Load Thresholds for Cuckoo Hashing with Overlapping Blocks

STEFAN WALZER, Universität zu Köln, Germany

We consider a natural variation of cuckoo hashing proposed by Lehman and Panigrahy (2009). Each of \( cn \) objects is assigned \( k = 2 \) intervals of size \( \ell \) in a linear hash table of size \( n \) and both starting points are chosen independently and uniformly at random. Each object must be placed into a table cell within its intervals, but each cell can only hold one object. Experiments suggested that this scheme outperforms the variant with blocks in which intervals are aligned at multiples of \( \ell \). In particular, the load threshold is higher, i.e., the load \( c \) that can be achieved with high probability. For instance, Lehman and Panigrahy (2009) empirically observed the threshold for \( \ell = 2 \) to be around 96.5% as compared to roughly 89.7% using blocks. They pinned down the asymptotics of the thresholds for large \( \ell \), but the precise values resisted rigorous analysis.

We establish a method to determine these load thresholds for all \( \ell \geq 2 \), and, in fact, for general \( k \geq 2 \). For instance, for \( k = \ell = 2 \), we get \( \approx 96.4995\% \). We employ a theorem due to Leconte, Lelarge, and Massoulié (2013), which adapts methods from statistical physics to the world of hypergraph orientability. In effect, the orientability thresholds for our graph families are determined by belief propagation equations for certain graph limits. As a side note, we provide experimental evidence suggesting that placements can be constructed in linear time using an adapted version of an algorithm by Khosla (2013).

CCS Concepts: • Theory of computation → Bloom filters and hashing; Sorting and searching;

Additional Key Words and Phrases: Cuckoo hashing, unaligned blocks, hypergraph orientability, load thresholds, randomised algorithms

ACM Reference format:
Stefan Walzer. 2023. Load Thresholds for Cuckoo Hashing with Overlapping Blocks. ACM Trans. Algor. 19, 3, Article 24 (May 2023), 22 pages.
https://doi.org/10.1145/3589558

1 INTRODUCTION

In standard cuckoo hashing [23], a set \( X = \{x_1, \ldots, x_n\} \) of objects (possibly with associated data) from a universe \( \mathcal{U} \) is to be stored in a hash table indexed by \( V = \{0, \ldots, n−1\} \) of size \( n \) such that each object \( x_i \) resides in one of two associated memory locations \( h_1(x_i), h_2(x_i) \), given by hash functions \( h_1, h_2 : \mathcal{U} \to V \). In most theoretic works, these functions are modelled as fully random functions, selected uniformly and independently from \( V^\mathcal{U} \).

A preliminary version of this work appeared in the Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP), 102:1–102:10, 2018. It was prepared at Technische Universität Ilmenau, Germany. Author’s address: S. Walzer, Universität zu Köln, Albertus-Magnus-Platz, Cologne, North Rhine-Westphalia, 50923, Germany; email: walzer@cs.uni-koeln.de.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1549-6325/2023/05-ART24 $15.00
https://doi.org/10.1145/3589558

ACM Transactions on Algorithms, Vol. 19, No. 3, Article 24. Publication date: May 2023.
The load parameter $c \in [0, 1]$ indicates the desired space efficiency, i.e., the ratio between objects and allocated table positions. Whether or not a valid placement of the objects in the table exists is well predicted by whether $c$ is above or below the threshold $c^* = \frac{1}{2}$: If $c \leq c^* - \varepsilon$ for arbitrary $\varepsilon > 0$, then a placement exists with high probability (whp), i.e., with probability approaching 1 as $n$ tends to infinity, and if $c \geq c^* + \varepsilon$ for $\varepsilon > 0$, then no placement exists whp.

If a placement is found, then we obtain a dictionary data structure representing $X \subseteq U$. To check whether an object $x \in U$ resides in the dictionary (and possibly retrieve associated data), only the memory locations $h_1(x)$ and $h_2(x)$ need to be computed and searched for $x$. Combined with results facilitating swift creation, insertion and deletion, standard cuckoo hashing has decent performance when compared to other hashing schemes at load factors around $\frac{1}{2}$ [23].

Several generalisations have been studied that allow trading rigidity of the data structure—and therefore performance of lookup operations—for load thresholds closer to 1.

- In $k$-ary cuckoo hashing, due to Fotakis et al. [8], a general number $k \geq 2$ of hash functions is used.
- Dietzfelbinger and Weidling [6] propose partitioning the table into $\frac{n}{T}$ contiguous blocks of size $\ell$ and assign two random blocks to each object via the two hash functions, allowing an object to reside anywhere within those blocks.
- By windows of size $\ell$, we mean the related idea—proposed in Lehman and Panigrahy [19] and the appendix in Reference [6]—where $x$ may reside anywhere in the intervals $[h_1(x), h_1(x) + \ell)$ and $[h_2(x), h_2(x) + \ell)$ (all indices understood modulo $n$). Compared to the block variant, the values $h_1(x), h_2(x) \in V$ need not be multiples of $\ell$, so the possible intervals do not form a partition of $V$.

The overall performance of a cuckoo hashing scheme is a story of multidimensional trade-offs and hardware dependencies, but based on experiments in References [6, 19] and roughly speaking, the following empirical claims can be made:

- $k$-ary cuckoo hashing for $k > 2$ is slower than the other two approaches. This is because lookup operations trigger up to $k$ evaluations of hash functions and $k$ random memory accesses, each likely to result in a cache fault. In the other cases, only the number of key comparisons rises, which are comparatively cheap.
- Windows of size $\ell$ offer a better trade-off between worst-case lookup times and space efficiency than blocks of size $\ell$.

Although our results are oblivious of hardware effects, they support the second empirical observation from a mathematical perspective.

### 1.1 Previous Work on Thresholds

Precise thresholds are known for $k$-ary cuckoo hashing [4, 10, 12], cuckoo hashing with blocks of size $\ell$ [3, 7], and the combination of both, i.e., $k$-ary cuckoo hashing with blocks of size $\ell$ with $k \geq 3, \ell \geq 2$ [9]. The techniques in the cited papers are remarkably heterogeneous and often specific to the cases at hand. Lelarge [20] managed to unify the above results using techniques from statistical physics that, perhaps surprisingly, feel like they grasp more directly at the core phenomena. Generalising further, Leconte, Lelarge, and Massoulié [18] solved the case where each object must occupy $j \in \mathbb{N}$ incident table positions, $r \in \mathbb{N}$ of which may lie in the same block (see also Reference [13]).

Lehman and Panigrahy [19] showed that, asymptotically, the load threshold is $1 - (2/(e + o_\ell(1)))^\ell$ for cuckoo hashing with blocks of size $\ell$ and $1 - (1/e + o_\ell(1))^{1.59\ell}$ in the case of windows, with no implication for small constant $\ell$. Beyer [2] showed in his master’s thesis that for $\ell = 2$ the threshold
Table 1. Some Thresholds \( c_{k,\ell} \) as Obtained by References [3, 4, 7, 9, 11, 12, 23] and Values of \( \gamma_{k,\ell} \) as Obtained from Our Main Theorem, Rounded to 10 Decimal Places

| \( \ell/k \) | 2    | 3               | 4               | 5               | 6               | 7               |
|-------------|------|----------------|----------------|----------------|----------------|----------------|
| 1           | 0.5  | 0.9179352767   | 0.9767701649   | 0.9924383913   | 0.997395528    | 0.9990637588   |
| 2           | 0.8970118682 | 0.9882014140   | 0.9982414840   | 0.9997243601   | 0.9999568737   | 0.9999934349   |
| 3           | 0.9591542686 | 0.9972857393   | 0.9997951434   | 0.9999851453   | 0.9999987975   | 0.999999929    |
| 4           | 0.9803697743 | 0.9992531564   | 0.9999720661   | 0.999990737    | 0.999999721    | 0.999999992    |

| \( \ell/k \) | 2    | 3               | 4               | 5               | 6               | 7               |
|-------------|------|----------------|----------------|----------------|----------------|----------------|
| 1           | 0.5  | 0.9179352767   | 0.9767701649   | 0.9924383913   | 0.997395528    | 0.9990637588   |
| 2           | 0.9649949234 | 0.9968991072   | 0.9996335076   | 0.9999529036   | 0.9999937602   | 0.999999631    |
| 3           | 0.9944227538 | 0.9998255112   | 0.9999998722   | 0.9999999843   | 0.9999999992   |
| 4           | 0.9989515932 | 0.9999896830   | 0.9999998577   | \( \approx 1 \) | \( \approx 1 \) |

In both tables, the line for \( \ell = 1 \) corresponds to plain \( k \)-ary cuckoo hashing, reproduced here for comparison.

is at least 0.829 and at most 0.981. To our knowledge, this is an exhaustive list of published work concerning windows.

In a spirit similar to cuckoo hashing with windows, Porat and Shalem [24] analyse a scheme where memory is partitioned into pages and a bucket of size \( k \) is a set of \( k \) memory positions from the same page (not necessarily contiguous). The authors provide rigorous lower bounds on the corresponding thresholds as well as empirical results.

1.2 Our Contribution

We provide precise thresholds for \( k \)-ary cuckoo hashing with windows of size \( \ell \) for all \( k, \ell \geq 2 \). In particular, this solves the case of \( k = 2 \) left open in References [6, 19]. Note the pronounced improvements in space efficiency when using windows over blocks, for instance in the case of \( k = \ell = 2 \), where the threshold is at roughly 96.5% instead of roughly 89.7%.

Formally, for any \( k, \ell \geq 2 \), we identify real analytic functions \( f_{k,\ell}, g_{k,\ell} \) (see Section 6), such that for \( \gamma_{k,\ell} = \inf_{\lambda > 0} \{ f_{k,\ell}(\lambda) \mid g_{k,\ell}(\lambda) < 0 \} \), we have

**Main Theorem.** The threshold for \( k \)-ary cuckoo hashing with windows of size \( \ell \) is \( \gamma_{k,\ell} \), in particular, for any \( \epsilon > 0 \),

1. if \( c > \gamma_{k,\ell} + \epsilon \), then no valid placement of objects exists whp, and
2. if \( c < \gamma_{k,\ell} - \epsilon \), then a valid placement of objects exists whp.

While \( f_{k,\ell} \) and \( g_{k,\ell} \) are very unwieldy, with ever more terms as \( \ell \) increases, numerical approximations of \( \gamma_{k,\ell} \) can be attained with mathematics software, as explained in Section 9. We provide some values in Table 1.

1.3 Methods

The obvious methods to model cuckoo hashing with windows either give probabilistic structures with awkward dependencies or the question to answer for the structure follows awkward rules. Our first non-trivial step is to transform a preliminary representation into a hypergraph with \( n \) vertices, \( cn \) uniformly random hyperedges of size \( k \), an added deterministic cycle, and a question strictly about the orientability of this hypergraph.

In the new form, the problem is approachable by a combination of belief propagation methods and the objective method [1], adapted to the world of hypergraph orientability by Lelarge [20]

ACM Transactions on Algorithms, Vol. 19, No. 3, Article 24. Publication date: May 2023.
in his insightful paper. These results were further strengthened by a Theorem in Reference [18], which we apply at a critical point in our argument.

As the method is fundamentally about approximate sizes of incomplete orientations, it leaves open the possibility of \( o(n) \) unplaced objects; a gap that can be closed in an afterthought with standard methods.

A comprehensive overview of our argument is given in Section 3. Key propositions are found in Sections 4 to 8.

1.4 Further Discussion

At the end of this article, we touch on three further issues that complement our results but are somewhat detached from our main theorem.

