

On Risks of Using Cuckoo Hashing with Simple Universal Hash Classes*

Martin Dietzfelbinger[†]

Ulf Schellbach[‡]

Abstract

Cuckoo hashing, introduced by Pagh and Rodler [10], is a dynamic dictionary data structure for storing a set S of n keys from a universe U , with constant lookup time and amortized expected constant insertion time. For the analysis, space $(2+\varepsilon)n$ and $\Omega(\log n)$ -wise independence of the hash functions is sufficient. In experiments mentioned in [10], several weaker hash classes worked well; however, a certain simple multiplicative hash family worked badly.

In this paper, we prove that the failure probability is high when cuckoo hashing is run with the multiplicative class or with the very common class of linear hash functions over a prime field, even if space $4n$ is provided. The key set S is fully random, but it must be relatively dense in the universe U of all keys (like $|S| \geq |U|^{11/12}$). The bad behavior and the fact that this effect depends on the density of S in U can also be observed in experiments. The result transfers to larger universes if the keys are chosen from a suitable smaller domain.

Viewed from a different perspective, our result illustrates that care must be taken when applying a recent result of Mitzenmacher and Vadhan ([12], SODA 2008) proving good behavior of universal hash classes in combination with key sets that have some entropy. Their result is applicable to cuckoo hashing. A technical hypothesis in [12], namely the assumption that either the “collision probability” or the “maximum probability” is small, translates into the condition that $|S|$ is relatively small in comparison to $|U|$. Our result shows that the result from [12] on 2-universal classes ceases to hold if $|S|/|U|$ is not small enough, even for very common 2-universal hash classes and fully random key sets.

1 Introduction

1.1 Background. Cuckoo hashing, introduced by Pagh and Rodler [10], is a strategy for maintaining hash tables for keys from U , $|U| = N$, so that lookups take constant time in the worst case. The data structure consists of two tables of size m each, and it uses two hash functions $h_1, h_2: U \rightarrow [m]$. For the scheme to work (with fully random hash functions) it is necessary and sufficient that $m \geq (1 + \varepsilon)n$ for an arbitrary constant $\varepsilon > 0$, where n is the number of keys stored.

Pagh and Rodler’s analysis establishes expected amortized constant time for insertion; for the analysis

to work it is required that the hash functions are $c \cdot \log n$ -wise independent. A further analysis of Devroye and Morin [9] establishes a similar result, assuming full independence and uniformity of all hash values. In experiments, cuckoo hashing works very well with weaker hash function classes. However, Pagh and Rodler [10] report on experimental results that indicate that cuckoo hashing will not work well in combination with the “multiplicative class” (which consists of functions $h_a: [2^k] \rightarrow [2^l]$ of the form $h_a(x) = ((a \cdot x) \bmod 2^k) \operatorname{div} 2^{k-l}$, for $0 < a < 2^k$ odd, and has a certain universality property). They state that they do not have an explanation for this phenomenon.

In 2008, Mitzenmacher and Vadhan [12] proved that if a universal hash class \mathcal{H} is used and the key set exhibits a certain degree of (Renyi) entropy, and further technical conditions are fulfilled, then the combination of the key set and a hash function chosen at random from \mathcal{H} will behave very close to full randomness.

1.2 Our Results. (a) We show that if cuckoo hashing with $2^l = m = 2n$ is employed (this table size is twice as large as the threshold sufficient for the standard analysis), then all function pairs from the multiplicative hash class will work badly with high probability for fully random key sets of size n , if $n/N > N^{1-\gamma}$, for some constant $\gamma > 0$. In other words, although the entropy of the input data is as large as possible given $|U|$ and $|S|$, for every pair of multiplicative hash functions the failure probability for a key set S relatively small in comparison to m is extremely high. This explains the experimental results obtained by Pagh and Rodler and justifies a warning against using this simple class of hash functions in combination with cuckoo hashing.

The result can be “lifted” to larger universes U , but then the key set S must be chosen randomly from a special subset $U' \subseteq U$, where again $n/|U'| > |U'|^{1-\gamma}$, for some constant γ .

Moreover, our result is relevant in a more general context. The results in [12] (and earlier results [4, 5], see [12]) require that either the “collision probability”

*Research supported in part by DFG grant DI 412/10-1.

[†]Technische Universität Ilmenau, Germany.
Email: martin.dietzfelbinger@tu-ilmenau.de

[‡]Technische Universität Ilmenau, Germany.
Email: ulf.schellbach@tu-ilmenau.de

or the “maximum probability” in the key set be small, which for a fully random key set translates into the requirement that it must not be too dense in the universe (or the “support”, the set of possible keys). Our result shows that this condition is necessary, and that it is relevant even in very natural circumstances (standard hash classes, fully random key sets).

(b) We show that cuckoo hashing with a standard almost 2-wise independent class of hash functions (functions of the form $h_{a,b} = ((ax+b) \bmod p) \bmod m$, $p \geq N$ a prime number) exhibits a similar behavior as the class in (a), again in the case where the key set is relatively dense in U . This is true even when the two hash functions use different prime moduli.

Our proof techniques are ad hoc and new. We study the “complete cuckoo graph” created by all keys in combination with hash functions from the considered classes, where an edge represents the hash values of a key, and identify certain “bad edge sets” with the property that the insertion of the corresponding sets of keys must fail. We show that random edge sets of relatively small size contain such a bad edge set with high probability.

1.3 Further Related Work. In [13], Cohen and Kane construct 2-, 3-, and even 5-wise independent hash families for which cuckoo hashing has high failure probability. However, these families are quite contrived and far from being common.

2 Preliminaries

2.1 Cuckoo Hashing. Given two hash tables T_1, T_2 , each of size $m \in \mathbb{N}$, and hash functions h_1, h_2 mapping a universe U of keys to $[m] = \{0, \dots, m-1\}$, a key x must be stored either in cell $h_1(x)$ of T_1 or in cell $h_2(x)$ of T_2 . We say that h_1 and h_2 are suitable for a set $S \subseteq U$ if it is possible to place the keys from S in the described way, such that any two distinct keys are stored in distinct table cells. This is all one needs to know about cuckoo hashing in our context. For a detailed description, see [10].

2.2 The Cuckoo Graph and Bad Edge Sets.

The *cuckoo graph* (see e.g. [9]) represents the hash value distribution of a set S of keys in U for hash functions $h_1, h_2: U \rightarrow [m]$. Its vertices correspond to the table cells of T_1 and T_2 , and each edge connects the two possible locations $T_1[h_1(x)]$ and $T_2[h_2(x)]$ for a key $x \in S$. Formally, the cuckoo graph $G(S, h_1, h_2)$ is defined as an undirected bipartite multigraph (V_1, V_2, E) with vertex sets $V_1 = [m]$ and $V_2 = [m]$, and edge set $E = \{(h_1(x), h_2(x)) \mid x \in S\}$.

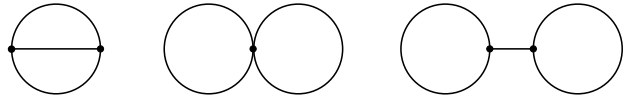
We refer to $G(U, h_1, h_2)$ as the *complete cuckoo graph*.

We call $E' \subseteq E$ a *bad edge set* (of $G(S, h_1, h_2)$) if $|E'|$ is larger than the number of distinct vertices that are incident with edges in E' . The following lemma will be useful.

