Graph Realizations: Maximum Degree in Vertex Neighborhoods

Amotz Bar-Noy
City University of New York (CUNY), NY, USA
amotz@sci.brooklyn.cuny.edu

Keerti Choudhary
Tel Aviv University, Israel
keerti.choudhary@cs.tau.ac.il

David Peleg
Weizmann Institute of Science, Rehovot, Israel
david.peleg@weizmann.ac.il

Dror Rawitz
Bar Ilan University, Ramat-Gan, Israel
dror.rawitz@biu.ac.il

Abstract

The classical problem of *degree sequence realizability* asks whether or not a given sequence of n positive integers is equal to the degree sequence of some n-vertex undirected simple graph. While the realizability problem of degree sequences has been well studied for different classes of graphs, there has been relatively little work concerning the realizability of other types of information *profiles*, such as the vertex neighborhood profiles.

In this paper, we initiate the study of *neighborhood degree* profiles, wherein, our focus is on the natural problem of realizing *maximum* neighborhood degrees. More specifically, we ask the following question: *"Given a sequence D of n non-negative integers $0 \leq d_1 \leq \cdots \leq d_n$, does there exist a simple graph with vertices $v_1, \ldots, v_n$ such that for every $1 \leq i \leq n$, the maximum degree in the neighborhood of v_i is exactly d_i?"*

We provide in this work various results for maximum-neighborhood-degree for general n vertex graphs. Our results are first of its kind that studies extremal neighborhood degree profiles. For closed as well as open neighborhood degree profiles, we provide a *complete realizability criteria*. We also provide tight bounds for the number of maximum neighbouring degree profiles of length n that are realizable. Our conditions are verifiable in linear time and our realizations can be constructed in polynomial time.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Graph realization, neighborhood profile, extremum-degree

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.10

Funding W911NF-09-2-0053 (the ARL Network Science CTA), US-Israel BSF grant 2018043.

1 Introduction

Background and Motivation. In many application domains involving networks, it is common to view vertex degrees as a central parameter, providing useful information concerning the relative significance (and in certain cases, centrality) of each vertex with respect to the rest of the network, and consequently useful for understanding the network's basic properties. Given an n-vertex graph G with adjacency matrix $Adj(G)$, its *degree sequence* is a sequence consisting of its vertex degrees,

$$\text{DEG}(G) = (d_1, \ldots, d_n).$$

17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2020).
Editor: Susanne Albers; Article No. 10; pp. 10:1–10:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Given a graph G or its adjacency matrix, it is easy to extract the degree sequence. An interesting *dual* problem, sometimes referred to as the *realization* problem, concerns a situation where given a sequence of nonnegative integers D, we are asked whether there exists a graph whose degree sequence conforms to D. A sequence for which there exists a realization is called a *graphic* sequence. Erdős and Gallai [10] gave a necessary and sufficient condition for deciding whether a given sequence of integers is graphic (also implying an $O(n)$ decision algorithm). Havel and Hakimi [12, 13] gave a recursive algorithm that given a sequence of integers computes in $O(m)$ time a realizing m-edge graph, if such a graph exists.

Over the years, various extensions of the degree realization problem were studied as well, cf. [1, 3, 23], concerning different characterizations of degree-profiles. The motivation underlying the current paper is rooted in the observation that realization questions of a similar nature pose themselves naturally in a large variety of *other* application contexts, where given *some* type of information profile specifying the desired vertex properties (be it concerning degrees, distances, centrality, or any other property of significance), it can be asked whether there exists a graph conforming to the specified profile. Broadly speaking, this type of investigation may arise, and find potential applications, both in scientific contexts, where the information profile reflects measurement results obtained from some natural network of unknown structure, and the goal is to obtain a model that may explain these measurements, and in engineering contexts, where the information profile represents a specification with some desired properties, and the goal is to find an implementation in the form of a network conforming to that specification.

This basic observation motivates a vast research direction, which was little studied over the last five decades. In this paper we make a step towards a systematic study of one specific type of information profiles, concerning *neighborhood degree* profiles. Such profiles are of theoretical interest in context of social networks (where degrees often reflect influence and centrality, and consequently neighboring degrees reflect "closeness to power"). Neighborhood degrees were considered before in [5], where the profile associated with each vertex i is the *list* of degrees of all vertices in i's neighborhood. In contrast, we focus here on "single parameter" profiles, where the information associated with each vertex relates to a single degree in its neighborhood. The first natural problem in this direction concern the *maximum* degrees in the vertex neighborhoods. For each vertex i, let d_i denote the maximum vertex degree in i's neighborhood. Then $\textsc{MaxNDeg}(G) = (d_1, \ldots, d_n)$ is the maximum neighborhood degree profile of G. The same realizability questions asked above for degree sequences can be posed for neighborhood degree profiles as well. This brings us to the following central question of our work:

MAXIMUM NEIGHBORHOOD DEGREE REALIZATION

Input: A sequence $D = (d_1, \ldots, d_n)$ of non-negative integers.

Question: Is there a graph G of size n such that the *maximum* degree in the neighborhood of *i-th* vertex in G is exactly equal to d_i ?

Our Contributions. We now discuss our contributions in detail. For simplicity, we represent the input vector D alternatively in a more compact format as $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$, where n_i's are positive integers with $\sum_{i=1}^{\ell} n_i = n$; here the specification requires that G contains exactly n_i vertices whose maximum degree in neighborhood is d_i. We may assume that $d_\ell > d_{\ell-1} > \cdots > d_1 \geq 1$ (noting that vertices with max degree zero are necessarily singletons and can be handled separately).

We perform an extensive study of maximum neighborhood degree profiles.

1. We obtain the necessary and sufficient conditions for $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ to be MaxNDeg realizable for closed neighborhoods in Section 3. For general graphs we obtain the following characterization.

   $$d_\ell \leq n_\ell - 1, \text{ and } (d_1 \geq 2 \text{ or } n_1 \text{ is even})$$

   We also study the version of the problem in which the realization is required to be connected. Our characterization is as follows.

   $$d_\ell \leq n_\ell - 1, \text{ and } (d_1 \geq 2 \text{ or } \sigma = (1^2))$$

2. Next, we consider the open neighborhoods, wherein a vertex is not counted in its own neighborhood. These are more involved, and are discussed in Section 4. Our results for open neighborhood are summarised in Table 1.

Table 1 Max-neighboring-degree realizability for open neighborhood.

Graph	Complete characterisation
Connected Graphs	$d_\ell \leq \min\{n_\ell, n-1\}$ $d_1 \geq 2$ or $\sigma = (d^d, 1^1)$ or $\sigma = (1^2)$ $\sigma \neq (d_\ell^{d_\ell+1}, 2^1)$
General graphs	σ can be split[1] into two profiles σ_1 and σ_2 such that (i) σ_1 has a *connected* MaxNDeg-open realization, and (ii) $\sigma_2 = (1^{2\alpha})$ or $\sigma_2 = (d^d, 1^{2\alpha+1})$, for integers $d \geq 2, \alpha \geq 0$.

3. Enumerating realizable maximum neighborhood degree profiles: The simplicity of above characterizations enables us to enumerate and count the number of realizable profiles. This gives a way to sample uniformly a random MaxNDeg realizable profile. In contrast, counting and sampling are open problems for the traditional degree sequence realizability problem. In the full version of this paper, we show that the number of realizable profiles of length n is $\lceil (2^{n-1} + (-1)^n)/3 \rceil$ for general graphs and 2^{n-3} for connected graphs. In comparison, the total number of non-increasing sequences of length n on the numbers $1, \ldots, n-1$ is $\Theta(4^n/\sqrt{n})$.

Through this work, we make the first crucial step by obtaining a closed form characterization for maximum-neighborhood-degree profiles, and solving the realization problem for such profiles with an efficient algorithm. As was done with degree sequences, we solve the problem for both *connected* graphs as well as *general* graphs. Our conditions are verifiable in linear time and our realizations are computable in polynomial time. Finally, in contrast to the degree-sequence case, we are able to count the number of distinct realizable maximum-neighborhood-degree sequences.

