Contemporary Mathematics

Variants of Product Replacement

C. R. Leedham-Green and Scott H. Murray

Abstract. We present new variations on the product replacement method for generating random elements in a black box group, together with some basic analysis and conjectures.

1. Introduction

The product replacement algorithm is widely used to construct random elements in finite groups. In particular, the computational algebra systems GAP and Magma both used this method until recently (GAP4 now uses the variant rattle described below). Given a generating set X for a black box group G (ie. a finite group with a bound on the encoding size of a element), we wish to construct random elements of G, starting with X and using only multiplication and division. The original product replacement algorithm does not produce independent uniformly distributed elements [**CLGM$^+$95, BP99**], though it is hard to find practical examples where this failure can be detected. We are primarily interested in the practical use of these methods in computational group theory—for an excellent introduction to the theoretical considerations see [**Pak01**].

The basic idea behind product replacement is to construct an array of generators for G, then at each stage multiply a randomly chosen entry in the array by another randomly chosen entry. In particular, this ensures that the random elements grow exponentially as words in the original generating set. Product replacement has two defects. Firstly, the limiting distribution of the elements returned is not uniform (ie. some elements are more likely to be returned than others)—we call this the *global problem*. The second weakness is that the elements returned are not independent—we call this the *local problem*.

In this paper we describe variants of product replacement which solve the global problem. This involves an overhead; in the simplest version, described in Section 4, two multiplications (or divisions) are required to produce each random element rather than one. In Section 5, we produce a variant that should have a faster mixing time than rattle when X is large. In Section 6, we describe a variant of product replacement to generate random elements of a normal subgroup in a black box group. This variant was used in [**LGO97**]. A slight modification of this

2000 *Mathematics Subject Classification.* Primary 20-04; Secondary 20D60, 60J05.
Key words and phrases. finite groups, black box groups, random elements, Markov chains.

gives a method for generating random elements in finite G-modules, as described in Section 7. We consider the problem of finding random elements of the derived subgroup or other verbal subgroups in Section 8.

It is intuitively clear that rattle has a mixing time that is no worse than the original product replacement algorithm, but we have been unable to prove any results in this direction. Instead we discuss why these algorithms behave so well in practice, and offer some conjectures that would explain this behaviour in Section 9.

2. Markov chains

The algorithms presented in this paper can be described as Markov chains. In this section we give some standard terminology and results from the theory of Markov chains. We consider only finite, homogeneous, discrete time chains. For more details see, for example, [**GS97**].

A *Markov chain* consists of a sequence $S(0), S(1), S(2), \ldots$ of random variables which take values in a finite *state space* Σ. These variables have the Markov property that the probability of $S(t+1)$ having a given state depends on the state of $S(t)$ but not on the state of $S(u)$ for $u < t$. We only consider homogeneous Markov chains, in which these probabilities are also independent of the time t. The probability that $S(t+1) = \tau$ given that $S(t) = \sigma$ is called a *transition probability* and is denoted by $p_{\sigma\tau}$. We frequently arrange these probabilities in a *transition matrix*

$$P = (p_{\sigma\tau})_{\sigma,\tau\in\Sigma}.$$

The probability that $S(t+n) = \tau$ given that $S(t) = \sigma$ is denoted by $p^{(n)}_{\sigma\tau}$, and it is easily seen that the matrix of these values is just P^n.

Given an initial state $\sigma(0)$, the probability that $S(t) = \sigma$ is denoted by $\pi_\sigma(t)$. The vector $\pi(t) = (\pi_\sigma(t))_{\sigma\in\Sigma}$ is called the *distribution* of the chain at time t. Clearly $\pi(t+1) = \pi(t)P$. We say there is a *path* from the state σ to the state τ if there is a chain of states $\sigma = \tau(0), \tau(1), \ldots, \tau(n) = \tau$ with $p_{\tau(k)\tau(k+1)} > 0$ for all k; we write $\sigma \rightsquigarrow \tau$ in this case.

A finite homogeneous Markov chain is *ergodic* if it satisfies the following properties

- *Irreducibility*: for every pair of states σ and τ, there is a path $\sigma \rightsquigarrow \tau$.
- *Aperiodicity*: for every state σ, the lowest common denominator of the lengths of the paths $\sigma \rightsquigarrow \sigma$ is 1. In other words, starting at σ, there is no $m > 1$ such that the chain can only return to σ at times which are multiples of m.

