

# Efficiency Considerations in Using Semi-random Sources.

(Extended Abstract)

*Umesh V. Vazirani*

Harvard University

Cambridge, MA 02138.

## 1. Introduction

Randomness is an important computational resource, and has found application in such diverse computational tasks as combinatorial algorithms, synchronization and deadlock resolution protocols, encrypting data and cryptographic protocols. Blum [Bl] pointed out the fundamental fact that whereas all these applications of randomness assume a source of independent, unbiased bits, the available physical sources of randomness (such as zener diodes) suffer seriously from problems of correlation. A general

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

mathematical model for such an imperfect source of randomness is the semi-random source [SV]. In this model, the next output bit is produced by an adversary by the flip of a coin of variable bias. The correlations in the output are modeled by allowing the adversary to look at the previously output bit sequence before setting the bias of the next bit. The only constraint on the adversary is that the coin must be biased between  $\delta$  and  $1 - \delta$ .

In this paper, we give efficient algorithms for using such semi-random sources in place of truly-random sources. Our results are of two different flavours. In the first we assume that we have two independent semi-random sources. It was shown in [Va] that there is an algorithm that produces  $n$ -bit quasi-random sequences ("high quality" random

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

sequences) from  $O(n \log n \log^* n)$  bits output by the two sources. In this paper, we show how to obtain quasi-random sequences at an optimal rate: we obtain  $n$ -bit quasi-random sequences from  $O(n)$  bits output by the two semi-random sources. The rate of production of quasi-random sequences is an important practical consideration.

An important feature of the algorithm in [Va] is its simplicity and ease of implementation: divide the  $O(n \log n \log^* n)$  output bits from the two sources into  $n$  pairs of blocks each of length  $O(\log n \log^* n)$ . Denote by  $\vec{x}_i$  and  $\vec{y}_i$  the  $i^{\text{th}}$  blocks output by the two sources. Then the  $i^{\text{th}}$  output bit is simply  $\vec{x}_i \vec{y}_i \bmod 2$ . The algorithm presented in this paper retains the simplicity of the algorithm in [Va], while achieving optimal rate: let  $p$  be a prime number such that 2 is a primitive root mod  $p$ , with  $p = O(\log n \log^* n)$ . Divide the  $O(n)$  output bits from the two sources into blocks of size  $p$  each. Denote by  $\vec{x}_i$  and  $\vec{y}_i$  the  $i^{\text{th}}$  blocks output by the two sources. Let  $\pi^j(\vec{y}_i)$  denote the cyclic shift of  $\vec{y}_i$  by  $j$  positions. Obtain  $O(p)$  bits  $b_j$  from  $x_i$  and  $y_i$  by letting  $b_j = \vec{x}_i \pi^j(\vec{y}_i)$ . Proving the validity of this algorithm involves showing that the bits,  $b_i$ , thus obtained are independent. Let us think of  $\vec{y}_i$  as extracting information from  $\vec{x}_i$  (by taking the xor of some subset of its bits). To show that the bits  $b_j$  are

independent it suffices to show that  $\pi^j(\vec{y}_i)$  extracts essentially distinct information out of  $\vec{x}_i$  for distinct values of  $j$ . The proof of this hinges upon the following lemma which says that if the length of  $\vec{y}_i$  satisfies the stated conditions, then  $\vec{y}_i$  has essentially no symmetries under cyclic shifts: Let  $p$  be a prime such that 2 is a primitive root mod  $p$ , and let  $A$  be a  $p \times p$  matrix whose rows are cyclic shifts of a 0/1 vector  $\vec{u}$  ( $\vec{u}$  is not the all 0's vector or the all 1's vector). Then the rank of  $A$  over  $\text{GF}(2)$  is at least  $p-1$ .

