Composed Degree-Distance Realizations of Graphs

Amotz Bar-Noy[1], David Peleg[2](✉), Mor Perry[2], and Dror Rawitz[3]

[1] City University of New York (CUNY), New York City, USA
amotz@sci.brooklyn.cuny.edu
[2] Weizmann Institute of Science, Rehovot, Israel
{david.peleg,mor.perry}@weizmann.ac.il
[3] Bar Ilan University, Ramat-Gan, Israel
dror.rawitz@biu.ac.il

Abstract. Network realization problems require, given a specification π for some network parameter (such as degrees, distances or connectivity), to construct a network G conforming to π, or to determine that no such network exists. In this paper we study *composed profile* realization, where the given instance consists of *two* or more profile specifications that need to be realized simultaneously. To gain some understanding of the problem, we focus on two classical profile types, namely, *degrees* and *distances*, which were (separately) studied extensively in the past. We investigate a wide spectrum of variants of the composed distance & degree realization problem. For each variant we either give a polynomial-time realization algorithm or establish NP hardness. In particular:

- We consider both *precise* specifications and *range* specifications, which specify a range of permissible values for each entry of the profile.
- We consider realizations by both weighted and unweighted graphs.
- We also study settings where the realizing graph is restricted to specific graph classes, including trees and bipartite graphs.

1 Introduction

This paper considers the family of *network realization* problems. A *Π-realization* problem concerns some type of network parameter Π on networks, such as the vertex degrees, inter-vertex distances, centrality, connectivity, and so on. With every network G one can associate its Π-profile, $\Pi(G)$, giving[1] the values of Π on G. An instance of the Π-realization problem consists of a *specification* π, detailing the requirements on Π. Given such a specification π, it is necessary to construct a network G conforming to it, i.e., satisfying $\Pi(G) = \pi$, or to determine that no such network exists. The motivation for network realization problems stems from both "exploratory" contexts where one attempts to reconstruct an

Supported in part by a US-Israel BSF grant (2018043).

[1] We consider profile types for which $\Pi(G)$ is polynomial-time computable given G.

P. Flocchini and L. Moura (Eds.): IWOCA 2021, LNCS 12757, pp. 63–77, 2021.
https://doi.org/10.1007/978-3-030-79987-8_5

existing network of unknown structure based on the outcomes of experimental measurements, and engineering contexts related to network design.

Two cannonical examples of profile types are vertex *degrees* and *distances*.

Degree Realization. The most well-studied family of realization problems concerns vertex degrees. The *degree profile* of a (simple undirected) graph $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_n\}$ is an integer sequence $\mathrm{DEG}(G) = (\delta_1, \ldots, \delta_n)$, where $\delta_i = \deg_{G,i}$ is the degree of vertex i in G. The *degree realization* problem asks to decide, given a sequence of n non-negative integers $\bar{\delta} = (\delta_1, \ldots, \delta_n)$, whether there exists a graph G whose degree sequence $\mathrm{DEG}(G)$ equals $\bar{\delta}$. A sequence that admits such a realization is called *graphic*. The main questions studied in the past concerned characterizations for a sequence to be graphic and algorithms for finding a realizing graph for a given sequence if exists. For a brief review and some references to previous work on degree realization see, e.g., [6] in these proceedings. Graphic sequences were studied also on specific graph families, such as trees and bipartite graphs [14,24,26,34,43].

Rather than *precise* degree requirements, some studies concerned *degree-range* specifications, which define a range of allowable degrees for each vertex. An entry in the specification $\bar{\delta}$ consists of a pair $[\delta_i^-, \delta_i^+]$, and the realizing graph G must satisfy $\delta_i^- \leq \deg_G(i) \leq \delta_i^+$. Again see [6] for references to prior work.

Distance Realization. In a graph G, the *distance* $dist_G(u, v)$ between two nodes u and v is the length of the shortest path connecting them in G. (The length of a path is the sum of its edge weights; in an unweighted graph, the weight of each edge is taken to be 1.) The *distance profile* of an n-vertex graph $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_n\}$ consists of an $n \times n$ matrix $\mathrm{DIST}(G) = D$, where $D_{i,j} \in \mathbb{N} \cup \{\infty\}$ for every $1 \leq i < j \leq n$, which specifies the required distance between every two nodes $i \neq j$ in the graph ($D_{i,j} = \infty$ when i and j are in different disconnected components).

The *unweighted distance realization* problem is defined as follows. An instance consists of an $n \times n$ matrix D, where $D_{i,j}$ is a nonnegative integer or ∞, for every $1 \leq i < j \leq n$. The goal is to compute an n-vertex unweighted undirected graph $G = (V, E)$ realizing D, i.e., such that $V = \{1, \ldots, n\}$ and $dist_G(i, j) = D_{i,j}$ for every $1 \leq i < j \leq n$, or to decide that no such realizing graph exists. In the *weighted* distance realization problem the edges of the realizing graph may have arbitrary integral weights. (We assume that the minimum edge weight is 1.)

Distance realization problems were introduced and studied in a seminal paper of Hakimi and Yau [27] (see [6] for a review). Precise distance realizations by trees and bipartite graphs were considered as well [4,11]. The distance realization problem was studied also for *distance ranges*, i.e., where an entry in the given distance matrix D consists of a pair $[D^-_{i,j}, D^+_{i,j}]$, and the realizing graph G must satisfy $D^-_{i,j} \leq dist_G(i, j) \leq D^+_{i,j}$, for every $1 \leq i < j \leq n$ [33,39].