Further Related Work. Many works 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

[1] A profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ is said to be split into two profiles $\sigma_1 = (d_\ell^{p_\ell}, \cdots, d_1^{p_1})$ and $\sigma_2 = (d_\ell^{q_\ell}, \cdots, d_1^{q_1})$ if $n_i = p_i + q_i$ for each $i \in [1, \ell]$.

a given degree sequence defines a unique realizing graph (a.k.a. the *graph reconstruction* problem), see [6, 8, 10, 11, 12, 13, 14, 15, 16, 19, 20, 21, 22, 24]. Other works such as [7, 9, 17] studied interesting applications in the context of social networks.

To the best of our knowledge, the MaxNDeg realization problems have not been explored so far. There are only two related problems that we are aware of. The first is the *shotgun assembly* problem [18], where the characteristic associated with the vertex i is some description of its neighborhood up to radius r. The second is the *neighborhood degree lists* problem [5], where the characteristic associated with the vertex i is the list of degrees of all vertices in i's neighborhood. We point out that in contrast to these studies, our MaxNDeg problem applies to a more restricted profile (with a single number characterizing each vertex), and the techniques involves are totally different from those of [5, 18]. Several other realization problems are surveyed in [2, 4].

2 Preliminaries

Let H be an undirected graph. We use $V(H)$ and $E(H)$ to respectively denote the vertex set and the edge set of graph H. For a vertex $x \in V(H)$, let $deg_H(x)$ denote the degree of x in H. Let $N_H[x] = \{x\} \cup \{y \mid (x,y) \in E(H)\}$ be the (closed) neighborhood of x in H. For a set $W \subseteq V(H)$, we denote by $N_H(W)$, the set of all the vertices lying outside set W that are adjacent to some vertex in W, that is, $N_H(W) = (\bigcup_{w \in W} N[w]) \setminus W$. Given a vertex v in H, the maximum degree in the neighborhood of v, namely $\textsc{MaxNDeg}_H(v)$, is defined to be the maximum over the degrees of all the vertices in the neighborhood of v. Similarly, the maximum degree in the open neighborhood $(N_H[v] \setminus v)$ of vertex v, namely $\textsc{MaxNDeg}^-{}_H(v)$ is the maximum over the degrees of all the vertices present in the open neighborhood of v. Given a set of vertices A in a graph H, we denote by $H[A]$ the subgraph of H induced by the vertices of A. For a set A and a vertex $x \in V(H)$, we denote by $A \cup x$ and $A \setminus x$, respectively, the sets $A \cup \{x\}$ and $A \setminus \{x\}$. When the graph is clear from context, for simplicity, we omit the subscripts H in all our notations. Finally, given two integers $i \leq j$, we define $[i,j] = \{i, i+1, \ldots, j\}$.

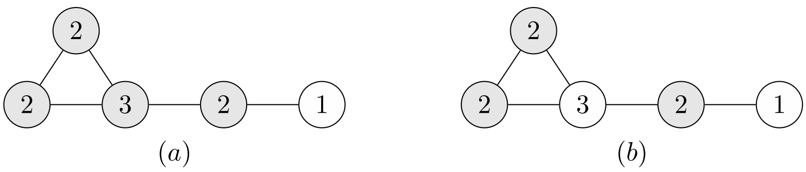

Figure 1 A comparison of the MaxNDeg realization of $(3^4, 2^1)$ and a $\textsc{MaxNDeg}^-$ realization of $(3^3, 2^2)$.

Next we formally define the realizable profiles.

▶ **Definition 1.** *A profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ satisfying $d_\ell > d_{\ell-1} > \cdots > d_1 > 0$ is said to be MaxNDeg realizable if there exists a graph G on $n = n_1 + \cdots + n_\ell$ vertices that for each $i \in [1,\ell]$ contains exactly n_i vertices whose MaxNDeg is d_i. Equivalently, $|\{v \in V(G) : \textsc{MaxNDeg}(v) = d_i\}| = n_i$.*

▶ **Definition 2.** *A profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ is said to be $\textsc{MaxNDeg}^-$ realizable if there exists a graph G on $n = n_1 + \cdots + n_\ell$ vertices that for each $i \in [1,\ell]$ contains exactly n_i vertices whose $\textsc{MaxNDeg}^-$ is d_i. Equivalently, $|\{v \in V(G) : \textsc{MaxNDeg}^-(v) = d_i\}| = n_i$*

The figure depicts a MaxNDeg realization of $(3^4, 2^1)$. (The numbers in the vertices represent their degrees.) Note that in the open neighborhood model, the corresponding MaxNDeg$^-$ profile becomes $(3^3, 2^2)$.

3 Realizing maximum neighborhood degree profiles

In this section, we provide a complete characterization of MaxNDeg profiles. For simplicity, we first discuss the uniform scenario of $\sigma = (d^k)$. Observe that a star graph $K_{1,d}$ is MaxNDeg realization of the profile (d^{d+1}). We show in the following lemma that, by identifying together vertices in different copies of $K_{1,d}$, it is always possible to realize the profile (d^k), whenever $k \geq d+1$.

▶ **Lemma 3.** *For any positive integers d and k, the profile $\sigma = (d^k)$ is MaxNDeg realizable whenever $k \geq d+1$. Moreover, we can always compute in $O(k)$ time a connected realization that has an independent set, say S, of size d such that all vertices in S have degree at most 2, and at least two vertices in S have degree 1.*

Proof. Let α be the smallest integer such that $k \leq 2 + \alpha(d-1)$. We first construct a caterpillar[2] T as follows. Take a path $P = (s_0, s_1, \ldots, s_\alpha, s_{\alpha+1})$ of length $\alpha + 1$. Connect each internal vertex s_i (here $i \in [1, \alpha]$) with a set of $d-2$ new vertices, so that the degree of s_i is d. (See Figure 2). Note that the MaxNDeg of each vertex $v \in T$ is d.

Now if $k = 2+\alpha(d-1)$, then T serves as our required realizing graph. If $k < 2+\alpha(d-1)$, then $\alpha \geq 2$ since $k \geq d+1$. The tree T is "almost" a realizing graph for the profile, except that it has too many vertices. Let $r = 2+\alpha(d-1)-k$ denote the number of excess vertices in T that need to be *removed*. The r vertices can be removed as follows. Take any two distinct internal vertices s_i and s_j on P, and let $s_i^1, \ldots, s_i^{d-2}$ and $s_j^1, \ldots, s_j^{d-2}$, respectively, denote the neighbors of s_i and s_j not lying on P. Let G be the graph obtained by merging vertices s_i^ℓ and s_j^ℓ into a single vertex for $\ell \in [1, r]$. (See Figure 2). Since the number of vertices was decreased by r, G now contains exactly n vertices. The degree of vertices $s_1, s_2, \ldots, s_\alpha$ remains d, and the degree of all other vertices is at most 2, therefore MaxNDeg$(v) = d$ for each $v \in G$, so G is a realization of the profile σ.

Finally, in the resultant graph G, the end points of P (i.e. s_0 and $s_{\alpha+1}$) have degree 1, and there are $d-2$ other vertices, namely $s_i^1, \ldots, s_i^{d-2}$ (or $s_j^1, \ldots, s_j^{d-2}$), that have degree bounded by 2. Therefore we set S to these d vertices. It is easy to verify that S is indeed an independent set. ◀

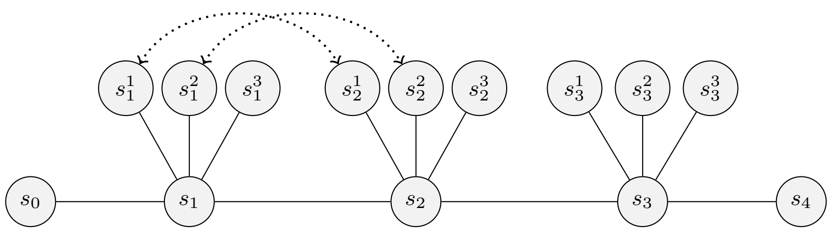

Figure 2 A caterpillar for $d = 5$ and $\alpha = 3$. If $k = 12$, then $r = 2$, and we merge (i) s_1^1, s_2^1, and (ii) s_1^2, s_2^2.

[2] A caterpillar is a tree in which all the vertices are within distance one of a central path.

