Relaxed and Approximate Graph Realizations

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

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

Abstract. A network realization problem involves a given specification π for some network parameters (such as vertex degrees or inter-vertex distances), and requires constructing a network G that satisfies π, if possible. In many settings, it may be difficult or impossible to come up with a precise realization (e.g., the specification data might be inaccurate, or the reconstruction problem might be computationally infeasible). In this expository paper, we review various alternative approaches for coping with these difficulties by relaxing the requirements, discuss the resulting problems and illustrate some (precise or approximate) solutions.

1 Introduction

1.1 Background: Precise (pure) Network Realization Problems

Network realization problems are fundamental questions pertaining to the ability to construct a network conforming to pre-defined requirements. Given a *specification* (or *information profile*) detailing constraints on some network parameters, such as the vertex degrees, distances or connectivity, it is required to construct a network abiding by the specified profile, i.e., satisfying the requirements, or to determine that no such network exists.

Realization problems may arise in two general types of contexts. In scientific contexts, the information profile may consist of the outcomes of some measurements obtained from observing some real world network (e.g., social networks and information networks) whose full structure is unknown. In such a setting, our goal is to construct a model that may possibly explain the empirical observations. Many of the studies in the field of phylogenetics and evolutionary trees (see, e.g., [16,23,41,42,51,63,68,84,88]) as well as in the field of discrete tomography and microscopic image reconstruction (see, e.g., [3–5,11,15,46,47,52,64,77]) belong to this class.

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

P. Flocchini and L. Moura (Eds.): IWOCA 2021, LNCS 12757, pp. 3–19, 2021.
https://doi.org/10.1007/978-3-030-79987-8_1

A second, engineering-related context where realization problems come up is network design. Here, the profile may be defined based on a specification dictated by the future users of the network, and the goal is to construct a network that obeys the specification. For example, the profile may specify the required connectivity, flow capacities, or distances between vertex pairs in the network. In particular, network realization techniques may be useful in the area of *software defined networks* (SDN). For example, in service chain placement, the specification can define a directed acyclic graph (DAG) of *virtual network functions* (VNF), and the realization must determine the placement of one of the paths of the DAG in the physical network [38,39,72].

Two of the most well-studied families of realization problems concern *vertex degrees* and *inter-vertex distances*. The following is a brief review of the literature on these two problems.

Degree Profile. The *degree sequence* of a simple (no parallel edges or self-loops) and undirected graph $G = (V, E)$ with the vertex set $V = \{1, \ldots, n\}$ is an integer sequence $\mathrm{DEG}(G) = (d_1, \ldots, d_n)$, where $d_i = \deg_{G,i}$ is the degree of vertex i in G. The degree sequence occurs as a central and natural parameter in many network applications, and provides information on the significance, centrality, connectedness, and influence of each vertex in the network, contributing to the understanding of the network structure and some of its important properties. Given a sequence D of n non-negative integers, the *degree realization* problem asks to decide whether there exists a graph G whose degree sequence $\mathrm{DEG}(G)$ equals D. A sequence admitting such a realization is called *graphic* (or *graphical*).

The two key questions studied extensively in the past concern identifying characterizations (or, necessary and sufficient conditions) for a sequence to be graphic, and developing effective and efficient algorithms for finding a realizing graph for a given sequence if exists. A necessary and sufficient condition for a given sequence of integers to be graphic (also implying an $O(n)$ decision algorithm) was presented by Erdös and Gallai in [36]. (For alternative proofs see [2,27,32,90–92].) Havel [60] and Hakimi [55] (independently) described an algorithm that given a sequence of integers computes in $O(m)$ time an m-edge graph realizing it, or proves that the given sequence is not graphic.

A number of related questions were considered in the literature, including the following: (a) Given a degree sequence D, find all the (non-isomorphic) graphs that realize it. (b) Given D, count all its (non-isomorphic) realizing graphs. (c) Given D, sample a random realization as uniformly as possible. (d) Determine the conditions under which a given D defines a unique realizing graph (a.k.a. the *graph reconstruction* problem). These questions are well-studied, cf. [27,36,55, 60,67,78,85,90,94,99,101], and have found several interesting applications, most notably in network design, randomized algorithms, and recently in the study of social networks [17,30,35,75] and chemical networks [89]. Degree realization with given constraints on some vertex was studied in [69]. For surveys on graphic sequences see [95–97].

The *subgraph realization* problem adds the restriction that the realizing graph must be a subgraph (or *factor*) of some fixed input graph. Subgraph realization problems are generally harder. For instance, it is easy to construct an n-vertex

connected graph whose degrees are all 2, but the same problem for subgraph-realization is NP-hard (being essentially the Hamiltonian cycle problem). For more on this interesting line of work see, e.g., [8,53,61,74,93]. Here we focus on the case where the host graph is the complete graph.

Graphic sequences were studied also on specific graph families. The problem is straightforward on trees (see [54] for an elegant proof) but of interest on more complex classes. For example, characterizations were given to sequence pairs that can be realized as the degree sequences of a *bipartite* graph [21,49,82,102]. The family of *planar* graphs was studied to some extent, but the degree realization problem in this setting is still far from being resolved, and existing results are restricted to characterizing planar graphic *k-sequences* (in which the difference between $\max d_i$ and $\min d_i$ is at most k) for $k = 0, 1, 2$ [1,83]. *Split* graphs and *difference* graphs are considered in [58,98] and in [57], respectively. *Chordal*, *interval*, and *perfect* graphs were studies in [25]. Degree realization in *directed* graphs was studied in [7,24,37,48,71], and the NP-hardness of degree realization by directed acyclic graphs was proved in [59].

Distance Profile. In a graph G, define the *distance* $\text{dist}_G(u, v)$ between two vertices u and v as the length of the shortest path connecting them in G. In the *distance realization* problem, the input DIST-profile consists of a matrix $D \in (\mathbb{N} \cup \{\infty\})^{n \times n}$, which specifies the required distance between every two vertices $i \neq j$ in the graph. $D_{i,j} = \infty$ represents the case where i and j are in different disconnected components. The goal is to compute a realizing graph G (if exists), such that any two vertices i, j in G satisfy $\text{dist}_G(i, j) = D_{i,j}$.

