Space Efficient
Hash Tables with
Worst Case Constant Access Time

Dimitris Fotakis and **Peter Sanders** (MPII)

Rasmus Pagh (IT U. Copenhagen)

Paul Spirakis (CTI)

Overview

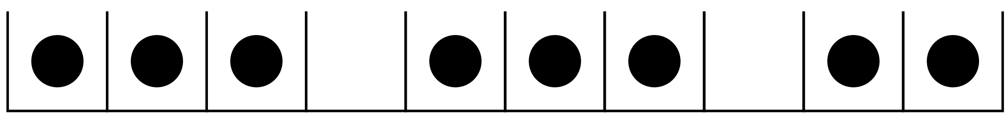
- The Problem and Related Work
- Cuckoo Hashing
- d-ary Cuckoo Hashing
- Analysis
- Relation to Bipartite Matching
- Filter Hashing
- Discussion

The Problem

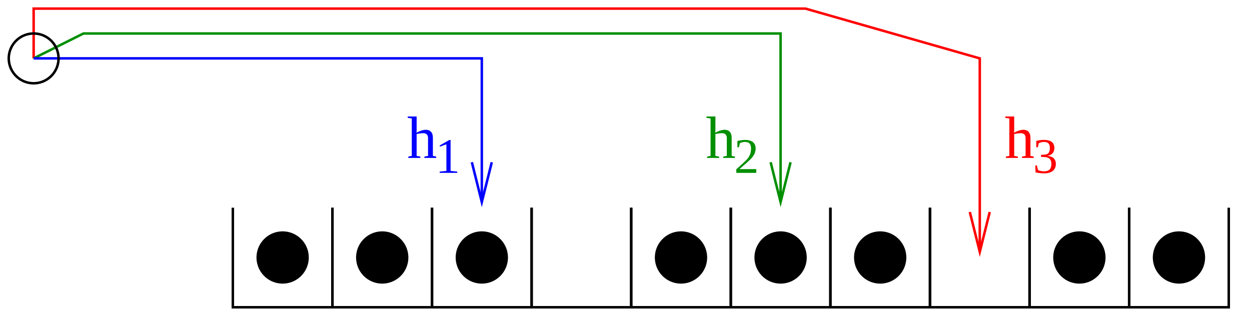
Represent a set of n elements (with associated information) using space $(1+\epsilon)n$.

Support operations insert, delete, lookup, (doall) efficiently.

Assume a truly random hash function h

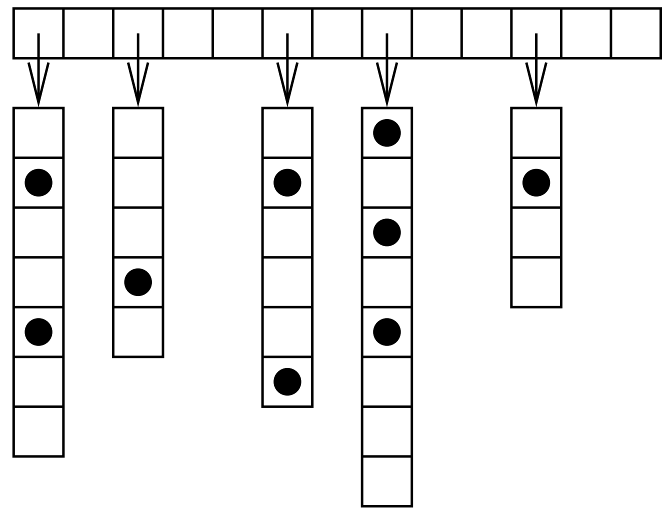


Related Work



Uniform hashing:

Expected time $\approx \frac{1}{\epsilon}$



Dynamic Perfect Hashing,

[Dietzfelbinger et al. 94]

Worst case constant time

for lookup but ϵ is not small.

Approaching the Information Theoretic Lower Bound:

[Brodnik Munro 99,Raman Rao 02]

Space $(1 + o(1))\times$lower bound without associated information

[Pagh 01] static case.

INFORMATIK

Cuckoo Hashing

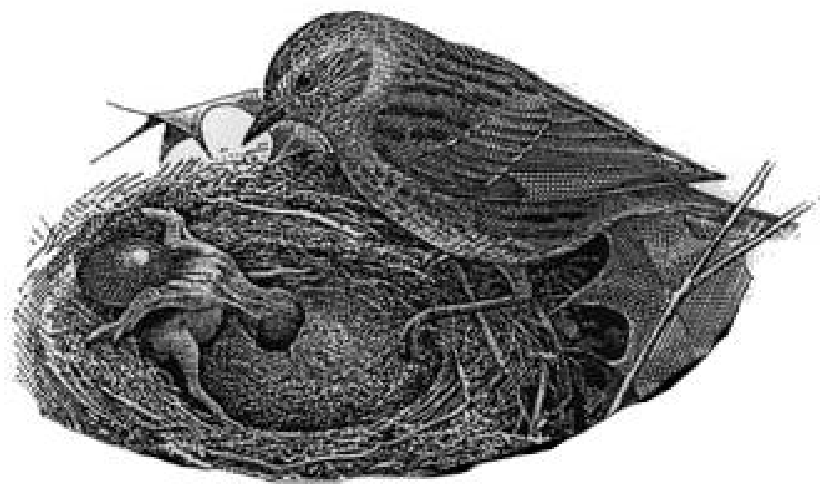
[Pagh Rodler 01]

Table of size $(2 + \epsilon)n$.