LEMMA 2.1. *The hash functions h_1 and h_2 are suitable for S if and only if $G(S, h_1, h_2)$ does not contain a bad edge set.*

Proof. In [9] it is shown that h_1 and h_2 are suitable for S if and only if $G(S, h_1, h_2)$ does not have a connected component that neither is a tree nor has exactly one cycle. It is not hard to see that this condition is equivalent to the graph having no bad edge set. \square

In our proofs, we will focus on minimal bad edge sets (of constant size). The abstract shape of such edge sets is as depicted below. (Lines represent simple paths of length at least 1, and dots are selected vertices.)



2.3 Hash Function Families. We consider two different kinds of hash function families. First, let $1 \leq l \leq k$. Then let

$$\mathcal{H}_{k,l}^{\text{mult}} := \{h_a: [2^k] \rightarrow [2^l] \mid a \in O_k\},$$

where $O_k := \{1, 3, \dots, 2^k - 1\}$, and for $x \in [2^k]$ we let $h_a(x) := (a \cdot x \bmod 2^k) \bmod 2^{k-l}$. We refer to this family as the *multiplicative class* [6]. Second, let p be a prime number, and $m < p$. Then let

$$\mathcal{H}_{p,m}^{\text{lin}*} := \{h_{a,b}: [p] \rightarrow [m] \mid a \in [p] - \{0\}, b \in [p]\},$$

where, for all $x \in [p]$, we define $h_{a,b}(x) := ((ax + b) \bmod p) \bmod m$. We refer to this family as the *linear class* [1]. More generally, we consider the hash family of polynomials of degree up to $d - 1$, i.e.,

$$\mathcal{H}_{p,m}^d := \{h_{a_0, \dots, a_{d-1}}: [p] \rightarrow [m] \mid a_0, \dots, a_{d-1} \in [p]\},$$

where $h_{a_0, \dots, a_{d-1}}(x) = ((a_0 + a_1x + \dots + a_{d-1}x^{d-1}) \bmod p) \bmod m$ for $x \in [p]$, see [1]. Obviously, $\mathcal{H}_{p,m}^{\text{lin}*}$ is practically the same as $\mathcal{H}_{p,m}^2$. We used $\mathcal{H}_{p,m}^3$ as a reference class in our experiments.

In order to classify families of hash functions, we use the following well known generalizations of the original notion of universality, which is due to Carter and Wegman [1].

DEFINITION 2.1. A family \mathcal{H} of hash functions $h: U \rightarrow [m]$ is called (c, k) -universal if for arbitrary k distinct keys $x_1, \dots, x_k \in U$, arbitrary k values $y_1, \dots, y_k \in [m]$, and a function $h \in \mathcal{H}$ chosen uniformly at random, we have

$$\Pr(h(x_1) = y_1, \dots, h(x_k) = y_k) \leq c/m^k .$$

It is called c -universal if for arbitrary keys $x \neq y$ and h chosen uniformly at random, we have

$$\Pr(h(x) = h(y)) \leq c/m .$$

In [6], the multiplicative class is proved to be 2-universal. For any fixed $d \geq 2$, the class of polynomials of degree up to $d - 1$ is known to be approximately $(2, d)$ -universal. The linear class is 1-universal and approximately $(1, 2)$ -universal. The results of [12] are formulated for 1-universal classes, but they hold equally well for 2-universal classes.

3 The Special Case of Very Dense Key Sets

This brief section deals with the special case of very dense key sets, in order to explain a technical condition of the following theorems. It turns out that the performance of cuckoo hashing combined with the multiplicative or linear class is best possible if m/N is at least $1/2$. Note that throughout the paper we focus on m/N (rather than n/N), because it is this ratio that determines the structure of the hash functions.

PROPOSITION 3.1. If $m/N \geq 1/2$, then all functions $h_1, h_2 \in \mathcal{H}_{k,l}^{mult}$ are suitable for all sets $S \subseteq U$. The same holds for $\mathcal{H}_{p,m}^{lin*}$.

Proof. It suffices to show that in these cases the complete cuckoo graph $G = G(U, h_1, h_2)$ has maximum degree of 2, i. e., its components are simple paths and simple cycles. It is clear that in this situation the keys can be arranged as required.

As for $\mathcal{H}_{k,l}^{mult}$, note that $O_k = \{1, 3, \dots, 2^k - 1\}$ is an Abelian group with respect to multiplication modulo 2^k . So, for each $a \in O_k$ the mapping

$$x \mapsto ax \bmod 2^k$$

is a permutation of $U = [2^k]$, and its restriction to O_k is a permutation of O_k . The assumption $m/N = 2^{l-k} \geq 1/2$ implies $k - l \in \{0, 1\}$, and hence

$$x \mapsto (ax \bmod 2^k) \operatorname{div} 2^{k-l}$$

is one-to-one on O_k , and one-to-one on $U - O_k$. Consequently, for all $j \in [m]$ there are at most two

keys x with $h_a(x) = j$, and hence G has maximum degree 2.

The argumentation for $\mathcal{H}_{p,m}^{lin*}$ is similar. It uses that $[p]$ is a field w.r.t. addition and multiplication modulo p . \square

4 High Failure Probability for the Multiplicative Class

We consider the multiplicative class. For fixed hash functions h_1, h_2 , and $S \subseteq U$ chosen randomly, we denote the probability that h_1 and h_2 are not suitable for S as *failure probability* p_F . The purpose of this section is to establish the following theorem.

THEOREM 4.1. Let $h_{a_1}, h_{a_2} \in \mathcal{H}_{k,l}^{mult}$ be arbitrary, and let a set $S \subseteq U = [2^k]$ of size $m/2 = 2^{l-1}$ be chosen uniformly at random. If $l \leq k - 2$ and $l/k > 11/12$, then $p_F = 1 - o(1)$, for $l \rightarrow \infty$.

Note that the number $m/2$ of keys in S is way below the threshold for cuckoo hashing with random sets in the case of $c \log n$ -wise independent hash functions, which permits sizes up to $(1 - \delta)m$ for an arbitrary constant $\delta > 0$. The case $l > k - 2$ was considered in Proposition 3.1.

Proof. The general idea is to show that the complete cuckoo graph $G = G(U, h_{a_1}, h_{a_2})$ contains many bad edge sets of constant size, of which any two do not overlap too much, and to conclude that the subgraph of G that corresponds to a randomly chosen set $S \subseteq U$ is very likely to contain one of these bad edge sets.

LEMMA 4.1. The graph $G = G(U, h_{a_1}, h_{a_2})$ contains a set $\{K_1, \dots, K_m\}$ of m distinct bad edge sets of size ≤ 10 such that for all $i, 1 \leq i \leq m$, we have $|\{j \in \{1, \dots, m\} \mid K_j \cap K_i \neq \emptyset\}| \leq 13$.

Proof. For analyzing the structure of G we may assume that $a_1 = 1$, as the following lemma shows.

LEMMA 4.2. The set $\{G(U, h_{a_1}, h_{a_2}) \mid a_2 \in O_k\}$ of complete cuckoo graphs for fixed h_{a_1} and variable h_{a_2} does not depend on h_{a_1} . The same holds for $\{G(O_k, h_{a_1}, h_{a_2}) \mid a_2 \in O_k\}$.