3.1 An incremental procedure for computing MaxNDeg realizations

We explain here our main building block, procedure AddLayer, that will be useful in incrementally building graph realizations in a decreasing order of maximum degrees. Given a partially computed connected graph H and integers d and k satisfying $d \geq 2$ and $k \geq 1$, the procedure adds to H a set W of k new vertices such that $\text{MaxNDeg}(w) = d$, for each $w \in W$. The reader may assume that $\text{MaxNDeg}(v) \geq d$, for each existing vertex $v \in V(H)$. The procedure takes in as an input a sufficiently large vertex list L (of size $d-1$) that forms an independent set in H, and whose vertices have small degree (that is, at most $d-1$). Moreover, in order to accommodate its iterative use, each invocation of the procedure also generates and outputs a *new* list, to be used in the further iterations.

Procedure AddLayer. The input to procedure AddLayer (H, L, k, d) is a connected graph H and a list $L = (a_1, \ldots, a_{d-1})$ of vertices in H whose degree is bounded above by $d-1$. The first step is to add to H a set of k new vertices $W = \{w_1, w_2, \ldots, w_k\}$. Next, the new vertices are connected to the vertices of L and to themselves so as to ensure that $\text{MaxNDeg}(w) = d$ for every $w \in W$. Depending upon whether or not $k < d$, there are two separate cases. (Refer to Algorithm 1 for pseudocode).

Algorithm 1 AddLayer (H, L, k, d).

Let the list L be $(a_1, a_2, \ldots, a_{d-1})$.
Add to H a set $W = \{w_1, \ldots, w_k\}$ of k new vertices.
case $(k < d)$ **do**
  Set $count = k$ and $i = d-1$.
  while $(count \neq 0)$ **do**
    Let $r = \min\{d - deg(a_i), count\}$.
    Add edges $(a_i, w_{count-t})$ to H for $t \in [0, r-1]$.
    Decrement i by 1 and $count$ by r.
  foreach $j \in [d-1, \ldots, 2, 1]$ **do**
    If $deg(a_i) = d$ then break the for loop.
    If $(j < i)$ then add edge (a_j, a_i) to H.
    If $(j > i)$ then add an edge between a_i and an arbitrary vertex in $N(a_j) \cap W$.
  Set L to be prefix of $(w_1, w_2, \ldots, w_k, a_1, a_2, \ldots, a_{i-1})$ of size $d-2$.
case $(k \geq d)$ **do**
  Use Lemma 3 to compute over independent set $(W \cup \{a_1\})$ the graph, say $\bar{H}$, realizing the profile (d^{k+1}) such that $deg_{\bar{H}}(a_1) = 1$.
  Add edges between a_1 and any arbitrary $d - deg(a_1)$ vertices in set $\{a_2, a_3, \ldots, a_{d-1}\}$.
  Let $b_1, \ldots, b_{d-1} \in \bar{H} \setminus a_1$ be such that $1 = deg_{\bar{H}}(b_1) \leq \cdots \leq deg_{\bar{H}}(b_{d-1}) \leq 2$.
  Set $L = (b_1, b_2, \ldots, b_{d-2})$.
Output L.

Let us first consider the case $k \leq d-1$. In this case we add edges from vertices in W to a subset of vertices from L such that those vertices in L will have degree d and therefore will imply $\text{MaxNDeg}(w) = d$, for every $w \in W$. We initialize two variables, *count* and i, respectively, to k and $d-1$. The variable *count* holds, at any instant of time, the number of vertices in W that still need to be connected to vertices in L. While $count > 0$, the procedure performs the following steps:

(i) compute $r = \min\{d - \mathit{deg}(a_i), \mathit{count}\}$, the maximum number of vertices in W that can be connected to vertex a_i;
(ii) connect a_i to following r vertices in W: $w_{count-(r-1)}, w_{count-(r-2)}, \dots, w_{count-1}, w_{count}$; and
(iii) decrease *count* by r, and i by 1.

When $\mathit{count} = 0$, the vertices $a_i, a_{i+1}, \dots, a_{d-1}$ are connected to at least one vertex in W (this implies $d - i \leq k$). It is also easy to verify that at this stage, $\mathit{deg}(a_{d-1}) = \mathit{deg}(a_{d-2}) = \cdots = \mathit{deg}(a_{i+1}) = d$, and $\mathit{deg}(a_i) \leq d$. Since the input graph H was connected, in the beginning of the execution $\mathit{deg}(a_i) \geq 1$, and by connecting a_i to at least one vertex in W, specifically to w_1, its degree is increased at least by one. So at most $d - 2$ edges need to be added to a_i to ensure that its degree is exactly d. The procedure performs the following operation for each $j \in [d-1, d-2, \dots, 2, 1]$ (in the given order) until $\mathit{deg}(a_i) = d$:
(i) if $j < i$ then add edge (a_j, a_i) to H, and
(ii) if $j > i$ then add an edge between a_i and an arbitrary neighbor of a_j lying in W.
Since $\mathit{deg}(a_i) = \mathit{deg}(a_{i+1}) = \cdots = \mathit{deg}(a_{d-1}) = d$, and $\mathit{deg}(w) \leq 2$ for every $w \in W$, it follows that $\textsc{MaxNDeg}(w) = d$, for each $w \in W$. In the end, we set a *new* list L containing the first $d - 2$ vertices in the sequence $(w_1, w_2, \dots, w_k, a_1, a_2, \dots, a_{i-1})$. This is possible since $k + i - 1 \geq d - 2$ due to the fact that $d - i \leq k$. (Later on we bound the degrees of the vertices in the new list.)

Now we consider the case $k \geq d$. The procedure uses Lemma 3 to compute over the independent set $W \cup \{a_1\}$ a graph $\bar{H}$ realizing the profile (d^{k+1}) such that $\mathit{deg}_{\bar{H}}(a_1) = 1$. Notice that in the beginning of the execution, $\mathit{deg}(a_1) \in [1, d-1]$, and it is increased by one by adding $\bar{H}$ over the set $W \cup \{a_1\}$. So now $\mathit{deg}(a_1) \in [2, d]$. To ensure $\mathit{deg}(a_1) = d$, at most $d - 2$ more edges need to be added to a_1. Edges are added between a_1 and any arbitrary $d - \mathit{deg}(a_1)$ vertices in set $\{a_2, a_3, \dots, a_{d-1}\}$. This ensures that every $w \in W$ has $\textsc{MaxNDeg}(w) = d$. By Lemma 3, $\bar{H} \setminus \{a_1\}$ contains an independent set of $d - 1$ vertices, say $b_1, \dots, b_{d-1}$, such that $1 = \mathit{deg}_{\bar{H}}(b_1) \leq \mathit{deg}_{\bar{H}}(b_2) \leq \cdots \leq \mathit{deg}_{\bar{H}}(b_{d-1}) \leq 2$. In the end, the procedure creates a *new* list $L = (b_1, b_2, \dots, b_{d-2})$.

For sake of better understanding, in the rest of paper, we denote by H_{old}, L_{old} and H_{new}, L_{new} respectively the graph and the list before and after the execution of Procedure $\textsc{AddLayer}$. Observe that $V(H_{new}) = V(H_{old}) \cup W$.

The following two lemmas follow from the description of algorithm.

▶ **Lemma 4.** *Each $w \in W$ satisfies $\textsc{MaxNDeg}(w) = d$, and $N(w) \subseteq W \cup L_{old}$.*

▶ **Lemma 5.** *Each $a \in L_{old} \setminus L_{new}$ satisfies $\mathit{deg}_{H_{new}}(a) \leq d$, and each $a \in L_{old} \cap L_{new}$ satisfies $\mathit{deg}_{H_{new}}(a) \leq \mathit{deg}_{H_{old}}(a) + 1$.*

It is also easy to verify that the total execution time of Procedure $\textsc{AddLayer}$ is $O(k + d)$.

The Inheritance Property. Till now, we showed that given an independent list of $d - 1$ vertices of degree at most $d - 1$ in a graph H, we can add $k \geq 1$ vertices to H such that the $\textsc{MaxNDeg}$ of these k vertices is d. In order to iteratively use this algorithm to add vertices of smaller $\textsc{MaxNDeg}$ values ($\lneq d$) we require that the list L_{new} computed by Procedure $\textsc{AddLayer}$ should satisfy following three constraints:
(i) The size of L_{new} should be $d - 2$;
(ii) the vertices of L_{new} should form an independent set; and most importantly,
(iii) the vertices in L_{new} should have degree at most $d - 2$.

