

Randomized Algorithms to Generate Hypergraphs with Given Degree Sequences

Michela Ascolese^a Matthias Lienau^b Matthias Schulte^b
Anusch Taraz^b

Submitted: Jan 29, 2024; Accepted: Mar 12, 2025; Published: Apr 25, 2025

© The authors. Released under the CC BY-ND license (International 4.0).

Abstract

The question whether there exists a hypergraph whose degrees are equal to a given sequence of integers is a well-known reconstruction problem in graph theory, which is motivated by discrete tomography. In this paper we approach the problem by randomized algorithms which generate the required hypergraph with positive probability if the sequence satisfies certain constraints.

Mathematics Subject Classifications: 05C65, 05C80, 06R10

1 Introduction and Results

One of the central topics in discrete tomography is the reconstruction of a discrete object based on partial knowledge, such as its horizontal and vertical projections, see e.g. [25, 26]. This task can be rephrased in the context of graph theory as the problem of reconstructing a hypergraph starting from some information concerning its structure. For example, one can wonder whether there exists a hypergraph with a given degree sequence. In contrast to graphs, this problem is NP-hard for hypergraphs. In the current paper, we analyze randomized algorithms that provide hypergraphs with given degree sequences in certain situations.

Notation. We briefly introduce some notation that is needed to formulate the basic questions, related results and our contributions. We shall slightly deviate from the standard graph theoretic notions, by first allowing for the possibility that edges contain multiple copies of the same vertex and that the edge set contains multiple copies of identical edges. More precisely, a k -hypergraph is a pair $H = (V, E)$ where $V = [n] := \{1, \dots, n\}$ denotes the set of vertices and $E = \{e_1, \dots, e_m\}$ the multi-set of edges. Here every edge

^aDepartment of Mathematics and Computer Science, University of Florence, Firenze, Italy
(michela.ascolese@unifi.it).

^bInstitute of Mathematics, Hamburg University of Technology, Hamburg, Germany
(matthias.lienau@tuhh.de, matthias.schulte@tuhh.de, taraz@tuhh.de).

e_i is a multi-set of vertices of cardinality $|e_i| = k$. We define $|e|_i$ to be the number of occurrences of the vertex i in the edge e . The degree of a vertex, $\deg(i) = \sum_{j=1}^m |e_j|_i$, is the number of edges containing i (counted with its multiplicity).

An edge e is called a *loop* if there exists $i \in [n]$ such that $|e|_i > 1$. Two edges $e_i, e_j \in E$ with $e_i = e_j$ and $i \neq j$ are called *parallel edges*. The k -hypergraph H is called *simple* if it does not contain loops or parallel edges.

We consider integer sequences $\pi = (d_1, \dots, d_n)$ of length n such that $d_1 \geq d_2 \geq \dots \geq d_n \geq 1$, and we define $\sigma = \sum_{i=1}^n d_i$. Such a sequence π is called *k -graphic* if there exists a simple k -hypergraph H with $d_i = \deg(i)$ for all i . In this case we say that H has π as degree sequence.

Clearly, for a sequence π to be k -graphic we must have that $d_1 \leq \sigma/k$ and that k divides σ . Hence we shall always assume from now on that these properties hold for any integer sequence we consider.

The following two tasks are well-known problems in graph theory that are motivated by questions in discrete tomography. Given a number $k \in \mathbb{N}$ and a sequence $\pi = (d_1, \dots, d_n)$,

- decide whether π is k -graphic (decision problem),
- find a simple k -hypergraph H that has π as degree sequence (reconstruction problem).

Throughout this paper, we consider k to be constant and n to be large.

History. Just as with Satisfiability and Colorability, the borderline between tractability and non-tractability, intended as the possibility of finding a solution in polynomial time (w.r.t. the size of the input), runs between $k = 2$ and $k \geq 3$. In the case of graphs, both the decision and reconstruction problem can be solved in polynomial time. A non-recursive characterization of graphic degree sequences was given by Erdős and Gallai in [16]. Later, many equivalent conditions were provided (see [27]). Moreover, Hakimi [21] and Havel [22] showed that an intuitive greedy algorithm solves the reconstruction problem in polynomial time.

Moving to hypergraphs, in 1975 Dewdney [13] characterized k -graphic sequences, but unfortunately his characterization cannot be checked in polynomial time and does not yield a feasible reconstruction algorithm. The same is true for a characterization given by Billington using the notion of tableaux [6].

Several papers contributed necessary [6, 12] or sufficient [5, 8] conditions for an integer sequence to be k -graphic. In 2018, Deza et al. [14] proved that the decision problem is NP-complete for $k \geq 3$. This hardness result motivated research into subclasses of sequences for which a polynomial time solution can be given. Many of them were identified, and reconstruction algorithms mainly based on greedy techniques were provided (see [3, 4, 17, 18]).

Randomized approach. In this paper we investigate the use of randomized algorithms to generate hypergraphs with a given degree sequence. In combinatorics, the use of randomness to prove the existence of certain structures with prescribed properties is

usually called the *probabilistic method* and was pioneered by Erdős. Its underlying idea can roughly be described as follows: perform a suitable random experiment, show that with positive probability the outcome yields the desired structure, hence such an object must exist (see e.g. [1]).