In the *unweighted distance realization* problem it is assumed that each edge is of length 1. It follows that the graph is fully determined by D: there is an edge (i, j) in the graph if and only if $D_{i,j} = 1$. It follows that there is only one graph that may serve as a realization. This was observed by Hakimi and Yau [56], who provided a characterization for distance specifications realizable by unweighted graphs, implying also a polynomial-time algorithm for distance realization.

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.) Hakimi and Yau [56] also studied the weighted problem. They proved that the necessary and sufficient condition for the realizability of a given martix D is that D is a metric. Furthermore, they gave a polynomial-time algorithm that given a distance specification D, which is a metric, computes a realization. More specifically, their algorithm constructs a unique realizing graph which contains edges that are necessary in every possible realization of D.

Patrinos and Hakimi [81] considered the case where weights can be negative. They showed that any symmetric matrix with zero diagonal is a distance matrix of some graph G. They gave necessary and sufficient conditions for realizing such a matrix as a tree, and they showed that the tree realization is unique.

Precise distance realization by *weighted trees* were considered in [9], which presented a characterization for realizability. (For unweighted trees, there is a straightforward realization algorithm, based on the algorithm of [56] for general unweighted graphs, and on the fact that the realization, if exists, is unique.) Precise distance realization restricted to bipartite graphs was studied in [14],

where it was observed that it is sufficient to check the unique realization in the unweighted case or the minimal realization in the weighted case.

1.2 Limitations of Precise Realizations

Most of the research activity on realizations dealt with precise realizations in which the specification must be met *exactly*, in terms of the graph size and the attributes of each vertex in the realizing graph. We refer to such realizations as pure realizations because they forbid any deviation or relaxation from the specifications. Unfortunately, pure realizations suffer from certain limitations, making them hard to utilize, and motivating the use of *relaxed* versions of network realization. We now discuss several of these limitations.

Hardness of Pure Realization. Our ability to solve a specific realization problem, i.e., construct networks satisfying a given specification, depends heavily on the attributes considered. While some profile types are handled easily, for other certain profile types it turns out to be very hard to derive characterizations for realizability or construction algorithms for pure realizations. For example, the realizability of vertex connectivity profiles is to date poorly understood. Moreover, for certain types of information profiles, it may be feasible to find characterizations or construction algorithms for general graphs, but hard to do so for some specific graph classes. For example, degree realizability is well understood on general graphs, but not on planar graphs. In such cases, it makes sense to resort to various relaxations, or to realizations that well-approximate the given specification. Specifically, efficient approximation algorithms for various hard to realize profiles may turn out to have practical significance in the context of network design, as it may provide designers with tools for constructing networks that obey a detailed pre-specified behavior in terms of the attributes most relevant to the purposes for which the network is designed.

Flexibility of Imprecise Realizations. When solutions are restricted to precise realizations, it could be the case that the space of all possible realizations may be too small or even empty. As a result, there is very little freedom in selecting a realization that optimizes some objective functions. By relaxing the specifications the space of all feasible realizations might become large enough to produce optimal or near optimal realizations in regard to predefined optimization goal. Consider for example the precise degree realization problem. By allowing the degrees of vertices to deviate by a constant number from the precise degree, one could find a realizing graph with a smaller diameter and/or higher connectivity. Now consider the precise distance realization problem. By allowing the actual distances to be larger by some factor, one could find a realizing graph with fewer edges and smaller degrees.

Imperfect Data. In certain cases, the input specification data is imprecise (e.g., describing the results of inaccurate measurements on a real-life network with unknown parameters), or ill-chosen (e.g., based on the unrealistic expectations of clients describing their "dream network"). In such cases, it might well

happen that the specification is infeasible with no network conforming to it. As a result, characterization tests and construction algorithms will simply return "non-realizable" and halt. This might be an unsatisfactory outcome in the practical world. As an alternative, we could aim for a more positive outcome, in the form of an actual realizing graph, if only approximate.

Hardness of Composition. In some settings we may be given a *composed* profile, combining two or more profiles that must be realized simultaneously (see [13] in these proceedings). As one might suspect, handling profile compositions may sometimes be significantly harder, especially when insisting on pure realizations. Hence for such composed profiles, finding relaxed or approximate realizations might often be our only recourse.

In short, the above discussion makes it evident that in many situations one may be interested in considering some *alternative* approaches as a substitute for pure realizations. In this expository paper we review and illustrate various alternative approaches to network realization, based on relaxed or approximate network realizations. Specifically, Sect. 2 gives an overview of relaxed realizations, Sect. 3 illustrates the notion of minimum deviation distance realizations and some related results, and Sect. 4 discusses *multigraph* realizations of degree profiles with minimum multiplicity.

2 Relaxed and Approximate Realizations

We next overview a number of variations studied in the literature, which introduce flexibility in one way or another. For simplicity, we focus only on attributes of vertices or vertex pairs.

Bounding Specifications. The specifications considered so far were *precise*, i.e., they specified the exact value of the attribute required for every vertex (or vertex pair). In contrast, a *lower-bounding* specification ϕ for an information profile f (f-profile) provides lower bounds on the required attributes. Such ϕ is realizable if there is a graph G whose f-profile satisfies $f(G) \geq \phi$. More explicitly, when ϕ and f concern *vertex* attributes of n-vertex graphs, G satisfies ϕ if $f_i \geq \phi_i$ for every $1 \leq i \leq n$ (a similar definition applies for attributes of vertex pairs). An analogue notion can be defined for *upper-bounding* specifications, as well as *range* specifications, specifying both lower and upper bounds on each attribute. Clearly, bounding specifications are more flexible than precise ones; whereas precise specifications sometimes (although certainly not always) force a unique realization, bounding specifications are more likely to admit many realizing graphs.

