A Note on the Advice Complexity of Multipass Randomized Logspace*

Peter Dixon[1], Debasis Mandal[2], A. Pavan[3], and N. V. Vinodchandran[4]

1 **Department of Computer Science, Iowa State University**
tooplark@cs.iastate.edu
2 **Synopsys, Inc.**
debasis.mandal@gmail.com
3 **Department of Computer Science, Iowa State University**
pavan@cs.iastate.edu
4 **Department of Computer Science and Engineering, University of Nebraska-Lincoln**
vinod@cse.unl.edu

Abstract

Investigating the complexity of randomized space-bounded machines that are allowed to make *multiple passes over the random tape* has been of recent interest. In particular, it has been shown that derandomizing such probabilistic machines yields a weak but new derandomization of probabilistic time-bounded classes.

In this paper we further explore the complexity of such machines. In particular, as our main result we show that for any $\epsilon < 1$, every language that is accepted by an $O(n^{\epsilon})$-pass, randomized logspace machine can be simulated in deterministic logspace with *linear amount of advice*. This result extends an earlier result of Fortnow and Klivans who showed that RL is in deterministic logspace with linear advice.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Space-bounded computations, randomized machines, advice

Digital Object Identifier 10.4230/LIPIcs.MFCS.2016.31

1 Introduction

In the standard definition of probabilistic space-bounded computations, a probabilistic machine accesses its random tape in a *one-way*, read-only manner. In particular, the machine cannot reread the random bits unless they are stored in its work tapes. This model captures machines that can toss coins and hence is the most natural and this leads to well-studied space-bounded probabilistic classes BPL and RL [7].

While one-way access to the random tape is the most natural notion for probabilistic space-bounded computations, researchers have explored space-bounded models where the base machines are allowed to read contents of the random tape multiple times. An interesting earlier result is due to Nisan who showed that two-sided error logspace machines with one-way access to the random tape can be simulated by zero-error logspace machines that have two-way access to the random tape (BPL $\subseteq$ 2-*way*ZPL) [12]. However, the progress in understanding such machines and corresponding complexity classes has been sporadic and

* Research Supported in part by NSF grants 1421163 and 1422668.

many relations are unknown. For example, while we know that BPL is in P, we do not know whether 2-*way*BPL is even in deterministic sub-exponential time (note that it is in BPP). A key issue is that allowing two-way access to the random tape for space-bounded machines brings the corresponding nonuniform classes closer to randomized circuit complexity classes, where progress is known to be difficult.

Allowing the probabilistic machine to have *multiple passes* over the random tape is an access mechanism that is in between one-way and two-way. In particular, a $k(n)$-pass probabilistic machine is allowed to make $k(n)$ passes over the random tape for deciding an input of length n. Such probabilistic space-bounded machines were first considered by David, Papakonstantinou, and Sidiropoulos [5]. They showed that any pseudorandom generator that fools traditional $k(n)s(n)$ space-bounded machines can also fool $k(n)$-pass $s(n)$ space-bounded machines. As a corollary, they obtain that polylog-pass, randomized logspace is contained in deterministic polylog-space. Very recently, Mandal, Pavan, and Vinodchandrian [10] showed that such multipass probabilistic machines are interesting from a derandomization point of view. In particular, they showed the following theorem.

▶ **Theorem 1** ([10]). *For some constant $k > 0$, if every language decided by a probabilistic logspace machine that uses $O(\log n \log^{(k+3)} n)$ random bits and makes $O(\log^{(k)} n)$ passes over its random tape is in* P*, then* $\mathrm{BPTIME}(n) \subseteq \mathrm{DTIME}(2^{o(n)})$.

Here $\log^{(k)} n$ denotes log function applied k times iteratively. Note that showing $\mathrm{BPTIME}(n)$ is a subset of $\mathrm{DTIME}(2^{o(n)})$ is a significant open problem. Thus derandomizing a slightly non-constant pass probabilistic space-bounded machine yields a non-trivial derandomization of $\mathrm{BPTIME}(n)$.

The Main Result

This note considers the *advice complexity* of multipass probabilistic machines. Using standard techniques, it can be shown that 2-*way*RL[1] is in L/poly. Can this simulation be improved for multipass machines? Indeed, for RL, Fortnow and Klivans established that RL is in $\mathrm{L}/O(n)$. Thus one-pass logspace probabilistic machines can be simulated deterministically in logspace using *linear* advice. Our main contribution is to show that even if the base probabilistic machine is allowed n^δ passes over the random tape, the corresponding complexity class can still be simulated in $\mathrm{L}/O(n)$. More formally,

▶ **Main Theorem.** *For any $\delta < 1$, n^δ-pass* RL *is in* $\mathrm{L}/O(n)$.

