Local expansion of vertex-transitive graphs and random generation in finite groups

László Babai [*]
Department of Computer Science
University of Chicago
Chicago, IL 60637, U.S.A.
and
Department of Algebra
Eötvös University
Budapest, Hungary H-1088

E-mail: laci@gargoyle.uchicago.edu

Abstract

Heuristic algorithms manipulating finite groups often work under the assumption that certain operations lead to "random" elements of the group. While polynomial time methods to construct uniform random elements of permutation groups have been known for over two decades, no such methods have previously been known for more general cases such as matrix groups over finite fields.

We present a Monte Carlo algorithm which constructs an efficient nearly uniform random generator for finite groups G in a very general setting. The algorithm presumes *a priori* knowledge of an upper bound n on $\log |G|$.

The random generator is constructed and works in time, polynomial in this upper bound n. The process admits high degree of parallelization: after a preprocessing of length $O(n \log n)$ with $O(n^4)$ processors, the construction of each random element costs $O(\log n)$ time with $O(n)$ processors.

We use the computational model of "black box groups": group elements are encoded as $(0,1)$-strings of uniform length; and an oracle performs group operations at unit cost. The group G is given by a list of generators. The random generator will produce each group element with probability $(1/|G|)(1 \pm \epsilon)$ where ϵ can be prescribed to be an arbitrary exponentially small function of n.

[*] Research supported in part by NSF Grant CCR-8710078.

The result is surprising because there does not seem to be any hope to estimate the order of a matrix group in polynomial time. A number of previous results have indicated close connection between nearly uniform random generation and approximate counting.

The proof involves elementary combinatorial considerations for finite groups as well as linear algebra and probabilistic techniques to analyse random walks over vertex-transitive graphs, i.e. graphs with all vertices "alike" (equivalent under the action of the automorphism group). The key tool is a local expansion lemma for groups, which generalizes to vertex-transitive graphs.

As a by-product, we obtain fairly general results on random walks on vertex-transitive graphs which may be of interest in their own right.

1 Introduction

1.1 Random generation in finite groups

In manipulating finite groups, it is often desirable to have access to uniformly distributed random elements of the group.

In [Ba1], "strong generators" for a chain of subgroups is constructed in polynomial time under the assumption of access to to random elements; an assumption justified by an application to a subcase of graph isomorphism. A particularly efficient version of this algorithm, exploiting random elements with great ingenuity, was found in [CFS]. Neumann and Praeger [NP] have recently constructed efficient algorithms for certain matrix group problems assuming access to random elements.

Heuristic algorithms often operate under the assumption that certain elements are "random". Group theory packages such as CAYLEY and GAP include routines that are fast under the assumption of access to random elements. On the other hand, in some cases, examples can be constructed to show that the supposedly random elements they use are so highly non-random that the algorithm is exponentially likely to output the wrong result.

For permutation groups (given by a list of generators), standard basic algorithms due to Sims [Sim] suffice in order to construct uniformly distributed random elements in polynomial time (cf. [FHL], [Je], [Kn], and the recent considerable speedup [BCFLS].) It is crucial for those methods that a permutation group G of degree n (= the number of elements permuted) possesses a subgroup chain $G = G_0 \geq G_1 \geq \ldots \geq G_n = 1$ with small jumps: $|G_{i-1} : G_i| \leq n$. This fact can be interpreted as a kind of *self-reducibility* of permutation groups and is largely responsible for the sizable polynomial time library available for permutation groups (cf. [KL]).

The situation is drastically different for another, potentially more important, class of representations of finite groups: *matrix groups* over finite fields. Such groups in general do not have subgroups of small index. While most finite simple groups are defined as matrix groups (cf. [Ca]), current algorithmic techniques to handle them first convert them into permutation groups, thus incurring a tremendous blowup of the size of the representation. For a group G of $d \times d$ matrices over the field of q elements, typically $|G| = q^{\Theta(d^2)}$, and the elements of G are represented as strings of length $\Theta(d^2 \log q)$. So this encoding of the group elements is optimal. On the other hand, these groups typically act on permutation domains of size at least $n = q^{\Theta(d)}$, exponentially large compared to the matrix representation. This effectively rules out handling matrix groups even of modest dimension.

It should therefore be of particular interest to perform efficient group computations in the matrix representation itself. *Generating random elements* in a group of which we have *no hope of determinig the approximate order* (cf. Section 9) might seem an exaggerated goal. Yet, we solve this problem in polynomial time in an even more general setting. While the random elements we construct will be slightly non-uniformly distributed, such a small (and prescribable) deviation from uniformity could hardly affect the potential applications.

1.2 Black box groups: the cost of random generation

Our model of computation is that of "black box groups". Group elements are encoded somehow (preferably by strings of uniform length, but the nature of the encoding is irrelevant for our discussion). Group operations (multiplication, inverse) are performed by an oracle (the black box). A "black box group" G is given by a list of generators. Our cost measure comprises three elements: the number of oracle calls (group operations), the cost of ordinary computation that controls the oracle calls, and the number of random bits used.

The algorithm presumes *a priori* knowledge of an upper bound N on the order of G. (In the matrix group case, $N = q^{d^2}$ is a convenient upper bound; typically $\log |G| = \Theta(\log N)$ in this case. More generally, for a black box group with elements encoded as binary strings of length n, we may set $N = 2^n$.) All the three cost measures will be bounded by a polynomial of $\log N$, clearly the best we can hope for, up to the implied constant in the exponent. In fact, while this statement describes the cost of the preprocessing phase (setting up the random generator), the cost per random element will be $O(\log N)$ only.

The random group elements we generate will not be truly uniformly distributed. But their deviation from the uniform distribution can be made arbitrarily small: if each group element is to have probability $(1/|G|)(1 \pm \varepsilon)$ to be selected, the costs will be polynomial in $\log N$ and $\log(1/\varepsilon)$.

To be more precise, our algorithm is Monte Carlo. If we wish that the algorithm succeed with probability $\geq 1 - \delta$ in constructing a random generator, uniform within $(1 \pm \varepsilon)$ in the above sense, then the cost is polynomial in $\log N$, $\log(1/\varepsilon)$, and $\log(1/\delta)$.

The cost per random group element requested will be $O(\log N + \log(1/\varepsilon) + \log(1/\delta))$ (after preprocessing).

In Section 9 we show that it is *impossible to determine the approximate order of black box groups.* Indeed the situation is so bad one cannot tell elementary abelian groups of such wildly differing orders as nearly 2^n and about $2^{\sqrt{n}}$ apart with a polynomial number of oracle queries.

This makes our main result more surprising since for self-reducible languages, approximate counting and nearly uniform random generation are known to be equivalent [JVV].

1.3 Erdős - Rényi generators and straight line programs

Let G be a finite group and S a set of generators of G. A *straight line program* in G with respect to S is a sequence of group elements $g_1, \ldots, g_m$ such that each g_i is either a member of S, or the inverse of g_j for some $j < i$, or a product $g_j g_k$ for some $j, k < i$.

The *Reachability Lemma* [BSz] (cf. Section 6.2 below) asserts that every element of a group G can be reached by a straight line program of length $m \leq (1 + \log |G|)^2$. (Throughout this paper, log stands for base 2 logarithms.) The lemma states the existence of such a straight line program but does not say how to construct one. (If we knew how to, we would in particular solve the *discrete logarithm* problem.)

The main result of this paper can be viewed as an efficient version of the Reachability Lemma. We show how to construct a straight line program that, starting from an arbitrary set of generators of G, leads to a set of $O(\log N)$ elements, from which nearly uniformly distributed random elements of G can be obtained at the cost of only $O(\log N)$ multiplications per random element. (As before, N is the upper bound on the order of G known *a priori.*)

Let $g_1, \ldots, g_k$ be a sequence of group elements. By a *subproduct* of this sequence we mean an element of the form $g_1^{e_1} \cdots g_k^{e_k}$, where $e_i \in \{0, 1\}$. The set of subproducts is the *cube* $C(g_1, \ldots, g_k) \subseteq G$ based on this sequence. A *random subproduct* is a subproduct obtained by choosing the exponents e_i by independent flips of a fair coin. Note that these products are not necessarily uniformly distributed over the cube. We shall be interested in the case when they are nearly uniformly distributed.

