FPT Algorithms for Embedding into Low Complexity Graphic Metrics

Arijit Ghosh
The Institute of Mathematical Sciences, HBNI, Chennai, India
arijitiitkgpster@gmail.com

Sudeshna Kolay
Eindhoven University of Technology, Eindhoven, Netherlands
s.kolay@tue.nl

Gopinath Mishra
Indian Statistical Institute, Kolkata, India
gopianjan117@gmail.com

Abstract

The Metric Embedding problem takes as input two metric spaces (X, D_X) and (Y, D_Y), and a positive integer d. The objective is to determine whether there is an embedding $F : X \to Y$ such that the distortion $d_F \leq d$. Such an embedding is called a *distortion d embedding*. In parameterized complexity, the Metric Embedding problem is known to be W-hard and therefore, not expected to have an FPT algorithm. In this paper, we consider the Gen-Graph Metric Embedding problem, where the two metric spaces are graph metrics. We explore the extent of tractability of the problem in the parameterized complexity setting. We determine whether an unweighted graph metric (G, D_G) can be embedded, or bijectively embedded, into another unweighted graph metric (H, D_H), where the graph H has low structural complexity. For example, H is a cycle, or H has bounded treewidth or bounded connected treewidth. The parameters for the algorithms are chosen from the upper bound d on distortion, bound Δ on the maximum degree of H, treewidth α of H, and the connected treewidth α_c of H.

Our general approach to these problems can be summarized as trying to understand the behavior of the shortest paths in G under a low distortion embedding into H, and the structural relation the mapping of these paths has to shortest paths in H.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Metric spaces, metric embedding, FPT, dynamic programming

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.35

Related Version A full version of the paper is available at [8], `https://arxiv.org/abs/1801.03253`.

1 Introduction

Given metric spaces (X, D_X) and (Y, D_Y), an *embedding* $F : X \to Y$ is an injective mapping from X to Y. The *expansion* e_F and *contraction* c_F of F are defined as $e_F = \max_{x_1,x_2(\neq x_1)\in X} \frac{D_Y(F(x_1),F(x_2))}{D_X(x_1,x_2)}$ and $c_F = \max_{x_1,x_2(\neq x_1)\in X} \frac{D_X(x_1,x_2)}{D_Y(F(x_1),F(x_2))}$, respectively. The *distortion* $d_F = e_F \cdot c_F$. Observe that $d_F \geq 1$. An embedding $F : X \to Y$ is *non-contracting* if $c_F \leq 1$.

The problem of low distortion embedding of a metric space into a simple metric space has been extensively studied in Mathematics and Computer Science (see [1, 11, 13, 14, 15]). Low distortion embedding algorithms have also found wide applications in other problems like

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 35; pp. 35:1–35:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Sparsest Cut, Nearest Neighbor Search, Clustering, Multicommodity Flow, Multicut, Small Balanced Separators (see [1, 9, 10, 13, 15]).

The need to obtain small distortion embeddings into simpler spaces naturally led to the question of finding a minimum distortion embedding of (X, D_X) into (Y, D_Y) when both the metric spaces have shortest path metrics on graphs with positive weights, and (Y, D_Y) has a simple topology as in paths, cycles, trees etc. Kenyon et al. [12] showed that this problem is APX-hard even when both the graphs are unweighted, have the same number of vertices, and one of the graphs is a simple wheel graph. Badoiu et al. [3] also proved APX-hardness when both the graphs are unweighted and (Y, D_Y) is the metric space of a path. Badoiu et al. [2] showed that computing the minimum distortion is hard to approximate up to a factor polynomial in $|X|$, even when (X, D_X) is a weighted tree with polynomial spread and (Y, D_Y) is a path. Fellows et al. [7] showed that the problem of embedding a weighted graph metric into a path with distortion at most $d > 2$ is NP-complete.

Badoiu et al. [3] gave the first algorithm for deciding if an unweighted graph metric has a non-contracting embedding into a path with distortion d. The running time of their algorithm was $n^{4d+2} \cdot d^{O(1)}$, where n denotes the number of vertices in the graph. Fellows et al. [7] gave the first fixed parameter tractable(FPT) algorithm with running time $O\left(n\, d^4 (2d+1)^{2d}\right)$ for finding a non-contracting embedding of an n vertex unweighted graph metric into a path with distortion at most d (d is the parameter of the algorithm). They also showed that their FPT algorithm can be extended to get an FPT algorithm for the case of non-contracting embeddings of weighted graphs into paths, where the parameters to the algorithm are both the distortion and the maximum weight of an edge in the graph. Nayyeri et al. [16] gave improved exact algorithms for embedding weighted path metrics into weighted paths.

Kenyon et al. [12] gave the first FPT algorithm for finding a bijective embedding f of an unweighted graph metric on n vertices into a tree with maximum degree bounded by Δ in $O(n^2 \cdot 2^{\Delta^{\mu^3}})$ time, where $\mu = \max\{e_f,\, c_f\}$. Fellows et al. [7] extended this result to give an algorithm for the problem of finding a non-contracting embedding of unweighted graphs into bounded degree trees with distortion at most d in $O(n^2 \cdot |V(T)|) \cdot 2^{O((5d)^{\Delta^{d+1}} \cdot d)}$ time, where $V(T)$ denotes the vertex set of the tree and where the maximum degree in T is bounded by Δ. In a follow-up paper, Nayyeri et al. [17] gave the first $(1+\epsilon)$-approximation algorithm to embed weighted graphs with spread Σ into graphs on m vertices with bounded treewidth α and doubling dimension λ in $m^{O(1)} \cdot n^{O(\alpha)} \cdot (d_{opt}\Sigma)^{\alpha \cdot (1/\epsilon)^{\lambda+2} \cdot \lambda \cdot (O(d_{opt}))^{2\lambda}}$ time, where d_{opt} denotes the minimum distortion.

Our Contributions. In this paper, we further investigate the problem of embedding a general graph metric (G, D_G) into a low complexity graph metric (H, D_H) with distortion at most d. We will denote by n and N the number of vertices in graphs G and H, respectively. Also, we denote distortion by d, and the maximum degree of H by Δ. We denote by ℓ_g the length of a largest induced cycle, or geodesic cycle, in H. We approach the metric embedding problem by trying to understand the behavior of the shortest paths in G under a low distortion embedding into H, and what relation the mapping of these paths has to shortest paths in H. Careful analysis of this connection helps us solve a number of problems in this area, in the parameterized setting. **All the algorithmic results mentioned below are regarding non-contracting bounded distortion embeddings. However, all these results can be extended to find bounded distortion embeddings, without the assumption on non-contraction. For all the results, if the running time of the stated algorithm is T, then the running time of finding a bounded distortion embedding will be $(nN)^{O(1)} \cdot T$.**

We first begin by proving the following open question in [7]. Independently, a similar result on the same problem was obtained in same time by Carpenter et al. [4].