The so-called *configuration model*, initially used for regular graphs, generates random graphs with a given degree sequence, see e.g. [7] as well as [23] and the references therein. In this model each vertex is equipped with so-called half-edges, where the number of these half-edges is equal to its desired degree. Then two half-edges are chosen uniformly at random and combined to create an edge until all half-edges are gone. In general, this procedure may yield loops or parallel edges, which are precisely the outcomes that we would like to avoid. Thus, one is interested in the probability that the obtained graph is simple. In [24], for example, a necessary and sufficient condition on the degree sequences is given that ensures that this probability does not converge to zero as the numbers of vertices and edges tend to infinity. For a non-asymptotic approximation of the probability we refer to [2].

One can generalize the configuration model to k -hypergraphs directly as done, for example, in [9, 10, 11]. Even though the half-edges are not truly half-edges anymore, but rather $1/k$ -edges for $k \neq 2$, we continue referring to them as half-edges. It seems intuitive that for larger k it becomes more unlikely to draw the exact same edge twice, so the probability to get parallel edges should be small. On the other hand, it becomes more likely to produce loops. The probability that two given half-edges v_1, v_2 of a vertex v are contained in the same edge is given by $(k-1)/(\sigma-1)$. Indeed, for the edge e containing the half-edge v_1 there are $\binom{\sigma-1}{k-1}$ possibilities to choose the remaining $k-1$ slots of e , $\binom{\sigma-2}{k-2}$ of which containing v_2 . By summing over all choices for pairs of half-edges adjacent to a single vertex and accounting for the fact that a single edge can contain at most $\binom{k}{2}$ pairs of equal vertices, we obtain the lower bound

$$\binom{k}{2}^{-1} \sum_{i=1}^n \frac{k-1}{\sigma-1} \binom{d_i}{2} = \frac{1}{k} \sum_{i=1}^n \frac{d_i(d_i-1)}{\sigma-1} \quad (1)$$

for the expected number of loops. This expression can tend to infinity with growing n , for example when $\min_{i \in [n]} d_i \rightarrow \infty$ as $n \rightarrow \infty$. The aim of the paper is to develop an approach that produces simple hypergraphs with positive probability in cases such as the previous example.

Our approach. We model the half-edges of the vertices $\{1, \dots, n\}$ as balls that are distributed and then drawn from a suitable number of boxes. Here is the rough idea:

1. Consider $k+1$ boxes with labels $1, \dots, k+1$, and for all $i \in [n]$ take d_i balls with label i , referring to the vertex i .
2. Distribute the balls among the boxes such that all the balls with the same label belong to the same box, and any box contains at most σ/k balls.

3. Consider the k boxes that contain the highest number of balls and, if there is a tie, take the boxes with the largest labels. Draw one ball from each of these boxes uniformly at random to construct an edge consisting of the labels (i.e. vertices) of the balls. Repeat until all boxes are empty.

Our goal is to show under some assumptions on the input data that the algorithm leads to the construction of a simple k -hypergraph with positive probability. We start with some remarks on the strategy.

The key idea of our algorithm is to prevent the emergence of loops and thus only having to deal with parallel edges, providing good results also when k is large. Indeed, in the second step we put the balls with the same label all into the same box, thus preventing the occurrence of loops. However, it is not obvious under which assumptions there exists an allocation of the balls to the boxes matching our requirement with at most σ/k balls in each box.

Note that it would of course be more intuitive to take only k boxes instead of $k+1$, but this would mean that every box needs to be filled with exactly σ/k balls, while still satisfying the constraints on putting all balls with the same label into the same box. This problem is called the *multi-way number partitioning problem* and is known to be NP-hard [19]. By taking $k+1$ boxes instead, we have some margin on the fill heights that allows us to find such an allocation, under mild assumptions.

Furthermore we remark that it is also not obvious that one can repeat the third step until all boxes are empty: it could be that we reach a stage where two boxes are empty, but there are still other non-empty boxes. It turns out that our assumptions on the degree sequence are sufficient to ensure that this will not happen.

Results. The following theorems state which assumptions guarantee that our general algorithmic approach of distributing balls into boxes will work. The pseudo-code of algorithms with the desired properties will be given in Section 2.

Theorem 1. *For $n \in \mathbb{N}$ and $k \geq 3$, let $\pi = (d_1, \dots, d_n)$ be a sequence such that $k(k+1)d_{k+2} \leq \sigma$. Then there is a polynomial time randomized algorithm that always returns a k -hypergraph H with degree sequence π and satisfies*

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{k+1}{2} \left(\frac{3k}{2}\right)^{k-2} \frac{d_1^k}{\sigma^{k-2}}.$$

In the setting of the previous theorem one obviously has

$$\mathbf{P}(H \text{ is simple}) \rightarrow 1 \tag{2}$$

as $n \rightarrow \infty$ if

$$d_1^k = o(\sigma^{k-2}). \tag{3}$$

This is the case when the degrees in the sequence π are either sufficiently small or sufficiently close to each other, as expressed in the following two corollaries.

Corollary 2. Let $k \geq 3$. If $d_1 \leq Cn^\alpha$ for $C > 0$ and $\alpha < 1 - \frac{2}{k}$, then (2) holds.

Corollary 3. Let $k \geq 3$ and define $\rho := d_1/d_n$. If

$$d_1^2 \left(\frac{\rho}{n} \right)^{k-2} \rightarrow 0,$$

then (2) holds.

