A Construction Method for Optimally Universal Hash Families and its Consequences for the Existence of RBIBDs

Philipp Woelfel[1]

Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4
Canada

Abstract

We introduce a method for constructing optimally universal hash families and equivalently RBIBDs. As a consequence of our construction we obtain minimal optimally universal hash families, if the cardinalities of the universe and the range are powers of the same prime. A corollary of this result is that the necessary conditions for the existence of an RBIBD with parameters v, k, λ, namely $v \equiv 0 \pmod k$ and $\lambda(v-1) \equiv 0 \pmod{k-1}$, are sufficient, if v and k are powers of the same prime. As an application of our construction, we show that the k-MAXCUT algorithm of Hofmeister and Lefmann [9] can be implemented such that it has a polynomial running time, in the case that the number of vertices and k are powers of the same prime.

1 Introduction and Results

The concept of universal hashing as introduced by Carter and Wegman [4] in 1979 has found a wide variety of applications. Besides of being an important tool for hashing schemes (see e.g. [4,8,7]), universal hashing has been used in many other areas of computer science such as complexity theory, cryptography or algorithmics. The importance of this concept has led to a search for practical hash families with good properties and to the investigation of the combinatorial structure of such hash families. In 1994, Stinson [17] has drawn the first connections between universal hash families and combinatorial designs such as resolvable balanced incomplete block designs or orthogonal

Email address: `pwoelfel@cs.toronto.edu` (Philipp Woelfel).

[1] Supported in part by DFG grant WO1232/1-1

arrays. Later on, more connections to combinatorial designs and other structures as, e.g., authentication codes were discovered. While on one hand, such connections have led to new applications of universal hashing in cryptography or the theory of combinatorial designs, they also allowed new constructions of universal hash families and new ways of analyzing them (for some references see e.g. [18,19,1,3]).

Definition 1 An $(N; u, r)$-*hash family* $\mathcal{H}$ is a family of N functions $U \to R$, where U and R are finite sets of cardinality $u > 1$ and $r > 1$, resp.

If $\mathcal{H}$ is a hash family of functions $U \to R$, then U is called *universe* and R is called *range* of $\mathcal{H}$. We call the elements from the universe *keys*, and say that two keys $x_1, x_2 \in U$ *collide* under a function $h \in \mathcal{H}$, if $h(x_1) = h(x_2)$. The *collision probability* of x_1 and x_2 is the probability that x_1 and x_2 collide under a randomly chosen function in $\mathcal{H}$, i.e., $\mathrm{Prob}_{h\in\mathcal{H}}\big(h(x_1) = h(x_2)\big)$.

It was the original intention behind using hash families in hashing schemes to guarantee that under a randomly chosen hash function the expected number of collisions among a set of keys is low. This motivates the following common definition.

Definition 2 ([4,17]) A hash family with universe U is ϵ-*universal*, if any two distinct keys $x_1, x_2 \in U$ have a collision probability of at most ϵ. A $(1/r)$-universal $(N; u, r)$-hash family is simply called *universal*.

Note that if $r \geq u$, then any injective function $f : U \to R$ forms a 0-universal $(1; u, r)$-hash family $\{f\}$. Therefore, we assume w.l.o.g. that $u \geq r$ if we talk about ϵ-universal hash families. However, we will later introduce other types of hash families (as e.g. Δ-universal hash families), where we also consider the case $r > u$.

Many constructions of ϵ-universal hash families have been proposed, and some of them can be implemented by means of simple arithmetic operations such as multiplication [6] or convolution [12]. But besides the search for efficient hash families, there has also been some interest in investigating the properties of hash families with extremely small cardinalities or collision probabilities.

We denote by *gcd* and *lcm* the greatest common divisor and the least common multiple, resp. It can be shown [15] that no ϵ-universal $(N; u, r)$-hash family exists, if $\epsilon < (u-r)/\big(r(u-1)\big)$, and that for any ϵ-universal $(N; u, r)$-hash family with $\epsilon = (u-r)/\big(r(u-1)\big)$, it is $N \geq (u-1)/\gcd(u-1,\, r-1)$. These properties justify the following definition.

Definition 3 ([15]) Let $u \geq r$. An ϵ-universal $(N; u, r)$-hash family $\mathcal{H}$ is called *optimally universal* or short OU, if $\epsilon = \epsilon_{\text{opt}}(u, r) := (u-r)/\big(r(u-1)\big)$. If in addition $N = (u-1)/\gcd(u-1,\, r-1)$, then $\mathcal{H}$ is called *minimal optimally universal.*

Sarwate [15] has presented constructions of minimal optimally universal $(N; q^n, q^m)$-hash families for any prime power q in the cases $m = 1$ and $m = n - 1$ $(n \geq 1)$. He states though, that while several ad hoc constructions of small OU universal $(N; q^n, q^m)$-hash families can be obtained for many values of n and m, there is no general construction method known that produces minimal OU hash families for all n and m. The main purpose of this paper is to present such a general construction method, as summarized by the following theorem.

Theorem 1 *Let q be an arbitrary prime power. For any $1 \leq m \leq n$ there exists a minimal optimally universal $(N; q^n, q^m)$-hash family.*

The construction is based on the definition of a new type of hash families, called *partitioned universal.* Our proof does not only show the existence of the optimally universal hash families, but describes in fact an efficient algorithm for constructing them. Moreover, the resulting hash functions can be evaluated fairly simple by means of finite field arithmetic. We show below, that our construction has algorithmic consequences as well as consequences for the existence of certain important combinatorial objects.