If a chain is ergodic then it has a unique *long-term distribution*

$$\pi = \lim_{t\to\infty} \pi(t) = \lim_{t\to\infty} \pi(0)P^n$$

which is independent of the initial state $\sigma(0)$. In fact, π is the unique vector with $\pi P = \pi$. If the column sums $\sum_\tau p_{\sigma\tau}$ are all equal to 1, then the *uniform distribution*, given by $\pi_\sigma = 1/|\Sigma|$, for all $\sigma \in \Sigma$ satisfies $\pi P = \pi$ and the long-term distribution is uniform on the state space.

Finally we mention that a chain is called *reversible* if $\tau \rightsquigarrow \sigma$ whenever $\sigma \rightsquigarrow \tau$. If a chain is reversible, then we can make it irreducible simply by restricting the state space to the *component* of $\sigma(0)$, i.e. the set Σ_0 of states which are connected to $\sigma(0)$ by some path.

3. Product replacement

In this section we review the Markov chain of the product replacement algorithm described in [**CLGM$^+$95**]. This is the basis of our analysis for the new algorithms. Let $X = \{x_1, \ldots, x_s\}$ be a generating set for the finite group G. Let $n > 1$ be a fixed integer which is at least as large as s. The state space Σ consists of all sequences $\sigma = (\sigma_1, \ldots, \sigma_n)$ of group elements such that $\langle \sigma_1, \ldots, \sigma_n \rangle = G$. We call these the *generating arrays* of length n. The initial state contains the elements of X, filled out if necessary with the identity, i.e. $\sigma(0) = (x_1, \ldots, x_s, 1, \ldots, 1)$.

Given distinct integers i and j between 1 and n, we define an operator T_{ij} which takes σ to the same array except that σ_i is replaced by $\sigma_i\sigma_j$. Clearly $T_{ij}\sigma$ is a generating array whenever σ is. It has an inverse ${T_{ij}}^{-1}$ which replaces σ_i by $\sigma_i\sigma_j^{-1}$. The transitions in our Markov chain are as follows: for each of the pairs (i, j) with $i \neq j$, we apply T_{ij} with equal probability. Hence the transition probabilities are

$$p_{\sigma\tau} = \frac{\#\{(i,j) : T_{ij}\sigma = \tau\}}{n(n-1)},$$

and the column sums are

$$\begin{aligned}\sum_{\tau\in\Sigma} p_{\sigma\tau} &= \sum_{\tau\in\Sigma} \frac{\#\{(i,j) : T_{ij}\sigma = \tau\}}{n(n-1)} \\ &= \frac{\#\{(i,j) : T_{ij}\sigma \in \Sigma\}}{n(n-1)} = 1\end{aligned}$$

The *product replacement* method works as follows: We start with the initial state $\sigma(0)$ and do some preprocessing, which basically means ignoring a certain number of steps in the chain. Then, for each random element, we take one step in the chain and return the most recently changed entry in the generating array.

This chain is reversible since ${T_{ij}}^{-1} = {T_{ij}}^{a-1}$, where a is the order of σ_j. Hence it is irreducible on the set Σ_0 of states connected to $\sigma(0)$. Let m be the smallest size of a generating set for G and let M be the largest size of a minimal generating set for G. In [**CLGM$^+$95**], it was shown that $\Sigma_0 = \Sigma$ whenever $n \geq m + M$. This bound has been improved in certain cases by [**DSC98**], [**BP99**].

We now show that the chain is aperiodic provided $n \geq s + 1$. Let σ be any state in Σ_0. Then there is a path $\sigma \rightsquigarrow \sigma(0)$ of length ℓ, say. By reversibility we get a loop $\sigma \rightsquigarrow \sigma(0) \rightsquigarrow \sigma$ of length 2ℓ. But $\sigma(0)_n = 1$, so $T_{1n}\sigma(0) = \sigma(0)$ and there is also a loop $\sigma \rightsquigarrow \sigma(0) \overset{T_{1n}}{\rightsquigarrow} \sigma(0) \rightsquigarrow \sigma$ of length $2\ell + 1$. Since $\gcd(2\ell, 2\ell + 1) = 1$, we are done.

