Selected Neighbor Degree Forest Realization

Amotz Bar-Noy
City University of New York (CUNY), NY, USA

David Peleg
Weizmann Institute of Science, Rehovot, Israel

Dror Rawitz
Bar Ilan University, Ramat-Gan, Israel

Elad Yehezkel
Weizmann Institute of Science, Rehovot, Israel

Abstract

The classical degree realization problem is defined as follows: Given a sequence $\bar{d} = (d_1, \ldots, d_n)$ of positive integers, construct an n-vertex graph in which each vertex u_i has degree d_i (or decide that no such graph exists). In this article, we present and study the related *selected neighbor degree realization* problem, which requires that each vertex u_i of G has a *neighbor* of degree d_i. We solve the problem when G is required to be acyclic (i.e., a forest), and present a sufficient and necessary condition for a given sequence to be realizable.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases network realization, graph algorithms, lower bound

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.27

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

1 Introduction

Background and motivation. Different properties of a given network can be described using “profiles” of the network. As a classical example, the *degree profile* (or *degree sequence*) of an n-vertex graph G is the sequence $\texttt{DEG}(G) = (d_1, \ldots, d_n)$, where $d_i = \deg(u_i)$ is the degree of the vertex u_i.

The extensively studied *degree realization* problem concerns the situation where given a sequence of positive integers $\bar{d} = (d_1, \ldots, d_n)$, we are asked whether there exists a graph whose degree sequence conforms to $\bar{d}$. (If so, then the sequence $\bar{d}$ is called *graphic*.) Erdös and Gallai [15] gave a necessary and sufficient condition for deciding if a given degree profile is realizable (also implying a $\Theta(n)$ time decision algorithm), and Havel and Hakimi [18, 19] gave a $\Theta(\sum_i d_i)$ time algorithm that given a degree profile $\bar{d}$ computes a realizing graph, or proves that the profile is not realizable. The problem is known to be particularly simple when the realizing graph is required to be acyclic, in which case the necessary and sufficient condition for realizability is simply that $\sum_i d_i = 2(n-k)$, where $k \in \{1, \ldots, n\}$ (see [16] for a short analysis for trees). Many extensions and variations of the degree realization problem were studied in the past, cf. [1, 11, 20, 24, 26, 31, 32, 34, 35]. Interesting applications in the context of social networks are studied in [9, 13, 21].

Other aspects of the graph structures may be described using other types of profiles. We focus on profiles that capture aspects of the *vertex neighborhoods* in the given graph. One reason for our interest in neighbor degrees is that in the context of social networks, it is often informative to observe not only the individual degree of each vertex, but also the degrees of nearby vertices, since obtaining a more complete picture of the degree distribution in a given neighborhood may reveal useful information regarding the interrelationships among vertices, and their relative standing in their immediate society.

© Amotz Bar-Noy, David Peleg, Dror Rawitz, and Elad Yehezkel;
licensed under Creative Commons License CC-BY 4.0

Let $N[u]$ denote the closed (inclusive) neighborhood of the vertex u in G, namely $N[u] = \{v \mid (v,u) \in E\} \cup \{u\}$. Clearly, the profile $N(G) = \langle N[u_1], \ldots, N[u_n] \rangle$ tells us everything we may need to know. However, this profile is costly, or "heavy", in the sense that storing it requires as much memory as storing the entire graph. Instead, one is often interested in studying "lighter" profiles, storing only a small amount of information per vertex.

For a graph $G(V,E)$, $V = \{u_1, \ldots, u_n\}$, let $\mathrm{ctr} : V \to V$ be a *neighbor selection* function, such that $\mathrm{ctr}(u) \in N[u]$ for each vertex u. We refer to the vertex $\mathrm{ctr}(u)$ as u's *selected neighbor* or *center*. For each vertex u, denote the degree of u's selected neighbor by $\mathsf{snd}(u) = \deg(\mathrm{ctr}(u))$. Then the sequence $\mathtt{SND}(G, \mathrm{ctr}) = (\mathsf{snd}(u_1), \ldots, \mathsf{snd}(u_n))$ is referred to as the *selected neighbor degree* (SND) profile of the pair (G, ctr).

Interesting special cases of the selected neighbor degree profile SND arise when the neighbor selection function ctr targets some specific neighbor for each vertex. The ordinary degree profile is obtained by using $\mathrm{ctr}^{self}(u) = u$. Picking the function ctr^{max}, which selects for every vertex u its neighbor of maximum degree, yields

$$\mathsf{maxnd}(u) = \deg(\mathrm{ctr}^{max}(u)) = \max\{\deg(w) \mid w \in N[u]\} ,$$

which gives the *maximum neighbor degree* profile $\mathsf{MaxND}(G) = (\mathsf{maxnd}(u_1), \ldots, \mathsf{maxnd}(u_n))$. The functions ctr^{min} and minnd and the *minimum neighbor degree* profile $\mathsf{MinND}(G)$ can be defined analogously. Note that the profile $\mathsf{MaxND}(G)$ (resp., $\mathsf{MinND}(G)$) is independent of the choice of ctr^{max} (resp., ctr^{min}). This is not the case for general SND profiles.

Recently, we studied the realization problem for the MinND and MaxND profiles. In particular, the *minimum* neighbor degree realization problem, where given a sequence $\bar{d}$ of n integers one must decide if there is a graph G such that $\mathsf{MinND}(G) = \bar{d}$ and construct such a graph (if exists), was studied in [3]. A complete characterization was given for realization by *forests* (i.e., acyclic graphs), but the problem over general graphs was left open. Surprisingly, when studying the realizability of the *maximum* neighbor degree profile MaxND [6], the picture was reversed: we were able to give a complete characterization for the realization problem of maximum neighbor degrees on general graphs, but on forests the problem appears to be harder, and was left open.

It is therefore natural to investigate the problem's behavior when, instead of MaxND and MinND, we look at general *selected neighbor* degree profiles. The current study addresses this question. We resolve the realizability of SND profiles by forests, although surprisingly even this more relaxed variant turned out to be subtle and considerably more difficult than anticipated initially. Formally, we study the *selected neighbor degree Forest realization* (SNDF) problem. Consider a given n-integer *SNDF-specification* $\bar{d}$. We say that $\bar{d}$ is a *forest-realizable SNDF-profile* if there exists an n-vertex forest F, and a neighbor selection function ctr, whose SNDF-profile satisfies $\mathtt{SND}(F, \mathrm{ctr}) = \bar{d}$.