Two choices for each element.

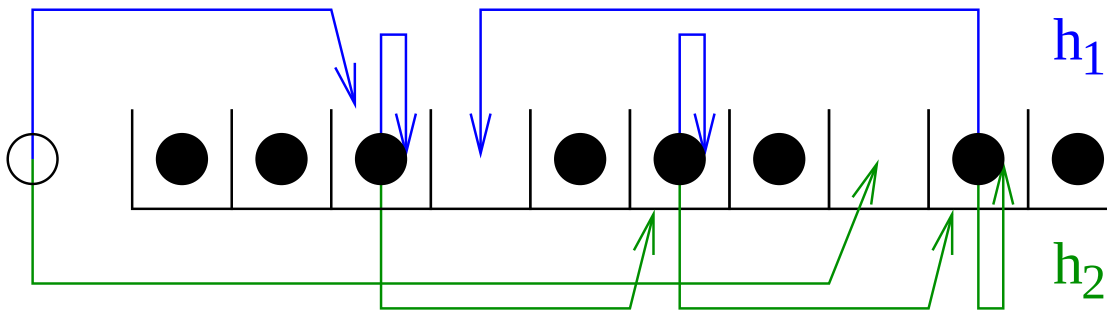
Insert moves elements;

rebuild if necessary.



Very fast lookup and delete.

Expected constant insertion time.



d-ary Cuckoo Hashing

d choices for each element.

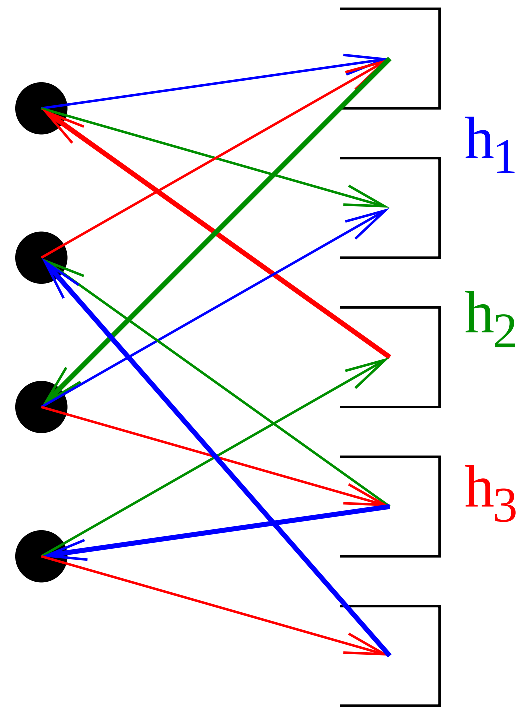
Worst case d probes for delete and lookup.