Definition 4 A *balanced incomplete block design* (short: BIBD) with parameters v, k, λ $(v, k, \lambda \in \mathbb{N})$ is a pair $(V, \mathcal{B})$, where V is a v-set of *points* and $\mathcal{B}$ is a set of k-subsets of V called *blocks*, such that for any two distinct points $p, p' \in V$ there are exactly λ blocks $b \in \mathcal{B}$ where $\{p, p'\} \subseteq b$. If some blocks form a partition of V, then the set of these blocks is called *parallel class.* A BIBD is *resolvable*, if its blocks can be partitioned into parallel classes. A resolvable BIBD with parameters v, k, λ is denoted by $\text{RBIBD}_\lambda[k; v]$.

RBIBDs are well investigated combinatorial structures and a lot of research has been spent on finding RBIBDs with certain parameters or on disproving their existence (a broad overview on results can be found in the monographs [2,5]). BIBDs and RBIBDs have also some algorithmic applications. E.g., the construction of layouts for redundant disk arrays in [10,16] is based on BIBDs and Hofmeister and Lefmann [9] show how to find a large k-cut in a graph with n vertices using an $\text{RBIBD}_\lambda[k; n]$. The authors complain, though, that although various algebraic construction methods for BIBDs are known in the literature, not a lot of attention has been paid to their exact running times.

Besides of the algorithmic problem of constructing RBIBDs, it is an important open question of design theory, for which parameters RBIBDs do exist. It is well-known that the following two conditions are necessary for the existence of an $\mathrm{RBIBD}_\lambda[k; v]$:

$$v \equiv 0 \pmod{k} \tag{C1}$$

and

$$\lambda(v-1) \equiv 0 \pmod{k-1}. \tag{C2}$$

It is easy to see that the second condition is fulfilled if and only if λ is a multiple of $\lambda_{\min}(k, v) := (k-1)/\gcd(k-1, v-1)$. Thus, $\lambda_{\min}(k, v)$ is the minimal value for λ such that an RBIBD with parameters v, k, λ may exist.

As Stinson [17] has shown, an OU hash family $\mathcal{H}$ of functions $U \to R$ describes an RBIBD, by taking U as point set and each set $h^{-1}(y)$ with $y \in R$ and $h \in \mathcal{H}$ as a block. Clearly, for a fixed h, the blocks $h^{-1}(y)$ with $y \in R$ form a parallel class. Taking into account that in an OU hash family any pair of keys has the same collision probability [15], it is easy to see that this block design is in fact an RBIBD. This construction can also be reversed such that one obtains from any RBIBD an OU hash family.

Theorem 2 ([17]) *Let $u \geq r$. An optimally universal $(N; u, r)$-hash family exists if and only if there is an $\mathrm{RBIBD}_\lambda[k; v]$ with $v = u$, $k = u/r$, and $\lambda = N(u-r)/\big(r(u-1)\big)$.*

Plugging the minimal possible λ-value, $\lambda_{\min}$, into this theorem, it is easy to see that a minimal optimally universal $(N; u, r)$-hash family exists if and only if there exists an $\mathrm{RBIBD}_\lambda[u/r; u]$ with $\lambda = \lambda_{\min}(u/r, u)$. Using this equivalence, our construction method of minimal OU hash families from Theorem 1 implies the existence of an $\mathrm{RBIBD}_{\lambda_{\min}(k,v)}[k; v]$ for any $k = q^{n-m}$ and $v = q^n$ where q is a prime power (and $m \leq n$). Since in addition any λ satisfying $(C2)$ is a multiple of $\lambda_{\min}(k, v)$, for any such λ an $\mathrm{RBIBD}_\lambda[k; v]$ can be obtained by taking multiple copies of an $\mathrm{RBIBD}_{\lambda_{min}(k,v)}[k; v]$.

Corollary 1 *The necessary conditions $(C1)$ and $(C2)$ for the existence of an $\mathrm{RBIBD}_\lambda[k; v]$ are sufficient, if k and v are powers of the same prime.*

According to the author's knowledge, such a general result about the existence of "optimal" RBIBDs for a wide range of parameters was not known, before. It should be noted, though, that the RBIBDs we obtain are not necessarily simple, i.e. they may contain repeated blocks. In fact, our construction of OU hash families requires to take several hash functions multiple times which yields multiple copies of the same resolvable classes in the corresponding RBIBDs.

Due to the fact that the hash families from Theorem 1 can be constructed efficiently, we also obtain an efficient algorithm for constructing these RBIBDs.

Therefore, they may be useful for algorithmic problems, as we demonstrate in the next section.

2 Application to k-MAXCUT

In [9], Hofmeister and Lefmann have presented a deterministic algorithm for computing a large k-cut in a graph, using RBIBDs. Let $G = (V, E)$ be an undirected graph whose edges are weighted by a function $w : E \to \mathbb{N}_0$. By $w(G)$ we denote the sum of edge weights, i.e., $w(G) = \sum_{e \in E} w(e)$. A *balanced k-cut* of G is a partition of the vertices V into k sets $V_1, \ldots, V_k$ of equal size. The *cut size* of a k-cut $(V_1, \ldots, V_k)$ is the sum of weights of all *crossing* edges, that is edges $e = \{v, w\}$ with $(v, w) \in V_i \times V_j$ with $i \neq j$.