Proof. Let r denote the mapping

$$x \mapsto a_1 x \bmod 2^k .$$

As a_1 is odd, we have for the edge set E of $G(U', h_{a_1}, h_{a_2})$, $U' \in \{U, O_k\}$:

$$\begin{aligned} E &= \{(h_{a_1}(x), h_{a_2}(x)) \mid x \in U'\} \\ &= \{(h_1(r(x)), h_{a_2 a_1^{-1}}(r(x))) \mid x \in U'\} , \end{aligned}$$

where the mapping

$$a_2 \mapsto a_2 a_1^{-1} \pmod{2^k}$$

is a permutation of O_k . It remains to observe that r is a permutation of U , and its restriction to O_k is a permutation of O_k . \square

So, for the proof of Lemma 4.1 we assume $a_1 = 1$, and consider $G = G(U, h_1, h_{a_2})$. We partition U into “grid sets” $G_m(c)$, $c \in [2^{k-l}]$, that are defined as follows:

$$G_m(c) := \{x_i(c) \mid i \in [m]\},$$

where $x_i(c) := (c + i \cdot 2^{k-l}) \pmod{2^k}$. A straightforward calculation proves the following.

LEMMA 4.3. *For each $h_a \in \mathcal{H}_{k,l}^{mult}$ and for all $i \in [m]$ we have $h_a(x_i(c)) = (h_a(c) + i \cdot a) \pmod{2^l}$.*

Proof. The calculation makes use of the following three equations, which hold for arbitrary natural numbers x and y , and whose correctness is immediate when the numbers are represented in binary:

$$(4.1) \quad (x \cdot 2^{k-l}) \pmod{2^k} = (x \pmod{2^l}) \cdot 2^{k-l},$$

$$(4.2) \quad (x \pmod{2^k}) \operatorname{div} 2^{k-l} = (x \operatorname{div} 2^{k-l}) \pmod{2^l},$$

$$(4.3) \quad x \operatorname{div} 2^{k-l} + (y \cdot 2^{k-l}) \operatorname{div} 2^{k-l} = (x + y \cdot 2^{k-l}) \operatorname{div} 2^{k-l}.$$

Now, the calculation is as follows, where the fourth equation makes use of (4.1) and (4.2), and the fifth equation applies (4.3).

$$\begin{aligned} h_a(x_i(c)) &= (a \cdot x_i(c)) \pmod{2^k} \operatorname{div} 2^{k-l} \\ &= (a \cdot (c + i \cdot 2^{k-l})) \pmod{2^k} \operatorname{div} 2^{k-l} \\ &= (ac \pmod{2^k} + (ai \cdot 2^{k-l}) \pmod{2^k}) \pmod{2^k} \operatorname{div} 2^{k-l} \\ &= (ac \pmod{2^k} + (ai \pmod{2^l}) \cdot 2^{k-l}) \operatorname{div} 2^{k-l} \pmod{2^l} \\ &= (h_a(c) + ai \pmod{2^l}) \pmod{2^l} \\ &= (h_a(c) + i \cdot a) \pmod{2^l}. \end{aligned}$$

\square

In other words, the image of $G_m(c)$ under h_a is also a grid set. Moreover, we shall see that the edge set corresponding to $G_m(c)$ in $G = (V_1, V_2, E)$ is a perfect matching with a structure that makes it possible to find many bad edge sets. If we denote the edge that corresponds to key $x_i(c)$ by $e_i(c)$, then

$e_i(c)$ is incident with $i \in V_1$, as shown by the following calculation which uses Lemma 4.3, (4.2) and the fact that $c < 2^{k-l}$ implies $h_1(c) = 0$:

$$\begin{aligned} e_i(c) &= (h_1(x_i(c)), h_{a_2}(x_i(c))) \\ &= ((h_1(c) + i) \pmod{2^l}, (h_{a_2}(c) + i \cdot a_2) \pmod{2^l}) \\ &= (i, (a_2 c \operatorname{div} 2^{k-l} + i \cdot a_2) \pmod{2^l}). \end{aligned}$$

For notational convenience, we consider a graph $\tilde{G} = (V_1, V_2, \tilde{E})$ derived from G by permuting V_2 . Precisely,

$$\tilde{E} = \{(h_1(x), (h_{a_2}(x) \cdot a_2^{-1}) \pmod{2^l}) \mid x \in U\},$$

where a_2^{-1} denotes the multiplicative inverse of a_2 modulo 2^k (and hence in particular we have $a_2 a_2^{-1} \pmod{2^l} = 1$ for $l < k$). Obviously, \tilde{G} is isomorphic to G , and hence bad edge sets in \tilde{G} correspond to bad edge sets in G . In \tilde{G} , edge $\tilde{e}_i(c)$ of key $x_i(c)$ is incident with $i \in V_1$ and with $i + o_c \in V_2$ for a constant

$$o_c := (a_2 c \operatorname{div} 2^{k-l} \cdot a_2^{-1}) \pmod{2^l},$$

$i \in [m]$. For any two distinct keys $c, c' < 2^{k-l}$, consider the edge set $C_i(c, c')$ defined as

$$\{\tilde{e}_i(c), \tilde{e}_{o_c+i}(0), \tilde{e}_{o_c+i}(c'), \tilde{e}_{o_{c'}+i}(c), \tilde{e}_{o_{c'}+i}(0), \tilde{e}_i(c')\},$$

of size at most 6, where arithmetic modulo m has to be applied to all indices. Now fix any three distinct elements $c, c', c'' \in [2^{k-l}]$. This is possible since $k - l \geq 2$. If two of the offsets $o_c, o_{c'}, o_{c''}$ are equal, say $o_c = o_{c'}$, then $C_i(c, c')$ is a bad edge set \tilde{K}_i of size 5, $i \in [m]$, since $\tilde{e}_{o_c+i}(0) = \tilde{e}_{o_{c'}+i}(0)$ (Figure 1(a)). Otherwise, $C_i(c, c')$ and $C_i(c, c'')$ are cycles of size 6 that overlap in two edges, and their union is a bad edge set \tilde{K}_i of size 10, $i \in [m]$ (Figure 1(b)).

Note that each of the bad edge sets \tilde{K}_i contains at most four distinct vertices in V_1 , and \tilde{K}_j , $j \in \{1, \dots, m\}$, is a copy of \tilde{K}_i , shifted modulo m . Therefore, the number of distinct bad edge sets \tilde{K}_j , $j \in \{1, \dots, m\}$, (where $j = i$ is included) overlapping with \tilde{K}_i is at most 13, because a necessary condition for $\tilde{K}_j \cap \tilde{K}_i \neq \emptyset$ is a common vertex in V_1 . Now, letting K_i denote the bad edge set in G that corresponds to \tilde{K}_i , respectively, completes the proof of Lemma 4.1. \square

By Lemma 4.1, we may fix a set $\{K_1, \dots, K_m\}$ of m distinct bad edge sets of size at most 10 such that for all $i = 1, \dots, m$ we have $|\{j \in \{1, \dots, m\} \mid K_j \cap K_i \neq \emptyset\}| \leq 13$. Let $S \subseteq U$, $|S| = m/2$, be chosen uniformly at random. Choosing S directly

