Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Mathematics 306 (2006) 420–428
DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Fast perfect sampling from linear extensions

Mark Huber

Department of Mathematics and Institute of Statistics and Decision Sciences, Duke University, Durham, NC 27713, USA

Received 19 November 2002; received in revised form 12 December 2005; accepted 4 January 2006
Available online 14 February 2006

Abstract

In this paper, we study the problem of sampling (exactly) uniformly from the set of linear extensions of an arbitrary partial order. Previous Markov chain techniques have yielded algorithms that generate approximately uniform samples. Here, we create a bounding chain for one such Markov chain, and by using a non-Markovian coupling together with a modified form of coupling from the past, we build an algorithm for perfectly generating samples. The expected running time of the procedure is $O(n^3 \ln n)$, making the technique as fast as the mixing time of the Karzanov/Khachiyan chain upon which it is based.

Keywords: MCMC; Perfect simulation; Linear extensions

1. The problem

Consider $[n] = \{1, \ldots, n\}$ and a partial order $P = (\preccurlyeq, [n])$ on this set. A *linear extension* of P is a permutation σ of the elements of $[n]$ that respects the partial order. That is, if $i < j$ then $\sigma(i) \preccurlyeq \sigma(j)$. Let Ω be the set of all linear extensions of P. Our goal is to generate a sample uniformly at random from Ω. This problem can also be viewed as choosing a uniformly random permutation subject to certain precedence constraints.

There are several applications associated with this problem, the most basic being an approximation algorithm for the probability that $\sigma^{-1}(i) < \sigma^{-1}(j)$ for unrelated i and j. Several combinatorial objects of interest (such as multiset permutations and binary trees) are isomorphic to linear extensions. Other applications include near-optimal sorting and decision theory. The problem is self-reducible, therefore a polynomial time algorithm for sampling also yields a fully polynomial time approximation scheme (*fpras*) for finding $|\Omega|$ [12] (the first *fpras* for this problem came from Dyer et al. [6]). The problem of finding $|\Omega|$ was shown by Brightwell and Winkler [2] to be $\sharp P$ complete.

Mathews [14] gave a geometric approach to the problem of approximate generation of variates, but later work concentrated on Markov chain approaches.

Karzanov and Khachiyan [13] used geometric and conductance arguments to show that a simple combinatorial Markov chain for the problem was rapidly mixing, although their bound on the running time was not tight. Bubley and Dyer [4] gave a slightly modified chain with $O(n^3 \ln n)$ mixing time that they proved using the technique of path coupling. Most recently, Wilson [16] was able to show that the original Karzanov/Khachiyan chain mixes in time $\Theta(n^3 \ln n)$.

E-mail address: mhuber@math.duke.edu.

0012-365X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2006.01.003

Protocols such as coupling from the past (CFTP) [15] and FMMR [8] allow perfect samples to be taken exactly from the stationary distribution of the Markov chain without the need to know the normalizing constant of the distribution. These methods require a means for detecting complete coupling such as monotocity or bounding chains. Felsner and Wernisch [7] were able to show that in the special case that the partial order is two-dimensional, the Karzanov/Khachiyan chain is monotonic, giving a means for detecting complete coupling, thereby allowing use of CFTP or FMMR to obtain perfect samples in this special case. Unfortunately, for general partial orders, the KK chain is not monotonic.

Our main result is a construction of a bounding chain for the KK chain that is guaranteed to detect complete coupling in polynomial time. Bounding chains were first developed in [10,9] as a means for detecting complete coupling in nonmonotonic situations. We will show that the bounding chain detects complete coupling with probability at least $1-\varepsilon$ after $(16/\pi^2)n^3(\ln n+(\frac{1}{2})\ln(2/[\pi\varepsilon]))$ steps. Utilizing the coupling lemma of Doeblin [5], this gives an independent proof of the order of the mixing time of the chain. Our secondary result is an extension of coupling from the past to deal with non-Markovian couplings. Together, these ideas give a perfect sampling algorithm for the problem whose running time has expectation $O(n^3 \ln n)$, and that concentrates strongly around the expected running time.

2. The Karzanov/Khachiyan chain

Let $\mathbf{K}(\sigma,\tau) = P(X_{t+1}=\tau|X_t=\sigma)$ be the kernel of our Markov chain. Given a permutation σ, let $\sigma_{i,j}$ be the permutation that applies σ and then transposes the elements in positions i and j, so $\sigma_{i,j}(j)=\sigma(i)$, $\sigma_{i,j}(i)=\sigma(j)$, and $\sigma_{i,j}(k)=\sigma(k)$ for all $k\notin\{i,j\}$. The chain of Karzanov and Khachiyan [13] moves in the following fashion. First, the chain holds its current position with probability $\frac{1}{2}$, if it does not hold it attempts to move. A move involves choosing an adjacent transposition at random from the $n-1$ such transpositions, and then transposing the elements as long as the precedence constraints are not violated by such a move. If we choose not to move, the next state is the same as the previous one. To be precise,

$$K(\sigma,\tau)=\begin{cases}1/[2(n-1)], & \exists\, i:(\tau=\sigma_{i,i+1})\wedge\neg(\sigma(i)\preccurlyeq\sigma(i+1)),\\ 1-\sum_{\tau\neq\sigma}K(\sigma,\tau), & \tau=\sigma,\\ 0 & \text{otherwise.}\end{cases}\tag{1}$$

