The k-orientability Thresholds for $G_{n,p}$*

Daniel Fernholz Vijaya Ramachandran

Abstract

We prove that, for $k \geq 2$, the *k-orientability threshold* for the random graph $G_{n,p}$ coincides with the threshold at which the $(k+1)$-core has average degree $2k$. The proof involves the analysis of a heuristic algorithm that attempts to find a k-orientation of the random graph.

The k-orientation threshold has several applications including offline balanced allocation with a limit of k on maximum bin-size, perfect hashing with a limit of k on maximum chain-length, and concurrent access to parallel memories through redundancy,

1 Introduction

A graph G is *k-orientable* if its edges can be directed such that every vertex has in-degree at most k. For the random graph $G_{n,p}$ with $p = d/n$ for constant d, we determine the k-orientability thresholds, defined by

$$d_k = \sup\{d : \ G_{n,d/n} \text{ is a.a.s. } k\text{-orientable}\}. \tag{1.1}$$

For $k = 1$, it is easily seen that $d_k = 1$, as the 1-orientability threshold coincides with the giant component threshold. For $k \geq 2$, k-orientability is closely related to the $(k+1)$-*core*, defined as the maximal induced subgraph of minimum degree at least $k+1$. The k-core thresholds for random graphs have been studied in [20, 4, 15, 7, 10].

It is evident that if the $(k+1)$-core of a graph G is empty, then G is k-orientable since one can repeatedly pick a vertex of degree at most k in the current graph and orient all of its incident edges as incoming to the vertex. It is also easy to see that if G contains any subgraph of average degree strictly greater than k, then G is not k-orientable. Hence, if we define

$$c_k = \sup\{c : \text{ the } (k+1)\text{-core of } G_{n,c/n} \text{ is a.a.s. empty}\}$$
$$c'_k = \sup\{c' : \text{ the } (k+1)\text{-core of } G_{n,c'/n} \text{ a.a.s. has average degree at most } 2k\}$$

we immediately have

$$c_k \leq d_k \leq c'_k. \tag{1.2}$$

The bounds for d_k in equation 1.2 were the best known until recently. A 2004 Ph.D. thesis [14] gives some improved results, but these are not tight. A conjecture by Karp and Saks on the k-orientability threshold is mentioned in [11]. Our main result is that the second inequality in (1.2) is tight.

THEOREM 1.1. *For any $k \geq 2$, $c'_k = d_k$.*

Concurrent with our paper the same result is reported in [3] using a different method.

For average degree $d > c'_k$ it is known the k-core a.a.s. has average degree strictly greater than $2k$, so we have a sharp threshold for k-orientability:

$$\begin{aligned} d_k &= \sup\{d : \ G_{n,d/n} \text{ is a.a.s. } k\text{-orientable}\} \\ &= \inf\{d : \ G_{n,d/n} \text{ is a.a.s. not } k\text{-orientable}\}. \end{aligned}$$

The proof of Theorem 1.1 involves the analysis of an algorithm for computing an upper bound on the probability that a random graph with a specified degree distribution is k-orientable. We show how to trace the execution of this algorithm by viewing the random graph as a random multigraph with a Poisson degree distribution.

The algorithm for computing the k-orientability threshold is constructive, and can be modified in trivial ways to yield a heuristic for k-orientation that succeeds a.a.s. on the random graph $G_{n,d/n}$ for any $d < d_k$ (though it may fail on an arbitrary k-orientable graph).

The main contribution of this paper is the proof of Theorem 1.1. As mentioned in the abstract this result has applications in many areas including perfect hashing, simulation of shared memory on DMM, and concurrent accesses to parallel disks (see e.g., [12, 1, 21, 5, 19, 14, 6]). As an example, in the context of perfect hashing [9] we wish to maximize c and hence m such that we can almost always store $m = c \cdot n$ keys in a hash table of size n with each slot containing at most k keys; we have two fully random and independent hash functions and each key can be placed in either one of the two slots into which it is hashed. The connection to k-orientability is seen as follows: the slots are the vertices of the random graph and keys are (random) edges; orienting edge (u, v) from u to v places the corresponding key in the slot corresponding to v. Clearly a valid placement of keys is possible if and only if this random graph is k-orientable. Our main theorem

*Department of Computer Sciences, University of Texas, Austin, TX 78712. Email: {fernholz,vlr}@cs.utexas.edu. This work was supported in part by NSF Grant CCF-0514876

gives a sharp bound d_k such that a placement is almost always possible if $2c < d_k$ and is almost never possible if $2c > d_k$. Our proof also gives an efficient algorithm to find such a placement when $2c < d_k$. Table 1 in the appendix lists some values of d_k.

In the above formulation, for $k = 1$ cuckoo hashing [18] corresponds to 1-orientability and achieves the best bound, however in that case $c = 1/2$ and there is a 50% wastage in space. The space usage for $k \geq 2$ is studied in [5, 8, 19, 6] among others, but their bounds are not tight. Our main theorem pins down the best possible bound for c for a given bound k on the maximum number of elements per slot. (These other results also consider the dynamic case (insertions and deletions); this is likely to become harder as we approach the threshold d_k and we leave this as a topic for future research.)

2 Preliminaries

2.1 Asymptotics and Probability This paper studies large random graphs asymptotically as the number of vertices $n \to \infty$. Hence, we are in effect considering a sequence of random graphs $G(1), G(2), \ldots$ where $G(n)$ is a random graph on n vertices. The asymptotic index "(n)" will generally remain implicit. Nevertheless, all asymptotic notation in this paper (i.e. big O, big Ω, etc) refers to the limit as $n \to \infty$. In this subsection only, we make the index "(n)" explicit in order to give precise asymptotic definitions.

Given a sequence of events $H(n)$, we say $H(n)$ occurs *asymptotically almost surely (a.a.s.)* if $\mathsf{P}[H(n)] = 1 - o(1)$, and H occurs *with high probability (w.h.p.)*, if $\mathsf{P}[H(n)] = 1 - n^{-\omega(1)}$. Note that the conjunction of constant number of w.h.p. events also occurs w.h.p., and that the conjunction of a polynomial number of events which hold uniformly w.h.p. also occurs w.h.p.

We shall frequently state lemmas in the form:

Assume $H(n)$; then $J(n)$ occurs w.h.p.

This means that the event $[H(n) \implies J(n)]$, or equivalently, $[\neg H(n) \vee J(n)]$, occurs w.h.p.; in particular, this does *not* mean that the conditional probability $\mathsf{P}[J(n)|H(n)]$ is always high.