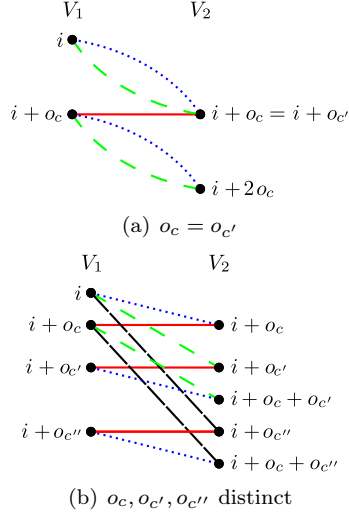


Figure 1: Bad edge sets for the multiplicative class

corresponds to choosing n edges in G at random. Let the random variable X_i , $1 \leq i \leq m$, take the value 1 if all edges of K_i have been chosen, 0 otherwise, and define $X := \sum_{1 \leq i \leq m} X_i$. Then $p_F \geq \Pr(X > 0)$. Note that the X_i are not independent. In order to establish a lower bound on $\Pr(X > 0)$, we invoke the *conditional expectation inequality* (for a proof, see [3]) and apply it for $(Y_1, \dots, Y_t) = (X_1, \dots, X_m)$:

LEMMA 4.4. *Let Y be the sum of arbitrary Bernoulli random variables Y_1, \dots, Y_t , $t \in \mathbb{N}$. Then we have*

$$\Pr(Y > 0) \geq \sum_{i=1}^t \frac{\Pr(Y_i = 1)}{E(Y | Y_i = 1)}.$$

As $\Pr(X_i = 1)$ only depends on $|K_i|$, we obtain under the given conditions $l \leq k - 2$ and $l/k > 11/12$:

$$\Pr(X_i = 1) \geq \frac{\binom{2^k - 10}{2^{l-1} - 10}}{\binom{2^k}{2^{l-1}}} > 2^{-10(k-l+2)}.$$

Furthermore, distinguishing between bad edge sets K_j that overlap with K_i and those which do not yields $E(X) + 13$ as an upper bound for $E(X | X_i = 1)$. Putting it all together, we get

$$p_F > \left(1 + 2^{4+10(k-l+2)-l}\right)^{-1}.$$

For $l/k > 11/12$, the latter term is at least $(1 + 2^{-(1/12) \cdot k + 24})^{-1}$, and hence $p_F = 1 - o(1)$ for $k, l \rightarrow \infty$. This concludes the proof of Theorem 4.1. \square

Theorem 4.1 can be lifted to larger universes U in the following way. Define $U' \subseteq U = [2^k]$ as the grid set

$$\{y \cdot 2^{k-l-2} \mid y \in [2^{l+2}]\}.$$

Observe that for an odd number $a \in O_k$, the mapping $x \mapsto ax \bmod 2^k$ is a permutation of U , and that this mapping preserves the lowest order 1 bit of x . Therefore the mapping is also a permutation of U' . This implies in particular that only the $l + 2$ lowest order bits of a are relevant for $h_a(x) = (ax \bmod 2^k) \operatorname{div} 2^{k-l}$, and hence if S is chosen randomly from U' then we are in the case $l = k - 2$ of Theorem 4.1. This leads to the following.

COROLLARY 4.1. *Let $h_{a_1}, h_{a_2} \in \mathcal{H}_{k,l}^{\text{mult}}$ be arbitrary and let a set $S \subseteq U'$ of size $m/2 = 2^{l-1}$ be chosen uniformly at random. If $l \leq k - 2$ and $l/(l + 2) > 11/12$, then $p_F = 1 - o(1)$, for $l \rightarrow \infty$.*

5 High Failure Probability for the Linear Class

In this section, we prove a theorem for the linear class, in analogy to Theorem 4.1. Note that the linear class is *very standard*.

THEOREM 5.1. *Let $h_{a_1, b_1}, h_{a_2, b_2} \in \mathcal{H}_{p,m}^{\text{lin}*}$ be arbitrary, and let $S \subseteq U = [p]$, $|S| = \lceil m/2 \rceil$, be chosen uniformly at random. If $m/p \in [p^{-(1/7-\varepsilon)}, 1/7]$ for a constant $\varepsilon \in (0, 1/7)$ then $p_F = 1 - o(1)$ for $m \rightarrow \infty$.*

Proof. The general approach is the same as in the proof of Theorem 4.1, but the details differ considerably. Fix hash functions $h_{a_1, b_1}, h_{a_2, b_2} \in \mathcal{H}_{p,m}^{\text{lin}*}$ and consider the complete cuckoo graph $G = G(U, h_{a_1, b_1}, h_{a_2, b_2})$.

LEMMA 5.1. *Let $m' = \lceil m/3 \rceil$. If $m/p \leq 1/7$, then G contains m' distinct bad edge sets $K_1, \dots, K_{m'}$ of size 6 such that for all i , $1 \leq i \leq m'$, we have $|\{j \in \{1, \dots, m'\} \mid K_j \cap K_i \neq \emptyset\}| \leq 5$.*

Proof. By a lemma and proof that is similar to Lemma 4.2 and its proof we may assume w.l.o.g. that $h_{a_1, b_1} = h_{1,0}$, where $h_{1,0}(x) = x \bmod m$:

LEMMA 5.2. *The set $\{G(U, h_{a_1, b_1}, h_{a_2, b_2}) \mid a_2 \in [p] - \{0\}, b_2 \in [p]\}$ of complete cuckoo graphs for fixed h_{a_1, b_1} and variable h_{a_2, b_2} does not depend on h_{a_1, b_1} .*

Proof. Let r denote the mapping

$$x \mapsto (a_1 x + b_1) \bmod p.$$

As $a_1 > 0$, we have for the edge set E of $G(U, h_{a_1}, h_{a_2})$:

$$\begin{aligned} E &= \{(h_{a_1, b_1}(x), h_{a_2, b_2}(x)) \mid x \in U\} \\ &= \{(h_{1,0}(r(x)), h_{a_2 a_1^{-1}, b_2 - a_2 a_1^{-1} b_1}(r(x))) \mid x \in U\}, \end{aligned}$$

where the mapping $a_2 \mapsto a_2 a_1^{-1} \bmod p$ is a permutation of $[p] - \{0\}$, and $b_2 \mapsto b_2 - a_2 a_1^{-1} b_1 \bmod p$ is a permutation of $[p]$. It remains to observe that r is a permutation of U . \square

Thus, consider $G = G(U, h_{1,0}, h_{a,b}) = (V_1, V_2, E)$ for an arbitrary $h_{a,b} \in \mathcal{H}_{p,m}^{\text{lin}^*}$. We study the neighborhood $\Gamma_j = h_{a,b}(h_{1,0}^{-1}(j)) \subseteq V_2$ of an arbitrary vertex $j \in V_1$. Every key x whose corresponding edge e_x is incident with $j \in V_1$ is in $h_{1,0}^{-1}(j)$, and hence has the form $x = im + j$ for some $i \in \mathbb{N}$. Note that $i \leq t := \lceil p/m \rceil$, and the degree of any vertex in G is either t or $t-1$. We refer to a key $im + j$ as $x_i(j)$, call the corresponding edge the i -edge of j , and refer to its endpoint in V_2 as the i -neighbor of j , or as $n_i(j)$. The following lemma will help us to understand the structure of Γ_j .

