From the Bernoulli Factory to a Dice Enterprise via Perfect Sampling of Markov Chains

Giulio Morina and Krzysztof Łatuszyński

Department of Statistics
University of Warwick
Coventry, CV4 7AL - UK
e-mail: G.Morina@warwick.ac.uk; K.G.Latuszynski@warwick.ac.uk

Piotr Nayar

Institute of Mathematics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Banacha 2, 02-097 Warsaw - Poland
e-mail: P.Nayar@mimuw.edu.pl

Alex Wendland

Mathematics Institute
Zeeman Building
University of Warwick
Coventry, CV4 7AL UK
e-mail: a.p.wendland@gmail.com

Abstract: Given a p-coin that lands heads with unknown probability p, we wish to produce an $f(p)$-coin for a given function $f : (0,1) \to (0,1)$. This problem is commonly known as the Bernoulli Factory and results on its solvability and complexity have been obtained in [23, 29]. Nevertheless, generic ways to design a practical Bernoulli Factory for a given function f exist only in a few special cases. We present a constructive way to build an efficient Bernoulli Factory when $f(p)$ is a rational function with coefficients in $\mathbb{R}$. Moreover, we extend the Bernoulli Factory problem to a more general setting where we have access to an m-sided die and we wish to roll a v-sided one; i.e., we consider rational functions $f : \Delta^{m-1} \to \Delta^{v-1}$ between open probability simplices. Our construction consists of rephrasing the original problem as simulating from the stationary distribution of a certain class of Markov chains - a task that we show can be achieved using perfect simulation techniques with the original m-sided die as the only source of randomness. In the Bernoulli Factory case, the number of tosses needed by the algorithm has exponential tails and its expected value can be bounded uniformly in p. En route to optimizing the algorithm we show a fact of independent interest: every finite, integer valued, random variable will eventually become log-concave after convolving with enough Bernoulli trials.

1. Introduction

Back in 1951, Von Neumann [39] proposed a method to produce a fair coin out of a biased one. Since then, the problem has been generalized into finding an algorithm that given a p-coin — a coin that lands heads with unknown probability p — can produce an $f(p)$-coin for a given function $f : \mathcal{D} \subseteq (0,1) \to (0,1)$. Keane and O'Brien [23] referred to this problem as the Bernoulli Factory and, motivated by problems in regenerative steady-state simulations [1, 14], identified the class of functions f for which it is solvable. Since then, other studies have been carried out to provide ways of constructing and analysing the Bernoulli Factory algorithms [17, 18, 19, 24, 26, 28, 29] as well as extending it to quantum settings [6, 31, 40] and specialised multivariate scenarios [7, 19]. Relations between the Bernnoulli Factory and other fundamental simulation questions in statistics and computer science have been explored in [21] and [9], respectively. More recently, Bernoulli Factory techniques have been successfully applied to perform exact simulations of diffusions [3, 24], develop perfect simulation algorithms [2, 10, 25], design MCMC algorithms that can tackle intractable likelihood models and perform Bayesian inference [11, 12, 15, 38], design particle filters in scenarios where weights are not available analytically [37], and also to reductions in mechanism design [5, 7, 30].

Nevertheless, designing a Bernoulli Factory algorithm for a given function f is still challenging. The strategy described in [29] can be generally applied for any real analytic function f, but the combinatorial complexity of the implementation is prohibitive. When combined with the reverse time martingale approach of [24], the implementation becomes feasible, but the running time depends on the speed of convergence of certain Bernstein polynomial envelopes and is often impractical. Specialised fast algorithms are available for linear functions [18, 19], under additional assumptions on the domain, or functions admitting specific series expansions [13, 24, 26]. However, for other classes of functions constructing a Bernoulli Factory is generally hard and even when an algorithm is available, its running time may be prohibitive. Moreover, the problem of extending the classic Bernoulli Factory setting to a multivariable one — that is producing rolls of a die given an arbitrary other one — has not been systematically studied.

In this paper we provide a novel constructive way to design a Bernoulli Factory for rational functions f with coefficients in $\mathbb{R}$. Our construction can be applied to rational functions mapping between probability simplices

$$f : \Delta^m \to \Delta^v, \qquad m, v \geq 1, \tag{1}$$

thus generalizing the classic Bernoulli Factory to a *Dice Enterprise*.

Our approach relies on rephrasing the original problem as sampling from the stationary distribution of a suitably designed Markov chain. This is achieved by first decomposing the given rational function in a fashion insipred by [28], but extended to take into account coefficients in $\mathbb{R}$ and multivariate scenarios. Then, the Markov chain is constructed so that its evolution can be simulated by just rolling the original die. Perfect simulations techniques, such as Coupling From

The Past (CFTP) [35] or Fill's interruptible algorithm [8], can then be employed to get a sample distributed precisely as its stationary distribution. Moreover, for $m = 1$ in (1), that includes the classic Bernoulli Factory setting $m = v = 1$ as a special case, a monotonic version of CFTP is proposed, improving the efficiency of implementation. Under this scenario, we show that the method has a "fast simulation" (i.e., the required number of tosses has exponentially decaying tail probabilities). To prove the result we demonstrate a fact of wider interest: the convolution of a Bin $\left(n, \frac{1}{2}\right)$ variable with any finite, integer valued random variable is log-concave when n is big enough.

The paper is organised as follows:

Section 2 introduces the notation and notions that will be used throughout the paper. In particular, it gives a brief introduction to CFTP and introduces ladders, a class of discrete probability distributions that are suitable as candidates for stationary distributions of Markov chains used in the sequel.

Section 3 develops the Dice Enterprise for rational functions $f : \Delta^m \to \Delta^v$, thus generalising the usual Bernoulli Factory setting. We first show how this problem can be rephrased as sampling from a ladder and construct a CFTP algorithm that performs it. We prove that our proposed construction is optimal in terms of Peskun's ordering. We then analyse the efficiency of the proposed algorithm in terms of the expected number of required rolls of the original die and notice that it is always finite under suitable assumptions. In the "coin to dice" scenario (of which the Bernoulli Factory is a special case), we notice that for log-concave ladders the tail probability of the number of tosses decays exponentially fast. We then prove that it is always possible to construct such log-concave distribution.

Section 4 presents an `R` package that implements the developed method and explicative examples, validating the developed theory. In particular, we also show how a Dice Enterprise can be used to deal with m independent coins and reproduce examples taken from [7, 12, 19].

Proofs of all the results are presented in Appendix.

2. Notation and Preliminaries

Define the open m dimensional probability simplex as

$$\Delta^m = \left\{ \boldsymbol{p} = (p_0, \ldots, p_m) \in (0,1)^{m+1} : \sum_{i=0}^{m} p_i = 1 \right\}$$

and by $\bar{\Delta}^m$ denote its closure. For $b \in \{0, \ldots, m\}$, by $e_b \in \bar{\Delta}^m$ denote the b^{th} standard unit vector, i.e. a vector of zeros with a 1 in the b^{th} position. We let the first element of a vector have index 0 and we interchangeably use $(p_0, p_1) \in \Delta^1$ and $p \in (0, 1)$, identifying p as p_1. We shall write $X \sim \boldsymbol{p}$ to denote that X is a draw from the categorical distribution with parameter $\boldsymbol{p} \in \Delta^m$ on $\Omega = \{0, \ldots, m\}$. If the vector $\boldsymbol{p}$ is not known explicitly, but there is a mechanism to sample $X \sim \boldsymbol{p}$ (e.g. via experiment or computer code), we call this mechanism a black box to sample from $\boldsymbol{p} \in \Delta^m$. Alternatively, if we want to stress that the vector $\boldsymbol{\mu} \in \Delta^n$ is given explicitly, we refer to it as known distribution $\boldsymbol{\mu}$.

Given a rational function $f : \Delta^m \to \Delta^v$, in Section 3 we will construct a new discrete probability distribution $\pi : \Delta^m \to \Delta^k$, named ladder, such that a draw from $f(\boldsymbol{p})$ can be transformed into a draw from $\pi(\boldsymbol{p})$ and vice-versa. To this end, consider pairs of distributions related by disaggregation defined as follows:

Definition 2.1 (Disaggregation)**.** Let $\boldsymbol{\mu} = (\mu_0, \ldots, \mu_k)$ and $\boldsymbol{\nu} = (\nu_0, \ldots, \nu_v)$ be probability distributions on Δ^k and Δ^v respectively with $v \leq k$. We say that $\boldsymbol{\mu}$ is a *disaggregation* of $\boldsymbol{\nu}$ if there exists a partition of $\{0, \ldots, k\}$ into $v + 1$ sets $A_0, A_2, \ldots, A_v$ such that

$$\nu_i = \sum_{j \in A_i} \mu_j, \qquad \text{for all } i \in \{0, \ldots, v\}.$$

If $\boldsymbol{\mu}$ is a disaggregation of $\boldsymbol{\nu}$, then we shall equivalently say that $\boldsymbol{\nu}$ is an *aggregation* of $\boldsymbol{\mu}$.

Remark 2.2. If $\boldsymbol{\mu} = (\mu_0, \ldots, \mu_k)$ is a disaggregation of $\boldsymbol{\nu} = (\nu_0, \ldots, \nu_v)$ then, sampling from $\boldsymbol{\mu}$ is equivalent to sampling from $\boldsymbol{\nu}$ in the following sense:
Given $X \sim \boldsymbol{\mu}$, define Y as $Y := i$ if $X \in A_i$. Then $Y \sim \boldsymbol{\nu}$.
Given $X \sim \boldsymbol{\nu}$, define Y by letting

$$\mathbb{P}(Y = i) = \mathbb{I}(i \in A_X) \frac{\mu_i}{\sum_{j \in A_X} \mu_j}. \tag{2}$$

Then $Y \sim \boldsymbol{\mu}$.

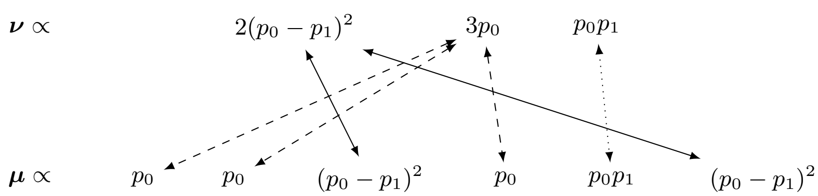


Figure 1: Disaggregation. A sample from $\boldsymbol{\mu}$ is directly mapped to a sample from $\boldsymbol{\nu}$. A sample from $\boldsymbol{\nu}$ can be mapped to $\boldsymbol{\mu}$ proportionally.

2.1. Sampling from known categorical distributions

Sampling from a known distribution $\boldsymbol{\mu} = (\mu_0, \ldots, \mu_k) \in \Delta^k$ is usually done by sampling $U \sim \text{Unif}(0, 1)$ and setting

$$Z = i \quad \text{if} \quad \sum_{j=0}^{i-1} \mu_j < U \leq \sum_{j=0}^{i} \mu_j.$$

In the spirit of [24, 29], we can consider B, the binary representation of U and notice that this is an iid sequence of Bern(1/2). Let $B_{1:l}$ denote the first l bits of U and let $(B_{1:l})_{10}$ be its representation in base 10. Clearly $(B_{1:l})_{10} \leq U \leq (B_{1:l})_{10} + 2^{-l}$, so that we could set

$$Z = i \qquad \text{if} \qquad \sum_{j=0}^{i-1} \mu_j \leq (B_{1:l})_{10} \quad \text{and} \quad (B_{1:l})_{10} + 2^{-l} \leq \sum_{j=0}^{i} \mu_j$$

where l is big enough so that there exists an i such that the condition above is satisfied.