Looking at the reduction for NP-hardness in [14], Corollary 2.1, it is clear that the decision problem remains NP-hard for all $k \geq 3$. On the other hand, using our Corollary 3 it is now clear that for example for $k = 15$ any sequence satisfying $n^{5/2} \leq d_n \leq d_1 \leq n^3$ is k -graphic if n is sufficiently large.

Obviously, the applicability of Theorem 1 depends heavily on the role of d_1 in π . Consider for example the sequence

$$\pi := \left(\underbrace{\frac{n}{\log(n)^3}, \dots, \frac{n}{\log(n)^3}}_{\log(n)}, \underbrace{\frac{\sqrt{n}}{\log(n)}, \dots, \frac{\sqrt{n}}{\log(n)}}_{n-\log(n)} \right), \quad (4)$$

after appropriate roundings to obtain integers. Here we have $\sigma \approx \frac{n^{3/2}}{\log(n)}$ and hence Condition (3) is not satisfied for $k = 4$. Therefore, we give another result which will allow us to ignore the first elements of the sequence π .

Theorem 4. For $k \geq 4$, $n \in \mathbb{N}$ and $\pi = (d_1, \dots, d_n)$, let $m \in [n]$ be maximal with

$$\frac{4\sigma}{k+1} \leq \sum_{i=m}^n d_i.$$

If $k(k+1)d_{k-2} \leq \sigma$ and $5k(k+1)d_m \leq 4\sigma$, then there is a polynomial time randomized algorithm that always returns a k -hypergraph H with degree sequence π and satisfies

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{3k(k+1)d_m^3}{4\sigma}.$$

Returning to our example sequence π in (4), we now have $d_m \approx \frac{\sqrt{n}}{\log(n)}$ and, again, $\sigma \approx \frac{n^{3/2}}{\log(n)}$, hence $d_m^3 = o(\sigma)$, thus proving that π is indeed 4-graphic for n large enough.

Related work. We briefly compare the above results to other activities in the area. Recently, Dyer et al. [15] tried to generate simple hypergraphs with given degree sequence uniformly at random using a bijection between bipartite graphs and k -hypergraphs, which requires less assumptions than the configuration model for hypergraphs. Their methods allow for scenarios where $d_1 = o(\min\{\sigma^{1/2}, \sigma^{1-2/k}\})$ (see Theorem 1.6 in [15]), while they require $d_1 = O(\log n)$ for the configuration model (see Lemma 2.3 in [15]). Our approach does not ask for uniform generation but works for scenarios up to $d_1 = o(\sigma^{1-2/k})$ (compare

Condition (3)), thus improving the previous result for any $k \geq 5$ (the same result is obtained for $k = 3, 4$). Moreover, our Theorem 4 allows us to ignore some vertices of higher degree.

Based on the characterization by Dewdney [13], in 2013 Behrens et al. gave sufficient conditions for a sequence to be k -graphic [5]. Among others they showed that a sequence is k -graphic if $d_1 = o(\sigma^{1-1/k})$ (or even if $d_1/\sigma^{1-1/k}$ is less than some constant, Corollary 2.2 in [5]). While this is a weaker constraint than our Condition (3), their result is non-constructive whereas our methods allow to generate a k -hypergraph with the given degree sequence in polynomial time.

It is also worth mentioning that so-called random constrained graph processes can be used to generate graphs and hypergraphs satisfying certain degree conditions. Here one starts with the empty hypergraph and adds edges chosen uniformly at random provided they do not violate some specified constraints. If, for instance, the constraint is such that no vertex is allowed to be contained in more than d edges, then it can be shown that, under some assumptions, the process generates a d -regular hypergraph with high probability (see [20]).

The remainder of this paper is organized as follows: in Section 2 we provide the implementation and analysis of our algorithms. Finally, in Section 3 we formulate and prove a general result (Theorem 8) from which we then deduce Theorems 1 and 4.

In principle our methods should also apply to situations involving non-uniform hypergraphs (i.e. hypergraphs whose edges may contain differently many vertices), but then the statements and computations will be less appealing.

2 Algorithms

We start with Step 2 of our approach sketched in the introduction, i.e. we need to distribute the balls representing half-edges among the $k + 1$ boxes. To this end we use the following algorithm, employing a greedy strategy.

Algorithm 1. *greedy_allocation*(π, ℓ)

Input: $\pi = (d_1, \dots, d_n)$ a non-increasing sequence of natural numbers, $\ell \in \mathbb{N}$

- 1 Set $B_1, \dots, B_\ell = \emptyset$;
- 2 **for** $i = 1, \dots, n$ **do**
- 3 Let $J \subseteq [\ell]$ be such that $|B_j|$ is minimal for all $j \in J$;
- 4 $j_{\min} = \min(J)$;
- 5 Add d_i copies of the vertex i to $B_{j_{\min}}$;

Output: (B_1, \dots, B_ℓ)

Without further assumptions on the integer sequence π , it is not clear that one can use Algorithm 1 to fill the boxes B_1, \dots, B_{k+1} in such a way that the demands in the second step of our approach are met: having all balls with the same label in a single box and no box exceeding σ/k balls. The following definition provides a set-theoretic description of these requirements.