Hence this Markov chain converges to the uniform distribution on the connected component Σ_0 containing our initial state. However, we are primarily interested in the distribution of a random element returned by the product replacement algorithm, which is σ_i for the appropriate value of i. Hence, in the long-term distribution, an element $g \in G$ is returned with probability

$$\frac{\#\{(\sigma, i) : \sigma \in \Sigma_0 \text{ and } \sigma_i = g\}}{n|\Sigma_0|}.$$

In [**CLGM$^+$95**], it was shown that when $n \geq m + M$ this becomes

$$\frac{\#\{\sigma \in \Sigma : \sigma_1 = g\}}{|\Sigma|}.$$

This distribution can often be shown to be close to uniform—this is proved for large simple groups in [**LS95**]. However, [**Pak99**] has shown that for certain groups it is very far from uniform. In addition, it is generally difficult to compute m and M, even for well known groups.

Note that, in many published versions of this algorithm, one allows multiplication by σ_j^{-1} as well as σ_j, or premultiplication as well as postmultiplication. This does not effect the long term distribution when $n \geq m + M$, but its effect on the rate of convergence is unknown.

4. Rattle

In this section we describe *rattle*, the most important variant of product replacement. This new method solves the global problem, since the long term distribution of the elements returned is uniform. Experimental evidence seems to suggest that it converges at least as quickly as ordinary product replacement.

We use the same notation as in the previous section. Let the state space Σ' be the set of all sequences $\sigma = (\sigma_0 \,|\, \sigma_1, \ldots, \sigma_n)$ of group elements such that $\langle \sigma_1, \ldots, \sigma_n \rangle = G$. There is no restriction on σ_0. The initial state is as in the previous section with $\sigma_0 = 1$. We call the subsequence $(\sigma_1, \ldots . \sigma_n)$ the *generating array*, and σ_0 the *accumulator*.

Given integers $1 \leq i, j, k \leq n$ with $i \neq j$, we define the operator T'_{ijk} which replaces σ_i by $\sigma_i \sigma_j$ and then replaces σ_0 by $\sigma_0 \sigma_k$. At each step in the Markov chain, one of these operators is applied with equal probability. We return the accumulator as our random element.

Since T'_{ijk} has the same effect on the generating array as T_{ij} from the previous section, all the results of that section apply to the generating array. Let Σ'_0 be the component of the initial state, so clearly the limiting distribution is uniform on Σ'_0. In order to show that the random elements returned have a uniform limiting distribution, it suffices to prove that if $(g \,|\, g_1, \ldots, g_n)$ is in Σ'_0 then so is $(h \,|\, g_1, \ldots, g_n)$ for every h in G. Now $(g \,|\, g_1, \ldots, g_n) \in \Sigma'_0$ means precisely that

$$\sigma_0 = (1 \,|\, x_1, \ldots, x_s, 1, \ldots, 1) \rightsquigarrow (g \,|\, g_1, \ldots, g_n).$$

It is easily seen that the same series of operators gives us

$$(hg^{-1} \,|\, x_1, \ldots, x_s, 1, \ldots, 1) \rightsquigarrow (h \,|\, g_1, \ldots, g_n).$$

And finally σ_0 is connected to $(hg^{-1} \,|\, x_1, \ldots, x_s, 1, \ldots, 1)$ by using operators of the form T_{inj}, which preserve everything to the left of the bar. Note that we do not require that Σ' be connected for this argument, so we no longer need the assumption that $n \geq m + M$.

5. Accelerators

Suppose we are doing product replacement with generating arrays of size n. Igor Pak (personal communication) has shown that the rate of convergence is polynomial in $\log |G|$ and n. In this section we describe a modification of the basic product replacement algorithm which may improve the rate of convergence when n is large.

We take the same state space, and the same initial state, as in product replacement. The operations S_{ij} for $i, j > 1$ are defined by $S_{ij} = T_{i1} T_{1j}$, where T is as defined for product replacement. Note that we operate on the 1st entry of the array at every step. This entry is called the *accelerator*. In order to avoid the

global problem, we should also have an accumulator. In either case it is easy to see that this defines a reversible ergodic Markov chain with column sums equal to 1.