Task: maintain L-perfect matching

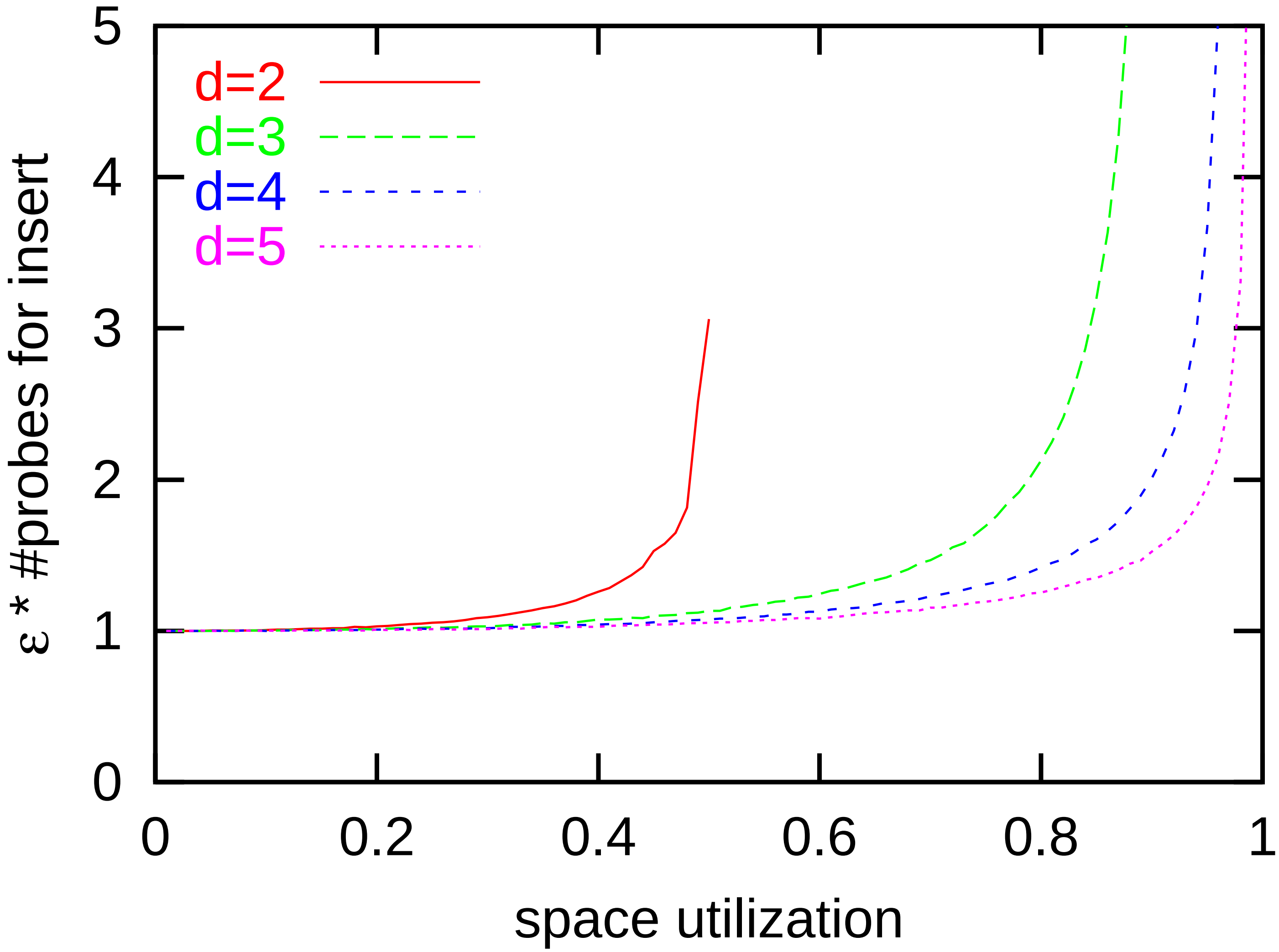
in the bipartite graph

$(L = \text{Elements}, R = \text{Cells}, E = \text{Choices})$,

e.g., insert by BFS.



Experiments



Tradeoff: Space $\leftrightarrow$ Lookup/Deletion Time

Lookup and Delete: $d = \mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ probes

Proof Outline:

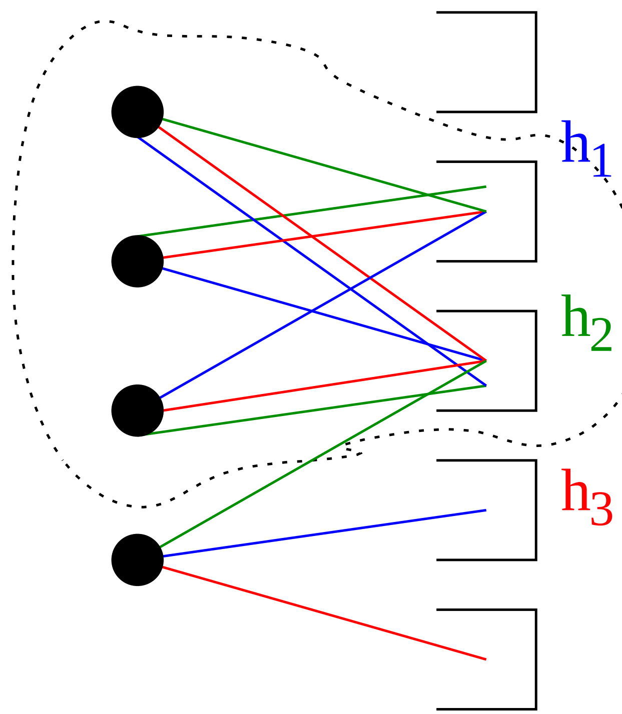
the bipartite graph (L, R, E)

has an L-perfect matching

$\Leftrightarrow$ Hall's Theorem

$\nexists M \subseteq L : |\text{neighbors}(M)| < |M|$

... Chernoff bounds ...



true whp if $d \geq 2(1+\epsilon)\ln(\frac{e}{\epsilon})$

Tradeoff:

Space $\leftrightarrow$ Insertion time

Insert: $\left(\frac{1}{\epsilon}\right)^{\mathcal{O}(\log\log(1/\epsilon))}$, (experiments) $\longrightarrow \mathcal{O}(1/\epsilon)$?