As an example, a natural generalization of the graphic sequence problem is the *degree range* profile. In this variant we are given a *range sequence* consisting of n ranges, $\mathcal{S} = ([a_1, b_1], \ldots, [a_n, b_n])$ such that $0 \leq a_i \leq b_i \leq n-1$ for every i, which is said to be realizable if there exists a graphic sequence $D = (d_1, \ldots, d_n)$ falling within the specified ranges, namely, such that $a_i \leq d_i \leq b_i$ for $1 \leq i \leq n$. (cf. [12,22,50,53] and the references therein). Two natural questions are to find

an efficient algorithm for verifying the realizability of a given range sequence $\mathcal{S}$, and to compute a certificate (that is, a suitable graphic sequence D) for a given realizable range sequence $\mathcal{S}$. (Given D, one can readily construct a graph whose degrees realize D, as discussed earlier.) An easy to verify characterization for realizable range sequences is provided in [22], crucially using the (g, f)-Factor Theorem of [74]. A constructive proof of this characterisation is given in [50]. Another algorithm for computing a graph whose degrees realize a given range sequence (if exists), based on [55,60], is presented in [62].

As another example, range variants of the distance realization problem, where we are given a range $[D^-_{i,j}, D^+_{i,j}]$ for every i, j and the realizing G must satisfy $D^-_{i,j} \leq \text{dist}_G(i,j) \leq D^+_{i,j}$, were studied in [14], and the resulting problems were classified as polynomial or NP hard.

Handling Unrealizable Specifications. When faced with a given *unrealizable* specification ϕ for some f-profile, a natural question that presents itself is to find a graph G that "resembles the specification most," namely, whose f-profile $f(G)$ "minimally deviates" from ϕ. This requirement is rather broad, and may lead to different optimization problems, depending on the setting. For our canonical example of degree realizations in general graphs, an interesting instantiation of this problem is the *minimum total discrepancy* degree realization problem [62,75], a.k.a. the *graphic deviation* problem [19]. It requires one to find, for a given non-graphic sequence D, a graph G whose degree sequence is closest to D.

Several measurements for closeness or the quality of an approximation are possible. As an example consider the unrealizable sequence $D_0 = (4, 4, 1, 1, 1)$. We may look for a realizing graph G that minimizes the *sum* of the deviations at all the vertices $\sum_i |\deg_{G,i} - d_i|$, i.e., with respect to the L_1-norm. The realizable sequence $(4, 4, 2, 2, 2)$ approximates D_0 with minimum total deviation 3.

Naturally, other norms may be of interest as well. In particular, the problem can be solved for the L_∞ norm (see Sect. 4) where the deviation measure to be minimized is $\max_i |\deg_{G,i} - d_i|$. A solution for D_0 is the sequence $(3, 3, 2, 1, 1)$, which is graphic and has maximum deviation 1. Beyond being a natural extension of the degree sequence problem, a practical motivation for this problem is raised and studied in [18,20], in the context of probabilistic and game-theoretic analyses of population models.

Another approach is to minimize the *number* of vertices that do not conform to their specification (are mismatches), that is, to find a realizing graph G minimizing the number of vertices i such that $\deg_{G,i} \neq d_i$. A possible solution for the example D_0 is the sequence $(4, 1, 1, 1, 1)$, which is graphic (the realizing graph being the 4-star) and has one mismatch.

Returning to the example of degree range profiles, for a non-realizable range sequence $\mathcal{S}$, the algorithm of [62] computes a graph whose deviation with respect to L_1-norm is minimum. The time complexity of the algorithm is $O(\sum_{i=1}^{n} b_i)$, where b_i is the upper limit of the ith range, which can be as high as $\Theta(n^2)$. In [12] we presented an $O(n \log n)$ time algorithm for computing a graphic certificate (if exists) for any given range sequence. In addition, given a range sequence $\mathcal{S}$, our algorithm can obtain (in the same time) a degree-certificate corresponding to

graphs with minimum (resp. maximum) possible edges. We also gave an $O(n \log n)$ time algorithm for efficiently computing graphic sequences having the least possible L_1-deviation when the input range sequence is non-realizable. Again, the problem can be solved also for the L_∞ norm, i.e., where the deviation measure to be minimized is $\max_i\{0, \deg_{G,i} - b_i, a_i - \deg_{G,i}\}$. These tools allow also studying other interesting applications, such as computing a minimum extension of non-graphic sequences to graphic ones.

Super-Realizations. It is sometimes acceptable to allow the realizing graph G to be *larger* than indicated by the specification (i.e., have more than n vertices). We refer to such a realizing graph as an *super-realization* of the given specification. The flexibility of adding new vertices is useful in tackling problems that are otherwise intractable. Consequently, super-realizations were extensively studied for various profile types.

As an example let us consider distance profiles. The *optimal distance realization* problem was introduced in [56]. In this problem, a distance matrix D is given over a set S of n *terminal* vertices, and the goal is to find a graph G including S, with possibly additional vertices, that realizes the given distance matrix for S. Necessary and sufficient conditions are given for a symmetric matrix with nonnegative entries to be realizable by a weighted or an unweighted graph. It is also shown that if G is an n-vertex realization of D without redundant edges (i.e., no edge can be removed without violating the distance matrix), then G is unique. It is shown in [34] that an optimal realization can have at most n^4 vertices (recall that the number of terminal vertices is n), and therefore, there is a finite (but exponential) time algorithm to find an optimal realization. In [6] it is shown that finding optimal realizations of distance matrices with integral entries is NP-complete, and evidence to the difficulties in approximating the optimal realization is provided in [28]. Over the years, various heuristics for optimal realizations are discussed in many papers [76,86,87,100].

Since optimal realization seems hard even to approximate, special cases and other variations have been studied. In [28], a *weak* realization is defined, where the distance matrix is a bounding specification that sets *lower bounds* on the required distances. it is shown that this weak realization problem is also NP-complete, but its optimum solution can be 2-approximated. In [40], an optimizing variant is defined to be the one with minimum *number of edges*. It is shown therein that if additional vertices are not allowed, then an edge-minimal graph can be found in polynomial time. On the other hand, in a setting allowing additional vertices, if the distance matrix has to be realized by an unweighted graph, then the problem is essentially as hard to approximate as graph coloring and maximum clique. In addition, polynomial approximation algorithms are presented for specific cases.

Special attention has been given to the optimal distance realization problem where the realizing graph is a tree. In [56], a procedure is given for finding a tree realization of D if exists. It is also shown therein that a tree realization, if exists, is unique and is the optimum realization of D. Necessary and sufficient conditions

for a distance matrix to be realizable by a tree were given in [10,33,87]. Finally, an $O(n^2)$ time algorithm for optimal tree-realization is given in [31].