To build our perfect sampling algorithm we will be using bounding chains. Bounding chains allow us to use protocols such as CFTP [15] or the FMMR method [8] to create a perfect sampling algorithm. The first step in creating a bounding chain is the development of a complete coupling for the chain.

In a complete coupling for a Markov chain, we have a sequence of deterministic functions, along with a countably infinite source of uniformly random bits. Our random bits will be represented by sequence of uniform random variables on [0, 1] which we shall denote by $U_0, U_1, \ldots$, and we denote our sequence of deterministic functions by $\phi_0, \phi_1, \ldots$. The functions $\phi_t : \Omega\times[0,1]^{t+1}\to\Omega$ each take the current state of the chain x_t, the uniform values $U_0,\ldots,U_t$, and outputs the next state of the chain x_{t+1}.

Let $H_t=(X_0,U_0,\ldots,U_{t-1})$ denote the history of the process up to time t. Call the $\{\phi_t\}_{t=0}^{\infty}$ a *complete coupling* of a Markov chain with kernel K if for all t

$$P(\phi_t(X_t,U_0,\ldots,U_t)\in A|X_t=x_t,H_t)=K(x_t,A)\tag{2}$$

for any measurable A. Note that conditioning on U_0 through U_{t-1} gives the values of X_1 through X_t as well. (The term complete coupling distinguishes it from the simple pairwise couplings that are all that is needed to prove rapid mixing of Markov chains.)

When there exists ϕ'_t such that $\phi_t(X_t,U_0,\ldots,U_t)=\phi'_t(X_t,U_t)$ for all t, we call $\{\phi_t\}$ *Markovian*, otherwise we refer to it as *non-Markovian*. It is important to note that even with a non-Markovian coupling, the Markov chain itself will always have the Markov property. Virtually all couplings of interest in proving mixing times of Markov chains have been Markovian, and the original coupling from the past [15] was designed with Markovian complete couplings in mind. However, the extension to non-Markovian complete couplings, while not trivial, is fairly straightforward, and is presented in Section 4.

In the meantime, we turn our attention to the complete coupling used for our chain of interest.
Karzanov/Khachiyan complete coupling

Random Choice:
Choose position i uniformly at random from $\{1, \ldots, n-1\}$
Choose c uniformly at random from $\{0, 1\}$

Evaluation of $X_{t+1} = \phi(X_t, i, c)$:
If $c = 0$ or $X_t(i) \preccurlyeq X_t(i+1)$
Set $X_{t+1} \leftarrow X_t$
Else
Set $\sigma \leftarrow X_t$, **Set** $X_{t+1} \leftarrow \sigma_{i,i+1}$

At each step we choose a random position i and flip a fair coin. Depending on the coin flip, we either make the adjacent transposition of the elements at positions i and $i+1$ if doing so would not violate the partial order constraint. Otherwise the state remains unchanged. This will be a Markovian complete coupling if our choice of i and c is solely a function of U_t, otherwise $\phi(X_t, i, c)$ also depends upon $U_0, \ldots, U_{t-1}$ and will be non-Markovian.

To measure the mixing time of such a chain, a metric on distributions such as the total variation distance can be used:

$$\|p - q\|_{\mathrm{TV}} = \frac{1}{2} \sum_{x \in \Omega} |p(x) - q(x)|. \tag{3}$$

If $p^t_{x_0}$ is the distribution of X_t given $X_0 = x_0$, then one means for bounding the mixing time is the coupling lemma [1,5].

Lemma 1. *Suppose* (A, B) *are a coupled random process* (*so marginally*, *A and B are both copies of the Markov chain*). *Suppose that* $A_0 = a_0$ *and* $p^t_{a_0}$ *is the distribution of* A_t. *If* B_0 *has distribution* π, *the stationary distribution of the chain*, *then*

$$\|p^t_{a_0} - \pi\|_{\mathrm{TV}} \leqslant \mathbf{P}(A_t \neq B_t).$$

3. The bounding chain

The purpose of a bounding chain is to start with an unknown state and by running the chain gain information about the state (see [11]). Let $\{X_i\}_{i=0}^{\infty}$ be an instance of the Markov chain. Then at each position i, $X_t(i)$ is a single value in $\{1, \ldots, n\}$.