This result extends Fortnow and Klivans' result and improves a result in [10] where it is shown that n^δ-pass RL is in deterministic $\log^2(n)$ space with linear advice.

2 Preliminaries

We refer the reader to [3] for standard notions and definitions of complexity theory. We first define probabilistic space-bounded computations. A probabilistic $s(n)$ space-bounded Turing Machine M has a *random tape* in addition to its input and work tapes. The machine has read-only access to both input and random tapes and it is allowed to read the contents on the random tape in a one-way manner. The total space used by the work tapes is bounded

[1] In this paper we consider one-sided error classes. Similar results can be obtained for two-sided error classes also.

by $s(n)$ and the machine can read at most $2^{s(n)}$ cells of the random tape. Thus the total number of random bits used by such machines is bounded by $2^{s(n)}$. The complexity class RL is the class of languages accepted by $O(\log n)$ space-bounded machines with one-sided error. We can relax the restriction on the machine's access to the random tape so that the machine M is allowed to read the contents of the random tape in a two-way manner. We use 2-*way*RL to denote the class that is analogous to RL but the base machine has two-way access to the random tape. In this paper, we use two-way, probabilistic machines that use limited amount of randomness. Let 2-*way*RL$[r(n)]$ denote the class of languages that are in 2-*way*RL and the base machine uses only $r(n)$ random bits on inputs of length n.

Next we define multipass, probabilistic, space-bounded machines.

▶ **Definition 2.** A probabilistic Turing machine M is a $k(n)$-*pass,* $s(n)$ *space-bounded machine* if

- M has read-only, two-way access to the input tape,
- total space used by the work tapes is bounded by $s(n)$,
- M is allowed to make $k(n)$ passes (on inputs of length n) over the random tape and during each pass it accesses the tape in a one-way, read only manner, and
- the total number of random bits used by the machine is bounded by $2^{s(n)}$.

▶ **Definition 3.** We say that a language L belongs to $k(n)$-*pass* RL if there exists a $k(n)$-pass, $O(\log n)$ space-bounded probabilistic Turing machine M such that for every input x, if $x \in L$, M accepts x with probability at least $1/2$ and if $x \notin L$, then the probability that M accepts x is 0.

Next we define the notion of advice [9].

▶ **Definition 4.** Let f be a function from natural numbers to natural numbers. A language L is in L/$f(n)$, if there is a logspace machine M and a sequence of strings $a_1, a_2, \cdots$ such that $|a_n| \leq f(n)$ and for every input x of length n, $M(x, a_n)$ accepts if and only if $x \in L$.

For a probabilistic machine M and an input x, we use $M(x; r)$ to denote the computation of M on x, where r is the contents of the random tape. We now define the notion of *pseudorandom generators* for space-bounded machines.

▶ **Definition 5.** A family of functions $G = \{G_n\}_{n\geq 0}$ is an $(m(n), r(n), \epsilon)$ pseudorandom generator for space $s(n)$ if for every probabilistic $s(n)$ space-bounded machine M that uses $r(n)$ random bits (on inputs of length n) and for every input x of length n,

$$\left|\Pr[M(x; r) = 1] - \Pr[M(x; G_n(y)) = 1]\right| \leq \epsilon,$$

where r is chosen uniformly at random from $\Sigma^{r(n)}$ and y is chosen uniformly at random from $\Sigma^{m(n)}$.

We can define a similar notion of *pseudorandom generators* for $k(n)$-pass, $s(n)$-space.

The following theorem of David *et al.* [5] connects pseudorandom generators for multipass machines to pseudorandom generators for single pass machines.

▶ **Theorem 6** ([5])**.** *Let $G = \{G_n\}_{n\geq 0}$ be an $(m(n), r(n), \epsilon)$ PRG for space $k(n)s(n)$. Then G is an $(m(n), r(n), \epsilon 2^{k(n)})$ PRG for $k(n)$-pass, $s(n)$-space.*

Our proofs use the pseudorandom generator of Babai, Nisan, and Szegedy [4] which is based on lowerbounds for multiparty communication complexity. We now define the necessary notions that are needed.