Our Contribution. We study an optimization version of the SNDF problem. As mentioned earlier, not every SNDF-specification $\bar{d}$ is realizable. To cope with unrealizable profiles, we define a measure for the deviation of a given profile $\bar{d}$ from realizability in Sect. 2, where we also introduce the basic elements of the problem, as well as some preliminary notions used in our solution. In Sect. 3 we introduce the framework, give a high-level overview of the general approach and present basic tools for handling SNDF profiles. In Section 4 we present a tight lower bound on the deviation of SNDF-profiles. This lower bound lays the foundation for our algorithm. We also outline our construction algorithm for the problem, which is optimal in the sense that when $\bar{d}$ is realizable by a forest, the resulting construction (F, ctr)

is a realization of $\bar{d}$, and when it is not realizable, the resulting (F, ctr) has the minimum possible deviation (matching the lower bound). Both the lower bound and the algorithm are rather involved, hence most of the details are omitted for lack of space and can be found in [36]. As a byproduct, our analysis also yields necessary and sufficient conditions for a specification $\bar{d}$ to be realizable by a forest, as well as a fast algorithm for deciding realizability, thus providing a complete solution for the SNDF problem. Finally, our algorithm can be easily modified into one that minimizes the number of centers.

Related Work. Over the years, various extensions of the degree realization problem were studied, cf. [1, 34]. Many studies have addressed related questions such as finding all the (non-isomorphic) graphs that realize a given degree sequence, counting all the (non-isomorphic) realizing graphs of a given degree sequence, sampling a random realization for a given degree sequence as uniformly as possible, or determining the conditions under which a given degree sequence defines a unique realizing graph (a.k.a. the *graph reconstruction* problem), cf. [11, 15, 18, 19, 20, 24, 26, 31, 32, 35]. Interesting applications in the context of social networks are studied in [9, 21, 13]. The somewhat related *shotgun assembly* problem [22] studies graph specifications consisting of a description of the r-neighborhood (up to radius r) of each vertex i. Realization questions of a similar nature were studied for *other* applications, where given *some* type of information profile specifying the desired vertex properties (concerning distances, connectivity, centrality, or any other property of significance), one may ask whether there exists a graph conforming to the specified profile (see, e.g., [2, 4, 5, 7, 8, 12, 14, 10, 17, 23, 25, 27, 28, 29, 30, 33, 37]). The selected neighbor degree realization problem belongs to this class of problems.

2 Preliminaries

Let $F = (V, E)$ be a forest. For a vertex set $U \subseteq V$, let $N[U] = \bigcup_{u \in U} N[u]$ be the *closed neighborhood* of U. Let $\mathrm{ctr} : V \to V$ be a *neighbor* function on F's vertices such that $\mathrm{ctr}(u) \in N[u]$ for each $u \in V$. For every $u \in V$, define

$$\mathsf{snd}(u) = \mathsf{snd}_{(F,\mathrm{ctr})}(u) = \deg_F(\mathrm{ctr}(u)) .$$

When F and ctr are clear from the context, we omit them and write $\mathsf{snd}(u)$. We refer to $\mathsf{snd}(u)$ as the snd *value of* u. The $\mathtt{SND}$ *profile* of (F, ctr) is the sequence

$$\mathtt{SND}(F, \mathrm{ctr}) = (\mathsf{snd}(u))_{u \in V} .$$

It is convenient to represent an SNDF profile in a condensed form as a list of non-negative integers $(k_i^{n_i})_{i=1}^{\ell}$, meaning that each value k_i appears in the list n_i times and the list contains ℓ distinct values, i.e. there are n_i vertices that have snd value k_i. We assume $n - 1 \geq k_1 > \ldots > k_\ell \geq 0$. Overall $n = \sum_{i=1}^{\ell} n_i$.

We are interested in the following SNDF realization problem. A given sequence $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ is viewed as an *SNDF profile*. It is realizable if there exists a pair, (F, ctr), where F is a forest, such that $\mathtt{SND}(F, \mathrm{ctr}) = \bar{d}$. We call (F, ctr) an *SNDF realization of* $\bar{d}$. (Note that $\bar{d}$ may or may not be realizable.) The problem concerns finding a realizing (F, ctr) for a given profile $\bar{d}$, if exists. Observe that the snd value 0 can only be realized by singleton vertices, independently of the rest of the profile. So hereafter assume that $k_\ell \geq 1$.

Star formations. When $k_i + 1$ vertices are required by the profile to have a neighbor of degree k_i, the requirement can be easily satisfied in a self-sufficient manner by a star composed of a root v and k_i leaves, all pointing at the root (i.e., with $\mathrm{ctr}(u) = v$). Likewise, if the profile contains $k_i^{n_i}$ where $n_i = c \cdot (k_i + 1)$ for some integer c, then this component of the profile can be realized on its own, using c stars of size k_i (where by the *size* of a star we refer to the number of leaves). This motivates the following alternative representation for a profile $\bar{d}$, using two sequences $(c_i)_{i=1}^{\ell}$ and $(\rho_i)_{i=1}^{\ell}$ such that

$$n_i = c_i(k_i + 1) - \rho_i \qquad \text{and} \qquad 0 \le \rho_i \le k_i, \tag{1}$$

where $\rho_i = c_i(k_i + 1) - n_i$ is the ith *residue*. We refer to the tuple $(c_i, \rho_i)_{i=1}^{\ell}$ as the *star formation* of $\bar{d}$. Note that

$$n = \sum_{i=1}^{\ell} n_i \;=\; \sum_{i=1}^{\ell} (c_i(k_i + 1) - \rho_i)\ .$$

As mentioned before, not every profile $\bar{d}$ is realizable. We therefore seek approximate solutions to the SNDF realization problem.

Upper realizations and deviation. Let $F = (V, E)$ be a forest and let $\mathrm{ctr} : V \to V$. We say that (F, ctr) is an *upper realization* (or U-realization) of the profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ if the SNDF profile of (F, ctr) is of the form $\mathtt{SND}(F, \mathrm{ctr}) = \bar{d}' = (k_i^{n'_i})_{i=1}^{\ell}$ (so in particular $\mathsf{snd}(u) \in \{k_1, \cdots, k_\ell\}$ for every u) and $n'_i \ge n_i$ for every i. Denote $n(F) = \sum_{i=1}^{\ell} n'_i$. Denote by $cc(F)$ the number of connected components in F. Define the *deviation* of (F, ctr) from the profile $\bar{d}$ as

$$Dev(\bar{d}, (F, \mathrm{ctr})) \;=\; \sum_{i=1}^{\ell} (n'_i - n_i) \;=\; n(F) - n\ .$$

The trivial U-realization. Observe that there is a straightforward way for constructing a U-realization to a given profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$. Define $(c_i)_{i=1}^{\ell}$ and $(\rho_i)_{i=1}^{\ell}$ as above. For each $1 \le i \le \ell$ create c_i stars of size k_i, and for each leaf v in a star with center u define $\mathrm{ctr}(v) = u$ and $\mathrm{ctr}(u) = u$. The resulting forest $\tilde{F}$ contains $cc(\tilde{F}) = \sum_{i=1}^{\ell} c_i$ connected components and $\sum_{i=1}^{\ell} c_i(k_i + 1)$ vertices. We refer to this construction as the *trivial* construction and denote it by $(\tilde{F}(\bar{d}), \tilde{\mathrm{ctr}}(\bar{d}))$. Note that

$$Dev(\bar{d}, (\tilde{F}, \tilde{\mathrm{ctr}})) = \sum_{i=1}^{\ell} (c_i(k_i + 1) - n_i) = \sum_{i=1}^{\ell} \rho_i\ . \tag{2}$$

