arXiv:1202.1111v2 [cs.DS] 7 Feb 2012

Thresholds for Extreme Orientability

Po-Shen Loh* Rasmus Pagh†

Abstract

Multiple-choice load balancing has been a topic of intense study since the seminal paper of Azar, Broder, Karlin, and Upfal. Questions in this area can be phrased in terms of *orientations* of a graph, or more generally a k-uniform random hypergraph. A (d, b)-*orientation* is an assignment of each edge to d of its vertices, such that no vertex has more than b edges assigned to it. In this paper we study the existence of $(k-1, 1)$-orientations, and establish:

- The density threshold below which an orientation exists with high probability, and above which it does not exist with high probability.
- An algorithm for finding an orientation that runs in linear time with high probability, with explicit polynomial bounds on the failure probability.

Previously, no closed-form expression for the threshold was known. The only known algorithms for constructing $(k-1, 1)$-orientations worked for $k \leq 3$, and were only shown to have *expected* linear running time.

Key words. Multiple-choice hashing, random hypergraphs, orientations.

1 Introduction

The efficiency of many algorithms and data structures rests on the fact that randomly and independently throwing m balls into n bins ensures a distribution that is, with high probability, close to uniform. Since the seminal paper of Azar et al. [3] a large literature has grown around even stronger *multiple-choice* load balancing schemes where the location of each ball is selected within a random *set* of $k > 1$ bins.

These problems have been studied both in the *on-line* setting, where balls and their possible locations are revealed one by one, and in the *off-line* setting where we are interested in the best allocation of a given set of balls. Most often, the focus of multiple-choice schemes is on minimizing the maximum number of balls contained in any bin. The question can also be turned around to ask for the largest number of balls that can be placed such that there are at most b balls in each bin. Of course, this number depends on the random choices made, but in the off-line setting it turns out that there is a well-defined *threshold* $m = (1 \pm o(1))\alpha n$, below which it is highly likely that the

*Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213. Email: ploh@cmu.edu.
Research supported by an NSA Young Investigators Grant and a USA-Israel BSF Grant.
†IT University of Copenhagen, Copenhagen, Denmark. Email: pagh@itu.dk.

allocation is possible, and above which it is highly unlikely that the allocation is possible. Here, α is a constant that depends on k and b, but not on n.

In this paper we consider the scenario where each ball comes in d *copies*, and must be placed in exactly d (distinct) out of k possible bins. Observe that the case $d = k$ is not so interesting, because it is equivalent to the single-choice case with md balls. Thus the interesting extreme case is $d = k - 1$, which is the focus of this paper. Motivation for copying each ball comes from parallel and distributed systems where we want high redundancy (resistance to $d-1$ failures), and/or want to ensure that any set of balls can be accessed in parallel with only a single request per bin. Early papers investigating such schemes include [6, 22, 20]. As a more recent example, Amossen and Pagh [2] considered the case $k = 3$, $d = 2$, $b = 1$, and showed that up to $\frac{(1-\varepsilon)}{6}m$ balls can be placed with high probability,[1] for any constant $\varepsilon > 0$. This was used to construct a data structure for sets that allows very fast computation of set intersections on graphics hardware. In this paper we show that the constant 6 in [2] cannot be reduced, i.e., that $m = (1 \pm o(1))n/6$ is the threshold for the problem of allocating balls into 2 of 3 bins with maximum load 1. In fact, we generalize this result to the extreme case $d = k - 1$, $b = 1$ for any $k > 2$, giving explicit bounds on the probability of successful allocation in terms of m. We also present a generalization of the algorithms of [2, 18] that computes a $(k-1, 1)$-orientation (if one exists) of a given k-regular hypergraph, and show that it runs in linear time with high probability. This strengthens [2] which only shows linear running time in expectation.

1.1 Related work

Multiple-choice balls and bins scenarios can be modeled as a random k-regular hypergraph with m edges (balls) on n vertices (bins), where edges are chosen i.i.d. uniformly from the set of all k-sets of vertices. Let $H_{n,m;k}$ be the random k-uniform hypergraph with n vertices and m hyperedges, where each such object is taken with equal probability. In the regime of interest in this work, when m is linear in n, there is essentially no difference between allowing and disallowing multiple edges, because for $k \geq 3$, the probability that the multi-hypergraph analogue repeats an edge is only $O(n^{-1})$. Given such a hypergraph, a (d, b)-*orientation* is an assignment of each edge to d of its vertices, such that no vertex has more than b edges assigned to it.

On-line setting. In our description of the on-line setting, we restrict attention to the case where balls cannot be moved, once placed into bins. Azar et al. [3] considered $(1, b)$-orientations in the on-line setting, and showed that the greedy algorithm that always assigns a ball to its least loaded bin achieves a $(1, O(m/n + \log\log m/\log k))$-orientation. Tighter bounds for the maximum load of $(1, b)$-orientations were later obtained by Berenbrink et al. [4].

Off-line setting. In the off-line setting, the threshold for $(1, 1)$-orientations with $k = 2$ can be shown (see, e.g. [18]) to coincide with the appearance of a *giant component* in the random graph, which is known to happen at $m = (1 \pm o(1))n/2$ with high probability [8]. Several groups of researchers [7, 11, 12] independently established the thresholds for $(1, 1)$-orientations for every $k > 2$. Generalizing in another direction, Fernholz and Ramachandran [9] and Cain, Sanders, and

[1] Meaning probability tending to 1 as $n \to \infty$.

Wormald [5] showed thresholds for $(1, b)$-orientations for $k = 2$, and gave expected linear time algorithms for computing an orientation. This result was later extended to $k > 2$ by Fountoulakis et al. [10].

