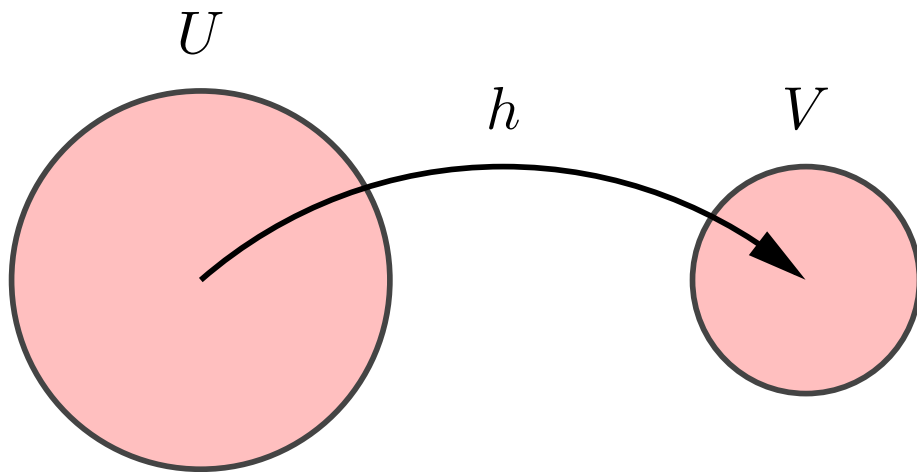

Uniform Hashing
in
Constant Time and Linear Space

Anna Östlin and Rasmus Pagh
IT University of Copenhagen

STOC 2003, San Diego

Presented by Martin Dietzfelbinger
TU Ilmenau

— Uniform hashing



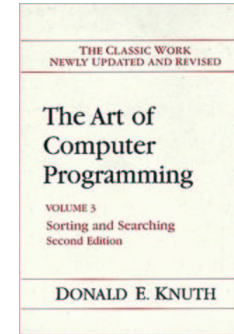
Uniform hashing assumption:

h maps elements of U uniformly at random and independently to V .

Hash functions, i.e., functions “mimicking” a uniform hash function, have applications in information retrieval, complexity theory, data mining, cryptography, etc.

— Usage of uniform hashing —

In analysis of algorithms, it is often assumed that the hash functions used are uniform. For example, all analyses of hashing schemes in *The Art of Computer Programming* use the uniform hashing assumption.



Is this reasonable?

For:

- In practice many simple hash functions perform as well as in the uniform hashing analysis.
- Often one can carry analyses over to explicit hash function classes with restricted randomness.

Against:

- True uniform hashing requires $|U| \log |V|$ bits of space. Mostly infeasible!
- Analyses for restricted randomness hash functions can be cumbersome (or undoable).

— The new result —

It is possible to get **very close** to the theoretical ideal of uniform hashing:

We construct a hash function that:

- Is uniform, *with high probability*, on any *particular* set S of size n .
- Can be stored in $O(n)$ space (which is optimal).
- Can be evaluated in constant time.

k -wise independence

One approach to “mimicking” a truly random function is to choose a hash function that is uniform on any set of size at most k , for some $k < |U|$.

This property is called *k -wise independence*.

Example:

For random $a_0, \dots, a_{k-1} \in \{0, \dots, p-1\}$, the function

$$h(x) = \left(\sum_{i=0}^{k-1} a_i x^i \bmod p \right) \bmod |V|$$

where p is prime, is k -wise independent.

— Usage of bounded independence —

Examples of analyses using bounded independence:

Independence	Algorithms	Type of analysis
2-wise	chained hashing dynamic perfect hashing	expected performance
4-wise	chained hashing dynamic perfect hashing	high probability bounds
$O(\log n)$ -wise	open addressing PRAM simulation	high probability bounds
n -wise	most hashing schemes	uniform hashing assumption

— Known n -wise independent hash functions —

Assume that $|U| = n^c$ for a constant c (see paper for general case).

Reference	Space	Eval. time	Error prob.
Polynomial	$O(n)$	$O(n)$	0
[Siegel 1989]	$n^{\sqrt{c}+\epsilon}$	$O(1)$	$n^{-O(1)}$
[Siegel 1989] (nonconstructive)	$n^{1+\epsilon}$	$O(1)$	0 (in general $n^{-O(1)}$)
New result	$O(n)$	$O(1)$	$n^{-O(1)}$

— The new result in detail —

RAM model: Unit cost with word size $\Theta(\log |U| + \log |V|)$.