2.2 Random Processes and Martingale Concentration Our concentration results will use Azuma's inequality. The setting we shall encounter repeatedly is the following. We have a random sequence of nested subsets $(A_0 \supseteq A_1 \supseteq A_2 \supseteq \ldots \supseteq A_\tau)$ of an initial set A_0, where τ is a possibly random stopping time. All random processes $z_0, z_1, \ldots$ are implicitly assumed to be functions of the history, so $z_t = z(A_0, \ldots, A_t)$.

For a real-valued random process $z_0, z_1, \ldots,$ we denote the change in z at a given time by $\Delta_t z = z_t - z_{t-1}$. We let E_t denote the conditional expectation at time t, and we abbreviate

$$E_t \Delta z = E_t \Delta_{t+1} z = \mathsf{E}[z_{t+1} - z_t | A_0, \ldots A_t].$$

Our main tool for deriving w.h.p. concentration is Azuma's inequality.

THEOREM 2.1. (COROLLARY TO AZUMA'S INEQUALITY) *Fix constants $\epsilon, c_1, c_2 > 0$, and consider a random process $z_0, z_1, \ldots, z_t$ for $t > m^{c_1}$ such that $|\Delta_s z| < c_2$ always for all $s \in [1, t]$. Then w.h.p.*

$$\sup_{s \in [0,t]} \left| z_s - \left(z_0 + \sum_{s=0}^{t-1} E_s \Delta z \right) \right| < \epsilon t.$$

Note that all w.h.p. results derived from Azuma's inequality are uniform as long as the Lipschitz constant and the length of the interval are uniformly bounded.

2.3 The Configuration Model Let π_d denote a Poisson distribution with expected value d, so $\pi_d(i) = \frac{e^{-d} d^i}{i!}$. It is well known and easy to see that the degree distribution (i.e. the fraction of vertices of various degrees, see §2.4) of the random graph $G_{n,d/n}$ is asymptotically Poisson. Hence, the random graph $G_{n,d/n}$ can be successfully studied by generating a uniformly random graph with specified sequence of vertex degrees which exhibits a Poisson degree distribution with expected value d.

We shall generate fixed-degree-sequence random graphs using the *configuration model* [2]. Our notation is as follows. We begin with an (even) set of *endpoints* A partitioned into a set of vertices V. A *multigraph* on (A, V) is a triple $G = (A, V, E)$, where the edge set E is a perfect matching of the endpoint set A. We let m denote the number of endpoints and n the number of vertices; note that the number of edges in the resulting graph is $m/2$.

Given a set of endpoints A, we let E_A denote a uniformly random matching of A, and given an endpoint partition (A, V), and we let $G(A, V) = G(A, V, E_A)$ denote the corresponding random multi-graph.

There are various technical difficulties which must be addressed when studying the random graph $G_{n,p}$ using the configuration model, most notably the fact that the configuration model does not necessarily produce a simple graph. Nevertheless, the use of the configuration model in this setting is standard, and has been employed by various authors, including [16, 17, 13, 7, 10] and in this paper we shall ignore the distinction between graphs and multigraphs, and simply study the k-orientability of the random multigraph $G(A, V)$.

2.4 The Degree Distribution Given an endpoint partition (A, V) we define the *degree* $\deg_{(A,V)}(v)$ of a vertex v to be the number of endpoints which the vertex v contains. We let

$$\Lambda_{(A,V)}(i) = \left| v \in V : \deg_{(A,V)}(v) = i \right|$$

denote the number of vertices of degree i. Accordingly, we define the *degree distribution* by

$$\lambda_{(A,V)}(i) = \frac{\Lambda_{(A,V)}(i)}{n},$$

so $\lambda_{(A,V)}$ is the distribution of the random variable corresponding to the degree of a randomly chosen vertex. We let $M(\lambda)$ denote the *average positive degree* (which excludes vertices of degree 0).

We define the degree $\deg_{(A,V)}(a)$ of an endpoint a to be the degree of the vertex containing a. For any distribution λ (degree distribution or otherwise), we define the corresponding *endpoint degree distribution* (or simply *endpoint distribution*) by

$$\mu_\lambda(i) = \frac{i\lambda(i)}{\sum_j j\lambda(j)}.$$

The endpoint distribution corresponds to the degree of a randomly chosen endpoint.

2.5 $G_{n,p}$ and the Poisson Distribution We assert without proof that the degree distribution of $G_{n,d/n}$ converges to a Poisson distribution.

PROPOSITION 2.1. *For any $d > 0$, the degree distribution of the random graph $G_{n,d/n}$ satisfies the following properties a.a.s.:*

1. *for any constant i, $\lambda(i) = \pi_d(i) \pm o(1)$;*
2. *there exist constants $C_1, C_2 > 0$ such that $\sum_{i=1}^{\infty} e^{C_1 i}\lambda(i) < C_2$.*

In particular, it is known that if a property holds a.a.s. for any random configuration $G(A, V)$ such that the degree distribution which satisfies the conditions of proposition 2.1, then the property holds a.a.s. for the random graph $G_{n,d/n}$ (see e.g. [16]).

3 A Heuristic for Finding a k-orientation

In this section we present a simple recursive algorithm which attempts to determine whether a graph G is k-orientable. For the purposes of analysis, we consider a slightly more general situation. Given a multigraph $G = (V, E)$, we consider a mapping $K : V \to \mathbb{N}$, and we define a K-orientation to be an orientation of the edges such that the indegree of each $v \in V$ is at most $K(v)$. For a constant k, the k-orientation problem is thus a special case of the K-orientation problem, by letting $K(v) = k$ for all k.

We say a vertex v is *unconstrained* if $\deg(v) \le K(v)$; hence if v is unconstrained, then every edge incident on v can be directed toward v. We say v is *partially constrained* if $K(v) < \deg(v) \le 2K(v)$. A partially constrained vertex has the property that at least half of its vertices can be directed towards v. Finally, we say v is *overconstrained* if $\deg(v) > 2K(v)$. Note that if all vertices are overconstrained then the graph is evidently not K-orientable.

We now give an informal overview of the k-orientation algorithm. The algorithm is recursive, in that, given a graph G, and a mapping K, the algorithm will produce a modified graph G' and a modified mapping K such that if G' is K'-orientable, then G is K'-orientable as well. The methods we used to construct the graph G' are based on two observations.

First, as noted above, all edges incident on any unconstrained vertex may be directed towards the given vertex. Hence, the first part of the algorithm simply directs edges towards unconstrained vertices in a greedy fashion. Specifically, if G contains an unconstrained vertex v, then we direct all incident edges towards v. Then we remove these edges and orient the edges in the residual graph G'.