Gao and Wormald [13] established thresholds for (d, b)-orientations, given that b is a sufficiently large constant (depending on d and k). Independently of our work, Lelarge [16] recently announced thresholds for any combination of parameters k, d, and b, and introduced technical machinery which enabled the analysis of all regimes. The thresholds are not explicit, however, and in general depend on the solutions to certain sets of transcendental equations. (This phenomenon also arises in the graph case, which had previously been studied by a number of authors.) If one plugs in the parameters corresponding to our situation of extreme orientability, then one can recover our threshold observation, but our alternative direct approach proves to be quite amenable to algorithmic analysis.

1.2 Our contribution

The main contribution of this paper is to highlight two links between Probabilistic Combinatorics and $(k-1, 1)$-orientations. First, we observe the connection between the literature on the phase transition in random hypergraphs and $(k-1, 1)$-orientations, which provides a natural explanation for the threshold phenomenon experimentally documented in [2]. Second, we derive explicit, quantitative high-probability bounds for the subcritical running time, by tracking a key parameter known as "susceptibility," through the Differential Equations method for analyzing discrete random processes. Previous bounds were only of expected-time type. Also, since we seek good polynomial-type dependencies in our probability bounds, we perform a more careful analysis of the susceptibility growth, which is substantially sharper than in previous published work (e.g., [21]) which was satisfied with error bounds that could tend to zero very slowly. Our main theorem refers to the pseudocode of the ORIENT algorithm, which can be found in section 3.1. Its running time is determined by the number of iterations, which we define to be the number of times the condition in the **while** loop is evaluated.

Theorem 1.1. *Let $0 < \epsilon < \frac{1}{2}$ be given, and assume that $\frac{n}{\log^6 n} > \frac{40000k^6}{\epsilon^{12}}$. Let $m = (1-\epsilon)\frac{n}{k(k-1)}$. With probability at least $1 - 3n^{-1}$, all edges of the random k-uniform hypergraph $H_{n,m;k}$ can be $(k-1, 1)$-oriented by the* ORIENT *procedure using a total of at most*

$$2k^2 \left(\frac{1}{\epsilon} + \frac{200k^3 \log^3 n}{\epsilon^7 \sqrt{n}} \right) \cdot n \, .$$

iterations, each taking constant time.

This paper is organized as follows. The next section observes the natural threshold for extreme orientability. Then, Section 3 applies the Differential Equations method to deduce quantitative high-probability bounds for algorithmic performance in the feasible regime. The following (standard) asymptotic notation will be utilized extensively. For two functions $f(n)$ and $g(n)$, we write $f(n) = o(g(n))$ or $g(n) = \omega(f(n))$ if $\lim_{n\to\infty} f(n)/g(n) = 0$, and $f(n) = O(g(n))$ or $g(n) = \Omega(f(n))$ if there exists a constant M such that $|f(n)| \leq M|g(n)|$ for all sufficiently large n.

2 Non-orientability

We now investigate why there is no $(k-1,k)$-orientation when the number of edges exceeds $\frac{n}{k(k-1)}$. This is done by exhibiting an obstruction that appears asymptotically almost surely as n approaches infinity. One may observe many types of possible obstructions to orientability. A simple example for $k > 3$ is the k-uniform hypergraph consisting of two hyperedges overlapping in three vertices. It is clearly impossible to pick $k-1$ vertices for each hyperedge, as there are only $2k-3$ vertices to share. Unfortunately, any fixed-size obstruction has a threshold for appearance in $H_{n,m;k}$ that is far beyond $\frac{n}{k(k-1)}$, so one cannot simply pinpoint a single such hypergraph as the culprit for non-orientability.

Instead, we draw inspiration from the case $k = 2$ (often referred to as "cuckoo hashing" [18]) where the desired threshold $\frac{n}{2}$ matches the appearance of the well-studied *giant component*. Indeed, the seminal result of Erdős and Rényi [8] established that in the uniformly random graph with cn edges, for constants $c < \frac{1}{2}$, the largest connected component has size $O(\log n)$, whereas for constants $c > \frac{1}{2}$, the largest connected component has size $\Omega(n)$. Further study (see, e.g., the book [14]) revealed that for $c < \frac{1}{2}$, all connected components are either trees or unicyclic (containing at most one cycle), whereas for $c > \frac{1}{2}$, the giant component is multicyclic. As any multicyclic component would have too many edges for vertices to be $(k-1,1)$-orientable, this would establish the result for $k = 2$.

The remainder of this section translates the random graph literature into the orientability context, to observe the threshold for $k \geq 3$. First, it is convenient to introduce a measure of how "crowded" a component is.

Definition 2.1. *Let $k \geq 3$ be a fixed integer, and let H be a k-uniform hypergraph. The **excess** of H is the difference $(k-1)e(H) - v(H)$, where $v(H)$ and $e(H)$ denote the numbers of vertices and edges in H, respectively.*

For connected hypergraphs H, the excess is always an integer greater than or equal to -1. When it is -1, the hypergraph is acyclic, and called a *hypertree*. When the excess is 0, we say that H is *unicyclic*, and when the excess is positive, we say that H is *complex*. Note that in the context of $(k-1,1)$-orientability, any complex component is an obstruction. Given an edge set E' we define its *capacity* as $\text{cap}(E') = \sum_{v \in V} \min(b, |\{e \in E' : v \in e\}|)$. We have the following consequence of the max-flow min-cut theorem (see, e.g. [19, Section 6.1]):

Theorem 2.2. *A k-regular hypergraph (V,E) has a (d,b)-orientation if and only if each subset $E' \subseteq E$ has capacity $cap(E') \geq |E'|d$.*

Proof. The capacity sums, over each vertex, an upper bound on how many edges in E' can be oriented towards it. If some edge set E' has capacity less than $|E'|d$ it is thus impossible to orient all its edges (even ignoring edges outside of E'). For the reverse direction consider the flow network with:

- Node set $E \cup V \cup \{s,t\}$, i.e., a node per edge and vertex in (V,E), plus a source node s, and a sink node t.
- Capacity 1 edges connecting the node of each $e \in E$ to the k nodes in V contained in e.