▶ **Theorem 1.1.** *Given an undirected unweighted graph G on n vertices, a cycle C and a distortion parameter d, there exists an algorithm that either finds a non-contracting distortion d embedding of G into C or decides that there does not exist such an embedding in* $O(n^3 \cdot d^{2d+3} \cdot (4d(2d+2))^{4d+4})$ *time.*

Due to the existence of the large geodesic cycle that is the graph H, techniques from the previous papers, like *pushing embeddings* [7], do not work and some new ideas are required to solve this problem. Moreover, our FPT algorithm can be extended to the case when the input graph G is a weighted graph, and we can parameterize by the distortion d and the maximum edge weight in G. We also show that the problem is NP-Complete when we do not take the maximum edge weight as a parameter for any distortion $d > 2$.

Observe that the *treewidth* of a cycle is 2, but the *connected treewidth* of a cycle is $\Omega(n)$ (see the definitions of treewidth and connected treewidth in Section 2). These two parameters (treewidth and connected treewidth of graphs) play important roles in this paper. In this direction, we first extend the result of Kenyon et al. [12] for bijection into bounded degree trees.

▶ **Theorem 1.2.** *Let G, H be two given graphs such that $|V(G)| = |V(H)| = n$, the maximum degree of H is Δ and the graph H has treewidth $tw(H) \leq \alpha$. Then there exists an algorithm that either finds a bijective non-contracting distortion d embedding of G into H or decides no such embedding exists in* $O(\alpha^2 n^{\alpha+3}) \cdot \Delta^{d+1} \cdot (\alpha\Delta^{d+1})^{\Delta^{O(\alpha d^2)}}$ *time.*

Note that the algorithm in Theorem 1.2 is not an FPT algorithm if $tw(H)$ is an input parameter to the problem. Therefore, it is natural to ask if we can still get FPT algorithms for a more general case, where $tw(H)$ is considered as a parameter instead of a constant. In this context, we prove the following result:

▶ **Theorem 1.3.** *Let G, H be two given graphs with n and N vertices, respectively, such that the maximum degree of H is Δ, treewidth $tw(H) \leq \alpha$ and the length of the longest geodesic cycle in H is ℓ_g. Then there exists an algorithm that either finds a non-contracting distortion d embedding of G into H or decides no such embedding exists in running time* $O(n^2 \cdot N) \cdot (\alpha \cdot \Delta^{d+1})^{\Delta^{O(\mu \cdot d + d^2)}} \cdot 2^{O((4(\mu+d))^{\alpha^2 \cdot \Delta^{d+1}})}$*, where* $\mu = 4(\alpha + \binom{\alpha}{2}(\ell_g(\alpha-2)-1))$.

This result crucially uses the result in [6] that a graph has bounded connected treewidth if and only if the graph has bounded treewidth and no long geodesic cycle. It is to be noted that a wheel graph has constant connected treewidth, and by a result in [12], embedding into wheel graphs is NP-hard even when the distortion $d = 2$. However, when the wheel graph has bounded degree, then the number of vertices in the wheel graph becomes bounded, and we obtain a trivial FPT algorithm parameterized by the degree and the distortion d. This motivated us to consider the above variant of metric embedding. Our FPT algorithm extends the result of Fellows et al. [7] for embedding into trees with bounded degree. Controlling the behavior of shortest paths in the graph G under a low distortion embedding into the class of graphs with bounded degree and bounded connected treewidth is algorithmically considerably harder than the case of bounded degree trees.

We also investigate bounded distortion embedding into *generalized theta graphs*: defined by the union of k internally vertex-disjoint paths all of which have common endpoints s and t. We prove the following result for generalized theta graphs.

▶ **Theorem 1.4.** Metric Embedding *into generalized theta graphs is FPT parameterized by distortion d and number k of $s-t$ paths. The algorithm runs in time $O(N) + n^5 \cdot k^{2k+1} \cdot (kd+1)^{(2d)^{O(kd)}} \cdot d^{O(d^2)}$, where n and N are the number of vertices in the input and output graph metrics, respectively.*

As mentioned earlier, it was shown in [6] that a graph has bounded connected treewidth if and only if the graph has bounded treewidth and no long geodesic cycle. In general, embedding into graphs with large geodesic cycles is not amenable to known algorithmic techniques in the parameterized settings. Intuitively, all known techniques for designing FPT algorithms in this area used the fact that if a low distortion embedding F exists, then the embedding of a shortest path between two vertices $u, v \in V(G)$ and the shortest path in H between $F(u)$ and $F(v)$ are somewhat structurally related. With the presence of large geodesic cycles this structural relation may completely break down: although the two paths have similar lengths, structurally they could be completely different. This poses a problem for designing dynamic programming algorithms, a staple for FPT algorithms in this area. The class of generalized theta graphs has treewidth 2, but may have large geodesic cycles. Hence, these graphs are more general than cycles and have constant treewidth, but they do not have bounded connected treewidth. Even for this very structured graph class, by virtue of the graphs having long geodesic cycles, we needed to develop completely new ideas in order to find low distortion embeddings into generalized theta graphs via FPT algorithms. The problem arises from the fact that any two geodesic cycles of a generalized theta graph intersect at at least two vertices, and there are many pairs of geodesic cycles with large intersections. Our algorithm is still a dynamic programming algorithm, but a more involved one. The way to work around the apparent barriers is to investigate more closely the structural properties of an input graph G that can be embedded with small distortion into a generalized theta graph. Independently, a generalization of this result was obtained in same time by Carpenter et al. [4]. We would like to mention that our algorithms for embedding into cycles and generalized theta graphs have better time complexity.

This is an extended abstract. For full details please refer to the full version of the paper [8].

2 Preliminaries

General Notation. We denote $\{1, \dots, t\}$ as $[t]$. For a set S, $|S|$ denotes the number of elements present in S. Given a function $f : U' \to D'$ and a function $F : U \to D$, where $U' \subseteq U$ and $D' \subseteq D$, we say that F *extends* f if for all $x \in U'$, $F(x) = f(x)$. For a set of functions $\Pi = \{f_i : A_i \to B_i, i \in [t]\}$ such that for any $i, j \in [t]$, $x \in A_i \cap A_j$ implies $f_i(x) = f_j(x)$, we define $\Phi_\Pi : \bigcup_{i=1}^{t} A_i \to \bigcup_{i=1}^{t} B_i$ such that $\Phi_\Pi(x) = f_i(x)$ for $i \in [t], x \in A_i$.

A graph is denoted by G while its vertex set and edge set are denoted by $V(G)$ and $E(G)$, respectively. We denote the set of neighbours of a vertex $v \in V(G)$ as $N_G(v)$. The degree of a vertex $v \in V(G)$ is denoted as $\mathsf{deg}_G(v)$. We also define $\Delta(G) = \max_{v \in V(G)} \mathsf{deg}_G(v)$. We also define the set $B(v, r) = \{u \in V(H) \mid D_H(u, v) \le r\}$, and refer to it as an r-ball around v. For a subgraph G' of G, $v \in V(G) \setminus V(G')$ is said to be a neighbour of G' if there is a vertex $u \in V(G')$ such that $(u, v) \in E(G)$. A subgraph G' of G is said to be an *induced subgraph* if $E(G') = \{(u, v) \in E(G) | u, v \in V(G')\}$. An induced cycle in a graph is also called a *geodesic cycle*.