While this realization may in some cases be near-optimal, our goal is to construct a U-realization (F, ctr) whose number of vertices, $n(F)$, is as close as possible to n, the specified number of vertices. Define the *realizable size* of a given profile $\bar{d}$ as the minimal size of any U-realization for it,

$$n^*(\bar{d}) = \min\{n(F) \mid \exists \mathrm{ctr} : (F, \mathrm{ctr}) \text{ is a U-realization for } \bar{d}\}.$$

Define the deviation of a given profile $\bar{d}$ as the minimum deviation over all of its U-realizations,

$$Dev(\bar{d}) \;=\; n^*(\bar{d}) - n \;=\; \min\{Dev(\bar{d}, (F, \mathrm{ctr})) \mid (F, \mathrm{ctr}) \text{ is a U-realization for } \bar{d}\}. \tag{3}$$

In this way, we can redefine realizability of SNDF profiles as follows. A given SNDF profile $\bar{d}$ is realizable if and only if $n^*(\bar{d}) = n$, or alternatively, $Dev(\bar{d}) = 0$. Denote by (F^*, ctr^*) an optimal U-realization of $\bar{d}$ (note that (F^*, ctr^*) is not necessarily unique), namely, such that $n(F^*) = n^*(\bar{d})$, or, $Dev(\bar{d}, (F^*, \mathrm{ctr}^*)) = Dev(\bar{d})$. Our goal is to find such a realization.

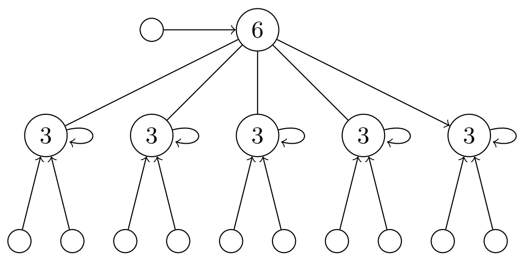


Figure 1 Realization of $\bar{d}^1 = (6^1, 3^{16})$ using $\bar{s} = (1, 5)$. The centers are marked by their degrees.

Centers and members. The sets of k_i-*centers* and k_i-*members* in upper realization (F, ctr) for a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ are

$$C_i(F, \text{ctr}) = \{u \in F \mid \deg(u) = k_i,\ \exists v \in N[u] \text{ s.t. } \text{ctr}(v) = u\} ,$$
$$M_i(F, \text{ctr}) = \{u \in F \mid \mathsf{snd}(u) = k_i\} .$$

(We may write simply C_i and M_i when clear from context.) Clearly, $C_i \cap C_j = \emptyset$ and $M_i \cap M_j = \emptyset$ for every $i \neq j$, and $\bigcup_i M_i = V$, i.e., the sets M_i form a partition of the vertices of F. Set $\sigma_i(F, \text{ctr}) = |C_i(F, \text{ctr})|$, and let

$$C(F, \text{ctr}) = \bigcup_{i=1}^{\ell} C_i(F, \text{ctr})$$

and

$$\overline{\sigma}(F, \text{ctr}) = (\sigma_1(F, \text{ctr}), \dots, \sigma_\ell(F, \text{ctr})) .$$

Center specification sequences (CSS). Clearly, any U-realization (F, ctr) for $\bar{d}$ must use at least c_i centers on layer i, so $\sigma_i \geq c_i$ for every i (see also Lemma 5(1)). Unfortunately, for some profiles, using exactly c_i centers for every i yields a suboptimal upper realization (with deviation greater than $Dev(\bar{d})$). A key component of the problem is thus to decide, given $\bar{d}$, on the right number of centers for each layer. These numbers are represented as a *center specification sequence* (or *CSS*) $\overline{s} = (s_1, \dots, s_\ell)$, which must satisfy $s_i \geq c_i$ for every i. A U-realization (F, ctr) *conforms* to $(\bar{d}, \overline{s})$ if the number of centers on each layer is as specified by $\overline{s}$, i.e., $\overline{\sigma}(F, \text{ctr}) = \overline{s}$. In our construction, we first select a CSS $\overline{s}$, and then look for a conforming U-realization (F, ctr) for it.

For example, the profile $\bar{d}^1 = (6^1, 3^{16})$ requires at least $\overline{c} = (1, 4)$ centers, but any U-realization with this many centers has deviation at least 1 (as follows from Thm. 8), whereas using the CSS $\overline{s} = (1, 5)$ yields the optimal $Dev(\bar{d}^1) = 0$, as demonstrated in the Figure 1.

Intuitively, in a conforming U-realization (F, ctr), each $x \in C_i$ acts as the center of a star of degree k_i, potentially allowing its "clients" $y \in M_i$ to have $\mathsf{snd}(y) = k_i$ by setting $\text{ctr}(y) = x$. Each center is also a member, possibly for a different i. Also, $\text{ctr}(u) \in C_i \cap N[u]$ for every $i \in [\ell]$ and $u \in M_i$, so $M_i \subseteq N[C_i]$ and $|M_i| \leq s_i(k_i + 1)$. Define the *residue* of k_i w.r.t. a CSS $\overline{s}$ as

$$\rho_i^s = s_i(k_i + 1) - n_i. \tag{4}$$

Note that Eq. (4) for $\overline{s} = \sigma(F, \text{ctr}) = (s_1, \dots, s_\ell)$ and $\overline{\rho^s} = \overline{\rho^\sigma}(F, \text{ctr}) = (\rho_1^s, \dots, \rho_\ell^s)$ is analogous to Eq. (1) for our star formation $(\overline{c}, \overline{\rho})$, since

$$n_i = s_i(k_i + 1) - \rho_i^s = c_i(k_i + 1) - \rho_i, \tag{5}$$

which is related to our definition of star formation, since each k_i-center u has $|N[u]| = k_i + 1$. We break the forest F into (possibly overlapping) stars centered around the k_i-centers in C_i.

The following two assumptions on U-realizations (F, ctr) are used hereafter without loss of generality.

Member independence. In (F, ctr), the non-centers are independent, namely, F contains no edge (u, v) between any $u, v \notin C$. (Such edges can always be removed without changing the profile of F, and our constructions never use them.)

No cross-pointing. In (F, ctr), there is no *cross-pointing*, i.e., there are no two centers u and v such that $\mathrm{ctr}(u) = v$ and $\mathrm{ctr}(v) = u$. (If cross-pointing occurs, one can change the two ctr values to $\mathrm{ctr}(u) = u$ and $\mathrm{ctr}(v) = v$ without changing the profile. Again, our constructions never use cross-pointing.)

3 Framework and basic tools

In this section, we describe the basic framework and the tools we use for realizing SNDF profiles.

Handling leaf centers

We first show how to handle the cases where $k_\ell = 1$. Consequently, in the rest of this article, we consider only SNDF profiles $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ where $k_\ell \geq 2$.