- Capacity d edges from s to each vertex in E, and capacity b edges from vertex in V to t.

Observe that an integer s-t flow corresponds to an orientation of edges with a flow of 1 from an edge to each vertex that the edge is oriented towards. This means that if there is no (d, b)-orientation, there is no integer s-t flow of value $|E|d$. Since all capacities in the network are integral, this in turn means that there exists no flow of value $|E|d$ at all. Using the max-flow min-cut theorem this implies that there is a minimum s-t cut (S, T) such that the total capacity of edges from S to T is $\text{cut}(S, T) < |E|d$. Let E' denote the set of edges that are members of S. Since (S, T) is minimal vertices in $V \cap S$ appear in at least b edges in E', and vertices in $V \cap T$ appear in at most b edges of E'. Thus we obtain:

$$\text{cap}(E') = \sum_{v \in V \cap S} b + \sum_{v \in V \cap T} |\{e \in E' : v \in e\}|) = \text{cut}(S, T) - |E \backslash E'|d < |E'|d \ .$$

□

For $b = 1$ the capacity of a set E' is exactly the number of distinct vertices in its edges, so the capacity of E' is $(k-1)|E'|$ minus the excess of E'. This means that $(k-1, 1)$-orientability exactly coincides with the appearance of a complex component. Much is known about the phase transition in random hypergraphs. The following results are from the paper [15] of Karoński and Łuczak, which actually determines several results of much higher precision.

Theorem 2.3. *(Theorem 4 in [15].) Let $k \geq 3$ be a fixed integer, and let $m = \frac{n}{k(k-1)} - t(n)$, where $t(n)$ is any function of higher order than $n^{2/3}$, i.e., $t(n) = \omega(n^{2/3})$. Then $H_{n,m;k}$ consists of hypertrees and unicyclic components with high probability (as n grows).*

Remark. The previous argument establishes that hypertrees and unicyclic components can be $(k-1, 1)$-oriented, although the running time for computing the orientation may increase with the component size. The earlier result of the second author established that in expectation, this could be done efficiently for $m = (1 - \epsilon)\frac{n}{k(k-1)}$ in the case $k = 3$. The connection above complements the earlier result by establishing feasibility, although not necessarily efficiency, when the number of edges differs from $\frac{n}{k(k-1)}$ by a sublinear term.

Theorem 2.4. *(Theorem 10 in [15].) Let $k \geq 3$ be a fixed integer, and let $m = \frac{n}{k(k-1)} + t(n)$, where $t(n)$ is any function of higher order than $n^{2/3}$ but smaller order than $n^{2/3}\big(\frac{\log n}{\log \log n}\big)^{1/3}$. Then with high probability, $H_{n,m;k}$ consists of one large complex component and some number of small components which are either hypertrees or unicyclic.*

Remark. Clearly, adding more edges only creates more complex components, so the upper bound on $t(n)$ plays a role only in limiting the number of large complex components.

Therefore, as soon as we exceed $\frac{n}{k(k-1)}$ by even a sublinear deviation, an obstruction appears, and hence $(k-1, 1)$-orientability fails. Note that we cannot bound the size of the complex component, and in fact its size grows with n. There remains a window of width roughly $n^{2/3}$ between the lower and upper bounds. It is worth noting that for the case of graphs, this is also well-understood, and when $m = \frac{n}{k(k-1)} + cn^{2/3}$ for (positive or negative) constants c, there is a constant probability of having a complex component. See, e.g., the discussion in the book [1].

3 High-probability running time bound

In this section we present and analyze a simple algorithm for finding $(k-1,k)$-orientations. The first step is to observe that the running time to orient each new edge is $O(k^2 s)$, where s is the size of the connected component formed by the new edge.

3.1 Algorithm description

The algorithm works by iteratively extending an orientation to more and more edges. We assume that vertices of each edge e can be traversed using methods e.first() (which returns an arbitrary vertex) and e.next(v) (which gives the next node in the order after v, cycling back to e.first() when all vertices have been traversed). For $v \in V$ let $T[v]$ refer to the edge that is oriented towards v, where $T[v] = \perp$ if no edge is oriented towards v. We maintain an array indexed by V that initially has all entries set to $\perp$. An edge e is directed to $k-1$ vertices by calling the following procedure, generalizing the procedure of [2]. We use the notation $\leftrightarrow$ to indicate exchange of two variable values.

```
procedure ORIENT(e)
  for i := 1 to k − 1 do
    τ = e
    v = e.first()
    while τ ≠ ⊥
      v = τ.next(v)
      τ ↔ T[v]
    end while
  end for
```

When ORIENT is called, each member of the set of previously oriented edges E_1 appears $k-1$ times in T. The procedure runs a while loop $k-1$ times that (if it terminates) inserts e in $T[v]$ for some $v \in e$, while ensuring that each edge $e' \in E_1$ is still oriented towards $k-1$ positions in T. The invariant of the while loop is that all edges in E_1 are oriented towards $k-1$ vertices, and e is oriented towards i vertices, with one exception: If $\tau \neq \perp$ the edge τ which is oriented towards one vertex less. Clearly, once $i = k-1$ and $\tau = \perp$ we have oriented all edges in $E_1 \cup \{e\}$.