▶ **Definition 7.** Let f be a Boolean function which takes k r-bit strings $x_1, \ldots, x_k$ as inputs. Suppose k people wish to collectively compute $f(x_1, \ldots, x_k)$ with the constraint that the i^{th} person does not know x_i. The k-party, ϵ-distributional communication complexity of f, denoted by $C_\epsilon(f)$, is the minimum number of bits that must be communicated among the k people (by, say, writing on a public whiteboard) in order to compute the function f on at least $\frac{1+\epsilon}{2}$ fraction of inputs (from Σ^{kr}).

▶ **Definition 8.** The Generalized Inner Product of r k-bit strings, denoted by $\mathrm{GIP}_{r,k}(x_1, \ldots, x_k)$, is 0 if there is an even number of indices i, $1 \leq i \leq r$, at which all of $x_1[i], \ldots, x_k[i]$ are 1; otherwise $\mathrm{GIP}_{r,k}(x_1, \ldots, x_k)$ is 1.

Babai, Nisan, and Szegedy obtained the following bound on the communication complexity of GIP.

▶ **Theorem 9** ([4])**.**

$$C_\epsilon(\mathrm{GIP}_{r,k}) = \Omega\left(\frac{r}{4^k} + \log \epsilon\right)$$

3 Two-way Simulation and Linear Advice

In this section, we prove the main result of the paper which is stated below.

▶ **Theorem 10.** *Let $0 \leq \delta < 1$ be a constant. Every language L that is in n^δ-pass* RL *is in* $\mathrm{L}/O(n)$.

The proof proceeds in two steps. We first show that any n^δ-pass randomized logspace machine can be simulated by a two-way randomized logspace machine that uses only n^γ ($\delta < \gamma < 1$) random bits. We then prove that such two-way machines can be decided in deterministic logspace with a linear amount of advice. The first step is proved in Theorem 11 and the second step is proved in Theorem 13.

▶ **Theorem 11.** *For every $0 \leq \delta < 1$, there exists a γ, where $\delta < \gamma < 1$, such that for every language L in n^δ-pass* RL*, L is in* 2-*way*RL *with n^γ random bits.*

Before we present a formal proof, we give an overview of the proof. Consider the class RL. The well-known result of Nisan [11] states that there exists a PRG for logspace that stretches an $O(\log^2 n)$-length seed to $p(n)$ bits, where $p(n)$ is a polynomial in n. Moreover, the generator can be computed in *logspace*. The logspace machine that computes the PRG accesses the seed in a two-way manner. Thus using this generator we obtain that RL is in 2-*way*RL$[O(\log^2 n)]$. In fact, this is the first step in the work of Fortnow and Klivans [6]. A natural approach to Theorem 11 is to use a PRG for multipass, space-bounded machines. Let M be an n^δ-pass, logspace machine. By Theorem 6, any PRG for $O(n^\delta \log n)$-space will also be a PRG for M. Since M uses at most polynomially many random bits, we only need a PRG for $O(n^\delta \log n)$-space machines that use a polynomial number of random bits (as opposed to potentially $2^{O(n^\delta \log n)}$ random bits). A natural candidate is a generalized version of Nisan's generator that uses $O(S \log R)$ seed length for space S machines that use R random bits. This leads to a PRG that stretches $O(n^{\delta'})$ bits to polynomially many bits (for some $\delta' > \delta$), and this PRG fools the multipass machine M. Thus M can be simulated by a two-way randomized machine that uses $O(n^{\delta'})$ random bits. However, there is a small caveat in this argument. The space needed to compute this PRG is $O(\log^2 n)$. Thus this simulation of M using a two-way probabilistic machine takes $O(\log^2 n)$ space, whereas our goal is to simulate M using a two-way probabilistic machine that only uses $O(\log n)$ space.

We get around this problem by using the generator due to Babai, Nisan, and Szegedy [4]. They exhibited a PRG for space S that stretches $2^{\sqrt{S}}$-bits to 2^S bits and this PRG can be computed in $O(S)$ space. More specifically, when applied to logspace, their generator uses $O(2^{\sqrt{\log n}})$ seed length and is inefficient compared to Nisan's generator (in seed length). However, we observe that their generator is much easier to compute than Nisan's generator. We show that the BNS generator for $O(n^\delta \log n)$-space machines that use only polynomially many random bits has a seed length of n^γ ($\delta < \gamma < 1$) and can be computed in $O(\log n)$ space.

We now provide a formal proof.

Proof. Let M be an n^δ-pass, RL machine that accepts a language L. Assume that M uses n^ℓ random bits on inputs of length n. We will use the BNS generator to reduce the number of random bits used by M.