The importance of randomization in computation arises from powerful stochastic sampling theorems such as the law of large numbers. However, independence and uniformity of samples is crucial to invoking these theorems. How important is this stochastic independence to random polynomial time computation? The result that  $RP = SRP$  [VV] shows that independence can be dispensed with by using more sophisticated sampling algorithms. More precisely, this result states that any random polynomial time algorithm can be simulated in polynomial time by an algorithm whose only source of randomness is a single semi-random source (a single source is provably too weak to produce quasi-random sequences [SV]). In the second part of this paper we consider We show that any randomized algorithm that runs in time  $T(n)$  and uses  $R(n)$  random bits

can be simulated in time  $O(T(n)\log^{(k)}R(n) + R(n)\log R(n)\log^{(k)}R(n))$  and using  $O(R(n))$  semi-random bits. Here  $k$  is any constant and  $\log^{(k)}R(n)$  denotes  $\log$  iterated  $k$  times on  $R(n)$ . Clearly, the number of semi-random bits used is within a constant factor of the optimal. We conjecture a lower bound of  $T(n) + M(n)\log M(n)$  for any algorithm that achieves this result, and provide arguments that support this conjecture.

## 2. Efficient Extraction of Quasi-random Sequences

In this chapter, we consider the problem of efficiently generating quasi-random sequences from the outputs of two independent semi-random sources. The following algorithm was given in [Va] for extracting  $n$ -bit quasi-random sequence from  $O(n\log n\log^* n)$  bits output by the two sources: divide the  $O(n\log n\log^* n)$  output bits from the two sources into  $n$  pairs of blocks each of length  $O(\log n\log^* n)$ . Denote by  $x$  and  $y$  respectively the  $i^{th}$  pair of blocks output by the two sources. Then the  $i^{th}$  output bit is simply  $\bar{x}^{\rightarrow}y^{\rightarrow} \bmod 2$ .

Here we show how to re-use  $x$  and  $y$  to extract  $O(\log n\log^* n)$  "high quality" random bits. Let

$|x| = |y| = p$  where  $p$  is prime such that 2 is a primitive root mod  $p$ .<sup>1</sup> Let  $\pi^j(\bar{y}^{\rightarrow})$  denote the cyclic shift of  $\bar{y}^{\rightarrow}$  by  $j$  positions. Obtain  $O(p)$  bits  $b_j$  from  $x$  and  $y$  by letting  $b_j = \bar{x}^{\rightarrow}\pi^j(\bar{y}^{\rightarrow})$ . We wish to show that the  $b_j$ 's are (super polynomially) close to being unbiased and uncorrelated. Certainly a necessary condition for this is that the parity of any subset of the  $b_j$ 's must be close to unbiased. The parity lemma below shows that this condition is also sufficient.

Let  $D$  be any probability distribution on  $\{0, 1\}^n$ .

Let  $Pr_D[E(x)]$  denote the probability that event  $E(x)$  occurs if  $x$  is picked according to the distribution  $D$ .

**Definition:**

$$bias_i(D) = | \sum_{x \in \{0,1\}^n} Pr_D[\bar{i}^{\rightarrow}x^{\rightarrow} = 1] - Pr_D[\bar{i}^{\rightarrow}x^{\rightarrow} = 0] |$$

**Parity-Lemma:** For any distribution  $D$  on  $\{0, 1\}^n$ ,

$$\sum_{x \in \{0,1\}^n} |Pr_D[x] - \frac{1}{2^n}| \leq \sum_{i \in \{0,1\}^n - 0^n} bias_i(D).$$

**Proof:** By induction on  $n$ .

$$\text{Let } p = \sum_{z \in \{0,1\}^{n-1}} Pr_D[1z].$$

$D$  induces two distributions  $D_0$  and  $D_1$  on  $\{0, 1\}^{n-1}$ ;

<sup>1</sup>By Artin's conjecture [Sh], asymptotically 3/8 th fraction of all primes have this property. It was shown in [Ho] that Artin's conjecture follows from the ERH. Picking such a prime is easy since its length is only  $O(\log \log n)$ .