Profile Composition. In reality, it is often required to address multiple network parameters simultaneously. In particular, one may be given specifications for two or more different profile types, and be requested to find a realizing graph conforming to all of these specifications simultaneously. We refer to the input as a "composed" profile specification. *Profile composition* is one of the fundamental, yet little understood, aspects of the realization problem. Specifically, we are interested in the following setting. Consider two different profile types, Π^A and Π^B. Given two profile specifications π^A and π^B corresponding to these profile types, the goal is to solve the realization problem of the composed profile specification $\pi^A \wedge \pi^B$, namely, construct a graph G that realizes both specifications simultaneously, if exists, or decide that this is impossible.

As a first concrete example, we focus in this paper on the composition of degree and distance profiles. In the composed *degree and distance (D&D) realization* problem, we are given both a degree vector $\bar{\delta}$ and a distance matrix D, and the goal is to decide whether there is an n-vertex (unweighted/weighted) undirected graph $G = (V, E)$ realizing $\bar{\delta}$ and D simultaneously, i.e., such that $V = \{1, \ldots, n\}$, $\deg_G(i) = \delta_i$ and $dist_G(i, j) = D_{i,j}$ for every $1 \leq i < j \leq n$.

We study such compositions in a variety of settings, including *precise* and *range* specifications, and realizations by *weighted* and *unweighted* graphs. In addition, we consider realizations by more restricted graph classes, such as (weighted or unweighted) *trees* and *bipartite graphs*. A somewhat different type of restriction that we consider is where the input to the problem consists also of a given graph G^+ (the *supergraph*), and the realizing graph G must be a *subgraph* of G^+, or in other words, G must belong to the class of subgraphs[2] of G^+.

Terminology. The above exposition outlines a wide collection of problems, classified by a number of characteristics, including the following:

- the *degree specification types* $\tilde{\delta} \in \{P, [\ ]\}$:
  indicates whether the input specifies exact degrees or degree-ranges,
- the *distance specification types* $\tilde{D} \in \{P, [\ ]\}$:
  indicates whether the input specifies exact distances or distance-ranges,
- the *class of graphs* $\tilde{g} \in \{U, W, UT, WT, UB, WB, US, WS\}$:
  indicates whether the realizing graph should be a (unweighted or weighted) general graph, tree, bipartite graph, or subgraph of a given graph.

Accordingly, the studied variants of the realization problem are:

- DEG$(\tilde{\delta}, \tilde{g})$: degree realization problems[3],
- DIST$(\tilde{D}, \tilde{g})$: distance realization problems,
- D&D$(\tilde{\delta}, \tilde{D}, \tilde{g})$: composed D&D realization problems.

Our Contributions. In Sect. 2 we present some basic properties of profile composition, concerning the relations between the properties of the two profile types Π^A and Π^B and the properties of their composition, $\Pi^A \wedge \Pi^B$. In Sect. 3 we focus

[2] Such a problem may arise naturally in a setting where it is known a priori that certain connections are impossible, infeasible or disallowed, due to the environment, the user specified requirements, or other reasons.

[3] For the pure degree-realization problem, there is no distinction between weighted and unweighted graphs.

on the properties of profile composition when one of the composed profiles is distances. Then, in Sect. 4, we turn to studying composed D&D realization problems. Most of the literature on realization problems concerns either efficient realization algorithms or characterizations (i.e., necessary or sufficient conditions for realizability), and negative results are scarce. Here, we are interested in classifying the resulting problems according to their complexity. We present, for each variant of this problem, either a polynomial-time algorithm or a proof that it is NP-hard. Specifically, we show that when the distance matrix specifies precise values, there are polynomial-time algorithms for all variants (precise degrees or degree-ranges realizations by weighted or unweighted arbitrary graphs, trees, bipartite graphs, and subgraphs of a given graph). However, when the distance matrix specifies distance-ranges, all variants are NP-hard. Along the way, we also fill in the gaps for some distance realization problems whose status was unresolved so far.

Related Work. A concrete type of composition that has received considerable attention concerns degree sequences that satisfy an additional graph property P. Let P be an invariant property of graphs. A graphic degree sequence $\bar{\delta}$ is said to be *potentially P-graphic* if there exists at least one graph G conforming with $\bar{\delta}$ that has the property P. The sequence $\bar{\delta}$ is *forcibly P-graphic* if every graph G conforming with $\bar{\delta}$ enjoys P. For example, consider the property that the graph is k-edge-connected. The composed profile specification consists of a degree specification and a k-edge-connectivity specification. An algorithm for this composed profile was given in [3]. Conditions for the existence of k-edge connected realizations of degree sequences, for $k \geq 1$, are also known [19]. Necessary and sufficient conditions for the realization of k-vertex connected graphs where $k \geq 2$, as well as a realization algorithm, were presented in [41]. For a survey on potentially and forcibly P-graphic sequences, see [32].

The *optimal distance realization* problem was also introduced in [27] and studied further in [1,5,13,15–18,20,28,30,31,33,35–38,40,42]. In this problem, a distance matrix D is given over a set S of n terminal vertices, and the goal is to find a *minimum-weight* graph G containing S, with possibly additional *auxiliary* vertices, that realizes the given D for S. In contrast, our "pure" distance realization problem requires the realizing graph to have exactly n vertices, and does not allow adding auxiliary ones. Hence, the results obtained for the optimal distance realization problem do not seem to carry over easily to our setting.

Realization questions were studied in the past for other types of network information *profiles*, including eccentricites [12,29], connectivity and flow [21–23,25] and maximum or minimum neighborhood degrees [7,10]. Several other realization problems are surveyed in [8,9].

