbitrary edge lengths, even with $\alpha = 1$.

▶ **Theorem 6.** *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.*

It proceeds by a reduction from a variant of the Partition problem. Via inapproximability of the Clique problem (see [21]), we extend this result in the following two ways:

▶ **Theorem 7.** *For all $\beta, \varepsilon > 0$ it is NP-hard to approximate the problem* Contraction *with $\varphi = x - \beta$, $\beta \geq 0$ to within a factor of $n^{1-\varepsilon}$.*

▶ **Theorem 8.** *For all $\varepsilon > 0$ it is NP-hard to approximate* Contraction *with $\varphi = x - 1$ on bipartite graphs with unit lengths ($\ell = 1$) to within a factor of $m^{1/2-\varepsilon}$.*

The next two theorems capture our results for the purely multiplicative case, where $\beta = 0$ (recall that Contraction is trivial in this case). To state the first result, recall that the *girth* of a graph is the length of the shortest cycle.

▶ **Theorem 9.** *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 $\ell = 1$.*

The proof of Theorem 9 uses a reduction from a variant of Planar 3SAT.

▶ **Theorem 10.** *For all $\varepsilon > 0$ it is NP-hard to approximate* Weak Contraction *with $\varphi = 2x/3$ to within a factor of $n^{1-\varepsilon}$.*

The proof of Theorem 10 proceeds via a reduction from Independent Set.

6 Asymptotic bounds

In this section we show how to compute contractions for graphs that are not optimal, but can be computed efficiently despite our hardness results from the previous section. In this vein, the main results of this section are Theorem 11 and the corresponding (not tight) lower bound (Theorem 14). Further we consider the factor by which a contraction can reduce the number of vertices (Theorem 16 and Theorem 17). Throughout this section, we assume all graphs to have unit length edges $\ell = 1$.

▶ **Theorem 11.** *Let $k \geq 1$ be a real number. Any graph G with unit length edges 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 $\mathcal{O}(m)$.*

Recall that here and throughout, n and m denote 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 11 yields the following corollary.

▶ **Corollary 12.** *Any graph G with unit length edges 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 11, we use a clustering approach as presented in [4], yielding the next lemma. 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 $\mathrm{dist}_\ell(p_i, u) \leq r - 1$ 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 13.** *Let $r \geq 1$ be a real number. Any graph G with unit length edges has an r-partition P with density $\rho(P) \leq n^{1+1/r}$, and such a partition can be computed in time $\mathcal{O}(m)$.*

For the proof of Lemma 13 we refer the reader to [12]. With Lemma 13 in hand, we are now ready to prove Theorem 11.

Proof of Theorem 11. Given $G = (V, E)$, we first compute a k-partition P into l clusters as described by Lemma 13. 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 $\mathrm{dist}_{\ell_C}(u, v) \geq \mathrm{dist}_\ell(u, v)/(2k - 1) - 1$ 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 $\mathrm{dist}_\ell(u, v) \leq d + (d + 1)(2k - 2)$.

Combining these observations we obtain

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

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 13. This completes the proof of the theorem. ◀

Erdős' girth conjecture [13] asserts that there exist graphs with $\Omega(n^{1+1/k})$ edges and girth $2k+1$. It has been verified for $k = 1, 2, 3, 5$ [20] and the strongest spanner lower bounds depend on it. We use the conjecture to derive the following (not tight) lower bound. For the proof we refer to [12].

▶ **Theorem 14.** *Assuming Erdős' girth conjecture, there exists for any integer $k \geq 2$ a graph G such that any $(k - 1, 1)$-contraction of G results in a graph G/C with $\Omega(n^{1+1/k})$ edges.*

Turning to the case of a purely additive error, we observe two simple $(1, k)$-contractions.