For degree realizations, we may look for a realizing graph G having a subset of n vertices whose parameters conform precisely to the given profile. The typical goal would be to minimize the number of additional vertices, i.e., $|V(G)| - n$. This type of relaxation may be useful, e.g., when each component f_i of the specification f represents a vertex that *must* exist in the constructed network, and cannot be omitted, whereas extra vertices imply extra cost but are not prohibitive. For the above example of $D = (4, 4, 1, 1, 1)$, one can obtain a super-realization with $n' = 7 = n + 2$ by realizing the sequence $D' = (4, 4, 2, 1, 1, 1, 1)$, which is graphic. In general graphs, this type of approximate degree realization problem can be solved efficiently, using algorithms for degree range profiles.

Another option is to allow a realizing graph G that is *smaller* than indicated by the specification, namely, has fewer than n vertices. We call such a graph an *sub-realization* of the given specification. For example, for the profile DEG, the specification $D = (4, 3, 1, 1, 1)$ cannot be realized, but by removing one vertex we obtain $D' = (3, 1, 1, 1)$ that can be realized using a star. The natural goal in such a setting is to compute a sub-realization with maximum number of vertices, i.e., with minimum number of vertex deletions. This type of relaxation may arise, e.g., when external constraints or physical resource bounds limit the size of the constructed network to no more than n vertices, and it is desired to utilize the available resources to the extent possible.

Yet another interesting variant of super-realizations is obtained if we allow only *splitting* vertices. For example, $D = (4, 3, 1, 1, 1)$ cannot be realized, but $(3, 2, 2, 1, 1, 1)$, obtained by splitting the requirement 4 into 2 and 2, is realizable.

Optimizing Realizations. Optimization goals may be of interest also for *realizable* profiles. It is often the case that the given profile has many possible realizing graphs, and it may be of interest to seek a realizing graph that also optimizes some desirable quality measure. This type of optimization problem often arises in contexts where the specification is bounding rather than precise, since the flexibility of bounding specifications, resulting in the larger variety of realizing graphs, makes it attractive to select the most suitable realizing graph for the quality measure at hand.

For example, in the context of connectivity realization, a natural problem to consider is *minimum edge connectivity threshold realization*. The input for this problem is an $n \times n$ connectivity threshold matrix CT^e serving as a bounding, rather than precise, specification, namely, specifying the required *minimum* edge connectivity $CT^e(i, j)$ between every two vertices $i \neq j$ in the graph. Hence a graph G satisfies the specification if $\mathsf{e\text{-}conn}_G(i, j) \geq CT^e(i, j)$ for any two vertices i, j. Note that for any connectivity threshold matrix CT^e, the complete graph on V is always a legal realization (provided that $CT^e(i, j) \leq n-1$ for every i and j). However, the minimum edge connectivity threshold realization problem imposes an additional desirable goal, i.e., to find the *sparsest possible* realizing graph G. In [45], this problem was studied in the context of survivable network design, and a 2-approximate solution is provided for this problem, guaranteeing that the

number of edges is at most twice the minimum possible. See also [26,43,44,66] for related studies.

Another approach to relax the the degree realization problem is to ask for a multigraph (with parallel edges and self-loops) instead of a simple graph. Then it is natural to try to maximize (or minimize) the number of edges in the underlying graph, i.e., to try to minimize the number of mismatches which are parallel edges or self-loops in this case. We take a closer look at these variants in Sect. 4.

We note that all the above deviations also have two one-sized (i.e., bounding) variants. For example, in the case of minimizing the number of mismatches, a subclass of possible bounding realizations arises when we allow only $\deg_{G,i} \geq d_i$, or only $\deg_{G,i} \leq d_i$, depending on the application at hand.

3 Minimum Deviation Distance Realization

If a given distance matrix D is unrealizable, one may want to find a graph G whose distance matrix is closest to D, say, with respect to the L_1-norm, i.e., such that the sum of deviations of all matrix entries is minimized, or with respect to the L_∞-norm, i.e., the maximum deviation of a matrix entry is minimized. Note that here we allow both *downwards deviation*, where the actual distance satisfies $\text{dist}_G(i,j) < D_{i,j}$, and *upwards deviation*, where $\text{dist}_G(i,j) > D_{i,j}$. Other deviation functions can be considered, for example, the number of matrix entries for which there is a deviation. For each of the above deviation functions (which allow both downwards and upwards deviation) we may consider also two more variants: one allowing only downward deviations, and one allowing only upward deviations.

In this section we focus on the version of only downward deviations, which is particularly interesting for distance realizations, since in system design, we would like to get as close as possible to the specification matrix, but never exceed the specified distances.

There are several variants of the distance realization problem, depending on whether the distance matrix specifies exact values or ranges at each entry, and whether the realizing graph is required to be unweighted or weighted. For all three deviation functions mentioned above, it turns out that finding a graph G whose distance matrix is closest to D is NP-hard when G must be unweighted, and is polynomial for weighted graphs. This holds for both types of distance matrix, namely, precise distances and distance range.

As a first illustration we show a polynomial algorithm for the weighted case.

Theorem 1 ([14]). *For every deviation function and every distance matrix D, there is a polynomial time algorithm that finds a weighted graph G with minimum downward deviation from D.*

The theorem is established by presenting a polynomial time algorithm for the most general case of a distance-range matrix (a precise-distance matrix is a special case, for which our algorithm applies as well). Given a distance-range matrix D, construct a weighted clique graph G by assigning the weight

$\omega(i,j) = D^+_{i,j}$ for every edge (i,j). We now sketch the argument showing that G is a minimum downward deviation graph. First, observe that by construction, $\text{dist}_G(i,j) \leq D^+_{i,j}$ for every i and j, since there exists an edge of weight $D^+_{i,j}$ connecting i and j, so there are no upward deviations in G. Next, we argue that for every pair (i,j), and for every graph whose distance matrix deviates only downwards from D, the deviation of (i,j) is at least $D^-_{i,j} - \text{dist}_G(i,j)$, i.e., G is a minimum deviation graph. To see this, consider a path $i = v_0, \ldots, v_\ell = j$ in G for which the total weight is $\text{dist}_G(i,j)$. This path implies a set of requirements $D_{v_k,v_{k+1}}$, for $k = 0, \ldots, \ell - 1$, such that the distance between v_k and v_{k+1} is at most $D^+_{v_k,v_{k+1}} = \omega(v_k, v_{k+1})$. Since every realizing graph G' is not allowed to deviate upwards, the distance between i and j in G' is at most $\text{dist}_G(i,j)$, i.e., $D^-_{i,j} - \text{dist}_{G'}(i,j) \geq D^-_{i,j} - \text{dist}_G(i,j)$.

