The power and weakness of randomness 
(when you are short on time)

Avi Wigderson
Institute for Advanced Study
Plan of the talk

• Computational complexity
  -- efficient algorithms, hard and easy problems, P vs. NP
• The power of randomness
  -- in saving time
• The weakness of randomness
  -- what is randomness?
  -- the hardness vs. randomness paradigm
• The power of randomness
  -- in saving space
  -- to strengthen proofs
**Easy and Hard Problems**

*asymptotic complexity of functions*

<table>
<thead>
<tr>
<th>Multiplication</th>
<th>Factoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>mult(23,67) = 1541</td>
<td>factor(1541) = (23,67)</td>
</tr>
</tbody>
</table>

grade school algorithm:  
\[ n^2 \text{ steps on } n \text{ digit inputs} \]

best known algorithm:  
\[ \exp(\sqrt[4]{n}) \text{ steps on } n \text{ digits} \]

**EASY**  
P - Polynomial time algorithm

**HARD?**  
--- we don’t know!  
--- the whole world thinks so!
Map Coloring and P vs. NP

**Input:** planar map M (with n countries)

- **2-COL:** is M 2-colorable? Easy
- **3-COL:** is M 3-colorable? Hard?
- **4-COL:** is M 4-colorable? Trivial

**Thm:** If 3-COL is Easy then Factoring is Easy

- **Thm [Cook-Levin ’71, Karp ’72]:** 3-COL is NP-complete
  - Numerous equally hard problems in all sciences

**P vs. NP problem:**

**Formal:** Is 3-COL Easy?

**Informal:** Can creativity be automated?
Fundamental question #1

Is $\text{NP} \neq \text{P}$? Is any of these problems hard?
- Factoring integers
- Map coloring
- Satisfiability of Boolean formulae
- Traveling salesman problem
- Solving polynomial equations
- Computing optimal Chess/Go strategies

Best known algorithms: exponential time/size. Is exponential time/size necessary for some?

Conjecture 1: YES
The Power of Randomness

Host of problems for which:

- We have \textit{probabilistic} polynomial time algorithms

- We (still) have \textit{no deterministic} algorithms of subexponential time.
Coin Flips and Errors

Algorithms will make decisions using coin flips
0111011000010001110101010111…
(flips are independent and unbiased)

When using coin flips, we'll guarantee:
“task will be achieved, with probability >99%”

Why tolerate errors?
- We tolerate uncertainty in life
- Here we can reduce error arbitrarily $<\exp(-n)$
- To compensate – we can do much more…
Number Theory: Primes

Problem 1: Given $x \in [2^n, 2^{n+1}]$, is $x$ prime?

1975 [Solovay-Strassen, Rabin]: Probabilistic

2002 [Agrawal-Kayal-Saxena]: Deterministic !!

Problem 2: Given $n$, find a prime in $[2^n, 2^{n+1}]$

Algorithm: Pick at random $x_1, x_2, \ldots, x_{1000n}$
For each $x_i$ apply primality test.
_prime Number Theorem $\Rightarrow \Pr \left[ \exists i \ x_i \text{ prime} \right] > .99$
Algebra: Polynomial Identities

Is $\det( ) - \prod_{i<k} (x_i-x_k) \equiv 0$?

Theorem [Vandermonde]: YES

Given (implicitly, e.g. as a formula) a polynomial $p$ of degree $d$. Is $p(x_1, x_2, ..., x_n) \equiv 0$?

Algorithm [Schwartz-Zippel '80]:
Pick $r_i$ indep at random in $\{1, 2, ..., 100d\}$

$p \equiv 0 \Rightarrow \Pr[ p(r_1, r_2, ..., r_n) = 0 ] = 1$

$p \neq 0 \Rightarrow \Pr[ p(r_1, r_2, ..., r_n) \neq 0 ] > .99$

Applications: Program testing, Polynomial factorization
Analysis: Fourier coefficients

Given (implicitly) a function $f: (\mathbb{Z}_2)^n \rightarrow \{-1,1\}$ (e.g. as a formula), and $\varepsilon > 0$,
Find all characters $\chi$ such that $|\langle f, \chi \rangle| \geq \varepsilon$

Comment: At most $1/\varepsilon^2$ such $\chi$

Algorithm [Goldreich-Levin '89] :
...adaptive sampling... $\Pr[\text{success}] > .99$

[AGS]: Extension to other Abelian groups.
Applications: Coding Theory, Complexity Theory
Learning Theory, Game Theory
Geometry: Estimating Volumes

Given (implicitly) a convex body $K$ in $\mathbb{R}^d$ (d large!)
(e.g. by a set of linear inequalities)
Estimate volume ($K$)
Comment: Computing volume($K$) exactly is $\#P$-complete

Algorithm [Dyer-Frieze-Kannan '91]:
Approx counting $\approx$ random sampling
Random walk inside $K$.
Rapidly mixing Markov chain.