A *generalized theta graph* is the union of k paths $\mathcal{P} = \{P_1, P_2, \ldots P_k\}$ such that the endpoints of all the paths are two vertices s and t, while every pair of paths are internally vertex and edge disjoint. Such a graph will also be referred to as a generalized theta graph defined at s,t, and the family $\mathcal{P}$ is said to define the generalized theta graph.

Treewidth. A *tree decomposition* [5] of a graph G is a tuple $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)})$, where T is a tree in which each vertex $\mathbf{u} \in V(T)$ has an assigned set of vertices $X_{\mathbf{u}} \subseteq V(G)$ (called a bag) such that the following properties hold: (i) $\bigcup_{\mathbf{u} \in V(T)} X_{\mathbf{u}} = V(G)$, (ii) for any $(x, y) \in E(G)$, there exists a $\mathbf{u} \in V(T)$ such that $x, y \in X_{\mathbf{u}}$, (iii) if $x \in X_{\mathbf{u}}$ and $x \in X_{\mathbf{v}}$, then $x \in X_{\mathbf{w}}$ for all $\mathbf{w}$ on the path from $\mathbf{u}$ to $\mathbf{v}$ in T.

The *treewidth* $tw_{\mathcal{T}}$ of a tree decomposition $\mathcal{T}$ is the size of the largest bag of $\mathcal{T}$ minus one. A graph may have several distinct tree decompositions. The treewidth $tw(G)$ of a graph G is defined as the minimum of treewidths over all possible tree decompositions of G. Note that for the tree T of a tree decomposition, we denote a vertex of $V(T)$ in bold font.

A tree decomposition $\mathcal{T} = (T, \{X_{\mathbf{u}}\}_{\mathbf{u} \in V(T)}))$ is called a *nice tree decomposition* if T is a tree rooted at some node $\mathbf{r}$ where $X_{\mathbf{r}} = \emptyset$, each node of T has at most two children, and each node is of one of the following kinds: (i) **Introduce node**: a node $\mathbf{u}$ that has only one child $\mathbf{u}'$ where $X_{\mathbf{u}} \supset X_{\mathbf{u}'}$ and $|X_{\mathbf{u}}| = |X_{\mathbf{u}'}| + 1$, (ii) **Forget vertex node**: a node $\mathbf{u}$ that has only one child $\mathbf{u}'$ where $X_{\mathbf{u}} \subset X_{\mathbf{u}'}$ and $|X_{\mathbf{u}}| = |X_{\mathbf{u}'}| - 1$, (iii) **Join node**: a node $\mathbf{u}$ with two children $\mathbf{u}_1$ and $\mathbf{u}_2$ such that $X_{\mathbf{u}} = X_{\mathbf{u}_1} = X_{\mathbf{u}_2}$, (iv) **Leaf node**: a node $\mathbf{u}$ that is a leaf of T, and $X_{\mathbf{u}} = \emptyset$.

One can show that a tree decomposition of width w can be transformed into a nice tree decomposition of the same width w and with $O(w|V(G)|)$ nodes, see e.g. [5]. For a node $\mathbf{u} \in V(T)$, let $T_{\mathbf{u}}$ denote the subtree rooted at $\mathbf{u}$ and $H_{\mathbf{u}}$ denote the subgraph induced by $\bigcup\limits_{\mathbf{v} \in V(T_{\mathbf{u}})} X_{\mathbf{v}}$. The set $\mathcal{B}(\mathbf{u}, r) = \bigcup\limits_{x \in X_{\mathbf{u}}} B(x, r)$

A *connected tree decomposition* is a tree decomposition where the vertices in every bag induce a connected subgraph of G [6]. The *connected treewidth* $ctw(G)$ of a graph G is defined as the minimum of treewidths over all possible connected tree decompositions of G.

Given a graph G, the function $D_G : V(G) \times V(G) \to \mathbb{R}$ is the shortest distance function defined on G; for any pair $u, v \in V(G)$, $D_G(u, v)$ is the length of the shortest path between u and v in the graph G. When we talk of a graph metric, then we denote it as the tuple (G, D_G). In this paper, unless otherwise mentioned, a graph metric is that of an *unweighted undirected graph.*

Metric Embedding. A metric embedding of a graph metric (G, D_G) into a graph metric (H, D_H) is a function $F : V(G) \to V(H)$. When the graph metrics are clear, we also use the terminology that the metric embedding is that of G into H, or that G is embedded into H. We also denote (G, D_G) as the *input metric space* and (H, D_H) as the *output metric space*. A non-contracting distortion d metric embedding implies that the expansion is at most d. Therefore, for any pair $u, v \in V(G)$, $D_G(u, v) \leq D_H(F(u), F(v)) \leq d \cdot D_G(u, v)$.

We consider the following two problems in this paper.

Gen-Graph Metric Embedding
Input: Two graph metrics (G, D_G) and (H, D_H), where G is a connected graph, and a positive integer d
Question: Is there a distortion d metric embedding of (G, D_G) into (H, D_H)?

Graph Metric Embedding
Input: Two graph metrics (G, D_G) and (H, D_H), where G is a connected graph, and a positive integer d
Question: Is there a non-contracting distortion d metric embedding of (G, D_G) into (H, D_H)?

The bijective versions of the above problems takes the same input but aims to determine whether the distortion d embedding is a bijective function. The Gen-Graph Metric Embedding problem or the Graph Metric Embedding problem for a graph class $\mathcal{G}$ is a variant where the output metric space (H, D_H) is such that $H \in \mathcal{G}$. In this extended abstract, we present results for the Graph Metric Embedding problem.

Parameterized Complexity. The instance of a *parameterized problem/language* is a pair containing the problem instance of size n and a positive integer k, which is called a parameter. The problem is said to be in FPT if there exists an algorithm that solves the problem in $f(k)n^{O(1)}$ time, where f is a computable function. Readers are requested to refer [5] for more details on Parameterized Complexity.

3 Graph Metric Embedding for Generalized Theta graphs

In this section, we design an FPT algorithm for embedding unweighted graphs into generalized theta graphs. Our FPT algorithm is parameterized by the distortion d and the number k of paths in the generalized theta graph. The strategy for the algorithm is still the same: that of putting together partial embeddings to obtain a non-contracting distortion d metric embedding. For this algorithm, we also observe structural properties of graphs that are embeddable into generalized theta graphs. We exploit these properties to obtain an FPT algorithm to compute a set of partial embeddings, and then use a dynamic programming algorithm to put together partial embeddings from the set to obtain the solution metric embedding. This makes the notion of partial embeddings more involved in this algorithm.