In order to ensure these constraints on L_{new}, we further impose the constraint that the list L_{old} is a valid list; this is formally defined as below.

▶ **Definition 6** (Valid List). *A list $L = (a_1, a_2, \ldots, a_t)$ in a graph G is said to be "valid" with respect to G if the following two conditions hold:*
(i) *for each $i \in [1, t]$, $deg(a_i) \leq i$, and*
(ii) *the vertices of L form an independent set in G.*

We next prove the inheritance property of our procedure.

▶ **Lemma 7** (Inheritance property). *If the input list L_{old} in Procedure* AddLayer *is valid, then the output list L_{new} is valid as well.*

Proof. We first consider the case $k \leq d-1$. Let i be the smallest index such that vertices $a_i, a_{i+1}, \ldots, a_{d-1}$ are adjacent to some vertex of W in H_{new}. (That is, i is the index when Procedure AddLayer exits the while loop). Recall that in the graph H_{new}, $w_1 \in W$ is a neighbor of a_i. Also, to increase the degree of a_i to d, we connect a_i to some/all vertices in $a_1, \ldots, a_{i-1}$, and some/all neighbors of $a_{i+1}, \ldots, a_{d-1}$ lying in W. Therefore the vertex set $W \cup \{a_1, \ldots, a_{i-1}\}$ is independent in H_{new}. Also, its size at least $d-1$, as we showed that $k \geq d-i$. Since the list $L_{old} = (a_1, a_2, \ldots, a_{d-1})$ is valid in the beginning of the execution of Procedure AddLayer, it follows that in H_{old}, $deg(a_j) \leq j$ for $j \in [1, d-1]$. So by Lemma 5, in H_{new}, (i) $deg(a_j) \leq j+1$ for $j \in [1, i-1]$, (ii) $deg(w_1) = 1$, and (iii) the degree of each other vertex in $W \setminus w_1$ is at most 2. Consequently, $(w_1, \cdots, w_k)$ is a valid list of length at least $d - i \geq 1$. Since $deg(a_j) \leq j+1$ for $j \in [1, i-1]$, the list $(w_1, \cdots, w_k, a_1, \ldots, a_{i-1})$ is valid and has length at least $d-1$. Truncating this to length $d-2$ again gives us a valid list.

We now consider the case $k \geq d$. By Lemma 3, $H[W \cup \{a_1\}] = \bar{H}$ contains an independent set $\{b_1, b_2, \ldots, b_{d-1}\} \subseteq W$ such that $deg(b_1) = 1$ and $deg(b_j) \leq 2$ for $j \in [2, d-1]$. Therefore, $(b_1, b_2, \ldots, b_{d-2})$ is a valid list of length $d-2$ in H_{new}. ◀

The following proposition summarizes the above discussion.

▶ **Proposition 8.** *For any integers $d \geq 2$, $k \geq 1$, and any connected graph H containing a valid list L of size $d-1$, procedure* AddLayer *adds to H in $O(k+d)$ time, a set W of k new vertices such that* MaxNDeg*$(w) = d$, for every $w \in W$. All the edges added to H lie in $W \times (W \cup L)$. Moreover, $deg_H(a) \leq d$, for every $a \in L$, and the updated graph remains connected and contains a new valid list of size $d-2$.*

3.2 The main algorithm

We now present the main algorithm for computing the realizing graph using Procedure AddLayer.

Let $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ be any profile satisfying $d_\ell \leq n_\ell - 1$ and $d_1 \geq 2$. The construction of a connected graph realizing σ is as follows (refer to Algorithm 2 for pseudocode). We first use Lemma 3 to initialize G to be the graph realizing the profile $(d_\ell^{n_\ell})$. Recall G contains an independent set, say $W = \{w_1, w_2, \ldots, w_{d_\ell}\}$, satisfying the condition that the degree of the first two vertices is one, and the degree of the remaining vertices is at most two. Set $L_{\ell-1} = (w_1, w_2, \ldots, w_{d_{\ell-1}-1})$ (notice that $d_{\ell-1} - 1 \leq d_\ell$). It is easy to verify that this list is valid. Next, for each $i = \ell - 1$ to 1, perform the following steps:

(i) Taking as input the valid list L_i of size $d_i - 1$, execute Procedure AddLayer (G, L_i, n_i, d_i) to add n_i new vertices to G. The procedure returns a valid list L_{i-1} of size $d_i - 2$.
(ii) Truncate the list L_{i-1} to contain only the first $d_{i-1} - 1 (\leq d_i - 2)$ vertices. The truncated list remains valid since any prefix of a valid list is valid.

Algorithm 2 MaxNDeg realization of $\sigma = (d_\ell^{n_\ell}, \ldots, d_1^{n_1})$.

Input: A sequence $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ satisfying $d_\ell \leq n_\ell - 1$ and $d_1 \geq 2$.

1 Initialize G to be the graph obtained from Lemma 3 that realizes the profile $(d_\ell^{n_\ell})$.
2 Let $L_{\ell-1}$ be a valid list in G of size $d_{\ell-1} - 1$.
3 **for** ($i = \ell - 1$ *to* 1) **do**
4 $\quad L_{i-1} \leftarrow$ AddLayer (G, L_i, n_i, d_i).
5 $\quad$ Truncate list L_{i-1} to contain only the first $d_{i-1} - 1 (\leq d_i - 2)$ vertices.
6 Output G.

Proof of Correctness. Let V_ℓ denote the set of vertices in graph G initialized in step 1, and for $i \in [1, \ell-1]$, let V_i denote the set of n_i new vertices added to graph G in iteration i of the for loop. Also for $i \in [1, \ell]$, let G_i be the graph induced by vertices $V_i \cup \cdots \cup V_\ell$. The following lemma proves the correctness.

▶ **Lemma 9.** *For any $i \in [1, \ell]$, graph G_i is a* MaxNDeg *realization of profile $(d_\ell^{n_\ell}, \cdots, d_i^{n_i})$, and for any $j \in [i, \ell]$ and any $v \in V_j$, $deg_{G_i}(v) \leq \text{MaxNDeg}_{G_i}(v) = d_j$.*

Proof. We prove the claim by induction on the iterations of the for loop. The base case is for index ℓ, and by Lemma 3 we have that $deg_{G_\ell}(v) \leq \text{MaxNDeg}_{G_\ell}(v) = d_\ell$, for every $v \in V_\ell$. For the inductive step, we assume that the claim holds for $i+1$, and prove the claim for i. Consider any vertex v in G_i. We have two cases.

1. $v \in V_i$: In this case by Proposition 8 we have that $deg_{G_i}(v) \leq \text{MaxNDeg}_{G_i}(v) = d_i$.
2. $v \in V_j$, for $j > i$: We first show that for any vertex $w \in N_{G_i}[v]$, $deg_{G_i}(w) \leq d_j$. If $w \in V_i$, then we already showed $deg_{G_i}(w) \leq d_i$. So let us consider the case $w \in V_{i+1} \cup \cdots \cup V_\ell$. Now if $w \in L_i$ participates in Procedure AddLayer (G, L_i, n_i, d_i), then by Proposition 8, in the updated graph $deg_{G_i}(w) \leq d_i \lneq d_j$. If $w \notin L_i$, then the degree of w is unaltered in the i^{th} iteration, and thus $deg_{G_i}(w) = deg_{G_{i+1}}(w) \leq \text{MaxNDeg}_{G_{i+1}}(v) = d_j$ by the inductive hypothesis. It follows that MaxNDeg(v) remains unaltered due to iteration i, and thus $\text{MaxNDeg}_{G_i}(v) = \text{MaxNDeg}_{G_{i+1}}(v) = d_j$. ◀

The execution time of the algorithm is $O\big(\sum_{i=1}^{\ell}(n_i + d_i)\big)$. This is also optimal. Indeed, any connected graph realizing σ must contain $\Omega(n_1 + n_2 + \cdots + n_\ell)$ edges as the degrees of all vertices must be non-zero. Also, the graph must contain at least one vertex of each of the degrees $d_1, d_2, \ldots, d_\ell$, and therefore must have $\Omega(d_1 + d_2 + \cdots + d_\ell)$ edges. In other words, any realizing graph must contain $\Omega\big(\sum_{i=1}^{\ell}(n_i + d_i)\big)$ edges, and thus the computation time must be at least $\Omega\big(\sum_{i=1}^{\ell}(n_i + d_i)\big)$. The following theorem is immediate from the above discussions.