We claim that the procedure always terminates if an orientation exists, and more specifically that the time spent if e is in a component of size s is $O(k^2 s)$. (Some stopping criterion is needed for termination in case no orientation exists, but this is left out for simplicity.) Suppose the while loop does not stop, i.e., it goes through an infinite sequence of edges. Let $e_1, e_2, e_3, \ldots$ denote this edge sequence, with consecutive identical edges combined into a single occurrence. We observe that there can be at most $k-1$ consecutive iterations involving a particular edge. Notice also that edge e_i shares at least one vertex with edge e_{i+1} for each i. Consider a minimal subsequence $e_i, \ldots, e_j$ containing 3 such occurrences of some edge, and without loss of generality, assume that $e = 1$. Let ℓ_1 and ℓ_2, $1 \leq \ell_1 < \ell_2 < j$, be the indexes of the edge in this subsequence that first appears for the second time (so ℓ_2 is minimal). Since all previously fully-oriented edges already are oriented towards all but one of their k vertices, one observes that then $e_{\ell_2+t} = e_{\ell_1-t}$ for $t = 0, \ldots, \ell_1 - 1$. This means that the $\ell_2 - 1$ distinct edges $e_1, \ldots, e_{\ell_2-1}$ contain exactly $(\ell_2 - 1)(k-1)$ distinct

vertices. From e_{ℓ_2} to $e_{\ell_2+\ell_1-1} = e_1$, the edges encountered are all repeats (in reverse order) of those already seen. After $e_{\ell_2+\ell_1-1}$ (which is equal to e_1) each new edge introduces at most $k-1$ new vertices until we reach an edge that overlaps with a previously visited edge and only $k-2$ vertices are introduced. At that point we have visited a set of edges having less than $k-1$ available vertices on average, meaning that no $(k-1,k)$-orientation exists.

Therefore, the length of the edge sequence is at most $2s$, while the number of consecutive identical edges consolidated into each element is at most $k-1$. Since the while loop is run $k-1$ times for each new edge to orient, we conclude that the full orientation of the edge completes in $O(k^2 s)$ time.

3.2 Probabilistic tools

We will need the following version of the Chernoff bound (see e.g. [17]):

Theorem 3.1. *For any $0 < \epsilon < 1$, every binomial random variable X with mean μ satisfies*

$$\mathbb{P}\left[X < (1-\epsilon)\mu\right] < e^{-\frac{\epsilon^2}{2}\mu} \qquad \text{and} \qquad \mathbb{P}\left[X > (1+\epsilon)\mu\right] < e^{-\frac{\epsilon^2}{3}\mu}.$$

A *supermartingale* is a sequence $X_0, X_1, \ldots$ of random variables such that each conditional expectation $\mathbb{E}\left[X_{t+1} \mid X_0, \ldots, X_t\right]$ is at most X_t. Azuma's inequality (see e.g. [17]) states the following:

Theorem 3.2. *Let $X_0, \ldots, X_n$ be a supermartingale, with bounded differences $|X_{t+1} - X_t| \le C$. Then for any $\lambda \ge 0$,*

$$\mathbb{P}\left[X_n \ge X_0 + \lambda\right] \;\le\; \exp\left\{-\frac{\lambda^2}{2C^2 n}\right\}.$$

3.3 Analysis of random hypergraphs

Throughout, we impose explicit bounds that keep n "sufficiently large" in order to simplify our calculations. Recall that $H_{n,m;k}$ is the random k-uniform hypergraph obtained by uniformly sampling one with n-vertices and m hyperedges. In this section, it will be substantially more convenient for us to work with a process that exhibits more independence. Specifically, we consider instead the following sequential process, which fortunately is quite similar to the original $H_{n,m;k}$.

Lemma 3.3. *Let $n > k \ge 2$, with $n > 2000$. Consider the random hypergraph process $H_0, H_1, \ldots$, where H_0 is the empty hypergraph with n isolated vertices. At each time t, sample k vertices independently and uniformly at random. If they are distinct, and form a hyperedge which does not yet appear in H_t, then add it to form H_{t+1}. Otherwise, let $H_{t+1} = H_t$. Then, with probability at least $1 - n^{-1}$, in the first $\frac{n}{k(k-1)}$ rounds, the number of times that we do not add is at most $\log n$.*

Proof. At time $t+1$, a union bound shows that the probability that the k sampled vertices are not distinct is at most

$$\frac{1}{n} + \frac{2}{n} + \cdots \frac{k-1}{n} = \frac{k(k-1)}{2n}.$$

Also, since $t < \frac{n}{k(k-1)}$, the probability that the k sampled vertices form a previously-added hyperedge is

$$\frac{tk!}{n^k} < \frac{(k-2)!}{n^{k-1}} < \frac{1}{n},$$

where we used $n > k$ for the final bound. Thus the probability that $H_{t+1} = H_t$ is at most $\frac{k(k-1)}{n}$, and so the probability that this happens at least $s = \log n$ times in the first $\frac{n}{k(k-1)}$ rounds is at most

$$\binom{\frac{n}{k(k-1)}}{s}\left(\frac{k(k-1)}{n}\right)^s \leq \left(\frac{e\cdot\frac{n}{k(k-1)}}{s}\right)^s\left(\frac{k(k-1)}{n}\right)^s = \left(\frac{e}{s}\right)^s = \left(\frac{e}{\log n}\right)^{\log n},$$

which is below n^{-1} for all $n > e^{e^2}$. □

It is sometimes more convenient to work with the related model $H_{n,p;k}$, which is the random k-uniform hypergraph formed by taking each of the $\binom{n}{k}$ potential hyperedges independently with probability p. Fortunately, the behavior of $H_{n,p;k}$ closely approximates that of $H_{n,m;k}$.