Analysis:
Spectral gap $\approx$ isoperimetric inequality

Applications:
Statistical Mechanics, Group Theory
Fundamental question #2
Does randomness help?
Are there problems with probabilistic polytime algorithm but no deterministic one?
Conjecture 2: YES

Fundamental question #1
Does \textit{NP} require exponential time/size?
Conjecture 1: YES

Theorem: One of these conjectures is false!
Hardness vs. Randomness

Theorems \([\text{Blum-Micali, Yao, Nisan-Wigderson, Impagliazzo-Wigderson…}]\):

If there are natural hard problems, then randomness can be efficiently eliminated.

Theorem \([\text{Impagliazzo-Wigderson ’98}]\)

NP requires exponential size circuits \(\Rightarrow\)
every probabilistic polynomial-time algorithm has a deterministic counterpart

Theorem \([\text{Impagliazzo-Kabanets’04, IKW’03}]\)
Partial converse!
Computational Pseudo-Randomness

Goldwasser-Micali '81 pseudorandom if for every efficient algorithm, for every input, output \( \approx \) output

\( k \sim c \log n \)
Hardness $\Rightarrow$ Pseudorandomness

Need $G$: $k$ bits $\rightarrow$ $n$ bits

NW generator

Show $G$: $k$ bits $\rightarrow$ $k+1$ bits

Need: $f$ hard on random input  Average-case hardness

Hardness amplification

Have: $f$ hard on some input  Worst-case hardness
Derandomization

Deterministic algorithm:
- Try all possible $2^k = n^c$ “seeds”
- Take majority vote

Pseudorandomness paradigm:
Can derandomize specific algorithms without assumptions!

*e.g.* Primality Testing & Maze exploration
Randomness and space complexity
Getting out of mazes
(when your memory is weak)

Theseus

Ariadne

Crete, ~1000 BC

Theorem [Aleliunas-Karp-Lipton-Lovasz-Rackoff '80]:
A random walk will visit every vertex in $n^2$ steps (with probability >99%).

Only a local view (logspace) $n$-vertex maze/graph

Theorem [Reingold '06]:
A deterministic walk, computable in logspace, will visit every vertex. Uses ZigZag expanders [Reingold-Vadhan-Wigderson '02]

Mars, 2003 AD

Comet
The power of pandomness in Proof Systems
Probabilistic Proof System
[Goldwasser–Micali–Rackoff, Babai '85]
Is a mathematical statement claim true? E.g.
claim: “No integers \(x, y, z, n>2\) satisfy \(x^n + y^n = z^n\”
claim: “The Riemann Hypothesis has a 200 page proof”

probabilistic
An efficient Verifier \(V(claim, argument)\) satisfies:

*) If claim is true then \(V(claim, argument) = \text{TRUE}\) for some argument always
(in which case claim=theorem, argument=proof)

**) If claim is false then \(V(claim, argument) = \text{FALSE}\) for every argument with probability \(> 99\%\)
Remarkable properties of Probabilistic Proof Systems

- Probabilistically Checkable Proofs (PCPs)
- Zero-Knowledge (ZK) proofs
Probabilistically Checkable Proofs (PCPs)

**claim**: The Riemann Hypothesis

**Prover**: (argument)

**Verifier**: (editor/referee/amateur)

Verifier's concern: Has no time...

**PCPs**: Ver reads 100 (random) bits of argument.

Th[Arora-Lund-Motwani-Safra-Sudan-Szegedy'90]

Every proof can be eff. transformed to a PCP

Refereeing (even by amateurs) in seconds!

**Major application** - approximation algorithms
Zero-Knowledge (ZK) proofs

[Goldwasser-Micali-Rackoff '85]

claim: The Riemann Hypothesis
Prover: (argument)
Verifier: (editor/referee/amateur)

Prover’s concern: Will Verifier publish first?
ZK proofs: argument reveals only correctness!

Theorem [Goldreich-Micali-Wigderson '86]:
Every proof can be efficiently transformed to a ZK proof, assuming Factoring is HARD
Major application - cryptography
Conclusions & Problems

When resources are limited, basic notions get new meanings (randomness, learning, knowledge, proof, ...).

- Randomness is in the eye of the beholder.
- Hardness can generate (good enough) randomness.
- Probabilistic algs seem powerful but probably are not.
- Sometimes this can be proven! (Mazes, Primality)
- Randomness is essential in some settings.

Is Factoring HARD? Is electronic commerce secure?
Is Theorem Proving Hard? Is P≠NP? Can creativity be automated?