A probability distribution over a set S is called *ε-uniform* if each element is selected with probability $(1/|S|)(1 \pm \varepsilon)$, i.e. with probability between $(1/|S|)(1 - \varepsilon)$ and $(1/|S|)(1 + \varepsilon)$.

We call a sequence of elements $h_1, \ldots, h_k \in G$ a *sequence of ε-uniform Erdős - Rényi generators* if every element of G is represented in $(2^k/|G|)(1 \pm \varepsilon)$ ways as a subproduct of the h_i. In other words, we require that random subproducts are ε-uniformly distributed over G.

Erdős and Rényi [ER, Theorem 1] proved that for

$$k \geq 2 \log |G| + 2 \log(1/\varepsilon) + \log(1/\delta), \tag{1}$$

a sequence of k random elements of G will be a sequence of ε-uniform Erdős - Rényi generators with probability $\geq 1-\delta$.

The target of the straight line programs to be constructed is a short sequence of ε-uniform Erdős - Rényi generators for G for any $\varepsilon > 0$, where "short" means length

$$k = 2\log N + 2\log(1/\varepsilon) + \log(1/\delta). \tag{2}$$

(Here, δ is the reliability parameter just mentioned, bounding the probability of failure of our Monte Carlo algorithm to produce ε-uniform Erdős - Rényi generators.) It is clear that once such generators have been constructed, ε-uniform random elements of G cost k random bits and $\leq (k-1)$ multiplications each.

1.4 The ingredients

The basic idea is to emulate the "cube doubling" technique of the proof of the Reachability Lemma [BSz] which we review in Section 6.2.

The method requires reaching, in each phase, elements that cannot be represented as too short words in the current generators. This will be achieved through the analysis of *random walks in groups.* The key ingredient in this analysis is a *local expansion property* of groups (Section 3). This property has independent interest in its own right and has already been applied in the context of interactive proofs [Ba2], [Ba3]. More recently it has played a key role in a very fast (nearly linear time) Monte Carlo algorithm to handle permutation groups with a small base (a case of particular importance in computational group theory) [BCFS]. Curiously, while in the present paper we have to assume an *a priori* bound on the order of G in order to know when to terminate the algorithm, under the circumstances of [BCFS] the local expansion property is utilized exactly in order to make such an a priori bound unnecessary.

The local expansion property generalizes (easily) to all vertex-transitive graphs (even to infinite ones with some restrictions), with implications of quite general nature on random walks. (In a vertex-transitive graph, all vertices are equivalent under automorphisms; cf. Section 2).

The local expansion property is employed to show that random walks over a vertex-transitive graph have a fair chance of being reasonably far from their origin at a random time within a short period. The proof of this fact involves elementary probabilistic arguments exploiting the symmetry of the graph (Section 5); and a linear algebra argument exploiting local expansion (Section 4). The latter requires a variant of the Cheeger-type [Ch] eigenvalue bound of Alon [Alo], which in turn is used to deduce fast exit of a random walk from an expanding subgraph (Section 4).

In Section 6.3 we indicate, reviewing ideas from [BCFLS], how to reduce, if necessary, the number of input generators to $O(\log N)$.

Having completed the description of the ingredients of Phase One of the algorithm, we describe this phase in Section 7. The result there is a set of $O(\log N)$ generators such that every element of G is representable as a product of length $O(\log N)$ of these generators.

In Section 6.1 we show that such a set suffices to reach nearly uniform random group elements by random walks of length, polynomial in $\log N$. This will follow from the expansion property and Alon's mentioned eigenvalue bound [Alo].

In Section 8.1 we review the required modification of the Erdős - Rényi result necessitated by the fact that the random elements generated by Phase One are not strictly uniform. Section 8.2 completes the description of the algorithm and its analysis.

We close this section with stating the main result of the paper.

Theorem 1.1 *Let $c, C > 0$ be given constants, and let $\varepsilon = N^{-c}$ where N is a given upper bound on the order of the group G. There is a Monte Carlo algorithm which, given any set of generators of G, constructs a sequence of $O(\log N)$ ε-uniform Erdős-Rényi generators at a cost of $O((\log N)^5)$ group operations. The probability that the algorithm fails is $\leq N^{-C}$.*

If the algorithm succeeds, it permits the construction of ε-uniformly distributed random elements of G at a cost of $O(\log N)$ group operations per random element.

The number of random bits required is $O(\log\log N)$ bits per group operation. The local computation consists merely of storing the labels of group elements considered and is therefore bounded by $O(n)$ time per group operation, where n is the length of the codewords representing each group element.

The proof of Theorem 1.1 will be completed in Section 8. For a contrast, we prove in Section 9 that in the "black box group" model, it is impossible to obtain even a rough estimate of the order of the group within polynomial time.

2 Definitions, notation

All graphs in this paper are undirected. We consider finite graphs and groups only, unless otherwise stated. Some of the results remain valid for the infinite case with some restrictions (see the remarks in each section).

Throughout the paper, X will denote a graph with vertex set V. For $v \in V$, the *ball* of radius t about v is the set $\Gamma^t_X(v) = \{u \in V : \text{dist}_X(v,u) \leq t\}$, where $\text{dist}_X(v,u)$ denotes the length of the shortest path in X between v and u. We omit the subscript X if the graph of reference is clear from the context. The diameter $\text{diam}(X)$ is the greatest distance between pairs of vertices of X.

The degree $\deg(v)$ of vertex v is the number of its neighbors; so $|\Gamma^1(v)| = \deg(v) + 1$. A graph is *regular* if each vertex has the same degree.

A *random walk* over X is a Markov chain with V as the set of states; from vertex v, a transition to each neighbor is allowed with probability $1/\deg(v)$. Random walks over undirected graphs are *reversible* Markov chains.

The *boundary* of a subset $D \subseteq V$ is the set $\partial D = \{w \in V \setminus D : w$ is adjacent to some $v \in D\}$.

Let $U \subset V$ be such that for every subset $W \subseteq U$ we have $|\partial W| \geq \varepsilon |W|$. Such a subset is called *ε-expanding.* Let Y denote the subgraph induced by U. We call Y an ε-expanding subgraph.

The graph X is an *ε-expander* if every subset $U \subset V$ with $|U| \leq |V|/2$ is ε-expanding. (Alon [Alo] calls these graphs *ε-magnifiers.*)

The *Cayley graph* $C(G, S)$ of the group G with respect to the set S of generators has G for its vertex set; two vertices $g, h \in G$ are adjacent iff $sg = h$ for some $s \in S \cup S^{-1}$. The assumption that S generates G ensures that $C(G, S)$ is connected.

The *automorphisms* of the graph X are its self-isomorphisms, i.e. those permutations $V \to V$ which preserve both adjacency and nonadjacency of pairs of vertices. The automorphisms form a group $\mathrm{Aut}(X)$ under composition; this is a subgroup of $\mathrm{Sym}(V)$, the symmetric group acting on V.

A permutation group $G \leq \mathrm{Sym}(V)$ is *transitive* if for every pair v, w of elements of the permutation domain V there exists $g \in G$ such that $v^g = w$. A graph X is *vertex-transitive* if $\mathrm{Aut}(X)$ is transitive. Informally this means that all vertices are alike, a condition we shall frequently use. It implies for instance that the expected time a random walk starting at $v \in V$ takes to exit $\Gamma^t(v)$ does not depend on v.

The group G acts on the Cayley graph $C(G, S)$ by right translations $\rho_g : x \mapsto xg$ $(g, x \in G)$. Hence all Cayley graphs are vertex-transitive. (The converse is false; the smallest counterexample is Petersen's graph.)

For additional definitions, see esp. Section 1.3 (ε-uniform probability distribution, Erdős - Rényi generators, straight line program, random subproducts, the cube over a sequence of group elements etc.).

3 Local expansion of vertex-transitive graphs

The following lemma is stated as Lemma 10.2 in [Ba3]. A weaker version was announced in [Ba2, p.428]. This lemma is true for finite as well as infinite groups.

Lemma 3.1 (Local Expansion of Groups). *Let S denote a set of generators of the group G, and set $T = S \cup S^{-1} \cup \{1\}$. Let D be any finite subset of T^t, the set of t-term products of members of T (in any order). Let, finally, $0 < \alpha \leq 1/(2t+1)$ be such that*

$$|D| \leq (1 - 2\alpha t)|G|. \tag{3}$$

Then for at least one generator $g \in S$,

$$|D \setminus Dg| \geq \alpha |D|. \tag{4}$$

For completeness, we include the short proof.