▶ **Theorem 12** ([4]). *Let $f_{r,k} : \Sigma^{rk} \to \{0,1\}$ be a Boolean function, $t > k$, and $N \leq \binom{t}{k}$. There is an $(rt, N, N\epsilon)$-pseudorandom generator G for s space-bounded machines that use N random bits, where $s < C_\epsilon(f)/k$. Also, the space required to compute G is the space required to compute the bits of f in antilexicographic order.*

We invoke this theorem for our choice of parameters. By Theorem 6, any $(m(n), n^\ell, \frac{1}{4\times 2^{n^\delta \log n}})$ generator for space $n^\delta \log n$ is an $(m(n), n^\ell, 1/4)$ generator for machine M. Note that we need a generator for $n^\delta \log n$ space-bounded machines that uses only n^ℓ random bits. Let $1 > \delta'$ be a constant that is greater than δ. We choose $r = n^{\delta'}$, $k = 2\ell\sqrt{\log n}$, $t = 2^k$, $N = n^\ell$, and the function f to be $\mathrm{GIP}_{k,r}$. By Theorem 9, the ϵ-distributional communication complexity of f is at least $\frac{r}{4^k} + \log \epsilon$. We pick ϵ as $\frac{1}{n^\ell} \times \frac{1}{4\times 2^{n^\delta \log n}}$. Thus $C_\epsilon(f)$ is at least

$$c\left(\frac{n^{\delta'}}{4^{2\ell\sqrt{\log n}}} - \ell \log n - 2 - \delta \log^2 n\right)$$

for some constant $c > 0$. With this we have $C_\epsilon(f)/k > c\left(\frac{n^{\delta'}}{4^{2\sqrt{\log n}}} - \ell \log n - 2 - \delta \log^2 n\right)/k > n^\delta \log n$ and $n^\ell < \binom{t}{k}$. Thus, by Theorem 12, there is an $(n^{\delta'} \times 2^{2\ell\sqrt{\log n}}, n^\ell, \frac{1}{4\times 2^{n^\delta \log n}})$-generator G for $n^\delta \log n$-space that uses n^ℓ random bits. Let γ be a constant that is greater than δ' (and less than 1). By theorem 6, G is an $(n^\gamma, n^\ell, 1/4)$-generator for n^δ-pass RL machines and hence fools M.

Our 2-*way*RL machine that simulates M works as follows. On any input x of length n, it has n^γ-bits written on its random tape. Let r denote the random string. It keeps track of the index of the random bit that M attempts to read. When M asks for ith random bit, it invoke the BNS generator on r, compute the ith bit of the generator, and continue the simulation of M. Note that the space needed by this machine is bounded by space needed by M plus the space needed to compute a bit of the BNS generator. We now claim that each bit of the generator can be computed in $O(\log n)$ space.

▶ **Claim 12.1.** *Each bit of the BNS generator can be computed in $O(\log n)$ space.*

Proof. The input to the BNS generator is an rt bit string which is viewed as t strings each of length r. Let $x_1, x_2, \cdots, x_t$ be these strings. We will describe the i^{th} bit of BNS generator. Consider the first $N = n^\ell$ k-subsets of $\{1, 2, \cdots, t\}$ in antilexicographic order: for two sets A and B, $A < B$ iff the largest element in $A \triangle B$ is an element of B. Let $S_1, S_2, \cdots, S_N$ be these subsets. Let $S_i = \{i_1, i_2, \cdots, i_k\}$. Then the i^{th} bit of BNS generator is $\mathrm{GIP}_{k,r}(x_{i_1}, x_{i_2}, \cdots, x_{i_k})$. Note that each S_i can be stored in $O(k \times \log t) = O(\log n)$ bits.

Also, $\mathrm{GIP}_{k,r}$ is computable in $\log k + \log r = O(\log n)$ space. We will describe an $O(\log n)$ space algorithm that takes set A and outputs B which is the next set in the antilexicographic order. Given A, initialize B with the maximal k-set. It generates all N k-subsets one by one and replaces B with the current set C if $A < C < B$. Note that given A and B, checking whether $A < B$ can be done in logspace. This leads to a logspace algorithm for generating the next set in the antilexicographic order. ◀

Thus we can simulate the n^δ-pass machine M using n^γ random bits in $O(\log n)$ space. Note that the simulating machine needs to access the random tape in a two-way manner. This completes the proof of the theorem. ◀

▶ **Theorem 13.** $2\text{-}way\mathrm{RL}[O(n)] \subseteq \mathrm{L}/O(n)$.

