Liftings of Tree-Structured Markov Chains (Extended Abstract)

Thomas P. Hayes[1] and Alistair Sinclair[2]

[1] Department of Computer Science, University of New Mexico*
[2] Computer Science Division, University of California at Berkeley**

Abstract. A "lifting" of a Markov chain is a larger chain obtained by replacing each state of the original chain by a set of states, with transition probabilities defined in such a way that the lifted chain projects down exactly to the original one. It is well known that lifting can potentially speed up the mixing time substantially. Essentially all known examples of efficiently implementable liftings have required a high degree of symmetry in the original chain. Addressing an open question of Chen, Lovász and Pak, we present the first example of a successful lifting for a complex Markov chain that has been used in sampling algorithms. This chain, first introduced by Sinclair and Jerrum, samples a leaf uniformly at random in a large tree, given approximate information about the number of leaves in any subtree, and has applications to the theory of approximate counting and to importance sampling in Statistics. Our lifted version of the chain (which, unlike the original one, is non-reversible) gives a significant speedup over the original version whenever the error in the leaf counting estimates is $o(1)$. Our lifting construction, based on flows, is systematic, and we conjecture that it may be applicable to other Markov chains used in sampling algorithms.

1 Introduction

1.1 Background and motivation

As the field of Markov chain Monte Carlo (MCMC) algorithms matures, attention is turning to refinements of these algorithms with improved running times. A general framework for speeding up MCMC algorithms, known as "lifting," was introduced ten years ago by Chen, Lovász and Pak [2]. A *lifting* of a Markov chain $\mathcal{M}$ is a larger chain $\mathcal{M}'$ obtained by replacing each state of $\mathcal{M}$ by a set of states; the lifting is required to preserve the structure of $\mathcal{M}$ in the sense that the obvious projection obtained by merging appropriate states of $\mathcal{M}'$ gets us back to $\mathcal{M}$ itself. (See Section 2 for a precise definition.) The intriguing fact, first observed by Diaconis, Holmes and Neal [4] and explored further by Chen, Lovász and Pak [2], is that lifting can in certain cases reduce the mixing time of the chain substantially, and hence potentially improve the running time of algorithms in which it is used.

The simplest example of lifting, due to [4], is for simple random walk on the path of length n, with uniform stationary distribution. The mixing time of this chain is well known to be $\Theta(n^2)$. This chain can be lifted by replacing each node by a pair of nodes,

* Email: hayes@cs.unm.edu

** Email: sinclair@cs.berkeley.edu. Supported in part by NSF grant CCF-0635153 and by a UC Berkeley Chancellor's Professorship.

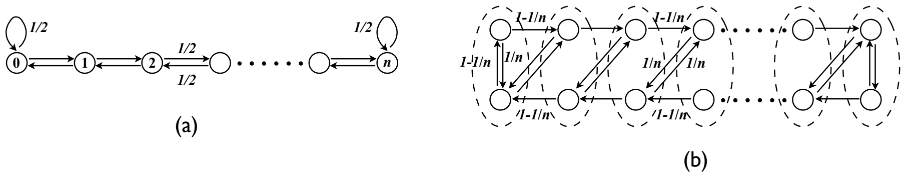

Fig. 1. (a) Simple random walk on a path of length n. (b) The lifted walk; dotted ovals indicate nodes that project to the same node in the original walk.

with the two sets of copies connected in two *directed* paths with opposite senses, and bidirected crossing edges between the paths. (See Fig. 1.) If the crossing probabilities are chosen appropriately (of order $1/n$), then the mixing time drops to $\Theta(n)$. The lifting achieves this speedup by almost eliminating the diffusive behavior of the original symmetric walk, and instead giving the walking particle "momentum" in its current direction of travel. In particular, after $t < n$ steps the lifted walk will typically be at distance $\Theta(t)$ from its starting point, in contrast to $\Theta(\sqrt{t})$ for the original walk.

This idea was extended by Chen et al. [2] to random walks on Cayley graphs. The strategy, roughly, is to lift the state space Ω to $\Omega \times \{1, \ldots, r\}$ where each $i \in \{1, \ldots, r\}$ is associated with a generator, and then to give the walk momentum around a carefully chosen cycle through the generators. The authors give several examples of significant speedups using this construction.

Chen et al. also give a general lifting construction that applies to arbitrary Markov chains, and achieves a mixing time of $O(\rho)$, where ρ is a multicommodity flow parameter (in the original chain) that is almost the inverse of the more familiar "conductance" (or sparsest cut); they also show that this is essentially best possible. Unfortunately, however, this construction is in general not feasible to implement, as simulating even one step of the lifted chain may be as hard as sampling from the stationary distribution π. Chen et al. pose the open question whether lifting can be used to speed up actual sampling algorithms.

In this paper we prove what is apparently the first result in this direction. We revisit a Markov chain introduced by Sinclair and Jerrum [26] which samples a leaf of a tree uniformly at random given crude estimates of the number of leaves in each subtree. This Markov chain was used in [26] to prove that approximate counting for all self-reducible problems in $\#P$ is robust, in the sense that such problems either have a fully-polynomial randomized approximation scheme, or cannot be approximated in polynomial time within *any* polynomial factor (even, say, n^{100}). It was also used in the same paper to give a polynomial time algorithm for uniformly generating random graphs with specified vertex degrees, based on analytic estimates for the number of such graphs [22].