**Numerical approximations of the thresholds.** In Section 9, we explain how mathematics software can be used to get approximations for the values \( y_{k,\ell} \), which have been characterised only implicitly.

**Speed of convergence.** In Section 10, we provide experimental results with finite table sizes to demonstrate how quickly the threshold behaviour emerges.

**Constructing orientations.** In Section 11, we examine the Local Search Allocation (LSA) algorithm by Khosla for insertion of elements, adapted to our hashing scheme. Experiments suggest an expected constant running time per element as long as the load is bounded away from the threshold, i.e., \( c < y_{k,\ell} - \varepsilon \) for some \( \varepsilon > 0 \).

2 DEFINITIONS AND NOTATION

A cuckoo hashing scheme specifies for each object \( x \in X \) a set \( A_x \subset V \) of table positions that \( x \) may be placed in. For our purposes, we may identify \( x \) with \( A_x \). In this sense, \( H = (V, X) \) is a hypergraph, where table positions are vertices and objects are hyperedges. The task of placing objects into admissible table positions corresponds to finding an *orientation* of \( H \), which assigns each edge \( x \in X \) to an incident vertex \( v \in x \) such that no vertex has more than one edge assigned to it. If such an orientation exists, then \( H \) is *orientable*.

We now restate the hashing schemes from the introduction in this hypergraph framework, switching to letters \( e \) (and \( E \)) to refer to (sets of) edges. We depart in notation, but not in substance, from definitions given previously, e.g., References [5, 8, 19]. Illustrations are available in Figure 1.

Concerning *k-ary cuckoo hashing* the hypergraph is given as

\[
H_n = H_{n, cn}^k := (\mathbb{Z}_n, E = \{e_1, e_2, \ldots, e_{cn}\}), \quad \text{for } e_i \leftarrow \left[\frac{\mathbb{Z}_n}{k}\right],
\]

where \( \mathbb{Z}_n = \{0, 1, \ldots, n-1\} \), and for a set \( S \) and \( k \in \mathbb{N} \), we write \( e \leftarrow \left[\frac{S}{k}\right] \) to indicate that \( e = \{s_1, \ldots, s_k\} \) is obtained by picking \( s_1, \ldots, s_k \) independently and uniformly at random from \( S \).

There is a subtle difference to picking \( e \) uniformly at random from \( \left[\frac{S}{k}\right] \), the set of all \( k \)-subsets of \( S \), as the elements \( s_1, \ldots, s_k \) need not be distinct. We therefore understand \( e \) as a multiset. Also, we may have \( e_i = e_j \) for \( i \neq j \), so \( E \) is a multiset as well.\(^1\)

\(^1\)While our choice for the probability space is adequate for cuckoo hashing and convenient in the proof, such details are inconsequential. Choosing \( H_n \) uniformly from the set of all hypergraphs with \( cn \) distinct edges all of which contain \( k \) distinct vertices would be equivalent for our purposes.
Assuming the table size $n$ is a multiple of $\ell$, \textbf{k-ary cuckoo hashing with blocks of size $\ell$} is modelled by the hypergraph

$$B_n = B_{n,cn}^{k,\ell} := (\mathbb{Z}_n, \{e'_1, e'_2, \ldots, e'_cn\}), \quad \text{where } e'_i = \bigcup_{j \in e_i} [j, (j + 1)\ell] \text{ and } e_i \sim \left[\mathbb{Z}_n/\ell \right]/k,$$

that is, each hyperedge is the union of $k$ blocks chosen uniformly at random from the set of all blocks, which are the $n/\ell$ intervals of size $\ell$ in $\mathbb{Z}_n$ that start at a multiple of $\ell$. Note that for $\ell = 1$, we recover $H_n$.

Similarly, \textbf{k-ary cuckoo hashing with windows of size $\ell$} is modelled by

$$W_n = W_{n,cn}^{k,\ell} := (\mathbb{Z}_n, \{e'_1, e'_2, \ldots, e'_cn\}), \quad \text{where } e'_i = \bigcup_{j \in e_i} [j, j + \ell] \text{ and } e_i \sim \left[\mathbb{Z}_n/k \right],$$

that is, each hyperedge is the union of $k$ windows chosen uniformly at random from the set of all windows, which are the $n$ intervals of size $\ell$ in $\mathbb{Z}_n$, this time without alignment restriction. Note that intervals wrap around at the ends of the set $\{0, \ldots, n - 1\}$ with no awkward “border intervals.” Again, for $\ell = 1$, we recover $H_n$.

3 \hspace{1em} OUTLINE OF THE PROOF

\textbf{Step 1: A tidier problem.} The elements of an edge $e$ of $B_n$ and $W_n$ are not independent, as $e$ is the union of $k$ intervals of size $\ell$. This poorly reflects the actual tidiness of the probabilistic object. We may obtain a model with independent elements in edges, by switching to a more general notion of what it means to orient a hypergraph.

Formally, given a weighted hypergraph $H = (V, E, \eta)$ with weight function $\eta : V \cup E \rightarrow \mathbb{N}$, an orientation $\mu$ of $H$ assigns to each pair $(e, v)$ of an edge and an incident vertex a number $\mu(e, v) \in \mathbb{N}_0$ such that

$$\forall e \in E : \sum_{v \in e} \mu(e, v) = \eta(e), \quad \text{and } \forall v \in V : \sum_{e \ni v} \mu(e, v) \leq \eta(v).$$

We will still say that an edge $e$ is oriented to a vertex $v$ (possibly several times) if $\mu(e, v) > 0$. One may be inclined to call $\eta(v)$ a capacity for $v \in V$ and $\eta(e)$ a demand for $e \in E$, but we use the same letter in both cases as the distinction is dropped later anyway.

Orientability of $H, B$, and $W$ from earlier is also captured in the generalised notion with implicit weights of $\eta \equiv 1$.  

ACM Transactions on Algorithms, Vol. 19, No. 3, Article 24. Publication date: May 2023.
Fig. 2. (a) In k-ary cuckoo hashing with blocks of size $\ell$ (here $k = \ell = 3$), we can contract each block into a single vertex of weight $\ell$ to obtain a simpler but equivalent representation of the orientation problem. (b) In k-ary cuckoo hashing with windows of size $\ell$, a similar idea works, but additional helper edges (drawn as ●) of weight $\ell - 1$ are needed (see Proposition 1).

A simplified representation of $B_n$ is straightforward to obtain. We provide it mainly for illustration purposes, see Figure 2(a):

\[
\hat{B}_n := \hat{B}_{n, cn} := (\mathbb{Z}_{n/\ell}, \{e_1, e_2, \ldots, e_{cn}\}, \eta), \quad \text{where } e_i \leftarrow \left[\mathbb{Z}_{n/\ell}\right], \quad \eta(v) = \ell \quad \text{for } v \in \mathbb{Z}_{n/\ell} \quad \text{and edge weight } \eta(e_i) = 1 \quad \text{for } 1 \leq i \leq cn.
\]

In $\hat{B}_n$, each group of $\ell$ vertices of $B_n$ representing one block is now contracted into a single vertex of weight $\ell$ and edges contain $k$ independent vertices representing blocks instead of $k\ell$ dependent vertices. It is clear that $B_n$ is orientable if and only if $\hat{B}_n$ is orientable.

In a similar spirit, we identify a transformed version $W_n$ for $W_n$, but this time the details are more complicated as the vertices have an intrinsic linear geometry, whereas $B_n$ featured essentially an unordered collection of internally unordered blocks. The ordinary edges in $\hat{W}_n$ also have size $k$ instead of size $k\ell$, but we need to introduce additional helper edges that capture the linear geometry of $\mathbb{Z}_n$, see Figure 2(b). We define

\[
\hat{W}_n := \hat{W}_{n, cn} := (\mathbb{Z}_n, C_n \cup \{e_1, \ldots, e_{cn}\}, \eta), \quad \text{with ordinary edges } e_i \leftarrow \left[\mathbb{Z}_n\right], \quad \text{helper edges } C_n = \{h_i := (i, i + 1) \mid i \in \mathbb{Z}_n\},
\]

vertex weight $\eta(v) = \ell$ for $v \in \mathbb{Z}_n$,

and edge weights $\eta(h) = \ell - 1$, $\eta(e) = 1$ for $h \in C_n, e \in \{e_1, \ldots, e_{cn}\}$.

Note that formally the graphs $W_n$ and $\hat{W}_n$ are random variables on a common probability space. An outcome $\omega = (e_i)_{1 \leq i \leq cn}$ from this space determines both graphs.

The following proposition justifies the definition. It is proved in Section 4.

**Proposition 1.** $\hat{W}_n$ is orientable if and only if $W_n$ is orientable.\(^2\)

An important merit of $\hat{W}_n$ that will be useful in Step 3 is that it is locally tree-like, meaning each vertex has a probability of $o(1)$ to be involved in a constant-length cycle. Here, by a cycle in a hypergraph, we mean a sequence of distinct edges $e_1, e_2, \ldots, e_j$ such that two consecutive edges share a vertex and $e_j$ and $e_1$ share a vertex.

\(^2\)Formally, this should read: The events \{$W_n$ is orientable\} and \{$\hat{W}_n$ is orientable\} coincide.
Note the interesting special case $\hat{W}^{2,2}_{n, cn}$, which is a cycle of length $n$ with $cn$ random chords, unit edge weights and vertices of weight 2. Understanding the orientability thresholds for this graph seems interesting in its own right, not just as a means to understand $\hat{W}^{2,2}_{n, cn}$.

**Step 2: Incidence Graph and Allocations.** The next step is by no means a difficult or creative one, we merely perform the necessary preparations needed to apply [18], introducing their concept of an allocation in the process.

This will effectively get rid of the asymmetry between the roles of vertices and edges in the problem of orienting $\hat{W}_n$, by switching perspective in two simple ways. The first is to consider the incidence graph $G_n$ of $\hat{W}_n$ instead of $\hat{W}_n$ itself, i.e., the weighted bipartite graph

$$G_n = G_{n, cn}^{k, \ell} = (\bigcup_{A \in \mathcal{A}_C} (A \cup \{e_1, \ldots, e_{cn}\}), \bigcup_{B \in \mathcal{A}_R} E(G_n), \eta). \quad (7)$$

We use $A = A_C \cup A_R$ to denote those vertices of $G_n$ that were edges in $\hat{W}_n$, and $B$ for those vertices of $G_n$ that were vertices in $\hat{W}_n$. Vertices $a \in A$ and $b \in B$ are adjacent in $G_n$ if $b \in a$ in $\hat{W}_n$. The weights $\eta$ on vertices and edges in $\hat{W}_n$ are now vertex weights with $\eta(a) = \ell - 1$, $\eta(a_R) = 1$, $\eta(b) = \ell$ for $a_C \in A_C$, $a_R \in A_R$, $b \in B$. The notion of $\mu$ being an orientation translates to $\mu$ being a map $\mu : E(G_n) \to \mathbb{N}_0$ such that $\sum_{b \in N(a)} \mu(a, b) = \eta(a)$ for all $a \in A$ and $\sum_{a \in N(b)} \mu(a, b) \leq \eta(b)$ for all $b \in B$. Note that vertices from $A$ need to be saturated ("$= \eta(a)$" for $a \in A$) while vertices from $B$ need not be ("$\leq \eta(b)$" for $b \in B$). This leads to the second switch in perspective.

Dropping the saturation requirement for $A$, we say $\mu$ is an allocation if $\sum_{u \in N(v)} \mu(u, v) \leq \eta(v)$ for all $v \in A \cup B$.