▶ **Theorem 10.** *There exists an algorithm that given any profile $\sigma = (d_\ell^{n_\ell}, \ldots, d_1^{n_1})$ satisfying $d_\ell \leq n_\ell - 1$ and $d_1 \geq 2$ computes in optimal time a connected* MaxNDeg *realization of σ.*

3.3 A complete characterization for MaxNDeg realizable profiles

The necessary conditions for MaxNDeg realizability is as follows.

▶ **Lemma 11.** *A necessary condition for a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ to be* MaxNDeg *realizable is $d_\ell \leq n_\ell - 1$.*

Proof. Suppose σ is MaxNDeg realizable by a graph G. Then G must contain a vertex, say w, of degree d_ℓ in G. Since d_ℓ is the maximum degree in G, the MaxNDeg of all the $d_\ell + 1$ vertices in $N[w]$ must be d_ℓ. Thus $n_\ell \geq d_\ell + 1$. ◀

Consider a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ realizable by a connected graph. If $d_1 = 1$, then the graph must contain a vertex, say v, of degree 1, and the vertices in $N[v]$ must also have degree 1. The only possibility for such a graph is a single edge graph on two vertices. Thus in this case $\sigma = (1^2)$. If $d_1 \geq 2$, then by Lemma 11, for σ to be realizable in this case we need that $n_\ell \geq d_\ell + 1$. Also, by Theorem 10, under these two conditions σ is always realizable. We thus have the following theorem.

▶ **Theorem 12.** *For a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ to be MaxNDeg realizable by a connected graph the necessary and sufficient condition is that either*
- **(i)** *$n_\ell \geq d_\ell + 1$ and $d_1 \geq 2$, or*
- **(ii)** *$\sigma = (1^2)$.*

Now if $d_1 = 1$, then n_1 must be even, since the vertices v with $\text{MaxNDeg}(v) = 1$ must form a disjoint union of exactly $n_1/2$ edges. So for general graphs we have the following theorem.

▶ **Theorem 13.** *For a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ to be MaxNDeg realizable by a general graph the necessary and sufficient conditions are that $d_\ell \geq n_\ell - 1$, and either n_1 is even or $d_1 \geq 2$.*

3.4 Discussion

We briefly discuss the reasons behind the innateness in our incremental construction. Let us consider the MaxNDeg profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ for a graph $G = (V, E)$. For $1 \leq i \leq \ell$, let $W_i \subseteq V$ be the set of vertices whose MaxNDeg in G is at least d_i. Note that for any vertex $v \in W_i$, a vertex having maximum degree in $N_G[v]$ (say x) must be contained in W_i. Moreover, all the neighbors of x must also lie in W_i. It follows that the degree of x remains unaltered when restricted to the induced subgraph $G[W_i]$, and $\text{MaxNDeg}_G(v) = \text{MaxNDeg}_{G[W_i]}(v)$. Hence, MaxNDeg profiles satisfy the following nice *substructure property*, which justifies our incremental algorithm for computing their realizations.

▶ **Substructure Property.** *The induced graph $G_i = G[W_i]$ is a MaxNDeg realization of the partial profile $(d_\ell^{n_\ell}, \cdots, d_i^{n_i})$, for each $1 \leq i \leq \ell$.*

Observe that in the case of MaxNDeg$^-$ profiles, unfortunately, the nice *sub-structure property* does not always hold, which in turn increases the complexity of the problem. For example, for the graph considered in Figure 1, the profile $\sigma = (3^3, 2^2)$ is MaxNDeg$^-$ realizable, however, the subsequence (3^3) is not MaxNDeg$^-$ realizable.

4 Realizing maximum open neighborhood-degree profiles

4.1 Pseudo-valid List

We begin by stating the following lemmas that are an extension of Lemma 3 and Proposition 8 presented in Section 3 for MaxNDeg profiles.

▶ **Lemma 14.** *For any positive integers d and k, the profile $\sigma = (d^k)$ is* MaxNDeg$^-$ *realizable whenever $k \geq d+2$. Moreover, we can always compute in $O(k)$ time a connected realization that contains an independent set having*
(i) *two vertices of degree 1, and*
(ii) *$d-2$ other vertices of degree at most 2.*

▶ **Proposition 15.** *For any integers $d \geq 2$, $k \geq 1$, and any connected graph H containing a valid list L of size $d-1$, procedure* AddLayer *adds to H in $O(k+d)$ time, a set W of k new vertices such that* MaxNDeg$^-(w) = d$*, for every $w \in W$. All the edges added to H lie in $W \times (W \cup L)$. Moreover, $deg_H(a) \leq d$, for every $a \in L$, and the updated graph remains connected and contains a new valid list of size $d-2$.*

It is important to note that though the Proposition 15 holds for the open-neighborhoods it can not be directly used to incrementally compute the realizations. This is because for the profiles $\sigma = (d_\ell^{d_\ell+1})$ unlike the scenario of MaxNDeg realization, there is no MaxNDeg$^-$ realization that contains a valid list (See Lemma 18 for further details). This motivates us to define pseudo-valid lists.

▶ **Definition 16.** *A list $L = (a_1, a_2, \ldots, a_t)$ in a graph H is said to be "pseudo-valid" with respect to H if*
(i) *for each $i \in [1,t]$, $deg(a_i) = 2$, and*
(ii) *the vertices of L form an independent set.*
Note that the only deviation that prevents L from being a valid list is that $deg(a_1)$ is 2 instead of 1.

We next state two lemmas that are crucial in obtaining MaxNDeg$^-$ realizations in the scenarios $n_\ell = d_\ell$ and $n_\ell = d_\ell + 1$.

▶ **Lemma 17.** *For any integers $d > \bar{d} \geq 2$, the profile $\sigma = (d^d, \bar{d}^1)$ is* MaxNDeg$^-$ *realizable. Moreover, in $O(d)$ time we can compute a connected realization that contains a valid list of size $d-1$.*

Proof. The construction of G is as follows. Take a vertex z and connect it to $d-1$ other vertices $v_1, \ldots, v_{d-1}$. Next take another vertex y and connect to $v_1, \ldots, v_{\bar{d}-1}$ (recall $2 \leq \bar{d} < d$). Also connect z to y. In the resulting graph G, $deg(z) = d$, $deg(y) = \bar{d}$, and $deg(v_i) \leq 2$ for $i \in [1, d-1]$. Also, v_{d-1} is not adjacent to y as $\bar{d} < d$, thus $deg(v_{d-1}) = 1$. Therefore, MaxNDeg$^-(z) = \bar{d}$, MaxNDeg$^-(y) = d$, and MaxNDeg$^-(v_i) = d$, for $i \in [1, d-1]$. It is also easy to verify that $(v_{d-1}, \ldots, v_{\bar{d}-1}, \ldots, v_2, v_1)$ is a valid list in G. ◀

▶ **Lemma 18.** *For any integer $d \geq 2$, the profile $\sigma = (d^{d+1})$ is* MaxNDeg$^-$ *realizable. Moreover, a connected realization that contains an independent set having $d-1$ vertices of degree 2 can be compute in $O(d)$ time. However, none of the graphs realizing σ can contain a vertex of degree 1.*

Proof. The construction of graph G realizing σ is very similar to the previous lemma. Take two vertex-sets, namely, $U = \{u_1, u_2\}$ and $W = \{w_1, \ldots, w_{d-1}\}$. Add to G the edge (u_1, u_2), and for each $i \in [1, d-1]$, add to G the edges (u_1, w_i) and (u_2, w_i). This ensures that $deg(u_1) = deg(u_2) = d$ and $deg(w_i) = 2$ for $i \in [1, d-1]$. So G contains $d+1$ vertices with MaxNDeg$^-$ equal to d. Also, W is an independent set of size $d-1$ in G and $deg(w_i) = 2$, for every vertex $w_i \in W$.