To describe the setting more precisely, let T be a binary[1] tree, all of whose leaves are at the same depth d. Our goal is to sample a leaf of T uniformly at random, in time polynomial in d. This fundamental problem goes back at least to Knuth [17]. Suppose we are given partial information about T in the form of an estimate $\widetilde{N}_v$ of the number

[1] We make this assumption for simplicity of presentation only; T may in fact have an arbitrary branching factor.

of leaves N_v in the subtree rooted at each node v. This estimate is guaranteed to be within ratio $1+\delta$, i.e., $(1+\delta)^{-1}N_v \leq \widetilde{N}_v \leq (1+\delta)N_v$. (Such estimates may be available, e.g., from a crude approximate counting algorithm as in the abstract framework of [26], from analytic approximations as for graphs with given degrees in [22], or from the solution of idealized approximations as in the derivative pricing framework discussed in [3]. In the Statistics literature, the use of such estimates is often referred to as "importance sampling.") If $\delta = O(\frac{1}{d})$ then we can solve the problem rather easily by choosing a random path from the root to a leaf, branching left or right at each node with probabilities proportional to the counting estimates at its two children. Because of the bound on δ, we will accumulate at most a constant bias at the leaves, which can be eliminated by "rejection sampling" with a constant number (in expectation) of repeated trials. (In rejection sampling, if a leaf ℓ is sampled with probability p_ℓ then we output the leaf with probability p^*/p_ℓ, where p^* is a lower bound on p_ℓ for all ℓ, and start again otherwise. The reader is referred to the full version of the paper for a detailed discussion of rejection sampling, including a comparison with the Markov chain approach.)

For larger values of δ the above approach breaks down. To overcome this obstacle, Sinclair and Jerrum [26] introduced a more involved sampling algorithm that runs in polynomial time provided δ is bounded by any constant (or indeed, by any polynomial in d). This algorithm works by simulating a Markov chain on T whose transition probabilities are proportional to the edge weights $\widetilde{N}_v$ (where we think of $\widetilde{N}_v$ as being associated with the edge whose lower endpoint is v). Note that transitions from a node to its parent are allowed, so backtracking occurs. The stationary distribution of this Markov chain is easily seen to be uniform over leaves, and to put a constant fraction of its weight on the leaves.[2] Perhaps surprisingly, the mixing time for $\delta = O(1)$ was shown in [26] to be $\widetilde{O}(d^2)$, implying that the algorithm outputs a uniformly random leaf with bias ε in expected time $\widetilde{O}(d^2 \log \varepsilon^{-1})$. The intuition for the effectiveness of this algorithm is that an overestimate $\widetilde{N}_v$, which leads the chain to choose a downward edge to v with too large probability, also acts to increase the probability of backtracking from v; thus the process is "self-correcting." We note also that the $\Omega(d^2)$ dependence on d is unavoidable as, even in the case of perfect estimates ($\delta = 0$), the process reduces to symmetric random walk on the levels $[0, d]$.

1.2 Results

In this paper we consider lifting the above Markov chain in the regime $\delta \in [\frac{1}{d}, 1]$. (Recall that the problem is trivial for $\delta = O(\frac{1}{d})$.) Our main result is a (non-reversible) lifting that speeds up the mixing time to $O(\delta d^2)$ throughout this range. Thus our lifted chain interpolates smoothly between a trivial linear time rejection sampling algorithm when $\delta = O(\frac{1}{d})$ and the Sinclair-Jerrum quadratic time algorithm when $\delta = \Omega(1)$. In particular, for all $\delta = o(1)$ the lifted chain overcomes the $\Omega(d^2)$ diffusion lower bound on the mixing time of the original chain. (For example, when $\delta = O(\frac{1}{\sqrt{d}})$, we are able to sample leaves in time $O(d^{3/2})$.) We leave open the question of whether a fast lifting exists for larger values of δ.

[2] The original chain in [26] puts weight $O(1/d)$ on the leaves; a simple modification, which we provide, improves this to a constant with the same bound on mixing time.

We believe that the main interest value of this result is as the first application of lifting to a complex Markov chain used in random sampling.[3] However, we briefly mention as an example one potential concrete application. Let $\mathbf{g} = (g_1, \ldots, g_n)$ be a graphical degree sequence on n vertices, and suppose we wish to sample a random graph in which vertex i has degree g_i for each i. We can construct such graphs edge-by-edge, giving rise to a "self-reducibility tree" in which each node corresponds to a partial graph (of edges previously chosen) and a residual degree sequence; the leaves of the tree are precisely the desired graphs (see [26] for details). Note that the depth of this tree is $d = |E(\mathbf{g})|$, the number of edges in the graphs. Classical work of McKay [22] (see also [10, 23]) provides analytical estimates for the number of graphs with given vertex degrees that are within ratio $1 + O(\frac{g_{\max}^4}{|E(\mathbf{g})|})$, where $g_{\max} = \max_i g_i$. In [26] the Sinclair-Jerrum Markov chain was used with these estimates to sample graphs from sequences in which $g_{\max} = O(|E(\mathbf{g})|^{1/4})$. The lifting in the present paper would potentially improve the mixing time of this Markov chain from $\widetilde{O}(|E(\mathbf{g})|^2)$ to $O(g_{\max}^4 |E(\mathbf{g})|)$, which is significantly less when $g_{\max} \ll |E(\mathbf{g})|^{1/4}$.

Since our construction is the main contribution of the paper, we say a few words about it here. We stress that the construction is purely local and can be implemented efficiently, unlike the optimal liftings discussed in [2]. Our lifting creates two copies of the tree, having "upward" and "downward" momentum respectively. To eliminate diffusive behavior, we need to arrange for small crossing probabilities between the two copies; this we achieve using a "flow cancellation" idea that is facilitated by our view throughout the paper of Markov chains as flows. Another key ingredient is smoothing of the holding time distribution at some nodes; we achieve this by lifting certain self-loops in the original chain to two-state "traps." This smoothing makes possible our analysis of the mixing time via a non-Markovian coupling argument.

While some of the above features can be identified with hindsight in the efficient liftings of [4, 2], our construction is considerably more general and systematic. In particular, we do not exploit strong symmetries in the original Markov chain which make the liftings in those papers rather simpler to construct and to analyze. Indeed, in our case the original Markov chain is not at all symmetrical, as the tree may have arbitrary structure and its edge weights may vary arbitrarily within their respective ranges. For the same reason, the tree is also very far from the one-dimensional processes analyzed in [4, 7, 8]. We conjecture that our flow-based approach may lead in future to a systematic framework for constructing liftings in a larger class of Markov chains where it is possible to identify generalized "directions" along which momentum can be defined.