LEMMA 5.3. (LEAP EFFECT) *Let r_1 and r_2 be fixed positive integers, and define $\Delta := (-r_1) \bmod r_2$. Then for all $x \in \mathbb{N}$ we have:*

$$(x \bmod r_1) \bmod r_2 = (x + \lfloor x/r_1 \rfloor \cdot \Delta) \bmod r_2.$$

Proof. Straightforward induction on $x \in \mathbb{N}$:

$x=0$: Clear. $x \rightarrow x+1$: We distinguish two cases. If r_1 divides $x+1$, then $\lfloor (x+1)/r_1 \rfloor = \lfloor x/r_1 \rfloor + 1$ and $r_1 = x \bmod r_1 + 1$. Together with the induction hypothesis, this yields

$$\begin{aligned} &(x+1) \bmod r_1 \bmod r_2 \\ &= 0 \bmod r_2 \\ &= (r_1 - r_1) \bmod r_2 \\ &= ((x \bmod r_1 + 1) + \Delta) \bmod r_2 \\ &= ((x + \lfloor x/r_1 \rfloor \cdot \Delta) + 1 + \Delta) \bmod r_2 \\ &= ((x+1) + \lfloor (x+1)/r_1 \rfloor \cdot \Delta) \bmod r_2. \end{aligned}$$

Otherwise, if r_1 does not divide $x+1$, then $\lfloor (x+1)/r_1 \rfloor = \lfloor x/r_1 \rfloor$ and $(x+1) \bmod r_1 = (x \bmod r_1) + 1$, and an application of the induction hypothesis yields

$$\begin{aligned} &(x+1) \bmod r_1 \bmod r_2 \\ &= (x \bmod r_1 + 1) \bmod r_2 \\ &= ((x + \lfloor x/r_1 \rfloor \cdot \Delta) + 1) \bmod r_2 \\ &= ((x+1) + \lfloor (x+1)/r_1 \rfloor \cdot \Delta) \bmod r_2. \end{aligned}$$

We want to simplify the term $n_i(j) = h_{a,b}(x_i(j))$. Applying Lemma 5.3 for $r_1 = p$ and $r_2 = m$ leads to the basic observation that Γ_j is nearly a grid, more precisely: For each $i \in [t-2]$ we have

$$n_{i+1}(j) \in \{(n_i(j) + s') \bmod m, (n_i(j) + s'') \bmod m\},$$

where $s' = s \bmod m$, $s'' = (s + \Delta) \bmod m$, $s = am \bmod p$, and $\Delta = (-p) \bmod m$.

For each i -edge of j with $i \in [t-2]$, there is still an $(i+1)$ -edge of j . So, call the former a *predecessor edge*, and the latter its *successor edge*. Furthermore, call a vertex in V_2 *obstructive* if it is incident with at least five predecessor edges. Let l be an obstructive vertex and fix any five of its incident predecessor edges. Their respective successor edges are incident with $(l + s') \bmod m$ or $(l + s'') \bmod m$ in V_2 , and therefore at least three of these successor edges must have the same endpoint $k \in V_2$. Fix three of the successor edges with endpoint k . Together with their predecessor edges, they form a bad edge set $K^{(l)}$ of size 6 (see Figure 2). If l and l' are distinct obstructive vertices, then their respective bad edge sets $K^{(l)}$ and $K^{(l')}$ must also be distinct, because otherwise there would be a predecessor edge which is its own successor edge, which is impossible. Moreover, if we fix $K^{(l)}$ for every obstructive vertex l , then for each obstructive vertex l , we have

$$|\{l' \in V_2 \mid l' \text{ obstructive vertex, } K^{(l)} \cap K^{(l')} \neq \emptyset\}| \leq 5,$$

as a necessary condition for $K^{(l)} \cap K^{(l')} \neq \emptyset$ is a common vertex in V_2 .

It remains to show that there are $m' = \lceil m/3 \rceil$ obstructive vertices. The number of predecessor edges in G is $(t-2)m$, where the assumption $m/p \leq 1/7$ implies $t-2 \geq 5$. With respect to maximizing the number of obstructive vertices, the worst case is $t-2 = 5$. Now it suffices to use the fact that the degree of each vertex of G is at most t . \square

We complete the proof of Theorem 5.1 in a way similar to the proof of Theorem 4.1: Lemma 5.1 guarantees the existence of a set $\{K_1, \dots, K_{m'}\}$ of bad edge sets in G . We fix such a set. Choose $\lceil m/2 \rceil$ edges from G uniformly at random. Define 0-1 random variables $X_1, \dots, X_{m'}$ as follows: $X_i = 1$ if and only if all edges of K_i are chosen, and let $X = \sum_{1 \leq i \leq m'} X_i$. Clearly, $p_F \geq \Pr(X > 0)$. Under the given condition $m/p \geq p^{-(1/7-\varepsilon)}$, $\varepsilon \in (0, 1/7)$, an application of the conditional expectation inequality (Lemma 4.4) yields the following lower bound for

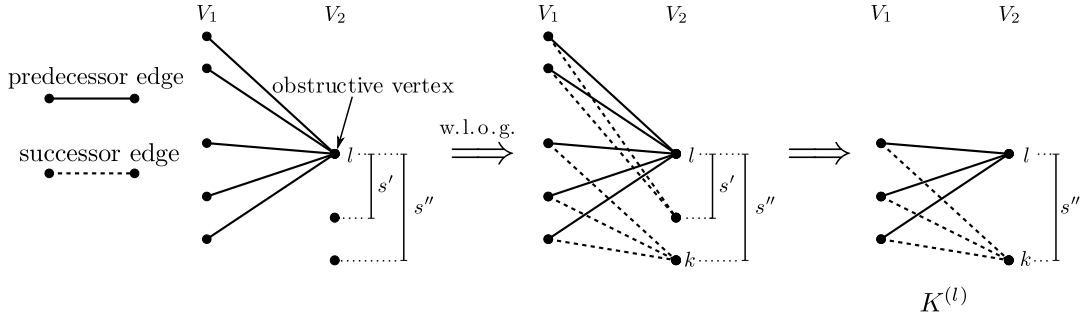


Figure 2: Bad edge sets for the linear class

$\Pr(X > 0)$:

$$\left(1 + 2^6 \cdot 3 \cdot 5 \cdot \left(1 - \frac{10}{p^{6/7+\varepsilon}}\right)^{-6} \cdot p^{-7\varepsilon}\right)^{-1} = 1 - o(1),$$

for $p, m \rightarrow \infty$. \square

6 High Failure Probability for Two Distinct Linear Classes

Again, we consider the linear class. It might seem plausible that the performance strongly improves if h_1 and h_2 are chosen from linear classes over fields given by distinct prime numbers p_1 and p_2 , respectively. It is the purpose of this section to show that this is not the case.

THEOREM 6.1. *Let h_{a_1, b_1} and h_{a_2, b_2} in $\mathcal{H}_{p_1, m}^{\text{lin}^*}$ and $\mathcal{H}_{p_2, m}^{\text{lin}^*}$, respectively, where $p_1 \leq p_2 \leq \alpha p_1$ holds for an arbitrary constant $\alpha \geq 1$, and let $S \subseteq U = [p_1]$, $|S| = \lceil m/2 \rceil$, be chosen uniformly at random. If $m/p_1 \in [p_1^{-(1/8-\varepsilon)}, 1/73]$ for any constant $\varepsilon \in (0, 1/8)$, then $p_F = 1 - o(1)$, for $m \rightarrow \infty$.*

Proof. The basic approach is the same as before. Fix arbitrary hash functions h_{a_1, b_1} and h_{a_2, b_2} in $\mathcal{H}_{p_1, m}^{\text{lin}^*}$ and $\mathcal{H}_{p_2, m}^{\text{lin}^*}$, respectively. Then there exist many bad edge sets of constant size, of which very few pairs have a common edge.