Let (G, D_G) be the graph metric that we want to embed into the graph metric (H, D_H). Here H is a generalized theta graph defined at s,t and let $\mathcal{P}$ be the family of $s - t$ paths that define H. To begin with, we try to guess the non-contracting distortion d embedding of (G, D_G) into (H, D_H), when restricted to a d-ball around s and around t.

▶ **Definition 3.1.** Let F be a non-contracting distortion d embedding of G into H. Define $B_s = \{v \in V(H) \mid D_H(v, s) \leq d\}$ and $B_t = \{v \in V(H) \mid D_H(v, t) \leq d\}$. For an embedding $F : V(G) \to V(H)$, define $\mathsf{Dom}_s^F = \{u \in V(G) \mid F(u) \in B_s\}$ and $\mathsf{Dom}_t^F = \{u \in V(G) \mid F(u) \in B_t\}$.

The following observation talks about the degree bound on the vertices of a graph that is embeddable into a generalized theta graph.

▶ **Observation 3.2.** If there exists a non-contracting distortion d embedding F of G into H, then:

(i) Each vertex in Dom_s^F can have degree at most $(k+1)d$. Similarly, each vertex in Dom_t^F can have degree at most $(k+1)d$,

(ii) All other vertices of G can have degree at most $2d$.

▶ **Observation 3.3.** The number of possible non-contracting distortion d embeddings of some $U \subseteq V(G)$ into $B_s \cup B_t$ is at most $n^2 \cdot (kd+1)^{(2d)^{O(kd)}}$.

We prove several properties of graphs that are embeddable into generalized theta graphs. For the given input graph G, let F be a non-contracting distortion d embedding and $\Psi : \mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F \rightarrow B_s \cup B_t$ be the restriction of F to $B_s \cup B_t$. Let $C_1, C_2, \dots C_a$ be the components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$.

▶ Remark. For simplicity of the presentation, we will assume the following:
1. For all $i \in [a]$, we have $|C_i| > 2kd^2 + 1$, and
2. for all $j \in [k]$, we have $|P_j| > 2k(2kd^2 + 1) + 3d$.

The details of the general case is handled in the full version [8].

We will derive certain properties of G with the help of the embedding F. For each $i \in [k]$, let $P_i' = P_i \setminus (B_s \cup B_t)$. If P_i' is a non-empty path, let s_i be the endpoint of P_i' that has an edge to B_s while t_i be the endpoint of P_i' that has an edge to B_t. Let S_i (T_i) denote the set of vertices of Dom_s^F (Dom_t^F) that are mapped into P_i.

▶ **Observation 3.4.** Let F be a non-contracting distortion d embedding, $\Psi : \mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F \rightarrow B_s \cup B_t$ be the restriction of F to $B_s \cup B_t$, and $C_1, C_2, \dots C_a$ be the components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$. Then each component of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ can have it's vertices mapped into exactly one P_i', $i \in [k]$. One the other hand, each P_i', $i \in [k]$, can have at most 2 connected components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ mapped into it, in the non-contracting distortion d embedding F.

Thus, there can be at most $2k$ components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$.

▶ **Definition 3.5.** Let F be a non-contracting distortion d embedding. An empty subpath of F is a subpath of the generalized theta graph where none of the vertices have any preimage. If a path P_i', $i \in [k]$, has an empty subpath with one endpoint at t_i, then such a subpath is called a *t-empty subpath*. Similarly, if a path P_i' has an empty subpath with one endpoint at s_i, then such a subpath is called a *s-empty subpath*. If a path P_i' contains an empty subpath that coincides with neither s_i nor t_i, then such a subpath is called an *internal-empty subpath*. Finally, it is possible that the path P_i' itself is an empty subpath and then P_i' is called a *fully-empty subpath*.

Note that a path P_i' can have at most one empty subpath with respect to F. Similarly, we classify the components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$.

▶ **Definition 3.6.** Let F be a non-contracting distortion d embedding. A component in $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ is called an *s-component* if it has neighbours to Dom_s^F and not to Dom_t^F. Similarly, we define a t-component. A *full component* is a component that has neighbours to both Dom_s^F and Dom_t^F.

Since F is a non-contracting distortion d embedding, the following observation is true.

▶ **Observation 3.7.** Let F be a non-contracting distortion d embedding. Any path P_i, $P_i' \neq \emptyset$, can be one of the following *forms*: **(i)** *form-1:* It has an s-component mapped into it by F, and a t-empty subpath, **(ii)** *form-2:* It has a t-component mapped into it by F, and an s-empty subpath, **(iii)** *form-3:* It has an s-component and a t-component mapped into it by F, and an internal-empty subpath, **(iv)** *form-4:* It has a full component mapped into it by F, and **(v)** *form-5:* It contains a fully-empty subpath.

If we refer P_i to be of *form-* ST , then P_i is of form-1 or form-2 or form-3. The objective is to find a non-contracting distortion d embedding F, if it exists. Although we do not know about F, we want to store a snapshot of F.

▶ **Definition 3.8.** A *configuration* $\mathcal{X}$ is a tuple $(\Psi, \mathcal{P}', \hat{\mathcal{P}})$ where:

(i) Let $U \subseteq V(G)$ be such that $G \setminus U$ creates a set of components $\{C_1, C_2, \ldots, C_a\}$, $a \leq 2k$. $\Psi : U \to B_s \cup B_t$ is a non-contracting distortion d embedding of U.

(ii) $\mathcal{P}' \subseteq \mathcal{P}$,

(iii) $\hat{\mathcal{P}}$ is a family of $|\mathcal{P} \setminus \mathcal{P}'|$ tuples such that for each path $P_i \in \mathcal{P} \setminus \mathcal{P}'$, there is a tuple $(\mathsf{form}_i, \mathcal{C}_{P_i}, \mathsf{comp}_i)$ with the following information: (a) form_i assigns the name of a form to P_i, (b) The set $\mathcal{C}_{P_i}$ is a set of at most 2 components of $G \setminus U$ that are assigned to P_i' and to no other $P_j', j \neq i$, (c) The function comp_i indicates for each $C \in \mathcal{C}_{P_i}$ whether it is an s-component or a t-component or full-component, with respect to Ψ.

(iv) $\bigcup_{P_i \in \mathcal{P} \setminus \mathcal{P}'} \mathcal{C}_{P_i}$ has all the components of $G \setminus U$.

For any fixed Ψ, the total number of configurations is $O(k^{2k})$. Next, we define feasible configurations that can be associated with metric embeddings.

▶ **Definition 3.9.** A configuration $\mathcal{X} = (\Psi, \mathcal{P}', \hat{\mathcal{P}})$ is said to be *feasible* with respect to a non-contracting distortion d embedding F of G into H if the following hold:

(i) $\Psi : \mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F \to B_s \cup B_t$ is the restriction of F to $\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F$.