In the full version of the paper, we discuss alternative approaches to the leaf-sampling problem for $\delta \in (0, 1]$ based on rejection sampling combined with Markov chain Monte Carlo.

1.3 Related work

The first authors to implicitly discuss lifting of Markov chains to speed up mixing were Diaconis, Holmes and Neal [4], who observed that the mixing time of simple random

[3] We mention that, in hindsight, the "hit-and-run" Markov chain [19] used for sampling points in a convex body has the flavor of a "lifting" of the more classical "ball walk" [20]. We return to this point in Section 5.

walk on a path of length n can be improved from $\Theta(n^2)$ to $\Theta(n)$. They also proposed an extension to more general one-dimensional chains with non-uniform stationary distribution, but did not provide bounds on the mixing time. Such an extension was subsequently analyzed by Hildebrand [7, 8], who showed that a similar acceleration to $\Theta(n)$ occurs when the stationary distribution is log-concave.

Chen, Lovász and Pak [2] studied lifting in a more general framework. In addition to giving several examples of liftings for random walks on Cayley graphs, they also proved general results on the scope and limitations of lifting. For example, they show that the best possible lifting of any given Markov chain has mixing time (suitably defined) $\Theta(\rho)$, where ρ is the flow parameter mentioned earlier. Since the mixing time is always $\widetilde{O}(\rho^2)$, the optimal speedup via lifting is at most roughly a square root. Chen et al. also give a theoretical construction that achieves this optimal lifting (up to a constant factor) for an arbitrary Markov chain; however, as mentioned earlier, this construction is in general not efficiently implementable. Moreover, they show that if the lifted Markov chain is reversible then the speedup obtainable is (relatively) negligible; hence any useful lifting needs to be non-reversible (as are all the liftings mentioned in this paper).

Jung, Shah and Shin [14] build on the work of Chen et al. by considering the problem of minimizing the size of the lifted Markov chain while still achieving a similar speedup. This measure has applications to distributed algorithms for computing averages in networks, which the same authors discuss in [15].

We mention that all of the above lifting constructions, like our own, seek to eliminate or reduce diffusive behavior in the Markov chain. This is also the idea behind other, more classical techniques for speeding up Markov chain Monte Carlo algorithms, notably Hybrid Monte Carlo [5] and Horowitz's method [9] (see also [27] for more recent work in this direction). However, to the best of our knowledge, these methods lack rigorous analysis in non-trivial examples.

The problem of sampling leaves of a tree can be traced back at least to Knuth [17] in his work on estimating the efficiency of branching programs. Knuth sampled leaves by branching uniformly to children regardless of the number of leaves in the corresponding subtree, which yields a non-uniform distribution $\{p_\ell\}$ over leaves ℓ; he then used the quantity p_ℓ^{-1} as an unbiased estimator of the number of leaves in the tree. This can be seen as the origin of the rejection sampling approach mentioned earlier. A paper by Rosenbaum [24] provides some further analysis and refinement of Knuth's scheme.

The Markov chain approach to leaf sampling appeared in the work of Sinclair and Jerrum [26], where the main application was to show robustness of approximate counting for self-reducible problems. The version of the Sinclair-Jerrum chain presented here is slightly more efficient than the original one. The same paper also applied this Markov chain to give the first polynomial time sampling algorithm for subgraphs of a given graph that have specified vertex degrees, under certain constraints on the maximum degree, using the fact that analytic approximations exist for the number of such graphs (see, e.g., [22]). For subsequent developments on this problem, see [11, 16, 1].

2 Preliminaries

2.1 Markov chains, liftings and mixing times

Markov chains. Let Ω be a finite state space. We shall specify Markov chains on Ω using the following weighted graph framework.

A reversible chain is specified by an undirected graph $G = (\Omega, E)$ (possibly with self-loops) with a positive weight Q_e on each edge $e \in E$. Transitions from any vertex $u \in \Omega$ are made with probabilities proportional to the edge weights: i.e., the transition probability from u to v is $P(u, v) = \frac{Q_{(u,v)}}{W_u}$, where $W_u = \sum_{e \ni u} Q_e$ is the sum of the edge weights incident at u.

This Markov chain is easily seen to be reversible with respect to the distribution $\pi(u) = \frac{W_u}{W}$, where $W := \sum_u W_u$ (i.e., π is proportional to the weighted vertex degrees). As is well known, if G is connected and not bipartite (e.g., a single self-loop suffices) then it is ergodic and converges to π from any initial state. Note that the edge weights Q_e are, up to scaling by W, the *ergodic flows* in the stationary distribution; i.e., $Q_{(u,v)} = W\pi(u)P(u, v) = W\pi(v)P(v, u) = Q_{(v,u)}$.

The above framework can be extended to general, non-reversible Markov chains by making G directed and requiring that the edge weights Q_e satisfy the flow condition $\sum_{u:(u,v)\in E} Q_{(u,v)} = \sum_{u:(v,u)\in E} Q_{(v,u)} =: W_v$ for all $v \in \Omega$. If G is strongly connected and aperiodic (again, a single self-loop suffices) then it again converges to the unique stationary distribution $\pi(v) \propto W_v$. Again Q_e is proportional to the ergodic flow along (directed) edge e.

Mixing times. For an ergodic Markov chain $(X_t)_{t\geq 0}$ on Ω with stationary distribution π, any $x \in \Omega$ and any $\varepsilon \in (0, 1]$, we define

$$\tau_x(\varepsilon) = \min\{t : \|\eta_{x,t} - \pi\| \leq \varepsilon\},$$

where $\eta_{x,t}$ denotes the distribution of X_t (the state at time t) starting from initial state $X_0 = x$, and $\| \cdot \|$ is total variation distance. We will refer to $\tau_x(\varepsilon)$ as the *mixing time starting from state* x. The *mixing time*, $\tau(\varepsilon)$, is defined as the maximum over $x \in \Omega$ of $\tau_x(\varepsilon)$. We shall sometimes abuse terminology by dropping the dependence on ε from the mixing time.