Clearly, any orientation is an allocation, but not vice versa; for instance, the trivial map $\mu \equiv 0$ is an allocation. Let $|\mu|$ denote the size of an allocation, i.e., $|\mu| = \sum_{e \in E} \mu(e)$. By bipartiteness, no allocation can have a size larger than the total weight of $A$, i.e.,

$$\text{for all allocations } \mu : |\mu| \leq \eta(A) = \sum_{a \in A} \eta(a) = |A_C| \cdot (\ell - 1) + |A_R| \cdot 1 = (\ell - 1 + c)n,$$

and the orientations of $G_n$ are precisely the allocations of size $\eta(A)$. We conclude:

**Proposition 2.** Let $M(G_n)$ denote the maximal size of an allocation of $G_n$. Then

$$\frac{M(G_n)}{n} = \ell - 1 + c \quad \text{if and only if } \hat{W}_n \text{ is orientable.}$$

**Step 3: The Limit T of G_n.** Reaping the benefits of step 1, we find $G_n$ to have $O(1)$ cycles of length $O(1)$ whp. To capture the local appearance of $G_n$ even more precisely, let the $r$-ball around a vertex $v$ in a graph be the subgraph induced by the vertices of distance at most $r$ from $v$. Then the $r$-ball around a random vertex of $G_n$ is distributed, as $n$ gets large, more and more like the $r$-ball around the root of a random infinite rooted tree $T = T^{k, \ell}_c$. The tree $T$ is distributed as follows, with weighted nodes of types $A_C, A_R, B$,

- The root of $T$ is of type $A_C, A_R, B$ with probability $\frac{1}{2\pi\ell}$, $\frac{e}{2\pi\ell}$, and $\frac{1}{2\pi\ell}$, respectively.
- If the root is of type $A_C$, then it has two children of type $B$. If it is of type $A_R$, then it has $k$ children of type $B$. If it is of type $B$, then it has two children of type $A_C$ and a random number $X$ of children of type $A_R$, where $X \sim \text{Po}(ck)$. Here $\text{Po}(\lambda)$ denotes the Poisson distribution with parameter $\lambda$.
- A vertex of type $A_C$ that is not the root has one child of type $B$. A vertex of type $A_R$ that is not the root has $k - 1$ children of type $B$.
- A vertex of type $B$ that is not the root has a random number $X$ of children of type $A_R$, where $X \sim \text{Po}(ck)$. If its parent is of type $A_C$, then it has one child of type $A_C$; otherwise, it has two children of type $A_C$.
- Vertices of type $A_C, A_R, B$ have weight $\ell - 1, 1, and \ell$, respectively.
All random decisions should be understood to be independent. A type is also treated as a set containing all vertices of that type. In Section 5, we briefly recall the notion of local weak convergence and argue that the following holds:

**Proposition 3.** Almost surely,\(^3\) \((G_n)_{n \in \mathbb{N}} = (G_{n,cn})_{n \in \mathbb{N}}\) converges locally weakly to \(T = T_{c,\ell}^k\).

**Step 4: The Method of Lelarge.** We are now in a position to apply a powerful theorem due to Leconte, Lelarge, and Massoulié [18] that characterises \(\lim_{n \to \infty} \frac{M(G_n)}{n}\) in terms of solutions to belief propagation equations for \(T\). Put abstractly: The numerical limit of a function of \(G_n\) is expressed as a function of the graph limit of \(G_n\). We elaborate on details and deal with the equations in Sections 6 and 7. After condensing the results into a characterisation of \(\gamma_{k,\ell} \in (0,1)\) in terms of “well-behaved” functions, we obtain

**Proposition 4.**

\[
\lim_{n \to \infty} \frac{M(G_{n,cn})}{n} = \begin{cases} 
\ell - 1 + c & \text{almost surely if } c < \gamma_{k,\ell}, \\
< \ell - 1 + c & \text{almost surely if } c > \gamma_{k,\ell}.
\end{cases}
\]

**Step 5: Closing the Gap.** It is important to note that we are not done, as

\[
\lim_{n \to \infty} \frac{M(G_{n,cn})}{n} = \ell - 1 + c \quad \text{a.s. does not imply} \quad M(G_{n,cn}) = n \cdot (\ell - 1 + c) \quad \text{whp.} \quad (8)
\]

We still have to exclude the possibility of a gap of size \(o(n)\) on the right hand side; imagine for instance \(M(G_{n,cn}) = (\ell - 1 + c)n - \sqrt{n}\) to appreciate the difference. In the setting of cuckoo hashing with double hashing [17], the analogue of this pesky distinction was in the way of obtaining precise thresholds, until recently tamed by a somewhat lengthy case analysis in Reference [21]. We should therefore treat this carefully.

Luckily the line of reasoning by Lelarge [20] can be adapted to our more general setting. The key is to prove that if not all objects can be placed into the hash table, then the configuration causing this problem has size \(\Theta(n)\) (and those large overfull structures do not go unnoticed on the left-hand side of Equation (8)).

**Lemma 3.1.** There is a constant \(\delta > 0\) such that whp no set of \(0 < t < \delta n\) vertices in \(\hat{W}_n\) (of weight \(\ell t\)) induces edges of total weight \(\ell t\) or more, provided \(c \leq 1\).

The proof of this Lemma (using first moment methods) and the final steps towards our main theorem are found in Section 8.

The following sections provide the details of the technical argument. Conclusion, outlook, and acknowledgments can be found at the end of the article.

### 4 EQUIVALENCE OF \(W_n\) AND \(\hat{W}_n\) WITH RESPECT TO ORIENTABILITY

In this section, we prove Proposition 1, i.e., we show that \(W_n\) is orientable if and only if \(\hat{W}_n\) is orientable. Recall the relevant definitions in Equations (3), (4), and (6). Some ideas are illustrated in Figure 3. We start with the hypergraph \(W_n\) in Figure 3(a). From this, Figure 3(b) is obtained by introducing one “broker-vertex” (●) for each interval of size \(\ell\) in the table, through which the incidences of the objects (○) are “routed” as shown. The purpose of each broker-vertex is to “claim” part of its interval on behalf of incident objects. To manage these claims, we imagine a “separator” (●) between each pair of adjacent broker-vertices that, by pointing between two table cells, indicates where the claim of one broker-vertex ends and the claim of the next broker-vertex

---

3Meaning: With probability 1.
begins; see Figure 3(c). There are \( \ell \) possible “settings” for each separator. The separators can be modelled as edges of weight \( \ell - 1 \) with \( \ell \) possible ways to distribute this weight among the two incident broker-vertices that have weight \( \ell \). The table is then fully implicit, which gives \( \hat{W}_n \) in Figure 3(d).

**Proof of Proposition 1** We introduce the shorthand \( w_i := [i, i + \ell) \) for this proof.

\( \Rightarrow \) Let \( \mu \) be an orientation of \( W_n \). We will define an orientation \( \hat{\mu} \) of \( \hat{W}_n \). Recall from Equations (3) and (6) how an edge \( e' = \bigcup_{j \in e} w_j \) of \( W_n \) is defined in terms of an edge \( e \) of \( \hat{W}_n \). If \( \mu \) directs \( e' \) to a table cell \( x \in Z_n \), then we pick \( j \in e \) with \( x \in w_j \). We let \( \hat{\mu} \) direct \( e \) to \( j \) and also assign to \( x \) the label \( w_j \).

Note that, since \( \mu \) is an orientation, each \( x \in Z_n \) receives at most one label this way, and that the label stems from \( \{w_{x-\ell+1}, \ldots, w_x\} \).

We still have to orient the helper edges \( h_i = (i, i + 1) \) of weight \( \ell - 1 \). For this, we count the number \( r_i \) of elements in \( w_i \cap w_{i+1} \) with a label that is to the right, i.e., stems from \( \{w_{i+1}, w_{i+2}, \ldots, w_{i+\ell-1}\} \). We then set \( \hat{\mu}(h_i, i) = r_i \) and \( \hat{\mu}(h_i, i + 1) = \ell - 1 - r_i \).

We now check that the weight of any vertex \( i \in Z_n \) is respected, i.e., we check that

\[
\hat{\mu}(h_i, i) + \hat{\mu}(h_{i-1}, i) + \sum_{\text{ordinary edge } e} \hat{\mu}(e, i) \leq \ell.
\]

From ordinary edges, the contribution is 1 for each \( x \in w_i \) with label \( w_i \). From \( h_i \) the contribution is the number of \( x \in w_{i-1} \cap w_i \) with label in \( \{w_{i+1}, \ldots, w_{i+\ell-1}\} \) and from \( h_{i-1} \), the contribution is the number of \( x \in w_{i-1} \cap w_i \) not having a label from \( \{w_i, \ldots, w_{i+\ell-2}\} \). The three conditions are clearly mutually exclusive, so each \( x \in w_i \) can contribute at most 1, giving a total contribution of at most \( |w_i| = \ell \) as required.

\( \Leftarrow \) Let \( \hat{\mu} \) be an orientation of \( \hat{W}_n \). Define \( s_i := \hat{\mu}(h_{i-1}, i) \) and \( t_i := \hat{\mu}(h_i, i) \). Let further \( \bar{w}_i := [i + s_i, i + \ell - t_i] \subseteq w_i \).

Crucially, \( \{\bar{w}_i \mid i \in Z_n\} \) forms a partition of \( Z_n \). This follows from the following properties:

\[
\max(\bar{w}_i) = i + \ell - t_i - 1, \quad \min(\bar{w}_{i+1}) = i + 1 + s_{i+1}
\]

and \( t_i + s_{i+1} = \hat{\mu}(h_i, i) + \hat{\mu}(h_i, i+1) = \eta(h_i) = \ell - 1 \).

Here, if cyclic intervals span the “seam” of the cycle, max and min should be reinterpreted in the natural way. Now let \( e_1^{(i)}, \ldots, e_p^{(i)} \) be the ordinary edges directed to \( i \) by \( \hat{\mu} \). Since \( \hat{\mu} \) respects \( \eta(i) = \ell \), we have \( p_i + s_i + t_i \leq \ell \), so \( p_i \leq \ell - s_i - t_i = |w_i| \). We can now define the orientation \( \mu \) of \( W_n \) to direct each \( e_j^{(i)} \) to \( i + s_j + j - 1 \in Z_n \) for \( 1 \leq j \leq p_i \) and \( i \in Z_n \).
5 LOCAL WEAK CONVERGENCE OF $G_n$ TO $T$

Recall the definitions of the finite graph $G_n$ in Equation (7) and the infinite rooted tree $T$ in Step 3 of Section 3. We obtain the rooted graph $G_n(\cdot)$ from $G_n$ by distinguishing one vertex—the root—uniformly at random. For any rooted graph $R$ and $d \in \mathbb{N}$, let $R_d$ denote the rooted subgraph of $R$ induced by the vertices at distance at most $d$ from the root. We treat two rooted graphs as equal if there is an isomorphism between them preserving root and vertex types. Refer to Figure 4 for a possible outcome of $T|_3$ and $G_n(\cdot)|_3$.

With this notation, we can clarify Proposition 3, i.e., what we mean by saying that almost surely $(G_n)_{n \in \mathbb{N}}$ converges locally weakly to $T$, namely,

$$\forall d \in \mathbb{N} : \forall \text{ rooted graph } H : \lim_{n \to \infty} \Pr_{\omega \in V(G_n)}[G_n(\cdot)|_d = H] = \Pr[T_d = H] \text{ almost surely.} \quad (9)$$

Here “almost surely” refers to the randomness in choosing the sequence $(G_n)_{n \in \mathbb{N}}$ while $\Pr_{\omega \in V(G_n)}$ refers to the randomness in rooting $G_n$. For a more general definition and a plethora of illustrating examples, refer to the excellent survey on the objective method by Aldous and Steele [1].