The bounding chain does not keep a single choice from $\{1, \ldots, n\}$, instead it keeps track of a set of possible states. That is, for $\{Y_t\}_{i=0}^{\infty}$ an instance of the bounding chain process, $Y_t(i)$, is a subset of $\{1, \ldots, n\}$. That is, $Y_t(i)$ is an element of $2^{[n]}$.

Definition 2. $Y \in (2^{[n]})^{[n]}$ *bounds* permutation X if for all i, $X(i) \in Y(i)$.

To be a bounding chain that respects the complete coupling of the chain, we require that for each step

$$X_t(i) \in Y_t(i) \quad \forall i \Rightarrow X_{t+1}(i) \in Y_{t+1}(i) \quad \forall i$$

when steps are taken according to a particular complete coupling. Equivalently, Y_t bounds X_t should imply that Y_{t+1} bounds X_{t+1}. By starting with $X_0(i) \in Y_0(i)$ for all i (which is easy to do, just make $Y_0(i) = \{1, \ldots, n\}$ for all i); this guarantees that $X_t(i) \in Y_t(i)$ for all i and for all $t \geqslant 0$.

Suppose that the state at time X_0 is stationary. The state may be unknown, but we do know the value of Y_0. If Y_t is a bounding chain, we run Y_0 until Y_t only bounds one state in Ω. At this point, the "unknown" starting state at X_0 has become the known state that is bounded by Y_t. Also, if Y_0 bounds all starting states a_0, then the process started from any starting state has coupled with the stationary process. Therefore, the total variation distance from any starting state bounding by Y_0 is at most the probability that Y_t bounds more than one state. This means that the bounding chain can be used to upper bound the mixing time of the chain. However, this is not the most valuable aspect of bounding chains.

A bounding chain can be used as a black box for the perfect sampling protocols such as CFTP [15] and FMMR [8]. These protocols require exactly what bounding chains provide: a means for moving from an unknown stationary state to a known state. Armed with this black box, CFTP or FMMR and other protocols like them can generate samples exactly from the stationary distribution of the chain. The running time of CFTP has the same order as the expected running time for the bounding chain to reach a singleton state.

In the particular case of the KK chain, we do something slightly different. Instead of bounding X directly by Y, we use Y' to bound X^{-1}. That is, we run Y' in such a way that $X^{-1}(a) \in Y'(a)$ for all a. Because permutations are one to one and onto, if $Y'(a)$ bounds a single state X^{-1} we can compute X easily.

We do this because the sets $Y'_t(a)$ will take on a very simple form, namely, $Y'_t(a) = \{1, \ldots, R_t(a)\}$ for all $a \in \{1, \ldots, n\}$. Because the values of R_t and Y_t are determined by one another, we only need to show how to take steps in the bounding chain on the R process.

Without loss of generality, we may assume that the identity permutation is a valid linear extension, so that $i \preccurlyeq j$ implies $i < j$. If it is not, then it is easy to create some linear extension S in $O(n \ln n)$ time via sorting. S^{-1} may then be applied to the labels to give a new problem where the identity permutation is a valid linear extension. We then generate a uniform linear extension from this problem, and because S is 1–1 and onto, applying S to this sample gives a uniform sample from the original problem.

At each step of the chain we pick a position i to move uniformly at random from all choices independent of prior $U_{t'}$. Now, the only values of $R_t(a)$ that can change are those with $R_t(a) = i$ or $R_t(a) = i + 1$. In fact, we will not concern ourselves with changes in R_t over the entire set of elements $[n]$, but rather over a restricted set of elements $[p_t]$. Elements i in $[n]\backslash[p_t]$ will always have $R_t(i) = n$, the trivial bound.

Two invariants on R_t will be maintained:

$$(\forall a, b \in [p_t]) \quad (a \neq b \rightarrow R_t(a) \neq R_t(b)), \tag{4}$$

$$(\forall a, b \in [p_t]) \quad (a \preccurlyeq b \rightarrow R_t(a) < R_t(b)). \tag{5}$$

The bounding state can be illustrated pictorially. For example, if $R = (5, 1, 4, 5, 5)$ with $p = 3$, and partial order $2 \preccurlyeq 4$, $2 \preccurlyeq 5$, then permutations 25 134 and 23 451 both lie within the bounding chain. Using a right bracket] to represent the bounding state and x to represent the item assigned each position in the permutation, this example is drawn in Fig. 1.

Roughly speaking, if the Markov chain does not hold, the bounding chain swaps $R_t(a)$ and $R_t(b)$ if $R_t(a) = i$ and $R_t(b) = i + 1$ where $i, i + 1$ are the desired transpositions and a and b are unrelated. Pseudocode for this procedure follows.