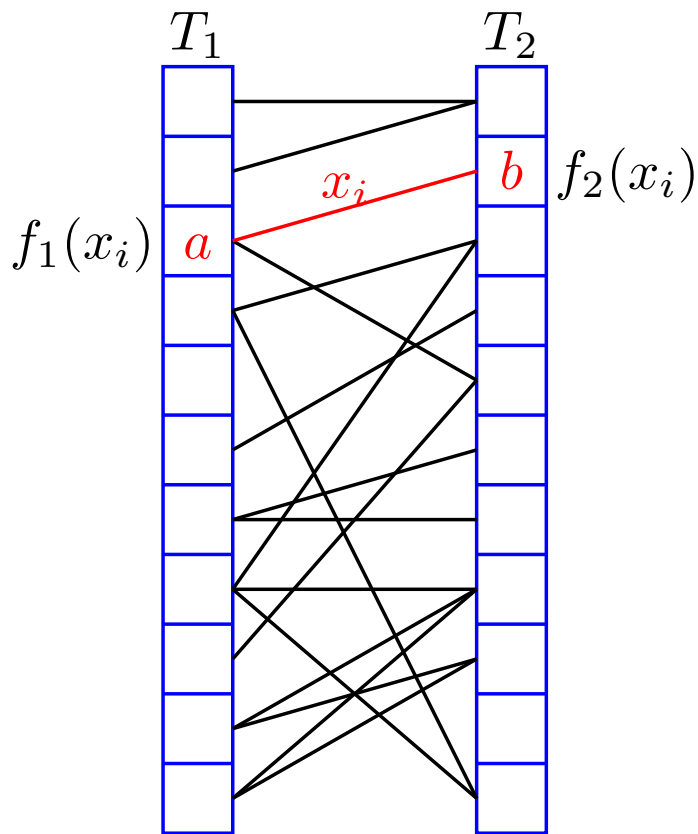
We can construct a random family of functions from U to V such that for any set $S \subseteq U$ of n elements:

- With high probability the family is uniform on S .
- There is a data structure of $O(n)$ words representing the family such that function values can be computed in constant time.
- The data structure can be set to a random function in $O(n)$ time.

The construction uses $o(n)$ words of space and takes expected time $o(n) + (\log \log |U|)^{O(1)}$.

The hash function family

$$h(x_i) = (a + b + g(x_i)) \bmod |V|, \text{ where } a = T_1[f_1(x_i)] \text{ and } b = T_2[f_2(x_i)].$$



Details:

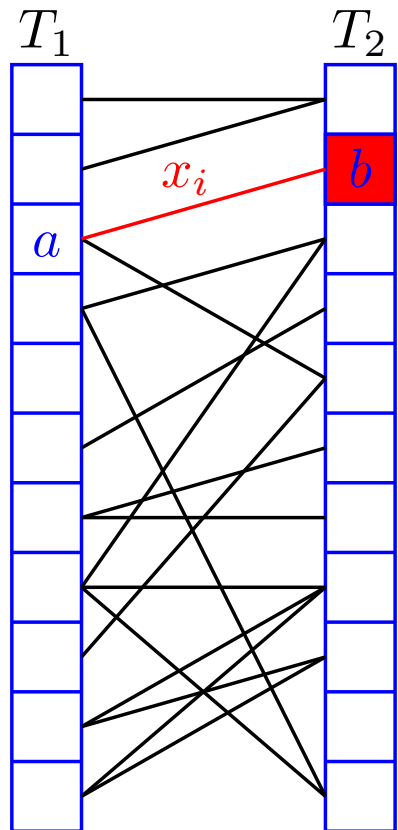
- g is a $O(\log n)$ -wise independent function from U to $\{0, \dots, |V| - 1\}$.
- f_1 and f_2 are $O(\log n)$ -wise independent functions from U to $\{0, \dots, 4n\}$
- g , f_1 and f_2 can be implemented in space $o(n)$ and with constant evaluation time using Siegel's construction.
- Entries in T_1 and T_2 are uniformly random in $\{0, \dots, |V| - 1\}$.

— Analysis (sketch) —

Let $S = \{x_1, \dots, x_n\}$.

Consider the bipartite graph with edges $(f_1(x_i), f_2(x_i))$, $i = 1, \dots, n$.