▶ **Theorem 15.** *Let G be a graph with unit length edges.*

(i) *For any even integer $0 \le k \le n$, the set of edges incident to the $k/2$ vertices of highest degrees is a $(1, k)$-contraction C in G with $\Phi(C) \ge km/(2n)$.*

(ii) *For any real number $0 < k \le n$, the set of edges incident to two vertices of degree at least n/k is a $(1, k)$-contraction C in G such that G/C has $\mathcal{O}(n^2/k)$ edges.*

These contractions can be computed in time $\mathcal{O}(m)$.

As mentioned in the introduction, Bernstein and Chechik used the contraction in Theorem 15 2 in their dynamic shortest paths algorithm [8].

Note that the information theoretic lower bound in [1] implies that for all $\varepsilon > 0$, any contraction C such that G/C has $\mathcal{O}(n^{4/3-\varepsilon})$ edges does not admit a constant additive error.

In contrast to spanners, contractions also reduce the number of vertices. Unfortunately, for constant distortion it is impossible to guarantee more than a constant-factor reduction in this parameter, as the example of a path shows. 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 contractions can lead to significant vertex reductions in many dense graphs. We ground this practical intuition with a theoretical result for the special case of graphs with large minimum degree.

▶ **Theorem 16.** *Let D be an integer. Any graph G with unit length edges and 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 $\mathcal{O}(m)$.*

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 17.** *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.*

The proofs of the two previous theorems are deferred to [12].

Acknowledgements. We thank Martin Skutella for stimulating discussions about the problems treated in this paper. We also thank the anonymous reviewers for their helpful comments and suggestions.

References

1 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.

2 Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999. doi:10.1137/S0097539796303421.

3 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993. doi:10.1007/BF02189308.

4 Baruch Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985. doi:10.1145/4221.4227.

5 David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, editors. *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop, Georgia Institute of Technology, Atlanta, GA, USA, February 13-14, 2012. Proceedings*, volume 588 of *Contemporary Mathematics*. American Mathematical Society, 2013. doi:10.1090/conm/588.

6 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.

7 Surender Baswana and Sandeep Sen. A simple linear time algorithm for computing a (2k-1)-spanner of $o(n^{1+1/k})$ size in weighted graphs. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 384–296. Springer, 2003. doi:10.1007/3-540-45061-0_32.

8 Aaron Bernstein and Shiri Chechik. Deterministic decremental single source shortest paths: beyond the o(mn) bound. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 389–397. ACM, 2016. doi:10.1145/2897518.2897521.

9 Hubert T.-H. Chan, Donglin Xia, Goran Konjevod, and Andréa W. Richa. A tight lower bound for the steiner point removal problem on trees. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, volume 4110 of *Lecture Notes in Computer Science*, pages 70–81. Springer, 2006. doi:10.1007/11830924_9.

10 Yun Kuen Cheung, Gramoz Goranci, and Monika Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 131:1–131:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.131.

11 E. Chlamtáč, M. Dinitz, G. Kortsarz, and B. Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 534–553, Philadelphia, PA, USA, 2017. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=3039686.3039720.

12 Karl Däubel, Yann Disser, Max Klimm, Torsten Mütze, and Frieder Smolny. Distance-preserving graph contractions. *CoRR*, abs/1705.04544, 2017. arXiv:1705.04544.

13 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.

14 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.

15 G. 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. URL: http://dl.acm.org/citation.cfm?id=646687.756924.

16 Guy Kortsarz and David Peleg. Generating sparse 2-spanners. *J. Algorithms*, 17(2):222–236, 1994. doi:10.1006/jagm.1994.1032.

17 Robert Krauthgamer, Huy L. Nguyen, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014. doi:10.1137/120888843.

18 Arthur L. Liestman and Thomas C. Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993. doi:10.1002/net.3230230417.

19 David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. doi:10.1002/jgt.3190130114.

20 Rephael Wenger. Extremal graphs with no C^4's, C^6's, or C^{10}'s. *J. Comb. Theory, Ser. B*, 52(1):113–116, 1991. doi:10.1016/0095-8956(91)90097-4.

21 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007. doi:10.4086/toc.2007.v003a006.