2 Profile Composition

Consider a Π-realization problem for some network parameter Π. An instance π of the Π-realization problem is *realizable* by a graph of the class $\mathcal{GC}$ if there exists a network $G \in \mathcal{GC}$ satisfying $\Pi(G) = \pi$. We assume that an instance π (of any profile type) always specifies also the network size n (i.e., the number of vertices). Let $\mathcal{RG}(\pi, \mathcal{GC})$ denote the set of graphs $G \in \mathcal{GC}$ realizing π.

We are interested in several properties of profile types. Consider a profile type Π and a graph class $\mathcal{GC}$.

– Π is *enumerable* over $\mathcal{GC}$ if for each of its instances π, the set $\mathcal{RG}(\pi, \mathcal{GC})$ has size polynomial in n, and moreover, there is a polynomial-time procedure that given π generates all the graphs of $\mathcal{RG}(\pi, \mathcal{GC})$ successively.
– Π is *unique* over $\mathcal{GC}$ if each instance π has $|\mathcal{RG}(\pi, \mathcal{GC})| \leq \infty$, and moreover, there is a polynomial-time procedure that given π generates the unique graph of $\mathcal{RG}(\pi, \mathcal{GC})$ or declares that $\mathcal{RG}(\pi, \mathcal{GC}) = \emptyset$.
– Π is *verifiable* over $\mathcal{GC}$ if there is a polynomial-time procedure that given a specification π and $G \in \mathcal{GC}$ checks whether G realizes π (or, $\Pi(G) = \pi$).
– Π is *realizable* over $\mathcal{GC}$ if there is a polynomial-time procedure that given a specification π finds a graph $G \in \mathcal{GC}$ such that $\Pi(G) = \pi$, if π is realizable, and otherwise indicates that no such graph exists[4].
– Π is *super-realizable* over $\mathcal{GC}$ if there is a polynomial-time procedure that given a specification π and $G = (V, E)$ finds a supergraph[5] $G' = (V, E') \in \mathcal{GC}$ of G such that $\Pi(G) = \pi$, if π is realizable by a supergraph of G, and otherwise indicates that no such graph exists.

Towards gaining an understanding of profile composition, a central goal is to identify relationships between the realizability of individual profile types and the realizability of their composition. For example, one might hope to establish some connections between the following two conditions:

$$\text{“both } \Pi^A \text{ and } \Pi^B \text{ are realizable.”} \tag{C1}$$

$$\text{“}\Pi^A \wedge \Pi^B \text{ is realizable.”} \tag{C2}$$

Unfortunally, profile composition turns out to be more intricate. To illustrate this, let us consider the following examples.

Example 1. Let Π^A be the *full information* profile type, i.e., an instance of it is of the form $\pi^A = \langle G_0 \rangle$, a complete description of some n-vertex graph G_0. This profile type is realizable, and has a unique realization for every instance, namely, G_0 itself. Let Π^B be the *clique* profile type, i.e., an instance for it consists of two integers, $\pi^B = \langle k_0, n \rangle$, where $k_0 \leq n$, and a realizing n-vertex graph is required to have a clique of size k_0. This profile type is also realizable, simply by taking the complete n-vertex graph K_n. Hence both profile types are realizable in a trivial way. However, for the composed profile type $\Pi^A \wedge \Pi^B$, the realization problem is NP-hard, since in order to decide if a given instance $\pi^A \wedge \pi^B = \langle G_0 \rangle \wedge \langle k_0, n \rangle$ is realizable, we must determine if G_0 has a k_0-clique. $\square$

Hence, (C1) is not a sufficient condition for (C2), unless NP=P.

[4] Note the difference between a realizable specification π (“π has a realization”) and a realizable profile type Π (“there is a polynomial-time algorithm deciding, for every specification π of Π, if it is realizable”).
[5] satisfying $E \subseteq E'$.

Example 2. Let Π^A be the *2-coloring* profile type, i.e., an instance of it is of the form $\pi^A = \langle n \rangle$, and a realizing n-vertex graph is required to have a legal 2-vertex coloring. This profile is realizable, simply by taking an n-vertex path. Let Π^B be the *graph 3-coloring* profile type, i.e., an instance of it is of the form $\pi^B = \langle G_0 \rangle$, a complete description of some n-vertex graph G_0 and the realizing graph is also required to have a legal 3-vertex coloring. This realization problem is NP-hard, since in order to decide if it is realizable, we must determine if G_0 has a legal 3-vertex coloring. Hence profile type Π^A is realizable in a trivial way, but profile type Π^B is not realizable, unless NP = P. However, the composed profile type $\Pi^A \wedge \Pi^B$, is realizable, since in order to decide if a given instance $\pi^A \wedge \pi^B = \langle n \rangle \wedge \langle G_0 \rangle$ is realizable, we only need to determine if G_0 has a legal 2-vertex coloring, which is known to have a polynomial algorithm. □

Hence, if P $\neq$ NP, then (C1) is not a necessary condition for (C2).

We establish the following sufficient condition.