Therefore, we do not need to have access to a generator of uniform random variables to sample from a categorical distribution — it is enough to obtain a sequence of independent tosses of a fair coin. Algorithm 1 is a variation of Von Neumann's algorithm [39] that outputs a fair coin given access to an iid sequence of rolls of an arbitrary die.

Algorithm 1 Fair coins from a die

Input: black box to sample from $\boldsymbol{p} \in \Delta^m$.
Output: a sample from Bern(1/2).

1: Sample $X_1, X_2 \overset{iid}{\sim} \boldsymbol{p}$
2: **if** $X_1 < X_2$ **then** set $Y := 0$
3: **else if** $X_1 > X_2$ **then** set $Y := 1$
4: **else if** $X_1 = X_2$ **then** discard X_1, X_2 and GOTO 1
5: **end if**
6: **Output** Y

Consequently, given a black box to sample from an unknown distribution $\boldsymbol{p} \in \Delta^m$, Algorithm 2 outputs a sample from a known distribution $\boldsymbol{\mu} \in \Delta^k$. In particular, if $\boldsymbol{\mu}$, is a disaggregation of $\boldsymbol{\nu}$, Algorithm 2 can be used to obtain a sample $Y \sim \boldsymbol{\mu}$ given $X \sim \boldsymbol{\nu}$ as in equation (2).

Algorithm 2 Categorical distribution from a die

Input: black box to sample from $\boldsymbol{p} \in \Delta^m$.
Output: a sample from a known distribution $\boldsymbol{\mu} \in \Delta^k$.

1: Set $l := 1$
2: Sample $Y \sim \text{Bern}(1/2)$ using Algorithm 1
3: Set $B_{1:l} = B_{1:l-1}|Y$
4: **if** there exists an $i \in \{0, \ldots, k\}$ such that

$$\sum_{j=0}^{i-1} \mu_j \leq (B_{1:l})_{10} < (B_{1:l})_{10} + 2^{-l} \leq \sum_{j=0}^{i} \mu_j$$

then set $Z := i$
5: **else**
6: Set $l = l + 1$ and GOTO 2
7: **end if**
8: **Output** Z

Remark 2.3. Let L be the random number of loops of Algorithm 2 before terminating. The computation verifies that $\mathbb{P}(L > l) \leq k2^{-l}$.

To ease the notation, from now on we shall assume that a generator of uniform random variables is available – as common in applications – since we demonstrated that this assumption is not restrictive from the theoretical viewpoint.

2.2. Coupling from the past

Perfect sampling is a well developed approach [20] to devise specialised algorithms, necessarily with random running time, that will output a single draw exactly from the stationary distribution of a Markov chain, rather than from its approximation. Coupling From the Past (CFTP) [35] is a pioneering technique in this field and illustrative for our purposes. The idea behind the method relies on starting the chain at time $-\infty$, so that at present time one would have a sample from the stationary distribution. This may not seem practical, but as pointed out in [35], one can make use of coupled chains to decide when to stop tracking the chain back in time. In practice, it is convenient to introduce an update function for the chain. Given a state i and a source of randomness, the update function returns the state of the chain at the next step. In case of a Markov chain dynamics driven by a die with $(m+1)$ faces, we define the update function as follows.

Definition 2.4 (Update function)**.** Let $(X_t)_{t\in\mathbb{N}}$ be a Markov chain on $\Omega = \{0, 1, \dots, k\}$. Assume $\boldsymbol{p} \in \Delta^m$ and let $B \sim \boldsymbol{p}$ and $U \sim \text{Unif}(0, 1)$. A function

$$\phi : \Omega \times \{0, 1, \dots, m\} \times [0, 1] \to \Omega$$

is an *update function* for the Markov chain $(X_t)_{t\in\mathbb{N}}$ if

$$\mathbb{P}(X_{t+1} = j|X_t = i) = \mathbb{P}(\phi(i, B, U) = j), \qquad \forall i, j \in \Omega.$$

We will write $\phi_t(x, \boldsymbol{B}, \boldsymbol{U}) = \phi(\phi(\dots(\phi(x, B_1, U_1), B_2, U_2), \dots), B_t, U_t)$ to indicate the state of the chain after t steps when starting from x.

Given an update function ϕ, CFTP is implementable via Algorithm 3.

Algorithm 3 Coupling From the Past

Input: an update function ϕ for a Markov chain $(X_t)_{t\in\mathbb{N}}$ on $\Omega = \{0, \dots, k\}$ with unique stationary distribution π; a black box to sample from $\boldsymbol{p} \in \Delta^m$.

Output: A sample from π.

Set $T \leftarrow 1$
for $i = 0, \dots, k$ **do** $X_0^{(i)} \leftarrow i$ **end for**
repeat
  Sample independently $B_{-T} \sim \boldsymbol{p}$ and $U_{-T} \sim \text{Unif}(0, 1)$
  for $i = 0, \dots, k-1$ **do** $X_0^{(i)} \leftarrow \phi_T(i, (B_{-T}, \dots, B_{-1}), (U_{-T}, \dots, U_{-1}))$ **end for**
  Set $T \leftarrow T + 1$
until $X_0^{(0)} = X_0^{(1)} = \dots = X_0^{(k)}$
Output $X_0^{(0)}$

Notice that CFTP needs to keep track of the trajectories of k coupled chains. If k is large implementing the algorithm may become infeasible. A more efficient version of CFTP can be designed for monotonic Markov chains [35]. In particular, assume that the state space Ω of the Markov chain $(X_t)_{t\in\mathbb{N}}$ admits a partial order $\preceq$, and there exist the maximum and the minimum states, say 0 and k, respectively; i.e., $\forall j \in \Omega$, $j \preceq k$ and $0 \preceq j$. The monotonic update function is defined as follows.

Definition 2.5 (Monotonic update function)**.** An update function ϕ as in Definition 2.4 is *monotonic* if for all $B \in \{0, 1, \dots, m\}$ and $U \in [0, 1]$

$$i \preceq j \quad \Longrightarrow \quad \phi(i, B, U) \preceq \phi(j, B, U). \tag{3}$$

In the monotonic case it is enough to track coalescence of just two chains, started from the minimum and the maximum state. Algorithm 4 presents CFTP with monotonic update function ϕ.

Algorithm 4 Monotonic Coupling From the Past

Input: a monotonic update function ϕ for a Markov chain $(X_t)_{t\in\mathbb{N}}$ on $\Omega = \{0, \dots, k\}$ with minimum and maximum states, 0 and k respectively and with unique stationary distribution π; a black box to sample from $\boldsymbol{p} \in \Delta^m$.

Output: A sample from π.

1: Set $T \leftarrow 1$, $X_0 \leftarrow 0$ and $Y_0 \leftarrow k$
2: **repeat**
3: Sample independently $B_{-T} \sim \boldsymbol{p}$ and $U_{-T} \sim \text{Unif}(0, 1)$
4: Set $X_0 \leftarrow \phi_T(1, (B_{-T}, \dots, B_{-1}), (U_{-T}, \dots, U_{-1}))$
5: Set $Y_0 \leftarrow \phi_T(k, (B_{-T}, \dots, B_{-1}), (U_{-T}, \dots, U_{-1}))$
6: Set $T \leftarrow T + 1$
7: **until** $X_0 = Y_0$
8: **Output** X_0

2.3. Ladder over $\mathbb{R}$

We now introduce a class of probability distributions $\pi : \Delta^m \to \Delta^k$ that will be of main interest in the remainder of the paper. Recall that the 1-norm of a vector $\boldsymbol{a} = (a_0, \dots, a_n)$ is $\|\boldsymbol{a}\|_1 = \sum_{j=0}^n |a_j|$.

Definition 2.6 (Multivariate ladder)**.** For every $\boldsymbol{p} = (p_0, \dots, p_m) \in \Delta^m$ let $\pi(\boldsymbol{p}) = (\pi_0(\boldsymbol{p}), \dots, \pi_k(\boldsymbol{p}))$ be a probability distribution on $\Omega = \{0, \dots, k\}$. We say that $\pi(\boldsymbol{p})$ is a *multivariate ladder* over $\mathbb{R}$ if every π_i is of the form

$$\pi_i(\boldsymbol{p}) = R_i \frac{\prod_{j=0}^m p_j^{n_{i,j}}}{C(\boldsymbol{p})}, \tag{4}$$

where

- $C(\boldsymbol{p})$ is a polynomial with real coefficients that does not admit roots in $\bar{\Delta}^m$;
- $\forall i, j$, R_i is a strictly positive real constant and $n_{i,j} \in \mathbb{N}_{\geq 0}$;
- Denote $\boldsymbol{n}_i = (n_{i,0}, n_{i,1}, \dots, n_{i,m})$. There exists an integer d such that $\|\boldsymbol{n}_i\|_1 = d$ for all i. We will refer to $\boldsymbol{n}_i$ as the degree of $\pi_i(\boldsymbol{p})$ and to d as the degree of $\pi(\boldsymbol{p})$.

Moreover, we say that $\pi(\boldsymbol{p})$ is a *connected ladder* if

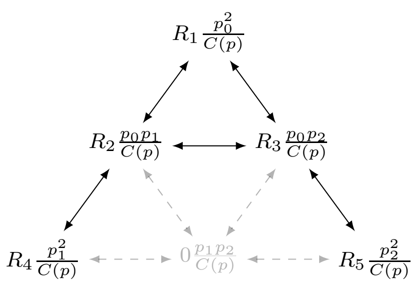
- Each $i, j \in \Omega$ are connected, meaning that there exists a sequence of states $(i = s_1, s_2, ..., s_t = j)$, such that $\left\|\boldsymbol{n}_{s_h} - \boldsymbol{n}_{s_{(h-1)}}\right\|_1 \leq 2$ for all $h \in \{2, \ldots, t\}$.

Finally, we say that $\pi(\boldsymbol{p})$ is a *fine ladder* if

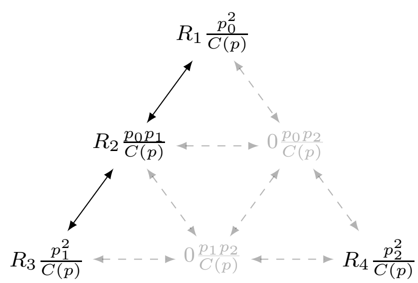
- $\boldsymbol{n}_i = \boldsymbol{n}_j$ implies $i = j$.

Figure 2 gives a graphical representation of a multivariate ladder and motivates the following concept of neighbourhood.

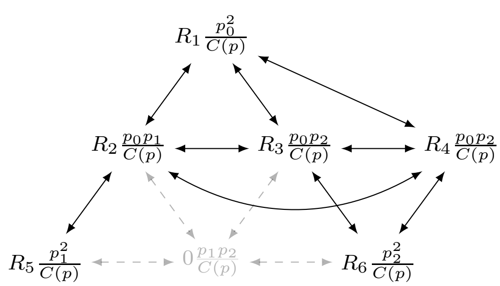
Definition 2.7 (Neighbourhoods on ladders)**.** On a multivariate ladder $\pi : \Delta^m \to \Delta^k$ define the *neighbourhood of* $i \in \Omega$ as $\mathcal{N}(i) = \{j \in \Omega \setminus \{i\} : \|\boldsymbol{n}_i - \boldsymbol{n}_j\| \leq 2\}$. Note that for connected ladders $\mathcal{N}(i)$ must have at least one element for each i (in the non-trivial case of $k > 1$).