Proof. For a contradiction, suppose (4) fails for every $g \in S$.

The fact that S generates G means that $G = \bigcup_{k \geq 0} T^k$.

Let us observe that for each $g \in S$, $|D \setminus Dg^{-1}| = |Dg \setminus D| = |D \setminus Dg| < \alpha|D|$. Observing in addition that

$$D \setminus Dxy \subseteq (D \setminus Dy) \cup (D \setminus Dx)y, \tag{5}$$

it follows by induction on k that for any $u \in T^k$, we have

$$|D \setminus Du| < k\alpha|D|. \tag{6}$$

As long as $k\alpha \leq 1$, this implies that $u \in D^{-1}D$. Since $\alpha \leq 1/(2t+1)$, we can choose k equal to $2t+1$ and so it follows that $T^{2t+1} \subseteq D^{-1}D \subseteq T^{2t}$ and therefore $T^{2t} = T^{2t+1} = \ldots = G$.

Next we observe that for any $u \in G$, the number of $x \in D$ such that $xu \in D$ is greater than $(1-2\alpha t)|D|$. This is the case because $u \in T^{2t}$ and thus $|D \setminus Du| < 2\alpha t|D|$.

Consequently, the number of pairs (x, u) such that $x \in D$ and $xu \in D$ is greater than $(1 - 2\alpha t)|G||D|$. On the other hand, the number of such pairs is precisely $|D|^2$. Hence

$$(1 - 2\alpha t)|G||D| < |D|^2, \tag{7}$$

contradicting assumption (3). ♠

This lemma states that Cayley graphs have a certain local expansion property which we make explicit below. The result extends in a very simple way to all vertex-transitive graphs. Recall that $\Gamma^t(v)$ denotes the set of vertices at distance $\leq t$ from $v \in V$.

Theorem 3.2 (Local Expansion of Vertex-Transitive Graphs). *Let X be a connected vertex-transitive graph with vertex set V. If $D \subseteq \Gamma^t(v)$ and $|D| \leq |V|/2$ then $|\partial D| \geq |D|/(4t)$.*

In other words, under these conditions, the set D is $1/(4t)$-expanding.

Proof. Let us first consider the case when X is a Cayley graph of a group G with respect to the set S of generators. Then, setting $\alpha = 1/(4t)$ in the previous lemma we obtain that $|gD \setminus D| \geq |D|/(4t)$ for at least one of the generators

$g \in G$. (We use the lemma with multiplication in the reverse order, clearly an equivalent statement.) But then $gD \setminus D \subseteq \partial D$ completing the proof in this case.

For the general case, let G be the automorphism group of X. Let us fix the vertex v in X. Take the following $|G|/|V|$-fold cover Y of X: set $V(Y) = G$; and join $g, h \in G$ if v^g and v^h are either adjacent in X or they coincide. (Y is the *lexicographic product* of X and the complete graph on $|G|/|V|$ vertices.) Clearly, Y is connected, and G acts on Y by right translations. Therefore Y is a Cayley graph of G. Let $\pi : G \to V$ be the projection defined by $\pi(g) := v^g$ ($g \in G$). We observe that π preserves distances, with the exception of the cases when $\mathrm{dist}_Y(g,h) = 1$ and $\mathrm{dist}_X(\pi(g), \pi(h)) = 0$. In particular, $D \subseteq \Gamma_X^t(v)$ implies $\pi^{-1}(D) \subseteq \Gamma_Y^t(1)$. We can thus apply the result to the set $\pi^{-1}(D)$ in the Cayley graph Y. Noting that $\partial(\pi^{-1}(D)) = \pi^{-1}(\partial(D))$ we conclude that $|\partial D|/|D| = |\pi^{-1}(\partial D)|/|\pi^{-1}(D)| \geq 1/(4t)$. ♠

Remark 3.3. Lemma 3.1 remains valid for infinite groups provided the set S of generators is finite. Our proof of Theorem 3.2 remains valid if X is locally finite (the vertices have finite degree) and Aut$X$ has a transitive subgroup G such that the stabilizer G_v of a vertex is finite.

We note that some of the most studied random walks are over Cayley graphs of infinite groups (cf. [MW], [Va]).

Remark 3.4. Theorem 3.2 implies that every *vertex-transitive graph of diameter* Δ is a $1/(4\Delta)$-expander. This result can be improved by a factor of 2, using a result of D. Aldous [Ald].

Proposition 3.5. *Let X be a vertex-transitive graph of diameter Δ. Then X is a $1/(2\Delta)$-expander.*

Proof. For Cayley graphs, this is stated as Lemma 3.1 in [Ald]. The reduction of the general case to Cayley graphs is identical with the corresponding argument in the proof of Theorem 3.2. ♠

Remark 3.6. For completeness, we indicate the short and elegant proof of Aldous for the case of Cayley graphs.

Let $X = C(G,S)$ be a Cayley graph of diameter Δ; and $A \subset G$. First we observe that for some $x \in G$,

$$|A \cap Ax| \leq |A|^2/|G|, \tag{8}$$

since, as seen by easy counting, the average over $x \in G$ of the left hand side is equal to the right hand side. (This observation, rediscovered in [Ald] and also in [CFS], is implicit in [ER, pp. 133-134] and is explicitly stated in [BE, Lemma].)

If $|A| \leq |G|/2$, it follows that $|Ax \setminus A| \geq |A|/2$. We now have $x = g_1 \cdots g_d$ for some $d \leq \Delta$ and $g_i \in S \cup S^{-1}$. Let $x_i = g_i \cdots g_d$. Observing that

$$|Ax \setminus A| \leq \sum_{i=1}^{d} |Ax_i \setminus Ax_{i+1}| = \sum_{i=1}^{d} |Ag_i \setminus A|, \tag{9}$$

we infer that $|Ag_i \setminus A| \geq |A|/(2\Delta)$ for some g_i, thus proving that X is a $1/(2\Delta)$-expander. ♠

4 Rapid exit from expanding subgraphs: the eigenvalue bound

In this section we show that random walks tend to exit rapidly from expanding subsets. The result is analogous to the fact that random walks over expanders mix rapidly.

Lemma 4.1. *Let λ denote the largest eigenvalue of the adjacency matrix of an ε-expanding subgraph Y of a regular graph X of degree d. Then $\lambda \leq d - \varepsilon^2/(4 + 2\varepsilon^2)$.*

This is a local variant of Alon's Cheeger-type inequality [Alo, Lemma 2.4]. The difference is that Alon requires the graph itself to be expanding in the sense that every subset U of V with $|U| \leq |V|/2$ has boundary $|\partial U| \geq \varepsilon|U|$. His conclusion is that the second largest eigenvalue of X is $\geq d - \varepsilon^2/(4 + 2\varepsilon^2)$. (The largest eigenvalue of X is d.)

The proof of this lemma follows, *mutatis mutandis*, the steps of Alon's proof. We indicate the required alterations.

Proof (sketch). Consider an eigenvector corresponding to λ. The components of this vector are labeled by the vertices in U; let x_i denote the component corresponding to $i \in U$. Then

$$\lambda = (\sum x_i x_j)/(\sum x_i^2), \tag{10}$$

where the summation in the numerator extends over all adjacent pairs of vertices $i, j \in U$. For convenience, let us introduce additional variables x_i for $i \in \partial U$ and set them to zero: $x_i = 0$ ($i \in \partial U$). This way equation (10) remains in force. We can now rewrite equation (10) as follows:

$$d - \lambda = (1/2)(\sum (x_i - x_j)^2)/(\sum x_i^2), \tag{11}$$

where the summation in the numerator extends over all adjacent pairs of vertices $i, j \in U \cup \partial U$.

Adapting Alon's trick, we use the max flow-min cut theorem to obtain a flow along which the terms x_i^2 can be broken down to edgewise increments "going toward ∂U".

Let $W = \{i' : i \in U \cup \partial U\}$ be a disjoint copy of $U \cup \partial U$.