The situation becomes non-trivial when there are no unconstrained vertices, in which case we employ a procedure called *excess degree reduction*. Consider a partially constrained vertex v, so $K(v) < \deg(v) \le 2K(v)$. We perform excess degree reduction on v as follows. First, we arbitrarily choose two endpoints s_1, s_2 which belong to v, with the intention of guaranteeing that at most one of s_1, s_2 are ultimately directed toward v. We consider two cases:

1. (s_1, s_2) is an edge in G (recall that we allow self-loops — see §2.3)
2. s_1 and s_2 are connected to two other endpoints r_1, r_2 in G.

In the first case, it is trivial that exactly one of $\{s_1, s_2\}$ must be directed inward. Hence we may remove s_1, s_2 from G along with the edge (s_1, s_2) to produce the graph G', and we set $K'(v) = K(v) - 1$. It is evident that G is K-orientable (if and) only if G' is K'-orientable.

In the second case, we note exactly one of $\{s_1, s_2\}$ will be directed inward if and only if exactly one of $\{r_1, r_2\}$ is directed inward. We can ensure that this second condition occurs by connecting r_1 and r_2 with an edge. Hence, we perform excess degree reduction on the pair $\{s_1, s_2\}$ by removing the endpoints $\{s_1, s_2\}$ and creating a new edge (r_1, r_2) in the graph G'. Once

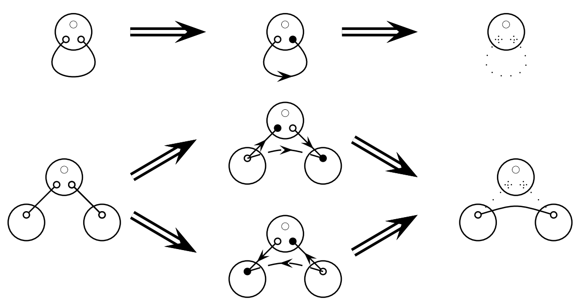

Figure 1: The two cases of excess degree reduction for two endpoints on a vertex of degree 3.

again, we set $K'(v) = K(v) - 1$, and therefore G is K-orientable only if G' is K'-orientable.

Note that, after excess degree reduction we have $\deg'(v) = \deg(v) - 2$ and $K'(v) = K(v) - 1$, and therefore after $\deg(v) - K(v)$ rounds, v will become unconstrained, in which case we may direct all incident edges towards v as discussed above. The procedure of excess degree reduction is illustrated in figure 1.

Hence, the k-orientation algorithm will proceed by repeatedly choosing a vertex v of minimum degree and applying the applicable steps in the following:

1. if v is unconstrained, direct all incident edges inwards, and remove v along with these edges;

2. if v is partially constrained, perform excess degree reduction until v is unconstrained, and then proceed as above;

3. if v is overconstrained terminate and and report failure.

3.1 The k-orientation Algorithm and the Configuration Model We incorporate the excess degree reduction steps into the configuration model, which we study algorithmically as in [16, 17]. We state the following lemma without proof.

LEMMA 3.1. *Each of the following two recursive procedures, P1 and P2, yields a uniformly random matching of a set A.*

(P1) Choose an arbitrary endpoint a_0, choose an endpoint a_1 uniformly at random from $A - \{a_0\}$, generate a uniformly random matching E' of $A' = A - \{a_0, a_1\}$, and let $E = E' \cup \{(a_0, a_1)\}$.

(P2) Choose an arbitrary pair of endpoints $a_0, a_1 \in A$, and generate a random matching E' of the set $A' = A - \{a_0, a_1\}$. With probability $1/(m-1)$, set $E = E' \cup \{(a_0, a_1)\}$. Otherwise choose an edge $e' = (b_0, b_1) \in E'$ uniformly at random (directed at random, so $\mathsf{P}[e' = (b_0, b_1)] = \mathsf{P}[e' = (b_1, b_0)]$), and let $E = (E' - \{e'\}) \cup \{(a_0, b_0), (a_1, b_1)\}$.

Parts (P1) and (P2) correspond to the two procedures used by the k-orientation algorithm described above when it does not fail. This leads to the configuration model version of the k-orientation algorithm:

ALGORITHM 3.1. *k-orientation(A, V)*
repeatedly execute the following on a vertex v of minimum degree until A is empty:

1. if $\deg(v) \leq k$, repeat until $\deg(v) = 0$:

   1a. remove an endpoint from v;

   1b. remove an endpoint chosen uniformly at random;

2. if $\deg(v) = k + j$ for $0 < j < k$, first remove $2j$ endpoints from v; then execute step 1;

3. if $\deg(v) \geq 2k$, terminate and return failure.

It is not difficult to verify the correctness of this algorithm in light of the above discussion; hence, we state the following lemma without proof.

LEMMA 3.2. *The probability that the k-orientation algorithm succeeds is at most equal to the probability that the random graph $G(A, V)$ is k-orientable.*

4 Analysis of the k-orientation Algorithm on a Random Graph

In this section, we analyze the execution of the k-orientation algorithm. We begin with some notation.

Note that the loop in algorithm 3.1 does not always remove the same number of endpoints from A. In order to simplify the analysis, we shall break the loop into individual *steps*. At each step, a single endpoint is removed from A. An *iteration* of the loop therefore consists of several consecutive steps.

We let A_t denote the set of endpoints that remain after t steps, and we let a_t denote the endpoint removed at step t, so $A_t = A_{t-1} - \{a_t\}$. The subscript t is used generally to indicate the state of the algorithm at time t. So, for example, λ_t denotes the degree distribution at time t, $\deg_t(v)$ denotes the degree of a given vertex v at time t, and so on.

We say step t is *random* if the endpoint a_t removed at time t is chosen uniformly at random, otherwise t is *deterministic*. We note that the type of step which will occur at time $(t+1)$ can be determined from the history at time t, and therefore, this information is included when computing conditional expectations. Specifically, if step $t+1$ is random, the probability of removing an endpoint of degree i is $\mu_t(i)$, and therefore, for a random step, we have $\mathsf{E}_t\,\Delta\Lambda(i) = \mu_t(i+1) - \mu_t(i)$.

We denote the minimum positive degree at time t by dmin_t. For any $h > 0$, we let

$$\tau_h = \min\{t \in [0, m] : \mathrm{dmin}_t = h\} \tag{4.3}$$

denote the first time that the minimum degree reaches h; if this never occurs, we set $\tau_h = m$. Note that the algorithm terminates at time τ_{2k}, and the algorithm succeeds if and only if the termination time is $\tau_{2k} = m$.

We define the *degree* of an iteration to be the degree of the vertex v chosen at the beginning of the iteration. Since the vertex v always has minimum positive degree, then the degree of an iteration which begins at time t is dmin_t.