(a) Fine and connected ladder
$\pi : \Delta^2 \to \Delta^4$



(b) Fine, but not connected ladder
$\pi : \Delta^2 \to \Delta^3$



(c) Connected, but not fine ladder
$\pi : \Delta^2 \to \Delta^5$

Figure 2: Multivariate ladders over $\mathbb{R}$. Edges represent connected states.

2.3.1. Operations on ladders

We now introduce three operations on ladders of which we will make extensive use: *increasing the degree* of a ladder, *thinning* and *augmenting* a ladder.

Definition 2.8 (Increasing the degree)**.** Let $\pi : \Delta^m \to \Delta^k$ be a multivariate ladder of degree d. *Increasing the degree of* π yields a new ladder $\pi' : \Delta^m \to \Delta^{(k+1)(m+1)-1}$ of degree $(d+1)$ with probabilities $\pi'_l(\boldsymbol{p})$ on $\Omega' = \{0, \ldots, (k+1)(m+1)-1\}$ of the form $\pi'_l(\boldsymbol{p}) := \pi_i(\boldsymbol{p})p_j$, where $i \in \{0, \ldots, k\}$ and $j \in \{0, \ldots, m\}$ are the unique solution of $l = i(m+1) + j$.

Increasing the degree corresponds to multiplication by $1 = \sum_{j=0}^{m} p_j$ and the resulting ladder $\pi'(\boldsymbol{p})$ is a disaggregation of $\pi(\boldsymbol{p})$. Indeed, let $A_0, \ldots, A_k$ be

$$A_i := \{i(m+1), \ldots, i(m+1) + m\}.$$

Then definition 2.1 is satisfied, since

$$\sum_{a \in A_i} \pi'_a(\boldsymbol{p}) = \sum_{j=0}^{m} \pi_i(\boldsymbol{p})p_j = \pi_i(\boldsymbol{p}) \sum_{j=0}^{m} p_j = \pi_i(\boldsymbol{p}).$$

Definition 2.9 (Thinning)**.** Let $\pi : \Delta^m \to \Delta^k$ be a multivariate ladder of degree d. *Thinning* π yields a fine ladder π' by joining all the states of π with the same monomial. Thus $\pi' : \Delta^m \to \Delta^w$ where $k \geq w := |\{\boldsymbol{n}_0, \ldots, \boldsymbol{n}_k\}|$, and by (4) the probabilities $\pi'_l(\boldsymbol{p})$ on $\Omega' = \{0, \ldots, w\}$ are of the form $\pi'_l(\boldsymbol{p}) := \frac{R'_l \boldsymbol{p}^{\boldsymbol{n}'_l}}{C(\boldsymbol{p})}$ where $R'_l = \sum_{i : \boldsymbol{n}_i = \boldsymbol{n}'_l} R_i$.

Clearly, $\pi(\boldsymbol{p})$ is a disaggregation of the resulting $\pi'(\boldsymbol{p})$. Moreover, if π is a connected ladder, then so is π'.

Increasing the degree will typically not result in a fine ladder as it produces "redundant" states, however thinning can be applied subsequently. We will refer to increasing the degree of the ladder first and thinning it afterwards, as *augmenting* the ladder.

Definition 2.10 (Augmenting)**.** Let $\pi : \Delta^m \to \Delta^k$ be a multivariate ladder of degree d. The augmented ladder $\pi' : \Delta^m \to \Delta^w$, where $w < \min\{(k+1)(m+1), \binom{d+m+1}{m}\}$, is obtained by first increasing the degree of π and then thinning it.

Remark 2.11. Sampling from π and its augmented ladder π' is equivalent in the sense of Remark 2.2, since it is enough to transform the sample in line with the disaggregation and aggregation steps applied.

Remark 2.12. The fact that $w < \binom{d+m+1}{m}$ in the augmented ladder of degree $d+1$, follows by noticing that there are at most $\binom{d+m-1}{m-1}$ homogeneous monomials of degree d in m variables.

Remark 2.13. Let $m = 1$, π be a fine ladder and assume $\boldsymbol{n}_i$'s are ordered lexicographically. Moreover, let $W \sim \text{Ber}(p)$ be independent of $X \sim \pi(p)$ and Y be the convolution of the two, that is $Y = X + W$. Then, $Y \sim \pi'(p)$, where π' is the augmented π.

Notice that given a multivariate ladder $\pi : \Delta^m \to \Delta^k$, augmenting it enough times yields a fine and connected ladder.

Proposition 2.14. *Let $\pi : \Delta^m \to \Delta^k$ be a multivariate ladder of degree d. Augment π d times to construct $\pi' : \Delta^m \to \Delta^w$, where $w < \min\{(k+1)(m+1)^d, \binom{2d+m}{m}\}$. Then π' is a fine and connected ladder and sampling from π' is equivalent to sampling from π.*

Remark 2.15. In practice, it may be enough to augment the ladder π less than d times to produce a fine and connected ladder π'.

2.3.2. Univariate fine and connected ladder over $\mathbb{R}$

Consider the special case of fine and connected ladders where $m = 1$, which will be of particular interest for the monotone CFTP implementation. We shall call such ladders univariate and denote $p_0 = 1 - p$ and $p_1 = p$. The condition that $C(p)$ has no roots in $[0, 1]$ implies $d = k$ and the probabilities take the form of

$$\pi_i(p) = R_i \frac{p^i(1-p)^{k-i}}{C(p)}, \qquad i = 0, \ldots, k. \tag{5}$$

This case corresponds to having access to a p-coin and simulating a $(k+1)$-sided die, so that the classic Bernoulli Factory setting falls in this scenario.

3. A dice enterprise for rational functions

We now tackle the problem of designing an algorithm that given a die where the probability $\boldsymbol{p}$ of rolling each face is unknown, produces rolls of a die (possibly having a different number of faces) where the probability associated to each face is given by a rational function $f : \Delta^m \to \Delta^v$. We first show how to decompose a rational function f into a fine and connected ladder $\pi : \Delta^m \to \Delta^k$ such that sampling from f is equivalent to sampling from π. Then, we detail how to construct a Markov chain that admits such π as its stationary distribution. Finally, we apply CFTP perfect sampling technique to get a sample exactly from π. For the case $m = 1$, the CFTP has a monotonic implementation with improved efficiency.

3.1. Construction of $\boldsymbol{\pi}$

Theorem 3.1 below shows that for any rational function $f : \Delta^m \to \Delta^v$, a fine and connected ladder $\pi : \Delta^m \to \Delta^k$ can be constructed, such that sampling from f is equivalent to sampling from π. The construction is explicit and among others, builds on the ideas of [28]. Roughly speaking the key steps are the following:

1: Let $f(\boldsymbol{p}) = (f_0(\boldsymbol{p}), \ldots, f_v(\boldsymbol{p})) = \left(\frac{D_0(\boldsymbol{p})}{E_0(\boldsymbol{p})}, \ldots, \frac{D_v(\boldsymbol{p})}{E_v(\boldsymbol{p})}\right)$ be a given rational function where $D_i(\boldsymbol{p})$ and $E_i(\boldsymbol{p})$ are positive and relatively prime polynomials.

2: Apply Lemma A.2 (presented in Appendix) to each $f_i(\boldsymbol{p})$, so that $f(\boldsymbol{p}) = \left(\frac{d_0(\boldsymbol{p})}{e_0(\boldsymbol{p})}, \ldots, \frac{d_v(\boldsymbol{p})}{e_v(\boldsymbol{p})}\right)$ and each $d_i(\boldsymbol{p})$ and $e_i(\boldsymbol{p})$ is an homogeneous polynomial with positive coefficients.

3: Rewrite $f(\boldsymbol{p})$ using a common denominator, so that $f(\boldsymbol{p}) = \frac{1}{C(\boldsymbol{p})}(G_0(\boldsymbol{p}), \dots, G_v(\boldsymbol{p}))$.
4: Rewrite each polynomial $G_i(\boldsymbol{p})$ as a homogeneous polynomial of degree d.
5: Using Proposition 2.14, construct a fine and connected ladder $\pi(\boldsymbol{p})$ sampling from which is equivalent to sampling from $f(\boldsymbol{p})$.

Detailed construction can be found in the proof of the following theorem and is also illustrated in Example 3.

Theorem 3.1. *Let $f : \Delta^m \to \Delta^v$ be a probability distribution such that every $f_i(\boldsymbol{p})$ is a rational function with real coefficients. Then, one can explicitly construct a fine and connected ladder $\pi : \Delta^m \to \Delta^k$ such that sampling from π is equivalent to sampling from f.*

Notice that we assume that $f(\boldsymbol{p}) \in \Delta^v$ for every $\boldsymbol{p} \in \Delta^m$. This rules out functions such as $f(p) = \min(2p, 1)$ - as expected since a Bernoulli Factory for such function cannot be constructed [23].

3.2. Construction of the Markov chain

Let $\pi : \Delta^m \to \Delta^k$ be a fine and connected ladder. We now consider the problem of designing a Markov chain that admits it as its stationary distribution. The main idea behind the construction is depicted in Figure 3: a roll of the die determines the possible directions for the next move. We then draw a Uniform r.v. and decide whether the chain stays still or moves to a specific state. We can then write the off diagonal entries of the transition matrix P of the chain as an entrywise product of V and W, i.e.

$$P_{i,j} = \begin{cases} V_{i,j} \cdot W_{i,j} & \text{if } i \neq j \\ 1 - \sum_{h \neq i} P_{i,h} & \text{if } i = j \end{cases} \tag{6}$$

where V is a matrix of real numbers in $[0, 1]$ and W is a matrix where the off diagonal elements are either null or equal to p_b for some $b \in \{0, \dots, m\}$.

It is convenient to introduce the following definition of neighbourhood that, once the $(m+1)$-sided die has been rolled, details where the chain may move.