Our second complementary example is an NP-hardness result for the unweighted case.

Theorem 2 ([14]). *The problem of finding an unweighted graph G with minimum sum of downward deviations from D, is NP-hard.*

The hardness of this problem is established for the case of a precise-distance matrix, implying also the hardness of the distance-range case. Hardness is established by reducing the *diameter*-2 *augmentation* problem, which is known to be NP-hard [73], to our deviation problem. Given a graph $G = (V,E)$, input to the diameter-2 augmentation problem, the goal is to find the minimal set of edges E' such that the diameter of the graph $G' = (V, E \cup E')$ is at most 2. This instance is reduced to an instance D of the *minimum sum of downward deviations* problem by setting $D_{i,j}$ to be 0 if $i = j$, 1 if $(i,j) \in E(G)$, and 2 otherwise, yielding a distance matrix for n vertices. It then remains to verify that D has a realization with minimum sum of downwards deviations k if and only if G has a diameter-2 augmentation of k edges, establishing the theorem.

The same reduction actually proves also the hardness of *minimum number of downward deviations* problem. Moreover, one can prove that the problem *min-max downward deviation* is NP-hard by a reduction from the k-coloring problem. A systematic study of these types of minimum-deviation realizations for distance profiles, and their classification into polynomial and NP-hard cases, can be found in [14], where we also consider other deviation functions, such as multiplicative deviation, two-sided deviation and only upward deviation.

4 Optimizing Multigraph Realizations of Degree Profiles

In this section, we consider degree profiles where the realization must satisfy the given specification, but it is allowed to use *parallel edges*. Namely, we allow realizations by *loopless multigraphs* rather than by only simple graphs. The goal is to find a realization which is as "close" to a simple graph as possible.

Let $H = (V,E)$ be a (loopless) multigraph. That is, E is a multiset. Denote by $E_H(v,w)$ the multiset of edges connecting v and w in H, and let $\#_H(v,w) = |E_H(v,w)|$. Given a vertex u, let $N(u)$ be the neighborhood of u,

namely $N(u) = \{v : (u, v) \in E\}$. Also, let $E(u) = \{(u, v) \in E : v \in V\}$ be the multiset containing edges that are adjacent to u. Observe that $\deg(u) = |E(u)|$, while it could be that $\deg(u) > |N(u)|$. Finally, let $\mathcal{P}(H) = \{(v, w) \mid E(v, w) \neq \emptyset\}$. Observe that $(V, \mathcal{P}(H))$ is the underlying simple graph of the multigraph H.

Define the *total multiplicity* as the number of parallel edges $|E| - |\mathcal{P}(H)|$. It follows that

$$\mathsf{TotMult}(H) = \sum_{(v,w)\in\mathcal{P}(H)} (\#_H(v, w) - 1).$$

Observe that in a simple graph $\mathsf{TotMult}(H) = 0$.

A realization minimizing the total multiplicity can be computed efficiently (see [70,79,80]). In [79] this problem is also solved when only loops and loops and parallel edges are allowed. Interestingly, finding a multigraph realization that maximizes the number of parallel edges is NP-hard [65].

Define the *maximum multiplicity* as

$$\mathsf{MaxMult}(H) = \max_{(v,w)\in\mathcal{P}(H)} (\#_H(v, w)).$$

Observe that in a simple graph $\mathsf{MaxMult}(H) = 1$.

Theorem 3 (Chungphaisan [29]). *Consider a sequence $d = (d_1, \ldots, d_n)$ where $\sum_{i=1}^{n} d_i$ is even and an integer $r \geq 1$. There is a multigraph H where $\mathsf{MaxMult}(H) \leq r$ if and only if for $k = 1, \ldots, n$ we have*

$$\sum_{i=1}^{k} d_i \leq rk(k-1) + \sum_{i=k+1}^{n} \min\{rk, d_i\} \ .$$

We conclude the discussion by illustrating how the generalized variant of the Havel-Hakimi algorithm outlined in [29] can be used to compute a multigraph H where $\mathsf{MaxMult}(H) \leq r$. The algorithm receives as input a sequence d and a parameter r and looks for a realization of d which uses at most r copies of each edge. An explicit description of the algorithm is given below. Each recursive call of this algorithm connects an arbitrary vertex ℓ to at most d_ℓ vertices with the highest degrees (not including itself) while using at most r copies per edge. The initial call is $\mathrm{MaxMult}(d, r)$.

Algorithm 1: $\mathrm{MaxMult}(d, r)$

```
if d = 0 then return ∅
Let ℓ be an arbitrary vertex such that d_ℓ > 0
V_ℓ ← V \ {ℓ}; E_ℓ ← ∅
while d_ℓ > 0 do
    j ← argmax_{q∈V_ℓ} d_q
    if d_j ≤ 0 then ABORT                    ▷ d is not r-graphic
    Add (ℓ, j) to E_ℓ
    d_ℓ ← d_ℓ − 1; d_j ← d_j − 1
    if #_{E_ℓ}(j, ℓ) = r then V_ℓ ← V_ℓ \ {j}
return E_ℓ ∪ MaxMult(d, r)
```

For completeness, we include an analysis of the algorithm, starting with the running time. There are $O(n)$ recursive calls, and each can be implemented to run in $O(n)$ time, yielding a total of $O(n^2)$ time. A more careful implementation yields a running time of $O(\sum_i d_i)$.

For correctness, the next lemma proves that if the algorithm terminates successfully, then the computed edge multiset induces a realization.