The proof of Equation (9) is technical but standard. We will take the time to sketch the proof of the slightly simpler statement:

$$\forall d \in \mathbb{N} : \forall \text{ rooted graph } H : \lim_{n \to \infty} \Pr_{G_n, \omega \in V(G_n)}[G_n(\cdot)|_d = H] = \Pr[T_d = H]. \quad (9')$$

Note that in Equation (9) the probability in the limit is a random variable (depending on $(G_n)_{n \in \mathbb{N}}$), while the probability in Equation (9') is not. We refer to Reference [17] where a similar proof is done in full, including concentration arguments to settle the difference between the analogues of Equations (9) and (9').

**Proof of Equation (9').** For simplicity, we shall assume that child vertices of the same type are ordered (for $G_n(\cdot)$ and $T$ just fix an ordering at random). Correspondingly, we prove the more fine-grained version of (9') where equality indicates the existence of an isomorphism preserving roots, types and child orderings.

Let $H$ be a possible outcome of $T_d$, let $t \in \{A_C, A_R, B\}$ be the type of the root of $H$ and let $v_1, \ldots, v_s$ be the vertices of type $B$ in $H$, except for leaves, in breadth-first-search ordering. Let further $x_i \in \mathbb{N}_0$ denote the number of $v_i$’s children of type $A_R$ for $1 \leq i \leq s$. Checking the definition of $T$, the sequence $(t, x_1, \ldots, x_s)$ contains all random decisions that $T$ has to “get right” in order.

Fig. 4. One possibility of what $T|_3$ may look like for $k = 3$. Since the infinite random tree $T$ is designed to reflect the local characteristics of $G_n$, it is also a possibility for $G_n(\cdot)|_3$. Actually, the distributions of $G_n(\cdot)|_3$ and $T|_3$ are asymptotically equal.
for $T|_d$ to coincide with $H$, i.e.,
\[
\Pr[T|_d = H] = \Pr[\text{root}(T) \in t] \cdot \prod_{i=1}^{s} \Pr[X_i = x_i],
\]
where $X_i \sim \text{Po}(ck)$ for $i \in \mathbb{N}$ are the independent random variables used in the construction of $T$, in breadth-first-search order.

To compare this to $\Pr_{G_n, o \in V(G_n)}[G_n(o)|_d = H]$, we shall reveal the type of $o$ and then the neighbourhoods of vertices from $G_n(o)|_d$ one by one in breadth-first-search order and check equality with $H$. Let $w_1, w_2, \ldots$ be the vertices of type $B$ in breadth-first-search ordering and $C(w_i) := N(w_i) \setminus \text{parent}(w_i)$ the neighbourhood of $w_i$, except for the vertex from which $w_i$ was discovered (if $w_i \neq o$). Recall that the $cn$ vertices from $A_R$ each choose $k$ among the $n$ vertices of $B$ uniformly at random, so for $v \in B$, we have $|N(v) \cap A_R| \sim \text{Bin}(ckn, \frac{1}{n})$. However, when revealing $Y_i := |C(w_i) \cap A_R|$, then a constant number $\alpha_i$ of incidences of vertices from $A_R$ are already revealed and the full neighbourhoods of a constant number $\beta_i$ of vertices of type $B$ are already revealed. Thus, conditioned on $G_n(o)$ matching with $H$ until before $N(w_i)$ is revealed, we have $Y_i \sim \text{Bin}(ckn - \alpha_i, \frac{1}{n-\beta_i})$. A second complication is that $G_n(o)$ can contain cycles. Therefore, whenever we reveal the identity of a vertex of type $A_R$, we shall assume that it is not one of the vertices already seen, and when we reveal the identity of a vertex of type $B$ found as a child of a vertex of type $A_R$, we shall assume that its position in $\mathbb{Z}_n$ is not within distance $d$ of a vertex of type $B$ already seen. The probability of these events is clearly $1 - o(1)$, since only a constant number of vertices are forbidden. It is important to note that, since $H$ is a possible outcome for $T|_d$, all remaining aspects of $H$ and $G_n(o)$ coincide by construction (e.g., the degree of vertices of types $A_R$ and $A_C$). We get
\[
\Pr_{G_n, o \in V(G_n)}[G_n(o)|_d = H] = (1 - o(1)) \Pr[o \in t] \cdot \prod_{i=1}^{s} \Pr[Y_i = x_i \mid G_n(o) \text{ matches with } H \text{ up to } v_i].
\]
Since $\Pr[o \in t] = \Pr[\text{root}(T) \in t]$ by construction and due to the convergence of Binomial to Poisson random variables, we get $\lim_{n \to \infty} \Pr_{G_n, o}[G_n(o)|_d = H] = \Pr[T|_d = H]$ as desired.

6 \hspace{1em} \text{BELIEF PROPAGATION ON THE LIMITING TREE } T

Recall the definition and relevance of large allocations from Proposition 2 and consider the task of finding a large allocation $\mu$ for $G_n$. Imagine the vertices as agents in a parallel endeavour that proceeds in rounds and that is designed to yield information useful to construct $\mu$. In each round, every vertex sends a message to each of its neighbours. Since two messages are sent between two adjacent vertices $u$ and $v$—one in each direction—it is convenient to distinguish the directed edges $(u, v)$ and $(v, u)$, the message from $u$ to $v$ being sent along $(u, v)$ and vice versa. Along $e = (u, v)$ the message is a number $I_e \in [0, \min(\eta(u), \eta(v))]$. We interpret this as the vertex $u$ suggesting that $\mu((u, v))$ be set to $I_e$. To determine $I_e$, the vertex $u$ sums up the messages it received from its other neighbours in the previous round, obtaining a value $\xi$. If $\xi < \eta(u)$, then, assuming the suggestions of the neighbours of $u$ were all followed, $u$ would want $\mu(e) = \eta(u) - \xi$ to fully utilise its weight $\eta(u)$. Taking into account the weight of $v$, $u$ sends $I_e = [\eta(u) - \xi]_0^{\eta(u)}$, where $[x]_0^y := \max(i, \min(j, x))$ is our shorthand for the “clamp function” (which we also occasionally use for one-sided clamping, leaving out the upper or lower index). Let $P$ be the operator that takes an assignment $I : \bar{E}(G_n) \to \mathbb{N}_0$ of messages to directed edges and computes the messages $P(I) : \bar{E}(G_n) \to \mathbb{N}_0$ of the next round.

On finite trees, iterated application of $P$ can easily be seen to converge to a unique fixed point $I^*$ of $P$, regardless of the initial assignment of messages. From $I^*$, the size of the maximum allocation can be obtained by local computations. It is plausible but non-trivial that the asymptotic behaviour
of a largest allocation of $G_n$ is similarly connected to fixed points of $P$ on the local weak limit $T$ of $G_n$. The details are found in Reference [18], but we say a few words trying to give some intuition.

Let $(u, v)$ be an edge of $T$ where $v$ is closer to the root than $u$ and let $T_u$ be the subtree of $T$ containing $u$, $v$ and all descendants of $u$. If we apply $P$ to $T$ repeatedly (starting with, say, the all-zero message assignment $I ≡ 0$), then the message $I_{(u, v)}$ in later rounds will depend on ever larger parts of $T_u$ but nothing else. Assume there was a magical (measurable) function $f$ that finds, by looking at all of $T_u$, the message $f(T_u)$ that is sent along $(u, v)$ in some fixed point of $P$. In particular, if $u_1, \ldots, u_i$ are the children of $u$, locally, the fixed point equation is

$$f(T_u) = \left[ \eta(u) - \sum_{j=1}^{i} f(T_{u_j}) \right]_{0}^{n(v)}.$$ 

Assume we have yet to reveal anything about $T_u$ and only know the types $t_u$ and $t_v$ of $u$ and $v$. Then $I_{t_u→t_v} := f(T_u)$ is a random variable. The four possible combinations of types yield random variables $I_{A_C→B}, I_{A_K→B}, I_{B→A_C}, I_{B→A_R}$, which must fulfill certain distributional equations.

Consider for instance $ε = (u, v)$ with $u ∈ B$ and $v ∈ A_C$. On the one hand, the message $f(T_u)$ is distributed like $I_{B→A_C}$. On the other hand, looking one layer deeper, $u$ has children $u_1, \ldots, u_2$ of type $A_R$, with $Z ∼ Po(ck)$ as well as one child $a$ of type $A_C$. The messages $f(T_{u_1}), \ldots, f(T_{u_2})$ and $f(T_a)$ are independent (since the subtrees are independent) and distributed like $I_{A_K→B}$ and $I_{A_C→B}$, respectively, which implies

$$I_{B→A_C} \overset{d}{=} \left[ \ell - I_{A_C→B} - \sum_{j=1}^{Z} f(A_{R→B}) \right]_{0}^{\ell-1}, \quad (\ast 3)$$

where a superscript in parentheses indicates an independent copy of a random variable and $\overset{d}{=} \approx$ denotes equality in distribution. Leconte, Lelarge, and Massoulié [18] show that, remarkably, such distributional equations suffice to capture the asymptotics of maximum allocations. Readers interested in Reference [18] but deterred by the measure theory may find Section 2 of Reference [20] illuminating, which gives a high-level description of the argument for a simpler case. Applying Theorem 2.1 from Reference [18] to our family $(G_n)_{n \in \mathbb{N}}$ yields the following Lemma. We postpone technical details to Section 7.

**Lemma 6.1 (Special Case of Reference [18, Theorem 2.1]).**

$$\lim_{n \to \infty} \frac{M(G_n, c)}{n} = \inf \left\{ F(I_{A_C→B}, I_{B→A_C}, I_{A_K→B}, I_{B→A_R}) : c \right\} \quad \text{almost surely, where}$$

$$F(I_{A_C→B}, I_{B→A_C}, I_{A_K→B}, I_{B→A_R}) = E \left[ \left[ I_{A_C→B}^{(1)} + I_{B→A_C}^{(2)} \right]_{0}^{\ell-1} + c \cdot \left[ \sum_{j=1}^{k} I_{B→A_R}^{(j)} \right]_{0}^{1} \right] + \left[ \ell - \sum_{i=1,2} \left[ \ell - I_{A_C→B} - \sum_{j=1}^{Z} f(A_{R→B}) \right]_{0}^{\ell-1} - \sum_{i=1}^{Z} \left[ \ell - I_{A_C→B} - I_{A_K→B} - \sum_{j=1, j \neq i}^{Z} f(A_{R→B}) \right]_{0}^{\ell-1} \right],$$

and the infimum is taken over distributions of $I_{A_C→B}, I_{A_K→B}, I_{B→A_C}, I_{B→A_R}$, fulfilling

$$I_{A_C→B} \overset{d}{=} \ell - 1 - I_{B→A_C} \quad (\ast 1), \quad I_{B→A_C} \overset{d}{=} \left[ \ell - I_{A_C→B} - \sum_{j=1}^{Z} f(A_{R→B}) \right]_{0}^{\ell-1} \quad (\ast 3),$$

$$I_{A_K→B} \overset{d}{=} \left[ 1 - \sum_{j=1}^{k-1} f(A_{R→B}) \right]_{0}^{1} \quad (\ast 2), \quad I_{B→A_R} \overset{d}{=} \left[ \ell - I_{A_C→B} - I_{A_K→B} - \sum_{j=1}^{Z} f(A_{R→B}) \right]_{0}^{1} \quad (\ast 4).$$

As before, superscripts in parentheses indicate independent copies and $Z \sim Po(ck)$.
To appreciate the usefulness of Lemma 6.1, understanding its form is more important than understanding the significance of the individual terms.