Definition 5. For $n \in \mathbb{N}$ and $\pi = (d_1, \dots, d_n)$, we define $\text{Allocation}_{k+1}(\pi)$ as the set of all $(k+1)$ -tuples (B_1, \dots, B_{k+1}) of multi-sets B_1, \dots, B_{k+1} such that B_1, \dots, B_{k+1} are pairwise disjoint, each $i \in [n]$ is contained exactly d_i times in one of the multi-sets and $\sigma/k \geq |B_1| \geq \dots \geq |B_{k+1}|$.

To ensure that the output of Algorithm 1 for $\ell = k+1$, after ordering by size, belongs to $\text{Allocation}_{k+1}(\pi)$, we only need to control the cardinalities of the multi-sets B_1, \dots, B_{k+1} , as all other requirements are obviously satisfied. In Section 3, we will check this condition via the following bound.

Lemma 6. For $\pi = (d_1, \dots, d_n)$ and $\ell \in \mathbb{N}$, the algorithm `greedy_allocation` (π, ℓ) yields for all $i = 1, \dots, \ell$,

$$|B_i| \leq \max\left(d_1, \frac{\sigma}{\ell} + d_{\ell+1}\right).$$

Proof. If we were able to distribute all vertices equally, we would obtain σ/ℓ objects in each box. Once some box B_i contains more than σ/ℓ elements, there must be another box with fewer than σ/ℓ elements, so that we no longer put balls into B_i .

In the first ℓ steps, we fill the vertices $1, \dots, \ell$ into the boxes B_1, \dots, B_ℓ , respectively. Should d_1 exceed σ/ℓ , we obtain d_1 as an upper bound on $|B_1|, \dots, |B_\ell|$ after the first ℓ steps. All boxes that are still below σ/ℓ can now overshoot σ/ℓ by at most $d_{\ell+1}$, as this is the largest degree that is left. \square

We continue with the algorithm for the third step of our approach, Algorithm 2, which samples the edges of the hypergraph.

Algorithm 2. `sample_edges` $(\pi, B_1, \dots, B_{k+1})$

Input: $\pi = (d_1, \dots, d_n)$ a non-increasing sequence of natural numbers and $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$

```

1 Set  $E = \emptyset$ ;
2 for  $i = 1, \dots, \sigma/k$  do
3   Let  $J \subseteq [k+1]$  be such that  $|B_j|$  is minimal for all  $j \in J$ ;
4    $j_{\min} = \min(J)$ ;  $\text{edge} = \emptyset$ ;
5   for  $\ell = 1, \dots, k+1, \ell \neq j_{\min}$  do
6     choose  $b_\ell$  uniformly at random from  $B_\ell$ ;
7      $\text{edge} = \text{edge} \cup \{b_\ell\}$ ;
8      $B_\ell = B_\ell \setminus \{b_\ell\}$ ;
9    $E = E \cup \{\text{edge}\}$ ;
```

Output: E

In Theorem 8 we will gather some properties of `sample_edges` which are key ingredients for our proofs of Theorem 1 and Theorem 4.

Finally, we present two algorithms that take a degree sequence and sample a k -hypergraph by combining Algorithm 1 and Algorithm 2.

Algorithm 3. *sample_hypergraph*(π, k)

Input: $\pi = (d_1, \dots, d_n)$ a non-increasing sequence of natural numbers, $k \geq 3$

- 1 $(B_1, \dots, B_{k+1}) = \text{greedy_allocation}(\pi, k + 1)$;
- 2 Relabel B_1, \dots, B_{k+1} such that they are decreasing in size;
- 3 **if** $(B_1, \dots, B_{k+1}) \notin \text{Allocation}_{k+1}(\pi)$ **then**
- 4 | **return** (Error)
- 5 $E = \text{sample_edges}(B_1, \dots, B_{k+1})$;

Output: E

Algorithm 4. *sample_hypergraph_2*(π, k, m)

Input: $\pi = (d_1, \dots, d_n)$ a non-increasing sequence of natural numbers, $k \geq 4, m \in [n]$

- 1 $(B_1, \dots, B_4) = \text{greedy_allocation}((d_m, \dots, d_n), 4)$;
- 2 $(B_5, \dots, B_{k+1}) = \text{greedy_allocation}((d_1, \dots, d_{m-1}), k - 3)$;
- 3 Relabel B_1, \dots, B_{k+1} such that they are decreasing in size;
- 4 **if** $(B_1, \dots, B_{k+1}) \notin \text{Allocation}_{k+1}(\pi)$ **then**
- 5 | **return** (Error)
- 6 $E = \text{sample_edges}(B_1, \dots, B_{k+1})$;

Output: E

Algorithm 3, where we simply concatenate Algorithm 1 and Algorithm 2, is more obvious, and is used to obtain Theorem 1. Algorithm 4 is designed for the situation of many vertices with small degree. It seems plausible that having many small degrees simplifies the task of avoiding parallel edges. An investigation of Algorithm 4 yields Theorem 4.