Trivial profiles of the form $\bar{d} = (1^n)$ for $n \geq 2$ can be realized by a star (F, ctr) with $n-1$ leaves, such that $\mathrm{ctr}(u) = u$ for each leaf u, and for the center, v, $\mathrm{ctr}(v)$ is defined to be one of the leaves.

It remains to handle profiles $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ where $\ell \geq 2$ and $k_\ell = 1$. This is done by the following approach. Given such a profile $\bar{d}$, denote the *truncated profile* (without k_ℓ) by $\bar{d}' = (k_i^{n_i})_{i=1}^{\ell-1}$. We show that the deviation of $\bar{d}$ is $Dev(\bar{d}) = \max\left\{Dev(\bar{d}') - n_\ell, 0\right\}$, and moreover, there is a polynomial time algorithm that given an optimal U-realization (of a special type, referred to as *a leaf-covered* U-realization) for $\bar{d}'$, transforms it into an optimal U-realization for $\bar{d}$. Hence the problem is reduced to finding optimal leaf-covered realizations for truncated profiles (with $k_\ell \geq 2$).

Leaf-covered U-realizations

Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be an SNDF profile with U-realization (F, ctr). Denote $e_i = |M_i| - n_i$, namely, the number of excess vertices with snd value k_i. Let $L = \{u \in V(F) \mid \deg(u) = 1\}$ be the set of leaves of F. We say that (F, ctr) is *a leaf-covered* U-realization for $\bar{d}$ if $|L \cap M_i| \geq e_i$ for every i.

Intuitively, a leaf-covered U-realization is easy to work with, since we may think of all the excess vertices as being among the leaves.

The reduction and its analysis are deferred to the full paper (see [36]). We get:

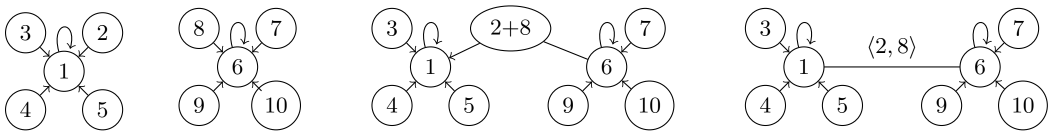
▶ **Proposition 1.** *Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be a profile with $\ell \geq 2$ and $k_\ell = 1$. Let (F', ctr') be an optimal leaf-covered U-realization for $\bar{d}' = (k_i^{n_i})_{i=1}^{\ell-1}$, namely, $Dev(\bar{d}', (F', ctr')) = Dev(\bar{d}')$. Then (F', ctr') can be converted, in polynomial time, into an optimal U-realization (F, ctr) for $\bar{d}$, with $Dev(\bar{d}, (F, ctr)) = Dev(\bar{d})$.*

By Prop. 1, it suffices to focus on finding optimal leaf-covered U-realizations for profiles without degree 1. Our main result is the following.

▶ **Theorem 2.** *Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be an SNDF profile such that $k_\ell \geq 2$. There exists an algorithm that constructs an optimal U-realization (F^*, ctr^*) for $\bar{d}$, namely, $Dev(\bar{d}, (F^*, ctr^*)) = Dev(\bar{d})$. In addition, (F^*, ctr^*) is leaf-covered. The run-time of the algorithm is $O(n^*(\bar{d}))$, which is optimal.*

Overview of the general approach

Consider a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ and let (F, ctr) be some U-realization of $\bar{d}$ with center classes $C_1, \ldots, C_\ell$ and $C = \bigcup_i C_i$. Consider a collection of stars $\{S_u\}_{u \in C}$, where S_u is centered at u and contains all of u's neighbors in F and their respective edges. Note that S_u and S_v are not necessarily disjoint, but they can share at most two vertices, namely, $|N[u] \cap N[v]| \leq 2$, since otherwise, there is a cycle in F. The trivial U-realization for $\bar{d}$, $(\tilde{F}, \tilde{\text{ctr}})$ is wasteful, since it employs $\sum_{i=1}^{\ell} c_i$ *pairwise disjoint* stars, resulting in $\sum_{i=1}^{\ell} c_i(k_i + 1)$ vertices. To improve it, we construct a realization by starting from disjoint stars and then forcing them to share vertices by performing certain merge operations. The key operation performed by our algorithm involves merging stars. Specifically, the algorithm employs two operations, referred to as *head-merges* and *leaf-merges*.



(a) (F_2, ctr_2). **(b)** (F_3, ctr_3). **(c)** (F_4, ctr_4).

Figure 2 The forests F_2, F_3 and F_4.

To illustrate these operations, consider the example profile $\bar{d}^2 = (4^{10})$. As $10 = 2 \cdot (4+1)$, an exact realization of this profile is obtained by the trivial U-realization composed of the following pair (F_2, ctr_2), consisting of two stars of size 4 (see Figure 2a). The directional edges represent the value of the ctr pointer of a specific vertex.

A *leaf-merge* of two stars fuses together two leaves, one of each star, into a single vertex, thus creating a single tree. Consider, for example, the profile $\bar{d}^3 = (4^9)$. As $9 = 2 \cdot (4+1) - 1$, the trivial U-realization composed of two stars of size 4, has one excess vertex (namely, a deviation of 1). To overcome this problem, we can leaf-merge vertices 2 and 8 of the forest F_2 into a single vertex denoted $2 + 8$, yielding the following pair (F_3, ctr_3) (see Figure 2b).

A *head-merge* of two stars is obtained by discarding one leaf of each star and connecting the star roots by a new edge. For example, to realize the profile $\bar{d}^4 = (4^8)$, whose trivial U-realization has a deviation of *two* excess vertices, we can head-merge the two stars of forest F_2 by completely removing vertices 2 and 8 and connecting the star heads by a new edge (marked by $\langle 2, 8 \rangle$ in the figure), yielding the following pair (F_4, ctr_4) (see Figure 2c).

Generally, satisfying part $k_i^{n_i}$ of the profile requires using stars with k_i leaves, but this will satisfy the profile only if n_i is a multiple of $k_i + 1$. For other n_i values, using $c_i = \lceil \frac{n_i}{k_i+1} \rceil$ stars yields more vertices than needed. We therefore use head and leaf merges to get rid of the excess vertices.

For example, consider the profile $\bar{d}^5 = (4^{12}, 3^7)$. As $12 = 3(4+1) - 3$ and $7 = 2(3+1) - 1$, we start by creating three size 4 stars and two size 3 stars, yielding the pair (F_5, ctr_5) shown in Figure 3.

The forest F_5 has three excess vertices with snd value 4 and one excess vertex with snd value 3. To correct it, we apply a head-merge operation and a leaf-merge operation on the size 4 stars, and a leaf-merge operation on the size 3 stars. This creates the following desired forest (F'_5, ctr'_5) depicted in Figure 4, which satisfies $\bar{d}$.