Lemma 3.4. *Assume that $0 < \epsilon < \frac{1}{2}$, $k \geq 2$, and $\frac{n}{\log n} > \frac{100k^2}{\epsilon^2}$. Let $m = (1-\epsilon)\frac{n}{k(k-1)}$, and $p = (1-0.8\epsilon)\frac{(k-2)!}{n^{k-1}}$. Then we may couple the probability spaces such that $H_{n,m;k}$ is contained in $H_{n,p;k}$ with probability at least $1 - n^{-1}$.*

Proof. Couple the probability spaces with the random hypergraph process, which corresponds to a uniformly random permutation of all $\binom{n}{k}$ potential edges. Then, $H_{n,m;k}$ corresponds to forming a hypergraph with precisely the first m edges in this permutation, and $H_{n,p;k}$ corresponds to generating an independent random variable $X \sim \text{Bin}\left[\binom{n}{k}, p\right]$, and then taking the first X edges in the permutation.

Therefore, it suffices to show that $X \geq (1-\epsilon)\frac{n}{k(k-1)}$ with high probability. We calculate

$$\mathbb{E}[X] = \binom{n}{k}p \geq (1-0.8\epsilon)\frac{(n-k)^k}{k(k-1)n^{k-1}}.$$

Next, observe that if $\left(\frac{n-k}{n}\right)^k \geq 1-0.01\epsilon$, then we will have $\mathbb{E}[X] \geq (1-0.81\epsilon)\frac{n}{k(k-1)}$. The Chernoff bound (Theorem 3.1) would then give

$$\mathbb{P}\left[X < (1-\epsilon)\frac{n}{k(k-1)}\right] < e^{-\frac{(0.19\epsilon)^2}{2}(1-0.81\epsilon)\frac{n}{k(k-1)}}.$$

Using $\epsilon < \frac{1}{2}$, and $n > \frac{100k^2}{\epsilon^2}\log n$, we conclude that this probability is at most $n^{-1.07}$. It remains to show that $\left(\frac{n-k}{n}\right)^k \geq 1 - \frac{\epsilon}{100}$. After rearrangement, we see that this is equivalent to

$$n \geq \frac{k}{1-\left(1-\frac{\epsilon}{100}\right)^{1/k}}. \tag{1}$$

However, $e^{-x} \leq 1 - \frac{x}{2}$ for all $0 \leq x \leq 1$, so

$$\left(1-\frac{\epsilon}{100}\right)^{1/k} \leq e^{-\frac{\epsilon}{100k}} \leq 1 - \frac{\epsilon}{200k}.$$

This, together with our assumption that $n > \frac{100k^2}{\epsilon^2}\log n > \frac{200k^2}{\epsilon}$, produces (1). □

Lemma 3.5. *Let $0 < \epsilon < \frac{1}{2}$ and $n > \frac{200k^2}{\epsilon}$. Let $p = (1-0.8\epsilon)\frac{(k-2)!}{n^{k-1}}$. In the random hypergraph $H_{n,p;k}$, with probability at least $1 - n^{-1}$, all connected components are of size at most $\frac{16k}{\epsilon^2}\log n$.*

Proof. Let V be the vertex set of the entire hypergraph. Let v be a fixed vertex, and let the random variable X_v be the size of the connected component containing v. We generate X_v by exposing hyperedges one at a time via breadth-first-search. Specifically, we maintain time-varying sets A_t of distinct active vertices and B_t of completed vertices, and build a labeling of the vertices $v_1, v_2, \ldots, v_n$, initializing $A_0 = \{v\}$ and $B_0 = \emptyset$. At time t, we arbitrarily select a vertex $w \in A_t$ (if A_t is empty, we stop), define the label $v_t = w$, and set $A_{t+1} = A_t \setminus \{w\}$ and $B_{t+1} = B_t \cup \{w\}$. Also, we expose all hyperedges which have exactly $k-1$ vertices in $V \setminus \{v_1, \ldots, v_{t-1}\}$, together with w as the k-th vertex. Here, "expose" means that we reveal whether or not the potential hyperedge in fact appears in this particular realization of $H_{n,p;k}$. Finally, for each vertex other than w which is in a newly exposed hyperedge, we add a single copy of that vertex to A_{t+1}.

Importantly, we never expose the same hyperedge twice, because the hyperedges exposed at time t have the property that their smallest labeled vertex is precisely v_t. Therefore, the decisions are independent at each stage, and the number of vertices added to the queue is stochastically dominated by $(k-1)$ times the Binomial random variable $\text{Bin}\left[\binom{n}{k-1}, p\right]$. In particular, if we define the random variables $Y_t = |A_t|$, then each successive difference $Y_{t+1} - Y_t$ is stochastically dominated by $(k-1)\text{Bin}\left[\binom{n}{k-1}, p\right] - 1$. Therefore, if we define the infinite sequence Z_t as $Z_0 = 1$, $Z_{t+1} = Z_t + (k-1)\text{Bin}\left[\binom{n}{k-1}, p\right] - 1$, we may couple the probability spaces such that $Y_t \leq Z_t$ until Y_t hits 0 (the breadth-first-search is exhausted).