Lemma 1. *Let d be a nonincreasing sequence and r be a positive integer. If the Algorithm MaxMult terminates without aborting, then the computed multiset induces a realization of d that contains at most r copies per edge.*

Proof. By induction on the number of recursive calls. In the base case, $d = 0$, and thus $E = \emptyset$ is a realization. For the induction step, let d' denote the value of d in Line 10, and let E' be the multiset of edges returned by the recursive call in Step 10. By the inductive hypothesis, E' induces a realization of d' that contains at most r copies per edge. The while loop realizes $d - d'$ and Line 9 makes sure that the number of copies per edge is bounded by r. Also, observe that E' does not contain edges adjacent to ℓ, since $d'_\ell = 0$ Hence $E' \cup E_\ell$ induces a realization of d that contains at most r copies per edge. □

The following lemma shows that if d admits a realization, then the algorithm will terminate successfully.

Lemma 2. *Let d be a nonincreasing sequence and r be a positive integer. If d has a realization that contains at most r copies per edge, then the algorithm will terminate successfully.*

Proof. By induction on the number of recursive calls. In the base case, $d = 0$, and in this case an $H = (V, \emptyset)$ is the only realization. For the induction step, let H be a realization of d that contains at most r copies per edge. First, observe that since there is a realization of d, the while loop would terminate successfully. If E_ℓ is contained in H, then we are done. Otherwise, we show that there is another realization H' of d that contains E_ℓ. Transform H into H' using edge-swaps. Let $v \in V$ such that $\#_H(\ell, v) < \#_{E_\ell}(\ell, v)$, where $\#_{E_\ell}(\ell, v)$ is the multiplicity of the edge (ℓ, v) in E_ℓ. It follows that there is a vertex u such that $\#_H(\ell, u) > \#_{E_\ell}(\ell, u)$. Since the last copy of (ℓ, v) was chosen over (ℓ, u), it must be that

$$d_v - \#_H(\ell, v) \geq d_v - (\#_{E_\ell}(\ell, v) - 1) \geq d_u - \#_{E_\ell}(\ell, u) > d_u - \#_H(\ell, u) \ .$$

Consequently, there must be a vertex $w \neq \ell, v, u$ such that $\#_H(v, w) > 0$ and $\#_H(u, w) < r$. Modify H by applying the following edge swap: remove the edges $(\ell, u), (v, w)$ and add the edges $(\ell, v), (u, w)$. It is not hard to verify that the resulting multigraph is a realization d that contains at most r copies per edge. Moreover, $\#_H(\ell, v)$ is increased by 1. Continue with edge swaps in this manner until H contains E_ℓ. □

The best r can be obtained with Algorithm 1 using binary search on r.

Lemma 3. *There exist a polynomial time algorithm that, given a sequence d, computes a realization H of d such that* $\mathsf{MaxMult}(H) = \mathsf{MaxMult}(d)$.

References

1. Adams, P., Nikolayevsky, Y.: Planar bipartite biregular degree sequences. Discr. Math. **342**, 433–440 (2019)
2. Aigner, M., Triesch, E.: Realizability and uniqueness in graphs. Discr. Math. **136**, 3–20 (1994)
3. Alpers, A., Gritzmann, P.: Reconstructing binary matrices under window constraints from their row and column sums. Fundamenta Informaticae **155**(4), 321–340 (2017)
4. Alpers, A., Gritzmann, P.: Dynamic discrete tomography. Inverse Probl. **34**(3), 034003 (2018)
5. Alpers, A., Gritzmann, P.: On double-resolution imaging and discrete tomography. SIAM J. Discr. Math. **32**, 1369–1399 (2018)
6. Althöfer, I.: On optimal realizations of finite metric spaces by graphs. Discret. Comput. Geom. **3**, 103–122 (1988)
7. Anstee, R.: Properties of a class of (0,1)-matrices covering a given matrix. Canad. J. Math. **34**, 438–453 (1982)
8. Anstee, R.P.: An algorithmic proof of Tutte's f-factor theorem. J. Algorithms **6**(1), 112–131 (1985)
9. Baldisserri, A.: Buneman's theorem for trees with exactly n vertices. CoRR (2014)
10. Bandelt, H.: Recognition of tree metrics. SIAM J. Discr. Math. **3**, 1–6 (1990)
11. Bar-Noy, A., Böhnlein, T., Lotker, Z., Peleg, D., Rawitz, D.: The generalized microscopic image reconstruction problem. In: 30th ISAAC, volume 149 of LIPIcs, pp. 42:1–42:15 (2019)
12. Bar-Noy, A., Choudhary, K., Peleg, D., Rawitz, D.: Efficiently realizing interval sequences. SIAM J. Discr. Math. **34**(4), 2318–2337 (2020)
13. Bar-Noy, A., Peleg, D., Perry, M., Rawitz, D.: Composed degree-distance realizations of graphs. In: 32nd IWOCA (2021)
14. Bar-Noy, A., Peleg, D., Perry, M., Rawitz, D., Schwartz, N. L.: Distance realization approximations. In: preparation (2021)
15. Battaglino, D., Frosini, A., Rinaldi, S.: A decomposition theorem for homogeneous sets with respect to diamond probes. Comput. Vis. Image Underst. **117**, 319–325 (2013)
16. Baum, D.A., Smith, S.D.: Tree Thinking: an Introduction to Phylogenetic Biology. Roberts and Company, Greenwood Village, CO (2013)
17. Blitzstein, J.K., Diaconis, P.: A sequential importance sampling algorithm for generating random graphs with prescribed degrees. Internet Math. **6**(4), 489–522 (2011)
18. Broom, M., Cannings, C.: A dynamic network population model with strategic link formation governed by individual preferences. J. Theoret. Biol. **335**, 160–168 (2013)
19. Broom, M., Cannings, C.: Graphic deviation. Discr. Math. **338**, 701–711 (2015)
20. Broom, M., Cannings, C.: Game theoretical modelling of a dynamically evolving network i: general target sequences. J. Dyn. Games **335**, 285–318 (2017)
21. Burstein, D., Rubin, J.: Sufficient conditions for graphicality of bidegree sequences. SIAM J. Discr. Math. **31**, 50–62 (2017)
22. Cai, M.-C., Deng, X., Zang, W.: Solution to a problem on degree sequences of graphs. Discr. Math. **219**(1–3), 253–257 (2000)
23. Camin, J.H., Sokal, R.R.: A method for deducing branching sequences in phylogeny. Evolution **19**, 311–326 (1965)