$D_0$  being the distribution on  $n$  bit strings whose first bit is 0. Now:

$$\begin{aligned}
\sum_{x \in \{0,1\}^n} |Pr_{D_0}[x] - \frac{1}{2^n}| &= \sum_{z \in \{0,1\}^{n-1}} |Pr_{D_0}[0z] - \frac{1}{2^n}| \\
&\quad + \sum_{z \in \{0,1\}^{n-1}} |Pr_{D_0}[1z] - \frac{1}{2^n}| \\
= \sum_{z \in \{0,1\}^{n-1}} |(1-p)Pr_{D_0}[z] - \frac{1}{2^n}| \\
&\quad + \sum_{z \in \{0,1\}^{n-1}} |pPr_{D_1}[z] - \frac{1}{2^n}| \\
\leq \sum_{z \in \{0,1\}^{n-1}} |(1-p)Pr_{D_0}[z] - \frac{1-p}{2^{n-1}}| \\
&\quad + \sum_{z \in \{0,1\}^{n-1}} |\frac{1-p}{2^{n-1}} - \frac{1}{2^n}| \\
&\quad + \sum_{z \in \{0,1\}^{n-1}} |pPr_{D_1}[z] - \frac{p}{2^{n-1}}| \\
&\quad + \sum_{z \in \{0,1\}^{n-1}} |\frac{p}{2^{n-1}} - \frac{1}{2^n}|
\end{aligned}$$

Applying the inductive hypothesis and simplifying, this is:

$$\begin{aligned}
&\leq |bias_{10^{n-1}}(D)| \\
&\quad + \sum_{i \in \{0,1\}^{n-1} - 0^{n-1}} (1-p)|bias_i(D_0)| + p|bias_i(D_1)| \\
&\leq |bias_{10^{n-1}}(D)| + \sum_{i \in \{0,1\}^{n-1} - 0^{n-1}} |\frac{1}{2} bias_i(D_0)| \\
&\quad + \frac{1}{2} |bias_i(D_1)| + |(1-p)bias_i(D_0) - p bias_i(D_1)| \\
&= |bias_{10^{n-1}}(D)| \\
&\quad + \sum_{i \in \{0,1\}^{n-1} - 0^{n-1}} |bias_{0i}(D)| + |bias_{1i}(D)|.
\end{aligned}$$

q.e.d.

Showing the quasi-randomness of the bit sequence  $b_1 b_2 \dots b_k$  is now reduced to showing that the parity of any subset of the  $b_i$ 's has small bias. This is more tractable since it is an assertion about a single bit. Let  $I \in \{1, \dots, k\}$ . Consider the bit  $b = \text{XOR}_{i \in I} b_i$ .

$$\begin{aligned}
\text{By definition } b &= \text{XOR}_{i \in I} \vec{x}^i \pi^i(\vec{y}) \\
&= \text{XOR}_{i \in I} \text{XOR}_{j=1}^p (x_j \text{and} y_{i+j}) \\
&= \text{XOR}_{j=1}^p \text{XOR}_{i \in I} (x_j \text{and} y_{i+j}) \\
&= \text{XOR}_{j=1}^p (x_j \text{and } \text{XOR}_{i \in I} y_{i+j}) \\
&= \vec{x}^i \sum_{i \in I} \pi^i(\vec{y}).
\end{aligned}$$

Thus the bit  $b$  is obtained simply by taking the inner product of  $\vec{x}^i$  with the sum of certain cyclic shifts of  $\vec{y}$ . We show below that taking (arbitrary) sums of cyclic shifts of such vectors is either an injective map or it is a two to one map, with vectors  $\vec{y}$  and  $1^p + \vec{y}$  mapped to the same element. In either case the inner product of  $\vec{x}^i$  with the sum of certain cyclic shifts of  $\vec{y}$  is stochastically similar to taking the inner product of two independent  $p$ -bit semi-random vectors. Thus the techniques of [Va] suffice to complete the proof that this inner product has small bias.

**Lemma 1:** The polynomial  $x^{p-1} + x^{p-2} + \dots + 1$  is irreducible over  $\text{GF}(2)$  iff  $p$  is a prime such that 2 is a primitive root mod  $p$ .