Proposition 7. *Algorithm 3 and Algorithm 4 have computational costs of $O(kn + \sigma + \frac{\sigma^2}{k}) = O(\sigma^2)$.*

Proof. Algorithm 1 has computational costs of $O(\ell n + \sigma)$ because, for each vertex $i \in [n]$, we need to choose the box with the smallest number of balls and then add the d_i balls with label i to it. Thus, summing over i yields $O(\ell n)$ for choosing the boxes and $O(\sigma)$ for the total number of balls. Regarding Algorithm 2, the steps in lines 6-8 have a complexity of $O(\frac{\sigma}{k})$ as we have to update B_ℓ after drawing a ball. Altogether, we obtain computational costs of $O(k\frac{\sigma^2}{k^2}) = O(\frac{\sigma^2}{k})$. Combining these observations concludes the proof. \square

Observe that the above bound on the running time of lines 6-8 of Algorithm 2 can be improved by a more efficient implementation. Moreover, it is important to note that the computational costs depend on the choice of the (fixed) parameter k , but this does not affect the polynomiality of our strategy. In this paper, we are not striving to optimize the degrees of the polynomials in our runtime analysis.

3 Proofs

The following theorem investigates the output of the algorithm *sample_edges* (see Algorithm 2). Later on, we apply it to prove Theorem 1 and Theorem 4. For a multi-set A we

denote by $\text{Supp}(A) = \{a : a \in A\}$ the underlying set, which no longer takes into account the multiplicities in A .

Theorem 8. Consider $n \in \mathbb{N}$ and let $\pi = (d_1, \dots, d_n)$. For

$$(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$$

we have that

1. the algorithm `sample_edges`(π, B_1, \dots, B_{k+1}) terminates,
2. provides a k -hypergraph H without loops and with degree sequence π ,
3. and

$$\begin{aligned} & \mathbf{P}(H \text{ has no parallel edges}) \\ & \geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in [k+1] \setminus \{\ell\}} \frac{|B_j|(|B_j| - 1)}{2} \prod_{i \in [k+1] \setminus \{\ell\}} \frac{\max_{u \in \text{Supp}(B_i)} d_u}{|B_i|}. \end{aligned} \quad (5)$$

Proof. We start by showing the first claim, i.e. that the algorithm terminates. From $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$ we deduce that B_{k+1} is, in the beginning, among the boxes that contain the fewest elements, and it stays that way by construction (in case of a tie concerning the cardinalities $|B_1|, \dots, |B_{k+1}|$, `sample_edges` chooses the highest label, see Algorithm 2, line 4). We will show that, as soon as B_{k+1} runs empty, all other boxes contain precisely one ball each. Since $\sigma/k \geq |B_1| \geq \dots \geq |B_{k+1}|$, there exist $r_1, \dots, r_k \geq 0$ such that

$$|B_i| = \sigma/k - r_i \text{ for } i \in [k] \quad \text{and} \quad |B_{k+1}| = \sum_{i=1}^k r_i.$$

We compare $|B_{k+1}|$ to the number of balls missing to fill the first k boxes to the height of B_1 , the fullest one. This number is given by

$$D = \sum_{i=2}^k (|B_1| - |B_i|) = \sum_{i=2}^k (r_i - r_1) \leq \sum_{i=1}^k r_i = |B_{k+1}|.$$

We update D when drawing balls and investigate its changes. Whenever we draw a vertex from the last box, there are only two possible cases.

- *We do not draw a vertex from the first box.* In this case, the first box must be among the boxes that contain the fewest vertices. But since the first box always contains the most vertices (again, by the choice made in case of a tie), we must have already reached $D = 0$.

- We draw a vertex from the first box. In this case, the discrepancy between the first box and the one we do not draw from gets reduced by one, whereas all the others stay the same. Therefore, D gets reduced by one.

From the inequality $D \leq |B_{k+1}|$ above, we conclude that we reach a point where $D = 0$, i.e. the first k boxes have the same number of elements in them before B_{k+1} runs out of balls. When $D = 0$, one keeps drawing balls from the first k boxes until all $k + 1$ boxes contain the same number of elements. From here on out, the difference between the number of balls in the fullest and least full box can be at most one. Since k divides σ , there must be one ball in each of the first k boxes when the last box runs empty, which shows the first claim.

The second claim follows from $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, where the pairwise disjointness of B_1, \dots, B_{k+1} ensures the absence of loops.

It remains to show the inequality in the third claim. For $\ell \in [k + 1]$, let E_ℓ denote the list of all edges that do not contain a vertex from B_ℓ . The order of the edges in the list E_ℓ shall be the order of their creation in the algorithm. Since B_1, \dots, B_{k+1} are pairwise disjoint, it follows that two lists E_i and E_j cannot share an edge for $i \neq j$. Defining A_ℓ as the event that some edge occurs twice in E_ℓ for $\ell \in [k + 1]$, we deduce that

$$\mathbf{P}(H \text{ has no parallel edges}) \geq 1 - \sum_{\ell=1}^{k+1} \mathbf{P}(A_\ell). \quad (6)$$

We proceed by giving an upper bound on $\mathbf{P}(A_\ell)$ for a fixed $\ell \in [k + 1]$. It may be assumed that the boxes B_i with $i \in [k + 1] \setminus \{\ell\}$ contain at least two elements each, otherwise we would get $|E_\ell| \leq 1$ and thus $\mathbf{P}(A_\ell) = 0$. Denote the elements in E_ℓ by e_i , with $i = 1, \dots, |E_\ell|$, so that

$$\mathbf{P}(A_\ell) \leq \sum_{1 \leq i < j \leq |E_\ell|} \mathbf{P}(e_i = e_j). \quad (7)$$