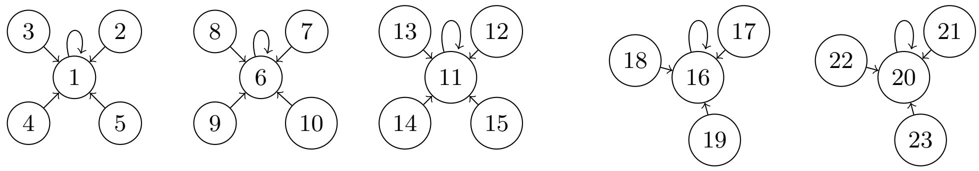


Figure 3 (F_5, ctr_5).

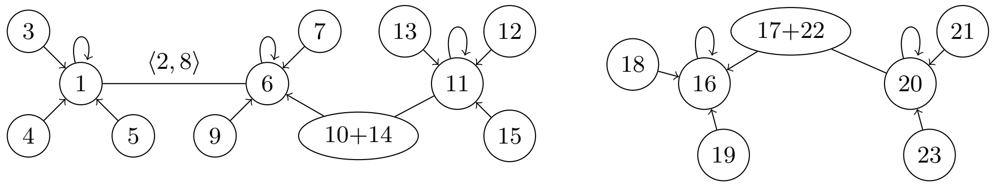


Figure 4 (F_5', ctr_5').

Our algorithm also exploits the fact that the value of ctr can be set so as to change the snd value of a vertex, by applying head or leaf merge operation to two stars of *different* sizes.

To illustrate this point, let us consider the following example. The profile $\bar{d}^6 = (4^4, 3^4)$ has one excess vertex with snd value 4. This profile is realizable in this way by starting from stars of size 3 and 4, and applying a leaf merge, to get the forest (F_6, ctr_6) shown in Figure 5.

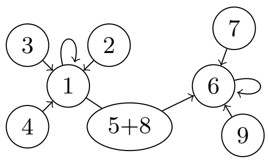


Figure 5 (F_6, ctr_6).

The analysis of our algorithm, and its proof of optimality, are based on the crucial observation that the *only* way to merge two individual stars is by head or leaf merges, namely, discarding one or two vertices, since trying to modify two stars by fusing together three or more vertices will create a cycle. This implies that certain profiles cannot be satisfied. For example, consider the profile $\bar{d}^7 = (4^7)$. Again, we must start with a forest F_2. However, there is no way to discard three vertices from F_2 and end up with a satisfying forest of 7 vertices. Indeed, based on Lemma 5 one can show that $Dev(\bar{d}^7) \geq 1$.

The $\mathcal{LM}$ operation and Procedure *Connect*

We now present Procedure $\mathcal{LM}$, which reduces the deviation (and the number of connected components in the forest) by 1, by performing a single leaf merge. Formally, given a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ and a U-realization (F, ctr) of $\bar{d}$ with $Dev(\bar{d}, (F, \text{ctr})) > 0$, such that F has $q = cc(F) \geq 2$ connected components, the operation constructs a U-realization (F', ctr') for $\bar{d}$ with $Dev(\bar{d}, (F', \text{ctr}')) = Dev(\bar{d}, (F, \text{ctr})) - 1$, such that F' has $cc(F') = q - 1$ connected components.

While the leaf merge operation is straightforward, applying it to an arbitrary forest raises some subtle points, as discussed next. Let $i \in [\ell]$ such that $|M_i(F, \text{ctr})| > n_i$ namely, the snd value k_i appears in (F, ctr) more than n_i times. Let $u \in V(F)$ be a k_i-center in F. The procedure aims to remove one of its k_i-members from F (possibly, u itself) in order to reduce the number of appearances of the `snd` value k_i in F (hence reducing the deviation of (F, ctr) from $\bar{d}$). However, this cannot always be done directly, since removing vertices from F results in an undesired change in the degree of their neighbors, which might affect the `snd` values of other vertices in F. Hence, removing vertices from F needs to be done carefully. The method details, as well as formal code and analysis, are deferred to the full paper (see [36]).

▶ **Lemma 3.** *Given $\bar{d}$ and (F, ctr) such that $cc(F) \geq 2$ and $Dev(\bar{d}, (F, ctr)) \geq 1$, $\mathcal{LM}$ returns a U-realization (F', ctr') for $\bar{d}$ such that $cc(F') = cc(F) - 1$ and $Dev(\bar{d}, (F', ctr')) = Dev(\bar{d}, (F, ctr)) - 1$.*

We now present a simple construction algorithm named $\mathcal{C}onnect$, based on performing only leaf merges. Given a profile $\bar{d}$ and a U-realization (F, ctr), The algorithm produces a new U-realization (F', ctr') by repeatedly invoking $\mathcal{LM}$ and applying leaf merges as long as possible. It halts when either the forest becomes connected, or the deviation becomes zero, yielding a proper realization.

Note that if the constructed (F', ctr') is a U-realization of $\bar{d}$, then $Dev(\bar{d}, (F', \text{ctr}')) = 0$. Otherwise, $\mathcal{LM}$ was invoked exactly $cc(F) - 1$ times, each decreasing the deviation by 1. We thus have the following.

▶ **Lemma 4.** *Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be a profile with a U-realization (F, ctr). Also, let (F', ctr') be the construction returned by Algorithm $\mathcal{C}onnect$. Then*

$$Dev(\bar{d}, (F', ctr')) = \max\{Dev(\bar{d}, (F, ctr)) - cc(F) + 1, 0\}.$$

A basic lower bound on $Dev(\bar{d})$. We next establish preliminary lower bounds on $Dev(\bar{d})$ and on the number of connected components in specific types of U-realizations.

For a forest F and a vertex subset $U \subseteq V(F)$, denote by $F[U]$ the induced forest on U's vertices, i.e., $V(F[U]) = U$ and $E(F[U]) = E(F) \cap (U \times U)$. For an SNDF profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ with a U-realization (F, ctr), our first lower bound expression involves $\overline{\sigma}$ and ρ^{σ} as defined in Sect. 2, and $cc(F[C])$, the number of connected components in the *centers forest* $F[C]$ induced by the centers:

$$LB_1^{dev} = LB_1^{dev}(\bar{d}, F, \text{ctr}) = \max\left\{\sum_{i=1}^{\ell} \rho_i^{\sigma} - 2\sum_{i=1}^{\ell} \sigma_i + cc(F[C]) + 1, 0\right\}. \quad (6)$$

The intuition for the lower bound is as follows. To minimize the deviation, we use head and leaf merges, each of which reduces $cc(F)$, the number of connected components, by 1. Hence the total number of both head and leaf merges that can be performed on some U-realization (F, ctr) is bounded by $cc(F) - 1$. However, a head-merge reduces the deviation by 2, whereas a leaf-merge reduces the deviation by 1. Since both of these merges "cost" the same (in the sense that they both decrease $cc(F)$ by 1), it is desirable to use as many head-merges as possible.