Assume that an RBIBD$_\lambda[n; k]$ is given by a description of the blocks, and that the list of these blocks is already arranged in such a way, that all blocks of each parallel class appear one after the other. Hofmeister and Lefmann have shown that in this case a balanced k-cut of size at least $w(G) \cdot \frac{k-1}{k} \cdot (1 + \frac{1}{n-1})$ can be computed for any graph G with n vertices in time $O\Big(k(n + m) + \lambda n^2\Big)$. Note that for any constant $0 < \epsilon < 1/(k - 1)$, $k \geq 2$, it is already NP-complete to decide whether a graph with m edges admits a k-cut of size at least $m \cdot \frac{k-1}{k} \cdot (1 + \epsilon)$ [13].

The algorithm of Hofmeister and Lefmann relies on having the explicit description of an RBIBD with appropriate parameters at hand. Since most known RBIBDs are ad-hoc constructions and algorithms for the construction of RBIBDs for a wide variety of parameters are rarely described in literature, it is not clear for which parameters the algorithm can be implemented. In fact, the only statement Hofmeister and Lefmann can make is that for n being a multiple of 3 and n, $n - 1$ or $n - 2$ being a prime power, a polynomial time algorithm exists for the construction of the desired RBIBDs.

Using our construction of optimally universal hash families, we obtain now explicit implementations of their algorithm for all n and k being powers of the same prime. We assume a RAM model allowing arithmetics over natural numbers and finite fields of order at most n in constant time.

Corollary 2 *If n and k are powers of the same prime, then for any graph with n vertices and m edges a balanced k-cut of weight at least $w(G) \cdot \frac{k-1}{k} \cdot (1 + \frac{1}{n-1})$ can be computed in time $O\Big(k(n + m) + k \cdot n^2/\gcd(n - 1,\, k - 1)\Big)$.*

Note that this result is obtained by simply plugging our RBIBD construction into the algorithm of Hofmeister and Lefmann. Furthermore, since we use

RBIBDs with the minimal possible parameter λ, we cannot improve on this result by using the same algorithm with other RBIBDs. Nevertheless, we can do better if we want to find k-cuts with $k > \sqrt{n}$ by a very similar and simple algorithm using OU hash families.

Let $G = (V, E)$ be a graph with n vertices and let $\mathcal{H}$ be an ϵ-universal $(N; n, k)$-hash family with universe V and range $\{1, \ldots, k\}$. Any hash function $h \in \mathcal{H}$ defines a cut $(V_1, \ldots, V_k)$ on the graph, by letting $V_i = \{v \in V \mid h(v) = i\}$. Moreover, if $\mathcal{H}$ is optimally universal, then the cut is balanced, because it is well-known that in this case $|h^{-1}(i)| = n/k$ (see e.g. [15]). Now choose a random hash function h from $\mathcal{H}$ and consider the cut $(V_1, \ldots, V_k)$ defined by h. Since $\mathcal{H}$ is ϵ-universal, each edge is a crossing edge with a probability of at least $1 - \epsilon$. Hence, by the linearity of expectation, the expected cut size is bounded below by $(1 - \epsilon)w(G)$. This means that there exists a hash function $h \in \mathcal{H}$ which defines a cut having at least that size. Using $\epsilon = \epsilon_{\text{opt}}(n, k) = (n - k)/\big(k(n - 1)\big)$ if $\mathcal{H}$ is optimally universal, yields an edge weight of at least $w(G) \cdot \Big(1 - \frac{n-k}{k(n-1)}\Big) = w(G) \cdot \frac{k-1}{k} \cdot (1 + \frac{1}{n-1})$.

As we will show in the remainder of this paper, the hash functions from the minimal OU hash family $\mathcal{H}$ claimed to exist in Theorem 1 can be enumerated in time $O(N)$, where $N = (n - 1)/\gcd(n - 1,\, k - 1)$. Moreover, under the assumption of a RAM where (finite field) arithmetics can be done in constant time, the hash functions can be evaluated in constant time. Once we have picked a hash function $h \in \mathcal{H}$, we can evaluate the hash function values of all n vertices in time $O(n)$ in order to compute the corresponding cut $(V_1, \ldots, V_k)$. Now, for each set V_i, the sum of weights of edges within V_i (that is edges $e = \{v, w\}$ with $v, w \in V_i$), can be computed in time $O\Big(n^2/k^2\Big)$ (assuming that the graph is given by an adjacency matrix). This way, we can compute the sum $s(V_1, \ldots, V_k)$ of weights of all non-crossing edges in time $O(n^2/k)$ and the k-cut with the minimal value $s(V_1, \ldots, V_k)$ has the maximal weight $w(G) - s(V_1, \ldots, V_k)$. Hence, the total running time is bounded by $O(N \cdot n^2/k) = O\Big(n^3/\big(k \cdot \gcd(n - 1,\, k - 1)\big)\Big)$. For $k > \sqrt{n}$ this is better than the result of Corollary 2. It should be noted, though, that the algorithm we have just described computes a cut but not its weight, because for some parameters the time bound may not allow us to spend $O(n^2)$ time for computing $w(G)$.

Theorem 3 *If n and k are powers of the same prime, then for any graph with n vertices given by an adjacency matrix, a balanced k-cut of weight at least $w(G) \cdot \frac{k-1}{k} \cdot (1 + \frac{1}{n-1})$ can be computed in time $O\Big(n^3/\big(k \cdot \gcd(n - 1,\, k - 1)\big)\Big)$.*

3 Partitioned Universal Hash Families

In the following, we define a new type of hash families, called partitioned universal. They are the main ingredients for our construction of optimally universal hash families.