**Lemma 2:** Let  $p$  be a prime with 2 a primitive root mod  $p$ , and let  $A$  be a  $p \times p$  matrix over  $\text{GF}(2)$ . Let the rows of  $A$  be the  $p$  cyclic shifts of a vector  $\vec{u}$ , which is not the all 0's or the all 1's vector. Then  $\text{rank}(A) \geq p-1$ .

**Proof:** By the lemma above, the polynomial  $x^{p-1} + x^{p-2} + \dots + 1$  is irreducible over  $\text{GF}(2)$ . Extend  $\text{GF}(2)$  by adjoining a root of this polynomial to obtain the galois field  $\text{GF}(2^{p-1})$ . Consider the redundant representation for this field given by  $\text{GF}(2)[x]/x^p - 1$ . In this representation the polynomial  $f(x)$  and  $x^{p-1} + x^{p-2} + \dots + 1 - f(x)$  represent the same field element. Moreover, a cyclic shift of  $f(x)$  is achieved by multiplying it by  $x$ . Now the rows of  $A$  are  $u(x), xu(x), x^2u(x), \dots, x^{p-1}u(x)$ . Since we are working in a field,  $u(x)$  has an inverse which may be written as a polynomial in  $x$ . Thus  $1 = u(x) \times u^{-1}(x)$  can be written as a linear combination of  $u(x)x^i$ . Similarly  $x, x^2, \dots, x^{p-1}$  lie in the span of the rows of  $A$ . Since these have rank  $p-1$ , it follows that  $A$  has rank at least  $p-1$ .

q.e.d.

**Theorem 1:** There is an algorithm to extract  $n$ -bit quasi-random sequences from  $O(n)$  bits output by two independent semi-random sources.

**Sketch of Proof:** It remains to show that a bit  $b$  obtained by taking the inner-product of  $\vec{x}$  with  $\vec{z} = \sum_{i \in I} \pi^i(\vec{y})$  has small bias. We shall use the Main Lemma of [Va], which states that for any fixed adversary for generating  $\vec{x}$ , there are very few  $\vec{z}$ 's such that  $\vec{x} \cdot \vec{z}$  has large bias. By Lemma 2, the number of bad  $\vec{y}$ 's is at most twice as large as the bound given by the Main Lemma of [Va]; finally the theorem follows by showing that the probability of generating such a bad  $\vec{y}$  is very small (using the arguments of Theorem 2.3 [Va]).

q.e.d.

### Memory Bounded Source:

The issue of mathematically modeling the dependences of a physical source was raised in [BI], and the finite state markov process was proposed as a model. In that paper, it was shown that a counter-intuitive generalization of von Neumann's algorithm [vN] produces truly random bits from the output of such a

source. A generalization of the finite state markov process is the memory bounded source:

**Definition:** [Va] Source  $S$  is *memory bounded* if the adversary strategy is a map  $T: \{0, 1\}^m \times N \rightarrow [\delta, 1 - \delta]$ , where the first argument denotes the contents of the  $m$ -bit memory of the adversary, and the second argument is the time (assume that one bit is output per unit time). The memory is updated by the function  $g: \{0, 1\}^m \times N \times \{0, 1\} \rightarrow \{0, 1\}^m$ ; the first argument to  $g$  is the contents of the memory, the second is time and the third is the latest bit output by the source.

Notice that whereas in the case of the semi-random source, there was no need to explicitly make the strategy  $T$  a function of time, this does increase the class of distributions that can be obtained when the memory is bounded. Interpreted physically, this time dependence is very important: the bias of the output of a Zener diode depends upon operating conditions such as temperature and humidity. Such effects cannot be expressed in the bounded memory, and must be represented separately as the time dependence of the bias.