Definition 3.2. Let $\pi : \Delta^m \to \Delta^k$ be a fine and connected multivariate ladder. Let $b \in \{0, \dots, m\}$ and let $\boldsymbol{e}_b \in \bar{\Delta}^m$ be the b^{th} standard unit vector. For each $i \in \Omega = \{0, \dots, k\}$ define the neighbourhood of i in the direction of b as

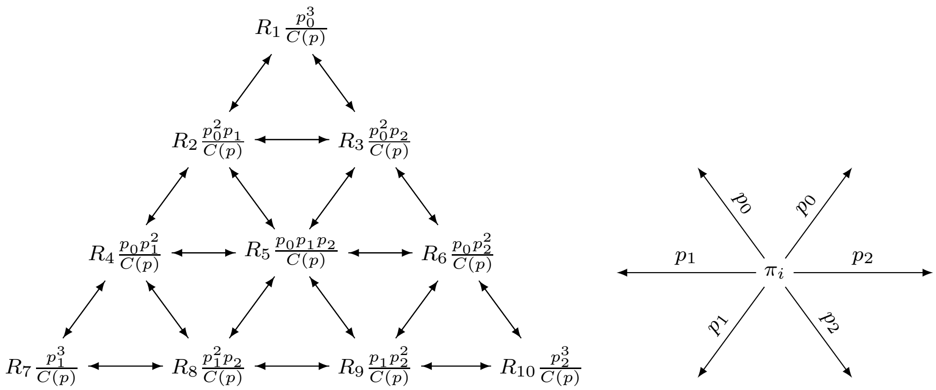
$$\mathcal{N}_b(i) = \{j \in \Omega \setminus \{i\} : \|\boldsymbol{n}_i - \boldsymbol{n}_j + \boldsymbol{e}_b\|_1 = 1\}. \tag{7}$$

We will also denote

$$\mathcal{S}_b(i) = \sum_{j \in \mathcal{N}_b(i)} R_j. \tag{8}$$

Remark 3.3. Unlike $\mathcal{N}(i)$ (cf. Definition 2.7), $\mathcal{N}_b(i)$ may be empty and

$$\mathcal{N}(i) = \bigcup_{b \in \{0, \dots, m\}} \mathcal{N}_b(i).$$



(a) A connected and fine ladder $\pi : \Delta^2 \to \Delta^9$

(b) Possible moves given a roll of the die

Figure 3: Markov chain structure for a multivariate ladder

We can now set the elements of the matrix W in (6) as

$$W_{i,j} = \begin{cases} p_b & \text{if } j \in \mathcal{N}_b(i) \\ 0 & \text{if } j \notin \mathcal{N}(i). \end{cases} \tag{9}$$

The matrix V in (6) shall be iteratively defined as follows: first, select the state $i \in \Omega$ and the roll $b \in \{0, \dots, m\}$ that maximises $\mathcal{S}_b(i)$. For each $j \in \mathcal{N}_b(i)$ assign $V_{i,j} = \frac{R_j}{\mathcal{S}_b(i)}$. Since the construction shall yield a reversible Markov chain, also set $V_{j,i} = \frac{R_i}{\mathcal{S}_b(i)}$. Then proceed in a similar fashion, taking into account the entries of the matrix that had already been fixed. The detailed procedure is described in Algorithm 5.

Proposition 3.4. *Let $\pi : \Delta^m \to \Delta^k$ be a fine and connected ladder. Consider a discrete-time Markov chain $(X_t)_{t\in\mathbb{N}}$ on $\Omega = \{0, \dots, k\}$. Let P as in (6) be the transition matrix of the chain, where W is as in (9) and V is the matrix output by Algorithm 5. Then, $(X_t)_{t\in\mathbb{N}}$ is a time-reversible Markov chain that admits $\pi(\boldsymbol{p})$ as its unique stationary distribution.*

Notice that the transition matrix of Proposition 3.4 is just one of many possible choices (see Remark 3.9 for an alternative construction). In principle, to speed up the convergence of CFTP it is good practice to reduce the mixing time of the chain [35]. A related and more operational criterion is that of Peskun ordering [32].

Definition 3.5. Given two reversible Markov chains with the same stationary distribution π and with transition matrices P and Q, we say that Q dominates P in Peskun sense, and write $Q \succeq_P P$, if each of the off diagonal elements of Q are greater or equal to the corresponding elements of P.

Algorithm 5 Construction of the Markov chain transition matrix

Input: A multivariate ladder $\pi : \Delta^m \to \Delta^k$ on $\Omega = \{0, \dots, k\}$
Output: The matrix V composing the transition kernel in equation (6).
Initialisation step
Initialise V as a $(k+1) \times (k+1)$ null matrix
For all $i \in \Omega$ and $b \in \{0, \dots, m\}$ set $\mathcal{N}_b(i)$ as in eq. (7), $\mathcal{S}_b(i)$ as in eq. (8), $\mathcal{W}_b(i) \leftarrow 0$
Main loop
repeat
  Set $b, i \leftarrow \arg\max_{b,i} \mathcal{S}_b(i)$
  for each $j \in \mathcal{N}_b(i)$ **do**
  Assign maximum probability of moving from state i having rolled b
    $V_{i,j} \leftarrow R_j/S_b(i)$, $\mathcal{N}_b(i) \leftarrow \mathcal{N}_b(i) \setminus \{j\}$, $\mathcal{W}_b(i) \leftarrow \mathcal{W}_b(i) + R_j/S_b(i)$
  Assign values for the reverse move
    Set c such that $i \in \mathcal{N}_c(j)$
    $V_{j,i} \leftarrow R_i/S_b(i)$, $\mathcal{N}_c(j) \leftarrow \mathcal{N}_c(j) \setminus \{i\}$, $\mathcal{W}_c(j) \leftarrow \mathcal{W}_c(j) + R_i/S_b(i)$
  Update $S_c(j)$ to take into consideration the new value of $V_{j,i}$
    $S_c(j) \leftarrow \left(\sum_{h \in \mathcal{N}_c(j)} R_h\right)(1 - \mathcal{W}_c(j))^{-1}$
  end for
  Update $\mathcal{S}_b(i)$
  $\mathcal{S}_b(i) \leftarrow 0$
until $\mathcal{N}_b(i) = \emptyset, \forall i \in \Omega, b \in \{0, \dots, m\}$

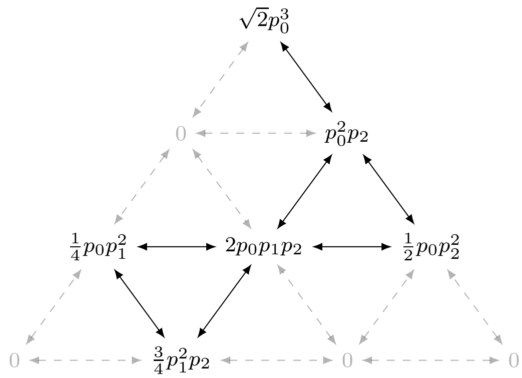
If $Q \succeq_P P$ then, for any π integrable target function f, using Q results in a smaller asymptotic variance in the Markov chain CLT than using P. Moreover, for positive operators, if $Q \succeq_P P$ then the Q-chain converges in total variation more rapidly towards the stationary distribution [27]. Consequently, Peskun ordering represents a valuable tool to assess our choice of the transition matrix of Proposition 3.4 and the following Proposition proves that it is optimal.

Proposition 3.6. *Let $\pi : \Delta^m \to \Delta^k$ be a fine and connected ladder. Consider the Markov chain defined in Proposition 3.4. Then, there does not exist a reversible Markov chain with the same adjacency structure and stationary distribution that dominates it in the Peskun sense.*

Example 1. Consider the multivariate ladder

$$\pi(p_0, p_1, p_2) \propto \left(\underbrace{\sqrt{2}p_0^3}_{\pi_0}, \underbrace{p_0^2 p_2}_{\pi_1}, \underbrace{\frac{1}{4}p_0 p_1^2}_{\pi_2}, \underbrace{2p_0 p_1 p_2}_{\pi_3}, \underbrace{\frac{1}{2}p_0 p_2^2}_{\pi_4}, \underbrace{\frac{3}{4}p_1^2 p_2}_{\pi_5}\right).$$

We can graphically represent the ladder as



The neighbourhoods of each state are

$$\begin{aligned} &\mathcal{N}(0) = \{1\}, && \mathcal{N}(1) = \{0,3,4\}, && \mathcal{N}(2) = \{3,5\}, \\ &\mathcal{N}(3) = \{1,2,4,5\}, && \mathcal{N}(4) = \{1,3\}, && \mathcal{N}(5) = \{2,3\}. \end{aligned}$$

and given how we defined $\mathcal{N}_b(i)$ we have

$$\begin{aligned} &\mathcal{N}_0(0) = \emptyset, && \mathcal{N}_1(0) = \emptyset, && \mathcal{N}_2(0) = \{1\}, \\ &\mathcal{N}_0(1) = \{0\}, && \mathcal{N}_1(1) = \{3\}, && \mathcal{N}_2(1) = \{4\}, \\ &\mathcal{N}_0(2) = \emptyset, && \mathcal{N}_1(1) = \emptyset, && \mathcal{N}_2(2) = \{3,5\}, \\ &\mathcal{N}_0(3) = \{1\}, && \mathcal{N}_1(3) = \{2,5\}, && \mathcal{N}_2(3) = \{4\}, \\ &\mathcal{N}_0(4) = \{1\}, && \mathcal{N}_1(4) = \{3\}, && \mathcal{N}_2(4) = \emptyset, \\ &\mathcal{N}_0(5) = \{2,3\}, && \mathcal{N}_1(5) = \emptyset, && \mathcal{N}_2(5) = \emptyset. \end{aligned}$$

The transition matrix obtained through Algorithm 5 is then equal to

$$P = \begin{pmatrix} \cdot & \frac{1}{\sqrt{2}}p_2 & 0 & 0 & 0 & 0 \\ p_0 & \cdot & 0 & p_1 & \frac{1}{2}p_2 & 0 \\ 0 & 0 & \cdot & \frac{8}{11}p_2 & 0 & \frac{3}{11}p_2 \\ 0 & \frac{1}{2}p_0 & \frac{1}{11}p_1 & \cdot & \frac{15}{44}p_2 & \frac{1}{3}p_1 \\ 0 & p_0 & 0 & p_1 & \cdot & 0 \\ 0 & 0 & \frac{1}{11}p_0 & \frac{10}{11}p_0 & 0 & \cdot \end{pmatrix}$$

where $\cdot$ represents the required quantity so that the rows sum up to 1.

3.3. *Perfect sampling*

We now introduce an update function for the Markov chain defined in Proposition 3.4 so that CFTP is implementable. We are then able to draw samples from a multivariate fine and connected ladder and thus solve the original problem via Theorem 3.1. For the general case of a die with more than 2 faces, the update function defined in the following proposition is not necessarily monotonic. However, in the Bernoulli Factory setting of $m = 1$, we can define a monotonic update function for the Markov chain as shown in Corollary 3.8. Notice that even when a monotonic construction is not possible, CFTP can still be used in practice. As numerical examples demonstrate (cf. Examples 4, 6), if the degree of the polynomials and the numbers of faces of the given die are not too large, running times are not prohibitive.

Proposition 3.7. *Given a fine and connected ladder $\pi : \Delta^m \to \Delta^k$, consider the Markov chain $(X_t)_{t\in\mathbb{N}}$ with transition matrix P of the form (6) and defined in Proposition 3.4. Let $B \sim \boldsymbol{p}$ and $U \sim Unif(0,1)$ be independent random variables. Given $i \in \Omega$ denote the elements of $\mathcal{N}_B(i)$ as $\mathcal{N}_B(i) = \{j_0, \dots, j_w\}$. Define the function $\phi : \{0,\dots,k\} \times \{0,\dots,m\} \times [0,1] \to \{0,\dots,k\}$ as*

$$\phi(i,B,U) = \begin{cases} j_0 & \text{if } U \le V_{i,j_0}, \\ j_1 & \text{if } V_{i,j_0} < U \le V_{i,j_0} + V_{i,j_1}, \\ \quad\dots & \\ j_l & \text{if } \sum_{h=0}^{l-1} V_{i,j_h} < U \le \sum_{h=0}^{l} V_{i,j_h}, \\ \quad\dots & \\ i & \text{otherwise.} \end{cases} \tag{10}$$

Then ϕ is an update function for the Markov chain $(X_t)_{t\in\mathbb{N}}$.