Definition 5 An $(N; u, r)$-hash family is *partitioned ϵ-universal* if it is ϵ-universal and if there exists an equivalence relation partitioning the universe in equivalence classes of size r such that any two distinct keys from the same equivalence class have a collision probability of 0. In the case $\epsilon = 1/r$ the hash family is simply called *partitioned universal*.

In the remainder of the text we use the convention that if the universe of a partitioned universal hash family with range R is written as $W \times R$, then the equivalence classes to which the above definition refers to, are the classes U_x, $x \in W$, where $U_x = \{(x, x') \mid x' \in R\}$.

While we have defined partitioned ϵ-universal hash families mainly in order to prove Theorem 1, there are several examples of well-known ϵ-universal hash families, which in fact turn out to be ϵ-partitioned universal. One example where this is the case is the multiplicative $(2/r)$-universal hash family defined in [6]. As we will see in the following, partitioned universal hash families are quite easy to construct by means of another type of hash families, defined by Stinson in 1996.

Definition 6 ([19]) Let $\mathcal{H} : U \to R$ be an $(N; u, r)$-hash family, where R is an additive abelian group. $\mathcal{H}$ is called *ϵ-Δ-universal* if for any two distinct keys $x_1, x_2 \in U$ and any $d \in R$, $\mathrm{Prob}_{h \in \mathcal{H}}\left(h(x_1) - h(x_2) = d\right) \leq \epsilon$. In the case $\epsilon = 1/r$, $\mathcal{H}$ is simply called Δ-universal.

Although not explicitly defined there, Δ-universal hash families have already been used by Carter and Wegman in 1979 [4] in order to construct universal hash families for long keys. In the last years, they have found applications mainly in message authentication [11,14], and Stinson [19] has used them to construct strongly universal hash families (a restricted type of universal hash families).

Here is a well-known construction (see e.g. [19]): Denote by $\mathbb{F}_q$ a finite field of order q and let $U = (\mathbb{F}_q)^n$ and $R = (\mathbb{F}_q)^m$ be extension fields of $\mathbb{F}_q$. If $\phi : U \to R$ is a surjective homomorphism then it is easy to see that the family $\{f_a \mid a \in \mathbb{F}_{q^n}\}$, where $f_a : U \to R$, $x \mapsto \phi(ax)$ is Δ-universal. If $n < m$, then any Δ-universal $(q^m; q^m, q^m)$-hash family is also Δ-universal for an arbitrary q^n-element subset of the universe. Thus, one gets the following result.

Lemma 1 ([19]) *Let q be a prime power. For any positive integers n and m, there exists a Δ-universal $(N; q^n, q^m)$-hash family, where $N = \max\{q^n, q^m\}$.*

Using ϵ-Δ-universal hash families, it is very easy to construct partitioned ϵ-universal ones as the following lemma shows.

Lemma 2 *If there is an ϵ-Δ-universal $(N; u, r)$-hash family, then there is also a partitioned ϵ-universal $(N; ur, r)$-hash family.*

Proof: Let $\mathcal{H} : U \to R$ be an ϵ-Δ-universal hash family where $|U| = u$ and $|R| = r$. For each $h \in \mathcal{H}$ we define the mapping $f_h : U \times R \to R$, $(x_1, x_2) \mapsto h(x_1) + x_2$. Then $\mathcal{F} = \{f_h \mid h \in \mathcal{H}\}$ is the desired partitioned ϵ-universal $(N; ur, r)$-hash family. To see this, define the equivalence relation $\sim$ on $U \times R$ as $(x_1, x_2) \sim (x_1', x_2') \Leftrightarrow x_1 = x_1'$. Clearly, two different keys (x_1, x_2) and (x_1', x_2') collide if and only if $h(x_1) - h(x_1') = x_2' - x_2$. Because $\mathcal{H}$ is ϵ-Δ-universal, this is for $x_1 \neq x_1'$ the case with a probability of at most ϵ. If $x_1 = x_1'$, then both keys belong to the same equivalence class and they differ only in the second component. Thus, they do not collide under any function in $\mathcal{F}$. Since the size of each equivalence class is r, $\mathcal{F}$ is partitioned ϵ-universal. $\square$

It is well-known that u/r is a lower bound on the size N of a $(1/r)$-universal $(N; u, r)$-hash family (see e.g. [18]). This is clearly not a tight lower bound for the case $r < u < r^2$, since for $r < u$, there is at least one key pair whose collision probability is not 0, and thus at least r hash functions are required to obtain a collision probability of at most $1/r$.

Definition 7 A (partitioned) universal $(N; u, r)$-hash family is called *minimal* if

$$N = \begin{cases} 1 & \text{if } u \leq r \\ \max\{u/r,\ r\} & \text{otherwise.} \end{cases}$$

Note that for $u \leq r$, a minimal partitioned universal $(N; u, r)$-hash family is the set consisting of the identity, only. Using Lemma 1 and Lemma 2 for the case $u > r$, we obtain minimal partitioned universal hash families for all u and r which are powers of the same prime.

Corollary 3 *Let q be a prime power. For any positive integers n and m, there there exists a minimal partitioned universal $(N; q^n, q^m)$-hash family .*

We present now two construction methods, which can later be combined to recursively construct the minimal OU hash families from Theorem 1. Both of them combine two hash families in order to obtain a new OU $(N; u, r)$-hash family, where $u = k \cdot r$ (note that $r|u$ is necessary for the existence for an OU hash family with universe size u and range size r). For the first construction method we assume $u|r^2$ and we combine a partitioned universal $(N_1; u, r)$-hash family with an OU $(N_2; r, r^2/u)$-hash family. The second method deals with the case $u \geq r$ and combines a $(N_1; u, r)$-hash family with an OU $(N_2'; u/r, r)$-hash family.