4.1 k-orientability and the $(k+1)$-core

As noted in the introduction, the k-orientability problem is related to the $(k+1)$-core of a random graph, which is the maximal induced subgraph of minimum degree at least $k+1$. The $(k+1)$ core of a random graph has been studied extensively[20, 4, 15, 7, 10], and given the average degree d, it is possible to determine whether the $(k+1)$-core of $G_{n,d/n}$ is nonempty, and if so, the size and degree distribution of the $(k+1)$-core.

The $(k+1)$-core of a graph can be found using a simple algorithm which removes vertices of degree less than $k+1$ from the graph until no such vertices remain. In fact, while the minimum degree remains at most k, the k-orientation algorithm defined in this paper coincides with this algorithm for finding the $(k+1)$-core. Hence, if the graph does not contain a $(k+1)$-core, the minimum degree remains at most k throughout, and the algorithm terminates successfully.

The following lemma from [10] allows us to rule out the possibility of finding a very small $(k+1)$-core.

Lemma 4.1. ([10]) *Fix constants $\epsilon, C_1, C_2 > 0$, and consider an endpoint partition (A, V) with degree distribution λ, such that $\lambda(0) < 1-\epsilon$ and $\sum_{i=1}^{\infty} e^{C_1 i}\lambda(i) < C_2$. Then there exists a $\beta > 0$ such that the random graph $G(A, V)$ a.a.s. does not contain a $(k+1)$-core containing fewer than βn vertices.*

The following corollary follows fairly immediately from this lemma, and allows us focus on the first $m - o(m)$ steps of the execution of the k-orientation algorithm.

Corollary 4.1. *Assume the initial degree distribution λ_0 satisfies both conditions of proposition 2.1, and suppose that there exists a constant $\delta_0 > 0$ such that for arbitrary $\delta > 0$, the we have $\mathrm{dmin}_t \le k$ for all times $m - \delta_0 m < t < m - \delta m$ w.h.p. Then the random graph $G(A, V)$ is a.a.s. k-orientable.*

4.2 Intervals and Iterations

Due to corollary 4.1, we shall focus our analysis on the time interval $[0, m - \delta m]$ for arbitrary but fixed $\delta > 0$. Noting that $\mu_t(h) = \frac{h\Lambda_t(h)}{m-t}$ and that $|\Delta_t\Lambda(h)| \le 1$ always, we have the following proposition.

Proposition 4.1. *Fix arbitrary constants $C, \delta > 0$, and consider a degree $h \le C$. Then:*

1. *for any time $t < m - \delta m$, we have $|\Delta_t \mu(h)| = O(1/m)$;*
2. *for times $t < t_1 < m - \delta m$, we have $|\mu_{t_1}(h) - \mu_t(h)| = O\left(\frac{t_1 - t}{m}\right)$.*

The first claim of this proposition allows us to invoke Azuma's inequality, while the second allows us to assume that the endpoint distribution remains essentially unchanged over intervals of length $o(m)$. Typically, we are interested in the values $\mu(h) \le 2k$, so we may set the constant $C = 2k$. In order to take advantage of this proposition, we offer the following definition.

Definition 4.1. *Fix constants $\delta > 0$ and $0 < c_1 < c_2 < 1$. A* regular interval *is a time interval $[t, t_1]$ such that* **(1)** *$t_1 - t > m^{c_1}$,* **(2)** *$t_1 < m - \delta m$, and* **(3)** *a new iteration begins at time t.*

A short regular interval *is a regular interval which also satisfies $t_1 - t < m^{c_2}$.*

In particular, note that for a short regular interval and any $h < 2k$, we have $\mu_s(h) = \mu_t(h) \pm o(1)$ for $s \in [t, t_1]$.

For any time interval $[t, t_1]$, regular or otherwise, we say a condition holds at *almost every step* (or *almost every iteration*) in $[t, t_1]$, if the condition holds for all but $o(t_1 - t)$ steps (respectively iterations) during the interval.

Note that, during a given iteration, all of the endpoints belonging to the chosen vertex v are removed. And, unless one of these endpoints is chosen randomly during the the iteration, they are all removed deterministically. Since the degree of an iteration is at most $2k - 1 = O(1)$, then at any time $t < m - \delta m$, the probability that any of these endpoints are chosen at random is $O(1/m)$. Hence, with probability $1 - O(1/m)$, an iteration of degree h includes exactly h deterministic steps.

Now, if $h \le k$, then each deterministic step is followed by a random step, while if $h > k$, the first $2(h-k)$ steps are deterministic and then deterministic and random steps alternate. Hence, with probability

$1 - O(1/m)$, an iteration of degree $h \geq k$ lasts a total of $2k$ steps. We thus have the following proposition.

PROPOSITION 4.2. *For any regular interval, almost every iteration of degree h includes $\max\{2h, 2k\}$ steps, of which $k - |k - h|$ are random w.h.p.*

Next, we note that during any iteration, the chosen vertex v becomes empty. A second vertex can only become empty if all of its endpoints are selected at random, an event which occurs with probability $O(1/m)$ if the minimum degree is at least 2. Hence we have the following proposition.

PROPOSITION 4.3. *For any regular interval, almost every iteration of degree at least 2 reduces the number of nonempty vertices by exactly 1 w.h.p.*

4.3 The Degree Distribution at High Degrees

Recall from equation (4.3) that τ_h denotes the first time the event $\mathrm{dmin}_{\tau_h} \geq h$ occurs. Until τ_h, endpoints belonging to vertices of degree h or greater are only subject to random selection. Hence, it is fairly easy to determine the degree distribution at degrees $j \geq h$ for any time $t \leq \tau_h$.

The degree distribution at time 0 satisfies $\lambda(i) = \pi_d(i) \pm o(1)$, where π_d is a Poisson distribution. As part of the analysis of k-core of sparse $G_{n,p}$, Pittel et al. [20] noted that the fact that high-degree endpoints are only selected at random implies that the distribution at high degrees maintains a Poisson distribution, but with a lower expected value parameter. The following lemma states that the same occurs for the k-orientability algorithm; its proof is omitted for this extended abstract.

LEMMA 4.2. *Assume the initial degree distribution satisfies the conditions of proposition 2.1 for a Poisson distribution π_d. For any fixed constants $C > j > 0$, and any time $0 < t < \tau_j$, there exists a value $d_t < d$ such that w.h.p. for all $j \leq h < C$*

$$\lambda_t(h) = \pi_{d_t}(h) \pm o(1)$$

4.4 The Minimum Degree

Using the result of the previous section, given d_t, we can determine the value of $\lambda_t(h)$ at any time $t \leq \tau_h$. In particular, for any $h \geq 2k$, lemma 4.2 is always applicable, since the algorithm terminates at time τ_{2k}.