LEMMA 6.1. *Let $m' := \lceil m/(\alpha(t+1)) \rceil$ for $t := \lceil p_1/m \rceil - 1$. If $m/p_1 \leq 1/73$, then $G(U, h_{a_1, b_1}, h_{a_2, b_2})$ contains m' distinct bad edge sets $K_1, \dots, K_{m'}$ of size 6 such that for each i , $1 \leq i \leq m'$, we have $|\{j \in \{1, \dots, m'\} \mid K_j \cap K_i \neq \emptyset\}| \leq 5$.*

Proof. In contrast to the proof of Lemma 5.1, we cannot assume that $h_{a_1, b_1} = h_{1,0}$, because of the distinct moduli of h_{a_1, b_1} and h_{a_2, b_2} . However, we may get rid of one modulus with the help of the following lemma, which says that a grid set modulo p contains a long arithmetic sequence in \mathbb{N} .

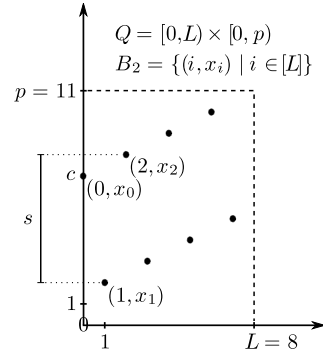


Figure 3: B_2

LEMMA 6.2. *Assume p is a prime and $L, \tilde{s} \in [p]$ are arbitrary, where $L > 2$ and $\tilde{s} > 0$. Then there exists an $\hat{s} \in [p]$, $0 < \hat{s} \leq p/\sqrt{L/2}$, such that for all $c \in [p]$ the grid set $B := \{x_i \mid i \in [L]\}$, where $x_i := (i \cdot \tilde{s} + c) \bmod p$, of size L contains an arithmetic sequence $(\hat{x}_k)_{k \in [l]}$ with step size \hat{s} and length $l = \lfloor \sqrt{L/2} \rfloor$, i. e., for $k = 0, 1, \dots, l-2$ we have $\hat{x}_{k+1} = \hat{x}_k + \hat{s} = \hat{x}_0 + (k+1) \cdot \hat{s}$.*

Proof. For each pair of distinct keys $x_i, x_j \in B$, define

$$A_{i,j} := \{x_{i+z \cdot d} \in B \mid d = |j-i|, z \in \mathbb{Z}, i+z \cdot d \in [L]\}$$

and $s_{i,j} := \min\{|x_j - x_i|, p - |x_j - x_i|\}$.

The subset $A_{i,j}$ can be viewed as a sequence $(y_k)_{k \in [l']}$ of length $l' \geq \lfloor L/d \rfloor$, which increases by $s_{i,j}$ in one step cyclically modulo p . Moreover, $s_{i,j}$ does not depend on the offset c of B . Now assume for the time being $l' \geq \lfloor \sqrt{2t} \rfloor$, i. e., $l' \geq 2 \lfloor \sqrt{L/2} \rfloor = 2l$, and $s_{i,j} \leq p/\sqrt{L/2}$. Then $(y_k)_{k \in [l']}$ contains the desired arithmetic sequence $(\hat{x}_k)_{k \in [l]}$ with step size $\hat{s} := s_{i,j} \leq p/\sqrt{L/2}$: Let w be the smallest index $w \in [l'] - \{0\}$ such that $y_{w-1} > y_w$. If no such w

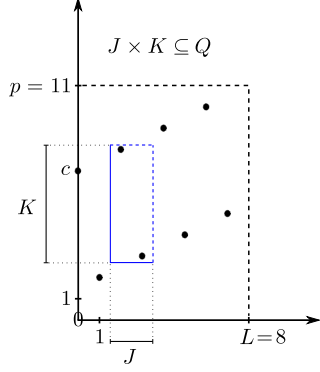


Figure 4: Suitable rectangle $J \times K \subseteq Q$ yields sufficiently long sequence (y_k)

exists, we are done. Otherwise, for each $k \in [l]$, set

$$\hat{x}_k := \begin{cases} y_k & \text{if } w' \geq l \\ y_{w'+k} & \text{otherwise.} \end{cases}$$

Regard B as a point set $B_2 := \{(i, x_i) \mid i \in [L]\}$ in the half-open rectangle $Q := [0, t) \times [0, p) \subseteq \mathbb{R}^2$ (Figure 3). Let $J \times K \subseteq Q$ for cyclic intervals $J := [l_1, r_1)$ and $K := [l_2, r_2)$, i. e., J and K may be wrapped around the boundaries of $[0, L)$ and $[0, p)$, respectively, and hence we have $|J| := (r_1 - l_1) \bmod L$ and $|K| := (r_2 - l_2) \bmod p$ for the lengths of J and K (Figure 4).

Observe the following. If $|J| \leq \sqrt{L/2}$, $|K| \leq p/\sqrt{L/2}$, and $J \times K$ contains $(i, x_i), (j, x_j) \in B_2$, $i \neq j$, then, $A_{i,j}$ yields the desired sequence $(y_k)_{k \in [l']}$ of length $l' \geq \lfloor L/|J| \rfloor \geq \lfloor \sqrt{2L} \rfloor$, with a step size $s_{i,j} \leq |K| \leq p/\sqrt{L/2}$.

It remains to show that a rectangle $J \times K$ with the above-mentioned properties exists, i. e., $|J| \leq \sqrt{L/2}$, $|K| \leq p/\sqrt{L/2}$, and $J \times K$ contains $(i, x_i), (j, x_j) \in B_2$, $i \neq j$. Assume this is not the case. Then define half-open rectangles $Q_i := [i, (i + \sqrt{L/2}) \bmod L) \times [x_i, (x_i + p/\sqrt{L/2}) \bmod p)$ for $0 \leq i \leq L-1$. By our assumption, the rectangles Q_0, \dots, Q_{L-1} are pairwise disjoint. This implies that the area of $\bigcup_{i \in [L]} Q_i$ is equal to the area of Q , and hence $Q = \bigcup_{i \in [L]} Q_i$. Consider Q_0 . Its bottom left corner is $(0, x_0) \in \mathbb{N}^2$. Our observation implies that, cyclically modulo L and p , there must be neighboring rectangles all around Q_0 that on the one hand do not overlap with Q_0 and on the other hand touch its borders. That is, there must be a rectangle with bottom left corner $(\sqrt{L/2}, \cdot)$ and another one with bottom left corner $(\cdot, (x_0 + p/\sqrt{L/2}) \bmod p)$, which in particular implies that $\sqrt{L/2}$ and $p/\sqrt{L/2}$ are in \mathbb{N} . For $L > 2$ and a

prime number p , this is impossible. This completes the proof of Lemma 6.2. \square

As in the proof of Lemma 6.1, Lemma 6.2 makes it possible to show for the complete cuckoo graph $G := (V_1, V_2, E) := G(U, h_{a_1, b_1}, h_{a_2, b_2})$ that the neighborhood $\Gamma_j \subseteq V_2$ of a vertex $j \in V_1$ contains a subset $\hat{\Gamma}_j$ of size $\lfloor \sqrt{t/2} \rfloor$ which has the same crucial property as the corresponding set in the proof of Lemma 5.1: $\hat{\Gamma}_j$ can be viewed as a sequence $(n_i(j))_{i \in [\lfloor \sqrt{t/2} \rfloor]}$ that increases modulo m with a step size s' and s'' for fixed values s' and s'' . From here, we complete the argumentation in direct analogy to the proof of Lemma 5.1. The details are as follows.

