DIMACS Technical Report 2000-28
October 2000

Explicit Interpolation Sets Using Perfect Hash Families [1]

by

Xiaodong Sun
Math Department, Rutgers University,
110 Frelinghuysen Road,
Piscataway, NJ 08854-8019

[1] This work is supported in part by NSF grant 9700239, NSF CCR 99-06105 (Special Year on Computational Intractability) and NSF STC 91-19999.

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, NEC Research Institute and Telcordia Technologies (formerly Bellcore).

DIMACS was founded as an NSF Science and Technology Center, and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

Let S be a set of functions with common domain D. We say X, a subset of D, an interpolation set for S if the values on X uniquely determine a function f in S. Recently, Piotr Indyk gave an explicit construction of interpolation sets of size $O(k^4 \log^4 n)$ for the family of boolean functions on n variables which depend symmetrically on at most k variables. Using perfect hash families, we have a variant of Indyk's method to get an explicit construction of interpolation sets of size $O(k^{10} \log n \log\log n / \log\log\log n)$ for this family of functions.

1 Introduction

Let S_k^n denote the family of boolean functions on n variables which depend symmetrically on at most k variables. We write each member in S_k^n as $f_{S,a} : \{0,1\}^n \longrightarrow \{0,1\}$, where S is the set of not more than k variables on which $f_{S,a}$ depends, $a = (a_0 \dots a_k)$ is a vector such that $f_{S,a}(x) = a_i$ whenever exactly i variables in S are set to 1 in x. An interpolation set for a family of functions is a set of assignments to the variables s.t. the values on this set uniquely determine the function in the family. In this paper we consider the problem of constructing interpolation set for S_k^n. Unless otherwise specified, interpolation set means the interpolation set for S_k^n through out this paper.

Finding interpolation set for certain family of functions is an interesting question in several different areas(See [5]). Here we just list two of them. In combinatorial group testing, this problem has applications in fault detection, design of efficient experiments and diagnosis. A typical question is to determine the k faulty components out of n components by asking questions of the form: is there any faulty component in a subset S of components? If all questions are to be asked in parallel (non-adaptively), then this problem is equivalent to finding interpolation set for a sub-family of S_k^n, the family of functions that are disjunctions of the variables in the faulty set. Another area of application is test sets for VLSI circuits. The interpolation set for the family of functions corresponding to functions of each components is useful for testing components of VLSI circuit in parallel. This is equivalent to our problem when each component depends only on at most k inputs symmetrically.

As for the existing result, Indyk [5] gave a proof of existence of an interpolation set of size $O(k^2 \log k \log n)$ and the work of Dyachkov and Rykov [3] and Chaudhuri and Radhakrishnan [1] implies a lower bound of $\Omega(k^2 \log n / \log k)$ on size of the interpolation set. However, the size of the explicit constructive interpolation set given in [5] is $O(k^4 \log^4 n)$ which is still far away from the truth.

The goal of this paper is to give an explicit construction of interpolation set of size $O(k^{10} \log n \log\log n / \log\log\log n)$ for S_k^n. The n term in size bound is almost optimum while the k term is still a low degree polynomial. This bound is better than the previous result when $poly(k) \ll \log n$. We use an interesting construction of perfect hash families which may be of separate interests in other directions.

In section 2, we give the construction of perfect hash family which will be used in section 3 where we present out main result, an explicit construction of the interpolation set.

2 Construction of perfect hash families

The theory of hash functions is an active area in theoretical computer science. In this section, we briefly review some existing results and give a new construction of a family of perfect hash functions that will be used to construct the interpolation set. Let us first give the following definition.

Definition 1 *A set H of functions $h : [N] \longrightarrow [m]$ is called an (N, m, k)-perfect hash family if for any subset $S \subset [N]$ of size k, there is a function $h \in H$ and $h(x) \neq h(y)$ for all $x, y \in S$, $x \neq y$.*

The theory of hash families is a well studied area. Here we are interested in explicit construction of perfect hash families. More results and proofs on this subject can be found in [2, 4, 6]. Fredman, Komlós and Szemerédi [4] gave the following two constructions of perfect hash families. To make this paper self-contained, we give proofs for the following two theorems.