Next, let H be any MaxNDeg$^-$ realizing graph of σ. Then H must contain two vertices, say x and y, of degree d, since a single vertex of degree d in H can guarantee MaxNDeg$^- = d$ for at most d vertices. Next notice that $N[x] = N[y]$, because otherwise H will contain more than $d+1$ vertices. This implies that all the vertices in H, other than x and y, are adjacent to both x and y. Therefore, each of the vertices in H must have degree at least two. ◀

The next lemma shows that AddLayer outputs a valid list, even when the input list is pseudo-valid.

▶ **Lemma 19.** *In procedure AddLayer, the list L_{new} is valid even when the list L_{old} is pseudo-valid and the parameter d satisfies $d \geq 3$.*

Proof. We borrow notations from the proof of Lemma 7. As before, we have two separate cases depending on whether or not $k < d$. We first consider the case $k \leq d-1$. We showed in Lemma 7 that $(w_1, \cdots, w_k, a_1, \ldots, a_{i-1})$ is a valid list of length at least $d-1$ when $deg_{H_{old}}(a_1) = 1$. We now consider the scenario when L_{old} is pseudo-valid, and $deg_{H_{old}}(a_1) = 2$. The list L_{new} is still valid if $k \geq 2$, since the degree of a_1 in H_{new} is at most 3 and its position in L_{new} is also 3 or greater. So the non-trivial case is $k = 1$. In such a case $i = d-1$, as the only vertex w_1 belonging to W is connected to a_{d-1} in Algorithm 1. Also, $deg_{H_{old}}(a_{d-1}) = 2$, and a_{d-1} is connected to vertex w_1, so to ensure that $deg(a_{d-1}) = d$, in the for loop in step 9 of Algorithm 1, it is connected to only $d-3$ vertices, namely, $a_2, a_3, \ldots, a_{d-2}$. Since a_{d-1} is never connected to vertex a_1, $deg_{H_{new}}(a_1) = deg_{H_{old}}(a_1) = 2$. This shows that the sequence $(w_1, \cdots, w_k, a_1, \ldots, a_{i-1}) = (w_1, a_1, \ldots, a_{d-2})$ is a valid list of length exactly $d-1$. Truncating it to length $d-2$ again yields a valid sequence. In case $k \geq d$, a_1's degree does not play any role, so the argument from the proof of Lemma 7 works as is. ◀

▶ Remark 20. The condition $d \geq 3$ is necessary in Lemma 19 because in a pseudo-valid list all the vertices have degree 2. However, Procedure AddLayer works only in the case when the degree of each vertex in the list is at most $d-1$, which does not hold true for a pseudo-valid list when $d = 2$. So we provide a different analysis for the profile $(d^{d+1}, 2^k)$.

4.2 MaxNDeg$^-$ realization of the profile $\sigma = (d^{d+1}, 2^k)$

The following lemmas shows that $\sigma = (d^{d+1}, 2^1)$, for $d \geq 3$, is not MaxNDeg$^-$ realizable when $d \geq 3$; and $\sigma = (d^{d+1}, 2^k)$ is MaxNDeg$^-$ realizable when $d \geq 3$ and $k \geq 2$.

▶ **Lemma 21.** *For any integer $d \geq 3$, the profile $\sigma = (d^{d+1}, 2^1)$ is not MaxNDeg$^-$ realizable.*

Proof. Let us assume on the contrary that σ is MaxNDeg$^-$ realizable by a graph G, and let $w \in V(G)$ be a vertex such that MaxNDeg$^-(w) = 2$. The graph G must contain at least two vertices, say x and y, of degree d, since a single vertex of degree d can guarantee MaxNDeg$^-$ of d for at most d vertices in the graph. Consider the following two cases.

(i) $N[x] = N[y]$: In this case the MaxNDeg$^-$ of all the vertices in $N[x] = N[y]$ is at least $d \geq 3$, as they are adjacent to either x or y. Thus $w \notin N[x]$, which implies that $V(G) = N[x] \cup \{w\}$ since $|N[x]| = d+1$ and $|V(G)| = d+2$. Also, w cannot be adjacent to any vertex in $N[x]$, because if w is adjacent to a vertex $w_0 \in N[x]$, then $deg(w_0)$ must be 3, in contradiction to the assumption MaxNDeg$^-(w) = 2$. Thus the only possibility left is that w is a singleton vertex, which is again a contradiction.

(ii) $N[x] \neq N[y]$: In this case the vertex set of G is equal to $N[x] \cup N[y]$ since size of $N[x] \cup N[y]$ must be at least $d+2$ (as $|N[x] \cap N[y]| \leq d$) and is also at most $|V(G)| = d+2$. This implies that all the vertices of G are adjacent to either x or y, which contradicts the fact that MaxNDeg$^-(w) = 2$, since $deg(x) = deg(y) = d \geq 3$. ◀

▶ **Lemma 22.** *For any integers $d \geq 3$ and $k \geq 2$, the profile $\sigma = (d^{d+1}, 2^k)$ is* MaxNDeg$^-$ *realizable. Moreover, we can compute a connected realization in $O(d + k)$ time.*

Proof. The construction of G is as follows. Take a vertex u_1 and connect it to d other vertices $v_1, \ldots, v_d$. Next, take another vertex u_2 and connect it to vertices $v_2, \ldots, v_d$, and a new vertex v_{d+1}. Finally, take a path $(a_1, a_2, \ldots, a_\alpha)$ on $\alpha = k - 2$ new vertices, and connect a_1 to v_{d+1}. In the graph G, $deg(u_1) = deg(u_2) = d$, and $deg(v_i), deg(a_j) \leq 2$, for $i \in [1, d+1]$ and $j \in [1, k-2]$. Vertices u_1 and u_2 has maximum degree in their neighborhood 2, thus MaxNDeg$^-(u_1) =$ MaxNDeg$^-(u_2) = 2$. Each v_i is adjacent to u_1, u_2, for $i \in [1, d+1]$, so its MaxNDeg$^-$ is d. And, the MaxNDeg$^-$ of vertices on the path $(a_1, a_2, \ldots, a_\alpha)$ is 2, since they have a neighbor of degree 2. ◀

4.3 Algorithm

We now explain the construction of a graph realizing the profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1}) \neq (d_\ell^{d_\ell+1}, 2^1)$ that satisfies the conditions

- **(i)** $d_\ell \leq \min\{n_\ell, n-1\}$, and
- **(ii)** $d_1 \geq 2$,

where $n = n_1 + \cdots + n_\ell$. If σ is equal to $(d_\ell^{d_\ell+1}, 2^k)$, for some $k \geq 2$, we use Lemma 22 to realize σ. If not, then depending upon the value of n_ℓ, we initialize G differently as follows (refer to Algorithm 3 for the pseudocode).

1. If $n_\ell \geq d_\ell + 2$, we use Lemma 14 to initialize G to be a MaxNDeg$^-$ realization of the profile $(d_\ell^{n_\ell})$. Recall G contains an independent set, say $W = \{w_1, w_2, \ldots, w_{d_\ell}\}$, satisfying the condition that the degree of first two vertices is one, and the degree of the remaining vertices is at most two. We set $L_{\ell-1}$ to be the list $(w_1, w_2, \ldots, w_{d_{\ell-1}-1})$ (notice $d_{\ell-1} - 1 < d_\ell$). It is easy to verify that this list is valid.
2. If $n_\ell = d_\ell + 1$, then a realization of $(d_\ell^{d_\ell+1})$ does not contains a valid list. So we use Lemma 18 to initialize G to be a MaxNDeg$^-$ realization of the profile $(d_\ell^{d_\ell+1})$ that contains a pseudo-valid list. This is possible since we showed G contains an independent set, say $W = \{w_1, w_2, \ldots, w_{d_\ell-1}\}$, such that degree of each $w \in W$ is two. We set $L_{\ell-1}$ to be the list $(w_1, w_2, \ldots, w_{d_{\ell-1}-1})$ (again notice $d_{\ell-1} - 1 < d_\ell - 1$).
3. If $n_\ell = d_\ell$, then the sequence $d_\ell^{d_\ell}$ is not realizable (see Lemma 25). So we initialize G to be the graph realization of $(d_\ell^{n_\ell}, d_{\ell-1})$ as obtained from Lemma 17. We set $L_{\ell-1}$ be a valid list in G of size $d_{\ell-1} - 1$. Also we decrement $n_{\ell-1}$ by one as G already contain a vertex whose MaxNDeg$^-$ is $d_{\ell-1}$.