In this section, we examine the lower degrees of the degree distribution. The essential part of this analysis is to determine the behavior of the minimum positive degree dmin_t over time. We begin by noting that if $\mathrm{dmin}_t = h$, then there are no vertices of degree less than h at time t. Hence, the only way the minimum degree can reach $j < h$ at the start of a future iteration if the degree of a vertex with $\deg_t(v) \geq h$ is reduced via random selection. This allows us to bound the number of iterations of degree $j < h$ by the following lemma, which we state without proof.

LEMMA 4.3. *Fix a constant $\epsilon > 0$ and consider a regular interval $[t, t_1]$ such that $\mathrm{dmin}_t \geq h$ and $\mu_s(h) < \epsilon$ for all $s \in [t, t_1]$. Then at most $\epsilon(t_1 - t)$ iterations in $[t, t_1]$ have degree $h - 1$ w.h.p.*

We now have a useful corollary which follows easily from lemma 4.3 and proposition 4.1.

COROLLARY 4.2. *Consider a short regular interval $[t, t_1]$ such that $\mathrm{dmin}_t = h$. Then w.h.p.:*

1. *almost all iterations in the interval $[t, t_1]$ have degree at least $h - 1$;*
2. *if $\mu_t(h) = o(1)$ then almost all iterations in the interval $[t, t_1]$ have degree at least h; and*
3. *if $\mu_t(h) = \Omega(1)$ then almost all iterations in the interval $[t, t_1]$ have degree either h or $h - 1$.*

Based on this corollary, we begin to understand the behavior of the minimum degree at the start of successive iterations. Specifically, if $\mathrm{dmin}_t = h$ and $\mu_t(h) = \Omega(1)$, then almost all iterations in a short regular interval will have degree either h or $h - 1$. In most cases, iterations of both degrees h and $h - 1$ will occur fairly regularly in such an interval. Intuitively, a larger value of $\mu_t(h)$ implies a greater frequency of iterations of degree $h - 1$. This is because iterations of degree $h - 1$ occur due to random selections of endpoints of degree h, and the frequency of these random selections is determined by $\mu_t(h)$.

Since the minimum degree at the start of successive iterations typically alternates between h and $h - 1$, it is often useful to think of the maximum degree of an iteration in the "recent past" (say the last $m^{1/100}$ steps) as an informal and qualitative guage of the current behavior of the algorithm. We call this value the *max-min degree* at time t. The notion of max-min degree is informal, and we do not use the term in any rigorous statements, but the idea is useful for intuitive explanations.

Our next task is to determine conditions which govern changes in the max-min degree. Typically, these changes occur in two ways. First, if the minimum degree is $h - 1$, and vertices of degree $h - 1$ become exhausted then the max-min degree increases to (at least) h. Second, if the max-min degree is h and $\mu(h)$ becomes sufficiently large, then vertices of degree $h - 1$ begin to accumulate due to random selections of endpoints of degree h, and the max-min drops to $h - 1$.

Both of these transitions are determined by the value of $\mu_t(h)$, and, as shown in the following two lemmas, for degree $h > k$x the critical value for a change in max-min degree is

$$\mu_t(h) \simeq \frac{1}{2k-h+1}.$$

The next lemma states that, if $\mu_t(h) < \frac{1}{2k-h+1} - \Omega(1)$ and dmin_t is at least h, then the minimum degree becomes at least h again in the "near future," and therefore the max-min degree cannot drop to $h-1$.

Lemma 4.4. *Fix a constant $\epsilon > 0$, and consider a short regular interval $[t, t_1]$ such that $\mathrm{dmin}_t \geq h > k$ and $\mu_t(h) \leq \frac{1}{2k-h+1} - \epsilon$. Then w.h.p.*

$$\max_{t<s\leq t_1} \mathrm{dmin}_s \geq h.$$

Proof. By corollary 4.2, almost every iteration in the interval $[t, t_1]$ has degree either $h-1$ or h. The proof is by contradiction, so let us assume the lemma fails; that is, assume that $\mathrm{dmin}_s \leq h-1$ for all $t < s \leq t_1$. In this case almost every iteration has degree exactly $h-1$.

By proposition 4.2, almost every iteration of degree $h-1$ includes $2k-h+1$ random steps. Each of these random steps produces a new vertex of degree $h-1$ with probability $\mu_s(h)$, and each random step reduces the number of vertices of degree $h-1$ by 1 with probability $\mu_s(h-1) = o(1)$. Also, a single vertex of degree $h-1$ is removed deterministically each iteration. Hence, the expected change in $\Lambda(h-1)$ for an iteration of degree $h-1$ is

$$\begin{aligned}(\mu_s(h) - \mu_s(h-1))(2k-h+1) - 1 \pm o(1) \\ = \mu_s(h)(2k-h+1) - 1 \pm o(1) \\ = -\frac{\epsilon}{2k-h+1} \pm o(1).\end{aligned}$$

Since $\Lambda_t(h-1) = 0$, and the expected change is $-\Omega(1)$ for almost every iteration, it follows by Azuma's inequality that the number of vertices at time t_1 is negative w.h.p. But this is a contradiction w.h.p., since the number of vertices cannot be negative. Hence it cannot be the case that the almost all iterations have degree $h-1$, and therefore some iterations must have degree h. In particular, the minimum degree must become h in this time period w.h.p.

A similar lemma demonstrates that, if $\mu_t(h) > \frac{1}{2k-h+1} + \Omega(1)$, and $\Lambda_t(h)$ (the number of vertices of degree h) is sufficiently small, then $\Lambda_t(h)$ increases over the course of a short regular interval, and therefore the max-min degree cannot increase from $h-1$ to h.

Lemma 4.5. *Fix constants $\epsilon > 0$ and $0 < c < 1$, and consider a short regular interval $[t, t_1]$ such that, for $h > k$, we have $\mathrm{dmin}_t \geq h-1$ and $\mu_t(h) \geq \frac{1}{2k-h+1} + \epsilon$. Assume also that $\Lambda_t(h-1) < m^c$. Then $\Lambda_{t_1}(h-1) > \Lambda_t(h-1)$ w.h.p.*

Proof. Note that, since $\Lambda_t(h-1) = o(m)$, then $\mu_t(h-1) = o(1)$, and therefore by lemma 4.3 almost all iterations in the interval $[t, t_1]$ have degree either h or $h-1$ w.h.p.