(ii) $P_i' = P_i \setminus (B_s \cup B_t)$ is empty for each $P_i \in \mathcal{P}' \subseteq \mathcal{P}$. ,

(iii) For each P_i that is non-empty with respect to F, $\hat{\mathcal{P}}$ contains a tuple $(\mathsf{form}_i, \mathcal{C}_{P_i}, \mathsf{comp}_i)$ with the following information: (a) form_i is the form of P_i in F, (b) The set $\mathcal{C}_{P_i}$ is the set of at most 2 components of $G \setminus U$ that are embedded into P_i' by F, (c) The function comp_i indicates for each $C \in \mathcal{C}_{P_i}$ whether it is an s-component or a t-component or full-component, with respect to F.

(iv) $\bigcup_{P_i \in \mathcal{P} \setminus \mathcal{P}'} \mathcal{C}_{P_i}$ has all the components of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$.

We denote a configuration feasible with respect to F as $\mathcal{X}(F)$.

Next, we define the notion of a last vertex for a component of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ with respect to the embedding F.

▶ **Definition 3.10.** Let F be a non-contracting distortion d embedding. Let C be a j-component, $j \in \{s, t\}$. A vertex ℓ in C is the *last* vertex of C with respect to embedding F if $D_H(j, F(\ell)) \geq D_H(j, F(x))$ for all $x \in C$.

The following Lemma gives a bound on the potential last vertices of a component of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ if G is embeddable into H.

▶ **Lemma 3.11.** *Let $\mathcal{F}$ be a family of non-contracting distortion d embedding of G into H such that $\mathcal{X}(F_1) = \mathcal{X}(F_2)$ for any $F_1, F_2 \in \mathcal{F}$. Then for any form-* ST *path P_i and any $s(t)$-component $C \in \mathcal{C}_{P_i}$, there are $d^{O(d^2)}$ vertices that are candidates for being the last vertex of C with respect to some $F \in \mathcal{F}$.*

Next, we define the notion of a shortest embedding of a component in a path of $\mathcal{P}$.

▶ **Definition 3.12.** Let $\mathcal{Y}$ be a feasible configuration such that $\mathcal{Y} = \mathcal{X}(F)$ for a non-contracting distortion d embedding F. Let P_i be a form- ST path, $C \in \mathcal{C}_{P_i}$ be a s-component of $G \setminus (\mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F)$ and $\ell \in C$ be a candidate to be the last vertex of C with respect F.

Recall that S_i is the set of vertices of Dom_s^F that are mapped into P_i. Let $\mathcal{A}$ be a family of non-contracting and distortion d embedding of $C \cup S_i$ into P_i such that the following conditions hold: **(i)** $f_1|_{S_i} = f_2|_{S_i}$ for any $f_1, f_2 \in \mathcal{A}$, **(ii)** For each $f \in \mathcal{A}$, $f(x)$ is a vertex of P_i' for any $x \in C$, **(iii)** For each $f \in \mathcal{A}$, $F|_{C \cup S_i} = f$ and ℓ is the last vertex of C with respect to F, and **(iv)** For each $f \in \mathcal{A}$, for any $x \in \mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F$, the path between $f(\ell)$ and $f(x)$ is non-contracting with expansion at most d.

Then the *shortest* embedding of $C \cup S_i$ into P_i with respect to $\mathcal{Y}$ and ℓ, is an embedding $f \in \mathcal{A}$ such that $D_H(s, f(\ell)) \leq D_H(s, f'(\ell))$ for all $f' \in \mathcal{A}$. If C is a t-component, T_i is taken to be the set of vertices of Dom_t^F that are mapped into P_i and we can define the shortest embedding of $C \cup T_i$ with respect to $\mathcal{Y}$ and ℓ in a similar way.

We can extend the notion of shortest embedding of a component into a path of $\mathcal{P}$ to that of a non-contracting distortion d embedding of G into H that has shortest embeddings for all s-components and t-components.

▶ **Definition 3.13.** Let us consider a non-contracting distortion d embedding F of G into H. We say F is a *special embedding with respect to feasible configuration* $\mathcal{X}(F)$ if for every path P_i of form- ST and s (t)-component $C \in \mathcal{C}_P$, the following holds: $F|_{C\cup S_i}$ ($F|_{C\cup T_i}$) is the shortest embedding of $C \cup S_i$ ($C \cup T_i$) into P_i with respect to the feasible configuration $\mathcal{X}(F)$ and the last vertex of C with respect to F.

The next lemma shows that it is enough to look for a special embedding of G into H.

▶ **Lemma 3.14.** *If there exists a non-contracting distortion d embedding of G into H, then there exists a special embedding of G into H with respect to some configuration.*

Therefore, we have shown that if G is embeddable into H then it is enough to find a special embedding. We design an FPT algorithm for finding a special embedding.

Proof Sketch of Theorem 1.4. By Lemma 3.14, it is sufficient to look for special embedding with respect to some configuration. We find $\Delta(G)$ and if $\Delta(G) > (k+1)d$, then we report NO. This is correct by Observation 3.2.

We first compute $D_H(s,u)$ and $D_H(t,u)$ for all $u \in V(H)$. We store this distance information in a matrix $\mathcal{D}_{st}$, such that the look-up time for the distance from any $u \in V(H)$ to s or t is $O(1)$. Next, let us fix a non-contracting distortion d embedding Ψ of $U \subseteq V(G)$ into $B_s \cup B_t$ and a configuration $\mathcal{Y}$ containing Ψ. If the degree of any vertex in $G \setminus U$ is more than $2d$, then we decide that there does not exist any desired embedding with respect to $\mathcal{Y}$. Otherwise, we proceed as follows. Let F be the special embedding of G into H with respect to $\mathcal{Y}$ that we want to find, if one exists. Note that $U = \mathsf{Dom}_s^F \cup \mathsf{Dom}_t^F$.

(i) If a path P_i is of form-5, we don't have to do anything for that.

(ii) Let a path P_i be of form-4, and suppose $C \in \mathcal{C}_{P_i}$ is the only full component mapping into P_i. Then we find a non-contracting distortion d embedding f_C, if possible, of $C \cup S_i \cup T_i$ into P_i such that $f_C|_{S_i} = \Psi|_{S_i}$ and $f_C|_{T_i} = \Psi|_{T_i}$. Such an algorithm is described in the full version of the paper. If we cannot find such an embedding, then there does not exist any special embedding of G into H with respect to $\mathcal{Y}$.