Consider the graph F with vertex set $\{u_0, u_\infty\} \cup U \cup W$ where u_0 is a new source, joined to each vertex in U with an edge of capacity $1+\varepsilon$; u_∞ is a new sink joined to each vertex in W by an edge of unit capacity, and we have unit capacity edges of the form (i, j') where either $i = j$ or i, j are adjacent in X $(i \in U, j \in W)$.

Claim. The maximum (u_0, u_∞)-flow in this network has value $|U|(1+\varepsilon)$.

We omit the easy proof which which closely follows Alon's argument. The rest of the proof is identical with Alon's, with minor difference in notation. For the reader's convenience, we recite the proof.

Let now $h(i, j)$ denote the value of an optimum flow through the edge (i, j') for $i, j \in U \cup \partial U$. (We set $h(i, j) = 0$ when $i \in \partial U$.)

Then by Kirchhoff's law, for every $i \in U$,

$$\sum_j h(i, j) = 1 + \varepsilon, \tag{12}$$

since the flow value $h(u_0, i) = 1+\varepsilon$. (The summation extends over the neighbors of i in X.)

Since $0 \le h(i, j) \le 1$, it follows that

$$\sum_j h(i, j)^2 \le 1 + \varepsilon^2. \tag{13}$$

Again by Kirchhoff's law, for every $i \in U \cup \partial U$,

$$\sum_j h(j, i) \le 1 \tag{14}$$

since the flow value $h(u_0, i) \le 1$. It follows that

$$\sum_j h(j, i)^2 \le 1. \tag{15}$$

Here is thus the breakdown of the denominator of eqn. (11) to increments:

$$\varepsilon \sum_i x_i^2 \le \sum_i x_i^2 (\sum_j (h(i, j) - h(j, i))) = \sum_{i,j} h(i, j)(x_i^2 - x_j^2), \tag{16}$$

where the summation extends over all adjacent pairs $i, j \in U \cup \partial U$. (We use the fact that $x_i = 0$ for $i \in \partial U$.)

We use inequalities (13) and (15) to estimate a related sum:

$$\begin{aligned} &\textstyle\sum_{i,j} h(i,j)^2(x_i+x_j)^2 \le 2\sum h(i,j)^2(x_i^2+x_j^2) \\ = \quad & 2\textstyle\sum_i x_i^2(\sum_j (h(i,j)^2+h(j,i)^2)) \le 2(2+\varepsilon^2)(\sum x_i^2). \end{aligned} \tag{17}$$

Combining inequalities (16) and (17), we obtain the following inequalities, where all summations extend over pairs $(i,j), i,j \in U \cup \partial U$. We shall apply the Cauchy-Schwarz inequality.

$$\begin{aligned} d-\lambda \quad &\ge \quad \frac{\sum_{i,j}(x_i-x_j)^2}{\sum x_i^2} \cdot \frac{\sum_{i,j} h(i,j)^2(x_i+x_j)^2}{\sum_{i,j} h(i,j)^2(x_i+x_j)^2} \\ &\ge \quad \frac{(\sum_{i,j} h(i,j)|x_i^2-x_j^2|)^2}{2(2+\varepsilon^2)(\sum_i x_i^2)^2} \ge \varepsilon^2/(4+2\varepsilon^2). \end{aligned} \tag{18}$$

This completes the proof of the Lemma. ♠

We continue to use the above notation. We estimate the probability of a random walk exiting U in terms of the largest eigenvalue.

Proposition 4.2. *Let v_0 be a vertex in U. Let us consider a random walk over X, starting from v_0. The probability that the first ℓ steps will all be within U is $\le (|U|)^{1/2}(\lambda/d)^\ell$.*

Proof. Let A denote the adjacency matrix of U. Then $(1/d)A$ describes the transition probabilities between pairs of vertices of U. (Note: this is not a stochastic matrix since it is possible to exit from the set U.) Let e_0 denote the column vector of length $|U|$ with 1 in position v_0 and 0 elsewhere. Let j be the all-ones vector. Then the probability that the random walk makes no exit from U during the first ℓ steps is $e_0^T(A/d)^\ell j$ where the superscript T refers to transpose. Now A is a symmetric matrix hence $A = C^TDC$ for some orthogonal matrix C and diagonal matrix D. All diagonal entries of D are at most λ in absolute value. Therefore

$$e_0^T A^\ell j = (Ce_0)^T D^\ell (Cj) \le \|Ce_0\| \cdot \|D\|^\ell \cdot \|Cj\| = 1 \cdot \lambda^\ell \cdot (|U|)^{1/2}. ♠ \tag{19}$$

Corollary 4.3. *Let U be an ε-expanding subset of the vertices of a regular graph X of degree d. Then the probability that during its first ℓ steps, a random walk over X starting in U does not exit U is less than*

$$|U|^{1/2}\exp(-\varepsilon^2\ell/(d(4+2\varepsilon^2))). ♠ \tag{20}$$

5 Random walks over vertex-transitive graphs

The results of the previous section guarantee that over a locally expanding graph, a random walk has a good chance of going reasonably far *within* a short time. The difficulty in using this fact is that we cannot point to a particular point in time at which the random walk is likely to be at a reasonably great distance.

In this section, we exploit the *symmetry* of our graphs to show that a random walk *stopping at a random time* has a fair chance of ending up reasonably far.

Let X be a vertex-transitive graph. Let x_0 be the start vertex (origin) and x_t the position at time t of a random walk over X. Let $\xi_t = \mathrm{dist}(x_0, x_t)$. Let $\eta_\ell = \max\{\xi_0, \xi_1, \dots, \xi_\ell\}$. Fix positive integers k and ℓ such that

$$\mathrm{Prob}(\eta_\ell \geq 4k+1) \geq 1/2, \tag{21}$$

i.e. with probability $\geq 1/2$, by time ℓ the random walk will have reached distance $\geq 4k+1$ from the origin at least once.

Lemma 5.1. *Let k and ℓ satisfy (21). Let τ be a random number selected uniformly from $\{k+1, k+2, \dots, \ell\}$. Then*

$$\mathrm{Prob}(\xi_\tau \geq k+1) \geq 1/16. \tag{22}$$

Informally, the Lemma says that a random walk of random length $\leq \ell$ has a fair chance of ending at distance $\geq k+1$.

Proof. Let B_j denote the event $(\xi_j \leq 2k)$.

Claim 1. There exists an $m \leq \ell$ such that $\mathrm{Prob}(B_m) \leq 2/3$.

We prove the Claim. Let A_j denote the event that j is the first time that $\xi_j \geq 4k+1$ happens. Then the A_j are mutually disjoint events, and $\sum_{j=1}^{\ell} \mathrm{Prob}(A_j) \geq 1/2$. By vertex-transitivity, the distribution of $\mathrm{dist}(x_j, x_{j+m})$ is the same as the distribution of ξ_m. By the triangle inequality we observe that if $\xi_j \geq 4k+1$ and $\mathrm{dist}(x_j, x_{j+m}) \leq 2k$ then $\xi_{j+m} \geq 2k+1$. Hence

$$\mathrm{Prob}(B_{j+m} | A_j) \leq 1 - \mathrm{Prob}(B_m). \tag{23}$$

For a contradiction assume now that $\mathrm{Prob}(B_m) > 2/3$ for all $m \leq \ell$. Then

$$1/3 > \mathrm{Prob}(\bar{B}_\ell) \geq \sum_{j=1}^{\ell} \mathrm{Prob}(\bar{B}_\ell | A_j)\mathrm{Prob}(A_j) > (2/3)\sum_{j=1}^{\ell} \mathrm{Prob}(A_j) \geq 1/3, \tag{24}$$

a contradiction, proving Claim 1.

Let now C_j denote the event that $(\xi_j \leq k)$. Let further $T = \{t : 0 \leq t \leq \ell; \;\; \mathrm{Prob}(C_t) > 3/4$.

Claim 2. $|T|/\ell \leq 3/4$.