To simplify notation, we write edges as vectors where we order the vertices according to the indices of B_1, \dots, B_{k+1} they belong to. Additionally, we assume that the elements of B_1, \dots, B_{k+1} are distinguishable even if they refer to the same vertex. Then, for distinct $i, j \in \{1, \dots, |E_\ell|\}$, the possible choices for (e_i, e_j) are of the form (f, g) given by

$$\begin{aligned} & ((f_1, \dots, f_{\ell-1}, f_{\ell+1}, \dots, f_{k+1}), (g_1, \dots, g_{\ell-1}, g_{\ell+1}, \dots, g_{k+1})) \\ & \in \left(\prod_{i \in [k+1] \setminus \{\ell\}} B_i \right)^2, \end{aligned}$$

where \times denotes the Cartesian product and $f_s \neq g_s$ for all $s \in [k + 1] \setminus \{\ell\}$, because we think of the elements as distinguishable. Since all random choices are with respect to uniform distributions, each possible combination (f, g) must have the same probability, so that

$$\mathbf{P}(e_i = f, e_j = g) = \prod_{s \in [k+1] \setminus \{\ell\}} \frac{1}{|B_s|(|B_s| - 1)}.$$

Now let us go back to indistinguishable objects in the boxes whenever they refer to the same vertex. Then we need to make up for the number of copies of a vertex, i.e. its degree, and obtain for a fixed possible edge $h \in \times_{i \in [k+1] \setminus \{\ell\}} \text{Supp}(B_i)$ that

$$\mathbf{P}(e_i = e_j = h) = \prod_{s \in [k+1] \setminus \{\ell\}} \frac{d_{h_s}(d_{h_s} - 1)}{|B_s|(|B_s| - 1)}.$$

Using the symmetry and summing over all possible choices for h , we obtain from (7) that

$$\begin{aligned} \mathbf{P}(A_\ell) &\leq \frac{|E_\ell|(|E_\ell| - 1)}{2} \mathbf{P}(e_1 = e_2) \\ &= \frac{|E_\ell|(|E_\ell| - 1)}{2} \prod_{i \in [k+1] \setminus \{\ell\}} \sum_{j \in \text{Supp}(B_i)} \frac{d_j(d_j - 1)}{|B_i|(|B_i| - 1)}. \end{aligned}$$

Next we observe that all edges in E_ℓ need to contain vertices from all B_i , with $i \in [k+1] \setminus \{\ell\}$. We obtain

$$|E_\ell| \leq \min_{j \in [k+1] \setminus \{\ell\}} |B_j|.$$

Moreover, it holds for all $i \in [k+1]$ that

$$\begin{aligned} \sum_{j \in \text{Supp}(B_i)} d_j(d_j - 1) &\leq \max_{u \in \text{Supp}(B_i)} d_u \sum_{j \in \text{Supp}(B_i)} (d_j - 1) \\ &\leq \max_{u \in \text{Supp}(B_i)} d_u(|B_i| - 1). \end{aligned}$$

Combining the three inequalities above with (6) yields (5). □

Proof of Theorem 1. We consider Algorithm 3, which has a polynomial runtime by Proposition 7. Once we have shown that $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, Theorem 8 yields that Algorithm 3 provides a k -hypergraph H with the desired degree sequence π as well as a bound on the probability of H being simple, which we simplify to obtain the bound from Theorem 1.

We start with proving $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$. The only property which is not clear by construction is that B_1 contains at most σ/k elements (after relabeling the boxes in the second line of Algorithm 3). From Lemma 6 with $\ell = k+1$ it follows that

$$|B_1| \leq \max \left(d_1, \frac{\sigma}{k+1} + d_{k+2} \right).$$

By our assumption on the input sequence, we know that $d_1 \leq \sigma/k$. On the other hand, using the assumed bound on d_{k+2} , we compute

$$\frac{\sigma}{k+1} + d_{k+2} \leq \frac{\sigma}{k+1} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k}.$$

This implies $|B_1| \leq \sigma/k$ and thus $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$.

It remains to show the lower bound on the probability of H being simple. By Theorem 8, the resulting k -hypergraph has no loops and the probability of having no parallel edges is bounded from below by

$$\begin{aligned} & \mathbf{P}(H \text{ has no parallel edges}) \\ & \geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in [k+1] \setminus \{\ell\}} \frac{|B_j|(|B_j| - 1)}{2} \prod_{i \in [k+1] \setminus \{\ell\}} \frac{\max_{u \in \text{Supp}(B_i)} d_u}{|B_i|}. \end{aligned}$$

From $|B_1| \geq \dots \geq |B_{k+1}|$ it follows for $\ell \in [k+1]$,

$$\min_{j \in [k+1] \setminus \{\ell\}} |B_j|(|B_j| - 1) \leq \begin{cases} |B_k|^2 & \text{for } \ell = k+1, \\ |B_{k+1}|^2 & \text{for } \ell \neq k+1, \end{cases}$$

as well as

$$\prod_{i \in [k+1] \setminus \{\ell\}} |B_i| \geq \begin{cases} |B_{k-1}|^{k-2} |B_k|^2 & \text{for } \ell = k+1, \\ |B_{k-1}|^{k-2} |B_{k+1}|^2 & \text{for } \ell \neq k+1. \end{cases}$$

Together with $d_1 \geq \dots \geq d_n$ this yields

$$\mathbf{P}(H \text{ has no parallel edges}) \geq 1 - \frac{k+1}{2} \frac{d_1^k}{|B_{k-1}|^{k-2}}.$$