Lemma 1. *For any profile types Π^A and Π^B and graph class $\mathcal{GC}$, if Π^A is enumerable over $\mathcal{GC}$ and Π^B is verifiable over $\mathcal{GC}$, then the composed profile type $\Pi^A \wedge \Pi^B$ is realizable over $\mathcal{GC}$.*

Proof. Let Π^A and Π^B be as in the lemma. The following algorithm is a polynomial realization for the composed profile type $\Pi^A \wedge \Pi^B$ over $\mathcal{GC}$. Let π^A and π^B be specifications of Π^A and Π^B, respectively. Succesively generate the graphs of $\mathcal{RG}(\pi^A, \mathcal{GC})$, and for each generated graph G, check whether it realizes π^B, and if so, return G and halt. If every $G \in \mathcal{RG}(\pi^A, \mathcal{GC})$ does not realize π^B, return "Impossible". The correctness of this algorithm is immediate. Since Π^A is enumerable over $\mathcal{GC}$, there is a polynomial-time procedure that generates all $G \in \mathcal{RG}(\pi^A, \mathcal{GC})$, and since π^B is verifiable over $\mathcal{GC}$, checking whether G realizes π^B is polynomial. Overall, the running time of this algorithm is polynomial. □

Interestingly, note that the verifiability of both profiles Π^A and Π^B is not always a *necessary* condition for the verifiability of the composed profile $\Pi^A \wedge \Pi^B$. To see this, consider the following example.

Example 3. Consider the following two profile types. Profile type Π^A requires that all degrees in G are 2. Profile type Π^B requires that G contains a Hamiltonian cycle. Consequently, the composed profile type $\Pi^A \wedge \Pi^B$ necessitates that G is a cycle. Note that over general graphs, Π^A is both always realizable and verifiable. Π^B is always realizable too, but verifying it is NP-hard. Yet their composition $\Pi^A \wedge \Pi^B$ is still both always realizable and verifiable. □

Lemma 2. *Let Π^A be a unique profile type over the graph class $\mathcal{GC}$. Then for any profile type Π^B, the composed $\Pi^A \wedge \Pi^B$ is realizable over $\mathcal{GC}$ if and only if Π^B is verifiable over $\mathcal{GC}$.*

Proof. Fix $\mathcal{GC}$, let Π^A be a unique profile type, and consider some profile type Π^B. Let ALG_A be the realization procedure of Π^A over $\mathcal{GC}$. We first show that if Π^B is verifiable over $\mathcal{GC}$ then $\Pi^A \wedge \Pi^B$ is realizable over $\mathcal{GC}$. Suppose Π^B

is verifiable over $\mathcal{GC}$, and let ALG_B be a verification algorithm for it. Then the following algorithm $ALG_{A\wedge B}$ solves the $\Pi^A \wedge \Pi^B$-realization problem over $\mathcal{GC}$. Given an instance specification $\pi^A \wedge \pi^B$ of $\Pi^A \wedge \Pi^B$, invoke procedure ALG_A on π^A. If procedure ALG_A returns "impossible", then return "impossible". If it returns the (unique) graph $G \in \mathcal{GC}$ realizing π^A, then invoke Alg_B on G and return its response. It is clear that algorithm $ALG_{A\wedge B}$ is correct.

In the opposite direction, we need to show that if $\Pi^A \wedge \Pi^B$ is realizable over $\mathcal{GC}$ then Π^B is verifiable over $\mathcal{GC}$. Suppose $\Pi^A \wedge \Pi^B$ is realizable over $\mathcal{GC}$, and let $ALG_{A\wedge B}$ be an algorithm for the $\Pi^A \wedge \Pi^B$-realization problem. Then the following procedure ALG_B verifies Π^B over $\mathcal{GC}$. Given an instance specification π^B of Π^B and a graph $G \in \mathcal{GC}$, do the following. Set $\pi^A = \Pi^A(G)$. Then $\pi^A \wedge \pi^B$ is an instance of the composed profile type $\Pi^A \wedge \Pi^B$. Invoke algorithm $ALG_{A\wedge B}$ to $\pi^A \wedge \pi^B$. If its response is a graph G', then return "yes", and if the response is "impossible", then return "no". Indeed, in the former case, the returned graph G' must equal the given G, implying that G satisfies π^B, and in the latter case, necessarily G (which is the only possible realization of π^A over $\mathcal{GC}$) has $\Pi(G) \neq \pi^B$, i.e., it fails to satisfy π^B. □

3 Composing the Distance Profile

We next focus on compositions where one of the profile types is distances. The following algorithm for precise distance realization by unweighted graphs has been presented as part of the proof of [27, Theorem 7].

Algorithm 1

for every pair (i, j) **do** ▷ Construction
  if $D_{i,j} = 1$, **then** add an edge (i, j) to G.
end for
for every pair (i, j) **do** ▷ Verification
  if $dist_G(i, j) \neq D_{i,j}$ **then return** "Impossible" and halt.
end for
return G.

Hereafter, we refer to the constructed graph as the *base graph* of D, and denote it by G_{base}. The next lemma is implicit in the proof of Thm. 7 of [27].

Lemma 3 [27]. *If D is realizable, then G_{base} is its unique realization.*

Note that in particular, the lemma implies the following.

Corollary 1. *The precise distance profile is unique over unweighted graphs.*

Hence by Lemma 2, we immediately have the following for the precise distance profile DIST.

Corollary 2. *For every profile Π, the composed profile $DIST \wedge \Pi$ is realizable by unweighted graphs if and only if Π is verifiable over unweighted graphs.*

We next consider profile compositions where one of the two profiles is the distance profile and the realization is by weighted graphs. The following algorithm solves the distance realization problem in the weighted case [27].