In this paper we will bound the mixing time using *couplings*. By a *coupling* of a Markov chain, we mean a joint distribution $(X_t, Y_t)_{t\geq 0}$ such that the two random processes $(X_t)_{t\geq 0}$ and $(Y_t)_{t\geq 0}$, considered separately, each obey the transition rule for the given chain. In addition, if $X_t = Y_t$ then we require $X_{t'} = Y_{t'}$ for all $t' \geq t$. One way of defining such a coupling is to specify a suitable transition matrix indexed by the product space $\Omega \times \Omega$, thereby defining a Markov chain with this state space. As long as the two marginal transition probabilities agree with the original Markov chain, this defines a coupling, often referred to as a "Markovian coupling." However, in general, couplings are not required to be Markovian, and in fact, even conditioned on the previous states X_{t-1}, Y_{t-1}, it is perfectly possible for the state X_t to be correlated nontrivially with the sequence of states $Y_0, \ldots, Y_{t-2}$ (as will be the case for the coupling we define in Section 4).

When we speak of couplings in the present paper, we will always mean that a class of couplings has been defined, one for each possible initial pair of states (X_0, Y_0). We say that the coupling has *coalesced by time* t if the event $\{X_t = Y_t\}$ occurs. The following theorem relates the mixing time to the worst-case time until coalescence, and dates back to work of Doeblin in the 1930's (see [18]).

Theorem 2.1 (Coupling Theorem). *Let $(X_t, Y_t)_{t\geq 0}$ be any coupling of a Markov chain on state space Ω, and define*

$$\tau_{\text{couple}}(\varepsilon) = \max_{(X_0,Y_0)\in\Omega\times\Omega} \min\{t\colon\ \Pr[X_t \neq Y_t] \leq \varepsilon\}.$$

Then, for every $\varepsilon > 0$, $\tau(\varepsilon) \leq \tau_{\text{couple}}(\varepsilon)$.

Liftings. Let $\mathcal{M}$ and $\widehat{\mathcal{M}}$ be Markov chains on finite state spaces Ω, $\widehat{\Omega}$ respectively. We use Q, π to denote the flows and stationary distribution of $\mathcal{M}$, and $\widehat{Q}, \widehat{\pi}$ for the same quantities in $\widehat{\mathcal{M}}$.

We say that $\widehat{\mathcal{M}}$ is a *lifting* of $\mathcal{M}$ if there is a function $f : \widehat{\Omega} \to \Omega$ such that

$$Q_{(u,v)} = \sum_{x\in f^{-1}(u),\ y\in f^{-1}(v)} \widehat{Q}_{(x,y)} \qquad \text{for all } u, v \in \Omega. \tag{1}$$

Informally, if we "collapse" $\widehat{\mathcal{M}}$ by merging into a single state all states that have the same image under f, and aggregate the flows between these merged states, then we obtain precisely the chain $\mathcal{M}$. Note that equation (1) can be viewed as a homomorphism between flows. An immediate consequence of (1) is that $\pi(v) = \sum_{x\in f^{-1}(v)} \widehat{\pi}(x)$ for all $v \in \Omega$. The reader may wish to verify that the construction in Fig. 1 is indeed a valid lifting. We observe that our definition of lifting based on flows makes it particularly easy to design liftings for a given Markov chain (cf. the equivalent definition given in [2]).

Note that $\widehat{\mathcal{M}}$ may be non-reversible even when $\mathcal{M}$ is reversible. Indeed, as Chen et al. [2] show, to substantially speed up a reversible chain one must consider non-reversible liftings. (Note that the lifting in Fig. 1(b) is non-reversible.)

2.2 Approximate counting and leaf sampling

Framework. Let $T = (V, E)$ be a binary[4] tree with root r, all of whose leaves are at the same depth d. As discussed in the Introduction, our goal is to sample a leaf of T u.a.r. We think of T as being very large, so we want an algorithm that is polynomial in the *depth* d of T.

For each node v, let N_v denote the number of leaves in the subtree rooted at v. (Thus $N := N_r$ is the total number of leaves of T, and $N_v = 1$ for each leaf v.) We are given an estimate $\widetilde{N}_v$ of each N_v satisfying

$$(1+\delta)^{-1} N_v \leq \widetilde{N}_v \leq (1+\delta) N_v, \tag{2}$$

and $\widetilde{N}_v = N_v = 1$ for leaves v.[5]

Throughout the paper, unless otherwise stated, we will assume that δ lies in the range $[\frac{1}{d}, 1]$. The case when $\delta = O(1/d)$ is of little interest, since in this case, as noted

[4] The assumption that the tree is binary is made for simplicity of presentation only.

[5] Note that it is not necessary to know the structure of T a priori: since (2) implies that $N_v = 0$ (the subtree below v is empty) iff $\widetilde{N}_v = 0$, we can actually infer the structure of T locally from the estimates $\widetilde{N}_v$ for all vertices v.

in the Introduction, there is a simple linear time sampling algorithm based on rejection sampling. On the other hand, for larger values, $\delta = \Omega(1)$, our lifting construction cannot offer more than a constant factor speedup over the original Sinclair-Jerrum Markov chain, which we now describe.

The Sinclair-Jerrum chain. Sinclair and Jerrum [26] proposed a reversible Markov chain for sampling leaves from a uniform distribution in polynomial time, even when δ is an arbitrarily large constant (or indeed polynomially large in d). We specify the chain by giving the flows Q_e on each edge of T. We set $Q_e = \widetilde{N}_v$, where v is the lower endpoint of e. Additionally we introduce at each non-leaf node a self-loop of weight $Q_{(v,v)}$ equal to the total weight of the other edges incident at v, and at each leaf v a self-loop of weight $Q_{(v,v)} = 4d - 1$. (Thus the self-loop probabilities are $\frac{1}{2}$ for non-leaves and $1 - \frac{1}{4d}$ for leaves.) The self-loops of $\frac{1}{2}$ are a standard device to make the chain aperiodic (the resulting chain is usually called "lazy"). The large self-loops at the leaves are included to ensure that the stationary distribution puts large weight on the leaves[6].