Let m be the integer guaranteed to exist by Claim 1; so $\mathrm{Prob}(B_m) \leq 2/3$. Using, as before, vertex-transitivity and the triangle inequality, we observe that for $t \in T$, we have

$$\mathrm{Prob}(C_{m\pm t}|\bar{B}_m) \leq 1 - \mathrm{Prob}(C_t) < 1/4; \tag{25}$$

and therefore

$$\begin{aligned}\mathrm{Prob}(C_{m\pm t}) \leq \;\; & \mathrm{Prob}(C_{m\pm t}|\bar{B}_m) \cdot \mathrm{Prob}(\bar{B}_m) + \mathrm{Prob}(B_m) < \\ & 1 - (3/4)\mathrm{Prob}(\bar{B}_m) \leq 1 - (3/4)(1/3) = 3/4. \end{aligned} \tag{26}$$

Hence if $t \in T$ then $m \pm t \notin T$. It follows that $|T| \leq 3\ell/4$. We also note that $T \supseteq \{0, 1, 2, \ldots, k\}$.

To conclude the proof of the Lemma we infer that $\mathrm{Prob}(\tau \in T) < |T|/\ell \leq 3/4$. Therefore $\mathrm{Prob}(\bar{C_\tau}) \geq \mathrm{Prob}(\bar{C_\tau}|\tau \notin T)\mathrm{Prob}(\tau \notin T) \geq (1/4)(1/4) = 1/16$. ♠

The results of the previous two sections guarantee that inequality (21) automatically holds in vertex-transitive graphs for some reasonable value of ℓ. The results thus add up to the following.

Theorem 5.2. *Let X be a connected vertex-transitive graph of degree d on the vertex set V. Assume, for some $k \geq 0$, that $|\Gamma^{4k}(v)| \leq |V|/2$ for some (any) $v \in V$. Let τ be a random number selected uniformly from $\{k+1, k+2, \ldots, \ell\}$, where*

$$\ell \geq 513k^2 d \cdot (2\ln 2 + \ln|\Gamma^{4k}(v)|). \tag{27}$$

Then inequality (22) holds, i.e. with probability $\geq 1/16$, a random walk of length τ, starting at v, will end outside $\Gamma^k(v)$.

Remark 5.3. A sufficient condition to ensure $|\Gamma^{4k}(v)| \leq |V|/2$ is that $k < \mathrm{diam}(X)/8$.

Remark 5.4. The following trivial estimate is useful in applications of Theorem 5.2.

$$\ln|\Gamma^{4k}(v)| < \min\{4k \ln d, \ln|V| - \ln 2\}. \tag{28}$$

Proof. Let $U = \Gamma^{4k}(v)$. By Theorem 3.2, this set is ε-expanding in the sense defined in Section 4 for $\varepsilon = 1/(16k)$. Substituting a value ℓ satisfying (27) we obtain $|U|^{1/2}\exp(-\varepsilon^2\ell/(d(4+2\varepsilon^2))) < 1/2$. Hence, by Corollary 4.3, inequality (21) holds. An application of Lemma 5.1 concludes the proof. ♠

6 Further preliminaries: rapid mixing, reachability, and reducing the number of generators

In the next section we describe Phase One of the algorithm. The output will be a logarithmic number of generators such that each element of G can be represented as a product of logarithmic length of these generators. First we show how such an output can be used to obtain ε-uniform random elements for any $\varepsilon > 0$.

6.1 Rapidly mixing random walks

We prove that random walks over vertex-transitive graphs of small degree and diameter rapidly approach the uniform distribution.

Lemma 6.1. *Let X be a vertex-transitive graph of degree d and diameter Δ. Then the second eigenvalue of the adjacency matrix of X is $\lambda_2 \leq d - 1/(16.5\Delta^2)$.*

Proof. By Proposition 3.5 we know that X is a $1/(2\Delta)$-expander. By Alon's eigenvalue bound [Alo, Lemma 2.4], we obtain

$$d - \lambda_2 \geq \gamma^2/(4 + 2\gamma^2) \geq 1/(16\Delta^2 + 2) \tag{29}$$

where $\gamma \geq 1/(2\Delta)$ is the expansion rate. ♠

The following well known estimate shows how to use the eigenvalue gap to find nearly uniformly distributed vertices.

Consider the following *lazy* random walk on the graph X: we begin each step by flipping a fair coin. If it comes out heads, we don't move in this step; else we move to a neighbor, each neighbor having equal probability to be visited. If A denotes the adjacency matrix of X, then the transition matrix of the lazy random walk is $(A + dI)/(2d)$. This matrix is positive semidefinite so we won't have to worry about negative eigenvalues.

Proposition 6.2. *Let X be a regular graph of degree d and let $v_0, v_i \in V$. Let λ_2 be the second largest eigenvalue of the adjacency matrix A of X. Let $p(\ell)$ denote the probability that after ℓ steps, the lazy random walk starting at v_0, arrives at v_1. Then*

$$|p(\ell) - (1/|V|)| \leq ((d + \lambda_2)/2d)^{\ell}. \tag{30}$$

Proof. Let e_h denote the column vector of length $|V|$ with 1 in position v_h and 0 elsewhere ($h = 0, i$). Set $B = (A + dI)/(2d)$ and $\mu = (d + \lambda_2)/(2d)$.

The largest eigenvalue of B is 1, the second largest is μ, and all eigenvalues are nonnegative. Clearly,

$$p(\ell) = e_0^T B^\ell e_i, \tag{31}$$

where the superscript T stands for transpose.

Let C^T denote the matrix whose columns form an orthonormal eigenbasis of A; let $|V|^{-1/2}j$ be the first column (corresponding to eigenvalue $\lambda_1 = d$) where j denotes the all-ones vector. Now $B = C^T DC$ where D is a diagonal matrix with diagonal entries $1, \mu, \dots$. Let E denote the diagonal matrix with diagonal entries $1, 0, 0, \dots$. Clearly, $C^T EC = (1/|V|)J$, where J is the all-ones matrix. Let $D_1 = D - E$. We note that $\|D_1\| = \mu$. We infer that

$$|p(\ell) - (1/|V|)| = |(Ce_0)^T(D_1^\ell(Ce_i)| \leq \|Ce_0\| \cdot \|D_1\|^\ell \cdot \|Ce_i\| = 1 \cdot \mu^\ell \cdot 1. \spadesuit \tag{32}$$

The last two results combine to an estimate on the deviation from uniform distribution of lazy random walks on vertex-transitive graphs. Recall that a probability distribution over V is ε-uniform if every element has probability $(1/|V|)(1 \pm \varepsilon)$ to be selected.

Theorem 6.3. *Let X be a vertex-transitive graph of degree d and diameter Δ. After ℓ steps, the lazy random walk, starting at a given vertex, ends at an ε-uniformly distributed random vertex, where*

$$\varepsilon < |V| \exp(-\ell/(33\Delta^2 d)). \tag{33}$$

Proof. Using the notation of the previous proof, we have $\mu = (d + \lambda_2)/(2d) < 1 - 1/(33\Delta^2 d) < \exp(-1/(33\Delta^2 d))$ by Lemma 6.1. Now apply Proposition 6.2. $\spadesuit$

6.2 The Reachability Lemma

The following result appears in [BSz] as Theorem 3.1.

Lemma 6.4 (Reachability Lemma, [BSz]). *Given a set S of generators of the finite group G, every element of G can be reached from S by some straight line program of length $< (1 + \log |G|)^2$.*

We briefly review the proof since it provides the basic motivation of Phase One of our algorithm.

What one proves in effect is the following.

Lemma 6.5. *Given a set S of generators of the finite group G, there exists a straight line program of length $< (\log |G|)^2$ which reaches a sequence of elements $h_1, \dots, h_t$ such that* *(i)* $t \leq \log |G|$;

(ii) *every element of G can be represented as a product $x^{-1}y$, where x, y belong to the cube $C(h_1, \dots, h_t)$.*

Recall (Section 1.3) that $C(h_1, \dots, h_t)$ is defined as the set of subproducts $h_1^{e_1} \cdots h_t^{e_t}$ where $e_i \in \{0, 1\}$. Hence every element of G can be represented as a product of length $\leq 2t - 1$ of the h_i and their inverses.

This clearly implies the Reachability lemma. We prove Lemma 6.5.

Proof. For $i \geq 0$, suppose h_j has already been defined for all j, $1 \leq j \leq i$. (This is certainly true in the initial case $i = 0$.) Let $C_i = C(h_1, \dots, h_i)$ denote the cube based on the sequence $h_1, \dots, h_i$. (For $i = 0$ we set $C_0 = \{1\}$.) Let $h_{i+1} \in C_i^{-1} C_i S$ be such that

$$C_i \cap C_i h_{i+1} = \emptyset. \tag{34}$$

If no such h_{i+1} exists, declare $t = i$ and halt.