Consider the complete cuckoo graph $G := (V_1, V_2, E) := G(U, h_{a_1, b_1}, h_{a_2, b_2})$. Observe (for later use) that each vertex in V_1 has a degree of either t or $t+1$. We analyze the neighborhood of an arbitrary vertex $j \in V_1$. For the set $B_j := h_{a_1, b_1}^{-1}(j)$ of keys whose corresponding edges are incident with $j \in V_1$, we have

$$B_j = \{x \in U \mid (a_1 x + b_1) \bmod p_1 \bmod m = j\} = \{x \in U \mid (a_1 x + b_1) \bmod p_1 \in \{im + j \in U \mid i \in \mathbb{N}\}\}.$$

The equation $(a_1 x + b_1) \bmod p_1 = im + j$ has the unique solution $x = (i \cdot a_1^{-1} m + a_1^{-1}(j - b_1)) \bmod p_1$. That is, B_j is a grid set $\{x_i^{(j)} \mid i \in \mathbb{N}, im + j \in U\}$, $x_i^{(j)} := (i \cdot s_1 + c^{(j)}) \bmod p_1$, with step size $s_1 := (a_1^{-1} m) \bmod p_1$ and offset $c^{(j)} := (a_1^{-1}(j - b_1)) \bmod p_1$. Note that $\{x_i^{(j)} \mid i \in [t]\}$ is a subset of B_j , and that the step size s_1 is independent of j .

If we try to compute the neighborhood $\Gamma_j := h_{a_2, b_2}(B_j)$ of j , then we have to deal with arithmetic w. r. t. three distinct moduli. However, for each $j \in V_1$, we apply Lemma 6.2 with $L = t$ and $B = \{x_i^{(j)} \mid i \in [t]\}$: There exists a step size $\hat{s} \in [p_1] - \{0\}$ such that each set $\{x_i^{(j)} \mid i \in [t]\} \subseteq B_j$, $j \in V_1$, contains an arithmetic sequence $(\hat{x}_k^{(j)})_{k \in [l]}$ of length $l \geq \lfloor \sqrt{t/2} \rfloor$ with step size \hat{s} . Now, if we restrict ourselves to considering the neighbors $\hat{\Gamma}_j$ of j that are given by the subset $\hat{B}_j := \{\hat{x}_k^{(j)} \mid k \in [\lfloor \sqrt{t/2} \rfloor]\}$ of B_j , then arithmetic modulo p_1 simply drops out.

Call the edge that corresponds to a key $\hat{x}_i^{(j)} \in \hat{B}_j$ the i -edge of j and call its endpoint in V_2 the i -neighbor of j , or $n_i(j)$. Then we have

$$\begin{aligned} n_i(j) &= h_{a_2, b_2}(\hat{x}_i^{(j)}) \\ &= (a_2 \hat{x}_i^{(j)} + b_2) \bmod p_2 \bmod m \\ &= (a_2(i \cdot \hat{s} + \hat{x}_0^{(j)}) + b_2) \bmod p_2 \bmod m \\ &= (i \cdot a_2 \hat{s} + (a_2 \hat{x}_0^{(j)} + b_2)) \bmod p_2 \bmod m. \end{aligned}$$

Define s as $(a_2 \hat{s}) \bmod p_2$, $o^{(j)}$ as $(a_2 \hat{x}_0^{(j)} + b_2) \bmod p_2$, and $y_i^{(j)}$ as $i \cdot s + o^{(j)}$. Then, by Lemma 5.3 (“leap effect”) applied for $r_1 = p_2$ and $r_2 = m$, we have

$$\begin{aligned} n_i(j) &= y_i^{(j)} \bmod p_2 \bmod m \\ &= (y_i^{(j)} + \lfloor y_i^{(j)} / p_2 \rfloor \cdot \Delta) \bmod m, \end{aligned}$$

where Δ is $(-p_2) \bmod m$. That is, the subsequence $(n_i(j))_{i \in \lfloor \sqrt{t/2} \rfloor}$ of Γ_j increases modulo m with steps of size $s' := s \bmod m$ and $s'' := (s + \Delta) \bmod m$.

Call an i -edge of j a *predecessor edge* if $i \leq \lfloor \sqrt{t/2} \rfloor - 2$, and call a vertex $l \in V_2$ *obstructive* if it is incident with at least five predecessor edges. Then the existence of $m' = \lceil m / (\alpha(t+1)) \rceil$ distinct bad edge sets $K_1, \dots, K_{m'}$ of size 6 such that for all $i = 1, \dots, m'$ the number of bad edge sets K_j , $j \in \{1, \dots, m'\}$, overlapping with K_i is at most 5, follows in complete analogy to the proof of Lemma 5.1:

- (i) Each obstructive vertex $l \in V_2$ belongs to a bad edge set $K^{(l)}$ of size 6 (Figure 2), and for any other obstructive vertex $l' \in V_2$, a corresponding bad edge set $K^{(l')}$ is distinct from $K^{(l)}$.
- (ii) The graph G contains at least $m' = \lceil m / (\alpha(t+1)) \rceil$ obstructive vertices. Here we use that $m/p_1 \leq 1/73$ implies $\lfloor \sqrt{t/2} \rfloor - 1 \geq 5$, and that $p_2 \leq \alpha p_1$ implies a vertex degree of at most $\alpha(t+1)$ in V_2 .
- (iii) We use (i) and (ii) to define the desired bad edge sets $K_1, \dots, K_{m'}$.

□

Using the condition $m/p_1 \geq p_1^{-(1/8-\varepsilon)}$ for an arbitrary constant $\varepsilon \in (0, 1/8)$, an application of Lemma 6.1 that is analogous to the application of Lemma 5.1 in the proof of Theorem 5.1, leads to the desired result:

$$\begin{aligned} p_F &\geq \left(1 + 2^7 \cdot 5 \cdot \alpha \cdot \left(1 - \frac{10}{p_1^{7/8+\varepsilon}} \right)^{-6} \cdot p_1^{-8\varepsilon} \right)^{-1} \\ &= 1 - o(1) \end{aligned}$$

for $m, p_1 \rightarrow \infty$. This completes the proof of Theorem 6.1. □

7 Experiments

We implemented cuckoo hashing in Java™ in a straightforward way, where generation of pseudo-random numbers was done via the Mersenne Twister