Now consider a given U-realization $(\tilde{F}, \tilde{\text{ctr}})$ consisting of s_i stars with k_i leaves for each $i \in [\ell]$, for some predetermined CSS $\overline{s}$. This U-realization has deviation $Dev(\bar{d}, (\tilde{F}, \tilde{\text{ctr}})) = \sum_{i=1}^{\ell} \rho_i^s$ and the number of connected components in $\tilde{F}$ is $cc(\tilde{F}) = \sum_{i=1}^{\ell} s_i$. In the best case, one can perform only head-merges to construct the final (F, ctr). Each head-merge reduces the deviation by 2 and $cc(F)$ by 1, therefore

$$Dev(\bar{d}, (F, \text{ctr})) = \sum_{i=1}^{\ell} \rho_i^s - 2\sum_{i=1}^{\ell} s_i + 2 \ .$$

However, in many profiles it is not possible to use only head-merges. So assume we used only x head-merges, and removed $2x$ excess vertices. Since the total number of merges is bounded by $cc(F) - 1 = \sum_{i=1}^{\ell} s_i - 1$, it is now possible to perform up to $\sum_{i=1}^{\ell} s_i - x - 1$ leaf-merges, removing $\sum_{i=1}^{\ell} s_i - x - 1$ additional excess vertices, totalling to $\sum_{i=1}^{\ell} s_i + x - 1$ excess vertices being removed. Note that each head-merge adds an edge (u, v) for some $u, v \in C$, so we have $x = |E(F[C])|$, where $E(F[C])$ is the set of edges in the centers forest $F[C]$. Since $F[C]$ is a forest,

$$|E(F[C])| = |V(F[C])| - cc(F[C]) = \sum_{i=1}^{\ell} s_i - cc(F[C]) \ ,$$

and therefore $x = \sum_{i=1}^{\ell} s_i - cc(F[C])$. In summary, at most

$$\sum_{i=1}^{\ell} s_i + x - 1 = 2 \cdot \sum_{i=1}^{\ell} s_i - cc(F[C]) - 1$$

out of the $\sum_{i=1}^{\ell} \rho_i^s$ initial excess vertices were removed.

This lower bound idea is formalized in the following lemma. (Proofs are deferred to the full paper, see [36].)

▶ **Lemma 5.** *For every profile $\bar{d}$ and U-realization (F, ctr):*
1. *$\sigma_i(F, ctr) \geq c_i$, for every i,*
2. *$Dev(\bar{d}, (F, ctr)) \geq LB_1^{dev}$.*

Reducing the error via head merges

We now present the main idea used later to construct an optimal U-realization (F, ctr) for a profile $\bar{d}$, namely, s.t. $Dev(\bar{d}, (F, \text{ctr})) = Dev(\bar{d})$. Our construction has two stages. In the first, we select a CSS $\bar{s} = (s_1, \dots, s_\ell)$ specifying the number of k_i-centers for every i. The selection ensures $\bar{s}$ is a CSS and has minimum deviation. The second stage builds a realization (F, ctr) that conforms to $(\bar{d}, \bar{s})$. A key observation, formalized by combining Lemmas 5(2) and 6, is that while $(\bar{d}, \bar{s})$ has many different conforming realizations (F, ctr), with different deviations, their deviations directly depend on $cc(F[C])$, the number of connected components in the centers forest $F[C]$ (where $C = C(F, \text{ctr})$ is the set of centers). Recall that a head merge is performed by taking two centers $u, v \in C$, removing one neighbor with degree 1 from each, and connecting u, v by an edge, thus it decreases the deviation by 2 while decreasing $cc(F[C])$ by 1. This means that in order to construct an optimal U-realization (with minimal $cc(F[C])$) that conforms to $(\bar{d}, \bar{s})$ for a CSS $\bar{s}$, we need to perform the maximal number of head merges. Suppose the resulting U-realization (F, ctr) allows no more head-merges. By Lemma 5(2), $Dev(\bar{d}, (F, \text{ctr})) \geq LB_1^{dev}(\bar{d}, F, \text{ctr})$. This bound can be matched by transforming (F, ctr) using leaf-merges.

▶ **Lemma 6.** *Consider a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ with a U-realization (F, ctr). There exists a U-realization (F', ctr') with deviation $Dev(\bar{d}, (F', ctr')) = LB_1^{dev}(\bar{d}, F, ctr)$.*

We conclude the discussion by stating that, given a profile $\bar{d}$, constructing an optimal U-realization for $\bar{d}$ boils down to
1. Choosing the "right" CSS $\bar{s}$.
2. Constructing a U-realization (F, ctr) that conforms to $(\bar{d}, \bar{s})$, such that $cc(F[C])$ is minimal among all other U-realizations with this property.

To formalize this workplan, we make the following definitions. Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be a profile with star formation $(c_i, \rho_i)_{i=1}^{\ell}$, and let $\overline{s}$ be a CSS for $\bar{d}$. Define the minimum number of connected components in a centers forest $F[C]$ for F that admits a conforming realization for $\bar{d}$ as

$$cc^*(\bar{d}, \overline{s}) = \min\left\{cc(F[C]) \mid \exists \text{ctr} : (F, \text{ctr}) \text{ conforms to } (\bar{d}, \overline{s})\right\}, \tag{7}$$

and define the expression obtained from Eq. (6) by replacing $cc(F[C])$ with $cc^*(\bar{d}, \overline{s})$ as

$$LB_3^{dev}(\bar{d}, \overline{s}) = \max\left\{\sum_{i=1}^{\ell} \rho_i^s - 2\sum_{i=1}^{\ell} s_i + cc^*(\bar{d}, \overline{s}) + 1, 0\right\} \tag{8}$$

A CSS $\overline{s^*}$ is *optimal* for $\bar{d}$ if $\overline{s^*}$ minimizes $LB_3^{dev}(\bar{d}, \overline{s^*})$ over all CSS's for $\bar{d}$. With the above definition, we state the following lemma.

▶ **Lemma 7.** *$Dev(\bar{d}) = LB_3^{dev}(\bar{d}, \overline{s^*}) = \min\{LB_3^{dev}(\bar{d}, \overline{s}) \mid \overline{s}$ is a CSS for $\bar{d}\}$.*

Hereafter, we focus on constructing, for a given profile $\bar{d}$ and any $\overline{s}$, a leaf-covered U-realization (F, ctr) conforming to $(\bar{d}, \overline{s})$. This U-realization has a minimal number of connected components in the centers forest, namely, $cc(F[C]) = cc^*(\bar{d}, \overline{s})$. In addition, in Section 4 we show how to find an optimal CSS $\overline{s^*}$ for $\bar{d}$. Combining these results, Theorem 2 follows.

Layer classification

To construct an optimal solution and analyze the structure of a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$, we classify each of its ℓ layers according to the values s_i and the residues ρ_i^s; later, the algorithm and analysis treat each class differently.

Consider a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ with a U-realization (F, ctr). Recall that the centers forest $F[C]$ is the induced forest of F on C. Note that for each $u \in V(F)$, $\text{ctr}(u) \in C$. For $\bar{d}$, (F, ctr), a residue sequence $(\rho_i^s)_{i=1}^{\ell}$ and $I \subseteq [\ell]$, define the partial residual deviation of I as $D_I = \sum_{i \in I} \rho_i^s$.