(iii) Let P_i be a form- ST path and $C \in \mathcal{C}_{P_i}$ be an (a) s (t)-component. Without loss of generality, assume that C is an s-component. Here, our objective is to find the shortest embedding f of $C \cup S_i$ into P_i with respect to $\mathcal{Y}$ and some ℓ, where ℓ is the last vertex of C with respect to F. We guess a vertex $\ell \in C$, as the last vertex. By Lemma 3.11, the total number of candidates for the last vertex of C with respect to F is $d^{O(d^2)}$. It is easy to see that $|C \cup S_i| \leq D_H(f(\ell), f(a)) \leq 2d.|C \cup S_i|$. Thus, the length of the shortest embedding of $C \cup S_i$, where ℓ is the last vertex, is also in this range. For each possible length $|C \cup S_i| \leq \mathsf{len} \leq 2d.|C \cup S_i|$, we try to find a non-contracting distortion d embedding f_{len} of $C \cup S_i$ into a path $P_{\mathsf{len}} = \{1, 2, \ldots, \mathsf{len}\}$ such that f_{len} restricted to the first $|S_i|$ vertices is same as the mapping by Ψ, and for each $u \in C \cup S_i$, $D_{P_{\mathsf{len}}}(1, f_{\mathsf{len}}(u)) \leq D_{P_{\mathsf{len}}}(1, f_{\mathsf{len}}(\ell))$. Such an algorithm is described in the full version of the paper. If the algorithm returns no for all lengths, for every candidate ℓ for the last vertex, then there does not exist any special embedding of G into H with respect to $\mathcal{Y}$. Otherwise, assume that for the current guess ℓ, f_C is an embedding that the algorithm returns for the shortest length.

Let $F = \Phi_\Pi$ be the function such that $\Pi = \{\Psi\} \cup \{f_C \mid C \text{ is a component of } G \setminus U\}$. We verify whether the obtained F is a non-contracting distortion d embedding from G to H. If yes, we are done. If not, then there does not exist any special embedding with respect to $\mathcal{Y}$. Observe that the distance between two given points in H, can be computed in $O(1)$ time using $\mathcal{D}_{st}$.

Note that in the worst case, we have to run the above steps for all possible configurations. If we decide that there does not exist a special embedding with respect to all configurations, then we report that G does not admit the desired embedding of G into H. The correctness of the algorithm follows from Lemma 3.14. ◀

4 Graph Metric Embedding and connected treewidth

In this Section, we will look at the Graph Metric Embedding problem with respect to the added parameters of treewidth and longest geodesic cycle of the output graph metric. Let (G, D_G) be the input connected graph metric to be embedded into (H, D_H). We show that this problem is FPT, when parameterized by the distortion d, the treewidth $tw(H) = \alpha$, the length ℓ_g of the longest geodesic cycle of H, and the maximum degree $\Delta(H) = \Delta$. From [6] it can be shown that for a graph with longest geodesic cycle ℓ_g, a tree decomposition of treewidth α' can be converted into a connected tree decomposition of width $\alpha' + \binom{\alpha'}{2}(\ell_g(\alpha' - 2) - 1)$ in polynomial time. Since trees have constant connected treewidth, our algorithm is a generalization of the FPT algorithm for Graph Metric Embedding for trees, parameterized by distortion d and maximum degree Δ [7]. As before, we employ a dynamic programming to build a non-contracting distortion d metric embedding using a set of partial embeddings that are computed in FPT time.

Before we give the details of the algorithm, we want to make the following remark about bijective Graph Metric Embedding. We extended the algorithm of Kenyon et al [12] for bijective embedding of unweighted graphs into bounded maximum degree trees to the case of graphs with bounded maximum degree and bounded treewidth (see Theorem 1.2). The techniques we use for the results in this section are a generalization of the techniques used to prove Theorem 1.2. For the details of the proof, please refer to the full version of the paper [8].

Let (G, D_G) be a graph metric to be embedded into (H, D_H). Here the parameters are the treewidth α of H, the length of the longest geodesic cycle ℓ_g in H, the distortion d and the maximum degree Δ of H. Let $\mathcal{T}$ be a nice tree decomposition of H with width μ. Since from [6] H has a connected tree decomposition of width μ, we may assume that the nice tree decomposition is derived from the connected tree decomposition [5] and therefore the maximum distance between any two vertices inside a bag in $\mathcal{T}$ is $\Gamma \leq \mu$.

Ensuring non-contraction for a non-contracting distortion d metric embedding F is more elaborate. Local non-contraction no longer implies global non-contraction. This problem was dealt with in [7] by introducing the notion of *types*. For our algorithm too, for a vertex $\mathbf{u} \in V(T)$ we need to define a *type* for every vertex of $V(G)$ that is mapped into the subgraph $H_{\mathbf{u}}$, to indicate how it behaves with the rest of the graph. Informally, the types store information of the interaction of vertices of the graph seen so far with the boundary vertices, and this is enough to ensure global non-contraction.

▶ **Definition 4.1.** Let $\mathbf{u} \in V(T)$, $f_{\mathbf{u}}$ be a feasible partial embedding and $X_{\mathbf{u}} = \{u_1, \ldots, u_k\}$, $1 \leq k \leq \alpha_c$. Then:

(i) For $\mathbf{v} \in N_T(\mathbf{u})$ and $u_i \in X_{\mathbf{u}}$, $[f_{\mathbf{u}}, \mathbf{v}, u_i]$ *type* is a function $t^{u_i} : \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v}) \rightarrow \{\infty, 2\Gamma + 3d + 3, \Gamma + d + 1, \ldots, -(\Gamma + d + 1)\}$,

(ii) A $[f_{\mathbf{u}}, \mathbf{v}]$ *type* $\mathbf{t}$ is a tuple $(t^{u_i}, \ldots, t^{u_k})$, where t^{u_i} is a $[f_{\mathbf{u}}, \mathbf{v}, u_i]$ type, and

(iii) A $[f_{\mathbf{u}}, \mathbf{v}]$ *type-list* is a set of $[f_{\mathbf{u}}, \mathbf{v}]$ types.

Intuitively, we want to define a type corresponding to each vertex mapped into $H_{\mathbf{u}}$. However, this blows up the number of types. In order to handle this, it can be shown that we do not need to remember the type of each vertex, and that it is enough to only remember the type of vertices "close to" the vertices in $X_{\mathbf{u}}$. Now we present the formal arguments. To bound the total number of possible types, we define a function β as follows: $\beta(k) = k$ if $k < 2\Gamma + 3d + 3$, and $\beta(k) = \infty$ otherwise. In the following definitions, treat $\beta(k) = k$ and the definition of β will be clear while we prove our claims.

▶ **Definition 4.2.** Let us consider $\mathbf{u} \in V(T)$, $\mathbf{v} \in N_T(\mathbf{u})$. Let $f_{\mathbf{u}}$ be a feasible partial embedding and $\mathcal{L}$ be a $[f_{\mathbf{u}}, \mathbf{v}]$ type-list. Then $\mathcal{L}$ is said to be *compatible* with $\mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v})$ if the following condition is satisfied: For each $x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v})$ there exists a type $\mathbf{t} \in \mathcal{L}$, such that for each $y \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v})$, for all $u_i \in X_{\mathbf{u}}$ $D_H(f_{\mathbf{u}}(x), u_i) - D_G(x, y) = t^{u_i}(y)$.