24. Chen, W.-K.: On the realization of a (p, s)-digraph with prescribed degrees. J. Franklin Inst. **281**(5), 406–422 (1966)
25. Chernyak, A.A., Chernyak, Z.A., Tyshkevich, R.I.: On forcibly hereditary p-graphical sequences. Discr. Math. **64**, 111–128 (1987)
26. W. Chou and H. Frank. Survivable communication networks and the terminal capacity matrix. IEEE Trans. Circ. Theory, CT-17, 192–197 (1970)
27. Choudum, S.A.: A simple proof of the Erdös-Gallai theorem on graph sequences. Bull. Austral. Math. Soc. **33**(1), 67–70 (1991)
28. 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**, 432–448 (2001)
29. Chungphaisan, V.: Conditions for sequences to be r-graphic. Discr. Math. **7**(1–2), 31–39 (1974)
30. Cloteaux, B.: Fast sequential creation of random realizations of degree sequences. Internet Math. **12**(3), 205–219 (2016)
31. Culberson, J.C., Rudnicki, P.: A fast algorithm for constructing trees from distance matrices. Inf. Process. Lett. **30**(4), 215–220 (1989)
32. Dahl, G., Flatberg, T.: A remark concerning graphical sequences. Discr. Math. **304**(1–3), 62–64 (2005)
33. Dahlhaus, E.: Fast parallel recognition of ultrametrics and tree metrics. SIAM J. Discr. Math. **6**(4), 523–532 (1993)
34. 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)
35. Erdös, D., Gemulla, R., Terzi, E.: Reconstructing graphs from neighborhood data. ACM Trans. Knowl. Discov. Data **8**(4), 23:1–23:22 (2014)
36. Erdös, P., Gallai, T.: Graphs with prescribed degrees of vertices [Hungarian]. Matematikai Lapok **11**, 264–274 (1960)
37. Erdös, P.L., Miklós, I., Toroczkai, Z.: A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs. Electr. J. Comb. **17**(1) (2010)
38. Even, G., Medina, M., Patt-Shamir, B.: On-line path computation and function placement in SDNs. Theory Comput. Syst. **63**(2), 306–325 (2019)
39. Even, G., Rost, M., Schmid, S.: An approximation algorithm for path computation and function placement in SDNs. In: Suomela, J. (ed.) SIROCCO 2016. LNCS, vol. 9988, pp. 374–390. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48314-6_24
40. 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
41. Felsenstein, J.: Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. **17**, 368–376 (1981)
42. Fitch, W.M.: Toward defining the course of evolution: Minimum change for a specific tree topology. Syst. Biol. **20**, 406–416 (1971)
43. Frank, A.: Augmenting graphs to meet edge-connectivity requirements. SIAM J. Discr. Math. **5**, 25–43 (1992)
44. Frank, A.: Connectivity augmentation problems in network design. In: Mathematical programming: state of the art, pp. 34–63. Univ. Michigan (1994)
45. Frank, H., Chou, W.: Connectivity considerations in the design of survivable networks. IEEE Trans. Circuit Theory, CT-17, 486–490 (1970)

46. Frosini, A., Nivat, M.: Binary matrices under the microscope: a tomographical problem. Theor. Comput. Sci. **370**(1–3), 201–217 (2007)
47. Frosini, A., Nivat, M., Rinaldi, S.: Scanning integer matrices by means of two rectangular windows. Theor. Comput. Sci. **406**(1–2), 90–96 (2008)
48. Fulkerson, D.: Zero-one matrices with zero trace. Pacific J. Math. **12**, 831–836 (1960)
49. Gale, D.: A theorem on flows in networks. Pacific J. Math. **7**, 1073–1082 (1957)
50. Garg, A., Goel, A., Tripathi, A.: Constructive extensions of two results on graphic sequences. Discr. Appl. Math. **159**(17), 2170–2174 (2011)
51. Gontier, N.: Depicting the tree of life: the philosophical and historical roots of evolutionary tree diagrams. Evol. Educ. Outreach **4**, 515–538 (2011)
52. Gritzmann, P., Langfeld, B., Wiegelmann, M.: Uniqueness in discrete tomography: three remarks and a corollary. SIAM J. Discr. Math. **25**, 1589–1599 (2011)
53. Guo, J., Yin, J.: A variant of Niessen's problem on degree sequences of graphs. Discr. Math. Theor. Comput. Sci. **16**, 287–292 (2014)
54. Gupta, G., Joshi, P., Tripathi, A.: Graphic sequences of trees and a problem of Frobenius. Czechoslovak Math. J. **57**, 49–52 (2007)
55. Hakimi, S.L.: On realizability of a set of integers as degrees of the vertices of a linear graph -I. SIAM J. Appl. Math. **10**(3), 496–506 (1962)
56. Hakimi, S.L., Yau, S.S.: Distance matrix of a graph and its realizability. Quart. Appl. Math. **22**, 305–317 (1965)
57. Hammer, P.L., Ibaraki, T., Simeone, B.: Threshold sequences. SIAM J. Algebra. Discr. **2**(1), 39–49 (1981)
58. Hammer, P.L., Simeone, B.: The splittance of a graph. Combinatorica **1**, 275–284 (1981)
59. Hartung, S., Nichterlein, A.: Np-hardness and fixed-parameter tractability of realizing degree sequences with directed acyclic graphs. SIAM J. Discr. Math. **29**, 1931–1960 (2015)
60. Havel, V.: A remark on the existence of finite graphs [in Czech]. Casopis Pest. Mat. **80**, 477–480 (1955)
61. Heinrich, K., Hell, P., Kirkpatrick, D.G., Liu, G.: A simple existence criterion for $(g < f)$-factors. Discr. Math. **85**, 313–317 (1990)
62. Hell, P., Kirkpatrick, D.: Linear-time certifying algorithms for near-graphical sequences. Discr. Math. **309**(18), 5703–5713 (2009)
63. Hennig, W.: Phylogenetic Systematics. Illinois University Press, Champaign (1966)
64. Herman, G.T., Kuba, A.: Discrete Tomography: Foundations, Algorithms, and Applications. Springer Science & Business Media (2012)
65. Hulett, H., Will, T.G., Woeginger, G.J.: Multigraph realizations of degree sequences: maximization is easy, minimization is hard. Oper. Res. Lett. **36**(5), 594–596 (2008)
66. Jayadev, S.P., Narasimhan, S., Bhatt, N.: Learning conserved networks from flows. Technical report, CoRR (2019). http://arxiv.org/abs/1905.08716
67. Kelly, P.: A congruence theorem for trees. Pacific J. Math. **7**, 961–968 (1957)
68. Kidd, K.K., Sgaramella-Zonta, L.: Phylogenetic analysis: Concepts and methods. American J. Hum. Gen. **23**, 235–252 (1971)
69. Kim, H., Toroczkai, Z., Erdos, P.L., Miklos, I., Szekely, L.A.: Degree-based graph construction. J. Phys. Math. Theor. **42**, 1–10 (2009)
70. Kleitman, D.J.: Minimal number of multiple edges in realization of an incidence sequence without loops. SIAM J. Appl. Math. **18**(1), 25–28 (1970)