Multiplicative class for fixed $k = 22$, uniform random S , $|S| = 0.9 \cdot m$

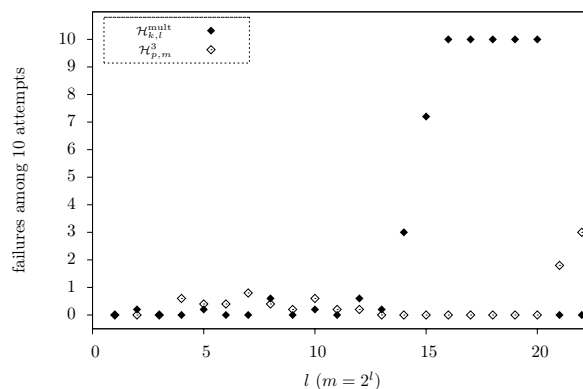


Figure 5: Results for the multiplicative class

from the colt distribution¹. We carried out some experiments in order to obtain estimates of the failure probability by counting average failure frequencies among 5 independently and uniformly random chosen sets S of size $(1 - \delta)m$, each set being inserted 10 times with independently and uniformly random chosen hash functions. This was repeated several times for different settings of the parameters k, l and p, m , respectively, where δ was fixed, as well as for different settings of δ , where k and l were fixed. Figure 5 depicts the results for the multiplicative class as well as for the reference class $\mathcal{H}_{p,m}^3$, i.e., the class of quadratic polynomials, where we fixed $k = 22$, $p = 8388593 \approx 2^{23}$ and $\delta = 0.1$, and repeated the experiment for table size $m = 2^l$, $l = 1, \dots, 22$. Figure 6 shows two results for the linear class, for fixed $p = 2097143 \approx 2^{21}$ and $\delta = 0.1$, as well as changing table size $m = \lceil p/t \rceil$, $t = 2, 3, \dots, 129$: First, only the linear class given by p was used, and second, we used the linear class given by $p_1 = p$ for h_1 and the linear class given by $p_2 = 4194301 \approx 2^{22}$ for h_2 . In case of the hash family being the same for both hash functions, one observes: For the linear class (Figure 6) and the multiplicative class (Figure 5), the failure probability close to 1 for large ratio $m/N < 1/2$ nicely reflect the assertions of Theorems 4.1 and 5.1. In the case of distinct linear classes for h_1 and h_2 (Figure 6) we do not see a significant performance improvement. This corresponds to Theorem 6.1. Note the good performance when quadratic polynomials are used (Figure 5).

¹<http://acs.lbl.gov/~hoschek/colt/>

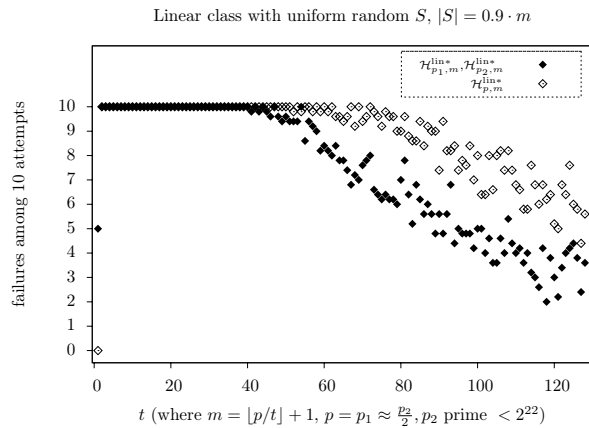


Figure 6: Results for the linear class

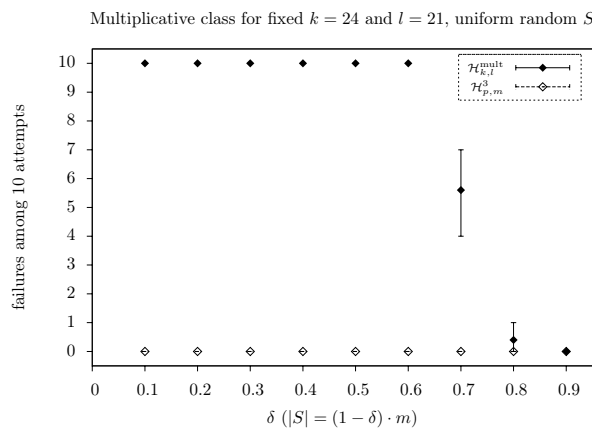


Figure 7: Further results for the multiplicative class

Experiments were carried out also for the multiplicative class with fixed $k = 24$, $l = 21$, and changing $\delta \in \{0.1, 0.2, \dots, 0.9\}$. The result is depicted in Figure 7. It can be seen that random key sets of relatively small size $0.4m$ still seem to be very unlikely to be inserted successfully, if the key set is very dense in the universe.

8 Conclusion

In the case $m/N \geq N^{1-\gamma}$ for a sufficiently small constant $\gamma > 0$, we have answered the question of why cuckoo hashing does not work well with the multiplicative class. We further showed that in the same sense cuckoo hashing performs badly when combined with the common linear class, even if h_1 and h_2 are chosen from distinct linear classes. Cuckoo

hashing should be used with these classes only in case U is sufficiently large compared to m and S does not have keys of a strongly restricted structure. Moreover, our results point out that care must be taken when interpreting the result by Mitzenmacher and Vadhan [12]—one has to check carefully that the hypotheses of the theorems are satisfied.

References

- [1] L. Carter and M. N. Wegman, *Universal Classes of Hash Functions*, J. Comput. Syst. Sci., 18 (1979), pp. 143–154.
- [2] K. Mehlhorn, *Data Structures and Algorithms*, vol. 1, Springer-Verlag, Berlin, 1984.
- [3] M. Mitzenmacher and E. Upfal, *Probability and Computing*, Cambridge University Press, Cambridge, UK, 2005.
- [4] B. Chor and O. Goldreich, *Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity*, SIAM J. Comput., 17 (1988), pp. 230–261.
- [5] D. Zuckerman, *Simulating BPP Using a General Weak Random Source*, Algorithmica, 16 (1996), pp. 367–391.
- [6] M. Dietzfelbinger, T. Hagerup, J. Katajainen, and M. Penttonen, *A Reliable Randomized Algorithm for the Closest-Pair Problem*, J. Algorithms, 25 (1997), pp. 19–51.
- [7] R. Pagh and F. F. Rodler, *Cuckoo Hashing*, in Proc. ESA 2001, LNCS 2161, Springer, 2001, pp. 121–133.
- [8] R. Pagh, *On the Cell Probe Complexity of Membership and Perfect Hashing* in Proc. 33rd Annual Symp. on Theory of Computing, ACM Press, 2001, pp. 425–432.
- [9] L. Devroye and P. Morin, *Cuckoo Hashing: Further Analysis*, Inf. Process. Lett., 86 (2003), pp. 215–219.
- [10] R. Pagh and F. F. Rodler, *Cuckoo Hashing*, J. Algorithms, 51 (2004), pp. 122–144.
- [11] A. Pagh, R. Pagh, and M. Ružić, *Linear Probing with Constant Independence*, in Proc. 39th Annual ACM Symp. on Theory of Computing, ACM Press, 2007, pp. 318–327.
- [12] M. Mitzenmacher and S. Vadhan, *Why Simple Hash Functions Work: Exploiting the Entropy in a Data Stream*, in Proc. 19th Annual ACM-SIAM Symp. on Discrete Algorithms, SIAM, 2008, pp. 746–755.
- [13] J. Cohen and D. M. Kane, *6.856 Project: Bounds on the Independence Required for Cuckoo Hashing*, <http://web.mit.edu/dankane/www/Independence%20Bounds.pdf>.