Bounding chain step for Karzanov/Khachiyan chain

Choose position i uniformly at random from $\{1, \ldots, n - 1\}$
Choose c' uniformly from $\{0, 1\}$
Set $R_{t+1} \leftarrow R_t$, $p_{t+1} \leftarrow p_t$
If $c' = 1$ then
  If $\exists a, b \leqslant p_t : R_t(a) = i$, $R_t(b) = i + 1$, and a and b unrelated
    Set $R_{t+1}(a) = i + 1$
    Set $R_{t+1}(b) = i$
  Elseif $\exists a \leqslant p_t$ with $R_t(a) = i$ and $\nexists b \leqslant p_t$ with $R_t(b) = i + 1$
    Set $R_{t+1}(a) = i + 1$
  Elseif $\exists b \leqslant p_t$ with $R_t(b) = i + 1$ and $\nexists a \leqslant p_t$ with $R_t(a) = i$
    Set $R_{t+1}(b) = i$
If for all $i \leqslant p_t$, $R_{t+1}(i) < n$
  Set $p_{t+1} \leftarrow p_t + 1$

It is straightforward to check that this bounding chain step maintains the two invariants in (4) and (5).

Theorem 3. *There exists a complete coupling for a process X_t that is evolving according to the Karzanov/Khachiyan kernel such that (p_t, R_t) is a bounding chain for X_t.*

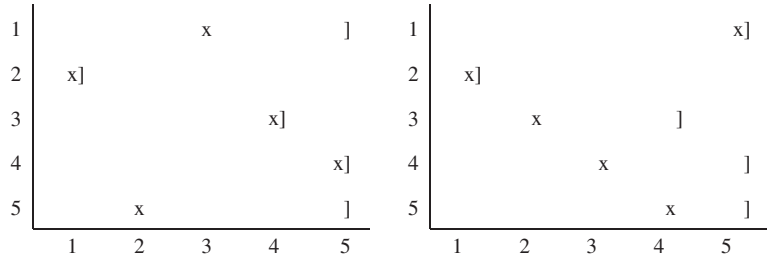

Fig. 1. Pictorial representation of bounding chain.

Proof. Given our choice of $i \sim \text{Unif}\{1, \ldots, n-1\}$ and $c' \sim \text{Unif}\{0, 1\}$, we wish to find $c \sim \text{Unif}\{0, 1\}$ such that

$$X_t^{-1}(a) \in \{1, \ldots, R_t(a)\} \quad \forall a \Rightarrow (\phi(X_t, i, c)^{-1}(a) \in \{1, \ldots, R_{t+1}(a)\} \quad \forall a, \tag{6}$$

where $\phi(X_t, i, c)$ is the complete coupling for the Karzanov/Khachiyan chain described earlier. In fact, c will be either c' or $1 - c'$ where the choice will depend on which of the several cases we are in.

Case 1: $R^{-1}(i) \cap [p] = \{a\}$, $R^{-1}(i+1) \cap [p] = \{b\}$, a and b are unrelated in the partial order. Then there are five types of states X bounded by R. Let $*$ be a wildcard symbol that denotes any element other than a and b. Then several subcases need to be considered.

Subcase $(X(i), X(i+1)) = (*, *)$ is easy to deal with: set $c = c'$. Since $X^{-1}(a)$ and $X^{-1}(b)$ are both at most $i - 1$ here, as long as we set $R'(a)$ and $R'(b)$ to at least i we are fine.

In the subcase where $(X(i), X(i+1)) = (a, *)$, set $c = c'$. Again we know that $X^{-1}(b)$ is less than i, so as long as $R'(b) > i - 1$ we are fine. $R(a)$ is a different matter. If $c = 0$, then we leave the state unchanged, so $R'(a) = i$ works. If $c = 1$ then a might be moved to position $i + 1$ or stay at i. Either way, setting $R'(a) = i + 1$ bounds this situation.

Subcase $(X(i), X(i+1)) = (b, *)$ sets $c = 1 - c'$. We know that $X^{-1}(a) < i$, so any $R'(a) > i - 1$ works. If $c' = 1$, then X is unchanged and $R'(b) = i$ is fine. If $c' = 0$, then we set $R'(b) = i + 1$ to bound the next state.

Subcase $(X(i), X(i+1)) = (*, b)$ sets $c = c'$. Our concern here is that the wildcard item might precede b, preventing it from swapping in the case $c' = 1$. Because of our invariants, this does not happen. Suppose $\omega \neq a$ and $\omega \preccurlyeq b$, we shall show that $X(i) \neq \omega$. Recall that the identity permutation is a valid linear extension, and so $b \leqslant p$ implies $\omega \leqslant p$ as well, and this means that ω is in play and our invariants apply. From our second invariant we know that $R(\omega) < i + 1 = R(b)$, and from our first invariant we know that $R(\omega) \neq i$ since $R(i) = a$. Together we have $R(\omega) < i$, and so $X(i) \neq \omega$.