   Next for each $i = \ell - 1$ to 1 we perform following steps.

   - **(i)** We take as an input the valid list L_i of size $d_i - 1$, and execute Procedure AddLayer (G, L_i, n_i, d_i) to add n_i new vertices to G. The procedure returns a valid list L_{i-1} of size $d_i - 2$.
   - **(ii)** Truncate list L_{i-1} to contain only the first $d_{i-1} - 1 (\leq d_i - 2)$ vertices. The truncated list remains valid since it is a prefix of a valid list.

Correctness. Let $\bar{V}_\ell$ denote the set of vertices in graph G initialized in steps 5, 8, or 11 of Algorithm 3, and for $i \in [1, \ell - 1]$, let $\bar{V}_i$ denote the set of new vertices added to graph G in iteration i of for loop. For $i \in [1, \ell]$, let G_i be the graph induced by vertices $\bar{V}_i \cup \cdots \cup \bar{V}_\ell$.

Recall that if $n_\ell = d_\ell$, then the graph is initialized in step 11 and contains $n_\ell + 1$ vertices, of which one vertex, say z, has MaxNDeg$^-(z) = d_{\ell-1}$, and the remaining vertices have MaxNDeg$^- = d_\ell$. If $n_\ell = d_\ell$, then let $Z = \{z\}$, otherwise let $Z = \emptyset$. We set $V_\ell = \bar{V}_\ell \setminus Z$, $V_{\ell-1} = \bar{V}_{\ell-1} \cup Z$, and $V_i = \bar{V}_i$ for $i \in [1, \ell - 2]$. Thus $|V_i| = n_i$, for $i \in [1, \ell]$. The next lemma proves the correctness.

Algorithm 3 MaxNDeg$^-$ realization of $\sigma = (d_\ell^{n_\ell}, \ldots, d_1^{n_1})$.

Input: A sequence $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1}) \neq ({d_\ell}^{d_\ell+1} 2^1)$ satisfying $d_\ell \leq \min\{n-1, n_\ell\}$ and $d_1 \geq 2$.

1 **if** $\sigma = (d_\ell^{d_\ell+1}, 2^k)$ *for some* $k \geq 2$ **then**
2 Use Lemma 22 to compute a realization G for profile σ.
3 **else**
4 **case** $n_\ell \geq d_\ell + 2$ **do**
5 Initialize G to be the graph obtained from Lemma 14 that realizes the profile $(d_\ell^{n_\ell})$.
6 Set $L_{\ell-1}$ to be a valid list in G of size $d_{\ell-1} - 1$.
7 **case** $n_\ell = d_\ell + 1$ **do**
8 Initialize G to be the graph obtained from Lemma 17 that realizes the profile $(d_\ell^{d_\ell+1})$.
9 Set $L_{\ell-1}$ to be a pseudo-valid list in G of size $d_{\ell-1} - 1$.
10 **case** $n_\ell = d_\ell$ **do**
11 Initialize G to be the graph obtained from Lemma 18 that realizes the profile $(d_\ell^{d_\ell} d_{\ell-1})$.
12 Set $L_{\ell-1}$ to be a valid list in G of size $d_{\ell-1} - 1$.
13 Decrement $n_{\ell-1}$ by 1.
14 **for** $(i = \ell - 1$ *to* $1)$ **do**
15 $L_{i-1} \leftarrow$ AddLayer (G, L_i, n_i, d_i).
16 Truncate list L_{i-1} to contain only the first $d_{i-1} - 1 (\leq d_i - 2)$ vertices.
17 Output G.

▶ **Lemma 23.** *For any $i \in [1,\ell]$, graph G_i is a* MaxNDeg$^-$ *realization of profile $(d_\ell^{n_\ell}, \cdots, d_i^{n_i})$, except for the case $n_\ell = d_\ell$ in which G_ℓ is* MaxNDeg$^-$ *realization of profile $(d_\ell^{n_\ell}, d_{\ell-1})$. Moreover, for any $j \in [i, \ell]$, we have*

1. *For every $v \in V_j \setminus Z$, $deg_{G_i}(v) \leq$ MaxNDeg$^-{}_{G_i}(v) = d_j$.*
2. *If $n_\ell = d_\ell$, then $deg_{G_i}(z) = d_\ell$ and* MaxNDeg$^-{}_{G_i}(z) = d_{\ell-1}$.

Proof. We prove the claim by induction on the iterations of the for loop. The base case is for index ℓ, and the claim follows from Lemmas 14, 17, and 18. Specifically, notice that every vertex $v \in V_\ell$ that is included in G in step 5, 8, or 11 of the algorithm has MaxNDeg$^-(v) = d_\ell$. In the case $n_\ell = d_\ell$, the vertex $z \in V_{\ell-1}$ included in step 11 of algorithm has MaxNDeg$^-(z) = d_{\ell-1}$. Also, in both the cases, $V_\ell \cup Z$ is the vertex set of G, and degree of all the vertices in this set is bounded by d_ℓ.

For the inductive step, we assume that the claim holds for $i+1$, and prove the claim for i. Consider any vertex v in G_i. We have two cases.

1. $v \in V_i \setminus Z$: In this case by Proposition 15 and Lemma 19, $deg_{G_i}(v) \leq$ MaxNDeg$^-{}_{G_i}(v) = d_i$.
2. $v \in V_j \setminus Z$, for $j > i$: In this case we first show that for any vertex $w \in N(v)$, $deg_{G_i}(w) \leq d_j$. If $w \in V_i \setminus Z$, then we already showed $deg_{G_i}(w) \leq d_i$. So we next consider the case $w \in (V_{i+1} \cup \cdots \cup V_\ell) \setminus Z$. Now if $w \in L_i$ participates in Procedure AddLayer(G, L_i, n_i, d_i), then by Proposition 15 in the updated graph $deg_{G_i}(w) \leq d_i \leq d_j$. If $w \notin L_i$, then the degree of w is unaltered in the i^{th} iteration, and thus $deg_{G_i}(w) = deg_{G_{i+1}}(w) \leq d_j$ by the inductive

hypothesis. If $n_\ell = d_\ell$ and $w = z \in Z$, then also $deg_{G_i}(w) = deg_{G_{i+1}}(w)$ since vertex z never participates in procedure AddLayer. It follows that MaxNDeg$^-(v)$ remains unaltered due to iteration i, and thus MaxNDeg$^-{}_{G_i}(v) =$ MaxNDeg$^-{}_{G_{i+1}}(v) = d_j$.

Now when $n_\ell = d_\ell$, then $deg_{G_\ell}(z) = d_\ell$ and MaxNDeg$^-{}_{G_\ell}(z) = d_{\ell-1}$. The degree of vertex z never changes since it does not participates in procedure AddLayer. The MaxNDeg$^-$ of z never changes from the same reasoning as above. ◀

The execution time of algorithm takes $O\big(\sum_{i=1}^{\ell}(n_i + d_i)\big)$ time, which can be easily shown to be optimal. The following theorem is immediate from the above discussions.

▶ **Theorem 24.** *There exists an algorithm that given any profile $\sigma = (d_\ell^{n_\ell}, \ldots, d_1^{n_1}) \neq ({d_\ell}^{d_\ell+1}2^1)$ with $n = n_1 + \cdots + n_\ell$ satisfying $d_\ell \leq \min\{n-1, n_\ell\}$ and $d_1 \geq 2$, computes in optimal time a connected MaxNDeg$^-$ realization of σ.*

4.4 Complete characterization of MaxNDeg$^-$ profiles.

We first give the sufficient conditions for a profile to be MaxNDeg$^-$ realizable.