For a profile $\bar{d}$ and a CSS $\overline{s}$, define the following four sets of layers, referred to as *very good*, *good*, *bad*, and *very bad* layers.

$$\begin{aligned} VG &= \{i \mid \rho_i^s \le s_i - 1\}, \\ G &= \{i \mid s_i \le \rho_i^s \le 2s_i - 1\}, \\ B &= \{i \mid 2s_i \le \rho_i^s \le 3s_i - 1\}, \\ VB &= \{i \mid \rho_i^s \ge 3s_i\}. \end{aligned}$$

As intuition for the terminology, note that a "very good" layer $i \in VG$ can take care of its deviation on its own (namely, by merging its own stars into a single tree) using leaf-merges alone. A good layer $i \in G$ can also take care of its deviation on its own, but it must apply some head-merges. In contrast, bad layers (in $B \cup VB$) require the help of other layers in order to reduce their deviation.

In particular, our SNDF problem is easy for *benign profiles*, namely, profiles in which all layers are good or very good w.r.t. $\overline{c}$, as well as for profiles in which each layer is either very good for $\overline{c}$ or ($c_i = 1$ and $\rho_i = 2$). (See the full paper or [36].) Hereafter, we consider only profiles that are not benign.

The completion forest

As described earlier, given a profile $\bar{d}$ and a CSS $\bar{s}$, it is sufficient to construct a U-realization (F, ctr) conforming to $(\bar{d}, \bar{s})$, such that the number of connected components in the centers forest $cc(F[C])$ is minimal. The completion forest serves as a basic tool in our algorithm for constructing a U-realization for $\bar{d}$, where we first construct the centers forest and then add the non-centers. This tool allows us to focus only on the centers forest for some U-realization (F, ctr), and use its structure to deduce lower and upper bounds on properties of F.

Consider a profile $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ with star formation $(c_i, \rho_i)_{i=1}^{\ell}$, a forest $\mathcal{F} = (C, E)$ and a neighbor function $\text{ctr} : C \to C$ where $\text{ctr}(u) \in N[u]$ for every $u \in C$. (Note that $(\mathcal{F}, \text{ctr})$ is not a U-realization of $\bar{d}$.) Let $\overline{C} = (C_1, \ldots, C_\ell)$ be a partition of C. The tuple $(\bar{d}, \mathcal{F}, \text{ctr}, \overline{C})$ is *legal* if $\deg(u) \leq k_i$ for every $u \in C_i$.

The *completion forest for a legal* $(\bar{d}, \mathcal{F}, ctr, \overline{C})$ is a pair (F, ctr) constructed by the following Procedure $\mathcal{C}ompForest$. For every i and every $u \in C_i$, the procedure adds $k_i - \deg(u)$ new vertices to the forest, connects them to u thus increasing its degree to k_i and making it a proper k_i-center. For each vertex v that was connected to u, the procedure sets $\text{ctr}(v) = u$, thus making it a k_i-member of its center.

Note that the tuple $(\bar{d}, \mathcal{F}, \text{ctr}, \overline{C})$ is legal if and only if its completion forest is well defined, since it has to satisfy $\deg(u) \leq k_i$ for each $u \in C_i$. Also note that the completion forest (F, ctr) for $(\bar{d}, \mathcal{F}, \text{ctr}, \overline{C})$ is not necessarily a U-realization of $\bar{d}$, since the number of k_i-members in (F, ctr) might happen to be smaller than n_i. For convenience, we refer to (F, ctr) as the completion forest for $\mathcal{F}$.

For example, consider the profile $\bar{d} = (4^8, 3^6)$, where $k_1 = 4$, $k_2 = 3$. Let $(\mathcal{F} = (C, E), \text{ctr})$, where the center set $C = \{1, 2, 3, 4\}$ is partitioned into $C_1 = \{1, 2\}$ and $C_2 = \{3, 4\}$, the sets of k_1 and k_2 centers respectively. The completion forest of the above tuple is defined by "completing" the neighborhoods of C centers according to $\bar{d}$ and the layers of C. See Figure 6.

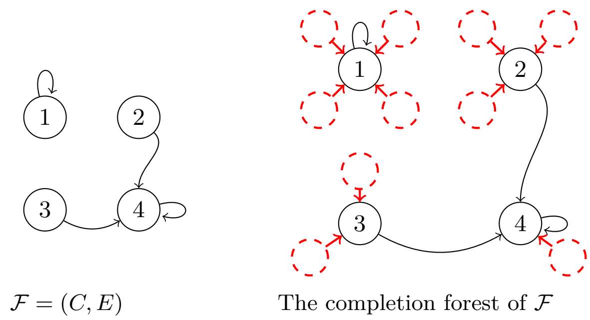


Figure 6 Completion forest example.

4 Lower bound and algorithm

Tight lower bound for SNDF profiles

Finally, we derive a tight lower bound on the minimum deviation of a given SNDF profile. To do that, we first lower bound $cc^*(\bar{d}, \bar{s})$, defined in Eq. (7), and then combine it with our lower bound on $LB_3^{dev}(\bar{d}, \bar{s})$ from Lemma 7. Our bound on $cc^*(\bar{d}, \bar{s})$ depends only on the CSS $\bar{s}$ and the profile $\bar{d}$. Our method of proving the bound is by considering an arbitrary U-realization (F, ctr), that conforms to $(\bar{d}, \bar{s})$, and proving a lower bound on $cc(F[C])$, which imply a lower bound on $cc^*(\bar{d}, \bar{s})$. This is done by decomposing the centers forest $F[C]$ in a special manner, and then reconstructing $F[C]$ while keeping track of the number of its connected components. As mentioned earlier, we may assume that the profile $\bar{d}$ is not benign.

The entire derivation of the lower bound is deferred to the full paper (see [36]). Letting

$$LB_4^{dev}(\bar{d}, \bar{s}) = \max\left\{0\ ,\ \ 2 + \sum_{i=1}^{\ell} \rho_i^s - 2\sum_{i=1}^{\ell} s_i + \max\left\{\sum_{i \in VG} s_i - \left(D_{VG} + \sum_{i \in G \cup B} \lfloor(\rho_i^s - s_i)/2\rfloor + \sum_{i \in VB} s_i\right), 0\right\}\right\}$$

we get the following.

▶ **Theorem 8.** *Let $\bar{d} = (k_i^{n_i})_{i=1}^{\ell}$ be an SNDF-profile and let $\overline{s'}$ be a CSS for $\bar{d}$. Assume that $\overline{s'}$ minimizes LB_4^{dev} over all choices of CSS $\bar{s}$ (Note that $\overline{s'}$ does not necessarily minimizes LB_3^{dev}). Then $Dev(\bar{d}) \geq LB_4^{dev}(\bar{d}, \overline{s'})$.*

Optimal algorithm

While our algorithm is based on the components introduced above, its actual operation is rather involved, hence its description is deferred to the full paper (see [36]) due to space constraints. It provides an optimal explicit construction of SNDF-realizations, which for a given profile $\bar{d}$ and sequence $\bar{s}$ yields a realization with deviation at most $LB_4^{dev}(\bar{d}, \bar{s})$, and also show how to select a sequence $\overline{s'}$ that minimizes LB_4^{dev}. Combining the above with Theorem 8 yields an explicit construction for optimal realizations. In the full paper (see [36]) we also show a more efficient solution for the decision version of SNDF, namely, decide if a given SNDF-profile $\bar{d}$ is realizable (i.e., $Dev(\bar{d}) = 0$).