In the final subcase, $(X(i), X(i+1)) = (a, b)$. Again set $c = c'$. So if $c' = 0$, the state is unchanged and we stay with $R(a) = i$, $R(b) = i + 1$. We know that a and b are unrelated in this case, so if $c' = 1$, we can make the transposition, and $R'(a) = i + 1$, $R(b) = i$ bounds the state.

Examining all the subcases, if $c' = 0$, then $R'(a) = i$, $R'(b) = i + 1$ bounds the next state, while if $c' = 1$, then $R'(a) = i + 1$, $R'(b) = i$ suffices.

Case 2: $R^{-1}(i) \cap [p] = \{a\}$, $R^{-1}(i+1) \cap [p] = \{b\}$, $a \preccurlyeq b$. There are fewer subcases in this case. In particular, we cannot have $(a, *)$, since if $X(i) = a$, then the fact that $a \preccurlyeq b$ and $R(b) = i + 1$ forces $X(i+1)$ to be b. This leaves four possibilities: $(X(i), X(i+1)) = (*, *)$, $(X(i), X(i+1)) = (b, *)$, $(X(i), X(i+1)) = (*, b)$, or $(X(i), X(i+1)) = (a, b)$.

Interestingly enough, in this case $R'(a) = i$ and $R'(b) = i + 1$ works regardless of the choice of c. Of course, setting $R'(b) = i + 1$ always works since b can at worst move to state $i + 1$ when position i is chosen.

When $(X(i), X(i+1)) \in \{(*, *), (b, *), (*, b)\}$ we have $X^{-1}(a) < i$, so again $R'(a) = i$ always bounds the next state. When $(X(i), X(i+1)) = (a, b)$ we cannot make the transposition since $a \preccurlyeq b$, so again a and b always stay in the same spot and $R'(a) = i$ as before.

Case 3: $R^{-1}(i) \cap [p] = \{a\}$, $R^{-1}(i+1) \cap [p] = \emptyset$. This means we need only consider $(X(i), X(i+1)) \in \{(*, *), (a, *)\}$, and case 1 analysis of these possibilities shows that $R'(a) = i$ when $c' = 0$ and $R'(a) = i + 1$ when $c' = 1$.

Case 4: $R^{-1}(i) \cap [p] = \emptyset$, $R^{-1}(i+1) \cap [p] = \{b\}$. Here $(X(i), X(i+1)) \in \{(*, *), (b, *), (*, b)\}$, and once more the analysis of case 1 indicates that $R'(b) = i + 1$ when $c' = 0$ and $R'(b) = i$ when $c' = 1$ yields a valid bounding chain.

Case 5: $R^{-1}(i) \cap [p] = \emptyset$, $R^{-1}(i+1) \cap [p] = \emptyset$. Here $R' = R$ works just fine.

Note that while we needed to consider the cases and subcases in designing the complete coupling, in running the bounding chain itself we can ignore such considerations and all we need is the value of c'.

Suppose that for all $i \in [p]$, $R(i) < n$. Then we can add $p+1$ to our in play items without violating an invariant. The first invariant follows from the fact that $R(p+1) = n$, and the second from the fact that the identity permutation is a valid linear extension, so $p+1$ cannot precede any element of $[p]$. □

Remark 4. The value of c (and hence the move X_t takes) depends on the value of the bounding chain state that in turn depends on $U_0, \ldots, U_{t-1}$, and so the coupling between (p_t, R_t) and X_t is a non-Markovian coupling.

In the end, a bounding chain is only useful if it ends up bounding only a single state. Suppose that $p_t = n$. Then for all $i \in [n]$, $R_t(i)$ is a different number from 1 up to n. Hence R is a 1–1 function from $[n]$ onto $[n]$. In particular, there is an element $x(1)$ such that $R_t(x(1)) = 1$, and so any X_t bounded by R_t has $X(1) = x(1)$. Similarly, there is an $x(2)$ such that $R_t(x(2)) = 2$, and position 1 is already taken, so any X_t bounded by R_t has $X(2) = x(2)$, and so on. In fact, $X_t = R_t^{-1}$ is the only state bounded by R.

Theorem 5. *For* $\varepsilon \leqslant 1$, *after* $(16/\pi)n^3[\ln n + (\frac{1}{2})\ln[2/(\pi\varepsilon)]]$ *steps, the probability that R bounds only one state is at least* $1-\varepsilon$.

Proof. As we have already pointed out, the bounding chain step maintains our two invariants. Given that we start with $p_0 = 1$, the invariants are both trivially true at time 0, and an induction verifies that for all t and $a \neq b$ for both elements of $[p_t]$, $R_t(a) \neq R_t(b)$ and if $a \preccurlyeq b$ then $R_t(a) \leqslant R_t(b)$.

When there is a position j where no a satisfies $R(a) = j$, say that there is a *hole* at j. Our analysis will proceed by examining the behavior of the holes in R_t. First note that if a hole is at the position i chosen in the bounding chain step, then if $c' = 1$ the hole moves one position to the right to $i+1$. On the other hand, if the hole is at $i+1$ and $c' = 1$, the hole moves to the left to i. When a hole reaches position n, it means that $p_{t+1} = p_t + 1$, and the hole vanishes permanently. This is a good thing: when all the holes have vanished, R_t bounds only a single state.