▶ **Lemma 25.** *A necessary condition for the profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ with $n = n_1 + \cdots + n_\ell$ to be MaxNDeg$^-$ realizable is $d_\ell \leq \min\{n_\ell, n-1\}$.*

Proof. Suppose σ is MaxNDeg$^-$ realizable by a graph H. Then there exists at least one vertex, say u, of degree exactly d_ℓ in H. Now $|N(u)| = d_\ell$ and $|N[u]| = d_\ell + 1$, which implies that the number of vertices in H whose MaxNDeg$^-$ is d_ℓ must be at least d_ℓ, so $n_\ell \geq d_\ell$. Also, the number of vertices in the graph H, n, must be at least $d_\ell + 1$. ◀

Consider a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1})$ realizable by a connected graph. If $d_1 = 1$, then the realizing graph must contain a vertex, say u, such that each vertex in $N(u)$ has degree 1. Let $d = deg(u)$, and $v_1, \ldots, v_d$ be the neighbors of u. Then $deg(v_1) = \cdots = deg(v_d) = 1$. So in this case the realizing graph is a star graph $K_{1,d}$ with MaxNDeg$^-$ profile $\sigma = (d^d, 1^1)$. If $d_1 \geq 2$, then by Lemma 25, for σ to be realizable in this case, we need that $d_\ell \leq \min\{n_\ell, n-1\}$. Also, Lemma 21 implies that σ must not be $(d^{d+1}, 2^1)$. By Theorem 24, under these conditions σ is always realizable. We thus have the following theorem.

▶ **Theorem 26.** *The necessary and sufficient condition for a profile $\sigma = (d_\ell^{n_\ell}, \cdots, d_1^{n_1}) \neq (d^{d+1}, 2^1)$ with $n = n_1 + \cdots + n_\ell$ to be MaxNDeg$^-$ realizable by a connected graph is*
- **(i)** *$d_\ell \leq \min\{n_\ell, n-1\}$ and $d_1 \geq 2$; or*
- **(ii)** *$\sigma = (d^d, 1^1)$ for some positive integer $d > 1$; or*
- **(iii)** *$\sigma = (1^2)$.*

For general graphs we have the following theorem.

▶ **Theorem 27.** *The necessary and sufficient condition for a profile σ to be MaxNDeg$^-$ realizable by a general graph is that σ can be split into two profiles σ_1 and σ_2 such that*
- **(i)** *σ_1 has a connected MaxNDeg$^-$ realization, and*
- **(ii)** *$\sigma_2 = (1^{2\alpha})$ or $\sigma_2 = (d^d, 1^{2\alpha+1})$ for some integers $d \geq 2, \alpha \geq 0$.*

Proof. Suppose σ is realizable by graph G. Let $\mathcal{C}(G)$ be a set consisting of all those components in G that contain a vertex of MaxNDeg$^-$ equal to 1 but is not an edge. As a long as $|\mathcal{C}(G)| > 1$, we perform following modifications to G. Take any two components $C_1, C_2 \in \mathcal{C}(G)$, and let σ_1 and σ_2 be their MaxNDeg$^-$ profiles. For $i = 1, 2$, component C_i must be of form K_{1,δ_i} and contain $\delta_i(\geq 2)$ vertices of MaxNDeg$^-$ equal to δ_i, and a single vertex of MaxNDeg$^-$ equal to 1. Let us assume $\delta_2 \geq \delta_1$. We replace C_1 and C_2 in G by two different components, namely, an edge and (i) a connected MaxNDeg$^-$ realization of profile $\delta_2^{\delta_1+\delta_2}$ if $\delta_2 = \delta_1$, or (ii) a connected MaxNDeg$^-$ realization of profile $(\delta_2^{\delta_2}, \delta_1^{\delta_1})$ if $\delta_2 > \delta_1$. In each iteration we decrease $|\mathcal{C}(G)|$ by a value two. In the end if $\mathcal{C}(G)|$ is non-empty we denote the only component in it by C_0. Next let $\bar{C}_1, \dots, \bar{C}_k$ be all those components in G that contain only the vertices of MaxNDeg$^-$ strictly greater than 1. Also let $\sigma_1, \dots, \sigma_k$ be their MaxNDeg$^-$ profiles. If $k > 0$, we replace the components $\bar{C}_1, \dots, \bar{C}_k$ by a single connected component, say $\bar{C}_0$, that realizes the profile $\sigma_1 + \cdots + \sigma_k$. It is easy to verify from Theorem 24 that $\sigma_1 + \cdots + \sigma_k$ will be MaxNDeg$^-$ realizable. The final graph G contains (i) at most one component, namely $\bar{C}_0$, having all vertices of MaxNDeg$^-$ greater than 1, (ii) at most one component, namely C_0, having exactly one vertex of MaxNDeg$^-$ equal to 1, and (iii) a union of some $\alpha \geq 0$ disjoint edges. This shows that σ can be split into two profiles σ_1 and σ_2 such that

(i) σ_1 has a connected MaxNDeg$^-$ realization, and
(ii) $\sigma_2 = (1^{2\alpha})$ or $\sigma_2 = (d^d, 1^{2\alpha+1})$ for some integers $d \geq 2, \alpha \geq 0$.

To prove the converse notice that $\sigma_2 = (1^{2\alpha})$ is realizable by a disjoint union of $\alpha \geq 0$ edges, and $\sigma_2 = (d^d, 1^{2\alpha+1})$ is realizable by a disjoint union of α edges and the star graph $K_{1,d}$. Thus any σ that can be split into two profiles σ_1 and σ_2 such that

(i) σ_1 has a connected MaxNDeg$^-$ realization, and
(ii) $\sigma_2 = (1^{2\alpha})$ or $\sigma_2 = (d^d, 1^{2\alpha+1})$ for some integers $d \geq 2, \alpha \geq 0$

is MaxNDeg$^-$ realizable. ◀

References

1 Martin Aigner and Eberhard Triesch. Realizability and uniqueness in graphs. *Discrete Mathematics*, 136:3–20, 1994.

2 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Realizability of graph specifications: Characterizations and algorithms. In *25th SIROCCO*, pages 3–13, 2018.

3 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Efficiently realizing interval sequences. In *30th Int. Symp. on Algorithms and Computation, ISAAC*, pages 47:1–47:15, 2019.

4 Amotz Bar-Noy, Keerti Choudhary, David Peleg, and Dror Rawitz. Graph profile realizations and applications to social networks. In *13th WALCOM*, pages 1–12, 2019.

5 Michael D. Barrus and Elizabeth A. Donovan. Neighborhood degree lists of graphs. *Discrete Mathematics*, 341(1):175–183, 2018.

6 Kevin E Bassler, Charo I Del Genio, Péter L Erdős, István Miklós, and Zoltán Toroczkai. Exact sampling of graphs with prescribed degree correlations. *New Journal of Physics*, 17(8):083052, August 2015.

7 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.

8 Sheshayya A. Choudum. A simple proof of the Erdös-Gallai theorem on graph sequences. *Bulletin of the Australian Mathematical Society*, 33(1):67–70, 1991.

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

10 Paul Erdös and Tibor Gallai. Graphs with prescribed degrees of vertices [hungarian]. *Matematikai Lapok*, 11:264–274, 1960.

11 C. Del Genio, H. Kim, Z. Toroczkai, and K.E. Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PLOS ONE*, 5:1–8, 2010.

12 S. Louis 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.

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

14 Ravi Kannan, Prasad Tetali, and Santosh S. Vempala. Simple markov-chain algorithms for generating bipartite graphs and tournaments. *Random Struct. & Algorithms*, 14(4):293–308, 1999.

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

16 H. Kim, Z. Toroczkai, P.L. Erdős, I. Miklós, and L.A. Székely. Degree-based graph construction. *J. Phys. A: Math. & Theor.*, 42:392001, 2009.

17 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.

18 Elchanan Mossel and Nathan Ross. Shotgun assembly of labeled graphs. *CoRR*, abs/1504.07682, 2015.

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

20 Gerard Sierksma and Han Hoogeveen. Seven criteria for integer sequences being graphic. *J. Graph Theory*, 15(2):223–231, 1991.

21 Amitabha Tripathi and Himanshu Tyagi. A simple criterion on degree sequences of graphs. *Discrete Applied Mathematics*, 156(18):3513–3517, 2008.

22 S.M. Ulam. *A Collection of Mathematical Problems*. Wiley, 1960.

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

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