A crude analysis suggests that if we carry out product replacement in a group that is freely generated by the initial entries of Σ then the average length of the words in the array grows exponentially, the basis of the exponent being $1 + 1/n$. If we suppose that mixing time in a finite group is approximately proportional to the time taken for the average length of the elements as words in X to reach a certain size (depending on G), then this gives a convergence rate that is $O(n)$ for a fixed G. This bound could become inconveniently large in practice. A similar analysis suggests that the use of an accelerator reduces the mixing time to $O(\sqrt{n})$. This would be useful in the case where we are unable to find a small generating set for our group, a situation that is particularly common for abelian and nilpotent groups.

6. Normal Subgroups

We now show how to adapt rattle to find random elements of normal subgroups. That is to say, the input now consists of a generating set X for the group G, and a subset Y of G that generates N as a normal subgroup of G. The ideas in this section were used in [**LGO97**]. The state space is now the set of arrays $(\sigma_0 \,|\, \sigma_1, \ldots, \sigma_m \,|\, \sigma'_1, \ldots, \sigma'_n)$, where $\sigma_i \in N$ and $\sigma'_a \in G$ for all $0 \leq i \leq m, 1 \leq a \leq n$. Initially $\sigma_0 = 1$, the N-generating array $(\sigma_1, \ldots, \sigma_m)$ consists of the elements of Y followed by a sequence of 1's, and the G-generating array $(\sigma'_1, \ldots, \sigma'_n)$ consists of the elements of X followed by a sequence of 1's. The basic operation T_{ijkabc}, where $1 \leq i, j, k \leq m$ and $1 \leq a, b, c \leq n$ and $i \neq j$ and $a \neq b$, is defined by

$$\sigma'_i := \sigma'_i \sigma'_j, \quad \sigma_a := \sigma_a \sigma_b^{\sigma'_k}, \quad \sigma_0 := \sigma_0 \sigma_c.$$

The arguments of Sections 3 and 4 can be modified to show that the chain is ergodic on the connected component of the initial state and also reversible. So to prove that the limiting distribution of σ_0 is uniform on N it suffices to show that, for every $u \in N$, a sequence of operations can be found that starts with the initial state, and replaces $\sigma_0 = 1$ by u while leaving the N- and G-generating arrays unchanged. In ordert to acheive this we assume that $m \geq \#Y + 2$ and $n \geq \#X + 1$, so initially $\sigma_{m-1} = \sigma_m = 1$ and $\sigma'_n = 1$. Now let $u = u_1^{v_1} \ldots u_k^{v_k}$, where each $u_i \in Y$ and each $v_i \in X$. We will only consider moves T_{ijkabc} in which $j = n$, so that the G-generating array is not changed. Now a sequence of operations with $a = m - 1$ and $c = m$ can be found that replaces σ_m by u, but changes nothing else. Taking $b = m$ and $c = m - 1$ then replaces σ_0 by u and changes nothing else. Now chosing $a = m - 1$ and $c = m$, again a sequence of operations can be found that returns σ'_{m-1} to its original value of 1 while changing nothing else.

7. Group modules

A trivial modification of the algorithm of Section 6 returns random elements of a finite dimensional FG-module V, where F is a finite field. Here V plays the role of N in the previous section, so the basic move now becomes $T_{ijkabc\gamma}$, defined by

$$\sigma'_i := \sigma'_i \sigma'_j, \quad \sigma_a := \sigma_a + \gamma \sigma_b \sigma'_k, \quad \sigma_0 := \sigma_0 + \sigma_c,$$

where γ is a random element of F.

8. Derived and other verbal subgroups

If G is given as $\langle X \rangle$, for a small set X, then there is no problem constructing random elements of the derived group G' of G by applying the algorithm of Section 6, since G' is generated as a normal subgroup by the commutators of pairs of elements of X. However, if X is large, this might be inefficient. Thus one might get much faster convergence by constructing the elements $g_n = [a_1, a_2][a_3, a_4] \dots [a_{2n-1}, a_{2n}]$, where the a_i are successive elements returned bu product replacement, or by rattle. Of course there are countless variations that might be tried. For example, one might get faster convergence by defining $g_n = [a_1, a_2]^{a_3} \dots [a_{3n-2}, a_{3n-1}]^{a_{3n}}$. Similar issues arise more acutely with the verbal subgroups arising from more complex laws; a search for random elements of verbal subgroups in a practical setting is discussed in [**LGO97**].

