Do Probabilistic Algorithms Outperform Deterministic Ones?

Avi Wigderson
Computer Science Institute
The Hebrew University
Jerusalem, Israel

Summary

The introduction of randomization into efficient computation has been one of the most fertile and useful ideas in computer science. In cryptography and asynchronous computing, randomization makes possible tasks that are impossible to perform deterministically. For function computation, many examples are known in which randomization allows considerable savings in resources like space and time over deterministic algorithms, or even "only" simplifies them.

But to what extent is this seeming power of randomness over determinism real? The most famous concrete version of this question regards the power of BPP, the class problems solvable by probabilistic polynomial time algorithms making small constant error. We know nothing beyond the trivial relation $P \subseteq BPP \subseteq EXP$, so both $P = BPP$ (read "randomness is useless") or $BPP = EXP$ (read "randomness is all-powerful") are currently equally possible. A major problem is shrinking this gap in our knowledge, or at the very least eliminating the (proposterous) second possibility.

A fundamental discovery (that emerged in the early 80's in the sequence of seminal papers [18, 4, 19]) regarding this problem is the "hardness versus randomness" paradigm. It relates this major problem to another equally important one: are there natural hard functions? Roughly speaking, "computationally hard" functions can be used to construct "efficient pseudo-random generators". These in turn lower the randomness requirements of any efficient probabilistic algorithm, allowing for a nontrivial deterministic simulation. Thus, under various complexity assumptions, randomness is weak or even "useless", and the challenge becomes to use the weakest possible assumption, at the hope of finally removing it altogether.

Only two methods are known for converting hard functions into pseudo-random sequences: the BMY-generator (introduced by Blum, Micali and Yao) and the NW-generator (introduced by Nisan and Wigderson). The BMY-generator [4, 19, 8, 9], in which the hardness versus randomness paradigm first appeared, uses one-way functions. Its construction facilitates using either nonuniform or uniform hardness assumptions. The results are (informally) summarized below, for nonuniform assumptions. We use $SIZE(s(n))$ to denote all functions computable with a family of Boolean circuits of size $s(n)$, and $P/poly = SIZE(n^{O(1)})$. Also, $SUBEXP = \cap_{\delta>0} DTIME(2^{n^\delta})$, and $\tilde{P} = DTIME(exp(\log n)^{O(1)})$, namely quasi-polynomial time.

Theorem 1 *[4, 19, 8, 9]*

If there are one-way functions not in $P/poly$, then $BPP \subset SUBEXP$.

If there are one-way functions not in $SIZE(exp(n^{o(1)}))$, then $BPP \subset \tilde{P}$.

The NW-generator [16, 17, 3] considerably weakened the hardness assumption needed in the nonuniform setting. It achieves the same deterministic simulation of BPP, from any function in EXP. The (wide ?) belief that $EXP \not\subseteq P/poly$ makes easy the belief in its corolarry $BPP \neq EXP$.

Theorem 2 *[16, 17, 3]*

If $EXP \neq P/poly$, then $BPP \subset SUBEXP$

If $EXP \neq SIZE(exp(n^{o(1)}))$, then $BPP \subset \tilde{P}$

While this already supplies considerable evidence to the weakness of probabilistic polynomial time algorithms, it leaves much room for progress. Recently significant steps were made in two different directions - tighter trade-offs and uniform assumptions.

The first deals with finding natural assumptions under which randomness is really "useless", namely $BPP = P$. The major obstacle was the fact that various "hardness amplification" techniques, most notably the XOR-lemma, which are key in the hardness to pseudorandomness conversion, significantly increased the input size. The combination of two papers, [10, 11] give much more efficient versions of the XOR lemma, and yield the following. Here $E = DTIME(exp(O(n))$.

Theorem 3 *[10, 11] If* $E \not\subseteq SIZE(exp(o(n))$ *then* $BPP = P$.

The same consequence was obtained, under the considerably stronger assumption $E \not\subseteq SIZE(o(2^n/n))$, but via totally different and interesting techniques, in [1, 2].