The advantage of putting this additional memory constraint on the adversary is that this makes it possible to obtain quasi-random sequences from a single

source. It was shown in [Va] how to generate  $n$ -bit quasi-random sequences from  $O(nm \log m + n \log \log^* n)$  bits of output of such a source. The algorithm was based on the following theorem:

**Theorem [Va]:** Let  $\vec{x}$  and  $\vec{y}$  denote successive  $k$ -bit blocks output by a memory-bounded source with  $m$  bits of memory. Then

$$|Pr[\vec{x}\vec{y} = 1] - 1/2| \leq (4m)^m 2^{-\delta^2 k}.$$

Here we show how to generate  $n$ -bit quasi-random sequences from  $O(n + m \log m \log^2 n)$  bits of output from such a source. The algorithm achieving this result uses a construction of Wyner [Wy]. Let  $x_1, x_2, \dots, x_k$  and  $y_1, y_2, \dots, y_k$  be two successive  $k$ -bit blocks produced by the memory-bounded source. Let  $b_i = x_i \text{ and } y_i$ . Let us extract  $l$  bits  $r_1, r_2, \dots, r_l$  as follows: each  $r_i$  is the parity of some subset of  $b_1, b_2, \dots, b_k$ . To make sure that the  $r_i$ 's are almost unbiased, the subsets must be large, and to ensure small correlations, they must be "well spread-out". This is precisely achieved by the rows of an  $l \times k$  parity-check code generator matrix (the rows are 0/1 vectors of dimension  $k$ , and represent subsets in a natural manner) which detects  $\alpha k$  errors for some constant  $\alpha > 0$  [Ga, Chapter 6]. An explicit construc-

tion of such asymptotically efficient codes appears in [Ju]; for practical values of  $k$ , the BCH code is more useful, although it is not asymptotically efficient. The following algorithm puts together these ideas:

*proc qgen(n):*

Let  $G$  be a  $n \times k$  parity check code generator matrix, which detects  $\alpha k$  errors, where  $k = \omega(n + m \log m \log^2 n)$ .

Let  $u_i = x_i$  and  $y_i$ .

Output  $G \cdot \vec{u}$ .

*end;*

**Theorem 2:** There is an algorithm that generates  $n$ -bit quasi-random sequences from  $O(n + m \log m \log^2 n)$  bits output by a memory bounded source with  $m$  bits of memory.

### 3. Efficient Simulation of Randomized Algorithms

It was shown in [VV] that any random polynomial time algorithm can be simulated in polynomial time by an algorithm whose only source of randomness is a single semi-random source. The result held for

both  $RP$  and the two-sided error  $BPP$ . The simulation time for an algorithm that ran in time  $T(n)$  and used  $R(n)$  random bits was  $O(T(n)R(n)^{1/\delta})$  and the number of semi-random bits used was  $O(R(n) \log R(n))$ . The main result in this chapter is:

**Theorem 3:** Any randomized algorithm that runs in time  $T(n)$  and uses  $R(n)$  random bits can be simulated by an algorithm using  $O(R(n))$  bits produced by a single semi-random source and in time  $O(T(n) \log^{(k)} R(n) + R(n) \log R(n) \log^{(k)} R(n))$ .

The proof of this theorem will appear in the final paper. The techniques of proof include those from the first part of this paper, as well as resampling techniques which can be found in [Va2]. The proof easily extends to the model defined in [CG].

### 4. Discussion

We conjecture that  $T(n) + R(n) \log R(n)$  is a lower-bound for any simulating algorithm. So the above algorithm is optimal to within a factor of  $\log^{(k)} R(n)$  for any constant  $k$ . It follows from a result of [SV] that any single bit that is almost unbiased must be a function of at least  $\log R(n)$  semi-random bits. To

effectively find a witness, we need  $R(n)$  bits such that the  $j^{\text{th}}$  bit is almost unbiased even though the values of the other  $R(n) - 1$  are fixed. Thus the  $j^{\text{th}}$  bit requires at least  $R(n)$  computational steps even assuming that the other  $R(n) - 1$  bits are given for free. This suggests that there is a lower bound of  $R(n) \log R(n)$ .