Algorithm 2

```
Initially set V = {1,...,n} and E = ∅.   ▷ At this stage dist_G(i,j) = ∞ for every i,j.
Sort the vertex pairs (i,j) for 1 ≤ i < j ≤ n by nondecreasing distances D_{i,j}.
 Go over the pairs in this order.
for every pair (i,j) do                                    ▷ Construction
    if dist_G(i,j) > D_{i,j} then add an edge (i,j) of weight D_{i,j} to E.
end for
for every pair (i,j) do                                    ▷ Verification
    if dist_G(i,j) < D_{i,j} then return "Impossible" and halt.
end for
Return G.
```

The algorithm's correctness follows from the next lemma, also due to [27].

Lemma 4 [27]**.** *If the precise distance specification D is realizable by a weighted graph, then the output G of Algorithm 2 is a minimal realization of D, i.e., every realization of D must contain the edges of G.*

Lemma 5. *For every profile type Π on unweighted graphs, the composed profile type $DIST \wedge \Pi$ is realizable on weighted graphs if and only if Π is super-realizable.*

Proof. Suppose Π is super-realizable on unweighted graphs and let ALG_π be the realization algorithm. We have to show that the profile type $\mathrm{DIST} \wedge \Pi$ is realizable on weighted graphs. The following algorithm $ALG_{D\wedge\pi}$ solves the $\mathrm{DIST} \wedge \Pi$-realization problem. Consider a specification $D \wedge \pi$. The algorithm first applies Algorithm 2 to the precise distance specification D. If the response is "Impossible", then return "Impossible". If the response is a weighted graph $G_{\min}(D) = (V, E_{\min})$ realizing D, then invoke Algorithm ALG_π on the graph $G_{\min}(D)$ (ignoring the weights). If the response of ALG_π is "Impossible", then return "Impossible". If the response is an unweighted graph G_π then return the weighted graph $G_{\pi,D}$ which is $G_\pi = (V, E_\pi)$ where the weight of every edge $(i,j) \in E_\pi$ is $D_{i,j}$. Note that if the algorithm returns "Impossible" after the execution of Algorithm 2, the specification $D \wedge \pi$ is indeed unrealizable (since D is unrealizable). Now consider the case where D is realizable. By Lemma 4, all of the edges of the returned graph $G_{\min}(D)$ are necessary in any realization of D. Therefore, any realization of $D \wedge \pi$ must be a supergraph of $G_{\min}(D)$. If an unweighted supergraph G_π of $G_{\min}(D)$ realizing π exists, then it will be found by Algorithm ALG_π (and if not, both ALG_π and $ALG_{D\wedge\pi}$ return "Impossible"). Note that by definition of a supergraph $E_{\min} \subseteq E_\pi$. Since The weights assigned to the edges of $G_{\pi,D}$ are all from the specification matrix D, if $G_{\min}(D)$ is a weighted realization of D, then so is $G_{\pi,D}$. It follows that $ALG_{D\wedge\pi}$ returns the correct answer.

Conversely, suppose DIST $\wedge$ Π is realizable on weighted graphs, and let $ALG_{D\wedge\pi}$ be the realization algorithm. We have to show that Π is super-realizable on unweighted graphs. The following algorithm ALG_{π} solves the Π-realization problem on supergraphs of a given unweighted graph. Consider a specification π of Π, and let G be the given unweighted graph. Let $D_G = \mathrm{DIST}(G)$, the distance matrix of G. Note that $G_{\min}(D_G) = G$, since G is unweighted, so $G_{\min}(D_G)$ consists only of the distance-1 entries of D_G, which are exactly the edges of G. Create the composed profile specification $D_G \wedge \pi$, apply Algorithm $ALG_{D\wedge\pi}$ to this specification, and return its response. Note that any super graph of $G_{\min}(D_G)$ can be a realization of D_G where the weight of every edge is exactly its corresponding distance in D_G. Therefore, there exists a realization of the composed profile specification $D_G \wedge \pi$ on weighted graphs if and only if π is realizable by a supergraph of G. Hence, the output of $ALG_{D\wedge\pi}$ is the correct response for ALG_{π}. $\square$

Note that these lemmas hold unconditionally, i.e., their proof does not rely on $NP \neq P$.

4 D&D Realizations

In this section, we illustrate profile composition by presenting our results concerning the concrete example of composing *degree and distance* profiles.

4.1 Precise Distance D&D Realizations

We begin with the problems that involve *precise* distance profiles. These variants turn out to be easy in general graphs and in specific graph classes.

Theorem 1. *Given a precise distance matrix (i.e., $\tilde{D} = P$), for both weighted and unweighted realizing graphs (i.e., $\tilde{g} \in \{U, W\}$), and for both precise degrees and degree-ranges (i.e., $\tilde{\delta} \in \{P, [\,]\}$), the composed D&D realization problem D&D$(\tilde{\delta}, P, \tilde{g})$ is solvable in polynomial time.*

Proof. We present our algorithms for degree-range sequences, which in particular solve also instances of precise degrees. In other words, our algorithms assume instances where the degree sequence $\bar{\delta} = ([\delta_1^-, \delta_1^+], \ldots, [\delta_n^-, \delta_n^+])$ is given with ranges and the distance matrix D is given with exact values. We start with the unweighted problem: D&D$([\,], P, U)$. A degree sequence $\bar{\delta}$ can be verified on a specific graph in linear time. Hence, by Corollary 2, the composed D&D$([\,], P, U)$ problem is realizable in polynomial time.