We obtain a lower bound on $|B_{k-1}|$ by observing that the first $k-2$ boxes all contain at most σ/k elements each, so that there are at least $\sigma - (k-2)\sigma/k$ vertices left to distribute between B_{k-1}, B_k and B_{k+1} . Since B_{k-1} contains the most elements among these three, it holds that

$$|B_{k-1}| \geq \frac{1}{3} \left(\sigma - (k-2) \frac{\sigma}{k} \right) = \frac{2\sigma}{3k}. \tag{8}$$

Inserting this into the formula above yields

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \frac{k+1}{2} \left(\frac{3k}{2} \right)^{k-2} \frac{d_1^k}{\sigma^{k-2}}$$

and finishes the proof. □

Proof of Theorem 4. We show that Algorithm 4 has the required properties. By Proposition 7 it has a polynomial runtime. If we show $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$, Theorem 8 yields that Algorithm 4 produces a k -hypergraph with the desired degree sequence π and establishes a bound on the probability of H being simple, which we simplify to obtain the probability bound of Theorem 4. Concerning $(B_1, \dots, B_{k+1}) \in \text{Allocation}_{k+1}(\pi)$,

the non-trivial condition to check is $|B_1| \leq \sigma/k$ (after relabeling in the third line of Algorithm 4). Suppose that B_1 is generated in the first line of the code. Then Lemma 6 implies that

$$|B_1| \leq \max \left(d_m, \frac{\sum_{i=m}^n d_i}{4} + d_{m+4} \right).$$

By assumption we have $d_m \leq d_1 \leq \sigma/k$. Since m is maximal with the property

$$\sum_{i=m}^n d_i \geq \frac{4\sigma}{k+1},$$

we deduce that

$$\sum_{i=m+1}^n d_i < \frac{4\sigma}{k+1}.$$

As $d_{m+4} \leq d_m$, the assumed bound on d_m allows us to compute

$$\frac{\sum_{i=m}^n d_i}{4} + d_{m+4} < \frac{\sigma}{k+1} + \frac{5d_m}{4} \leq \frac{\sigma}{k+1} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k}.$$

Now suppose that B_1 is generated in the second line of the code of Algorithm 4. Then Lemma 6 provides

$$|B_1| \leq \max \left(d_1, \frac{\sum_{i=1}^{m-1} d_i}{k-3} + d_{k-2} \right).$$

We have $d_1 \leq \sigma/k$ whereas the definition of m , and the assumed bound on d_{k-2} gives us

$$\begin{aligned} \frac{\sum_{i=1}^{m-1} d_i}{k-3} + d_{k-2} &= \frac{\sigma - \sum_{i=m}^n d_i}{k-3} + d_{k-2} \leq \frac{\sigma - \frac{4\sigma}{k+1}}{k-3} + \frac{\sigma}{k(k+1)} \\ &= \frac{\sigma}{k+1} + \frac{\sigma}{k(k+1)} = \frac{\sigma}{k}. \end{aligned}$$

In both cases we obtain $|B_1| \leq \sigma/k$, which allows to apply Theorem 8. So, the algorithm terminates, and provides a k -hypergraph with the desired degree sequence. Moreover, the probability of the k -hypergraph being simple satisfies

$$\begin{aligned} &\mathbf{P}(H \text{ is simple}) \\ &\geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in [k+1] \setminus \{\ell\}} \frac{|B_j|(|B_j| - 1)}{2} \prod_{i \in [k+1] \setminus \{\ell\}} \frac{\max_{u \in \text{Supp}(B_i)} d_u}{|B_i|}. \end{aligned}$$

Let $a, b, c, d \in [k+1]$ be the indices of the boxes that were filled with the copies of the vertices m, \dots, n . For each $\ell \in [k+1]$, we can choose distinct $x_\ell, y_\ell, z_\ell \in \{a, b, c, d\} \setminus \{\ell\}$.

In particular, one has $x_\ell, y_\ell, z_\ell \in [k+1] \setminus \{\ell\}$ and the respective boxes contain copies of the vertices m, \dots, n . Thus, we can bound the numerators of the factors corresponding to x_ℓ, y_ℓ and z_ℓ in the rightmost product above by d_m . Bounding the $k-3$ remaining factors in the product by 1 yields

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \sum_{\ell=1}^{k+1} \min_{j \in [k+1] \setminus \{\ell\}} \frac{|B_j|(|B_j| - 1)}{2} \frac{d_m^3}{|B_{x_\ell}| |B_{y_\ell}| |B_{z_\ell}|}.$$

Since $x_\ell, y_\ell, z_\ell \neq \ell$, we deduce that $|B_{x_\ell}|, |B_{y_\ell}|$ and $|B_{z_\ell}|$ are all at least as large as $\min_{j \in [k+1] \setminus \{\ell\}} |B_j|$. As they are also pairwise distinct, their maximum is larger than or equal to $|B_{k-1}|$ as $|B_1| \geq \dots \geq |B_{k+1}|$. We obtain

$$\mathbf{P}(H \text{ is simple}) \geq 1 - \sum_{\ell=1}^{k+1} \frac{1}{2} \frac{d_m^3}{|B_{k-1}|} \geq 1 - \frac{3k(k+1)}{4} \frac{d_m^3}{\sigma},$$

where we inserted $3k|B_{k-1}| \geq 2\sigma$ from (8) above as lower bound for $|B_{k-1}|$ in the last inequality. Note that (8) does not depend on how the balls were allocated to the boxes and is also applicable here. This finishes the proof. \square

Acknowledgments

We thank an anonymous referee for the thorough reading and the valuable comments.