Let $T = \frac{16k}{\epsilon^2}\log n$. Since a binomial random variable is the sum of independent Bernoullis, the sum of independent and identically distributed binomials is another binomial. Thus the distribution of Z_T is precisely $1 + (k-1)\text{Bin}\left[\binom{n}{k-1}T, p\right] - T$. The Chernoff bound will control the probability that $Z_T \geq 1$, and this will be sufficient because if the integer $Z_T < 1$, then the breadth-first-search must have completed, as $Y_t \leq Z_t$ during it. Observe that $Z_T \geq 1$ happens precisely when $\text{Bin}\left[\binom{n}{k-1}T, p\right] \geq \frac{T}{k-1}$. Yet the expectation of this binomial is

$$\binom{n}{k-1}T(1-0.8\epsilon)\frac{(k-2)!}{n^{k-1}} \leq (1-0.8\epsilon)\frac{T}{k-1}\,,$$

so when $Z_T \geq 1$, that binomial exceeds its expectation μ by a factor of at least 0.8ϵ. Hence

$$\mathbb{P}\left[Z_T \geq 1\right] \leq e^{-\frac{(0.8\epsilon)^2}{3}\mu}\,.$$

To continue, we need a lower bound on μ. At the end of the proof of the previous lemma, we showed that $n \geq \frac{200k^2}{\epsilon}$ implies that $\left(\frac{n-k}{n}\right)^k \geq 1 - 0.01\epsilon$. Since $\left(\frac{n-k}{n}\right)^{k-1} > \left(\frac{n-k}{n}\right)^k$, and we assume $\epsilon < \frac{1}{2}$, we therefore have that

$$\mu = \binom{n}{k-1}T(1-\epsilon)\frac{(k-2)!}{n^{k-1}} \geq (1-0.81\epsilon)\frac{T}{k-1} \geq 0.595 \cdot \frac{T}{k-1}\,.$$

Thus using $T = \frac{16k}{\epsilon^2}\log n$, we have

$$\mathbb{P}\left[Z_T \geq 1\right] < e^{-\frac{(0.8\epsilon)^2}{3}\cdot 0.595 \cdot \frac{T}{k-1}} < n^{-2}\,,$$

i.e., a fixed vertex v has probability at least $1 - n^{-2}$ of having its component size at most $\frac{16k}{\epsilon^2}\log n$. A final union bound over the n vertices yields the desired result. $\square$

We now move to introduce the key parameter which characterizes the overall running time of our algorithm. This parameter has been successfully used to analyze various discrete random processes, ranging from percolation (where its name originated from statistical physics) to the theory of random graphs and stochastic coalescence processes.

Definition 3.6. *The **susceptibility** of a given hypergraph H, denoted by $\chi(H)$, is the expected size of the component which contains a uniformly random vertex. Equivalently, if the connected components are $C_1, C_2, \ldots, C_t$, then $\chi = \frac{1}{n}\sum_i |C_i|^2$.*

It turns out that the susceptibility evolves smoothly with the number of edges m. The following theorem applies the Differential Equations method to estimate its growth. Its analysis builds upon the approach used in [21], but improves the error bounds from exponential to polynomial (in both $\frac{1}{\epsilon}$ and n).

Theorem 3.7. *Let $0 < \epsilon < \frac{1}{2}$ be given, and assume that $\frac{n}{\log^6 n} > \frac{40000k^6}{\epsilon^{12}}$. Let $m = (1-\epsilon)\frac{n}{k(k-1)}$. With probability at least $1 - 3n^{-1}$, the random k-uniform hypergraph $H_{n,m;k}$ has susceptibility at most*

$$\frac{1}{\epsilon} + \frac{200k^3 \log^3 n}{\epsilon^7 \sqrt{n}}\,. \tag{2}$$

Proof. Define

$$T = (1-\epsilon)\frac{n}{k(k-1)}\,.$$

Consider the random hypergraph process $H_0, H_1, \ldots$ of Lemma 3.3. We will run this process to time $T + \log n$ which by Lemma 3.3 will contain $H_{n,m;k}$ with probability at least $1 - n^{-1}$, because $\log n < \epsilon \cdot \frac{n}{k(k-1)}$. It therefore suffices to show that with probability at least $1 - 2n^{-1}$, the susceptibility of $H_{T+\log n}$ is at most (2). We track the evolution of susceptibility by defining X_t to be the susceptibility of H_t. Suppose that in the $(t+1)$-st round, the k vertices of the incoming hyperedge lie in components $C_1, \ldots, C_k$, where some of the components may be repeated. Then, the susceptibility increases by at most

$$\frac{1}{n}\left[(|C_1| + \cdots + |C_k|)^2 - (|C_1|^2 + \cdots + |C_k|^2)\right] = \frac{2}{n}\sum_{r<s} |C_r||C_s|\,,$$

with equality only if $C_1, \ldots, C_k$ are distinct. Define the filtration $\mathcal{F}_0, \mathcal{F}_1, \ldots$ such that $\mathcal{F}_t$ captures the outcomes up to and including time t. Let us bound $\mathbb{E}\left[X_{t+1} - X_t \mid \mathcal{F}_t\right]$. For this, let $c_1, \ldots, c_z$ be the component sizes after time t. Since our process selects k independent vertices for the next hyperedge, and the hyperedge is added only if they are distinct from each other, and form a new

hyperedge, we then have

$$\begin{aligned}\mathbb{E}\left[X_{t+1} - X_t \mid \mathcal{F}_t\right] &\leq \sum_{i_1,\ldots,i_k \in [z]} \left(\frac{c_{i_1}}{n} \cdot \frac{c_{i_2}}{n} \cdots \frac{c_{i_k}}{n}\right) \cdot \frac{2}{n} \sum_{1 \leq r < s \leq k} c_{i_r} c_{i_s} \\ &= \frac{2}{n} \cdot \binom{k}{2} \cdot \sum_{i_1,\ldots,i_k \in [z]} \frac{c_{i_1}^2 c_{i_2}^2 c_{i_3} c_{i_4} \cdots c_{i_k}}{n^k} \\ &= \frac{k(k-1)}{n} \left(\sum_{i_1} \frac{c_{i_1}^2}{n}\right) \left(\sum_{i_2} \frac{c_{i_2}^2}{n}\right) \left(\sum_{i_3} \frac{c_{i_3}}{n}\right) \cdots \left(\sum_{i_k} \frac{c_{i_k}}{n}\right) \\ &= \frac{k(k-1)}{n} \left(X_t\right) \left(X_t\right) (1) \cdots (1) = \frac{k(k-1)}{n} X_t^2 \,.\end{aligned}$$