We now turn to show a polynomial-time algorithm for the weighted problem: D&D$([\,], P, W)$. According to Lemma 5, we need to show that finding a realization of a degree sequence which is lower bounded by a graph G can be done efficiently. Let $\bar{\delta}^0 = (\delta_1^0, \ldots, \delta_n^0)$ be the degree sequence of G, and let $f = (f_1, \ldots, f_n)$, where $f_i = \delta_i^+ - \delta_i^0$ for every $1 \leq i \leq n$, is the *degree shortage* vector. There are two cases to consider. If $f_i < 0$ for some $1 \leq i \leq n$, then there is no realizing graph for $\bar{\delta}$ which is lower bounded by G. Else, $f_i \geq 0$ for all $1 \leq i \leq n$. In this case, let $\bar{G} = K_n \backslash G$, where K_n is the complete graph on n vertices. Edges of $\bar{G}$ can now be added to G in order to increase some of the degrees. Let $g = (g_1, \ldots, g_n)$, where $g_i = \max\left\{0, \delta_i^- - \delta_i^0\right\}$. The problem now reduces to that of finding a (g, f)-factor in $\bar{G}$, which is known to be solvable in polynomial time [2]. □

(Hereafter, most proofs are omitted due to lack of space.)

Theorem 2. *Given a precise distance matrix (i.e., $\tilde{D} = P$), for weighted or unweighted tree or bipartite realizing graphs (i.e., $\tilde{g} \in \{UT, WT, UB, WB\}$), as well as for weighted or unweighted subgraphs of a given graph (i.e., $\tilde{g} \in \{US, WS\}$), and for both precise degrees and degree-ranges (i.e., $\tilde{\delta} \in \{P, [\ ]\}$), the following realization problems are all solvable in polynomial time:*

- *The distance realization problems DIST$(P, \tilde{g})$,*
- *The composed D&D realization problems D&D$(\tilde{\delta}, P, \tilde{g})$.*

4.2 Distance-Range D&D Realizations

We next consider problems involving distance-range profiles. As mentioned earlier, a polynomial-time algorithm for distance-range realization (not composed with degrees) by a weighted graph was given in [33,39]. However, no algorithm was known for the unweighted version. We now show that this is no coincidence: testing distance-range realizability by an unweighted graph is NP-hard. The proof method serves also (with small variations) for deriving hardness results for some of the D&D realization problems discussed later on.

Theorem 3. *The unweighted distance-range realization problem DIST$([\ ], U)$ is NP-hard.*

Proof. We prove that the problem DIST$([\ ], U)$ is NP-hard by a reduction from the coloring problem. Consider an instance (G, k) of the coloring problem, namely, an unweighted undirected graph G and an integer k, where it is required to decide whether G can be legally colored with k or fewer colors. We reduce this instance to an instance D of DIST$([\ ], U)$ defined as follows. D is a distance matrix for $n+k+1$ vertices, $\{1, \ldots, n+k+1\}$. Intuitively, we think of the first n vertices of D, $V_{orig} = \{1, \ldots, n\}$, as representing the *original* n vertices of the given graph G, and of additional k vertices of D, $V_{col} = \{n+1, \ldots, n+k\}$, as representing the *colors*, and the last vertex $n+k+1$ represents a *coordinator*.

First, we impose the following requirements on the color vertices and the coordinator. Let $D_{n+\ell, n+k+1} = 1$ for every $1 \leq \ell \leq k$, and $D_{n+\ell, n+t} = 2$ for every $1 \leq \ell < t \leq k$. This allows only one possible realization for the subgraph induced by the vertices $V_{col} \cup \{n+k+1\}$, namely, a star rooted at $n+k+1$ (with no edges between the leaves).

Next, for the connections between the original vertices and the star structure, define the distance constraints as follows. For every $1 \leq i \leq n$, let $D_{i, n+k+1} = 2$

and let $D_{i,n+\ell} = [1,3]$ for every $1 \le \ell \le k$. This forces each of the original vertices to be connected to one (or more) of the color vertices, but not to the coordinator.

Finally, for every two original vertices $1 \le i < j \le n$, let

$$D_{i,j} = \begin{cases} 4, & (i,j) \in E(G), \\ [1,4], & \text{otherwise.} \end{cases}$$

It remains to show that the input G is k-colorable if and only if the distance matrix D is realizable.

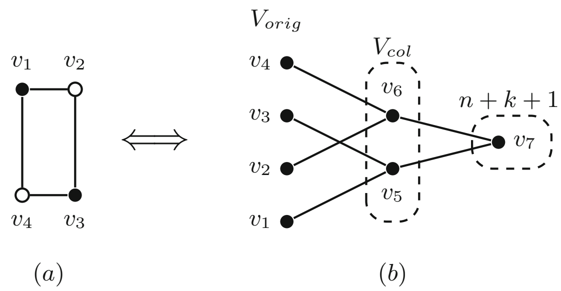

Fig. 1. An example of the reduction in the proof of Theorem 3, for $n = 4$ and $k = 2$. (a) The graph G is 2-colorable. (b) A realization of D which is the result of the reduction from the 2-coloring problem on G.

(⇒): Suppose G is k colorable. Identify the colors as $1, \ldots, k$ and let $\varphi : V \mapsto \{1, \ldots, k\}$ be the coloring function. For the matrix D defined from G, construct a realizing graph G as follows. Start with a star rooted at $n+k+1$ with the k color vertices $n+1, \ldots, n+k$ as leaves. Connect each original vertex i to the color vertex $\varphi(i)$. It is easy to verify that G is a valid realization for D (see Fig. 1 for an example).