References

- [1] N. Alon and J. Spencer. The Probabilistic Method, 4th edition. *John Wiley & Sons, Hoboken, NJ*, 2016.
- [2] O. Angel, R. van der Hofstad, and C. Holmgren. Limit laws for self-loops and multiple edges in the configuration model. *Ann. Inst. H. Poincaré Probab. Statist.*, 55(3): 1509–1530, 2019.
- [3] M. Ascolese and A. Frosini. Characterization and reconstruction of hypergraphic pattern sequences. In *Combinatorial image analysis*, vol. 13348 of *Lecture Notes in Comput. Sci.*, pages 303–316, 2023.
- [4] M. Ascolese, A. Frosini, W. L. Kocay, and L. Tarsissi. Properties of unique degree sequences of 3-uniform hypergraphs. In *Discrete geometry and mathematical morphology*, vol. 12708 of *Lecture Notes in Comput. Sci.*, pages 312–324, 2021.
- [5] S. Behrens, C. Erbes, M. Ferrara, S. G. Hartke, B. Reiniger, H. Spinoza, and C. Tomlinson. New results on degree sequences of uniform hypergraphs. *Electron. J. Combin.*, 20(4), 2013.
- [6] D. Billington. Lattices and degree sequences of uniform hypergraphs. *Ars Combin.* 21A: 9–19, 1986.

- [7] B. Bollobés. Random Graphs, 2nd edition. *Cambridge Studies in Advanced Mathematics* 73. Cambridge University Press, Cambridge, 2001.
- [8] S. Brlek and A. Frosini. A tomographical interpretation of a sufficient condition on h -graphical sequences. In *Discrete geometry for computer imagery*, vol. 9647 of *Lecture Notes in Comput. Sci.*, pages 95–104, 2016.
- [9] P. S. Chodrow. Configuration models of random hypergraphs. *J. Complex Netw.* 8(3), cnaa018, 2020.
- [10] C. Cooper. The cores of random hypergraphs with a given degree sequence. *Random Structures Algorithms* 25(4): 353–375, 2004.
- [11] C. Cooper, A. Frieze, M. Molloy, and B. Reed. Perfect matchings in random r -regular, s -uniform hypergraphs. *Combin. Probab. Comput.* 5(1): 1–14, 1996.
- [12] S. Choudum. On graphic and 3-graphic sequences. *Discrete Math.* 87(1): 91–95, 1991.
- [13] A. K. Dewdney. Degree sequences in complexes and hypergraphs. *Proc. Amer. Math. Soc.* 53(2): 535–540, 1975.
- [14] A. Deza, A. Levin, S. M. Meesum, and S. Onn. Optimization over degree sequences. *SIAM J. Discrete Math.* 32(3): 2067–2079, 2018.
- [15] M. Dyer, C. Greenhill, P. Kleer, J. Ross, L. Stougie. Sampling hypergraphs with given degrees. *Discrete Math.* 344(11) vol. 112566, 2021.
- [16] P. Erdős, T. Gallai. Graphs with prescribed degrees of vertices (in Hungarian). *Mat. Lapok* 11: 264–274, 1960.
- [17] A. Frosini, G. Palma, and S. Rinaldi. Combinatorial properties of degree sequences of 3-uniform hypergraphs arising from Saind arrays. In *Beyond the horizon of computability*, vol. 12098 of *Lecture Notes in Comput. Sci.*, pages 228–238, 2020.
- [18] A. Frosini, C. Picouleau, and S. Rinaldi. New sufficient conditions on the degree sequences of uniform hypergraphs. *Theoretical Computer Science* 868: 97–111, 2021.
- [19] M. R. Garey, and D. S. Johnson. Computers and intractability: A Guide to the Theory of NP-Completeness. *Freeman*, New York, 1979.
- [20] C. Greenhill, A. Ruciński, and N. Wormald. Random hypergraph processes with degree restrictions. *Graphs Combinatorics* 20: 319–332, 2004.
- [21] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph I. *J. Soc. Indust. Appl. Math.* 10: 496–506, 1962.
- [22] V. Havel. A remark on the existence of finite graphs (in Czech). *Časopis Pěst. Mat.* 80: 477–480, 1955.
- [23] R. van der Hofstad. Random Graphs and Complex Networks Vol 1. *Cambridge University Press*, Cambridge, 2017.
- [24] S. Janson. The probability that a random multigraph is simple. *Combin. Probab. Comput.* 18(1-2): 205–225, 2009.
- [25] A. Kuba and G. T. Herman (editors). Discrete tomography. *Birkhäuser Boston, Inc.*, Boston, MA, 1999.

- [26] A. Kuba and G. T. Herman (editors). Advances in discrete tomography and its applications. *Birkhäuser Boston, Inc.*, Boston, MA, 2007.
- [27] G. Sierksma and H. Hoogeveen. Seven criteria for integer sequences being graphic. *J. Graph Theory* 15(2): 223–231, 1991.