As discussed above, the stationary distribution is given by $\pi(v) \propto W_v$, where $W_v := \sum_{e \ni v} Q_e$ is the sum of the edge weights incident at v. Now for any non-leaf node v, since $W_v = 2(\widetilde{N}_v + \sum_{u \text{ a child of } v} \widetilde{N}_u)$ we have $W_v \in [4(1+\delta)^{-1} N_v, 4(1+\delta) N_v]$. And for any leaf v we have $W_v = 4d$. This implies the following properties of the stationary distribution π:

1. π is uniform over the leaves.
2. $\sum_{v \text{ a leaf}} \pi(v) \geq \frac{1}{2+\delta}$. [To see this, note that the sum of W_v over all nodes v in any level above the leaves is at most $4(1+\delta)\sum_v N_v = 4(1+\delta)N$, while the sum of W_v over leaves is $4dN$.]

Therefore, we can sample leaves as follows. Simulate the Markov chain, starting from the root, until the distribution is close to π. If the final node is a leaf then output it, else fail and repeat. This gives us an almost uniformly distributed leaf (within any desired variation distance ε) in expected time $O(\tau_r(\varepsilon))$, where $\tau_r(\varepsilon)$ is the mixing time starting from the root r. The following theorem, which is a slightly improved version of the original result of Sinclair and Jerrum [26], bounds the mixing time. A proof is given in the full version of the paper.

Theorem 2.2. *For any $\delta \geq 0$, the mixing time of the Sinclair-Jerrum chain starting from the root satisfies $\tau_r(\varepsilon) = O(d^2(1+\delta)^2 \log(d\varepsilon^{-1}))$.*

Thus, for δ bounded by a constant (which is our range of interest in this paper), the mixing time is $\widetilde{O}(d^2)$. (The Theorem actually also shows that the mixing time remains polynomial for any $\delta \leq \mathrm{poly}(d)$.)

We note that a lower bound of $\Omega(d^2)$ follows easily, even in the case where the counting estimates are all exact (i.e., $\widetilde{N}_v = N_v \; \forall v$), since the height of the walking particle then behaves like symmetric random walk on $[0, d]$. Our main goal in this paper is to give a lifting that improves the above mixing time to $O(\delta d^2)$, thus beating the $\Omega(d^2)$ lower bound for all $\delta = o(1)$.

[6] The construction in [26] did not include these large self-loops; this simple modification actually leads to greater efficiency, since without it a leaf is sampled only with probability $O(\frac{1}{d})$, leading to a factor $O(d)$ overhead in the time to output a leaf.

3 The lifted chain

We will define a non-reversible lifted Markov chain having exactly two states for every node of the tree, with the exception of the root which will only have one lifted state. Roughly speaking, one set of these nodes correspond to "particles with downward momentum," and the others to "particles with upward momentum." The root and the leaves are exceptions. In the case of the root there is no need for the "upward" copy, so we retain just a single root node. In the case of a leaf we correspondingly have no need for a "downward" copy; however, we do need a second copy to act as a "trap" node, whose purpose will be to give the distribution of the departure time from the leaf a heavier tail than that provided by the self-loop in the original chain. We describe our construction in three steps:

Step 1: Lazy edges become 4-cycles. Let e be any non-loop edge in the original tree, joining nodes v, w, and with bidirectional flow Q_e through it. In the lifted chain, the original node v corresponds to two nodes, v^+ and v^-, and likewise for w. Suppose v is the parent of w. The new chain has a directed 4-cycle, (v^+, w^+, w^-, v^-), with each of the four directed edges carrying flow Q_e. Under the "projection" sending $v^+, v^- \mapsto v$ and $w^+, w^- \mapsto w$, this directed 4-cycle maps down to the original bidirectional flow Q_e on edge e, plus self-loops at v and w, each also of flow Q_e. Note that Q_e is exactly the contribution of edge e to the lazy self-loops at v, w in the original chain. (See Fig. 2(a).)

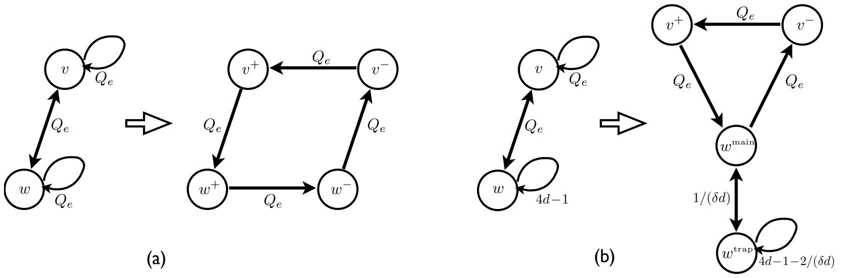

Fig. 2. (a) Lifting of an internal edge $\{v, w\}$. (b) Lifting of a leaf node w.

Applying the above construction to every non-loop edge e in the original tree yields a directed flow which exactly projects back onto the original undirected flow, with the sole exception of the large self-loops on the leaves.

Before proceeding, we first modify the above construction slightly. In the case when w is a leaf and $e = \{v, w\}$ is the edge joining it to its parent, our lifted flow looks slightly different. In this case, the self-loop of flow Q_e at v, and the bidirectional flow Q_e on e lift to a directed 3-cycle, $(v^+, w^{\text{main}}, v^-)$, with each of the three edges carrying flow Q_e. Similarly, in the case when $v = r$ is the root and $e = \{v, w\}$ is the edge joining it to one of its children, the self-loops of flow Q_e at v and w, and the bidirected flow Q_e on e lift to a directed 3-cycle, (r, w^+, w^-), plus a self-loop at r, with each of these four edges carrying flow Q_e.