If $X$ is a random variable on a finite set $D$, then the distribution of $X$ is captured by real numbers $(\Pr[X = i])_{i \in D}$ that sum to 1. In this sense, the four distributions of $I_{A_C \rightarrow B}$, $I_{A_R \rightarrow B}$, $I_{B \rightarrow A_C}$, $I_{B \rightarrow A_R}$ are given by numbers $\vec{\rho} \in [0, 1]^{|D|}$ such that satisfies $2$ for $\vec{\rho}$. We say $\vec{\rho} \in [0, 1]^{2\ell+2}$ is a solution to the system $(\star)$ if the four groups of numbers belonging to the same distribution each sum to 1 and if setting up the four random variables according to $\vec{\rho}$ satisfies $(\star 1)$, $(\star 2)$, $(\star 3)$, and $(\star 4)$.

If we treat $c$ as a variable instead of as a constant, then we obtain the relaxed system $(\star c)$ where solutions are pairs $(\vec{\rho}, c) \in [0, 1]^{2\ell+2} \times (0, \infty)$. The value $\vec{\rho}_{\text{triv}}$ that corresponds to

$$
1 = \Pr[I_{A_C \rightarrow B} = 0] = \Pr[I_{A_R \rightarrow B} = 0] = \Pr[I_{B \rightarrow A_C} = \ell - 1] = \Pr[I_{B \rightarrow A_R} = 1]
$$

easily checked to give rise to a solution $(\vec{\rho}_{\text{triv}}, c)$ of the relaxed system for any $c > 0$; we call such a solution trivial. Evaluating $F$ for a trivial solution yields $\ell - 1 + c$ so Lemma 6.1 implies the trivial assertion $\lim_{n \to \infty} \frac{M(G_n)}{n} \leq \ell - 1 + c$ for all $c > 0$.

We now give a "nice" characterisation of the space of non-trivial solutions for $(\star c)$.

**Lemma 6.2.** For any $k, \ell \geq 2$, there is a bijective mapping $\lambda \mapsto (\vec{\rho}_\lambda, c_\lambda)$ from $\mathbb{N}_0$ to the set of non-trivial solutions for $(\star c)$.

Moreover, (each component of) this mapping is an explicit real analytic function.

**Proof.** We have $I_{A_R \rightarrow B} \in [0, 1]$ and thus $I_{A_R \rightarrow B} \sim \text{Bernoulli}(q)$ for some $q \in [0, 1]$. Therefore $Y := \sum_{j=1}^{|D|-1} I_{A_R \rightarrow B}$ is the sum of $Z$ independent Bernoulli random variables, where $Z \sim \text{Po}(ck)$. It is well known that such a "thinned out" Poisson distribution is again Poisson distributed, and we have $Y \sim \text{Po}(\lambda)$ for $\lambda = ck$. Since $q = 0$ only occurs in trivial solutions, each non-trivial solution to $(\star c)$ has such a parameter $\lambda > 0$. We will now show that, conversely, $\lambda$ uniquely determines this solution. From $(\star 1)$ and $(\star 3)$, we obtain

$$
I_{B \rightarrow A_C} \overset{d}{=} \left[ \ell - (\ell - 1 - I_{B \rightarrow A_C}) - Y \right]^{\ell-1}_0 = \left[ I_{B \rightarrow A_C} + 1 - Y \right]^{\ell-1}_0.
$$

With $p_i := \Pr[I_{B \rightarrow A_C} = i]$ for $0 \leq i \leq \ell - 1$, we can write this equation in matrix form as

$$
\begin{pmatrix}
p_0 \\
p_1 \\
p_2 \\
p_{\ell-1}
\end{pmatrix} = e^{-\lambda} \begin{pmatrix}
1 & \lambda & \lambda^2/2 & \ldots & \lambda^{\ell-1} \\
1 & \lambda & \lambda^2/2 & \ldots & \lambda^{\ell-1} \\
1 & \lambda & \lambda^2/2 & \ldots & \lambda^{\ell-1} \\
1 & \lambda & \lambda^2/2 & \ldots & \lambda^{\ell-1}
\end{pmatrix} \begin{pmatrix}
p_0 \\
p_1 \\
p_2 \\
p_{\ell-1}
\end{pmatrix}.
$$

This uses $\Pr[Y = j] = e^{-\lambda} \lambda^j / j!$ for $j \in \mathbb{N}_0$. Each "*" is such that the columns of the matrix sum to $(e^\lambda, \ldots, e^\lambda)$, which is implicit in the fact that we deal with distributions. The unique solution for a fixed $\lambda$ can be found by using the equations from bottom to top to express $p_{\ell-2}, p_{\ell-3}, \ldots, p_0$ in terms of $p_{\ell-1}$ and then choosing $p_{\ell-1}$ such that the probabilities sum to 1. This yields a closed expression $p_{\ell-1} = p_{\ell-1}(\lambda)$ for $0 \leq i \leq \ell - 1$.

Using first $(\star 1)$, then $(\star 4)$ (with the definition of $Y$), and finally $(\star 2)$ the distributions of $I_{A_C \rightarrow B}$, $I_{B \rightarrow A_R}$ and $I_{A_R \rightarrow B}$ fall into place, completing the unique solution candidate $\vec{\rho}_\lambda$. The only loose end is the definition of $\lambda$, which gives a final equation: If $q(\lambda) = \Pr[I_{A_R \rightarrow B} = 1]$ is the value we computed after choosing $\lambda$, then we need $\lambda = c k \cdot q(\lambda)$, which uniquely determines a value $c_\lambda = \frac{\lambda}{k \cdot q(\lambda)}$ (it is easy to check that $\lambda > 0$ guarantees $q(\lambda) > 0$). Thus, $(\vec{\rho}_\lambda, c_\lambda)$ is the unique solution with parameter $\lambda$. Retracing our steps it is easy to verify that we only composed real analytic functions. □
With the parametrisation of the solutions of \((\star c)\), we can, with a slight stretch of notation, rewrite Lemma 6.1. For any \(c > 0\), we have
\[
\lim_{n \to \infty} \frac{M(G_{n,c}, cn)}{n} = \inf\{\{F(\tilde{\rho}_\lambda, c_\lambda) \mid \lambda \in (0, \infty), c_\lambda = c\} \cup \{\ell - 1 + c\}\} \quad \text{almost surely.} \quad (11)
\]
We now define the value \(y_{k,\ell}\) and by proving Proposition 4 demonstrate its significance:
\[
y_{k,\ell} := \inf_{\lambda > 0} \{c_\lambda \mid F(\tilde{\rho}_\lambda, c_\lambda) < \ell - 1 + c_\lambda\}. \quad (12)
\]
\section*{Proof of Proposition 4.}
- **Case** \(c < y_{k,\ell}\). By definition of \(y_{k,\ell}\) there is no parameter \(\lambda\) with \(c_\lambda = c\) and \(F(\tilde{\rho}_\lambda, c_\lambda) < \ell - 1 + c\). Thus, Equation (11) implies \(\lim_{n \to \infty} \frac{1}{n} M(G_{n,cn}) = \ell - 1 + c\) almost surely.
- **Case** \(c > y_{k,\ell}\). By definition of \(y_{k,\ell}\), for \(c = y_{k,\ell} + \varepsilon\) there is some \(\lambda\) with \(c_\lambda \in [y_{k,\ell} + c, \ell - 1 + c_\lambda - \varepsilon'\) for some \(\varepsilon' > 0\). This implies \(\lim_{n \to \infty} \frac{1}{n} M(G_{n,cn}) \leq \ell - 1 + c_\lambda - \varepsilon'\) almost surely. Since \(G_{n,cn}\) can be obtained from \(G_{n,c_\lambda n}\) by adding \((c - c_\lambda)\) vertices of weight 1 with random connections, and this can increase the size of a maximum allocation by at most \((c - c_\lambda)n\), we also have \(\lim_{n \to \infty} \frac{1}{n} M(G_{n,cn}) \leq \ell - 1 + c - \varepsilon'\) almost surely. \(\square\)

\section{Proof of Lemma 6.1 Using a Theorem by Lelarge et al.}

On a technical level, Lemma 6.1 is obtained by mechanically plugging the right distributions into Theorem 7.1 (see below), which is imported from Reference [18]. The details can be dizzying, however. This is partly due to differences in notation but also due to our use of edge weights (which are supported in Reference [18]) to simulate vertex types (which are not supported in Reference [18]).

We must establish some context. Lelarge et al. are interested in the maximum size \(M(G_{n},c)\) of an allocation, where \(G_{n}\) is a bipartite graph with bipartitions \(A = A_{n}\) and \(B = B_{n}\) and weighted vertices as well as weighted edges. In addition to respecting vertex weights as before, an allocation \(\mu : E(G_{n}) \to N_{0}\) must also respect edge weights, i.e., \(\mu(e)\) must not exceed \(e\)'s weight. If the sequence \((G_{n})_{n \in \mathbb{N}}\) converges locally weakly, then the most natural candidate for the limit is the bipartite Galton-Watson tree \(T_{\Phi^{A},\Phi^{B}}\) defined as follows. The distributions \(\Phi^{A}\) and \(\Phi^{B}\) describe the incidences of a random vertex from \(A\) and \(B\) in \(G_{n}\) as \(n \to \infty\). A sample \((D^{A},W^{A},\{C_{1}^{A},\ldots,C_{D^{A}}^{A}\})\) \(\sim \Phi^{A}\) indicates a degree of \(D^{A}\), a weight of \(W^{A}\) and a multiset \(\{C_{1}^{A},\ldots,C_{D^{A}}^{A}\}\) of weights on the incident edges. The same applies to \((D^{B},W^{B},\{C_{1}^{B},\ldots,C_{D^{B}}^{B}\})\) \(\sim \Phi^{B}\). As for the distribution of \(T_{\Phi^{A},\Phi^{B}}\), the root is from \(A\) or \(B\) with probability \(\mathbb{E}[D^{A}]\) and \(\mathbb{E}[D^{B}]\) and \(\mathbb{E}[D^{A}]\) and \(\mathbb{E}[D^{B}]\), respectively. Say it is from \(A\). We then take a sample from \(\Phi^{A}\) and attach \(D^{A}\) "dangling" edges with weights \(\{C_{1}^{A},\ldots,C_{D^{A}}^{A}\}\) to the root. To each dangling edge of weight \(c\), we attach a new vertex from \(B\) with weight \(W^{B}\), and to that vertex, we attach \(D^{B}\) new dangling edges with weights \(\{C_{1}^{B},\ldots,C_{D^{B}}^{B}\}\), where \((D^{B},W^{B},\{C_{1}^{B},\ldots,C_{D^{B}}^{B}\})\) \(\sim \Phi^{B}(\cdot|c)\) comes from an edge-biased distribution; see below. A corresponding distribution \(\Phi^{B}(\cdot|c)\) also exist for \(A\)-vertices and each possible weight \(c\). The (possibly infinite) random tree \(T_{\Phi^{A},\Phi^{B}}\) is obtained by continuing to attach vertices to dangling edges, alternating between using distributions of the form \(\Phi^{A}(\cdot|c)\) and \(\Phi^{B}(\cdot|c)\). The intuition for \(\Phi^{A}(\cdot|c)\) (and similarly for \(\Phi^{B}(\cdot|c)\)) is this: When sampling a uniformly random edge \(e\) of weight \(c\) from \(G_{n}\), then \(\Phi^{A}(\cdot|c)\) is the distribution of the set of edges incident to \(e\) on the \(A\)-side, in the limit as \(n \to \infty\). Formally, the probability of any outcome \((D^{A},W^{A},\hat{C})\) under \(\Phi^{A}(\cdot|c)\) is the probability of the outcome \((D^{A} + 1, W^{A}, \hat{C} + \{c\})\) under \(\Phi^{A}\), except multiplied by the number of occurrences of \(c\) in \(\hat{C} + \{c\}\) and a normalisation factor.