Clearly $C_{i+1} = C_i \cup C_i h_i$, hence $|C_{i+1}| = 2|C_i|$. Consequently, $t \leq \log |G|$, verifying condition (i).

Set $D = C_i^{-1} C_i$. If some $x \in DS$ is not an appropriate choice for h_{i+1} because it violates equation (34) then $x \in D$. If none of the elements of DS are appropriate then $DS \subseteq D$, therefore $G = DS^N \subseteq D$, hence $D = G$. This proves that it was correct to conclude that $i = t$: condition (ii) holds.

Finally since $h_{i+1} \in C_i^{-1} C_i S$, the "straight line cost" of adding h_{i+1} to $S \cup \{h_1, \dots, h_i\}$ is $\leq (2i - 1)$. Hence the total cost is $\leq \sum_{i=1}^{t} (2i - 1) = t^2$. ♠

6.3 Reducing the number of generators: Phase Zero

A group G of order $\leq N$ cannot have subgroup chains of length greater than $\log N$. Therefore any set of $> \log N$ generators is redundant. We may, however, not be able to recognize which generators can be omitted.

A simple Monte Carlo algorithm to reduce the number of generators of a black box group to $O(\log N)$ is described in [BCFLS]. In this section, we review the result. For the definition of "subproducts", see Section 1.3.

Lemma 6.6 [BCFLS] *Let m denote the length of the longest subgroup chain in the group G. (Note: $m \leq \log |G|$.) Let S be an ordered sequence of generators of G. Let further T be a set of $2m + t$ random subproducts of S. Then the probability that T does not generate G is less than $\exp(-t^2/(4m + 2t))$.*

It follows that for any $\varepsilon > 0$, a set of $\geq 2m + \ln(1/\varepsilon) + 2(m \ln \varepsilon)^{1/2}$ random subproducts generate G with probability $\geq 1 - \varepsilon$.

Corollary 6.7. *Let G be a group given by a set S of generators and an upper bound N for $|G|$. For any constant $c > 0$, a Monte Carlo algorithm constructs, with probability $\geq 1 - N^{-c}$, a set of $O(\log N)$ generators for G. The cost of the algorithm is $O(|S| \log N)$ group operations.*

The algorithm simply consists of taking the stated number of random subproducts.

Remark 6.8. Another Monte Carlo algorithm, also described in [BCFLS], requires $O(|S| \log |S| \log(1/\varepsilon))$ group operations to obtain $O(\log N)$ generators with probability $\geq 1 - \varepsilon$.

Since $|S|$ would normally not be greater than $(\log N)^{O(1)}$, this algorithm is considerably faster than the one described in Corollary 6.7, as long as we do not insist on the same degree of reliability (which is exponential in $\log N$ in Cor. 6.7).

We remark that these algorithms, like those in the rest of the paper, are not Las Vegas; we have no way of knowing that the algorithm succeeded in actually finding a set of generators. But it is up to us to set the reliability parameter ε; the cost will be proportional to $\log(1/\varepsilon)$.

7 The algorithm: Phase One

Now we turn to the description of the first (main) phase of the algorithm. Let G be a group of order known to be $\leq N$.

The *input* of Phase One is the integer N and a set S of generators of G.

If successful, the *output* of Phase One will be another set S' of generators such that

(i) $|S'| = |S| + c_1 \log N$;

(ii) every element of G can be represented as a product of length $\leq c_2 \log N$ of elements of S' and their inverses.

Our Monte Carlo algorithm is not Las Vegas; we have no way of checking whether or not Phase One was successful, i.e. whether or not objective (ii) was met. However, the probability that Phase One fails is exponentially small as a function of $\log N$.

For definiteness, we shall state concrete values of the constants in the algorithm. In particular, we can choose $c_1 = xxx$ and $c_2 = xxx$.

The algorithm will construct an increasing sequence $S = S_1 \subset S_2 \subset \ldots \subset S_m = S'$ of subsets of G, where $m = c_3 \log N$. The sets will have cardinality

$$|S_i| = |S| + c_4(i - 1), \tag{35}$$

where $c_4 = c_1/c_3$ (cf. (i)).

In each round, we have to augment S_i by a set R_{i+1} of c_4 elements, to obtain S_{i+1} $(1 \le i \le m-1)$. We set $R_1 = S$.

To obtain the elements of R_{i+1} for $i \ge 1$, we consider random walks on the Cayley graph $X_i = C(G, S_i)$. Each of the c_4 elements of R_{i+1} is obtained as the result of a random walk of random length, starting at the identity. The length of the random walk is a random integer between $2i+1$ and ℓ_i where

$$\ell_i = \lceil 2052 i^2 |S_i| \ln(2N) \rceil. \tag{36}$$

(Distinct random choices should be made for each new element.)

Here is a pseudo-code of the algorithm.

```
**procedure** *PHASE_ONE(N,S)*
Initialize: $S' := S$
**for** $i = 1$ **to** $c_3 \log N$ **do**
    initialize: $R = \emptyset$
    **for** $j = 1$ **to** $c_4$ **do**
        select random integer $\tau \in \{2i+1, \dots, \ell_i\}$
        make random walk of length $\tau$, starting from 1,
            in the Cayley graph $C(G, S')$
        add the element reached to $R$
    **end**
    set $S' := S' \cup R$
**end**
```

In this procedure, ℓ_i is defined by equation (36), where $|S_i| = |S| + c_4(i-1)$ (eqn. (35)).

Theorem 7.1 *For any constant $c_5 > 0$ and appropriate positive constants c_3, c_4, procedure PHASE_ONE constructs, with probability $> 1 - \exp(-c_5 \log N)$, a set S' such that the Cayley graph $C(G, S')$ has diameter $\le 16 c_3 \log N$. The cost of the algorithm is $O((\log N)^5)$ group operations and $O((\log N)^5 \log\log N)$ random bits.*

The cost estimate presumes that $|S| = O(\log N)$, an assumption justified in Section 6.3.

Proof. Let $R'_i = R_i \cup \{1\}$; $C_i = R'_1 \cdots R'_i$, $C_0 = \{1\}$. As in the proof of the Reachability Lemma, if

$$R_{i+1} \not\subseteq C_i^{-1} C_i, \tag{37}$$

then $|C_{i+1}| \ge 2|C_i|$. Our objective was to select a small number of elements to guarantee that (37) has a good chance to hold.

Let $X_i = C(G, S_i)$. Observe that

$$C_i^{-1} C_i \subseteq \Gamma_i^{2i} \tag{38}$$

where Γ_i^t denotes the ball of radius t about the identity element in X_i. It follows by Theorem 5.2 and Remark 5.4 that each element added to R_{i+1} has probability $\geq 1/16$ to be outside Γ_i^{2i}, unless $2i \geq \mathrm{diam} C(G, S_i)/8$ (see Remark 5.3).

Consequently, as long as $16i < \mathrm{diam} C(G, S_i)$, the probability that (37) fails is $\leq (15/16)^{c_4}$. E.g. for $c_4 = 11$, this probability is $< 0.4917 < 1/2$.

Let now $c_3' = (16/15)^{c_4}$ and $c_3 > c_3'$. Then by a Chernoff estimate, a Bernoulli trial with probability of success $1/c_3'$, if repeated $c_3 \log N$ times, has more than $\log N$ successes with probability $> 1 - \exp(-c_5 \log N)$. Since the cardinality of C_i cannot double more than $\log N$ times, an easy argument shows that with probability $> 1 - \exp(-c_5 \log N)$ we must reach $\mathrm{diam} X_i \leq 16i$. ♠

We call PHASE_ONE *successful* if its output meets the diameter bound stated in Theorem 7.1. The probability that PHASE_ONE is unsuccessful is exponentially small in $\log N$.

Proposition 7.2 *After a successful completion of PHASE_ONE (at a cost of $O((\log N)^5)$ group operations), we are able to generate, for any $\varepsilon > 0$, ε-uniformly distributed random elements of G at a cost of $O((\log N)^4 + (\log N)^3 \log(1/\varepsilon))$ group operations per random element.*

Proof. According to Theorem 6.3, lazy random walks of length $\ell \geq 33\Delta^2 d(\ln N + \ln(1/\varepsilon))$ will produce ε-uniformly distributed random elements of G, where Δ is the diameter and d the degree of the Cayley graph $C(G, S')$ obtained in Phase One. Both quantities are $O(\log N)$. ♠

8 The Erdős - Rényi generators

The aim of the second phase is to set up, at a cost not greater than that of the first phase, a generator producing ε-uniform random elements of G at greatly reduced cost per random element compared to the cost stated in Proposition 7.2.