Consider the random walk taken by a particular hole, and call this process B_t. Wilson [16] studied the Karzanov/Khachiyan chain using a sinusoidal potential function, and the same approach can be used here with some slight modifications. Let

$$\Phi(i) := \frac{\sin(Ci)}{\sin(C)}, \quad C := \frac{\pi}{2(n-1)}. \tag{7}$$

It is straightforward to verify that $1/\sin C \leqslant 2n/\pi$. Since $\sin(Ci) \leqslant 1$, we have $\Phi(n - B_t) \leqslant 2n/\pi$. Also, $\Phi(i) = 0$ if and only if $i = 0$. When $i \geqslant 1$, $\Phi(i) \geqslant 1$ as well. Hence $\Phi(n - B_t) = 0$ when $B_t = n$ (and the hole vanishes) and at least 1 when $B_t < n$. Finally, the expected value of $\Phi(n - B_{t+1})$ given $n - B_t = i$ is just

$$\frac{1}{2(n-1)}\Phi(i-1) + \left(1 - \frac{1}{n-1}\right)\Phi(i) + \frac{1}{2(n-1)}\Phi(i+1). \tag{8}$$

This can be written more compactly using the second forward difference operator: $\Delta^2\Phi(i) := \Phi(i+2) - 2\Phi(i+1) + \Phi(i)$:

$$E[\Phi(n - B_{n+1}) | n - B_t = i] = \Phi(i) + \pi^{-1}C\Delta^2\Phi(i-1). \tag{9}$$

The sum of angle formula for $\sin x$ yields

$$\Delta^2 \sin(C(i-1)) = \sin(Ci)[2\cos(C) - 2], \tag{10}$$

$$E[\Phi(n - B_{t+1}) | n - B_t = i] \leqslant \Phi(i)[1 - \pi^{-1}C(2 - 2\cos(C))]. \tag{11}$$

Note $1 - \pi^{-1}C(2 - 2\cos C) \leqslant \exp(-\pi^2/[8n^3])$. This can be proved by using the first three nonzero terms of the Taylor series for $\cos x$ and the first two terms of the Taylor series for $\exp(-x)$. Also $E[E[X|Y]] = E[X]$. This (together with an induction on t) gives

$$E[\Phi(n - B_t)] \leqslant E[\Phi(n - B_0)]\exp(-t\pi^2/[8n^3]). \tag{12}$$

As noted earlier, $\Phi(n - B_0) \leqslant 2n/\pi$. Markov's inequality on $\Phi(n - B_t)$ says that $P(\Phi(n - B_t) \geqslant 1) \leqslant E[\Phi(n - B_t)]$. Setting $t = (8/\pi^2)[\ln([2n^2]/[\pi\varepsilon])]$ makes the probability that the hole has not reached the right-hand side after t steps at most ε/n.

Finally, note that the bounding chain begins with n holes: the probability that there exists one of these holes that has not reached the right-hand side and vanished after t steps is at most $n\varepsilon/n = \varepsilon$, completing the proof. □

Remark 6. In Wilson [16] the mixing time for the adjacent transposition chain is shown to be $(4/\pi)n^3(\ln n + \ln(1/\varepsilon))$. Our bound loses a factor of 4. This difference comes from the fact that Wilson coupled two processes together, thereby doubling the probability that movement occurs. Second, the two processes can wander anywhere in the space, whereas for the bounding chain analysis the holes are required to reach the right-hand side.

Remark 7. In the same paper [16], Wilson proves a lower bound on the variation threshold of $(1/\pi)n^3(\ln n + \ln(1/\varepsilon))$ and conjectures that this is correct. This result shows that the bounding chain time to obtain an exact sample differs from the mixing time of the chain (as measured by total variation distance) by at most a factor of 16.

4. Non-Markovian coupling from the past

At each step, the state of the bounding chain depends only on the value of the random coin flip c' and the previous state of the bounding chain. Therefore, the bounding chain itself evolves in a Markovian fashion, and it makes sense to call it a bounding chain rather than a more general bounding process.

On the other hand, the complete coupling for the original process depends on the current state of the bounding chain, which in turn depends on all of the $U_0, \ldots, U_{t-1}$ at time t. Hence the complete coupling itself is non-Markovian.

Now, the original formulation of CFTP was given in terms of a Markovian coupling. This is because this type of coupling does not require the storage of extra information about the past. This bounding chain for linear extensions appears to be the first nontrivial instance where a non-Markovian coupling is needed for speedy perfect sampling.