9. Conjectures

The conjectures in this section are motivated by a desire to understand the good behaviour of rattle.

Rattle can be regarded as defining a random walk on the set $\overline{\Sigma}$ of $(n+1)$-tuples in G, the first component of which is the accumulator. We take G and n to be given.

Conjecture 1. *Let $S \subseteq G$ be any subset of G. Let*

$$c = \max(|G|/|S|, |G|/|G \setminus S|).$$

Let V be the set of vertices in $\overline{\Sigma}$ that lie within a distance of $10 + \log_n(c)$ of v. Then for most v in $\overline{\Sigma}$ the proportion p of vertices in V whose accumulators lie in S satisfies

$$\frac{1}{2} - \frac{|S|}{2|G|} > p > \frac{|S|}{2|G|}.$$

The truth of this conjecture would imply that searching for elements of S, starting from a random initial configuration, would be about as successful as one would expect from a random search with an independent uniformly distributed sample. The appearance of 10 is simply to avoid edge effects when c is small. The contribution of $\log_n(c)$ is to ensure that one would expect, with a uniform distribution, that there should be a reasonably large number of elements of V whose accumulators lie in S, and similarly for $G \setminus S$.

This conjecture on its own is not sufficient to justify the use of rattle, as we do not start from a random configuration. This is not a purely theoretical observation: the need to carry out a number of preprocessing steps is clearly needed in many practical situations.

Let us call a vertex v that satisfies conjecture 1 *virtuous*; a property that depends on S. To justify rattle one needs to prove a conjecture along the following lines.

Conjecture 2. *Let $d > n \log n \log |G|$. Then every vertex v in $\overline{\Sigma}$ has the property that most random walks on $\overline{\Sigma}$ of length d starting at v end in a virtuous vertex.*

This conjecture is marred by the appearance of the word 'most'. Clearly the proportion of random walks of length d starting at v and ending at a virtuous vertex

cannot tend to 0 as d tends to infinity, unless all vertices are virtuous, so it is hard to make a convincing guess at what should replace 'most'.

The factor of $n \log n$ is inserted into the bound for d to allow every member of the generating array to be changed an appropriate number of times. The factor of $\log |G|$ arises from more interesting considerations. Define the *diameter* of a finite group G to be the largest integer d such that G has a generating set X, and, for some $g \in G$, it is possible to construct g from X in d multiplications, but not in fewer. Thus if G is cyclic then the diameter of G is approximately $2 \log_2 |G|$. The factor $\log |G|$ in the above conjecture is an estimate of the diameter of G.

This brings us to the next conjecture.

CONJECTURE 3. *Every finite group G has diameter at most $2 \log_2 |G|$.*

It is easy enough to provide evidence for this conjecture. Thus an explicit recognition algorithm for a group G enables one to write an arbitrary element g of G as a straight line program on an arbitrary generating set X for G; that is to say, g will be constructed from X by a given sequence of multiplications and divisions. Allowing divisions as well as multiplications is, in the absence of better evidence than we can provide, a mere bagatelle. These straight line programs are generally expected to have length around $\log |G|$. Although such algorithms produce evidence, and may well prove specific instances of the conjecture when X and g as well as G are specified, they do not provide much help with constructing a proof, since they need product replacement or rattle to provide random elements of G. The strategy used in these algorithms consists of two steps. First one finds, by probabilistic means, a canonical generating set for G in terms of X, and then constructs g deterministically from the canonical generating set. Thus the crucial part of proving the above conjecture, at least for a given G, is to bound the probabilistic part of the algorithm. By way of an example, suppose that G is the symmetric group S_u for some u. Then one could find, by random search, an element g that is a product of disjoint cycles, one a transposition, the others of odd length. Powering g gives rise to a transposition t. An n-cycle h can also be found by a random search. Replacing g or t by suitable conjugates, and replacing g by a suitable power of itself, one soon reduces to the case where we may assume that $g = (1, 2, \ldots, u)$, and $t = (1, 2)$. Now one can deterministically construct any element of G from g and t in $O(u \log u)$ multiplications. The main remaining problem is to bound the number of multiplications needed to find g and h.