Lemma 3 *Let $u = k^2 \cdot \ell$ and $r = k \cdot \ell$. If $\mathcal{G}$ is a partitioned universal $(N_1; k^2\ell, k\ell)$-hash family and $\mathcal{F}$ is an optimally universal $(N_2; k\ell, \ell)$-hash family, then there is an optimally universal $(N; u, r)$-hash family $\mathcal{H}$, where*

$$N \;=\; \frac{N_1 N_2 (u-1)}{\gcd\big(N_1(r-1),\, N_2(u-r)\big)}.$$

Moreover, if $\mathcal{G}$ and $\mathcal{F}$ are both minimal, then so is $\mathcal{H}$.

For the proof, the following simple statement is needed.

Remark 1 *If $m, n \in \mathbb{N}$ and n is a multiple of m, then* $\gcd(m-1,\, n-1) = \gcd(m-1,\, n/m-1)$.

Proof: Using the euclidian algorithm we obtain that $\gcd(m-1,\, n-1)$ equals $\gcd\Big(m-1,\, n-1-\frac{n}{m}(m-1)\Big) = \gcd\Big(m-1,\, \frac{n}{m}-1\Big)$. $\square$

Proof of Lemma 3: Let K and L be sets of cardinality k and ℓ, resp. We may assume w.l.o.g. that $\mathcal{G}$ consists of mappings $K \times R \to R$, where $R = K \times L$, and $\mathcal{F}$ consists of mappings $K \times L \to L$. Let $z = \mathrm{lcm}\big(N_1(r-1),\, N_2(u-r)\big)$ and $z_1 = z/\big(N_1(r-1)\big)$ as well as $z_2 = z/\big(N_2(u-r)\big)$. Let the family $\mathcal{G}'$ consist of z_1 copies of each hash function in $\mathcal{G}$ and the family $\mathcal{F}'$ consist of z_2 copies of the hash functions in $\mathcal{F}$. For each function $f \in \mathcal{G}' \cup \mathcal{F}'$ we define the mapping

$$h_f : K \times R \to K \times L, \quad (x_1, x_2) \mapsto \begin{cases} f(x_1, x_2) & \text{if } f \in \mathcal{G}', \\ \big(x_1, f(x_2)\big) & \text{if } f \in \mathcal{F}'. \end{cases}$$

Finally, the hash family $\mathcal{H}$ consists of the functions h_f with $f \in \mathcal{G}' \cup \mathcal{F}'$.

In the following, we first show that the cardinality of $\mathcal{H}$ is in fact N and then prove that $\mathcal{H}$ is optimally universal. Finally, we give evidence that $\mathcal{G}$ and $\mathcal{F}$ being minimal implies $N = (u-1)/\gcd(u-1,\, r-1)$ (i.e., $\mathcal{H}$ has the size of a minimal OU hash family).

The following computation shows that $|\mathcal{H}| = N$. Using the identity $\mathrm{lcm}(m,\, n) = mn/\gcd(m,\, n)$ we obtain

$$z \;=\; \frac{N_1(r-1) \cdot N_2(u-r)}{\gcd\big(N_1(r-1),\, N_2(u-r)\big)}.$$

Therefore,

$$\begin{aligned}|\mathcal{H}| = z_1N_1 + z_2N_2 \;&=\; \frac{z}{r-1} + \frac{z}{u-r} \;=\; z \cdot \frac{u-1}{(r-1)(u-r)} \\ &= \frac{N_1N_2(r-1)(u-r)}{\gcd\left(N_1(r-1),\, N_2(u-r)\right)} \cdot \frac{u-1}{(r-1)(u-r)} \;=\; N.\end{aligned}$$

Now we prove that $\mathcal{H}$ is optimally universal. Let $c = |\mathcal{G}'|/|\mathcal{F}' \cup \mathcal{G}'|$, which is the probability, that a randomly chosen $f \in \mathcal{G}' \cup \mathcal{F}'$ is an element of $\mathcal{G}'$. Hence,

$$c = \frac{z_1N_1}{z_2N_2 + z_1N_1} = \frac{1/(r-1)}{1/(u-r) + 1/(r-1)} = \frac{u-r}{r-1+u-r} = \frac{u-r}{u-1}. \tag{1}$$

Thus, by definition of optimally universality it suffices to show that any two distinct keys $x = (x_1, x_2)$ and $x' = (x'_1, x'_2)$ in $K \times R$ have a collision probability of at most c/r.

Assume first $x_1 = x'_1$, thus $x_2 \neq x'_2$. Since in this case x and x' are elements of the same equivalence class with respect to the partitioned universality of $\mathcal{G}'$, they do not collide under any function h_f with $f \in \mathcal{G}'$. Under the condition that $f \in \mathcal{F}'$, it follows from $\mathcal{F}'$ being OU that $h_f(x)$ equals $h_f(x')$ with a probability of at most $(r-\ell)/\big(\ell(r-1)\big)$. Therefore, we have for randomly chosen $f \in \mathcal{G}' \cup \mathcal{F}'$ a total collision probability of at most

$$(1-c) \cdot \frac{r-\ell}{\ell(r-1)} = \frac{r-1}{u-1} \cdot \frac{r-\ell}{\ell(r-1)} = \frac{r-\ell}{\ell(u-1)} = \frac{rk-\ell k}{k\ell(u-1)} = \frac{u-r}{r(u-1)} = \frac{c}{r}.$$

Let now $x_1 \neq x'_1$. Under the condition that f was chosen from $\mathcal{G}'$, the collision probability of x and x' is at most $1/r$. If on the other hand f was chosen from $\mathcal{F}'$, then the keys do not collide at all, which follows straight from the definition of h_f. Therefore, we again have a total collision probability (for f chosen randomly from $\mathcal{G}' \cup \mathcal{F}'$) of at most c/r.