Theorem 2 *(Fredman, Komlós and Szemerédi [4]). For any $S \subseteq \{0, \ldots, N-1\}$ with $|S| = k$, there exists an $a \in [1, N-1]$ such that the function $h_{N,a} : x \longmapsto (ax \text{ mod } N) \text{ mod } k^2$ operates injectively on S.*

Proof. Let $s_{a,i} = |S_{a,i}|$, where $S_{a,i} = \{x | x \in S, \quad (ax \bmod N) \bmod k^2 = i\}$ for $i \in [0, k^2 - 1]$, i.e. $s_{a,i}$ denotes the number of keys in S hashed onto the ith slot in the hash table by $h_{N,a}$.

Note that $\binom{s_{a,i}}{2}$ gives the number of colliding pairs of keys at the ith location of the hash table for $h_{N,a}$. We have

$$\sum_{a=1}^{N-1} \sum_{i=0}^{k^2} \binom{s_{a,i}}{2} = \sum_{x,y \in S, x<y} |\{a : k^2 \text{ divides } a(y-x) \bmod N\}|$$

It is easy to see that for a single pair of distinct keys, x and y, the number of a's satisfying the condition cannot exceed $2N/k^2$. Summing over all $\binom{k}{2}$ pairs, we get

$$\sum_{a=1}^{N-1} \sum_{i=0}^{k^2} \binom{s_{a,i}}{2} < \frac{Nk(k-1)}{k^2} < N - 1.$$

Therefore, there exists an $a \in [1, N-1]$ s.t. $\sum_{i=0}^{k^2} \binom{s_{a,i}}{2} = 0$ which means $h_{N,a}(x) = (ax \bmod N) \bmod k^2$ is an injection on S. ∎

Theorem 3 *(Fredman, Komlós and Szemerédi [4]). Given a set $S \subseteq [N]$, $|S| = k$, there exists a prime $p < k^2 \log N$ such that the function $\zeta_p : x \longmapsto x \text{ mod } p$ is injective on S, i.e. for any $x, y \in S$, $x \neq y$, $x \text{ mod } p \neq y \text{ mod } p$.*

Proof. Let $S = \{x_1, x_2, \ldots, x_k\}$, and let $t = \prod_{i<j}(x_i - x_j)$. Then $\log |t| \leq \binom{k}{2} \log N$. The prime number theorem gives $\log(\prod_{p<x, p \text{ prime}} p) = x + o(x)$, so we conclude that there is some prime $p < k^2 \log N$ that cannot divide t and this is the prime p we are looking for. ∎

Now let us compare the efficiency of the above two constructions. In the first construction in Theorem 2, the size of the range is k^2 but the size of the hash family is $N-1$ which is quite large. In the second construction, the size the hash family is $\pi(k^2\log N) \approx k^2\log N/\log(k^2\log N)$ but the size of the hash table is $k^2\log N$ which is large because of the term $\log N$.

Now the idea is to combine the above two constructions to get a construction so that both hash table size and hash family size are small.

Theorem 4 *Given a set $S \subseteq [N]$, $|S| = k$, there exist two primes p and q, $p < k^2\log N$ and $q < k^2\log(k^2\log N)$, and $a \in [1, q-1]$, such that $h_{q,a} \circ \zeta_q \circ \zeta_p$ is perfect for S.*

Proof. First by applying Theorem 3, we have a prime number $p < k^2\log N$ s.t. ζ_p is perfect for S. Then we take the image of ζ_p, $[0, p-1]$, as the new universe and apply Theorem 3 again. This time we get a prime number $q < k^2\log p \le k^2\log(k^2\log N)$, s.t. ζ_q is perfect for $\zeta_p(S)$. Finally, we take $[0, q-1]$ as the new universe and apply Theorem 2. We have an $a \in [1, q-1]$, s.t. $h_{q,a}$ is perfect for $\zeta_q(\zeta_p(S))$. Therefore, the function $h_{q,a} \circ \zeta_q \circ \zeta_p$ is perfect for S. ■