3.4. A special case: from coins to dice

Assume that we are given a p-coin, and write $p_0 = 1-p$ and $p_1 = p$, to consider a fine and connected ladder of the form $\pi : (0,1) \to \Delta^k$ as in equation (5). In this case the definition of neighbourhoods simplifies and so does the Markov chain defined in Proposition 3.4. Moreover, given the simplified structure of the state space, the update function defined in Proposition 3.7 is monotonic, so that monotonic CFTP can be employed. These observations are summarised in the following Corollary. Figure 4 gives a graphical representation of the dynamics of the Markov chain.

Corollary 3.8. *Let $\pi : (0,1) \to \Delta^k$ be a fine and connected ladder as in equation (5). The transition matrix of the Markov chain $(X_t)_{t\in\mathbb{N}}$ defined in Proposition 3.4 can be rewritten as*

$$P_{i,j} = \begin{cases} R_{i-1}\frac{1-p}{R_{i-1}\vee R_i} & \text{if } j = i-1, j > 0, \\ 1 - R_{i-1}\frac{1-p}{R_{i-1}\vee R_i} - R_{i+1}\frac{p}{R_i \vee R_{i+1}} & \text{if } j = i, \\ R_{i+1}\frac{p}{R_i\vee R_{i+1}} & \text{if } j = i+1, j < k, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Let $U \sim Unif(0,1)$ and $p-$coin B be an independent r.v. (i.e. $\mathbb{P}(B = 1) = 1 - \mathbb{P}(B = 0) = p$). The update function defined in Proposition 3.7 can be *rewritten as*

$$\phi(i,B,U) = \begin{cases} i-1 & \textit{if } i>0, B=0, U \le \frac{R_{i-1}}{R_{i-1}\vee R_i}, \\ i+1 & \textit{if } i<k, B=1, U \le \frac{R_{i+1}}{R_i \vee R_{i+1}}, \\ i & \textit{otherwise.} \end{cases} \tag{12}$$

Moreover, ϕ is a monotonic update function for the Markov chain $(X_t)_{t\in\mathbb{N}}$ where 0 and k are the minimum and maximum states respectively.

Remark 3.9. The explicit form of the transition matrix P in equation (11) can be extended to the multivariate case. Consider a fine and connected ladder $\pi : \Delta^m \to \Delta^k$ and for $i,j \in \Omega$ let $b,c \in \{0,\dots,m\}$ such that $j \in \mathcal{N}_b(i)$ and $i \in \mathcal{N}_c(j)$. Define a transition matrix P for the chain $(X_t)_{t\in\mathbb{N}}$ as

$$P_{i,j} = \begin{cases} \dfrac{R_j}{\displaystyle\sum_{h\in\mathcal{N}_b(i)} R_h \vee \sum_{h\in\mathcal{N}_c(j)} R_h} p_b & \text{if } j\in\mathcal{N}_b(i) \text{ and } i \in \mathcal{N}_c(j) \\ 1-\sum_{k\in\mathcal{N}(i)} P_{i,j} & \text{if } i=j \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

It turns out that this is a valid choice and the chain admits $\pi(\boldsymbol{p})$ as its unique stationary distribution. However, unlike the transition kernel defined in defined in Proposition 3.4, P in (13) may not be optimal in Peskun sense.

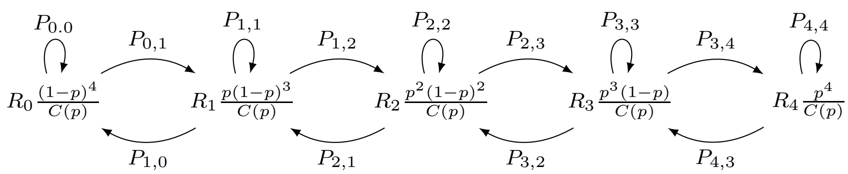


Figure 4: Transition probabilities on a fine and connected univariate ladder.

3.5. Efficiency of the algorithm

We now provide some results on the expected number of rolls of the original die required by CFTP and give insights on how the algorithm can be sped up. In particular, we provide conditions for the expected number of rolls to be bounded uniformly in $\boldsymbol{p}$. Interestingly, this is always the case when $m=1$, thus also in the Bernoulli Factory scenario. Moreover, we give tighter bounds when the univariate ladder is strictly log-concave, as defined below, and show that univariate ladders can be always transformed into an equivalent log-concave ladder through augmentation.

Definition 3.10 (Log-concave discrete distribution)**.** A discrete distribution $\boldsymbol{\mu}$ on $\Omega = \{0,\dots,k\}$ is log-concave if for all $0<i<k$,

$$\mu_i^2 \ge \mu_{i-1}\mu_{i+1}. \tag{14}$$

If the inequality is strict, then $\boldsymbol{\mu}$ is said to be strictly log-concave.

We shall also show that augmenting the degree of the ladder produces a log-concave distribution out of a ladder that is not itself log-concave. To that end, we provide the following general theorem that is of independent interest.

Theorem 3.11. *For every discrete random variable W on $\Omega = \{0, \dots, n_0\}$ where $n_0 < \infty$, there exists a number $n = n(W)$ such that $W + B_n$ is strictly log-concave, where B_n is an independent Binomial$(n, 1/2)$.*

Proof of the theorem is deferred to the Appendix. Several works have looked into whether log-concavity is preserved [16, 36], but checking whether some operations *introduce* log-concavity seems to be a harder problem [22] and the above result appears to be the first in this direction.

Building on Theorem 3.11, we will show that augmenting the degree of the ladder may lead to faster implementation. This is expanded in Proposition 3.17 and empirically verified in Examples 2 and 4.

3.5.1. General case

Proposition 3.12. *Let $\pi(\boldsymbol{p}) : \Delta^m \to \Delta^k$ be a fine and connected ladder of degree d. Write $\boldsymbol{n}_i$ as in Definition 2.6 and assume $E := \{b \in \{0, \dots, m\} : \exists i, n_{i,b} = d\}$ is nonempty. Denote by N the number of rolls of the original die required by CFTP when the update function of Proposition 3.7 is used. Then, one can explicitly construct a new ladder $\pi'(\boldsymbol{p}) : \Delta^m \to \Delta^w$ of degree $2d$ where $w < \min\{k(m+1)^d, \binom{2d+m}{m}\}$ that is a disaggregation of π and such that*

$$\mathbb{E}[N] \leq \min_{b \in E} \frac{(ap_b)^{-2d} - 1}{1 - ap_b},$$

where $a \in (0, 1]$ is a constant independent on $\boldsymbol{p}$.

Remark 3.13. If $E = \{0, \dots, m\}$, then we can bound N by a quantity independent of $\boldsymbol{p}$, that is

$$\mathbb{E}[N] \leq \min_{b \in E} \frac{(ap_b)^{-2d} - 1}{1 - ap_b} \leq \frac{\left(\frac{a}{m+1}\right)^{-2d} - 1}{1 - \frac{a}{m+1}} \tag{15}$$

3.5.2. From coins to dice

We now restrict our analysis to rational functions of the form $f : (0, 1) \to \Delta^v$, where an implementation of monotonic CFTP is possible. We study the efficiency of the proposed method in terms of the required number of tosses of the given p-coin. A direct consequence of Proposition 3.12 is the following.

Corollary 3.14. *Let $\pi(p) : (0, 1) \to \Delta^k$ be a fine and connected ladder. Denote by N the number of tosses of the p-coin required by CFTP when the update* *function of Corollary 3.8 is used. Then, one can explicitly construct a new ladder* $\pi'(\boldsymbol{p}) : (0,1) \to \Delta^{2k}$ *that is a disaggregation of* π *and such that*

$$\mathbb{E}[N] \leq \min\left\{ \frac{(ap)^{-2(k-1)} - 1}{1 - ap}, \frac{(a(1-p))^{-2(k-1)} - 1}{1 - a(1-p)} \right\} \leq \frac{\left(\frac{a}{2}\right)^{-2(k-1)} - 1}{1 - \frac{a}{2}}$$

where $a = \min_i \left\{ \frac{R_{i-1}}{R_{i-1} \vee R_i} \wedge \frac{R_i}{R_{i-1} \vee R_i} \right\} \in (0,1]$ *is a constant independent of* $\boldsymbol{p}$.

Therefore, the expected running time of the algorithm can be bounded uniformly in p. However, the bound proposed in Corollary 3.14 is generally very loose and does not give insights into how the algorithm could potentially be sped up. We now provide a tighter bound under the condition that the ladder $\pi(p)$ is a log-concave distribution.

The proof of the following Proposition is in spirit similar to the Path Coupling technique of [4].

Proposition 3.15. *Let* $\pi : (0,1) \to \Delta^k$ *be a univariate fine and connected ladder as in (5). Assume further that* π *is strictly log-concave and that the Markov chain and update function defined in Corollary 3.8 are used. Then*

$$\mathbb{P}(N \geq n) \leq (k-1)\rho^n$$

where $\rho \in (0,1)$ *for all* $p \in (0,1)$ *is given by*

$$\rho = \max_{i \in \{0,\ldots,k-2\}} [1 - (P_{i,i+1} - P_{i+1,i+2}) - (P_{i+1,i} - P_{i,i-1})] \tag{16}$$

with $P_{i,j}$ *given by (11) with the convention that* $P_{k,k+1} = P_{0,-1} = 0$.

Remark 3.16. Given a univariate fine and connected ladder, ρ as in equation (16) can be explicitly computed for a fixed $p \in (0,1)$. Moreover, the algorithm still requires a finite number of tosses even if $p = 1$ or $p = 0$. In this case the expected number of tosses $\mathbb{E}[N]$ can be exactly computed:

$$\mathbb{E}[N] = \begin{cases} \frac{R_0 \vee R_1}{R_1} + \ldots + \frac{R_{k-2} \vee R_{k-1}}{R_{k-1}} & \text{if } p = 1, \\ \frac{R_0 \vee R_1}{R_0} + \ldots + \frac{R_{k-2} \vee R_{k-1}}{R_{k-2}} & \text{if } p = 0. \end{cases}$$

Given a rational function $f : (0,1) \to \Delta^v$, we have proved in Theorem 3.1 that it is always possible to construct a univariate fine and connected ladder $\pi : (0,1) \to \Delta^k$. However, π may not be strictly log-concave. Using Theorem 3.11, we now show that augmenting the ladder enough times produces a new ladder that is strictly log-concave, so that Proposition 3.15 applies.

Lemma 3.17. *Let* $\pi : (0,1) \to \Delta^k$ *be a univariate fine and connected ladder as in Section 2.3.2. Then, one can explicitly construct a new univariate fine and connected ladder* $\pi' : (0,1) \to \Delta^w$ *where* $w \geq k$ *such that* π' *is a disaggregation of* π *and* π' *is strictly log-concave.*

Hence increasing the degree of a generic univariate fine and connected ladder $\pi : (0,1) \to \Delta^k$ may lead to a faster implementation of monotonic CFTP despite an increased number of states that the chain needs to visit. Clearly, this leads to a trade-off that the user may want to calibrate, as shown in Examples 2 and 4.