The second direction deals with the possible use of a uniform of Theorem 2, i.e. requiring the hard function be hard for probabilistic Turing machines rather than Boolean circuits. While this presents no major problems if the function is one-way and we are using the BMY-generator (as was pointed out in the original papers), for 10 years since the introduction of the NW-generator no way was found to "uniformize" that conversion of hardness to randomness. Very recently, this was achieved in [12], and is stated (informally) below. Note that the simulation of BPP here is only in $Av - SUBEXP$, namely requires deterministic subexponential time on average whenever the inputs are drawn from an efficiently samplable distribution [13].

Theorem 4 *[12] If* $BPP \neq EXP$, *then* $BPP \subseteq Av - SUBEXP$

This result is naturally interpreted as a gap theorem on derandomization - either no derandomization of BPP is possible at all (BPP is "all-powerful"), or otherwise a highly nontrivial dernadomization is possible.

We believe that the basic question of the power of BPP deserves more attention, and that an unconditional result is possible. The challenge is to prove the weakest such statement:

Conjecture 5 $EXP \neq BPP$

To conclude, we refer the reader to some general surveys that contain some of this material in an organized fashion - the three surveys of Oded Goldreich [5, 6, 7] and the monograph of Mike Luby [14].

References

[1] A. Andreev, A. Clementi and J. Rolim, "Hitting Sets Derandomize BPP", in *XXIII International Colloquium on Algorithms, Logic and Programming (ICALP'96)*, 1996.

[2] A. Andreev, A. Clementi, and J. Rolim, "Hitting Properties of Hard Boolean Operators and its Consequences on BPP", manuscript, 1996.

[3] L. Babai, L. Fortnow, N. Nisan and A. Wigderson, "BPP has Subexponential Time Simulations unless EXPTIME has Publishable Proofs", *Complexity Theory*, Vol 3, pp. 307–318, 1993.

[4] M. Blum and S. Micali. "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits", *SIAM J. Comput.*, Vol. 13, pages 850–864, 1984.

[5] O. Goldreich, *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, to be published by *Springer*.

[6] O. Goldreich, "Randomness, Interaction, Proofs and Zero-Knowledge", *The Universal Turing Machine: A Half-Century Survey*, R. Herken (ed.), Oxford University Press, 1988, London, pp. 377 406. A revised version of the section on pseudorandomness is available from `http://theory.lcs.mit.edu/pub/people/oded/prg88.ps`.

[7] O. Goldreich. "Pseudorandomness", Chapter 3 of *Foundation of Cryptography - Fragments of a Book*, February 1995. Available from `http://theory.lcs.mit.edu/~oded/frag.html`.

[8] O. Goldreich and L.A. Levin. "A Hard-Core Predicate for all One-Way Functions", in *ACM Symp. on Theory of Computing*, pp. 25-32, 1989.

[9] J. Hastad, R. Impagliazzo, L.A. Levin and M. Luby, "Construction of Pseudorandom Generator from any One-Way Function", to appear in *SICOMP*. (See preliminary versions by Impagliazzo et. al. in *21st STOC* and Hastad in *22nd STOC*.)

[10] R. Impagliazzo, "Hard-core Distributions for Somewhat Hard Problems", in *36th FOCS*, pages 538-545, 1995.

[11] R. Impagliazzo and A. Wigderson, "P=BPP unless E has sub-exponential circuits: Derandomizing the XOR Lemma", in *29th STOC*, pp. 220-229, 1997.

[12] R. Impagliazzo and A. Wigderson, "A Gap Theorem for Derandomization", In preparation.

[13] L.A. Levin, "Average Case Complete Problems", *SIAM J. Comput.*, 15:285-286, 1986.

[14] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton Computer Science Notes, Princeton University Press, 1996.

[15] R. Lipton, "New directions in testing", In J. Fegenbaum and M. Merritt, editors, *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Volume 2, pp. 191-202. American Mathematical Society, 1991.

[16] N. Nisan, "Pseudo-random bits for constant depth circuits", *Combinatorica 11* (1), pp. 63-70, 1991.

[17] N. Nisan, and A. Wigderson, "Hardness vs Randomness", *J. Comput. System Sci.* 49, 149-167, 1994

[18] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences", *8th ICALP, Lecture Notoes in Computer Science* 62, Springer-Verlag, pp. 544-550, 1981.

[19] A.C. Yao, "Theory and Application of Trapdoor Functions", in *23st FOCS*, pages 80-91, 1982.