We now present modified non-Markovian coupling from the past in its entirety. The original proof does not carry over to the non-Markovian case, and so we give a new proof of correctness. Given the sequence of deterministic functions $\{\phi_t\}$ and random bits $U_0, U_1, \ldots$, define F_t recursively. Let F_0 be the identity function, and $F_t(x)$ be $\phi_{t-1}(F_{t-1}(x), U_0, U_1, \ldots, U_{t-1})$. Hence if $X_0 = x_0$, at any time t, $X_t = F_t(x_0)$.

CFTP has several different forms and variants; here we present the recursive approach. Suppose that the chain has stationary distribution π. Run the chain forward for a fixed number of steps T. Note that if $X_0 \sim \pi$, $F_T(X_0) \sim \pi$ as well. Check to see if F_T is a constant function. If it is, then it does not matter that we do not know X_0, the value $X_T = F_T(X_0)$ is determined anyway. With bounding chains, F_T will be constant when the bounding chain state Y_T bounds exactly one state.

If F_T is not a constant, then call CFTP recursively, doubling the time to run to $2T$. The output of this recursive call then becomes our stationary state X_0. We then run the chain forward to time X_T and output this state. In pseudocode:

Non-Markovian CFTP

`Input:` T, the number of time steps to run
`Output:` $X_T \sim \pi$

Choose $U_0, \ldots, U_{T-1}$ uniformly from $[0, 1]^{\mathrm{T}}$
If F_T is a constant function
  Let X_T be the constant value taken by F_T and exit
Else
  Let $X_0 \leftarrow \mathrm{CFTP}(2T)$
  Let $X_T \leftarrow F_T(X_0)$ and exit

As with the original CFTP, we note here that it is very important to only choose the random variates $U_0, \ldots, U_{t-1}$ once within a single call to CFTP. Once we have our $U_0, \ldots, U_{t-1}$ in hand, we must use the same values for finding F_T in the if expression and in the else clause of the if expression. In practice, this usually means storing the seed of the random number generator used to create the random variates.

Theorem 8. *For a finite state Markov chain with stationary distribution π, suppose that there exists a T such that the probability that F_T is constant is nonzero for all $T' \geqslant T$. Then non-Markovian CFTP terminates in finite time with probability* 1, *with output that comes exactly from π.*

Proof. Suppose the probability that F_T is constant is $\alpha > 0$. Then for $T' \geqslant T$, the probability that $F_{T'}$ is a constant is at least α. Each recursive call is independent of the others and so r recursions after the call of length T; the probability that more recursions are needed is bounded above by $(1-\alpha)^r$ which goes to 0 as r goes to infinity. In fact, it is easy to extend this idea to show that with probability 1, an infinite number of recursion levels will have F_T as a constant function.

Now suppose that we are dealing with such an outcome where F_T is constant in an infinite number of recursive levels. Consider the output X_T conditioned on the fact that we are dealing with such an outcome.

Then this output is the same if we modify CFTP so that it always takes at least r recursions. That is, even if F_T is constant, we call CFTP again until we are calling it with time 2^r. If the number of recursions it actually needs is greater than r, then forcing it to do at least r recursions does not change the run of the algorithm at all. And if it would stop recursing at $r' < r$, then in the r' recursion F_T is a constant function, and so it does not matter what is given as X_0.

Given $\varepsilon > 0$, there exists a time t such that taking t steps of the Markov chain from any starting state w_0 results in a random state W_t whose distribution is within ε total variation distance of π. After $\lceil \log_2 t \rceil$ recursions, one level of CFTP will get X_0 from its recursive call according to some distribution since we are in an outcome where the process always terminates no matter how many recursions we go back. It will then take $T \geqslant t$ steps in the Markov chain, and so the output to the $r-1$ recursive call of CFTP will have a distribution that is within ε of π. Total variation distance only decreases, and so after all the recursive calls have been evaluated, the final answer has distribution within ε of π as well.

But ε was arbitrarily small, and so this final answer must have a distribution within 0 total variation of π, and must come exactly from π. □

Remark 9. It is not necessary to double T every time, but this is a simple method that ensures that we quickly pass the point where F_T has a nonzero chance of being constant.

When we know a value of T such that F_T has a positive chance of being constant, we need not increase T in the recursive call at all, but can keep it the same. In fact, for linear extensions we can directly bound the expected running time of CFTP for doubling T and keeping T constant using Theorem 5. From the proof of Theorem 5 we know that if t steps are taken in the bounding chain, the probability that it has not collapsed to bound a single state is at most $(2/\pi)n^2 \exp(-t\pi^2/[8n^3])$. This can be used to show the following.

Lemma 10. *If we recursively call CFTP doubling the time allotted at each step, the expected running time of the perfect sampling algorithm for linear extensions is at most* $4.3n^3 \ln n$. *The probability the running time is more than* $k \cdot (16/\pi^2)n^3 \ln n$ *is at most* $1/n^{k-1}$ *for integer k. If we always recursively call CFTP with* $T = (16/\pi^2)n^3[\ln n + 1]$, *the expected time to run is at most* $1.8n^3[\ln n + 1]$ *and the probability that the running time is more than kT for integer k is at most* e^{-k}.