Example 2. Consider sampling from the following ladder

$$\pi(p) \propto ((1-p)^4, 1000p(1-p)^3, p^2(1-p)^2, 500p^3(1-p), p^4).$$

Clearly, π is not log-concave. We can augment the ladder up to two times to obtain respectively

$$\pi^{(1)} \propto ((1-p)^5, 1001p(1-p)^4, 1001p^2(1-p)^3, 501p^3(1-p)^2, 500p^4(1-p), p^5)$$
$$\pi^{(2)} \propto ((1-p)^6, 1002p(1-p)^5, 2002p^2(1-p)^4,$$
$$1502p^3(1-p)^3, 1001p^4(1-p)^2, 500p^5(1-p), p^6).$$

Notice that now $\pi^{(2)}$ is strictly log-concave. Table 1 shows the empirical number of tosses required by the algorithm when sampling from either $\pi(p)$, $\pi^{(1)}(p)$ or $\pi^{(2)}(p)$ for different values of p. Notice that even if $\pi^{(1)}(p)$ is not log-concave, it still leads to a slightly faster implementation than when targeting $\pi^{(2)}(p)$.

	True value of p						
	0.01	0.1	0.25	0.5	0.75	0.9	0.99
$\hat{\mathbb{E}}[N_\pi]$	561.31	621.73	827.86	1332.63	1433.59	1209.28	1090.54
$\hat{\mathbb{E}}[N_{\pi^{(1)}}]$	92.35	12.21	7.72	8.65	9.48	12.59	87.05
$\hat{\mathbb{E}}[N_{\pi^{(2)}}]$	93.17	13.33	9.43	11.29	11.67	14.47	89.56

Table 1: Average number of required tosses of the p-coin over 1,000 runs of the algorithm when targeting $\pi, \pi^{(1)}$ and π_2.

4. Examples and implementation

An R package implementing the method and reproducing the examples is available at https://github.com/giuliomorina/DiceEnterprise. The user is just required to define the function $f(\boldsymbol{p})$ and to provide a function that rolls the original die. Then, the package automatically constructs the fine and connected ladder and implements CFTP. If the original die has only two faces, the monotonic version of CFTP is automatically employed.

We now show how the method works and performs on some examples, all of which can be reproduced using the provided package. We start with a toy example to better explain and highlight the construction proposed in Theorem 3.1 and Proposition 3.4. Next, we examine efficiency of the monotonic and general versions of the algorithm by considering higher order rational functions. We also consider the so-called logistic Bernoulli factory as studied in [19]. We show that our method leads to a simple algorithm which on average requires the same number of tosses as the approach of [19]. Finally, we deal with a slightly different scenario where instead of an m-sided die, m independent coins are provided where the probability $\boldsymbol{p} = (p_0, \dots, p_{m-1}) \in (0,1)^m$ of tossing heads is unknown. In particular, we notice how we can construct a Dice Enterprise for the "Bernoulli Race" function considered in [7] which again has the same performance in terms of the expected number of required tosses.

Example 3 (Toy example of Bernoulli Factory)**.** Let $p \in (0,1)$ and assume we wish to generate a coin that lands heads with probability

$$\frac{\sqrt{2}p^3}{(\sqrt{2}-5)p^3 + 11p^2 - 9p + 3},$$

having access only to a p-coin. Our proposed construction produces the following fine and connected ladder

$$\pi(p) = (3(1-p)^4, 3p(1-p)^3, 2p^2(1-p)^2, (\sqrt{2}+2)p^3(1-p), \sqrt{2}p^4),$$

via the following steps:

1. Let $C(p) = (\sqrt{2}-5)p^3 + 11p^2 - 9p + 3$ and consider

$$f(p) = \frac{1}{C(p)}(\underbrace{-5p^3 + 11p^2 - 9p + 3}_{D_0(p)}, \underbrace{\sqrt{2}p^3}_{D_1(p)}).$$

   Convert $D_0(p)$ and $D_1(p)$ into homogeneous polynomials in the variables p and $(1-p)$ with positive coefficients and of the same degree. This can be achieved by using the multinomial theorem (cf. proof of Theorem 3.1). We get

$$\begin{aligned} D_0(p) &= 2p^2(1-p) + 3(1-p)^3, \\ D_1(p) &= \sqrt{2}p^3, \end{aligned}$$

   so that we can equivalently consider the ladder

$$\pi'(p) = \frac{1}{C(p)}(3(1-p)^3, 2p^2(1-p), \sqrt{2}p^3).$$

   and notice that if $X \sim \pi'$, then $W = \mathbb{I}(X \in \{3\})$ is distributed as $f(p)$.
2. Notice that π' is not a connected ladder, as there is no term proportional to $p(1-p)^2$. By applying the binomial theorem, we can construct a new ladder

$$\tilde{\pi}(p) = \frac{1}{C(p)}(\tilde{\pi}_0(p), \tilde{\pi}_1(p), \tilde{\pi}_2(p), \tilde{\pi}_3(p), \tilde{\pi}_4(p), \tilde{\pi}_5(p)),$$

where

$$\begin{aligned}
\tilde{\pi}_0(p) &= \pi'_0(p)\binom{1}{0}(1-p) = 3(1-p)^4,\\
\tilde{\pi}_1(p) &= \pi'_0(p)\binom{1}{1}p = 3p(1-p)^3,\\
\tilde{\pi}_2(p) &= \pi'_1(p)\binom{1}{0}(1-p) = 2p^2(1-p)^2,\\
\tilde{\pi}_3(p) &= \pi'_1(p)\binom{1}{1}p = 2p^3(1-p),\\
\tilde{\pi}_4(p) &= \pi'_2(p)\binom{1}{0}(1-p) = \sqrt{2}p^3(1-p),\\
\tilde{\pi}_5(p) &= \pi'_2(p)\binom{1}{1}p = \sqrt{2}p^4.
\end{aligned}$$

Notice that if $Y \sim \tilde{\pi}$, then $X = \mathbb{I}(Y \in \{2,3\}) + 2 \cdot \mathbb{I}(Y \in \{4,5\})$ is distributed as π'.

3. Finally, we can construct a fine and connected ladder by adding up together the terms where the same monomial appears:

$$\pi(p) = \frac{1}{C(p)}(3(1-p)^4, 3p(1-p)^3, 2p^2(1-p)^2, (\sqrt{2}+2)p^3(1-p), \sqrt{2}p^4).$$

Assume $Z \sim \pi$, $U \sim \mathrm{Unif}(0,1)$ and let

$$\begin{aligned}
Y = &\cdot\mathbb{I}(Z=1) + 2\cdot\mathbb{I}(Z=2) +\\
&3\cdot\mathbb{I}\left(Z=3, U \leq \frac{2}{2+\sqrt{2}}\right) + 4\cdot\mathbb{I}\left(Z=3, U > \frac{2}{2+\sqrt{2}}\right) + 5\cdot\mathbb{I}(Z=4)
\end{aligned}$$

so that $Y \sim \tilde{\pi}$.

Table 2 shows the performance of CFTP for different value of the unknown probability p.

p	0.01	0.1	0.25	0.5	0.75	0.9	0.99
$f(p)$	0.00	0.00	0.02	0.22	0.65	0.86	0.99
Lower C.B.	0.00	0.00	0.01	0.18	0.63	0.86	0.98
$\hat{f}(p)$	0.00	0.00	0.01	0.21	0.66	0.88	0.99
Upper C.B.	0.00	0.00	0.02	0.23	0.69	0.90	0.99
$\hat{\mathbb{E}}[N]$	4.80	5.61	7.45	10.61	8.05	6.51	5.94

Table 2: Implementation of the Bernoulli Factory for the function $\frac{\sqrt{2}p^3}{(\sqrt{2}-5)p^3+11p^2-9p+3}$ and for different values of the true unknown probability p. The algorithm has been run 1,000 times to obtain tosses of the $f(p)$-coin and $\hat{f}(p)$ is the sample average. The confidence interval were computed with a 95% confidence level. The empirical expected number of tosses in a run of the CFTP algorithm is given by $\hat{\mathbb{E}}[N]$.

Example 4 (Augmenting the number of states can lead to faster running time)**.** Given a p-coin, consider constructing a 3-sided die where the probability of rolling each face is given by

$$\pi(p) \propto \{p^{20}, p^{10}(1-p)^{10}, (1-p)^{20}\}. \tag{17}$$

A naive approach to construct a Bernoulli Factory for $\pi(p)$ would be the following: toss the p-coin 20 times and with probability 1/3 output 1 if all the tosses are heads, with probability 1/3 output 2 if the first 10 tosses are heads and the last 10 tosses are tails, with probability 1/3 output 3 if all the tosses are tails. In all other cases, restart the algorithm.

Assume now that $p = 1/2$, so that the expected number of tosses of this procedure would be $\mathbb{E}[N] = 2^{20}$. Table 3 shows the performance of our novel algorithm on the same example when targeting the ladder of equation (17), as well as when targeting the augmented ladder where extra states are added. Indeed, Lemma 3.17 and Proposition 3.15 suggest that doing so may lead to faster performance, as empirically confirmed. Notice that to get a strictly log-concave ladder, we need to augment π at least 203 times. In practice, it is enough to augment it around 40 times to obtain optimal performance, due to the trade-off effect discussed in Section 3.5.

	Number of states added to the original ladder $\pi(p)$						
	+0	+20	+40	+60	+80	+100	+120
$\hat{\mathbb{E}}[N]$	5337.7	585.7	471.4	481.7	529.3	590.4	647.9
Log-concave	No	No	No	No	No	No	No
	+140	+160	+180	+200	+220	+240	+260
$\hat{\mathbb{E}}[N]$	717.2	774.2	840.2	892.3	927.3	996.4	1038.9
Log-concave	No	No	No	No	Yes	Yes	Yes

Table 3: Implementation of the Dice Enterprise for the function of eq. (17) when $p = 1/2$. The algorithm has been run 1,000 times and $\hat{\mathbb{E}}[N]$ is the empirical number of tosses of the p-coin required. It is also reported whether the augmented ladder is strictly log-concave.

Consider now a slightly different example, where a 3-sided fair die is given, i.e. $\boldsymbol{p} = (1/3, 1/3, 1/3)$, and the aim is to construct a 4-sided die where the probability of rolling each face is given by

$$\pi(\boldsymbol{p}) \propto \{p_0^5 p_1^5 p_2^5, p_0^{15}, p_1^{15}, p_2^{15}\}. \tag{18}$$

A naive approach as the one before would require on average $\mathbb{E}[N] = 3^{15}$ rolls of the $\boldsymbol{p}$-die. Although the result of Proposition 3.15 does not hold here, as a monotonic implementation of CFTP is not possible, augmenting the ladder may still lead to faster performance. This is indeed the case, as shown in Table 4, where targeting the ladder with 60 extra states leads to an implementation that requires on average around 840 tosses of the $\boldsymbol{p}$-die, instead of more than 100,000 when directly targeting the original $\pi(\boldsymbol{p})$.

	Number of states added to the original ladder $\pi(\boldsymbol{p})$							
	+0	+10	+20	+30	+40	+50	+60	+70
$\hat{\mathbb{E}}[N]$	174246.4	2569.0	1341.5	1032.9	912.5	874.0	841.4	860.1

Table 4: Implementation of the Dice Enterprise for the function of eq. (18) when $\boldsymbol{p} = (1/3, 1/3, 1/3)$. The algorithm has been run 1,000 times and $\hat{\mathbb{E}}[N]$ is the empirical number of rolls of the $\boldsymbol{p}$-die required.