8.1 The Erdős - Rényi theorem

Erdős and Rényi [ER, Theorem 1] proved that for

$$k \geq 2 \log |G| + 2 \log(1/\varepsilon) + \log(1/\delta), \tag{39}$$

a sequence of k random elements of G will be a sequence of ε-uniform Erdős - Rényi generators with probability $\geq 1-\delta$.

The slight problem with applying this result is that the random elements obtained after Phase One (Proposition 7.2) are not truly uniformly distributed.

However, an easy modification of the Erdős-Rényi argument yields the following result.

Theorem 8.1 *Let* $\gamma > 1$,

$$k \geq 2\log|G| + 2\log(1/\varepsilon) + \log(1/\delta) + \log(\gamma), \tag{40}$$

and $0 < \omega \leq (\gamma-1)|G|/2^k$.

Then a sequence of k random elements of G from an ω-uniform distribution will form a sequence of ε-uniform Erdős - Rényi generators with probability $\geq 1-\delta$.

The proof is identical with that of the original result of Erdős and Rényi except that in their Lemma, the upper bound on the quantity D_k^2 should be $D_k^2 < \gamma \cdot 2^k$ (rather than $2^k(1-1/|G|)$). (This quantity is defined as the variance of the number of representations of a truly uniform random element of G as a subproduct of the given k-tuple.)

8.2 The algorithm: Phase Two

Phase Two of the algorithm assumes successful completion of Phase One.

The *input* of Phase Two is a pair of positive parameters ε, δ, together with the output of a (successful) Phase One, i.e. a set S' of $\leq c_6 \log N$ generators of G such that every element of G can be represented as a product of length $\leq c_2 \log N$ of elements of S' and their inverses. (We assume Phase Zero (Section 6.3) was successfully completed, if necessary, before Phase One.)

The *output* of Phase Two is a sequence of k elements of G which, with probability $\geq 1-\delta$, form a sequence of ε-uniform Erdős-Rényi generators. Here

$$k = \lceil 2\log N + 2\log(1/\varepsilon) + \log(1/\delta) + 1 \rceil \tag{41}$$

procedure *PHASE_TWO*$(\varepsilon, \delta, S')$.

Set $\omega = \varepsilon^2\delta/N^2$. Generate k random elements of G which are ω-uniformly distributed, according to Proposition 7.2, where k is defined by eqn. (41). End.

Theorem 8.2. *With probability $\geq 1-\delta$, procedure PHASE_TWO produces a sequence of ε-uniform Erdős-Rényi generators. The cost of the procedure is* $O((\log N)^5 + (\log N)^3(\log(1/\varepsilon) + (\log(1/\delta))^2))$ *group operations.*

Proof. We set $\gamma = 2$ and apply Theorem 8.1. Our choice of k satisfies the condition in Theorem 8.1 since $|G| \le N$. Our choice of ω satisfies the second condition in Theorem 8.1, namely $\omega \le (\gamma - 1)|G|/2^k$ since now

$$\omega = \varepsilon^2\delta/N^2 \le (\varepsilon^2\delta/N^2)(|G|/2) = (\gamma - 1)|G|/2^k.$$

By Theorem 8.1, all that remains to be justified is the claimed cost.

By Proposition 7.2, the cost is $O(k((\log N)^4 + (\log N)^3 \log(1/\omega)))$. Since $\log(1/\omega) = O(\log N + \log(1/\varepsilon) + \log(1/\delta))$, the stated bound is immediate. ♠

Remark 8.3. Presumably in most applications, ε and δ need not be smaller than $(1/N)^{O(1)}$. In these cases, we conclude that the combined total cost of Phases One and Two is $O((\log N)^5)$ group operations. After Phase Two, the cost of ε-uniform random elements is $O(\log N)$ group operations per random element. This justifies Theorem 1.1.

Remark 8.4. The length of the input strings (generators of the group G) is $n \ge \log N$, so our algorithms are polynomial time.

The repeated doubling process seems inherently sequential and there does not seem any way to make this algorithm RNC. But from a practical point of view, *massive parallelization* is possible. Indeed, every round of Phase One can be performed in RNC optimally, i.e. with only a logarithmic loss in time × number of processors (because a random walk of length r in a group can be performed in $\log r$ rounds, using $r/2$ parallel processors, each capable of performing a single group operation per round).

Once we have the Erdős-Rényi generators, each random group element can be obtained in $O(\log n)$ rounds using $O(n)$ processors.

9 Impossibility of approximating the order

Below, when we talk about black box groups, we have an infinite sequence of black box groups in mind, one for every n.

The elements of the n^{th} black box group are encoded as binary strings of length n; so $N = 2^n$ is an upper bound on the order of the group.

Let p be a fairly large prime, not bounded by any polynomial of n, and let $m \ge 2$ be such that $p^m < 2^n$. (So m can still be quite large.) Let G be a cyclic group of order p and H an elementary abelian group of order p^m, i.e. the direct sum of m copies of G. Assume H is given by a basis T, i.e. a set of m generators. On the other hand, assume G is given by a list S of m randomly selected elements (which of course form a redundant set of generators of G). The encoding of each group is done by random injections $G, H \to \{0,1\}^n$.

We claim that no polynomial time Monte Carlo algorithm has a chance of distinguishing the two groups.

Indeed, let $\varphi : H \to G$ be the homomorphism obtained by extending the bijection $T \to S$ of the (ordered) lists of generators.

Let us follow the course of a Monte Carlo algorithm, applied to (H, T). The algorithm computes a sequence of group elements $h_1, \ldots, h_t$. Let $g_i = \varphi(h_i)$.

Claim. The probability that there exist $i, j \leq t$ such that $h_i \neq h_j$ but $g_i = g_j$ is less than $t^2/(2p)$.

Proof. The probability here is understood to refer to a fixed set T and a random choice of S. The probability that any particular element $h_i^{-1}h_j \neq 1$ belongs to the kernel of φ is $(p^{m-1} - 1)/(p^m - 1) < 1/p$; thus the probability that this happens to at least one element of this type is less than $\binom{t}{2}/p$. ♠

In the cases when this does not happen, there is a measure-preserving transformation between the runs of the algorithm on (H, T) (with random labeling of the elements of H) and the runs of the algorithm on G (with random list of m generators and random labeling of the elements) which preserves all the codewords (code(h_i) =code(g_i) for every $i \leq t$).

Let us now consider a black box group defined as follows: we flip a coin to decide whether the group will be (G, S) or (H, T); and perform the randomization in encoding as well as the randomization of S in case of (G, S).

It follows that a (fixed) Monte Carlo algorithm running in time t is expected to have no more than t^2/p advantage at guessing the order of the group. (The actual advantage is a random variable, depending on the random choices made in the previous paragraph. We consider the advantage to be a nonnegative value; i.e. being able to guess wrong 60% of the time also counts as 10% advantage. Clearly, this is as unlikely as being able to guess right 60% of the time.)

Let us now consider the following *group oracle.* The oracle consists of a black box group for every n; the elements of the n^{th} group are encoded by strings of length n. Each of the groups is selected at random as just described, with $p = p_n$ between $2^{\sqrt{n}}$ and $2^{1+\sqrt{n}}$ and m approximately $\sqrt{n}$.

Then, for any randomized oracle Turing machine M running in time $t(n)$ on inputs of length n, the following event is true with probability 1: For all but finitely many values of n, the machine M has no more than $2n^2 t(n)^2 2^{-\sqrt{n}}$ advantage at guessing whether the order of the n^{th} group is less than $2^{1+\sqrt{n}}$ or more than $2^{n/2}$.

(The role of the n^2 factor inserted is to make the probability for the n^{th} group less than $1/n^2$; then by the Borel-Cantelli lemma, almost surely this happens a finite number of times only.)

Hence this is true for all Turing machines simultaneously with probability 1. It follows that there exists a group oracle for which the above statement is still true for every Turing machine M.