Three points should be noted about the running time of CFTP versus approximate algorithms. First, if the Markov chain is known to mix within ε total variation of stationarity in time $f(\varepsilon)$, then the running time of the approximate algorithm is $\Theta(f(\varepsilon))$. That is, unless we are willing to reanalyze the chain for each particular partial order, we must commit to running the chain for bound on the mixing time for all partial orders in order to generate approximate samples.

Bounding chains and the CFTP algorithms derived from them do not suffer from this flaw. If the partial order results in a faster mixing chain, then the bounding chain can reduce to a single state more quickly. Second, the output of CFTP comes exactly from the stationary distribution, and so there is no dependence on ε in any way. Third, while a bound on the expected running time is good, it is also important to note that the probability that the run time is very much larger than the expected time declines exponentially. The actual running time will be very tightly concentrated around the expected running time.

The advantage Markovian couplings usually have is that they do not require any additional storage of the values of the past sample path in order to progress forward. In our algorithm, the extra storage about the past is entirely contained within the bounding chain state itself, which does move in a Markovian fashion, and so although the underlying coupling is non-Markovian, the algorithm does not need to record extra information about the past above and beyond the state of the bounding chain itself.

5. Conclusions

The previous bounds on the mixing time of the KK chain were proved using the simple but powerful idea of path coupling [3,4]. While a number of path coupling arguments have now been extended to perfect sampling algorithms [10] it remains to be seen whether a general methodology for moving from approximate to perfect sampling exists, even in this restricted setting. For now, generating a useful bounding chain and a complete coupling is more difficult than finding a path coupling, but ultimately more powerful, as it allows for perfect samples to be obtained.

It is usually easy to write down bounding chains for a problem such as this; however, it is not always the case that the bounding chain running time is the same as the mixing time of the chain, and often it is far slower. This is a nontrivial example of a bounding chain for a chain that is not monotonic, but which actually does probably reduce to a single bounded state in the same time as the mixing time of the chain.

Because the bounding chain gives an upper bound on the mixing time and the chain is known to require $\Omega(n^3 \ln n)$ steps to mix, more complex bounding chains for the KK chain could only change the running time by a constant factor, and a new chain must be used to beat this bound.

References

[1] D. Aldous, Some inequalities for reversible Markov chains, J. London Math. Soc. 25 (2) (1982) 561–576.
[2] G. Brightwell, P. Winkler, Counting linear extensions, Order 8 (3) (1991) 225–242.
[3] R. Bubley, M. Dyer, Path coupling: a technique for proving rapid mixing in Markov chains, in: 38th Annual Symposium on Foundations of Computer Science, 1997, pp. 223–231.
[4] R. Bubley, M. Dyer, Faster random generation of linear extensions, Discrete Math. 201 (1999) 81–88.
[5] W. Doeblin, Exposé de la théorie des chains simples constantes de Markov à un nombre fini d'états, Rev. Math. de l'Union Interbalkanique 2 (1933) 77–105.
[6] M. Dyer, A. Frieze, R. Kannan, A random polynomial-time algorithm for approximating the volume of convex bodies, J. Assoc. Comput. Mach. 38 (1) (1991) 1–17.
[7] S. Felsner, L. Wernisch, Markov chains for linear extensions, the two-dimensional case, in: Proceedings of the Eighth ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana, 1997, pp. 239–247.
[8] J.A. Fill, M. Machida, D.J. Murdoch, J.S. Rosenthal, Extension of fill's perfect rejection sampling algorithm to general chains, Random Structures Algorithms 17 (2000) 290–316.
[9] O. Häggström, K. Nelander, On exact simulation from Markov random fields using coupling from the past, Scand. J. Statist. 26 (3) (1999) 395–411.
[10] M.L. Huber, Exact sampling and approximate counting techniques, in: Proceedings of the 30th Symposium on the Theory of Computing, 1998, pp. 31–40.
[11] M.L. Huber, Perfect sampling with bounding chains, Ph.D. Thesis, Cornell University, 1999.
[12] M. Jerrum, L. Valiant, V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theoret. Comput. Sci. 43 (1986) 169–188.
[13] A. Karzanov, L. Khachiyan, On the conductance of order Markov chains, Order 8 (1) (1991) 7–15.
[14] P. Mathews, Generating a random linear extension of a partial order, Ann. Probab. 19 (3) (1991) 1367–1392.
[15] J.G. Propp, D.B. Wilson, Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Structures Algorithms 9 (1–2) (1996) 223–252.
[16] D.B. Wilson, Mixing times of lozenge tiling and card shuffling Markov chains, Ann. Appl. Probab. 14 (1) (2004) 274–325.