In the new construction, the hash table size is k^2, the perfect hash family is

$$H = \{h_{q,a} \circ \zeta_q \circ \zeta_p \,|\, p < k^2\log N,\ q < k^2\log(k^2\log N),\ a \in [1, q-1]\}$$

and we have

$$\begin{aligned}
|H| &\le \pi(k^2\log N)\cdot\pi(k^2\log(k^2\log N))\cdot k^2\log(k^2\log N)\\
&\approx \frac{k^2\log N}{\log(k^2\log N)}\cdot\frac{k^2\log(k^2\log N)}{\log(k^2\log(k^2\log N))}\cdot k^2\log(k^2\log N)\\
&= \frac{k^6\log N\log(k^2\log N)}{\log(k^2\log(k^2\log N))}\\
&= \frac{k^6\log N(\log k^2 + \log\log N)}{\log k^2 + \log\log(k^2\log N)}\\
&\le \frac{k^6\log N\log\log N}{\log\log\log N}.
\end{aligned}$$

Therefore, $|H| = O(k^6\log N\log\log N/\log\log\log N)$.

Remark 5 *We can slightly improve the dependence on N at the cost of increasing the exponent of k by apply the above method with more iterations.*

3 Construction of interpolation set

In this section, we give the construction of the interpolation set using the hash family constructed in the previous section. Let us first give the definition of interpolation set.

Definition 6 *Let $\mathcal{F}$ be a family of functions $f : \mathcal{D} \to \mathcal{R}$. A subset X of $\mathcal{D}$ is called interpolation set for $\mathcal{F}$ if for any two distinct function $f, g \in \mathcal{F}$, there is a point $x \in X$ s.t. $f(x) \neq g(x)$, i.e. the values of the function $f \in \mathcal{F}$ on X uniquely determine f.*

Indyk [5] studied the interpolation set for S_k^n, the family of boolean functions on n variables which depend symmetrically on a subset of no more than k variables. Using probabilistic method, he showed the existence of an interpolation set for S_k^n of size $O(k^2 \log k \log n)$ which is within a factor of $\log^2 k$ from the lower bound $O(k^2 \log n/\log k)$. In [5], Indyk also gave an explicit construction of an interpolation set of size $O(k^4 \log^4 n)$ using universal hash families. Here, we will use perfect hash families with a simpler argument to obtain a new construction of interpolation set of size $O(k^{10} \log n \log\log n/\log\log\log n)$ which reduce the power of $\log n$ from 4 to $1+\varepsilon$. In the case $poly(k) \ll \log n$, the size of this construction is better than the result in [5].

The intuition of using hash functions is that the function in S_k^n only depends on a small number of variables. For $f_{S,a} \in S_k^n$, we may take the hash function h that is injective on S and this hash function h will reduce the universe to a significantly smaller universe where we pick elements of the interpolation set. Now let us give the construction of the interpolation set formally. Given an $(n, m, 2k+1)$-perfect hash family H, the interpolation set I_H is defined as follows.

$$I_H = \{h^{-1}([1,j] \oplus \{l\} | h \in H, \quad j, l \in [m]\}$$

where the assignments to n boolean variables are viewed as index vectors of subsets of $[n]$, and

$$[1,j] = \{i | 1 \leq i \leq j, i \in [m]\}$$
$$[1,j] \oplus \{l\} = \begin{cases} [1,j] \cup \{l\}, & \text{if } l \notin [1,j], \\ [1,j] \backslash \{l\}, & \text{if } l \in [1,j]. \end{cases}$$

The size of the set I_H is less than $m^2|H|$. The following theorem shows that I_H is an interpolation set for S_k^n.

Theorem 7 *Given an $(n, m, 2k+1)$-perfect hash family, there exists an interpolation set for S_k^n of size $O(m^2|H|)$.*

Proof. Recall every function in S_k^n is equivalent to some $f_{S,a}$, where S is a subset and a is a vector. For $f \in S_k^n$, let S_f and a_f be the associated set and vector.

It is enough to show that I_H is an interpolation set for S_k^n, i.e. for any two distinct functions $f, g \in S_k^n$, there exists $x \in I_H$ s.t. $f(x) \neq g(x)$. Without loss of generality, we may assume that $f(x)$ is not constant.

Let S_f, S_g be the sets and a_f, a_g the vectors associated with f and g respectively. Since $|S_f \cup S_g| \leq 2k$, there is an $h \in H$ s.t. $h|_{S_f \cup S_g}$ is an injection.

Now we consider two cases:

Case 1: $S_f = S_g$

Since $f \neq g$, there exists an index i s.t. $a_f(i) \neq a_g(i)$. Since $h|_S$ is an injection, there exists $j, l \in [m]$ s.t. $|h^{-1}([1,j] \oplus \{l\}) \cap S_f| = i$. Therefore, at the point $x = h^{-1}([1,j] \oplus \{l\}) \in I_H$, $f(x) \neq g(x)$.