This suggests that the evolution of X_t may resemble that of the differential equation $x'(\theta) = k(k-1)x(\theta)^2$, where we parameterize $\theta = \frac{t}{n}$. So, let us define

$$x(\theta) = \frac{1}{1 - k(k-1)\theta} \,,$$

which is the solution of that differential equation with initial condition $x(0) = 1$. We use $x(\theta)$ to convert (X_t) into a supermartingale, defining

$$Y_t = X_t - x\left(\frac{t}{n}\right) - f\left(\frac{t}{n}\right) \Delta \,,$$

where

$$f(\theta) = \frac{1}{(1 - k(k-1)\theta)^3} \,, \qquad \Delta = \frac{199k^3 \log^3 n}{\epsilon^4 \sqrt{n}}$$

is the solution to the differential equation

$$f'(\theta) = 3k(k-1) \cdot x(\theta) f(\theta) \,; \qquad f(0) = 1 \,.$$

Also, define E_t to be the event that **(i)** $X_t \leq x\left(\frac{t}{n}\right) + f\left(\frac{t}{n}\right)\Delta$ and **(ii)** all components of H_t have size at most $\frac{16k}{\epsilon^2} \log n$. Then, define the stopping time τ to be either T, or the first moment that E_t fails. Define

$$Z_t = Y_{\min\{t,\tau\}} \,.$$

Then,

$$\mathbb{E}\left[Z_{t+1} - Z_t \mid \mathcal{F}_t, \overline{E_t}\right] = 0$$

and

$$\mathbb{E}\left[Z_{t+1} - Z_t \mid \mathcal{F}_t, E_t\right] \leq \frac{k(k-1)}{n} X_t^2 - \left[x\left(\frac{t+1}{n}\right) - x\left(\frac{t}{n}\right)\right] - \left[f\left(\frac{t+1}{n}\right) - f\left(\frac{t}{n}\right)\right] \Delta \,,$$

which by convexity of $x(\theta)$ and $f(\theta)$ is at most

$$\begin{aligned}\mathbb{E}\left[Z_{t+1} - Z_t \mid \mathcal{F}_t, E_t\right] &\leq \frac{k(k-1)}{n} X_t^2 - \frac{1}{n} x'\left(\frac{t}{n}\right) - \frac{1}{n} f'\left(\frac{t}{n}\right)\Delta \\ &\leq \frac{k(k-1)}{n}\left[x\left(\frac{t}{n}\right) + f\left(\frac{t}{n}\right)\Delta\right]^2 - \frac{1}{n} x'\left(\frac{t}{n}\right) - \frac{1}{n} f'\left(\frac{t}{n}\right)\Delta \\ &= \frac{k(k-1)}{n}\left[2x\left(\frac{t}{n}\right) f\left(\frac{t}{n}\right)\Delta + f\left(\frac{t}{n}\right)^2 \Delta^2\right] - \frac{1}{n} f'\left(\frac{t}{n}\right)\Delta\,.\end{aligned}$$

Our condition on n is essentially equivalent to the fact that $\Delta \leq \epsilon^2$. This implies that over the range $0 \leq \theta \leq \frac{1-\epsilon}{k(k-1)}$, we always have

$$f(\theta)\Delta \leq x(\theta)\,, \tag{3}$$

so

$$\mathbb{E}\left[Z_{t+1} - Z_t \mid \mathcal{F}_t, E_t\right] \leq \frac{\Delta}{n}\left[3k(k-1)\cdot x\left(\frac{t}{n}\right) f\left(\frac{t}{n}\right) - f'\left(\frac{t}{n}\right)\right] = 0\,,$$

and we conclude that Z_t is in fact a supermartingale. Furthermore, by part (ii) of the definition of E_t, we know that the addition of a single hyperedge cannot increase the susceptibility by more than

$$\frac{1}{n}\left[\left(k \cdot \frac{16k}{\epsilon^2}\log n\right)^2 - k \cdot \left(\frac{16k}{\epsilon^2}\log n\right)^2\right] < \frac{256k^4 \log^2 n}{\epsilon^4 n}\,.$$

Since $x(\theta)$ and $f(\theta)$ are both increasing functions, this is an upper bound for the incremental change $Z_{t+1} - Z_t$. On the other hand, the susceptibility can never decrease, and on the range $\theta < \frac{1-\epsilon}{k(k-1)}$, the derivatives $x'(\theta)$ and $f'(\theta)$ increase to $\frac{k(k-1)}{\epsilon^2}$ and $\frac{3k(k-1)}{\epsilon^4}$, respectively.

Since $x(\theta)$ and $f(\theta)$ are convex, we conclude that as t ranges from 0 to T, the maximum one-step change in Z_t is bounded by

$$C = \max\left\{\frac{256k^4 \log^2 n}{\epsilon^4 n}\,, \frac{k(k-1)}{\epsilon^2}\cdot\frac{1}{n} + \frac{3k(k-1)}{\epsilon^4}\cdot\frac{\Delta}{n}\right\} = \frac{256k^4 \log^2 n}{\epsilon^4 n}\,.$$

Yet $Z_0 = -\Delta$, so Hoeffding-Azuma (Theorem 3.2) implies that

$$\mathbb{P}\left[Z_T \geq 0\right] \leq \exp\left\{-\frac{\Delta^2}{2C^2 T}\right\} < \exp\left\{-\frac{4n\log^2 n}{k^2 T}\right\} < n^{-1}\,.$$