(⇐): Suppose D has a valid realization G. Note that the restrictions of D force the color vertices to form a star rooted at $n+k+1$. Note that every original vertex i must be connected to at least one of the leaves of that star. Define a coloring function for G as follows. For every original vertex i, let ℓ be some color vertex connected to i, and let $\varphi(i) = \ell$. The distance constraints defined for the original vertices specify that if two original vertices i and j are connected by an edge in G, then their distance must be 4. This ensures that none of the color vertices are connected to both i and j (as this would make their distance 2). It follows that if $(i,j) \in E$ then i and j are assigned different colors.

This establishes the correctness of the reduction and shows that DIST$([\,], U)$ is NP-hard. □

As shown next, all variants of the D&D realization problem are hard both in general graphs and in specific graph classes.

Theorem 4. *Given a distance matrix with distance-ranges (i.e., $\tilde{D} = [\,]$), for both weighted and unweighted realizing graphs (i.e., $\tilde{g} \in \{U, W\}$), and for both precise degrees and degree-ranges (i.e., $\tilde{\delta} \in \{P, [\,]\}$), the composed D&D realization problem $D\&D(\tilde{\delta}, [\,], \tilde{g})$ is NP-hard.*

Theorem 5. *Given a distance matrix with distance-ranges (i.e., $\tilde{D} = [\,]$), for weighted and unweighted general, tree and bipartite realizing graphs (i.e., $\tilde{g} \in \{UT, WT, UB, WB\}$), as well as for weighted and unweighted subgraphs of a given graph (i.e., $\tilde{g} \in \{US, WS\}$), and for both precise degrees and degree-ranges (i.e., $\tilde{\delta} \in \{P, [\,]\}$), the following distance realization problems are NP-hard:*

- *The distance realization problems $DIST([\,], \tilde{g})$,*
- *the composed D&D realization problems $D\&D(\tilde{\delta}, [\,], \tilde{g})$.*

Note that the difference between our results for arbitrary realizing graphs and specific graph classes is for the weighted distance-range realization problem. Specifically, a polynomial algorithm exists in the general case, but when the realizing graph is required to be one of the structures we consider, this problem becomes NP-hard.

One of our results for trees is that the weighted distance-range realization problem, DIST($[\,], WT$), is NP-hard. It is interesting to note that in fact, this problem is NP-hard even if the realizing graph is required to be a simple path (denoted DIST($[\,], Wpath$)), but when the realizing graph is required to be a star (denoted DIST($[\,], Wstar$)), the problem becomes polynomial, as shown in the following two theorems.

Theorem 6. *The distance-range realization problem on weighted paths, $DIST([\,], Wpath)$, is NP-hard.*

Theorem 7. *There exists a polynomial-time algorithm for the distance-range realization problem on weighted stars, $DIST([\,], Wstar)$.*

5 Conclusions and Open Problems

This paper initiates the study of composed profiles and their realization, so naturally, many interesting research questions are ignored or touched upon only cursorily. Let us briefly mention some of those. We focused on the questions of deciding the realizability of a given specification, and generating a realizing graph if one exists. Some equally important questions, studied in the literature for various profile types, involve determining the *number* of different realizations, efficiently generating all realizations, establishing conditions for the existence of a unique realization, and so on. Our study of profile composition assumed that the two given specifications are *aligned*, i.e., vertex i in π^A is the same as vertex i in π^B. A different set of problems arises when we decouple the two specifications. In the context of D&D degree and distance profile composition, we may consider a graph G over $V = \{1, \ldots, n\}$ as an acceptable realization for the two given specifications if there exists a permutation π such that $dist_G(i, j) = D_{i,j}$ for every $1 \leq i < j \leq n$ and $\deg_G(\pi(i)) = \delta_i$ for every $1 \leq i \leq n$.

References

1. Althöfer, I.: On optimal realizations of finite metric spaces by graphs. Discret. Comput. Geom. **3**(2), 103–122 (1988). https://doi.org/10.1007/BF02187901
2. Anstee, R.P.: An algorithmic proof of Tutte's f-factor theorem. J. Algorithms **6**(1), 112–131 (1985)
3. Asano, T.: Graphical degree sequence problems with connectivity requirements. In: Ng, K.W., Raghavan, P., Balasubramanian, N.V., Chin, F.Y.L. (eds.) ISAAC 1993. LNCS, vol. 762, pp. 38–47. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57568-5_233
4. Baldisserri, A.: Buneman's theorem for trees with exactly n vertices. CoRR (2014)
5. Bandelt, H.: Recognition of tree metrics. SIAM J. Discret. Math. **3**(1), 1–6 (1990)
6. Bar-Noy, A., Böhnlein, T., Peleg, D., Perry, M., Rawitz, D.: Relaxed and approximate graph realizations. In: 32nd IWOCA (2021)
7. Bar-Noy, A., Choudhary, K., Cohen, A., Peleg, D., Rawitz, D.: Minimum neighboring degree realization in graphs and trees. In: 28th ESA, vol. 173 of LIPIcs, pp. 10:1–10:15 (2020)
8. Bar-Noy, A., Choudhary, K., Peleg, D., Rawitz, D.: Realizability of graph specifications: characterizations and algorithms. In: Lotker, Z., Patt-Shamir, B. (eds.) SIROCCO 2018. LNCS, vol. 11085, pp. 3–13. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01325-7_1
9. Bar-Noy, A., Choudhary, K., Peleg, D., Rawitz, D.: Graph profile realizations and applications to social networks. In: Das, G.K., Mandal, P.S., Mukhopadhyaya, K., Nakano, S. (eds.) WALCOM 2019. LNCS, vol. 11355, pp. 3–14. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-10564-8_1
10. Bar-Noy, A., Choudhary, K., Peleg, D., Rawitz, D.: Graph realizations: maximum degree in vertex neighborhoods. In: 17th SWAT, vol. 162 of LIPIcs, pp. 10:1–10:17 (2020)
11. Bar-Noy, A., Peleg, D., Perry, M., Rawitz, D., Schwartz, N.L.: Distance realization approximations. In: Preparation (2021)
12. Behzad, M., Simpson, J.E.: Eccentric sequences and eccentric sets in graphs. Discrete Math. **16**(3), 187–193 (1976)
13. Buneman, P.: A note on the metric properties of trees. J. Comb. Theory B **17**, 48–50 (1974)
14. Burstein, D., Rubin, J.: Sufficient conditions for graphicality of bidegree sequences. SIAM J. Discrete Math. **31**, 50–62 (2017)
15. Chung, F.R.K., Garrett, M.W., Graham, R.L., Shallcross, D.: Distance realization problems with applications to internet tomography. J. Comput. Syst. Sci. **63**(3), 432–448 (2001)
16. Culberson, J.C., Rudnicki, P.: A fast algorithm for constructing trees from distance matrices. Inf. Process. Lett. **30**(4), 215–220 (1989)
17. Dahlhaus, E.: Fast parallel recognition of ultrametrics and tree metrics. SIAM J. Discret. Math. **6**(4), 523–532 (1993)
18. Dress, A.W.M.: Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: a note on combinatorial properties of metric spaces. Adv. Math. **53**, 321–402 (1984)
19. Edmonds, J.: Existence of k-edge-connected ordinary graphs with prescribed degrees. J. Res. Natl. Bur. Stand. **68B**(2), 73–74 (1964)