Case 2: $S_f \neq S_g$

Since f is not constant, there exists an index i s.t. $a_f(i-1) \neq a_f(i)$. Since $h|_S$ is an injection, there exists $j, j' \in [m]$, $j < j'$ s.t. $|h^{-1}([1,j]) \cap S_f| = i-1$ and $|h^{-1}([1,j']) \cap S_f| = i$. Since $|S_f \cup S_g| \leq 2k$, there exists a spare index $l \in [m]$ s.t $h^{-1}(l) \notin S$.

Now, if $S_f \backslash (h^{-1}([1,j]) \cup S_g) \neq \emptyset$, there exists an index l' s.t. $l' \notin [1,j]$, $|h^{-1}(l') \cap S_f| = 1$ and $|h^{-1}(l') \cap S_g| = 0$. Therefore, we have

$$
\begin{aligned}
&|h^{-1}([1,j] \oplus \{l\}) \cap S_f| = i-1, \\
&|h^{-1}([1,j] \oplus \{l'\}) \cap S_f| = i, \\
&|h^{-1}([1,j] \oplus \{l\}) \cap S_g| = |h^{-1}([1,j] \oplus \{l'\}) \cap S_g|.
\end{aligned}
$$

Then we pick take $x_1 = h^{-1}([1,j] \oplus l)$, $x_2 = h^{-1}([1,j] \oplus l')$. we have $g(x_1) = g(x_2)$ and $f(x_1) \neq f(x_2)$. Therefore either $f(x_1) \neq g(x_1)$ or $f(x_2) \neq g(x_2)$.

Otherwise, if we have $S_f \subseteq (h^{-1}([1,j]) \cup S_g)$, there must be an index l' s.t. $l' \in [1,j]$, $|h^{-1}(l') \cap S_f| = 1$ and $|h^{-1}(l') \cap S_g| = 0$. Similarly, we have

$$
\begin{aligned}
&|h^{-1}([1,j'] \oplus \{l'\}) \cap S_f| = i-1, \\
&|h^{-1}([1,j'] \oplus \{l\}) \cap S_f| = i, \\
&|h^{-1}([1,j'] \oplus \{l\}) \cap S_g| = |h^{-1}([1,j] \oplus \{l'\}) \cap S_g|.
\end{aligned}
$$

We pick $x_1 = h^{-1}([1,j'] \oplus l)$, $x_2 = h^{-1}([1,j'] \cup l')$. We have either $f(x_1) \neq g(x_1)$ or $f(x_2) \neq g(x_2)$.

We have just shown that I_H is an interpolation set for S_k^n. ■

With the explicit construction of the perfect hash families in the previous section, we have the following.

Corollary 8 *For the boolean function family S_k^n, there is a constructive interpolation set of size $O(k^{10} \log n \log\log n / \log\log\log n)$.*

Proof. Let H be the $(n, (2k+1)^2, 2k+1)$-perfect hash family constructed in Theorem 4. Therefore $|H| \leq O(k^6 \log n \log\log n / \log\log\log n)$. Using Theorem 7, we have the proof. ■

Acknowledgment

The author would like to thank his advisor Michael Saks for many helpful discussions and comments.

References

[1] S. Chaudhuri and R. Radhakrishnan, "Deterministic restrictions in circuit complexity", STOC'96, 30-36.

[2] Z.J. Czech, G. Havas and B.S. Majewski, "Perfect hashing", *Theoretical Computer Science*, Vol. 182, (1997) 1-143.

[3] A.G. Dyachkov, V.V. Rykov, "A survey of superimposed code theory", *Problem of Control and Information Theory(1983)*, Vol. 12, No. 4(English translation)

[4] M.L. Fredman, J. Komlós and E. Szemerédi, "Storing a sparse table with $O(1)$ worst case access time", *J. ACM* Vol. 31, No. 3 (1984) 538-544.

[5] Piotr Indyk, "Interpolation of symmetric functions and a new type of combinatorial design", *Proc. 31st ACM Symp. on Theory of Computing* (1999) 736-740.

[6] Kurt Mehlhorn, Sorting and searching, Springer-Verlag, 1984.