It remains to show that if $\mathcal{G}$ and $\mathcal{F}$ are minimal, then so is $\mathcal{H}$. Since by assumption we have $u = kr$ and $r = k\ell$, obviously $r \leq u \leq r^2$. Thus, by minimality of $\mathcal{G}$ we have $N_1 = \min\{u/r,\, r\} = r$. Further, it follows from the minimality of $\mathcal{F}$ using Remark 1

$$N_2 \;=\; \frac{r-1}{\gcd\left(r-1,\, \ell-1\right)} \;=\; \frac{r-1}{\gcd\left(r/\ell-1,\, \ell-1\right)} \;=\; \frac{r-1}{\gcd\left(k-1,\, r/k-1\right)}. \tag{2}$$

Therefore, we obtain

$$(u-r)N_2 \;=\; r(k-1) \cdot \frac{r-1}{\gcd\left(k-1,\, r/k-1\right)} \;=\; r(r-1) \cdot \frac{\operatorname{lcm}\left(k-1,\, r/k-1\right)}{r/k-1}.$$

Since the last fraction obviously is an integer, it follows that $N_2(u-r)$ is a multiple of $r(r-1)$. Using $N_1 = r$, this implies

$$\gcd\Big(N_1(r-1),\, N_2(u-r)\Big) \;=\; N_1(r-1).$$

By the already proven result on the cardinality of $\mathcal{H}$, we obtain

$$|\mathcal{H}| \;=\; N \;=\; \frac{N_1 N_2(u-1)}{\gcd\left(N_1(r-1),\, N_2(u-r)\right)} \;=\; \frac{N_1 N_2(u-1)}{N_1(r-1)}.$$

Using equation (2), this simplifies to

$$N \;=\; \frac{r-1}{\gcd\left(k-1,\, r/k-1\right)} \cdot \frac{u-1}{r-1} \;=\; \frac{u-1}{\gcd\left(k-1,\, r/k-1\right)}.$$

Finally, by Remark 1 it is true that

$$\begin{aligned}\gcd\left(u-1,\, r-1\right) \;=\; \gcd\left(u/r-1,\, r-1\right) \;=\; &\gcd\left(k-1,\, r-1\right)\\ =\; &\gcd\left(k-1,\, r/k-1\right),\end{aligned}$$

and it follows

$$N \;=\; \frac{u-1}{\gcd\left(u-1,\, r-1\right)}.$$

This shows that $\mathcal{H}$ is minimal. $\square$

Now we come to the case $u \geq r^2$.

Lemma 4 *Let $u = kr$ and $k \geq r$. If $\mathcal{G}$ is a partitioned universal $(N_1; u, r)$-hash family and $\mathcal{F}$ is an optimally universal $(N_2; k, r)$-hash family, then there exists an optimally universal $(N; u, r)$-hash family $\mathcal{H}$, where*

$$N \;=\; \frac{N_1 N_2(ur-r)}{\gcd\Big(N_1(u-r),\, N_2(ur-u)\Big)}.$$

Moreover, if $\mathcal{G}$ and $\mathcal{F}$ are both minimal, then so is $\mathcal{H}$.

Proof: Let K be a set of size k and assume w.l.o.g. that $\mathcal{G}$ and $\mathcal{F}$ consist of mappings $K \times R \to R$ and $K \to R$, resp. Let $z = \operatorname{lcm}\Big(N_1(u-r),\, N_2(ur-u)\Big)$ and $z_1 = z/\Big(N_1(u-r)\Big)$ as well as $z_2 = z/\Big(N_2(ur-u)\Big)$. Further, let the family $\mathcal{G}'$ consist of z_1 copies of each function in $\mathcal{G}$ and the family $\mathcal{F}'$ consist of z_2 copies of each function in $\mathcal{F}$. For each $f \in \mathcal{G}' \cup \mathcal{F}'$ we define the hash function

$$h_f : K \times R \to R, \quad (x_1, x_2) \mapsto \begin{cases} f(x_1, x_2) & \text{if } f \in \mathcal{G}', \\ f(x_1) & \text{if } f \in \mathcal{F}'. \end{cases}$$

Finally, let $\mathcal{H}$ be the family of functions h_f with $f \in \mathcal{G}' \cup \mathcal{F}'$.

In the following, we first show that the cardinality of $|\mathcal{H}|$ is in fact N and after that we prove that $\mathcal{H}$ is optimally universal. Finally, we show that $\mathcal{H}$ is minimal if $\mathcal{G}$ and $\mathcal{F}$ are minimal.