We can now state Theorem 2.1 from Reference [18] almost verbatim and derive Lemma 6.1 from it.
Theorem 7.1 (Maximum Allocation for Bipartite Galton-Watson Limits; from Reference [18]). Assume \((G_n)_{n \in \mathbb{N}}\) converges locally weakly to \(T_{\Phi^A,\Phi^B}\). Provided that \(\mathbb{E}[W^A]\) and \(\mathbb{E}[W^B]\) are finite, the limit \(M(\Phi^A, \Phi^B) := \lim_{n \rightarrow \infty} M(G_n)/|A_n|\) exists and equals

\[
\inf \left\{ \mathbb{E}\left[ \min \left( W^A, \sum_{i=1}^{D^A} X_i(C_i^A) \right) \right] + \frac{\mathbb{E}[D^A]}{\mathbb{E}[D^B]} \mathbb{E}\left[ \left( W^B - \sum_{i=1}^{D^B} Y_i(C_i^B) \right)_0^c \right] \right\} 
\cdot \left( W^B < \sum_{i=1}^{D^B} C_i^B \right),
\]

where for all \(i\), \((X_i(c), Y_i(c))_{c \in \mathbb{N}}\) is an independent copy of \((X(c), Y(c))_{c \in \mathbb{N}}\) and the infimum is taken over all distributions for \((X(c), Y(c))_{c \in \mathbb{N}}\) satisfying the recursive distributional equations

\[
Y(c) = \left\{ \left( \hat{W}^A - \sum_{i=1}^{D^A} X_i(C_i^A) \right)_0^c \mid c \right\}, \quad X(c) = \left\{ \left( \hat{W}^B - \sum_{i=1}^{D^B} Y_i(C_i^B) \right)_0^c \mid c \right\}.
\]

Proof of Lemma 6.1 using Theorem 7.1. In Proposition 3, we have proved that our sequence \((G_{n,c})_{n \in \mathbb{N}}\) of graphs almost surely has local weak limit \(T_{\Phi^A,\Phi^B}\) as defined in Step 3 of Section 3. But does \(T\) have the form required by Theorem 7.1? An immediate problem is that \(T\) distinguishes two types \(A_C\) and \(A_R\) of \(A\)-vertices that affect the number of children spawned by the vertex. At first \(\Phi^A\) does not allow such a distinction. It is a lucky coincidence that we can use edge weights, which we do not need for their original purpose, to fix this. We pick two numbers \(C \neq R\) both large enough to never impose a real constraint on allocations when used as edge weights in \(G_n\) and \(T\), e.g., \(C = \ell\) and \(R = \ell + 1\). The edges incident to vertices of type \(A_C\) and \(A_R\) receive weight \(C\) and \(R\), respectively, meaning the type of an \(A\)-vertex is encoded in each of its incident edges. With \(T\)modified in this way, it is not hard to check that \(T \overset{\text{d}}{=} T_{\Phi^A,\Phi^B}\) for \(\Phi^A\) and \(\Phi^B\) as follows:

\[
\Phi^A : \begin{cases} 
\text{with probability } \frac{1}{1+c} & D^A = 2, \quad W^A = \ell - 1, \quad (C_1, C_2) = (C, C), \\
\text{with probability } \frac{c}{1+c} & D^A = k, \quad W^A = 1, \quad (C_1, \ldots, C_k) = (R, \ldots, R),
\end{cases}
\]

\[
\Phi^B : D^B = 2 + Z \text{ for } Z \sim \text{Po}(ck), \quad W^B = \ell, \quad (C_1^B, \ldots, C_{2+Z}^B) = (C, C, R, \ldots, R).
\]

The edge-biased distributions \(\tilde{\Phi}^A(\cdot|C)\) and \(\tilde{\Phi}^A(\cdot|R)\) are degenerate, assigning probability 1 to the outcomes \((1, \ell - 1, (C))\) and \((k - 1, 1, (R, \ldots, R))\), respectively. Moreover, \((\tilde{D}^B, \tilde{W}^B, C) \sim \tilde{\Phi}^B(\cdot|C)\) arises by removing one edge of weight \(C\) after sampling from \(\Phi^B\), i.e., we have \((\tilde{D}^B + 1, \tilde{W}^B, C + \{C\}) \sim \Phi^B\). Last, we have \(\Phi^B(\cdot|R) = \Phi^B\).

We now specialise the RDEs from Theorem 7.1 for our case. From now on let \(Z \sim \text{Po}(ck)\).

\[
Y(C) \overset{\text{d}}{=} \left[ \ell - 1 - Y(C) \right]_0^C \quad (\star \star \star 1), \quad X(C) \overset{\text{d}}{=} \left[ \ell - Y(C) - \sum_{i=1}^{Z} Y_i(R) \right]_0^C \quad (\star \star \star 3),
\]

\[
Y(R) \overset{\text{d}}{=} \left[ 1 - \sum_{i=1}^{k-1} X_i(R) \right]_0^R \quad (\star \star \star 2), \quad X(R) \overset{\text{d}}{=} \left[ \ell - Y_1(C) - Y_2(C) - \sum_{i=1}^{Z} Y_i(R) \right]_0^R \quad (\star \star \star 4).
\]

These equations directly correspond to \((\star 1)\) to \((\star 4)\) in Lemma 6.1 after renaming \((Y(C) \mapsto I_{A_C \rightarrow B}, Y(R) \mapsto I_{A_R \rightarrow B}, X(C) \mapsto I_B \mapsto A_C, X(R) \mapsto I_B \mapsto A_R)\), except that we made some of the clamping...
“[..]” more natural in ways that are inconsequential in the context of Lemma 6.1. Using the same substitutions, we can now work out the terms contributing to \( M(\Phi^A, \Phi^B) \) as follows:

\[
\begin{align*}
\mathbb{E} \left[ \min(W^A, \sum_{i=1}^{D^A} X_i(C_i^A)) \right] &= \frac{1}{1 + c} \mathbb{E} \left[ \min(\ell - 1, X_1(C) + X_2(C)) \right] + \frac{c}{1 + c} \mathbb{E} \left[ \min(1, \sum_{i=1}^{k} X_i(R)) \right] \\
&= \frac{1}{1 + c} \mathbb{E} \left[ [I_{B \to A_c}^{(1)} + I_{B \to A_c}^{(2)}]^{\ell-1} \right] + \frac{c}{1 + c} \mathbb{E} \left[ \sum_{i=1}^{k} I_{B \to A_R}^{(i)} \right], \\
\mathbb{E} [D^A] &= \frac{\ell + 2 + \frac{c}{1 + c} \cdot k}{2 + c} = \frac{1}{1 + c}, \\
\mathbb{E} \left[ W^B - \sum_{j=1}^{D^B} Y_j(C_j^B) \right]_0 &= 1 \left( W^B < \sum_{i=1}^{D^B} C_i^B \right)_0 \\
&= \mathbb{E} \left[ \ell - \sum_{i=1}^{Z} \ell - Y_i(C) - \sum_{j=1}^{Z} Y_j(R) \right]_0 - \sum_{i=1}^{Z} \left( \ell - Y_i(C) - Y_j(C) - \sum_{j=1}^{Z} Y_j(R) \right)_0 \\
&= \mathbb{E} \left[ \ell - \sum_{i=1}^{Z} \ell - I_{A_c \to B}^{(i)} - \sum_{j=1}^{Z} I_{A_R \to B}^{(j)} \right]_0 - \sum_{i=1}^{Z} \left( \ell - I_{A_c \to B}^{(1)} - I_{A_R \to B}^{(2)} - \sum_{j=1}^{Z} I_{A_R \to B}^{(j)} \right)_0.
\end{align*}
\]

From these terms, we can assemble a formula for \( M(\Phi^A, \Phi^B) = \lim_{n \to \infty} \frac{M(G_n)}{\alpha_n} = \frac{1}{1 + c} \lim_{n \to \infty} \frac{M(G_n)}{n} \). Multiplying with \( 1 + c \) yields the exact formula from Lemma 6.1, concluding the proof.

\[ \square \]

**8 CLOSING THE GAP—PROOF OF THE MAIN THEOREM**

The key ingredient still missing to prove the main theorem is Lemma 3.1, stated on page 8.

**Proof of Lemma 3.1.** Call a set \( X \subset \mathbb{Z}_n = V(\hat{W}_n) \) a bad set if it induces (hyper-)edges of total weight \( \ell |X| \) or more. We now consider all possible sizes \( t \) of \( X \) and each possible number \( at \) \((0 < at \leq 1)\) of contiguous segments of \( X \) separately, using the first moment method to bound the probability that a bad set \( X \) with such parameters exists, later summing over all \( t \) and \( \alpha \).

For now, let \( t \) and \( \alpha \) be fixed and write \( X \) as the union of non-empty, non-touching\(^4\) intervals \( X = X_1 \cup X_2 \cup \ldots \cup X_{at} \) arranged on the cycle \( \mathbb{Z}_n \) in canonical ordering and with \( X_1 \) being the interval containing \( \min X \). We write the complement \( \mathbb{Z}_n - X = Y_1 \cup Y_2 \cup \ldots \cup Y_{at} \) in a similar way. It is almost possible to reconstruct \( X \) from the sets \( \{x_1, \ldots, x_{at}\} \) and \( \{y_1, \ldots, y_{at}\} \) where \( x_i := |X_1 \cup \ldots \cup X_i| \) and \( y_i := |Y_1 \cup \ldots \cup Y_i| \), we just do not know where \( X_1 \) starts. To fix this, we exploit that \( x_{at} \) and \( y_{at} \) are always \( t \) and \( n - t \), respectively, and do not really encode information. In the case \( 0 \in X \), we set \( x'_{at} := \max X_1 \in \{0, 0, \ldots, x_1\} \) and \( y'_{at} := y_{at} \); if \( 0 \notin X \), we set \( y'_{at} := \max Y_1 \in \{0, \ldots, y_1\} \) and \( x'_{at} := x_{at} \). The sets \( \{x_1, \ldots, x_{at-1}, x'_{at}\} \subseteq \{0, \ldots, t\} \) and \( \{y_1, \ldots, y_{at-1}, y'_{at}\} \subseteq \{0, \ldots, n - t\} \) now uniquely identify \( X \), meaning there are at most \( \binom{t+1}{at} \binom{n-t+1}{at} \) choices for \( X \). No matter the choice, \( X \) induces helper edges of weight precisely \((\ell - 1)(t - at)\), since for each \( x \in X \) the edge \((x, x + 1)\) is induced, except if \( x \) is the right endpoint of one of the \( at \) intervals. In order for \( X \) to induce a total weight of \( \ell t \) or more another

\[ \ell t - (\ell - 1)(t - at) = t + (\ell - 1)at \geq t + at \]

\(^4\)Two intervals touch if their union is an interval, i.e., if there is no gap in between them.
ordinary edges (of weight 1) need to be induced. There are \( \binom{cn}{t+at} \) ways to choose such a set of edges and each edge has all endpoints in \( X \) with probability \( \left( \frac{t}{n} \right)^k \leq \left( \frac{1}{n} \right)^2 \).