Simplifying this problem, we ask: find a good bound (in terms of u) to the number of multiplications that may be required to obtain an element of S_u of even order from an arbitrary generating set X of G.

It is possible that there is a universal bound to this number, independent of u. However, by estimating the number of conjugacy classes of generating pairs for S_u, the number of elements of S_u that can be constructed from a given generating pair in a small number of multiplications, and the proportion of elements of S_u of even order, one arrives at the following conjecture.

CONJECTURE 4. *Let $e(u)$ be the largest integer such that S_u has a generating set X with the property that $e(u)$ multiplications are required to construct an element of even order from X. Then $e(u) = O(\log u)$.*

If this conjecture is approximately correct then, if $G = S_u$ for big enough u, there are points in $\overline{\Sigma}$ that are not virtuous with respect to being of even order. Note that $\log(u) = O(\log \log |G|)$.

Note that the proportion of elements of S_u of odd order is known exactly; see for example [**Cam94**], p72, Exercise 20. The above heuristics in fact suggest that $e(u)! \sim 2\sqrt{u}$.

10. Conclusion

It appears that the good behaviour of product replacement and its variants may be connected to a number of elementary and plausible conjectures which may be very hard to prove. It is instructive to compare the use of these algorithms to find elements of the finite group G as straight line programs in a given generating set X with the methods used in the matrix recognition project [**LG01**] to express a given element of G in this form. The method used in matrix recognition is to find a composition series for G, and use specialist algorithms for the simple components. While [**Pak99**] has shown that product replacement behaves badly when G is, for example, the direct product of a large number of copies of the alternating group A_5, this objection does not apply to rattle; and it seems quite plausible to suppose that the performance of rattle is to some extent determined by its performance on the composition factors of G.

References

[BP99] László Babai and Igor Pak, *Strong bias of group generators: an obstable to the "product replacement algorithm"*, Preprint, July 1999.

[Cam94] Peter J. Cameron, *Combinatorics: topics, techniques, algorithms*, Cambridge University Press, Cambridge, 1994.

[CLGM+95] Frank Celler, Charles R. Leedham-Green, Scott H. Murray, Alice C. Niemeyer, and E. A. O'Brien, *Generating random elements of a finite group*, Comm. Algebra **23** (1995), no. 13, 4931–4948.

[DSC98] P. Diaconis and L. Saloff-Coste, *Walks on generating sets of groups*, Invent. Math. **134** (1998), no. 2, 251–299.

[GS97] Charles M. Grinstead and J. Laurie Snell, *Probability*, American Mathematics Society, 1997, Second revised edition.

[LG01] Charles R. Leedham-Green, *The computational matrix group project*, Groups and computation, III (Columbus, OH, 1999), de Gruyter, Berlin, 2001, pp. 229–247.

[LGO97] C. R. Leedham-Green and E. A. O'Brien, *Recognising tensor products of matrix groups*, Internat. J. Algebra Comput. **7** (1997), no. 5, 541–559.

[LS95] Martin W. Liebeck and Aner Shalev, *The probability of generating a finite simple group*, Geom. Dedicata **56** (1995), no. 1, 103–113.

[Pak99] Igor Pak, *On the graph of generating sets of a simple group*, preprint, July 1999.

[Pak01] Igor Pak, *What do we know about the product replacement algorithm?*, Groups and computation, III (Columbus, OH, 1999), de Gruyter, Berlin, 2001, pp. 301–347.

C. R. Leedham-Green, Queen Mary, University of London, Mile End Road, London E1 4NS, United Kingdom

E-mail address: C.R.Leedham-Green@qmw.ac.uk

Scott H. Murray, School of Mathematics and Statistics F07, University of Sydney NSW 2006, Australia

E-mail address: murray@maths.usyd.edu.au

URL: www.win.tue.nl/~smurray