5 Discussion

In this paper we introduced the selected neighbor degree realization problem and solved it when the graph is required to be acyclic. We presented a necessary and sufficient condition for realizability. In addition, we provided an algorithm that given a specification computes an upper realization with minimum deviation from the given specification. In particular, if the specification is realizable, the algorithm computes a realization.

A natural open question is to solve the realization problem on general graphs. One may also consider the problem on other graph families, such as trees or bipartite graphs. (Note that the realization problem is easy in regular graphs.) Finally, another interesting direction for future study is to consider variants of the realization problem on directed graphs.

References

1 M. Aigner and E. Triesch. Realizability and uniqueness in graphs. *Discrete Math.*, 136:3–20, 1994.

2 Takao Asano. Graphical degree sequence problems with connectivity requirements. In *4th ISAAC*, volume 762 of *LNCS*, pages 38–47, 1993.

3 A. Bar-Noy, K. Choudhary, A. Cohen, D. Peleg, and D. Rawitz. Minimum neighboring degree realization in graphs and trees. In *28th ESA*, volume 173 of *LIPIcs*, pages 10:1–10:15, 2020.

4 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Realizability of graph specifications: Characterizations and algorithms. In *25th International Colloquium on Structural Information and Communication Complexity*, volume 11085 of *LNCS*, pages 3–13, 2018.

5 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Graph profile realizations and applications to social networks. In *13th WALCOM*, volume 11355 of *LNCS*, pages 3–14, 2019.

6 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Graph realizations: Maximum degree in vertex neighborhoods. In *17th SWAT*, volume 162 of *LIPIcs*, pages 10:1–10:17, 2020.

7 Amotz Bar-Noy, David Peleg, Mor Perry, and Dror Rawitz. Composed degree-distance realizations of graphs. In *32nd IWOCA*, volume 12757 of *LNCS*, pages 63–77, 2021.

8 Vladimir Batagelj, Tomaz Pisanski, and J. M. S. Simões-Pereira. An algorithm for tree-realizability of distance matrices. *Int. J. Comput. Math.*, 34(3-4):171–176, 1990.

9 Joseph K. Blitzstein and Persi Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet Mathematics*, 6(4):489–522, 2011.

10 W. Chou and H. Frank. Survivable communication networks and the terminal capacity matrix. *IEEE Trans. Circuit Theory*, CT-17:192–197, 1970.

11 S.A. Choudum. A simple proof of the Erdös-Gallai theorem on graph sequences. *Bulletin of the Australian Mathematical Society*, 33:67–70, 1986.

12 Fan R. K. Chung, Mark W. Garrett, Ronald L. Graham, and David Shallcross. Distance realization problems with applications to internet tomography. *J. Comput. Syst. Sci.*, 63(3):432–448, 2001.

13 Brian Cloteaux. Fast sequential creation of random realizations of degree sequences. *Internet Mathematics*, 12(3):205–219, 2016.

14 Joseph C. Culberson and Piotr Rudnicki. A fast algorithm for constructing trees from distance matrices. *Inf. Process. Lett.*, 30(4):215–220, 1989.

15 P. Erdös and T. Gallai. Graphs with prescribed degrees of vertices [hungarian]. *Matematikai Lapok*, 11:264–274, 1960.

16 G. Gupta, P. Joshi, and A. Tripathi. Graphic sequences of trees and a problem of frobenius. *Czechoslovak Math. J.*, 57:49–52, 2007.

17 S. Louis Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quart. Appl. Math.*, 22:305–317, 1965.

18 S.L. Hakimi. 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.

19 V. Havel. A remark on the existence of finite graphs [in Czech]. *Casopis Pest. Mat.*, 80:477–480, 1955.

20 P.J. Kelly. A congruence theorem for trees. *Pacific J. Math.*, 7:961–968, 1957.

21 Milena Mihail and Nisheeth Vishnoi. On generating graphs with prescribed degree sequences for complex network modeling applications. *3rd Workshop on Approximation and Randomization Algorithms in Communication Networks*, 2002.

22 Elchanan Mossel and Nathan Ross. Shotgun assembly of labeled graphs. *CoRR*, abs/1504.07682, 2015. `arXiv:1504.07682`.

23 Juhani Nieminen. Realizing the distance matrix of a graph. *J. Inf. Process. Cybern.*, 12(1/2):29–31, 1976.

24 P.V. O'Neil. Ulam's conjecture and graph reconstructions. *Amer. Math. Monthly*, 77:35–43, 1970.

25 A. N. Patrinos and S. L. Hakimi. The distance matrix of a graph and its tree realizability. *Quarterly of Applied Math.*, 255, 1972.

26 G. Sierksma and H. Hoogeveen. Seven criteria for integer sequences being graphic. *J. Graph Theory*, 15:223–231, 1991.

27 J. M. S. Simões-Pereira. A note on the tree realizability of a distance matrix. *J. Combinat. Theory B*, 6:303–310, 1969.

28 J. M. S. Simões-Pereira. A note on distance matrices with unicyclic graph realizations. *Discrete Mathematics*, 65:277–287, 1987.

29 J. M. S. Simões-Pereira. An algorithm and its role in the study of optimal graph realizations of distance matrices. *Discrete Mathematics*, 79(3):299–312, 1990.

30 Hiroshi Tamura, Masakazu Sengoku, Shoji Shinoda, and Takeo Abe. Realization of a network from the upper and lower bounds of the distances (or capacities) between vertices. In *IEEE ISCAS*, pages 2545–2548. IEEE, 1993.

31 A. Tripathi and H. Tyagi. A simple criterion on degree sequences of graphs. *Discrete Applied Math.*, 156:3513–3517, 2008.

32 S.M. Ulam. *A collection of mathematical problems*. Wiley, 1960.

33 Sacha C. Varone. A constructive algorithm for realizing a distance matrix. *Eur. J. Oper. Res.*, 174(1):102–111, 2006.

34 D.L. Wang and D.J. Kleitman. On the existence of n-connected graphs with prescribed degrees ($n > 2$). *Networks*, 3:225–239, 1973.

35 N.C. Wormald. Models of random regular graphs. *Surveys in combinatorics*, 267:239–298, 1999.

36 Elad Yehezkel. Selected neighbor degree tree realization. Master's thesis, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, 2020.

37 K. A. Zaretskii. Constructing a tree on the basis of a set of distances between the hanging vertices. *Uspekhi Mat. Nauk*, 20:90–92, 1965.