Together, we obtain the following upper bound on the probability that a bad set of size \( t \) with \( at \) contiguous regions exists:

\[
\sum_{t+1}^{t+1} \left( \binom{cn}{t+at} \right) \left( \frac{t}{n} \right)^{2(t+at)} \leq 2^{t+1} \left( \frac{n}{at} \right)^t \left( \frac{t}{n} \right)^{2(t+at)}
\]

choices for \( X \), choices for edges

\[
\leq 2^{t+1} \left( \frac{ne}{at} \right)^t \left( \frac{t}{n} \right)^{2(t+at)} \leq 2 \left( \frac{ne}{at} \right)^t \left( \frac{t}{n} \right)^{2(1+\alpha)}
\]

We used that \( \alpha^2 \) is bounded (has limit 1) for \( \alpha \to 0 \). The resulting term is \( o(1) \) for \( t = 1, 2, 3 \) (and only 1, 2 or 3 choices for \( \alpha \) are possible). For \( 4 \leq t \leq \sqrt{n} \), it is

\[
2 \left( \frac{t}{n} \right)^t \leq 2 \left( \frac{C}{\sqrt{n}} \right)^t \leq 2 \left( \frac{C}{\sqrt{n}} \right)^4 = O(n^{-2}).
\]

Summing over all combinations of \( O(\sqrt{n} \cdot \sqrt{n}) \) choices for \( \alpha \) and \( t \), we get a sum of \( O(n^{-1}) \). For \( \sqrt{n} \leq t \leq \delta n \) with \( \delta = \frac{1}{2\sqrt{c}} \), we get

\[
2 \left( \frac{t}{n} \right)^t \leq 2 \left( \frac{1}{2} \right)^t \leq 2 \cdot 2^{-\sqrt{n}},
\]

which is clearly \( o(1) \) even if we sum over all \( O(n^2) \) combinations for choosing \( t \) and \( \alpha \).

**Proof of the Main Theorem.** It suffices to show that \( \gamma_{k,\ell} \) is the threshold for the event \( \{ M(G_n) = n(\ell - 1 + c) \} \), since by Propositions 1 and 2 this event coincides with the events that \( W_n \) and \( W_n \) are orientable.

- \( c > \gamma_{k,\ell} \). If \( c = \gamma_{k,\ell} + \epsilon \) for \( \epsilon > 0 \), then by Proposition 4, we have \( \lim_n \frac{M(G_{n,\epsilon})}{n} = \ell - 1 + c - \epsilon' \) almost surely for some \( \epsilon' > 0 \). This clearly implies that \( M(G_{n,\epsilon}) < n(\ell - 1 + c) \) whp.

- \( c < \gamma_{k,\ell} \). Let \( c = \gamma_{k,\ell} - \epsilon \) for some \( \epsilon > 0 \) and define \( c' := \gamma_{k,\ell} - \frac{\epsilon}{2} \). We generate \( G_{n,\epsilon} \) from \( G_{n,c'} \) by removing \( \frac{\epsilon}{2} n \) vertices from \( A_R \). The idea is to derive orientability of \( G_{n,\epsilon} \) from Lemma 3.1 and “almost-orientability” of \( G_{n,c'} \).

More precisely, let \( G^{(0)} := G_{n,c'} = (A = A_C \cup A_R, B, e'') \) and let \( G^{(i+1)} \) be obtained from \( G^{(i)} \) by removing a vertex from \( A_R \cap V(G^{(0)}) \) uniformly at random, \( 0 \leq i < \frac{\epsilon}{2} n \). By Proposition 4, we have \( \lim_n \frac{M(G_{n,c'})}{n} = \ell - 1 + \epsilon' \) almost surely, which implies \( M(G_{n,c'}) = n(\ell - 1 + c - 0(n)) = \eta(A) - o(n) \) whp. For any subgraph \( G \) of \( G^{(0)} \), define \( gap(G) := \eta(A \cap V(G)) - M(G) \), which measures how far \( G \) is away from being orientable. This ensures \( gap(G^{(i)}) = o(n) \) whp as well as \( gap(G^{(i+1)}) \) \in \( \{ gap(G^{(i)}), gap(G^{(i)}) - 1 \} \). We say a vertex \( a \in V(G^{(i)}) \cap A_R \) is good for \( G^{(i)} \) if \( gap(G^{(i)}) - a = gap(G^{(i)}) - 1 \).

Assume \( gap(G^{(i)}) > 0 \). We now show that \( \Theta(n) \) vertices are good for \( G^{(i)} \) whp. Let a be one vertex of \( G^{(i)} \) that is not saturated in a maximum allocation \( \mu \) of \( G^{(i)} \). Let \( X \subseteq B \) and \( Y \subseteq A \) be the vertices from \( B \) and \( A \) reachable from \( a \) via an alternating path, i.e., a path of the form \( (a = a_1, b_1, a_2, b_2, \ldots) \) such that \( \mu(b_j, a_{j+1}) > 0 \) for all \( j \).

It is easy to check that all vertices from \( X \) are saturated in \( \mu \)(otherwise \( \mu \) could be increased), \( Y \) exceeds \( X \) in total weight and every vertex from \( Y \cap A_R \) is good for \( G^{(i)} \). Moreover, when...
viewed as a subset of $V(\hat{W}, c\ell)$, $X$ induces at least $Y$. Discounting the low probability event that Lemma 3.1 does not apply to $\hat{W}$, we conclude $|X| > \delta n$, and taking into account $|Y \cap A_C| \leq |X|$ (clear from definition of $A_C$) together with $(\ell - 1)|Y \cap A_C| + |Y \cap A_R| = \eta(Y) > \eta(X) = \ell |X|$, we obtain $|Y \cap A_R| > \delta n$. This means that whp on the way from $G^0$ to $G(\frac{\ell}{n})$, we start with a gap of $o(n)$ and have up to $\frac{\ell}{n} = \Omega(n)$ chances to reduce a non-zero gap by 1, namely, by choosing a good vertex for removal, and the probability is at least $\delta = \Omega(1)$ every time. Now simple Chernoff bounds imply that the gap vanishes whp meaning $G(\frac{\ell}{n})$ is orientable whp. Since the distributions of $G_n$ and $G(\frac{\ell}{n})$ coincide, we are done. \hfill \Box

9 NUMERICAL APPROXIMATIONS OF THE THRESHOLDS
Rewriting the definition of $\gamma_{k, \ell}$ in Equation (12) in the form promised in the Introduction, we get

$$\gamma_{k, \ell} = \inf_{\lambda > 0} \{ f_{k, \ell}(\lambda) \mid g_{k, \ell}(\lambda) < 0 \},$$

where $f_{k, \ell}(\lambda) = c_3 \lambda$ and $g_{k, \ell}(\lambda) = F(\tilde{\rho}_3, c_3) - \ell + 1 - c_4$. We give the plots of $f_{2, 2}$ and $g_{2, 2}$ in Figure 5.

It looks as though $\{ \lambda \mid g_{2, 2}(\lambda) < 0 \}$ is an interval $(\lambda^* \approx 1.74, \infty)$ on which $f_{2, 2}$ is monotonically increasing, meaning $\gamma_{2, 2} = f_{2, 2}(\lambda^*) \approx 0.965$. Here is a semi-rigorous argument that properly done plots cannot be misleading: Regardless of $k, \ell \geq 2$ it is fairly easy to see that $c_3 = \Omega(\lambda^{-k-1})$ for $\lambda \to 0$ and $c_4 = \Omega(\lambda)$ for $\lambda \to \lambda_1$. In particular, for fixed $k, \ell$ it is easy to obtain bounds $0 < \lambda_0 < \lambda_1 < \infty$ such that for $\lambda \in (0, \lambda_0) \cup (\lambda_1, \infty)$, we can guarantee $f_{k, \ell}(\lambda) > 1$. Because $\gamma_{k, \ell} < 1$, only the interval $[\lambda_0, \lambda_1]$ can be relevant. Being real analytic, the functions $f_{k, \ell}$ and $g_{k, \ell}$ have bounded first and second derivatives on $[\lambda_0, \lambda_1]$, which vindicates plots of sufficient resolution: There cannot be unexpected zeroes in between sampled positions and what looks strictly monotonic in the plot actually is. So, starting with a golden ratio search close to the apparent root of $g_{2, 2}$, we are guaranteed to find $\lambda^*$ and $\gamma_{2, 2} = f_{2, 2}(\lambda^*)$. This can be made formal.

While handling it this way saves us the trouble of having to deal with unwieldy functions, our lack of analytical insight means we have to consider each pair $(k, \ell)$ separately to make sure that $f_{k, \ell}$ and $g_{k, \ell}$ do not exhibit a qualitatively different behaviour. We did this for $\ell \leq 4$ and $k \leq 7$, i.e., for the values we provided in Table 1.

10 SPEED OF CONVERGENCE AND PRACTICAL TABLE SIZES
It is natural to wonder to what degree the asymptotic results of this article predict the behaviour of hash tables for realistic (large-ish but fixed) values of $n$, say a hash table of size $n = 10^5$. Formally,
Fig. 6. Approximate probabilities for $\hat{W}_{n,cn}^{k,\ell}$ to not be orientable depending on $c$. From left to right, the plots correspond to $(k, \ell) \in \{(2, 2), (2, 3), (3, 2)\}$. In each plot the three curves correspond to $n \in \{10^4, 10^5, 10^6\}$, with curves for larger $n$ visibly getting closer to the step function that we know is the limit for $n \to \infty$. Details are given in the text.

for $k, \ell \geq 2$ and $t \in [0, 1]$, we define the functions

$$p_n^{k,\ell}(c) := \Pr[\hat{W}_{n,cn}^{k,\ell} \text{ is not orientable}] \quad \text{and} \quad \text{step}_t(c) = \begin{cases} 0 & \text{if } c < t, \\ \frac{1}{2} & \text{if } c = t, \\ 1 & \text{if } c > t. \end{cases}$$

This article shows that $p_n^{k,\ell} : [0, 1] \to [0, 1]$ converges point-wise to $\text{step}_{\gamma_k,\ell}$, except, possibly, at the point of the threshold $c = \gamma_k,\ell$ itself. To give an idea of the speed of this convergence, we plotted approximations of $p_n^{k,\ell}$ for $n \in \{10^4, 10^5, 10^6\}$ and $(k, \ell) \in \{(2, 2), (2, 3), (3, 2)\}$ in Figure 6.

To obtain the approximation of $p_n^{k,\ell}$, we carried out 1,000 trials. In each trial, a graph of the form $\hat{W}_{n,m}^{k,\ell}$ was generated by adding random\(^5\) edges one by one until the graph was not orientable any more. The number of edges $m$ where orientability breaks down corresponds to the load $c = \frac{m}{n}$. As an estimate for $p_n^{k,\ell}(c)$, we take the fraction of the 1,000 trials where orientability broke at a value less than $c$.

The plots suggest that, at the threshold, $p_n^{k,\ell}$ assumes a value of $\frac{1}{2}$ and has a slope proportional to $\sqrt{n}$.

11 LINEAR TIME CONSTRUCTION OF AN ORIENTATION?

Constructing a placement of objects in a hash table is not the focus of this article, but we provide experimental support for the following conjecture:

**Conjecture 1.** Let $k \geq 2$ and $\ell \geq 2$ and $c = \gamma_k,\ell - \varepsilon$ for some $\varepsilon > 0$. An orientation of $\hat{W}_{n,cn}^{k,\ell}$ or, equivalently, a placement of $cn$ objects into $n$ table cells for $k$-ary cuckoo hashing with windows of size $\ell$ can be constructed in time $O(C(\varepsilon)n)$ whp using the adapted LSA Algorithm explained below. Here, $C(\varepsilon)$ does not depend on $n$.