20. Feder, T., Meyerson, A., Motwani, R., O'Callaghan, L., Panigrahy, R.: Representing graph metrics with fewest edges. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 355–366. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36494-3_32
21. Frank, A.: Augmenting graphs to meet edge-connectivity requirements. SIAM J. Discrete Math. **5**, 25–43 (1992)
22. Frank, A.: Connectivity augmentation problems in network design. In: Mathematical Programming: State of the Art, pp. 34–63. University of Michigan (1994)
23. Frank, H., Chou, W.: Connectivity considerations in the design of survivable networks. IEEE Trans. Circuit Theory, CT **17**, 486–490 (1970)
24. Gale, D.: A theorem on flows in networks. Pac. J. Math. **7**, 1073–1082 (1957)
25. Gomory, R., Hu, T.: Multi-terminal network flows. J. Soc. Ind. Appl. Math. **9**, 551–570 (1961)
26. Gupta, G., Joshi, P., Tripathi, A.: Graphic sequences of trees and a problem of Frobenius. Czechoslovak Math. J. **57**, 49–52 (2007)
27. Hakimi, S.L., Yau, S.S.: Distance matrix of a graph and its realizability. Q. Appl. Math. **22**, 305–317 (1965)
28. Imrich, W., Simões-Pereira, J.M.S., Zamfirescu, C.: On optimal embeddings of metrics in graphs. J. Comb. Theory Ser. B **36**(1), 1–15 (1984)
29. Lesniak, L.: Eccentric sequences in graphs. Periodica Mathematica Hungarica **6**(4), 287–293 (1975)
30. Nieminen, J.: Realizing the distance matrix of a graph. J. Inf. Process. Cybern. **12**(1/2), 29–31 (1976)
31. Patrinos, A.N., Hakimi, S.L.: The distance matrix of a graph and its tree realizability. Q. Appl. Math. **30**, 255 (1972)
32. Rao, S.B.: A survey of the theory of potentially P-graphic and forcibly P-graphic degree sequences. In: Rao, S.B. (ed.) Combinatorics and Graph Theory. LNM, vol. 885, pp. 417–440. Springer, Heidelberg (1981). https://doi.org/10.1007/BFb0092288
33. Rubei, E.: Weighted graphs with distances in given ranges. J. Classif. **33**, 282–297 (2016)
34. Ryser, H.: Combinatorial properties of matrices of zeros and ones. Can. J. Math. **9**, 371–377 (1957)
35. Simões-Pereira, J.M.S.: A note on the tree realizability of a distance matrix. J. Comb. Theory B **6**, 303–310 (1969)
36. Simões-Pereira, J.M.S.: A note on distance matrices with unicyclic graph realizations. Discret. Math. **65**, 277–287 (1987)
37. Simões-Pereira, J.M.S.: An optimality criterion for graph embeddings of metrics. SIAM J. Discret. Math. **1**(2), 223–229 (1988)
38. Simões-Pereira, J.M.S.: An algorithm and its role in the study of optimal graph realizations of distance matrices. Discret. Math. **79**(3), 299–312 (1990)
39. Tamura, H., Sengoku, M., Shinoda, S., Abe, T.: Realization of a network from the upper and lower bounds of the distances (or capacities) between vertices. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2545–2548. IEEE (1993)
40. Varone, S.C.: A constructive algorithm for realizing a distance matrix. Eur. J. Oper. Res. **174**(1), 102–111 (2006)

41. Wang, D., Kleitman, D.: On the existence of n-connected graphs with prescribed degrees ($n >$). Networks **3**, 225–239 (1973)
42. Zaretskii, K.A.: Constructing a tree on the basis of a set of distances between the hanging vertices. Uspekhi Mat. Nauk **20**, 90–92 (1965)
43. Zverovich, I.E., Zverovich, V.E.: Contributions to the theory of graphic sequences. Discret. Math. **105**(1–3), 293–303 (1992)