We start by computing the cardinality of $|\mathcal{H}|$. By definition we have

$$z \;=\; \frac{N_1(u-r)\cdot N_2(ur-u)}{\gcd\Big(N_1(u-r),\; N_2(ur-u)\Big)}.$$

Therefore,

$$|\mathcal{H}| \;=\; z_1N_1 + z_2N_2 \;=\; \frac{z}{u-r} + \frac{z}{ur-u} \;=\; z\cdot\frac{ur-r}{(u-r)(ur-u)}$$

$$=\; \frac{N_1N_2(ur-r)}{\gcd\Big(N_1(u-r),\; N_2(ur-u)\Big)} \;=\; N.$$

We now show that $\mathcal{H}$ is optimally universal. Let $\epsilon = |\mathcal{F}'|/|\mathcal{F}'\cup\mathcal{G}'|$. Then

$$\epsilon \;=\; \frac{z_2N_2}{z_1N_1+z_2N_2} \;=\; \frac{1/(ur-u)}{1/(u-r)+1/(ur-u)} \;=\; \frac{u-r}{ur-u+u-r} \;=\; \frac{u-r}{ur-r}.$$

Thus, it suffices to show that any two distinct keys $x=(x_1,x_2)$ and $x'=(x_1',x_2')$ in $K\times R$ collide under a randomly chosen function in $\mathcal{H}$ with a probability of at most ϵ. Assume first that $x_1 = x_1'$ and thus $x_2 \neq x_2'$. Then, by partitioned universality of $\mathcal{G}$, x and x' do not collide under any function in $\mathcal{G}'$ but under all functions in $\mathcal{F}'$. Therefore, the collision probability is exactly ϵ. Let now $x_1 \neq x_1'$. Under the condition $f\in\mathcal{G}'$ the keys collide with a probability of at most $1/r$. If on the other hand $f\in\mathcal{F}'$, then the collision probability is at most $(k-r)/\big(r(k-1)\big)$ since $\mathcal{F}'$ is optimally universal. Therefore, the probability that $h_f(x)$ equals $h_f(x')$ for a randomly chosen $f\in\mathcal{H}$ is bounded above by

$$(1-\epsilon)\cdot\frac{1}{r}+\epsilon\cdot\frac{k-r}{kr-r} = \left(1-\frac{u-r}{ur-r}\right)\cdot\frac{1}{r}+\frac{u-r}{ur-r}\cdot\frac{u/r-r}{u-r} = \frac{u-r}{ur-r} = \epsilon. \quad (3)$$

It remains to prove that $\mathcal{H}$ is minimal if $\mathcal{G}$ and $\mathcal{F}$ are minimal. It is

$$N_2(ur-u) \;=\; \frac{k-1}{\gcd\,(k-1,\, r-1)}\cdot u(r-1) \;=\; \operatorname{lcm}\,(k-1,\, r-1)\cdot u$$

$$=\; u\cdot\operatorname{lcm}\,(u/r-1,\, r-1)\,.$$

Since $N_1 = u/r$ (because of $u/r \geq r$), we have $N_1(u-r) = u\,(u/r-1)$. So, clearly $N_2(ur-u)$ is a multiple of $N_1(u-r)$ and thus $\gcd\big(N_1(u-r),\; N_2(ur-u)\big) =$

$N_1(u-r)$. By the already proven value for N, we obtain

$$N \;=\; \frac{N_1 N_2(ur-r)}{N_1(u-r)} \;=\; \frac{ur-r}{u-r}\cdot\frac{u/r-1}{\gcd\,(u/r-1,\; r-1)} \;=\; \frac{u-1}{\gcd\,(u-1,\; r-1)}.$$

This shows, that $\mathcal{H}$ is minimal. $\square$

We can now combine the two construction methods in order to prove our main result.

Proof of Theorem 1: Let q be an arbitrary prime power. We show by induction on n that for all $1 \le m \le n$ there exists a minimal optimally universal hash family $U \to R$, where U and R are sets of cardinalities $u = q^n$ and $r = q^m$, resp. Note that the statement in Theorem 1 merely claims the existence of the hash families. However, the following proof in fact shows how to construct such a minimal OU $(N; q^n, q^m)$-hash family, which we call $\mathcal{H}^q_{n,m}$.

We use as the universe and range the extension fields $U = (\mathbb{F}_q)^n$ and $R = (\mathbb{F}_q)^m$, resp. First, we define a minimal partitioned universal hash family $\mathcal{G}^q_{n,m}$ with universe $K \times R$, where $K = (\mathbb{F}_q)^{n-m}$. Let $\ell = \max\{n-m,\, m\}$ and $\phi : (\mathbb{F}_q)^\ell \to (\mathbb{F}_q)^m$ be a projection from $(\mathbb{F}_q)^\ell$ to m arbitrary coordinates of the extension field. For all $m \le n$ the hash family $\mathcal{G}^q_{n,m}$ consists of the functions

$$f_a : (\mathbb{F}_q)^{n-m} \times (\mathbb{F}_q)^m \to (\mathbb{F}_q)^m, \quad (x_1, x_2) \mapsto \phi(a \cdot \mu(x_1)) + x_2,$$

where $a \in (\mathbb{F}_q)^\ell$ and $\mu : (\mathbb{F}_q)^{n-m} \to (\mathbb{F}_q)^\ell$ is an aribtrary injective mapping (the identity in the case $n - m = \ell$). Note that for the computation of f_a the multiplication is in the field $(\mathbb{F}_q)^\ell$ and the addition is in $(\mathbb{F}_q)^m$. With the same arguments as in the discussion before Lemma 1 and in the proof of Lemma 2 it follows that $\mathcal{G}^q_{n,m}$ is partitioned universal. Moreover, $\left|\mathcal{G}^q_{n,m}\right| = \max\{q^m,\, q^{n-m}\}$, and thus $\mathcal{G}^q_{n,m}$ is even minimal partitioned universal.