We now argue as in the previous lemma that an iteration of degree $h-1$ produces an expected change in $\Lambda(h-1)$ of

$$(\mu_s(h) - \mu_s(h-1))(2k-h+1) - 1 \pm o(1) > \frac{\epsilon}{2k-h+1} + o(1),$$

and therefore the expected change is $\Omega(1)$. Also, evidently, the expected change in $\Lambda(h-1)$ for an iteration of degree h is also $\Omega(1)$. Hence, by martingale concentration, the number of vertices of degree $h-1$ increases over the interval $[t, t_1]$ w.h.p.

It is useful to restate the previous lemmas in terms of a quantity related to the endpoint distribution. For a distribution μ, let us define

$$\mu^*(h) = \frac{\mu(h)}{\sum_{j \geq h} \mu(j)}. \tag{4.4}$$

Intuitively, $\mu^*(h)$ is the probability of choosing an endpoint of degree exactly h, conditional on choosing an endpoint of degree at least h.

If h is the minimum degree then $\mu(h) = \mu^*(h)$, and if $\sum_{j<h} \mu(j) = o(1)$, then $\mu(h) = \mu^*(h) \pm o(1)$. Therefore, we may replace μ with μ^* in the previous lemmas.

Corollary 4.3. *Lemmas 4.4 and 4.5 hold if the conditions $\mu_t(h) \geq \frac{1}{2k-h+1} + \epsilon$ and $\mu_t(h) \leq \frac{1}{2k-h+1} - \epsilon$ are replaced by $\mu_t^*(h) \geq \frac{1}{2k-h+1} + \epsilon$ and $\mu_t^*(h) \leq \frac{1}{2k-h+1} - \epsilon$, respectively.*

Although this is a trivial observation, as we shall see in the next section, it is easier in certain cases to bound $\mu^*(h)$ rather than $\mu(h)$.

5 k-orientability of Random Graphs

In this section, we shall prove our main theorem, Theorem 1.1, by showing that the k-orientability algorithm succeeds a.a.s. for $G_{n,d/n}$ with average degree $d < c'_k$. Informally, we show that while the max-min degree is at least $k+1$ and the average degree is $2k - \Omega(1)$, the average degree decreases with almost every iteration. This implies that the max-min degree must eventually drop from $k+1$ back down to k. We also show that it does not go back up to $k+1$ for the remainder of the algorithm. Hence, if the average degree is at most $2k-\Omega(1)$ at the time τ_{k+1} when the $(k+1)$-core is found, then the algorithm will terminate successfully a.a.s.

The following lemma states a useful characteristic of the Poisson distribution, namely that μ^*, defined in equation (4.4), satisfies $\mu^*(h+1) > \mu^*(h)$ for all h.

Lemma 5.1. *Let μ_{π_c} denote the endpoint distribution for the Poisson distribution π_c. Then, for fixed constants $C_2 > C_1 > 0$ and a given h, there exists an $\epsilon > 0$ such that*

$$\mu^*_{\pi_c}(h+1) - \mu^*_{\pi_c}(h) > \epsilon \quad \text{for all } C_1 < c < C_2$$

Proof. We first note that the endpoint distribution for a Poisson distribution π_c is given by

$$\mu_{\pi_c}(i) = \frac{i\frac{e^{-c}c^i}{i!}}{c} = \frac{e^{-c}c^{i-1}}{(i-1)!} = \pi_c(i-1).$$

Hence, it suffices to prove that $\pi^*_c(h+1) > \pi^*_c(h)$, where $\pi^*_c(h) = \frac{\pi_c(h)}{\sum_{j\geq h}\pi_c(j)}$. Since $\pi_c(j) = \frac{e^{-c}c^j}{j!}$, we compute $\frac{\pi_c(j)}{\pi_c(h)} = \frac{h!}{j!}c^{j-h}$, and therefore $\frac{1}{\pi^*_c(h)} = \sum_{i=0}^{\infty}\frac{h!}{(h+i)!}c^i$. Hence

$$\begin{aligned}\frac{1}{\pi^*_c(h+1)} &= \sum_{i=0}^{\infty}\frac{(h+1)!}{(h+i+1)!}c^i \\ &= \sum_{i=0}^{\infty}\frac{h!}{(h+i)!}\frac{h+1}{h+i+1}c^i < \frac{1}{\pi^*_c(h)}.\end{aligned}$$

It follows that if $\pi^*_c(h+1) > \pi^*_c(h)$ for all $c > 0$. And, since $\pi^*_c(h)$ is evidently a continuous function of c for $c > 0$, the minimum value of $\pi^*_c(h+1) - \pi^*_c(h)$ is strictly positive in the interval $[C_1, C_2]$.

For the remainder of this section, we shall assume that the degree distribution at time 0 satisfies the convergence conditions of proposition 2.1 for a Poisson distribution π_d with expected value d.

Noting that $\mu^*(h)$ depends only on $\lambda(j)$ for $j \geq h$, and that by lemma 4.2, the $\lambda_t(j)$ are Poisson distributed for $j \geq 2k$, we have the following corollary to lemma 5.1.

Corollary 5.1. *For fixed $\delta > 0$, there exists a constant $\epsilon > 0$ such that w.h.p. for all $t < m - \delta m$*

$$\mu^*_t(2k+1) - \mu^*_t(2k) > \epsilon.$$

We now prove the condition $\mu^*(h+1) - \mu^*(h) - \Omega(1)$ holds, not only for $2k$, but also for any degree $h \leq 2k$, provided that the max-min degree is at least h. First, we compute the expected change in $\mu^*(h)$. The proof of the following lemma is routine calculations and is omitted.

Lemma 5.2. *For any fixed $\delta > 0$, and any time $t < m - \delta m$, the expected change in $\mu^*(h)$ during a random step is given by*

$$\begin{aligned}\mathsf{E}_t\,\Delta\mu^*(h) &= \frac{1-\mu^*_t(h)}{m-t}\left(h\mu^*_t(h+1) - (h-1)\mu^*_t(h)\right) \\ &\qquad \pm O(m^{-2}).\end{aligned}$$

We can now obtain the following corollary.

Corollary 5.2. *Fix constants $\epsilon, \delta > 0$, and a degree $h \leq 2k$. For a random step at time $t = m - \delta m$ such that $\mu^*_t(h) < 1 - \epsilon$, the following hold:*

1. *for all $\delta_0 > 0$, there exists a constant $C > 0$ such that if $\mu^*_t(h+1) - \mu^*_t(h) > \delta_0$ then $\mathsf{E}_t\,\Delta\mu^*(h) > C/m$;*

2. *for all $\delta_0 > 0$, there exist constants $\epsilon_0 > 0$ and $C_0 > 0$ such that if $\mu^*_t(h+2) - \mu^*_t(h+1) > \delta_0$ and $\mu^*_t(h+1) - \mu^*_t(h) < \epsilon_0$ then $\mathsf{E}_t\,\Delta(\mu^*(h+1) - \mu^*(h)) > C_0/m$.*

In particular, we have the following lemma.