71. Kleitman, D.J., Wang, D.L.: Algorithms for constructing graphs and digraphs with given valences and factors. Discr. Math. **6**, 79–88 (1973)
72. Kutiel, G., Rawitz, D.: Service chain placement in SDNs. Discr. Appl. Math. **270**, 168–180 (2019)
73. Li, C., McCormick, S., Simchi-Levi, D.: On the minimum-cardinality-bounded-diameter and the bounded-cardinality-minimum-diameter edge addition problems. Oper. Res. Lett. **11**, 303–308 (1992)
74. Lovász, L.: Subgraphs with prescribed valencies. J. Comb. Theory **8**, 391–416 (1970)
75. Mihail, M., Vishnoi, N.: On generating graphs with prescribed degree sequences for complex network modeling applications. In: 3rd ARACNE (2002)
76. Nieminen, J.: Realizing the distance matrix of a graph. J. Inf. Process. Cybern. **12**(1/2), 29–31 (1976)
77. Nivat, M.: Sous-ensembles homogénes de z2 et pavages du plan. Comptes Rendus Mathematique **335**(1), 83–86 (2002)
78. O'Neil, P.V.: Ulam's conjecture and graph reconstructions. Am. Math. Mon. **77**, 35–43 (1970)
79. Owens, A., Trent, H.: On determining minimal singularities for the realizations of an incidence sequence. SIAM J. Appl. Math. **15**(2), 406–418 (1967)
80. Owens, A.B.: On determining the minimum number of multiple edges for an incidence sequence. SIAM J. Appl. Math. **18**(1), 238–240 (1970)
81. Patrinos, A.N., Hakimi, S.L.: The distance matrix of a graph n and its tree realizability. Q. Appl. Math. **30**(3), 255 (1972)
82. Ryser, H.: Combinatorial properties of matrices of zeros and ones. Canad. J. Math. **9**, 371–377 (1957)
83. Schmeichel, E.F., Hakimi, S.L.: On planar graphical degree sequences. SIAM J. Appl. Math. **32**, 598–609 (1977)
84. Schuh, R.T., Brower, A.V.: Biological Systematics: Principles and Applications. Cornell University Press (2009)
85. Sierksma, G., Hoogeveen, H.: Seven criteria for integer sequences being graphic. J. Graph Theory **15**(2), 223–231 (1991)
86. Simões-Pereira, J.M.S.: An optimality criterion for graph embeddings of metrics. SIAM J. Discr. Math. **1**(2), 223–229 (1988)
87. Simões-Pereira, J.M.S.: An algorithm and its role in the study of optimal graph realizations of distance matrices. Discr. Math. **79**(3), 299–312 (1990)
88. Swofford, D.L., Olsen, G.J.: Phylogeny reconstruction. Mol. Syst. 411–501 (1990)
89. Tatsuya, A., Nagamochi, H.: Comparison and enumeration of chemical graphs. Comput. Struct. Biotechnol. **5**(6), e201302004 (2013)
90. Tripathi, A., Tyagi, H.: A simple criterion on degree sequences of graphs. Discr. Appl. Math. **156**(18), 3513–3517 (2008)
91. Tripathi, A., Venugopalan, S., West, D.B.: A short constructive proof of the Erdös-Gallai characterization of graphic lists. Discr. Math. **310**(4), 843–844 (2010)
92. Tripathi, A., Vijay, S.: A note on a theorem of Erdös & Gallai. Discr. Math. **265**(1–3), 417–420 (2003)
93. Tutte, W.T.: Graph factors. Combinatorica **1**, 79–97 (1981)
94. Tyshkevich, R.: Decomposition of graphical sequences and unigraphs. Discr. Math. **220**, 201–238 (2000)
95. Tyshkevich, R.I., Chernyak, A.A., Chernyak, Z.A.: Graphs and degree sequences: a survey I. Cybernetics **23**, 734–745 (1987)
96. Tyshkevich, R.I., Chernyak, A.A., Chernyak, Z.A.: Graphs and degree sequences: a survey II. Cybernetics **24**, 137–152 (1988)

97. Tyshkevich, R.I., Chernyak, A.A., Chernyak, Z.A.: Graphs and degree sequences: a survey III. Cybernetics **24**, 539–548 (1988)
98. Tyshkevich, R.I., Mel'nikov, O.I., Kotov, V.M.: Graphs and degree sequences: canonical decomposition. Cybernetics **17**, 722–727 (1981)
99. Ulam, S.: A Collection of Mathematical Problems. Wiley, Hoboken (1960)
100. Varone, S.C.: A constructive algorithm for realizing a distance matrix. Eur. J. Oper. Res. **174**(1), 102–111 (2006)
101. Wormald, N.: Models of random regular graphs. Surveys Combin. **267**, 239–298 (1999)
102. Zverovich, I.E., Zverovich, V.E.: Contributions to the theory of graphic sequences. Discr. Math. **105**(1–3), 293–303 (1992)