Algorithms to orient certain random graphs have been known for a while, for instance, the *selfless algorithm* analysed by Cain, Sanders, and Wormald [3] or an algorithm by Fernholz and Ramachandran [7] involving so-called *excess degree reduction*. While a generalisation of the selfless algorithm has been suggested by Dietzfelbinger et al. [4], the algorithm that seemed easiest to adapt to our particular hypergraph setting is the LSA algorithm by Khosla [15] (with improved analysis in Reference [16]).

**The Algorithm.** Following Khosla’s terminology, we describe the task of orienting $\hat{W}_{n,cn}^{k,\ell}$ as a problem of placing balls into bins. There are $n$ bins of capacity $\ell$ arranged in a circle and for each

\(^5\)Using Pseudo-Random numbers produced by the mt19937 implementation of the c++ standard library.
pair of adjacent bins there are \((\ell-1)\) helper balls that may be placed into one of those two bins. Moreover there are ordinary balls, each of which has \(k\) random bins it may be placed into.

We start with all helper balls placed in their “left” option, in particular, all bins have room for just one more ball. Now the ordinary balls are placed one by one. We maintain a label for each bin. All labels are natural numbers, initially zero. We place a ball simply by putting it in the admissible bin with least label. If placing a ball results in an overloaded bin \(b\), then one ball must be evicted from \(b\) and inserted again. The ball to be evicted is chosen to have, among all balls in \(b\), an alternative bin of least label.

Whenever the content of a bin \(b\) changed (after insertion or insertion + eviction), its label is updated. Note that \(b\) is necessarily full. For each of the \(\ell\) balls currently placed in \(b\), all alternate bins are considered. The updated label of \(b\) is one more than the least label among these bins.

**Analysis.** Labels can be thought of as lower bounds on the distance of a bin to the closest non-full bin in the directed graph where bins are vertices and an edge from \(b_1\) to \(b_2\) indicates that a ball in \(b_1\) has \(b_2\) as an alternative bin. It is fairly easy to see that this algorithm finds a placement in quadratic time whenever a placement exists. To show that running time is linear whp, it suffices to show that the running time is linear in the sum of all labels in the end and that the sum of the aforementioned distances is linear whp. We do not attempt a proof here, although we expect it to be possible with Khosla’s techniques.

**Experiments.** The results from Figure 7 suggest that the expected number of evictions per insertion is bounded by a constant as long as the load \(c\) is bounded away from the threshold. Eviction counts sharply increase close to the threshold.

### 12 CONCLUSION AND OUTLOOK

We established a method to determine load thresholds \(\gamma_{k,\ell}\) for \(k\)-ary cuckoo hashing with (unaligned) windows of size \(\ell\). In particular, we resolved the cases with \(k = 2\) left open in References [6, 19], confirming corresponding experimental results by rigorous analysis.

The following four questions may be worthwhile starting points for further research.

**Is there more in this method?** It is conceivable that there is an insightful simplification of Lemma 6.1 that yields a less unwieldy characterisation of \(\gamma_{k,\ell}\). We also suspect that the threshold
for the appearance of the $(\ell + 1)$-core of $\hat{W}_n$ can be identified with some additional work (for cores see, e.g., References [14, 22]). This threshold is of interest because it is the point where the simple peeling algorithm to compute an orientation of $\hat{W}_n$ breaks down.

Can we prove efficient insertion? Given our experiments concerning the performance of Khosla’s LSA algorithm for inserting elements in our hashing scheme, it seems likely that its running time is linear, see conjecture 1 in Section 11. But one could also consider approaches that do not insert elements one by one but build a hash table of load $c = y_{k,\ell} - \varepsilon$ given all elements at once. Something in the spirit of the selfless algorithm [3] or excess degree reduction [4] may offer linear running time with no performance degradation as $\varepsilon$ gets smaller, at least for $k = 2$.

How good is it in practice? This article does not address the competitiveness of our hashing scheme in realistic practical settings. The fact that windows give higher thresholds than (aligned) blocks for the same parameter $\ell$ may just mean that the “best” $\ell$ for a particular use case is lower, not precluding the possibility that the associated performance benefit is outweighed by other effects. Reference [6] provide a few experiments in their appendix suggesting slight advantages for windows in the case of unsuccessful searches and slight disadvantages for successful searches and insert operations, in one very particular setup with $k = 2$. Further research could take into account precise knowledge of cache effects on modern machines, possibly using a mixed approach, respecting alignment only insofar as it is favoured by the caches. Ideas from Porat and Shalem [24] could prove beneficial in this regard.

What about other geometries? We analysed linear hash tables where objects are assigned random intervals. One could also consider a square hash table $(\mathbb{Z}_n^2)^2$ where objects are assigned random squares of size $\ell \times \ell$ (with no alignment requirement). We suspect that understanding the thresholds in such cases would require completely new techniques.

**ACKNOWLEDGMENTS**

I am indebted to my advisor Martin Dietzfelbinger for drawing my attention to this problem as well as for providing a constant stream of useful literature recommendations. When discussing a preliminary version of this work at the Dagstuhl Seminar 17181 on Theory and Applications of Hashing, Michael Mitzenmacher and Konstantinos Panagiotou provided useful comments. The full version of this article also profited significantly from helpful reviewers who, among other things, pointed out the need for Sections 7 and 9 to 11 as well as more details in Section 5.

**REFERENCES**

[1] David Aldous and J. Michael Steele. 2004. *The Objective Method: Probabilistic Combinatorial Optimization and Local Weak Convergence*. Springer, Berlin, 1–72. DOI: https://doi.org/10.1007/978-3-662-09444-0_1

[2] Stephan Beyer. 2012. *Analysis of the Linear Probing Variant of Cuckoo Hashing*. Master’s Thesis. Technische Universität Ilmenau. Retrieved from http://gso.gbv.de/DB=2.1/PPNSET?PPN=685166759.

[3] Julie Anne Cain, Peter Sanders, and Nicholas C. Wormald. 2007. The random graph threshold for $k$-orientability and a fast algorithm for optimal multiple-choice allocation. In *Proceedings of the 18th SODA*. 469–476. Retrieved from http://dl.acm.org/citation.cfm?id=1283383.1283433.

[4] Martin Dietzfelbinger, Andreas Goerdt, Michael Mitzenmacher, Andrea Montanari, Rasmus Pagh, and Michael Rink. 2010. Tight thresholds for cuckoo hashing via XORSAT. In *Proceedings of the 37th ICALP (1)*. 213–225. DOI: https://doi.org/10.1007/978-3-642-14165-2_19

[5] Martin Dietzfelbinger and Christoph Weidling. 2005. Balanced allocation and dictionaries with tightly packed constant size bins. In *Proceedings of the 32nd ICALP*. 166–178. DOI: https://doi.org/10.1007/11523468_14

[6] Martin Dietzfelbinger and Christoph Weidling. 2007. Balanced allocation and dictionaries with tightly packed constant size bins. *Theor. Comput. Sci*. 380, 1-2 (2007), 47–68. DOI: https://doi.org/10.1016/j.tcs.2007.02.054

[7] Daniel Fernholz and Vijaya Ramachandran. 2007. The $k$-orientability Thresholds for $G_{n,p}$. In *Proceedings of the 18th SODA*. 459–468. Retrieved from http://dl.acm.org/citation.cfm?id=1283383.1283432.
[8] Dimitris Fotakis, Rasmus Pagh, Peter Sanders, and Paul G. Spirakis. 2005. Space efficient hash tables with worst case constant access time. Theory Comput. Syst. 38, 2 (2005), 229–248. DOI: https://doi.org/10.1007/s00224-004-1195-x
[9] Nikolaos Fountoulakis, Megha Khosla, and Konstantinos Panagiotou. 2011. The multiple-orientability thresholds for random hypergraphs. In Proceedings of the 22nd SODA. 1222–1236. Retrieved from http://www.siam.org/proceedings/soda/2011/SODA11_092_fountoulakism.pdf.
[10] Nikolaos Fountoulakis and Konstantinos Panagiotou. 2010. Orientability of random hypergraphs and the power of multiple choices. In Proceedings of the 37th ICALP (1). 348–359. DOI: https://doi.org/10.1007/978-3-642-14165-2_30
[11] Nikolaos Fountoulakis and Konstantinos Panagiotou. 2012. Sharp load thresholds for cuckoo hashing. Random Struct. Algorithms 41, 3 (2012), 306–333. DOI: https://doi.org/10.1002/rsa.20426
[12] Alan M. Frieze and Páll Melsted. 2012. Maximum matchings in random bipartite graphs and the space utilization of cuckoo hash tables. Random Struct. Algorithms 41, 3 (2012), 334–364. DOI: https://doi.org/10.1002/rsa.20427
[13] Pu Gao and Nicholas C. Wormald. 2010. Load balancing and orientability thresholds for random hypergraphs. In Proceedings of the 42nd STOC. 97–104. DOI: https://doi.org/10.1145/1806689.1806705
[14] Svante Janson and Malwina J. Luczak. 2007. A simple solution to the k-core problem. Random Struct. Algorithms 30, 1-2 (2007), 50–62. DOI: https://doi.org/10.1002/rsa.20147
[15] Megha Khosla. 2013. Balls into bins made faster. In Proceedings of the 21st ESA. 601–612. DOI: https://doi.org/10.1007/978-3-642-40450-4_51
[16] Megha Khosla and Avishek Anand. 2019. A faster algorithm for cuckoo insertion and bipartite matching in large graphs. Algorithmica 81, 9 (2019), 3707–3724. DOI: https://doi.org/10.1007/s00453-019-00595-4
[17] Mathieu Leconte. 2013. Double hashing thresholds via local weak convergence. In Proceedings of the 51st Annual Allerton Conference. 131–137. DOI: https://doi.org/10.1109/Allerton.2013.6736515
[18] M. Leconte, M. Lelarge, and L. Massoulié. 2013. Convergence of multivariate belief propagation, with applications to cuckoo hashing and load balancing. In Proceedings of the 24th SODA. 35–46. http://dl.acm.org/citation.cfm?id=2627817.2627820.
[19] Eric Lehman and Rina Panigrahy. 2009. 3.5-way cuckoo hashing for the price of 2-and-a-Bit. In Proceedings of the 17th ESA. 671–681. DOI: https://doi.org/10.1007/978-3-642-04128-0_60
[20] Marc Lelarge. 2012. A new approach to the orientation of random hypergraphs. In Proceedings of the 23rd SODA. 251–264. Retrieved from http://dl.acm.org/citation.cfm?id=2095139.
[21] Michael Mitzenmacher, Konstantinos Panagiotou, and Stefan Walzer. 2018. Load thresholds for cuckoo hashing with double hashing. In SWAT (Leibniz International Proceedings in Informatics (LIPIcs)18), David Eppstein (Ed.), Vol. 101. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 29:1–29:9. DOI: https://doi.org/10.4230/LIPIcs.SWAT.2018.29
[22] Michael Molloy. 2005. Cores in random hypergraphs and Boolean formulas. Random Struct. Algorithms 27, 1 (2005), 124–135. DOI: https://doi.org/10.1002/rsa.20061
[23] Rasmus Pagh and Flemming Friche Rodler. 2004. Cuckoo hashing. J. Algorithms 51, 2 (2004), 122–144. DOI: https://doi.org/10.1016/j.jalgor.2003.12.002
[24] Ely Porat and Bar Shalam. 2012. A cuckoo hashing variant with improved memory utilization and insertion time. In Proceedings of the 22nd DCC. DOI: https://doi.org/10.1109/DCC.2012.41

Received 19 October 2018; revised 11 September 2021; accepted 7 March 2023