Example 5 (Logistic Bernoulli Factory)**.** Consider constructing a Bernoulli factory for the function

$$\frac{Cp}{1+Cp}, \qquad C > 0.$$

Such problem is considered in [19] where it is referred as constructing a logistic Bernoulli factory. In the same paper, the author proposes an ad-hoc algorithm that exploits properties of thinned Poisson processes and requires on average $\mathbb{E}[N_H] = C/(1+Cp)$ tosses of the p-coin. We now show that our proposed method leads to an alternative algorithm that requires on average the same number of tosses. The fine and connected ladder for this target is

$$\pi(p) = \frac{1}{1+Cp}((1+C)p, (1-p)).$$

Given $Y \sim \pi$ and $U \sim \mathrm{Unif}(0,1)$, we output heads if $Y = 1, U < C/(1+C)$ and tails otherwise. Sampling from $\pi(p)$ boils down to sampling from the stationary distribution of a Markov chain consisting of only two states, as depicted in Figure 5. CFTP needs to keep track of only two chains starting in the two states and the algorithm stops as soon as one of the two chain moves, as they cannot both move at the same time. In particular, the particles coalesce if heads is tossed or if the uniform r.v. U drawn by the algorithm is such that $U \leq 1/(1+C)$. Therefore, CFTP is equivalent to algorithm 6 which is a special case of the 2-coin algorithm presented in [11, 12] with $c_1 = C, c_2 = 1, p_1 = p, p_2 = 1$.

Algorithm 6 Logistic Bernoulli Factory

Input: black box to sample from $\mathrm{Ber}(p)$, a constant $C > 0$.
Output: a sample from $\mathrm{Ber}(Cp/(1+Cp))$.

```
Sample U ~ Unif(0,1)
if U ≤ 1/(1+C) then set Y := 0
else
    Sample B ~ Bern(p)
    if B = 1 then set Y := 1
    else discard U, B and GOTO 1
    end if
end if
Output Y
```

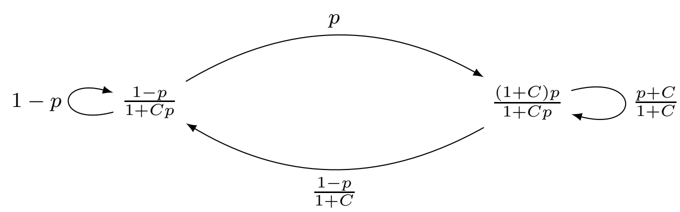


Figure 5: Dynamic of the Markov chain with stationary distribution $\pi(p) = \frac{1}{1+Cp}((1+C)p, (1-p))$.

The probability that the algorithm stops at a specific iteration is $\frac{1+Cp}{1+C}$. Since each iteration is independent of the others and the probability that a toss of the p-coin is required is $\frac{C}{1+C}$, the average number of tosses is then given by

$$\mathbb{E}[N_{\text{CFTP}}] = \frac{1+C}{1+Cp}\frac{C}{1+C} = \frac{C}{1+Cp}$$

and it is thus equal to $\mathbb{E}[N_H]$.

Example 6 (Independent coins and Bernoulli Race)**.** We now deal with a slightly different scenario where instead of having access to a die, m independent coins are given. Similarly, the probability of tossing heads on each of the coin is unknown and given by $\boldsymbol{p} = (p_0, \ldots, p_{m-1}) \in (0,1)^m$, so that the problem is now obtaining a sample from a rational function $f : (0,1)^m \to \Delta^v$. There are several ways to transform tosses of m coins into a roll of a die. In particular, we can construct an $(m+1)$-sided die in the following fashion. Firstly, we choose uniformly which coin to toss, say the i^{th}. If the result is heads we output $i \in \{0, \ldots, m-1\}$, otherwise we output m. The probabilities of obtaining each face by rolling the so constructed $(m+1)$-sided die is then given by $\tilde{\boldsymbol{p}} = \left(\frac{p_0}{m}, \ldots, \frac{p_{m-1}}{m}, 1 - \frac{1}{m}\sum_{i=0}^{m-1} p_i\right) \in \Delta^m$. The function $f(\boldsymbol{p})$ can be transformed into a function of $\tilde{\boldsymbol{p}}$ by substituting $p_i = m\tilde{p}_i$.

We now consider the function $f(\boldsymbol{p}) = \frac{1}{\sum_{i=1}^m p_i}(p_1, \ldots, p_m)$ as in [7], where the problem of tossing such $f(\boldsymbol{p})$-die is named Bernoulli Race. Their proposed algorithm requires on average $\mathbb{E}[N_D] = m/\sum_{i=1}^m p_i$ tosses of the m coins. After applying the required variable transformation, we then consider $f(\tilde{\boldsymbol{p}}) = \frac{1}{\sum_{i=0}^{m-1}\tilde{p}_i}(\tilde{p}_0, \ldots, \tilde{p}_m)$ and we can then employ our Dice Enterprise methodology. In this particular problem, $f(\tilde{\boldsymbol{p}})$ is already a multivariate ladder and the transition matrix of the chain constructed as in Proposition 3.4 is given by

$$P = \begin{pmatrix} 1 - \sum_{i\neq 0}\tilde{p}_i & \tilde{p}_1 & \tilde{p}_2 & \ldots & \tilde{p}_{m+1} \\ \tilde{p}_0 & 1 - \sum_{i\neq 1}\tilde{p}_i & \tilde{p}_2 & \ldots & \tilde{p}_{m+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{p}_0 & \tilde{p}_1 & \tilde{p}_2 & \ldots & 1 - \sum_{i\neq m}\tilde{p}_i \end{pmatrix}$$

Notice that CFTP terminates as soon as either $0, 1, \ldots, m-1$ is rolled and continues only when the outcome of the roll is the m^{th} face. In this case, each iteration of CFTP is independent of the other and the probability that the algorithm stops is given by

$$\mathbb{P}(\text{CFTP stops}) = 1 - \frac{1}{m}\sum_{i=1}^{m}\mathbb{P}(\text{i}^{\text{th}} \text{ coin returns tails}) = \frac{1}{m}\sum_{i=0}^{m-1} p_i$$

so that $\mathbb{E}[N_{CFTP}] = m/\sum_{i=0}^{m-1} p_i$ and the algorithm is actually equivalent to the one proposed in [7].

5. Conclusions

The Dice Enterprise algorithm introduced in this paper is a generalisation of the celebrated Bernoulli Factory algorithms to rational mappings of categorical distributions. It offers a fully automated procedure that does not require further user intervention or case specific design tweaks. Furthermore, in the "coin to dice" case the efficiency of the algorithm can be automatically boosted by increasing the degree of the target polynomials until the distribution is log-concave which guarantees fast convergence. The version we developed is based on Coupling From the Past and enjoys an efficient monotonic implementation in the "coin to dice" case, however CFTP can be replaced by any other Markov chain perfect sampling routine, including Fill's interruptible algorithm. We demonstrated that several specialised Bernoulli factory algorithms introduced in literature, such as the two coin algorithm, the logistic Bernoulli factory or the Bernoulli race can be regarded as special versions of the Dice Enterprise. A natural open problem that follows from this paper is to design a monotone version of the Dice Enterprise in the "dice to dice" scenario. Further studies may also look into providing bounds for the degree of the decomposition of rational functions into ladders (based on Pólya positive homogenous polynomial theorem [33, 34]) and the number of Bernoulli trials needed to introduce log-concavity when convoluted with a discrete random variable.

6. Acknowledgments

We thank Susanna Brown, Oliver Johnson and Krzysztof Oleszkiewicz for helpful discussions. KŁ acknowledges funding form the Royal Society via the University Research Fellowship scheme. AW has been supported by EPSRC and GM has been supported by EPSRC through the OxWaSP Programme. Finally, AW and KŁ thank the Warwick Undergraduate Research Scholarship Scheme for supporting the project in its initial stages in the summer of 2013.

Appendix A: Appendix

Define the scaled by d discrete m dimensional simplex as

$$\Lambda_d^m = \left\{ \boldsymbol{n} = (n_0, \ldots, n_m) \in \{0, 1, \ldots, d\}^{m+1} : \sum_{i=0}^{m} n_i = d \right\}.$$

Proof of Proposition 2.14

By construction π' is a fine ladder on $\Omega' = \{0, \dots, w\}$. Since one augmentation operation yields a ladder sampling from which is equivalent to sampling from π, so does d-fold augmentation. It remains to show that π' is connected. To this end note that each state of π' is of the form $C\pi_i(\boldsymbol{p})\binom{d}{\boldsymbol{n}}\prod_{l=0}^{m} p_l^{n_l}$ for some constant C, some $i \in \{0, ..., k\}$ and some $\boldsymbol{n} \in \Lambda_d^m$. Define sets $A_0, \dots, A_k$ as

$$A_i = \{a \in \Omega' : \pi'_a(\boldsymbol{p}) = C_a \pi_i(\boldsymbol{p}) \binom{d}{\boldsymbol{n}} \prod_{l=0}^{m} p_l^{n_l} \text{ for some } \boldsymbol{n} \in \Lambda_d^m,\ C_a \in \mathbb{R}\}. \quad (19)$$

First notice that for a fixed i all the states in A_i are connected by construction due to $d-$fold augmentation. It is then enough to show that $A_i \cap A_j \neq \emptyset$ for all $j \neq i$. Indeed, let $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ be the degree of $\pi_i(\boldsymbol{p})$ and $\pi_j(\boldsymbol{p})$ respectively. Then, the numerators of $\pi_j(\boldsymbol{p})\binom{d}{\boldsymbol{n}_i}\prod_{l=0}^{m} p_l^{n_{i,l}}$ and $\pi_i(\boldsymbol{p})\binom{d}{\boldsymbol{n}_j}\prod_{l=0}^{m} p_l^{n_{j,l}}$ have the same degree $\boldsymbol{n}_i + \boldsymbol{n}_j$ and the respective state $a \in \Omega'$ with probability $\pi_{a'}(\boldsymbol{p})$ of degree $\boldsymbol{n}_i + \boldsymbol{n}_j$ satisfies $a \in A_i \cap A_j$. ☐

Lemma A.1 (Pólya [33])**.** *Let $f : \Delta^m \to \mathbb{R}$ be a homogeneous and positive polynomial in the variables $p_0, \dots, p_m$, i.e. all the monomials of the polynomial have the same degree. Then, for all sufficiently large n, all the coefficients of $(p_0 + \ldots + p_m)^n f(p_0, \dots, p_m)$ are positive.*

Lemma A.2. *Let $f : \Delta^m \to (0, 1)$ be a rational function over $\mathbb{R}$. Then, there exist homogeneous polynomials*

$$d(\boldsymbol{p}) = d(p_0, \dots, p_m) = \sum_{\boldsymbol{n} \in \Lambda_d^m} d_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j},$$

$$e(\boldsymbol{p}) = e(p_0, \dots, p_m) = \sum_{\boldsymbol{n} \in \Lambda_d^m} e_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j},$$

where $d_{\boldsymbol{n}}$ and $e_{\boldsymbol{n}}$ are real coefficients such that $0 \leq d_{\boldsymbol{n}} \leq e_{\boldsymbol{n}}$ and $f(\boldsymbol{p}) = d(\boldsymbol{p})/e(\boldsymbol{p})$. We will refer to d as the degree of the decomposition.