Observation:

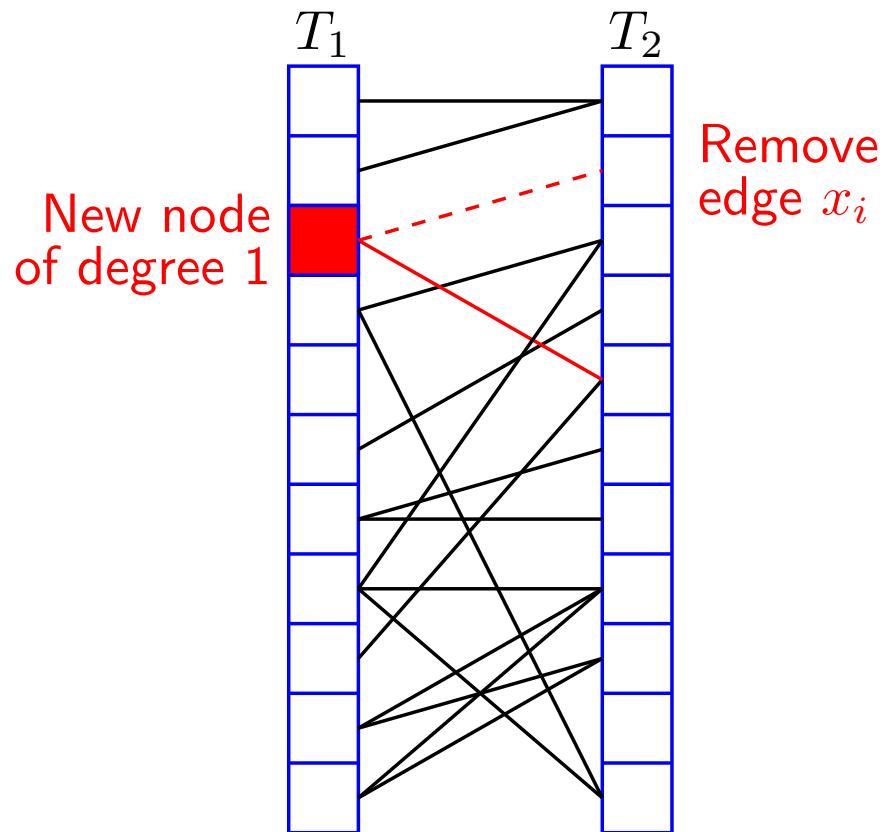


Node of degree 1

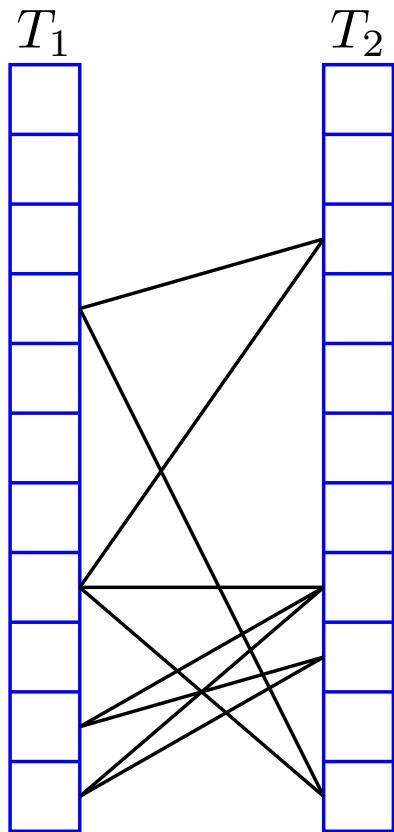
$h(x_i) = (a + b + g(x_i)) \bmod |V|$
is independent of all other
function values since b is
a random number

— Analysis (sketch) —

h is uniform on S if and only if it is uniform on $S \setminus \{x_i\}$.



Analysis (sketch)



- Repeatedly remove edges with degree 1.
- What remains is the cyclic part.
- h is uniform on S if g is k -wise independent and the cyclic part has size at most k .
- It can be shown that the cyclic part has size $O(\log n)$ w.h.p.
- Recall that we chose g to be $O(\log n)$ -wise independent w.h.p.

Conclusion:

The hash function is uniform on S with high probability.

— Implications

For many hashing schemes, the new hash function is the first to make their uniform hashing analysis come true, with high probability, without incurring overhead in time or space.

— Concluding remarks —

Following this work, Dietzfelbinger and Woelfel (STOC '03) have devised a simple uniform hashing scheme with similar properties that does *not* use Siegel's (impractical) construction.

Open problems:

- Can the error probability be reduced?
(Siegel has shown that it cannot be zero.)
- Devise explicit expanders for Siegel's construction.
(This could perhaps make it practical.)