Now we recursively construct the optimally universal hash family $\mathcal{H}^q_{n,m}$. If $m = n$, then $\mathcal{H}^q_{n,m}$ contains only the identity $id : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^n$, which is obviously minimal optimally universal. Hence, for $n = 1$ we obtain a trivial hash family $\mathcal{H}^q_{1,1}$. Let now $1 \le m < n$ and assume that the hash families $\mathcal{H}^q_{m',n'}$ have been constructed for all $1 \le m' \le n' < n$. In the case $m < n \le 2m$, we choose $\mathcal{H}^q_{n,m}$ as the union of z_1 copies of $\mathcal{G}^q_{n,m}$ and z_2 copies of all mappings $(\mathbb{F}_q)^{n-m} \times (\mathbb{F}_q)^m \to (\mathbb{F}_q)^{n-m} \times (\mathbb{F}_q)^{2m-n}$, $(x_1, x_2) \mapsto \big(x_1, f(x_2)\big)$, with $f \in \mathcal{H}^q_{m,2m-n}$, where z_1 and z_2 are the integers determined in the proof of Lemma 3. In the case $n > 2m$, we choose $\mathcal{H}^q_{n,m}$ as the union of z'_1 copies of $\mathcal{G}^q_{n,m}$ and z'_2 copies of the mappings $(\mathbb{F}_q)^{n-m} \times (\mathbb{F}_q)^m \to (\mathbb{F}_q)^m$, $(x_1, x_2) \mapsto f(x_1)$, with $f \in \mathcal{H}^q_{n-m,m}$, where z'_1 and z'_2 are the integers determined in the proof of Lemma 4. Then, according to the proofs of Lemma 3 and Lemma 4 and by the induction hypothesis, $\mathcal{H}^q_{n,m}$ is minimal optimally universal. $\square$

We finally remark that the hash functions from $\mathcal{H}^q_{n,m}$ may in fact be evaluated by simple finite field arithmetic. By the above proof it is easy to see that each hash function $h \in \mathcal{H}^q_{n,m}$ has the form $g_{i,j}(x_1,\ldots,x_n) \mapsto \big(x_1,\ldots,x_i,f(x_{i+1},\ldots,x_j)\big)$, where either $i = j$ and $g_{i,j}$ is the identity, or $i < j$ and f is a function in $\mathcal{G}^q_{j-i,m-i}$. Hence, h can essentially be evaluated by one finite field multiplication and one finite field addition. Although the evaluation of these hash functions is simple, there seems to be no obvious uniform algorithm allowing us to choose a function from $\mathcal{H}^q_{n,m}$ in constant time. Nevertheless, it is easy to enumerate all hash functions in $\mathcal{H}^q_{n,m}$ in time $\left|\mathcal{H}^q_{n,m}\right|$. This means also that the blocks of the corresponding $\mathrm{RBIBD}_\lambda[k;v]$ can be constructed in linear time (with respect to the size of their description) ordered by the parallel classes (recall that a parallel class in the RBIBD corresponds to a hash function in the hash family). Therefore, the algorithm of Hofmeister and Lefmann can be used to compute a k-cut in a graph as stated in Corollary 2 and our modified algorithm has in fact the running time stated in Theorem 3.

Acknowledgements

The author thanks Malcolm Greig for providing helpful information regarding RBIBDs and Ingo Wegener for comments on an early version of the proofs.

References

[1] M. Atici and D. R. Stinson. Universal hashing and multiple authentication. In *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pp. 16–30. 1996.

[2] T. Beth, D. Jungnickel, and H. Lenz. *Design Theory*, volume 1. Cambridge University Press, second edition, 1999.

[3] J. Bierbrauer. Universal hashing and geometric codes. *Designs, Codes and Cryptography*, 11:207–221, 1997.

[4] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[5] C. J. Colbourn and J. H. Dinitz, editors. *The CRC Handbook of Combinatorial Designs.* CRC Press, first edition, 1996.

[6] M. Dietzfelbinger, T. Hagerup, J. Katajainen, and M. Penttonen. A reliable randomized algorithm for the closest-pair problem. *Journal of Algorithms*, 25:19–51, 1997.

[7] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, and R. E. Tarjan. Dynamic perfect hashing: Upper and lower bounds. *SIAM Journal on Computing*, 23:738–761, 1994.

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

[9] T. Hofmeister and H. Lefmann. A combinatorial design approach to MAXCUT. *Random Structures and Algorithms*, 9:163–173, 1996.

[10] M. Holland and G. Gibson. Parity declustering for continuous operation in redundant disk arrays. In *Proceedings of the 5th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 23–35. 1992.

[11] H. Krawczyk. New hash functions for message authentication. In *Advances in Cryptology – EUROCRYPT '95*, pp. 301–310. 1995.

[12] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107:121–133, 1993.

[13] V. Nguyen and Z. Tuza. Linear-time approximation algorithms for the max cut problem. *Combinatorics, Probability and Computing*, 2:201–210, 1993.

[14] P. Rogaway. Bucket hashing and its application to fast message authentication. *Journal of Cryptology*, 12:91–115, 1999

[15] D. V. Sarwate. A note on universal classes of hash functions. *Information Processing Letters*, 10:41–45, 1980.

[16] E. J. Schwabe and I. M. Sutherland. Flexible usage of redundancy in disk arrays. *Theory of Computing Systems*, 32:561–587, 1999.

[17] D. R. Stinson. Combinatorial techniques for universal hashing. *Journal of Computer and System Sciences*, 48:337–346, 1994.

[18] D. R. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4:369–380, 1994.

[19] D. R. Stinson. On the connections between universal hashing, combinatorial designs and error-correcting codes. *Congressus Numerantium*, 114:7–27, 1996.