Proof. Let L be a language in $2\text{-}way\mathrm{RL}[O(n)]$ and let M be a machine that witnesses this with error probability $\leq 1/2$. The idea is to reduce the error probability of M to $1/2^{2n}$ using additional $O(n)$ random bits. Then a standard counting argument implies that there exists an $O(n)$-length string y for which $M(x; y)$ is correct on all strings x of length n. Thus y can be used as an advice. For reducing the error probability, we will use a *space-efficient expander walk* given by Gutfreund and Viola [8].

The general technique of using constant degree expander graphs to reduce the error probability of probabilistic machines is due to Ajtai, Komlos, and Szemeredi [1]. Let $r(n)$ denotes the number of random bits used by M on inputs on length n. Let G be a constant degree expander over $2^{r(n)}$ vertices. Consider the following process of producing k vertices: randomly pick a node v_0. For $1 \leq i \leq k-1$, v_i is a random neighbor of v_{i-1}. Note that each v_i is described using an $r(n)$ bit string. Also, the total random bits used in this process is $r(n) + O(k)$.

Consider the following simulation of M by M': on input x of length n, M' simulates M k times where the i^{th} simulation uses the encoding of v_i as the random string. If one of the simulations accept, M' accepts. It is well known that for any constant degree expander, there is a k where $k = O(n)$ so that the error probability of M' is $1/2^{2n}$.

▶ **Theorem 14** ([1]). *Given a constant degree expander G, there is a k where $k = O(n)$ so that the error probability of the above simulation is $\leq 1/2^{2n}$.*

To make this work we need be able to perform the random walk in logspace. The following theorem due to Allender, Jiao, Mahajan, and Vinay shows that random walk on certain constant degree expanders can be done in logspace. In a latter work, Gutfreund and Viola show that random walks on certain constant degree expanders can be done in $\mathrm{AC}_0[2]$.

▶ **Theorem 15** ([2, 8]). *There exist an infinite family $\{G_n\}_{n\geq 0}$ of expander graphs where G_n has 2^n nodes and constant degree D, and an $O(\log n)$-space algorithm A such that A on input $v_0 \in \{0,1\}^n$ and indices $\ell_1, \cdots, \ell_k$ where $1 \leq \ell_i \leq D$, outputs $v_0, v_1, \cdots, v_k$ where v_i is the ℓ_i^{th} neighbor of v_{i-1}. The algorithm runs in space $O(\log n + \log k)$.*

Proof of Theorem 13 follows from Theorem 14 and Theorem 15. ◀

Acknowledgements. We thank anonymous reviewers for helpful comments which improved the presentation of the paper.

References

1. M. Ajtai, J. Komlos, and E. Szemeredi. Deterministic Simulation in LOGSPACE. In *Proc. 19th ACM Symposium on Theory of Computing (STOC)*, volume 2, pages 132–140, 1987. doi:10.1145/28395.28410.
2. Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theor. Comput. Sci.*, 209(1-2):47–86, 1998.
3. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.1088/1742-6596/1/1/035.
4. L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992. doi:10.1016/0022-0000(92)90047-M.
5. M. David, P.A. Papakonstantinou, and A. Sidiropoulos. How strong is Nisan's pseudorandom generator. *Information Processing Letters*, 111(16):804–808, 2011. doi:10.1016/j.ipl.2011.04.013.
6. L. Fortnow and A.R. Klivans. Linear advice for randomized logarithmic space. In *Proc. STACS*, 2006. URL: http://link.springer.com/chapter/10.1007/11672142_38.
7. J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977. URL: http://epubs.siam.org/doi/abs/10.1137/0206049.
8. D. Gutfreund and E. Viola. Fooling parity tests with parity gates. In *Proc. APPROX and RANDOM*, pages 381–392. Springer-Verlag Berlin, Heidelberg, 2004.
9. R.M. Karp and R.J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. 12th ACM Symp. on Theory of Computing*, pages 302–309, 1980.
10. Debasis Mandal, A. Pavan, and N. V. Vinodchandran. On probabilistic space-bounded machines with multiple access to random tape. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*, pages 459–471, 2015.
11. N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. doi:10.1007/BF01305237.
12. N. Nisan. On read once vs. multiple access to randomness in logspace. *Theoretical Computer Science*, 107(1):135–144, 1993. URL: http://www.sciencedirect.com/science/article/pii/030439759390258U.