Lemma 5.3. *Consider a regular interval $[t, t_1]$ and a degree $k \leq h \leq 2k$ such that* $\mathrm{dmin}_s \leq h$ *for all $s \in [t, t_1]$. Then, we have $\mu^*_s(h+1) - \mu^*_s(h) = \Omega(1)$ for all $s \in [t, t_1]$ w.h.p.*

Proof. [Proof Sketch] Intuitively, by corollary 5.2, the expected change in $\mu^*(h+1) - \mu^*(h)$ is positive for a random selection, so $\mu^*(h+1) - \mu^*(h)$ continues to increase so long as both $\mu^*(h+2) > \mu^*(h+1)$ and $\mu^*(h+1) > \mu^*(h)$. And, if the minimum degree is at most h, then deterministic selections do not affect $\mu^*(h+1)$, and deterministic selections cannot increase $\mu^*(h)$.

Recall that by corollary 4.1, it suffices to prove that there exists a $\delta_1 > 0$ such that for all $\delta > 0$, we have $\mathrm{dmin}_s \leq k$ for all $m - \delta_1 m \leq s \leq m - \delta m$ w.h.p. We shall now prove the existence of such an interval. First, we show that it suffices to find any interval of length $\Omega(m)$ which begins after τ_{k+1}, during which the minimum degree does not exceed k.

Lemma 5.4. *Fix $\epsilon > 0$ and $h > k$, and suppose there is a regular interval $[t, t_1]$ of length $t_1 - t > \epsilon m$ such that $t \geq \tau_h$ and such that the* $\mathrm{dmin}_s \leq h-1$ *for all $s \in [t, t_1]$. Then* $\mathrm{dmin}_s \leq h-1$ *for all $s \in [t, m-\delta m]$ w.h.p.*

Proof. Without loss of generality, assume t is the least value $t > \tau_h$ such that $\mathrm{dmin}_s < h-1$ for all $s \in [t, t+\epsilon m]$. It follows the degree of the iteration which ends at time $t-1$ is at least h. By corollary 4.3 to lemma 4.4, if

$\mu_t^*(h) \leq \frac{1}{2k-h+1} - \Omega(m)$, then we must have $\mathrm{dmin}_s \geq h$ for some $t \leq s \leq t+\epsilon m$ w.h.p. But, since by assumption this does not occur, then it must be the case that have $\mu_t^*(h) \geq \frac{1}{2k-h+1} - o(1)$ w.h.p.

Moreover, by lemma 5.3, we have $\mu_s^*(h+1) - \mu_s^*(h) = \Omega(1)$ for all $s \in [t, t_1]$, so by corollary 5.2, it follows that the expected change in $\mu^*(h)$ is $\Omega(1/m)$ for a random selection throughout this interval. And, since endpoints of degree h or greater are subject only to random selection during the interval $[t, t+\epsilon m]$, it follows that $\mu_{t+\epsilon m}^*(h) \geq \frac{1}{2k-h+1} + \Omega(1)$ w.h.p. by martingale concentration.

We now note that, so long as $\mu^*(h) \geq \frac{1}{2k-h+1} + \Omega(1)$, by corollary 4.3 to lemma 4.5, the minimum degree cannot increase to h w.h.p. And, so long as the minimum degree remains at most $h-1$, the condition $\mu^*(h+1) - \mu_t^*(h) = \Omega(1)$ continues to hold, and therefore the expected change in $\mu^*(h)$ remains positive. Therefore $\mu^*(h)$ continues to increase w.h.p., and the minimum degree never exceeds $h-1$ in the interval $[t, m-\delta m]$.

All that remains is to prove that there exists an interval $[t, t_1]$ of length $\Omega(m)$ beginning at time $t > \tau_{k+1}$, such that $\mathrm{dmin}_s \leq k$ for all $s \in [t, t_1]$. Informally, in order to find such an interval, we note that, while $\mu(h) = o(1)$ for all $h \leq k$, then almost all iterations have degree at least k, and therefore almost all iterations remove exactly $2k$ endpoints and reduce the number of nonempty vertices by exactly 1. Hence, if the average degree is $2k - \epsilon$, and the number of nonempty vertices is n, the change in the average positive degree can be computed by

$$\frac{(2k-\epsilon)n}{n} \longmapsto \frac{(2k-\epsilon)n - 2k}{n-1} = \frac{(2k-\epsilon)(n-1) - \epsilon}{n-1} = 2k - \epsilon - \frac{\epsilon}{n-1}.$$

In particular, the average degree decreases by $\Omega(1/m)$ during each iteration.

This fact has two consequences. First, the average positive degree remains less than $2k$, and so the minimum degree remains less than $2k$ as well; this implies that algorithm does not terminate during this time. Second, if the pattern of removing $2k$ endpoints and 1 nonempty vertex each iteration continues, then the average positive degree will drop below k at some time before $m - \delta m$ for a constant $\delta > 0$. But, clearly, the average degree cannot drop below $k - o(1)$ if all but $o(1)$ vertices have degree at least k. Hence, it must be the case that before $m - \delta m$, we must have $\Omega(m)$ iterations of degree less than k, which implies that $\sum_{h \leq k} \mu(h) = \Omega(1)$.

The informal argument in the previous paragraph is made formal in the following lemma.

Lemma 5.5. *Assume the $(k+1)$-core has average positive degree at most $2k - \epsilon$ w.h.p., and choose arbitrarily small constants $\epsilon_0, \delta > 0$, which may depend on ϵ. Let ρ denote the first time after τ_{k+1} such that $\sum_{j \leq k} \mu_\rho(j) > \epsilon_0$, and let $\rho = m$ if this event never occurs.*

Then, for $\epsilon_0, \delta > 0$ sufficiently small, we have $\rho < m - \delta m$ w.h.p.

Proof. Note that the $(k+1)$-core of the graph $G(A, S)$ consists precisely of the endpoints which remain at time τ_{k+1}. It follows that, for δ sufficiently small, we have $\tau_{k+1} < m - \delta m$ w.h.p., and average positive degree at this time satisfies $M_{\tau_{k+1}} < 2k - \epsilon$ w.h.p. To simplify computations, we may assume without loss of generality that $\tau_{k+1} = 0$, so the algorithm begins with minimum degree $k+1$, and that there are no non-empty vertices at this time, so the average positive degree is $M_{\tau_{k+1}} = M_0 = m/n < 2k - \epsilon$.