Step 2: Set traps at the leaves. Let w be any leaf of the original tree. In the lifted chain, there will be two nodes, w^{main} and w^{trap}, corresponding to w. We next describe the lifted version of the self-loop of flow $4d - 1$ at w. This consists of a self-loop of flow $4d - 1 - \frac{2}{\delta d}$ at w^{trap}, together with a bidirectional flow of $1/(\delta d)$ between w^{main} and w^{trap}. (See Fig. 2(b).)

Step 3: Cancel the crossing edges. After the above two steps we have a lifted flow which projects down onto the original flow. However, in order to avoid the diffusive behavior of the original Markov chain, we need to reduce the "crossing flows" between nodes v^+ and v^-. We do this in a systematic way which preserves the projection onto the original flow. Let v be a non-leaf node in the tree, with flow Q_{up} to its parent and aggregated flow Q_{down} to its children. Then, as described in Step 1, we have crossing flows of value Q_{up} from v^+ to v^-, and Q_{down} from v^- to v^+.

Let $Q_{\min} = \min\{Q_{\mathrm{up}}, Q_{\mathrm{down}}\}$. We now cancel $Q_{\min}$ of the crossing flow in each direction, replacing it with self-loops at v^+ and v^-, each of flow $Q_{\min}$. This leaves us with crossing flow in just one direction, of value $|Q_{\mathrm{up}} - Q_{\mathrm{down}}|$. Note that this modification does not violate the flow condition, nor does it change the projection onto the original Markov chain. (See Fig. 3.)

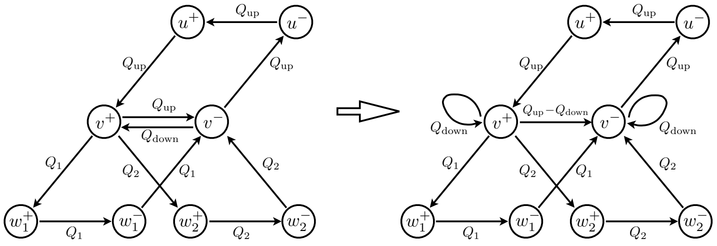

Fig. 3. Cancelling the crossing edges. Here $Q_{\mathrm{down}} = Q_1 + Q_2$, where Q_1, Q_2 are the flows between node v and its two children. The diagram assumes that $Q_{\mathrm{up}} \geq Q_{\mathrm{down}}$.

Since each step of the above construction preserves the lifting condition (1), the resulting chain is indeed a lifting of the Sinclair-Jerrum chain defined in Section 2.2. An immediate consequence is that the stationary distribution $\widehat{\pi}$ is the pull-back of π along the projection, and hence has the desired properties on the lifted copies of the leaves, namely that it projects down to a uniform distribution over the leaves having probability mass at least $1/(2+\delta)$. Thus, in order to use our lifted chain as an improved sampler of leaves, all that remains is to prove that its mixing time is faster than the original undirected chain.

4 Analysis of the lifted chain

In this section we prove our main result, which is the following bound on the mixing time of the lifted chain of the previous section. As we have noted earlier, the mixing time overcomes the $\Omega(d^2)$ diffusion lower bound for the original chain for all $\delta = o(1)$, and interpolates smoothly between the trivial $O(d)$ rejection sampling algorithm for $\delta = O(\frac{1}{d})$ and the original $\widetilde{O}(d^2)$ Sinclair-Jerrum algorithm for $\delta = \Omega(1)$.

Theorem 4.1. *For any $\delta \in [\frac{1}{d}, 1]$, the mixing time of the lifted Markov chain defined in Section 3 satisfies $\tau(\varepsilon) = O(\delta d^2 \log(1/\varepsilon))$.*

Proof. We proceed by constructing a non-Markovian coupling for the lifted chain. Let $X_0 \neq Y_0$ be arbitrary states of this lifted chain. We will define a coupled joint evolution $(X_t, Y_t)_{t \geq 0}$ in such a way that each of (X_t) and (Y_t), considered separately, obeys the law of our lifted Markov chain. We will do this in three asynchronous stages. First, let (X_t), (Y_t) each run independently until reaching the root, r, at times ρ_X, ρ_Y, respectively.

Subsequently, let both (X_t) and (Y_t) follow the same trajectory until they reach a "leaf trap" node, at respective times σ_X and $\sigma_Y = \sigma_X + (\rho_Y - \rho_X)$. Since ρ_X may not equal ρ_Y, this portion of the coupling is non-Markovian.

The third stage is empty for whichever chain had reached the root later, and lasts for $|\rho_X - \rho_Y|$ steps for the chain that reached the root earlier. This means that at the end of the third stage, the same (random) number of time steps will have elapsed for both chains. Also note that, since both chains begin stage 3 at the same leaf trap node, there is at least a probability of

$$(1 - 1/((\delta d)(4d - 1 - \delta d)))^{|\rho_X - \rho_Y|} = \exp(-O(|\rho_X - \rho_Y|/(\delta d^2))) \qquad (3)$$

that both chains remain at this node throughout stage 3, and have therefore coalesced by the end. If not, we can simply start over again with the first stage.

Our analysis of this coupling rests on two lemmas.