▶ **Definition 4.3.** Let $\mathbf{u} \in V(T)$ and $f_{\mathbf{u}}$ be a feasible partial embedding. Also consider $\mathbf{v}, \mathbf{w} \in N_T(\mathbf{u})$ along with a $[f_{\mathbf{u}}, \mathbf{v}]$ type-list $\mathcal{L}_1$ and a $[f_{\mathbf{u}}, \mathbf{w}]$ type-list $\mathcal{L}_2$ such that $\mathbf{v} \neq \mathbf{w}$. Then $\mathcal{L}_1$ and $\mathcal{L}_2$ *agree* if the following condition is satisfied for all $u_i \in X_{\mathbf{u}}$: For every $\mathbf{t}_1 \in \mathcal{L}_1$ and $\mathbf{t}_2 \in \mathcal{L}_2$, there exists $x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v})$ and $y \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{w})$ such that $t_1^{u_i}(x) + t_2^{u_i}(y) \geq D_G(x, y)$ for all $u_i \in X_{\mathbf{u}}$.

Next, we define a state with respect to a vertex in T.

▶ **Definition 4.4.** Let $\mathbf{u} \in V(T)$. A $\mathbf{u}$*-state* constitutes of a feasible partial embedding $f_{\mathbf{u}}$, a $[f_{\mathbf{u}}, \mathbf{v}]$ type-list $\mathcal{L}[f_{\mathbf{u}}, v]$ for each $\mathbf{v} \in N_T(\mathbf{u})$.

Notice that it is no longer enough to consider feasibility and succession of partial embeddings. We also need to take care of the types of vertices. Therefore, we define feasibility and succession of states.

▶ **Definition 4.5.** A $\mathbf{u}$-state is said to be *feasible* if the following conditions are satisfied:

(i) $\mathcal{L}[f_{\mathbf{u}}, \mathbf{v}]$ is compatible with $\mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v})$, for each $\mathbf{v} \in N_T(\mathbf{u})$, and

(ii) $\mathcal{L}[f_{\mathbf{u}}, \mathbf{v}]$ agrees with $\mathcal{L}[f_{\mathbf{u}}, \mathbf{w}]$, for any $\mathbf{v}, \mathbf{w} \in N_T(\mathbf{u})$ and $\mathbf{v} \neq \mathbf{w}$.

▶ **Definition 4.6.** Let $\mathbf{u} \in V(T)$ and $\mathbf{v} \in C_T(\mathbf{u})$. Let $\mathcal{S}_{\mathbf{u}}, \mathcal{S}_{\mathbf{v}}$ be feasible $\mathbf{u}$-state and $\mathbf{v}$-state, respectively. $\mathcal{S}_{\mathbf{v}}$ is said to *succeed* $\mathcal{S}_{\mathbf{u}}$ if the following properties hold.

(i) $f_{\mathbf{v}}$ succeeds $f_{\mathbf{u}}$.

(ii) For every $\mathbf{w} \in N_T(\mathbf{v}) \setminus \mathbf{u}$ and a type $\mathbf{t}_1 \in \mathcal{L}[f_{\mathbf{v}}, \mathbf{w}]$ there exists a type $\mathbf{t}_2 \in \mathcal{L}[f_{\mathbf{u}}, \mathbf{v}]$ satisfying the following conditions: **(a)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v}) \cap \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})$ and $a \in X_{\mathbf{u}} \cap X_{\mathbf{v}}$, $t_2^a(x) = t_1^a(x)$. **(b)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v}) \cap \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})$ and $a \in X_{\mathbf{u}} \setminus X_{\mathbf{v}}$, $t_2^a(x) = \beta(\min_{b \in X_{\mathbf{v}}} (D_H(a, b) + t_1^b(x)))$. **(c)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v}) \setminus \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})$ and $a \in X_{\mathbf{u}} \cap X_{\mathbf{v}}$, $t_2^a(x) = \beta(\max_{y \in \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})} (t_1^a(y)) - D_G(x, y)))$. **(d)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{v}) \setminus \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})$ and $a \in X_{\mathbf{u}} \setminus X_{\mathbf{v}}$, $t_2^a(x) = \beta(\max_{y \in \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{w})} (\min_{b \in X_{\mathbf{v}}} (D_H(a, b) + t_1^b(y)) - D_G(x, y)))$.

(iii) For every $\mathbf{w} \in N_T(\mathbf{u}) \setminus \mathbf{v}$ and a type $\mathbf{t}_1 \in \mathcal{L}[f_{\mathbf{u}}, \mathbf{w}]$ there exists a type $\mathbf{t}_2 \in \mathcal{L}[f_{\mathbf{v}}, \mathbf{u}]$ satisfying the following conditions: **(a)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{u}) \cap \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{w})$ and $a \in X_{\mathbf{u}} \cap X_{\mathbf{v}}$, $t_2^a(x) = t_1^a(x)$. **(b)** $\forall x \in \mathsf{Dom}_{f_{\mathbf{v}}}(\mathbf{u}) \cap \mathsf{Dom}_{f_{\mathbf{u}}}(\mathbf{w})$ and $a \in X_{\mathbf{v}} \setminus X_{\mathbf{u}}$, $t_2^a(x) = \beta \min_{b \in X_{\mathbf{u}}} (D_H(a, b) +$

$t_1^b(x))$. **(c)** $\forall x \in \mathsf{Dom}_{f_\mathbf{v}}(\mathbf{u}) \setminus \mathsf{Dom}_{f_\mathbf{u}}(\mathbf{w})$ and $a \in X_\mathbf{u} \cap X_\mathbf{v}$, $t_2^a(x) = t_1^a(x)$. **(d)** $\forall x \in \mathsf{Dom}_{f_\mathbf{v}}(\mathbf{u}) \setminus \mathsf{Dom}_{f_\mathbf{u}}(\mathbf{w})$ and $a \in X_\mathbf{v} \setminus X_\mathbf{u}$, $t_2^a(x) = \beta(\max_{y \in \mathsf{Dom}_{f_\mathbf{u}}(\mathbf{w})}(\min_{b \in X_\mathbf{u}}(D_H(a,b) + t_1^b(y)) - D_G(x,y)))$.

Now, we define the embeddability of a set of feasible states.

▶ **Definition 4.7.** For $u \in V(T)$, let $\mathcal{S}_u$ denote a u-state. The set $\{\mathcal{S}_\mathbf{u} : \mathbf{u} \in V(T)\}$ is said to be an *embeddable* set of feasible states if the following conditions are satisfied: (i) For each $\mathbf{u} \in V(T)$, $\mathcal{S}_\mathbf{u}$ is a feasible state, and (ii) For $\mathbf{u} \in V(T)$ and $\mathbf{v} \in C_T(\mathbf{u})$, $\mathcal{S}_\mathbf{v}$ succeeds $\mathcal{S}_\mathbf{u}$.