Also, by Lemma 3.5, the probability that H_T has a component with size exceeding $\frac{16k}{\epsilon^2}\log n$ is at most n^{-1}. Hence with probability at least $1 - 2n^{-1}$, we have that both $Z_T < 0$ and all components of H_T have size at most $\frac{16k}{\epsilon^2}\log n$. When this happens, we must never have had any E_t fail, and hence we conclude that $Y_T = Z_T < 0$, and so the susceptibility after T rounds is

$$X_T \leq x\left(\frac{T}{n}\right) + f\left(\frac{T}{n}\right)\Delta = x\left(\frac{1-\epsilon}{k(k-1)}\right) + f\left(\frac{1-\epsilon}{k(k-1)}\right)\Delta = \frac{1}{\epsilon} + \frac{1}{\epsilon^3}\cdot\frac{199k^3\log^3 n}{\epsilon^4\sqrt{n}}\,.$$

Adding $\log n$ more rounds to reach time $T + \log n$, we see that these can link at most $k \log n$ clusters, and since we conditioned on all clusters having size at most $\frac{16k}{\epsilon^2} \log n$, this can further increase the susceptibility by at most

$$\frac{1}{n} \cdot \left(k \log n \cdot \frac{16k}{\epsilon^2} \log n \right)^2 = \frac{256k^4 \log^4 n}{\epsilon^4 n} < \frac{k^3 \log^3 n}{\epsilon^7 \sqrt{n}},$$

by our initial assumption on the size of n. Therefore, with probability at least $1 - 2n^{-1}$, the total susceptibility after $T + \log n$ rounds is at most $\frac{1}{\epsilon} + \frac{200k^3 \log^3 n}{\epsilon^7 \sqrt{n}}$, as required. □

We now combine all of our results to produce our main theorem, which provides a single high-probability bound for the final sum of squared component sizes in $H_{n,m;k}$.

Proof of Theorem 1.1. As explained in section 3.1, the time for inserting a key in a component of size s is $O(k^2 s)$. This means that if the final hypergraph contains a component of size s, it took only $O(\sum_{i=1}^{s} k^2 s)$ time to insert all edges of that component, i.e., $O(k^2 s^2)$ operations. Each edge is in exactly one component, and we recognize that summing the squares of the final component sizes gives exactly n times the final susceptibility. Thus, we can bound the total running time by $O(k^2 n)$ times the final susceptibility, which by Theorem 3.7 is bounded by a constant with high probability. □

4 Acknowledgments

We thank Alan Frieze for invigorating discussions which inspired us to pursue this project.

References

[1] N. Alon and J. H. Spencer. *The probabilistic method, 3rd edition.* Wiley, New York, 2007.

[2] R. R. Amossen and R. Pagh. A new data layout for set intersection on gpus. In *Proceedings of 25th International Parallel and Distributed Processing Symposium (IPDPS)*, pages 698–708. IEEE, 2011.

[3] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM Journal on Computing*, 29(1):180–200, Feb. 1999.

[4] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking. Balanced allocations: the heavily loaded case. In *32nd Annual ACM Symposium on Theory of Computing (STOC '00)*, pages 745–754. ACM Press, 2000.

[5] J. A. Cain, P. Sanders, and N. Wormald. The random graph threshold for k-orientability and a fast algorithm for optimal multiple-choice allocation. In *Proceedings of the 18th Symposium on Discrete Algorithms (SODA)*, pages 469–476. ACM Press, 2007.

[6] M. Dietzfelbinger and F. M. auf der Heide. Simple, efficient shared memory simulations. In *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 110–119. SIGACT and SIGARCH, June 30–July 2, 1993. Extended abstract.

[7] M. Dietzfelbinger, A. Goerdt, M. Mitzenmacher, A. Montanari, R. Pagh, and M. Rink. Tight thresholds for cuckoo hashing via xorsat. In *Proceedings of 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 213–225, 2010.

[8] P. Erdős and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[9] D. Fernholz and V. Ramachandran. The k-orientability thresholds for $g_{n,p}$. In *Proceedings of the 18th Symposium on Discrete Algorithms (SODA)*, pages 459–468. ACM Press, 2007.

[10] N. Fountoulakis, M. Khosla, and K. Panagiotou. The multiple-orientability thresholds for random hypergraphs. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, pages 1222–1236, 2011.

[11] N. Fountoulakis and K. Panagiotou. Orientability of random hypergraphs and the power of multiple choices. In *Proceedings of 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 6198 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2010.

[12] A. M. Frieze and P. Melsted. Maximum matchings in random bipartite graphs and the space utilization of cuckoo hashtables. *CoRR*, abs/0910.5535, 2009. Submitted on 29 Oct 2009 (v1), revised 11 Nov 2009 (v2).

[13] P. Gao and N. C. Wormald. Load balancing and orientability thresholds for random hypergraphs. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*. ACM, 2010.

[14] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. Wiley, 2000.

[15] M. Karoński and T. Luczak. The phase transition in a random hypergraph. *J. Comput. Appl. Math.*, 142:125–135, 2002.

[16] M. Lelarge. A new approach to the orientation of random hypergraphs. In *Proceedings of the 23rd Symposium on Discrete Algorithms (SODA)*, 2012. To appear.

[17] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[18] R. Pagh and F. F. Rodler. Cuckoo hashing. *Journal of Algorithms*, 51:122–144, 2004.

[19] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.

[20] L. Stockmeyer and U. Vishkin. Simulation of parallel random access machines by circuits. *SIAM J. Comput.*, 13(2):409–422, May 1984.

[21] M. K. P.-S. L. Tom Bohman, Alan Frieze and B. Sudakov. Ramsey games with giants. *Random Structures and Algorithms*, 38:1–32, 2011.

[22] E. Upfal and A. Wigderson. How to share memory in a distributed system. *Journal of the ACM*, 34(1):116–127, Jan. 1987.