Lemma 4.2. *There exists an absolute constant C such that, from any initial node $X_0 = v$, the expected hitting time from v to the root is $\leq C\delta d^2$.*

Proof. We split the proof into three cases, according to whether v is a downward node, a leaf node, or an upward node. (In the case when v is the root, the hitting time is 0.)
Case 1: $v = w^+$ is a downward node. Since v is a downward node, every non-self-loop move either increases the depth by 1, or crosses to a rootward-oriented node. Hence, since the self-loop probability at v is at most $1/2$, in expected time at most $2d = O(\delta d^2)$ we will reach one of the other two cases. Thus, it suffices to handle cases 2 and 3.
Case 2: $v = w^{\text{main}}$ or $v = w^{\text{trap}}$ is a leaf node. Let u denote the parent of w in the original tree. Now, in our lifted chain, starting from $X_0 = v$, the first node reached by X_t that is not in $\{w^{\text{main}}, w^{\text{trap}}\}$ must be u^-. What is the hitting time to u^-? Solving a system of two linear equations in two unknowns, we find that this hitting time is $4d$ when starting from w^{main}, and $(4\delta d^2 - \delta d - 1 + 4d)$ when starting from w^{trap}. Since in both cases this is $O(\delta d^2)$, and u^- is an upward node, it thus suffices to handle case 3.
Case 3: $v = w^-$ is an upward node. Let u be the parent of w in the original tree. As in case 2, note that, starting from $X_0 = w^-$, the first node that will be reached by X_t that does not project into the subtree rooted at w must be u^-.

Let $Q_{\text{up}} = \widetilde{N}_w$ denote the flow up from w, and Q_{down} the aggregated flow down from w to its children. When $Q_{\text{up}} \geq Q_{\text{down}}$ (as in Fig. 3), the only edges out from w^- are a self-loop and the edge (w^-, u^-), so it is easy to calculate that the expected hitting time from w^- to u^- equals $1 + Q_{\text{down}}/Q_{\text{up}}$, which is at most $1 + (1 + \delta)^2 = O(1)$.

Claim. Suppose $Q_{\text{up}} < Q_{\text{down}}$. Let H denote the hitting time from w^- to u^-. Then $\mathbb{E}(H) = O(\delta d)$.

Assuming the Claim is true, we have shown that the expected hitting time from w^- to u^- is always $O(\delta d)$. It follows by induction that the hitting time from w^- to r is $O(\delta d^2)$, since the depth of w is at most d, which completes our analysis of case 3 and the proof of the lemma.

All that remains is to prove the Claim. To see this, consider what happens to our lifted walk if we re-route the flow on the edge (w^-, u^-) to instead go along the edge (w^-, w^+). In this case, starting from w^-, we can never leave the subtree rooted at w, and in fact the random walk is exactly the same as would be produced by our lifting construction applied just to the subtree rooted at w, except that the transition probabilities at the leaves are still based on d rather than on the height of the subtree below w. Let us compute the stationary probability of w^- in this modified chain.

Using the well-known fact that the stationary probability at any node is the reciprocal of the expected return time to that node, it follows that

$$\frac{1}{\widetilde{\pi}(w^-)} = 1 + \frac{Q_{\mathrm{down}}}{Q_{\mathrm{down}} + Q_{\mathrm{up}}}\mathbb{E}(H'), \tag{4}$$

where H' is the hitting time from w^+ to w^-, and $\widetilde{\pi}$ is the stationary distribution for the modified lifted chain rooted at w. A straightforward calculation yields $\widetilde{\pi}(w^-) \geq 1/((1+\delta)^2(2d+i)) \geq 1/(4d)$, where i is the height of node w, whence by (4) it follows that $\mathbb{E}(H') \leq 2(4d-1)$.

Returning now to the full lifted chain, since from w^- the flows out are Q_{up} to u^-, Q_{up} in a self-loop, and $Q_{\mathrm{down}} - Q_{\mathrm{up}}$ to w^+, it follows that

$$\mathbb{E}(H) = 1 + \frac{Q_{\mathrm{up}}}{Q_{\mathrm{down}} + Q_{\mathrm{up}}}\,\mathbb{E}(H) + \frac{Q_{\mathrm{down}} - Q_{\mathrm{up}}}{Q_{\mathrm{down}} + Q_{\mathrm{up}}}\,\mathbb{E}(H'),$$

which implies

$$\mathbb{E}(H) = 1 + \frac{Q_{\mathrm{up}}}{Q_{\mathrm{down}}} + \left(\frac{Q_{\mathrm{down}} - Q_{\mathrm{up}}}{Q_{\mathrm{down}}}\right)\mathbb{E}(H') \leq 2 + 2P_{\mathrm{cross}}(w^-)\mathbb{E}(H'),$$

where $P_{\mathrm{cross}}(w^-) = O(\delta)$ is the transition probability from w^- to w^+ in the lifted chain. Since we already know that $\mathbb{E}(H') = O(d)$, it follows that $\mathbb{E}(H) \leq 2 + O(\delta d) = O(\delta d)$. This concludes the proof of the Claim, and of Lemma 4.2. □

Lemma 4.3. *There exists an absolute constant C' such that the expected hitting time from the root to the set of "leaf trap" nodes is $\leq C'\delta d^2$.*

Proof. Consider an infinite run of the Markov chain, and partition the positive integers into epochs, where the even epochs end at the first time (after they start) that the root is reached, and the odd epochs end at the first time (after they start) that a leaf trap is reached. Let us denote by L the set of all leaf trap nodes. Since no leaf traps are visited during the odd epochs, the fraction of time in even epochs is at least $\pi(L)$. But the average length of an even epoch is at most $C\delta d^2$, by Lemma 4.2. Hence the average length of an odd epoch must be at most $C\delta d^2/\pi(L)$, which is $O(\delta d^2)$ since we arranged for $\pi(L) = \Theta(1)$. This concludes the proof, as the average length of an odd epoch equals the expected hitting time from the root to the set of leaf trap nodes. □

We now continue with the proof of Theorem 4.1. By Lemmas 4.2 and 4.3, the expected total length of stages 1 and 2 combined is $O(\delta d^2)$. Hence, by Markov's inequality, with probability at least $7/8$, the total length is at most eight times the expectation, which is $O(\delta d^2)$. An application of the triangle inequality implies that therefore $\mathbb{E}(|\rho_X - \rho_Y|) = O(\delta d^2)$ (where the O hides an explicit constant of moderate size). By Markov's inequality, it follows that with probability $\Omega(1)$, $|\rho_X - \rho_Y| = O(\delta d^2)$. By (3), the coupling has coalesced by the end of the third stage with probability $\Omega(1)$. Thus the chain coalesces within $O(\delta d^2 \log(1/\varepsilon))$ time steps with probability at least $1 - \varepsilon$. The corresponding bound on the mixing time follows from Theorem 2.1. □

5 Conclusions and future work

We have shown that non-reversible liftings can be used to speed up MCMC sampling (and hence also approximate counting) algorithms, even without the high degree of symmetry present in previous examples. Although it is still highly specialized, the class of Markov chains we consider, being random walks on trees with an approximation oracle for the number of leaves, is nevertheless natural in the context of computation, and encompasses many combinatorial problems with interesting and complex structure.