Now, if $\rho < m^{c_1}$, then the conditions of the lemma are satisfied. Otherwise, we consider a regular interval $[0, t]$, where $t < \rho$. Since $\sum_{j \leq k} \mu(j) < \epsilon_0$ during this interval, it follows by lemma 4.3 that at most $t\epsilon_0$ iterations in this interval have degree less than k w.h.p.

By proposition 4.2, almost all iterations of degree at least k remove $2k$ endpoints produce a single empty vertex. And, evidently, at most one empty vertex can be produced each step during any iteration of any degree. Hence, by making $\epsilon_0 > 0$ sufficiently small, we can guarantee that at time t, for arbitrary $\epsilon_1 > 0$ there are at least

$$n - t\left(\frac{1}{2k} - \epsilon_1\right)$$

non-empty vertices w.h.p.

Hence, the average positive degree at time t satisfies

$$M_t \leq \frac{m-t}{n - t/2k - \epsilon_1} = M_0 \left(\frac{m-t}{m - \frac{M_0}{2k}t - \epsilon_1 t M_0}\right) \geq M_0 \left(\frac{m-t}{m - t(1 + \frac{\epsilon}{2k} - 2k\epsilon_1)}\right).$$

In particular, if we set $t = m - \delta m$ this equation yields

$$\frac{M_t}{M_0} = \frac{\delta}{\delta + (1-\delta)(\frac{\epsilon}{2k} - 2k\epsilon_1)}.$$

Hence, by choosing $\epsilon_1 < \epsilon/(2k)^2$ and making δ sufficiently small, we can make M_t arbitrarily small as well. Clearly, though, at time $t < \rho$, since $\sum_{j \leq k} \mu_t(j) < \epsilon$ we must have $M_t > k - \epsilon_0$. It follows that $\rho < m - \delta m$ w.h.p.

We can now prove the main result of this paper.

k	c_k	d_k
2	3.351	3.588
3	5.149	5.755
4	6.799	7.843
5	8.365	9.896

Table 1: The k-orientability threshold for $G_{n,d/n}$ for small values of k. In this table $d = d_k$ gives the k-orientability threshold we establish, and $d = c_k$ gives the threshold for the emergence of the $(k+1)$ core; c_2 has been widely used as a lower bound for d_2, and c_k or lower bounds based on c_2 have been used as lower bounds for d_k, $k > 2$.

Proof. [Proof of Theorem 1.1] By assumption, the $(k+1)$-core of the random graph has average positive degree at most $2k-\epsilon$ w.h.p. It follows that $\tau_{k+1} < m-\delta m$, and that the average positive degree at this time is at most $2k-\epsilon$. By lemma 5.5, it follows that w.h.p. there exists a time $\tau_{k+1} < \rho < m-\delta m$ such that $\sum_{j\leq k} \mu_\rho(j) > \epsilon_0$. Moreover, since $\rho < m - \delta m$, by proposition 4.1 there exists a constant $\epsilon_1 > 0$ such that

$$\left|\sum_{j\leq k} \mu_s(j) - \sum_{j\leq k} \mu_\rho(j)\right| < \epsilon_0$$

for all times s satisfying $|s-\rho| < \epsilon_1 m$, and therefore $\text{dmin}_s \leq k$ for all $\rho-\epsilon_1 m < s < \rho$. Hence, by lemma 5.4, it follows that $\text{dmin}_s \leq k$ for all $\rho - \epsilon_1 m < s < m-\delta m$ w.h.p. Finally, by corollary 4.1, this implies that the algorithm terminates successfully a.a.s.

Hence, by lemma 3.2, the graph $G(A, V)$ is a.a.s. k-orientable, and therefore the random graph $G_{n,d/n}$ is a.a.s. k-orientable for any d such that the $(k+1)$-core has average degree $2k - \Omega(1)$.

References

[1] Y. Azar, A. Z. Broder, A. Karlin, and E. Upfal. Balanced allocations. *SIAM Jour. on Computing*, 29:180–200, 2000.

[2] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *Eur. J Combinatorics*, 1:311–316, 1980.

[3] J. Cain, P. Sanders, and N. Wormald. The random graph threshold for k-orientability and a fast algorithm for optimal multiple-choice allocation, 2006. This proceedings.

[4] C. Cooper. The cores of random hypergraphs with a given degree sequence. *Random Structures and Algorithms*, 25:353–375, 2004.

[5] A. Czumaj and V. Stemmann. Randomized allocation processes. *RSA*, 18:297–331, 2001.

[6] M. Dietzfelbinger and P. Weidling. Balanced allocation and dictionaries with tightly packed constant size bins. In *Proc. ICALP*, 2005.

[7] D. Fernholz and V. Ramachandran. Cores and connectivity in sparse random graphs. Technical Report UTCS TR04-13, Dept of Comp Sci, Univ Texas, Austin, 2004.

[8] D. Fotakis, R. Pagh, P. Sanders, and P. Spirakis. Space efficient hash tables with worst case constant access time. *Theory of Computing Systems*, 38:229–248, 2005.

[9] M. L. Fredman, J. Komlos, and E. Szemeredi. Storing a sparse table with $o(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.

[10] S. Janson and M. Luczak. A simple solution to the k-core problem. Technical Report 2005:31, Uppsala, 2005.

[11] R. M. Karp. Random graphs, random walks, differential equations and the probabilistic analysis of algorithms. In *Proc. STACS*, pages 1–2, 1998.

[12] R. M. Karp, M. Luby, and F. M. auf der Heide. Efficient pram simulations on a distributed memory machine. *Algorithmica*, 16:517–542, 1996.

[13] J. H. Kim. The poisson cloning model for random graphs, random directed graphs and random k-sat problems. In *Proc. COCOON*, page 2, 2004.

[14] E. Malalla. *Two-way Hashing with Separate Chaining and Linear Probing.* PhD Thesis, School of Computer Science, McGill University, 2004.

[15] M. Molloy. The pure literal rule threshold and cores in random hypergraphs. In *SODA*, pages 665–674, 2004.

[16] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures Algorithms*, 6:161–179, 1995.

[17] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Comb. Prob. Comput.*, 7:295–305, 1998.

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

[19] R. Panigrahy. Efficient hashing with lookups in two memory accesses. In *Proc. SODA*, pages 830–839, 2005.

[20] B. Pittel, J. Spencer, and N. Wormald. Sudden emergence of a giant k-core in a random graph. *J. of Combinatorial Theory, Series B*, 67:111–151, 1996.

[21] P. Sanders, S. Egnes, and J. Korst. Efficient hashing with lookups in two memory accesses. In *Proc. SODA*, pages 830–839, 2005.