We summarize the result.

Proposition 9.1. *There exists a group oracle, i.e. a sequence of black box groups B_n with the following properties:*

(i) for every n, the elements of B_n are encoded as binary strings of length n;

(ii) B_n is an elementary abelian group of order either $< 2^{1+\sqrt{n}}$ or more than $2^{n/2}$;

(iii) for every $t(n)$-time-bounded randomized oracle Turing machine M and for every $n > n_0(M)$, the machnie M has no more than $2n^2t(n)^2 2^{-\sqrt{n}}$ advantage at guessing whether $|B_n| < 2^{1+\sqrt{n}}$ or $|B_n| > 2^{n/2}$.

In particular, no polynomial time Monte Carlo algorithm can guess the logarithm of the order of a black box group within a factor of $\sqrt{n}$.

10 Some applications

10.1 Permutation groups with a small base

A *base* of a permutation group $G \leq \mathrm{Sym}(\Omega)$ is a set $\Delta \subseteq \Omega$ such that no element of $G \setminus \{1\}$ fixes Δ pointwise. Δ is a *small base* if $|\Delta|$ is bounded by polylog(n), where $n = |\Omega|$. Such groups are particularly significant in computational group theory. Clearly, $2^{|\Delta|} \leq |G| < n^{|\Delta|}$ (the first inequality assumes sequential irredundance of Δ in some ordering), so G has a small base precisely if $\log |G|$ is bounded by polylog(n).

A combination of the nearly uniform generator of this paper and the efficient data structure of [CFS] yield a *nearly linear time* algorithm for basic manipulation (membership, order, etc.) for groups with a small base. More specifically we obtain:

If N is an a priori bound on the order of $G \leq S_n$ then basic group manipulation can be solved in Monte Carlo time $O(n(\log N)^c)$ for an absolute constant c.

The constant c does not seem small enough to make this algorithm competitive in practice. A substantially better constant has been obtained in [BCFS]; work on that paper has been a source of motivation for this work. In addition, an application of the Local Expansion Lemma to the product of partial transversals in [BCFS] allows to avoid the need for an *a priori* bound on $|G|$.

10.2 Sylow subgroups of small index

Let G be a black box group, N a known upper bound on $|G|$, and r a known upper bound on the index of a Sylow p-subgroup. Then, *a Sylow p-subgroup can be constructed in Monte Carlo time $r(\log N)^c$.*

The algorithm uses the Monte Carlo polynomial time recognition algorithm of nilpotence [BCFLS]. A group is a p-group if and only if it is nilpotent and all generators have order a power of p. We build a p-subgroup P, starting from $P = \{1\}$, by adding a random element $g \in G$ whenever $\langle P, g \rangle$ is a p-group. We stop after $O(\log N)$ rounds.

10.3 Interactive proofs

This author introduced his version of interactive proofs [Ba2] in order to put the problems of matrix group order and nonmembership into suitably low complexity classes [Ba3]. The Local Expansion Lemma was a key tool. The result of the present paper allows a very simple "nonmembership" protocol for black box groups, along the lines of the quadratic nonresiduosity protocol of [GMR]: to verify that $g \notin G$, the verifier privately generates random elements $h_i \in G$, and for each i flips a coin to decide whether to show h_i or $h_i g$ to the prover. Subsequently, the prover has to guess for each i the outcome of the coin flip. If indeed $g \notin G$, he can answer correctly all the time; otherwise he is unlikely to answer correctly more than 51% of the time. – I do not know such a simple protocol to verify the order of G [Ba3].

Acknowledgment

The author is indebted to Gene Cooperman, Larry Finkelstein, Gene Luks, and Ákos Seress for sharing their insights in permutation group algorithms and to Persi Diaconis and Pál Takácsi-Nagy for illuminating discussions of Markov chains on groups.

References

[Ald] D. Aldous: On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing, *Probability in Engineering and Informational Sciences* **1** (1987), 33-46.

[Alo] N. Alon: Eigenvalues and expanders, *Combinatorica* **6** (1986), 83-96.

[Ba1] L. Babai: Monte Carlo algorithms in graph isomorphism testing, Université de Montréal Tech. Rep. DMS 79-10, 1979.

[Ba2] L. Babai: Trading group theory for randomness, *in: Proc. 17th ACM STOC*, Providence RI, 1985, pp. 421-429.

[Ba3] L. Babai: Bounded-round interactive proofs in finite groups, *SIAM J. Discr. Math.*, to appear

[Ba4] L. Babai: Computational complexity in finite groups, *Proc. International Congress of Mathematicians*, Kyoto 1990, Springer, to appear

[BCFS] L. Babai, G. Cooperman, L. Finkelstein, Á. Seress: Permutation groups with small base in almost linear time, in preparation

[BCFLS] L. Babai, G. Cooperman, L. Finkelstein, E. M. Luks, Á. Seress: Fast Monte Carlo algorithms for permutation groups, in preparation

[BE] L. Babai, P. Erdős: Representation of group elements as short products, *in: Theory and Practice of Combinatorics* (A. Rosa et al., eds.), *Ann. Discr. Math.* **12** (1982), 27-30.

[BLS] L. Babai, E. M. Luks, Á. Seress, Permutation groups in *NC*, *in: Proc. 19th ACM STOC*, 1987, pp. 409-420.

[BSz] L. Babai, E. Szemerédi: On the complexity of matrix group problems I, *in: Proc. 25th IEEE FOCS*, Palm Beach FL, 1984, pp. 229-240.

[Ca] R. Carter: Simple groups of Lie type, *Wiley* 1972, 1989.

[Ch] J. Cheeger: A lower bound for the smallest eigenvalue of the Laplacian, *in: Problems in Analysis* (R. C. Gunning, ed.), Princeton Univ. Press 1970, pp. 195-199.

[CFS] G. Cooperman, L. Finkelstein, N. Sarawagi: A random base change algorithm for permutation groups, *in: Proc. Int. Symp. Symbolic and Algebraic Comp.*, 1990.

[Di] P. Diaconis: *Group Representations in Probability and Statistics*, Inst. Math. Stat. Hayward CA 1988.

[ER] P. Erdős, A. Rényi: Probabilistic methods in group theory, *J. d'Analyse Math.* **14** (1965), 127-138.

[FHL] M. L. Furst, J. Hopcroft, E. M. Luks: Polynomial-time algorithms for permutation groups, *in: 21st IEEE FOCS*, 1980, pp. 36-41.

[GMR] S. Goldwasser, S. Micali, C. Rackoff: The knowledge complexity of interactive proofs, *SIAM Journal on Computing*, **18** (1989), 186-208. (Preliminary version in *Proc. 18th STOC*, 1985, pp. 291-304.)

[Je] M. R. Jerrum: A compact representation for permutation groups, *J. Algorithms* **7** (1986), 60-78.

[JVV] M. R. Jerrum, L. G. Valiant, V. V. Vazirani: Random generation of combinatorial structures from a uniform distribution, *Theoret. Computer Science* **43** (1986), 169-188.

[KL] W. M. Kantor, E. M. Luks: Computing in quotient groups, *in: Proc. 22nd ACM STOC*, Baltimore, 1990, pp. 524-534.

[Kn] D. E. Knuth: Efficient representation of perm groups, *Combinatorica*, to appear

[MW] B. Mohar, W. Woess: A survey on spectra of infinite graphs, *Bull. London Math. Soc.* **21** (1989), 209-234.

[Lu] E. M. Luks: Examples that beat GAP's Monte Carlo routines (personal communication)

[NP] P. M. Neumann, Cheryl E. Praeger: A recognition algorithm for the special linear groups, manuscript, 1990.

[Ré] A. Rényi: *Selected Papers* (P. Turán, ed.), Akadémiai Kiadó, Budapest, 1976.

[Sim] C. C. Sims, Some group theoretic algorithms, *in: Lecture Notes in Math.* **697** (1978), pp. 108-124.

[SJ] A. Sinclair, M. R. Jerrum: Approximate counting, uniform generation and rapidly mixing Markov chains, *Information and Computation* **82** (1989), 93-133.

[Sze] M. Szegedy, Notes on the expansion property of symmetric graphs, in preparation

[Va] N. Th. Varopoulos: Isoperimetric inequalities and Markov chains, *J. Funct. Anal.* **63** (1985), 215-239.