The first open question is whether our construction can be improved to reduce the mixing time of the lifted chain down to the asymptotically optimal value $O(\rho)$, where $\rho = O(d(1+\delta)^2)$ is the flow parameter for the original chain, while retaining the local character of the current construction which makes it a practical tool for sampling. In the case of large bias, $\delta = \Omega(1)$, our current lifting exhibits (potentially) nearly as much diffusive behavior as the unlifted chain; intuitively this happens because "excursions" upward or downward may typically be of length $1/\delta = O(1)$, as is the case for symmetric random walk. However, at least in the special case when the tree is a path, we have developed a more complex (yet still local) construction that eliminates this diffusive behavior to a large extent; the idea is to keep track of multiple momentum values (rather than just "up" and "down"). This will be discussed in the full version of the paper.

A second natural question is whether our techniques can be profitably applied to other Markov chains used in sampling algorithms. Prime candidates here are Markov chains for matchings [12, 13] and for sampling points in a convex body [6, 21]. The latter example seems particularly intriguing as there is a well-defined notion of "direction" along which momentum can be preserved. Indeed, we note that lifting ideas have already appeared, albeit not explicitly, in this example: the "hit-and-run" Markov chain [19], which at each step moves to a random point on a randomly chosen chord of the body through the current point, has the flavor of a "lifting" of the more local "ball walk"[20],[7] which moves to a random point within a ball centered at the current point. We conjecture that understanding this connection more formally within a lifting framework may illuminate previous work on random walks on convex bodies, and perhaps even lead to further algorithmic improvements.

References

1. M. BAYATI, J.-H. KIM and A. SABERI. A sequential algorithm for generating random graphs. *Proc. APPROX-RANDOM* 2007, pp. 326–340.

[7] See the full version for a more precise discussion of this point.

2. F. Chen, L. Lovász and I. Pak. Lifting Markov chains to speed up mixing. *Proc. 17th Annual ACM Symposium on Theory of Computing*, 1999, pp. 275–281.
3. S.R. Das and A. Sinclair. A Markov chain Monte Carlo method for derivative pricing and risk assessment. *J. Investment Management* **3** (2005), pp. 29–44.
4. P. Diaconis, S. Holmes and R. Neal. Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability* **10** (2000), pp. 726–752.
5. S. Duane, A. Kennedy, B. Pendleton and D. Roweth. Hybrid Monte Carlo. *Physics Letters B* **195** (1987), pp. 216–222.
6. M. Dyer, A. Frieze and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *JACM* **38** (1991), pp. 1–17.
7. M. Hildebrand. Rates of convergence of the Diaconis-Holmes-Neal Markov chain sampler with a V-shaped stationary probability. *Markov Proc. Rel. Fields* **10** (2004), pp. 687–704.
8. M.Hildebrand. Analysis of the Diaconis-Holmes-Neal Markov chain sampler for log-concave probabilities. Preprint, 2002. Available from http://nyjm.albany.edu:8000/∼martinhi/preprints.html
9. A.M. Horowitz. A generalized guided Monte Carlo algorithm. *Physics Letters B* **268** (1991), pp. 247–252.
10. S. Janson. The probability that a random multigraph is simple. *Combinatorics, Probability and Computing* **18** (2009), pp. 205–225.
11. M. Jerrum and A. Sinclair. Fast uniform generation of regular graphs. *Theoretical Computer Science* **73** (1990), pp. 91–100.
12. M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing* **18** (1989), pp. 1149–1178.
13. M.Jerrum, A.Sinclair and E.Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *JACM* **51** (2004), pp. 671–697.
14. K. Jung, D. Shah and J. Shin. Distributed averaging via lifted Markov chains. Preprint, August 2009. Available at arxiv.org/pdf/0908.4073v1
15. K. Jung, D. Shah and J. Shin. Fast and slim lifted Markov chains. *Allerton Conference on Communication, Control and Computing*, 2007.
16. J.-H. Kim and V. Vu. Generating random regular graphs. *Proc. 21st Annual ACM Symposium on Theory of Computing*, 2003, pp. 213–222.
17. D. Knuth. Estimating the efficiency of backtrack programs. *Mathematics of Computation* **29** (1975), pp. 121–136.
18. T. Lindvall. *Lectures on the coupling method.* Dover, Mineola, NY, 2002.
19. L. Lovász. Hit-and-run mixes fast. *Mathematical Programming* **86** (1998), pp. 443–461.
20. L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms* **4** (1993), pp. 359–412.
21. L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Computer and System Sciences* **72** (2006), pp. 392–417.
22. B. McKay. Asymptotics for symmetric 0-1 matrices with prescribed row sums. *Ars Combinatorica* **19A** (1985), pp. 15–25.
23. B.D. McKay and N. Wormald. Asymptotic enumeration by degree sequence of graphs with degrees $o(n^{1/2})$. *Combinatorica* **11** (1991), pp. 369–382.
24. P. Rosenbaum. Sampling the leaves of a tree with equal probabilities. *Journal of the American Statistical Association* **88** (1993), pp. 1455–1457.
25. A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability and Computing* **1** (1992), pp. 351–370.
26. A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information & Computation* **82** (1989), pp. 93–133.
27. K. Turitsyn, M. Chertkov and M. Vucelja. Irreversible Monte Carlo algorithms for efficient sampling. Preprint, 2008. Available at arxiv.org/pdf/0809.0916v2