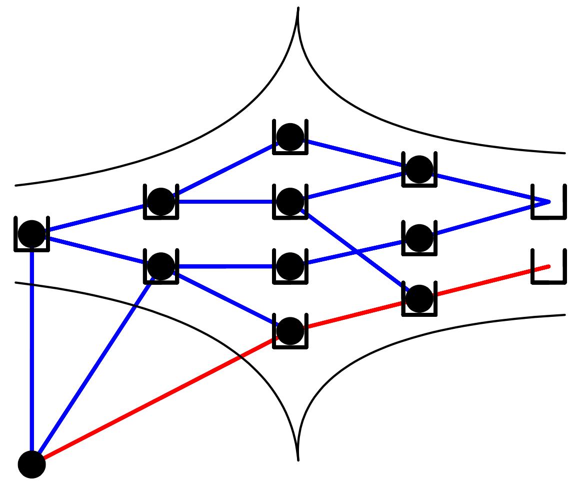
Expansion property: half the nodes within

$\mathcal{O}(\log(1/\epsilon))$ from a free node

Shrinking property: number of far-away nodes

shrinks geometrically with distance

$\Rightarrow$ short average augmenting path length



Average Case Analysis of Bipartite Matching

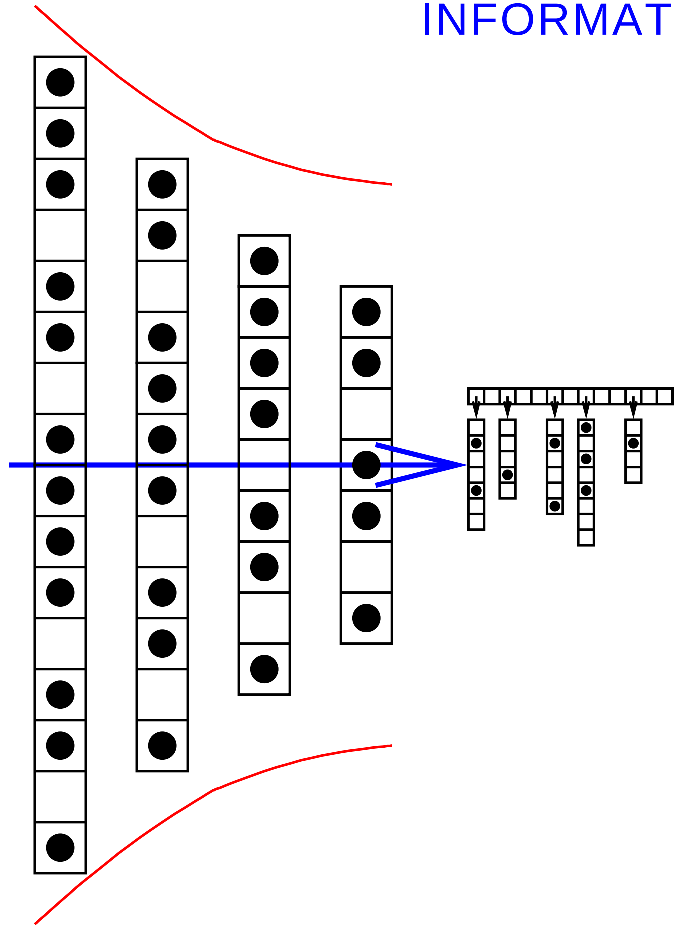
[Motwani 94]: A bipartite graph (L, R, E) with $|L| = |R|$ and $|E| > n \ln n$ random edges has a perfect matching whp.

Time $\mathcal{O}(|E| \log |L| / \log \log |L|)$

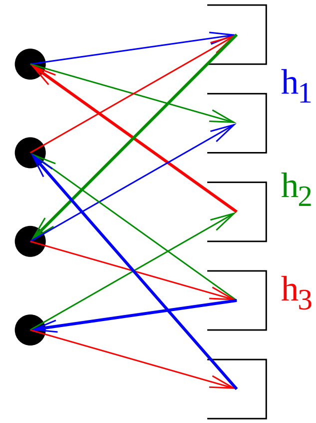
Here: slight assymmetry, very sparse, linear time

Filter Hashing

- $\mathcal{O}\left(\log^2 \frac{1}{\epsilon}\right)$ layers
- shrinking geometrically
- perfect hashing for the overflow table
- realistic hash functions



Discussion



d-ary Cuckoo: fast, practical, very space efficient

Open Question

- "real" hash functions
- Tighten insertion time
- average case lookup time
- average case max cardinality bipartite matching for sparse symmetric graphs