The above definitions are enough to show the relation between the existence of a non-contracting distortion d embedding of G into H and the existence of an embeddable set of feasible states. This is proved over the following two Lemmas. Lemma 4.9 is the most important structural Lemma for the design of this algorithm. We give a brief sketch of this Lemma and refer to the full details in the full version. For the proof of Lemma 4.8 refer to the full version.

▶ **Lemma 4.8.** *Let F be a non-contracting and distortion d embedding of G into H. Then there exists an embeddable set of feasible states.*

▶ **Lemma 4.9.** *Let $\Pi = \{f_\mathbf{u} : \mathbf{u} \in V(T)\}$ be an embeddable set of feasible states. Then there exists a non-contracting and distortion d embedding of G into H.*

Proof Sketch. To prove this lemma we first show the following:
1. For every $x \in V(G)$, there exists a feasible u-state such that $x \in \mathsf{Dom}_{f_\mathbf{u}}$, and
2. The subgraph of T induced by $A_x = \{\mathbf{u} \in V(T) : x \in \mathsf{Dom}_{f_\mathbf{u}}\}$ is connected. Moreover, $x \in \mathsf{Dom}_{f_\mathbf{u}} \cap \mathsf{Dom}_{f_\mathbf{v}}$ implies $f_\mathbf{u}(x) = f_\mathbf{v}(x)$.

Next, using the family Π, we construct an embedding F that satisfies the following (Please refer to the full version for this construction):

(i) F is a metric embedding with expansion at most d,

(ii) Consider a path $P = \mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_k$ from $\mathbf{u} = \mathbf{u}_1$ to $\mathbf{v} = \mathbf{v}_k$ in T. Then for every $x \in \mathsf{Dom}_{f_\mathbf{v}}$, at least one of the following properties hold.

Prop-1: There exists a $\mathbf{u}_j \in P$ and $y \in \mathsf{Dom}_{f_{\mathbf{u}_j}}$ such that $D_H(F(x), u') - D_G(x,y) \geq 2\Gamma + 3d + 3$ for all $u' \in X_{\mathbf{u}_j}$.

Prop-2: There exists a type $\mathbf{t}_x \in \mathcal{L}[f_\mathbf{u}, \mathbf{u}_2]$ such that $\mathbf{t}_x^{u'}(y) = D_H(F(x), u') - D_G(x,y)$ for all $y \in \mathsf{Dom}_{f_\mathbf{u}}(\mathbf{u}_2)$ and $u' \in X_\mathbf{u}$.

(iii) Consider a path $P = \mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_k$ from $\mathbf{u}$ to $\mathbf{v}$ in T, where $\mathbf{u} = \mathbf{u}_1$ and $\mathbf{v} = \mathbf{u}_k$. Then F restricted to $\bigcup_{\mathbf{u}_i \in P} \mathsf{Dom}_{f_{\mathbf{u}_i}}$ is non-contracting.

Now we will be done if we prove that F is a non-contracting embedding for any two vertices $x, y \in V(G)$. Note that each of $F(x)$ and $F(y)$ is in some bag. Fix $\mathbf{u}, \mathbf{v} \in V(T)$ such that $F(x) \in X_\mathbf{u}$ and $F(y) \in X_\mathbf{v}$. Consider the path $P = \mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_k$ from $\mathbf{u}$ to $\mathbf{v}$ in T, where $\mathbf{u} = \mathbf{u}_1$ and $\mathbf{v} = \mathbf{u}_k$. We can show that the shortest path between $F(x)$ to $F(y)$ is non-contracting as $x, y \in \bigcup_{\mathbf{u}_i \in P} \mathsf{Dom}_{f_{\mathbf{u}_i}}$. ◀

Proof Ideas for Theorem 1.3. A graph that is embeddable into the given H must have bounded maximum degree. This helps in proving bounds for the total number of feasible partial embeddings and the total number of feasible states. After this, the proof of Theorem 1.3 uses the standard dynamic programming approach over a bounded tree-decomposition of a graph. ◀

5 Open Questions

The parameterized complexity of embedding into trees of unbounded degree, asked in [7], still remains open. A generalization of that question is to determine the parameterized complexity of Graph Metric Embedding for bounded treewidth graphs, and this is also open.

References

1 I. Abraham, Y. Bartal, and O. Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011.
2 M. Badoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos. Low-Distortion Embeddings of General Metrics into the Line. In *Proc. of STOC*, pages 225–233, 2005.
3 M. Badoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Räcke, R. Ravi, and A. Sidiropoulos. Approximation Algorithms for Low-Distortion Embeddings into Low-Dimensional spaces. In *Proc. of SODA*, pages 119–128, 2005.
4 T. Carpenter, F. V. Fomin, D. Lokshtanov, S. Saurabh, and A. Sidiropoulos. Algorithms for low-distortion embeddings into arbitrary 1-dimensional spaces. *CoRR*, abs/1712.06747, 2017. To appear in *SoCG*, 2018.
5 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Parameterized algorithms. *Springer*, 5(4):16:1–16:20, 2015.
6 Reinhard Diestel and Malte Müller. Connected tree-width. *Combinatorica*, 2017.
7 M. R. Fellows, F. V. Fomin, D. Lokshtanov, E. Losievskaja, F. A. Rosamond, and S. Saurabh. Distortion is Fixed Parameter Tractable. *ACM Transactions on Computation Theory*, 5(4):16:1–16:20, 2013.
8 A. Ghosh, S. Kolay, and G. Mishra. FPT algorithms for embedding into low complexity graphic metrics. *CoRR*, 2018. `arXiv:1801.03253`.
9 A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, Trees and l_1-Embeddings of Graphs. *Combinatorica*, 24(2):233–269, 2004.
10 P. Indyk. Algorithmic Applications of Low-Distortion Geometric Embeddings. In *Proc. of FOCS 2001*, pages 10–33, 2001.
11 P. Indyk and J. Matoušek. Low-Distortion Embeddings of Finite Metric Spaces. In *Handbook of Discrete and Computational Geometry, Second Edition.*, pages 177–196. CRC, 2004.
12 C. Kenyon, Y. Rabani, and A. Sinclair. Low Distortion Maps Between Point Sets. *SIAM Journal on Computing*, 39(4):1617–1636, 2010. Peliminary version in *Proc. of STOC*, 2004.
13 N. Linial, E. London, and Y. Rabinovich. The Geometry of Graphs and Some of its Algorithmic Applications. *Combinatorica*, 15(2):215–245, 1995.
14 J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., 2002.
15 J. Matoušek. Lecture Notes on Metric Embeddings, 2013.
16 A. Nayyeri and B. Raichel. Reality Distortion: Exact and Approximate Algorithms for Embedding into the Line. In *Proc. of FOCS*, pages 729–747, 2015.
17 A. Nayyeri and B. Raichel. A Treehouse with Custom Windows: Minimum Distortion Embeddings into Bounded Treewidth Graphs. In *Proc. of SODA*, pages 724–736, 2017.