The shift-register (and linear congruential) pseudo-random sequence generators are known to fail the polynomial time unpredictability test. Why do they work so well in practice? Say that a pseudo-random sequence generator is quasi-perfect if on an input seed of length  $O(n)$ , its output can be used to simulate  $k$  runs of a random polynomial time algorithm (which uses  $n$  random bits per invocation and has probability  $1/2$  of success) with success probability  $1 - \frac{1}{2^k}$ , for some constant  $\epsilon$ . We conjecture that the following is a quasi-perfect pseudo-random sequence generator: Let  $\vec{x}$  and  $\vec{y}$  denote random seeds of length  $n$ . Assume that  $n$  is a prime with 2 a generator mod  $n$ . The output sequence is  $\vec{u}_1, \dots, \vec{u}_k$  where  $\vec{u}_i = \vec{x} + \pi^i(\vec{y})$ .

**Acknowledgements:** I am grateful to Miller Puckette and Vijay Vazirani for some very useful discussions. I am particularly grateful to Michael Rabin for

suggesting the proof of Lemma 1.

## References

- [Al] N. Alon, "On VV-s result  $SRP = RP$ ," unpublished manuscript, May 1985.
- [Bl] M. Blum, "Independent Unbiased Coin Flips From a Correlated Biased Source: a Finite State Markov Chain," *25th. IEEE Symposium on the Foundations of Computer Science*, 1984.
- [CG] B. Chor and O. Goldreich, "Unbiased Bits from Weak Sources of Randomness," *26th. IEEE Symposium on the Foundations of Computer Science*, 1985.
- [El] P. Elias, "The Efficient Construction of an Unbiased Random Sequence," *Ann. Math. Statist.* Vol 43, No. 3, 1972, 865-870.
- [Ga] R. Gallager, *Information Theory and Reliable Communication*, New York: John Wiley, 1968.
- [Ho] C. Hooley, "On Artin's Conjecture," *Crelle's Journal*, 225, (1967), pp 209-220.
- [Ju] J. Justesen, "A Class of Constructive Asymptotically Good Algebraic Codes," *IEEE Trans. Inform. Theory*, vol IT-18, pp 652-656, Sept



- 1972.
- [vN] J. von Neumann, "Various Techniques Used in Connection with Random Digits," Notes by G. E. Forsythe, National Bureau of Standards, Applied Math Series, 1951, Vol 12, 36-38. Reprinted in von Neumann's Collected Works, Vol 5, Pergamon Press (1963), 768-770.
- [SV] M. Santha and U. V. Vazirani, "Generating Quasi-random Sequences from Semi-random Sources," *Journal of Computer Systems and Sciences*, Vol. 33, No 1, Aug 1986, pp 75-87.
- [Sh] D. Shanks, *Solved and Unsolved Problems in Number Theory*, Chelsea Publishing Co., NY.
- [Va] U. V. Vazirani, "Towards a Strong Communication Complexity Theory or Generating Quasi-random sequences from two communicating semi-random sources," *15th Annual ACM Symp. on Theory of Computing*, pp 366-378, 1983.
- [Va2] U. V. Vazirani, "Randomness, Adversaries and Computation," Ph.D. Dissertation, U. C. Berkeley, 1986.
- [VV1] U. V. Vazirani and V. V. Vazirani, "Random Polynomial Time is Equal to Semi-Random Polynomial Time," *26th. IEEE Symposium on the Foundations of Computer Science*, 1985.
- [VV2] U.V. Vazirani and V.V. Vazirani, "Sampling a Population with a Semi-random source," *Proceedings Sixth Ann. FST-TCS Conference*, New Delhi, 1986.
- [Wy] A. Wyner, "Wire-tap Channel," *Bell System Technical Journal*, pp 1355-1387, Oct. 1975.
- [Ya2] A. Yao, "Theory and Applications of Trap-door Functions," *23th. IEEE Symposium on the Foundations of Computer Science*, 1982.