Proof. The lemma is a variation of Lemma 2.7 of [28], where $m = 1$ and coefficients are integers, and the proof follows the reasoning therein.

As $f(\boldsymbol{p})$ is a rational function, it may be written as

$$f(\boldsymbol{p}) = \frac{\overline{D}(\boldsymbol{p})}{\overline{E}(\boldsymbol{p})},$$

and we can assume that $\overline{D}(\boldsymbol{p})$ and $\overline{E}(\boldsymbol{p})$ are relatively prime polynomials. Since $f(\boldsymbol{p}) \in (0, 1)$ for all $\boldsymbol{p} \in \Delta^m$ and $\overline{D}(\boldsymbol{p})$ does not share any common root with $\overline{E}(\boldsymbol{p})$, it follows that $\overline{D}(\boldsymbol{p})$ and $\overline{E}(\boldsymbol{p})$ do not change sign in Δ^m so that we can assume without loss of generality that $\overline{D}(\boldsymbol{p})$ and $\overline{E}(\boldsymbol{p})$ are positive polynomials.

Let d_0 be the maximum degree of the polynomials $\overline{D}(\boldsymbol{p})$ and $\overline{E}(\boldsymbol{p})$. A general representation of the polynomials is given by

$$\overline{D}(\boldsymbol{p}) = \sum_{i=0}^{d} \sum_{\boldsymbol{n} \in \Lambda_i^m} a_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j}, \qquad \overline{E}(\boldsymbol{p}) = \sum_{i=0}^{d} \sum_{\boldsymbol{n} \in \Lambda_i^m} b_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j}.$$

Notice that in general $\overline{D}(\boldsymbol{p})$ and $\overline{E}(\boldsymbol{p})$ are not homogeneous polynomials, but it is possible to increase the degree of each term of the summation to be equal to d_0. In fact, since $p_0 + \ldots + p_m = 1$, one can use the multinomial theorem to define homogeneous polynomials $D(\boldsymbol{p})$ and $E(\boldsymbol{p})$ as

$$\begin{aligned}
\overline{D}(\boldsymbol{p}) &= \sum_{i=0}^{d} \sum_{\boldsymbol{n} \in \Lambda_i^m} a_{\boldsymbol{n}} (p_0 + \ldots + p_m)^{d-i} \prod_{j=0}^{m} p_j^{n_j} \\
&= \sum_{i=0}^{d} \sum_{\boldsymbol{n} \in \Lambda_i^m} \sum_{\boldsymbol{n}' \in \Lambda_{d-i}^m} a_{\boldsymbol{n}} \binom{d-i}{\boldsymbol{n}'} \prod_{j=0}^{m} p_j^{n_j + n_j'} \\
&= \sum_{\boldsymbol{n} \in \Lambda_d^m} d_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j} =: D(\boldsymbol{p}),
\end{aligned}$$

where

$$d_{\boldsymbol{n}} = \sum_{i=0}^{d} \sum_{\tilde{\boldsymbol{n}} \in \Lambda_i^m} \sum_{\boldsymbol{n}' \in \Lambda_{d-i}^m : \tilde{\boldsymbol{n}} + \boldsymbol{n}' = \boldsymbol{n}} a_{\tilde{\boldsymbol{n}}} \binom{d-i}{\boldsymbol{n}'}.$$

Analogously

$$\overline{E}(\boldsymbol{p}) = \sum_{i=0}^{d} \sum_{\boldsymbol{n} \in \Lambda_i^m} b_{\boldsymbol{n}} (p_0 + \ldots + p_m)^{d-i} \prod_{j=0}^{m} p_j^{n_j} = \sum_{\boldsymbol{n} \in \Lambda_d^m} e_{\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j} := E(\boldsymbol{p}).$$

Notice that $D(\boldsymbol{p})$ and $E(\boldsymbol{p})$ are positive polynomials. Moreover, since $f(\boldsymbol{p}) < 1$, it follows that also $E(\boldsymbol{p}) - D(\boldsymbol{p})$ is a positive polynomial. Therefore, by Lemma A.1 there exists a sufficiently large n, such that the polynomials $d(\boldsymbol{p}) = (p_0 + \ldots + p_m)^n D(\boldsymbol{p})$, $e(\boldsymbol{p}) = (p_0 + \ldots + p_m)^n E(\boldsymbol{p})$ and $e(\boldsymbol{p}) - d(\boldsymbol{p})$, all have positive coefficients. Hence, as required, $0 \leq d_{\boldsymbol{n}} \leq e_{\boldsymbol{n}}$ and

$$f(\boldsymbol{p}) = \frac{\overline{D}(\boldsymbol{p})}{\overline{E}(\boldsymbol{p})} = \frac{D(\boldsymbol{p})}{E(\boldsymbol{p})} = \frac{d(\boldsymbol{p})}{e(\boldsymbol{p})}. \tag{20}$$

The degree of the decomposition is therefore $d = d_0 + n$. □

Proof of Theorem 3.1

Since $f(\boldsymbol{p} = (f_0(\boldsymbol{p}), \ldots, f_v(\boldsymbol{p}))$ is a rational function, we can apply Lemma A.2 to each $f_i(\boldsymbol{p})$ and write

$$f(\boldsymbol{p}) = \left(\frac{d_0(\boldsymbol{p})}{e_0(\boldsymbol{p})}, \frac{d_1(\boldsymbol{p})}{e_1(\boldsymbol{p})}, \ldots, \frac{d_v(\boldsymbol{p})}{e_v(\boldsymbol{p})} \right).$$

Let $C(\boldsymbol{p})$ be the lowest common multiple of the denominators $e_i(\boldsymbol{p})$ and express $f(\boldsymbol{p})$ as

$$f(\boldsymbol{p}) = \frac{1}{C(\boldsymbol{p})}(g_0(\boldsymbol{p}), \dots, g_v(\boldsymbol{p})).$$

Assume w.l.o.g. that each polynomial $g_i(\boldsymbol{p})$ has degree d (if this is not the case, let d_i be the degree of $g_i(\boldsymbol{p})$ and multiply it by $(p_0 + \ldots + p_m)^{d-d_i}$) and write

$$\frac{g_i(\boldsymbol{p})}{C(\boldsymbol{p})} = \frac{1}{C(\boldsymbol{p})} \sum_{\boldsymbol{n}\in\Lambda_d^m} a_{i,\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j}. \tag{21}$$

Having applied Lemma A.2 it follows $a_{i,\boldsymbol{n}} \geq 0$ for all $i \in \{0, \dots, v\}$, $\boldsymbol{n} \in \Lambda_d^m$. Therefore, we can construct a distribution $\pi' : \Delta^m \to \Delta^w$ on $\Omega' = \{0, \dots, w\}$, where $w < (v+1)\binom{d+m}{m}$ and where each state is one term of the summation in (21) for a fixed i and thus of the form $\frac{1}{C(\boldsymbol{p})} a_{i,\boldsymbol{n}} \prod_{j=0}^{m-1} p_j^{n_j}$.

By construction π' is a disaggregation of f. Indeed, consider v sets $A_0, \dots, A_v$ defined as

$$A_i = \{a \in \Omega' : \pi'_a(\boldsymbol{p}) = \frac{1}{C(\boldsymbol{p})} a_{i,\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j} \text{ for a } \boldsymbol{n} \in \Lambda_d^m\}.$$

It then follows

$$f_i(\boldsymbol{p}) = \frac{g_i(\boldsymbol{p})}{C(\boldsymbol{p})} = \sum_{h\in A_i} \pi'_h(\boldsymbol{p}) = \frac{1}{C(\boldsymbol{p})} \sum_{\boldsymbol{n}\in\Lambda_d^m} a_{i,\boldsymbol{n}} \prod_{j=0}^{m} p_j^{n_j}.$$

By discarding any null term in $\pi'(\boldsymbol{p})$, it follows that π' is a multivariate ladder. Finally, via Proposition 2.14 we construct a fine and connected multivariate ladder $\pi : \Delta^m \to \Delta^k$ where $k < \min\{(w+1)(m+1)^d, \binom{2d+m}{m}\}$, such that sampling from each f, π' and π is equivalent.

□

Proof of Proposition 3.4

We shall prove the result by showing that P is a stochastic matrix and that the detailed balance condition is satisfied for all $\boldsymbol{p} \in \Delta^m$. Recall that the off-diagonal elements of P are given by the off-diagonal elements of $V \circ W$ where $\circ$ denotes the entrywise product, W is defined in equation (9) and V is the output of Algorithm 5. We first prove that

$$\sum_{j\in\mathcal{N}_b(i)} V_{i,j} \leq 1, \quad \forall b \in \{0, \dots, m\}, i \in \Omega.$$

Notice that by how the weights $\mathcal{W}_b(i)$ are defined within the algorithm, we have $\sum_{j\in\mathcal{N}_b(i)} V_{i,j} = \mathcal{W}_b(i)$.

Having fixed i and b, assume that one of the $V_{i,j}$, where $j \in \mathcal{N}_b(i)$, is obtained in line 6 of the algorithm. Denote by $\mathcal{W}^\star_b(i)$ the new value of $\mathcal{W}_b(i)$ after it has been updated for all $j \in \mathcal{N}_b(i)$. It follows

$$\mathcal{W}^\star_b(i) = \mathcal{W}_b(i) + \sum_{j\in\mathcal{N}_b(i)} \frac{R_j}{\mathcal{S}_b(i)} = \mathcal{W}_b(i) + \sum_{j\in\mathcal{N}_b(i)} \frac{R_j}{\sum_{h\in\mathcal{N}_b(i)} R_h}(1-\mathcal{W}_b(i)) = 1,$$

where the value of $S_b(i)$ is given in line 9 of the algorithm. At this point the algorithm has assigned a value to $V_{i,j}$ for all $j \in \mathcal{N}_b(i)$ and thus $\sum_{j\in\mathcal{N}_b(i)} V_{i,j} = \mathcal{W}^\star_b(i) = 1$.

Assume now that all the $V_{i,j}$ for $j \in \mathcal{N}_b(i)$ have been assigned in line 8 of the algorithm. For fixed i, we then have that $j \in \mathcal{N}_b(i)$ and let $d \in \{0,\dots,m\}$ such that $i \in \mathcal{N}_d(j)$. Then $V_{j,i}$ is assigned in line 6 of the algorithm. Denote the new value of $\mathcal{W}_b(i)$ assigned in line 8 of the algorithm as $\mathcal{W}^\star_b(i)$. It follows

$$\mathcal{W}^\star_b(i) = \mathcal{W}_b(i) + \frac{R_j}{\mathcal{S}_d(j)} \le \mathcal{W}_b(i) + \frac{R_j}{\mathcal{S}_b(i)} \le \mathcal{W}_b(i) + \sum_{j\in\mathcal{N}_b(i)} \frac{R_j}{\mathcal{S}_b(i)} = 1,$$

where the fact that $\mathcal{S}_d(j) \ge \mathcal{S}_b(i)$ follows by the fact that b and i are chosen in line 4 of the algorithm to maximise $\mathcal{S}_b(i)$. The value of $\mathcal{W}_b(i)$ will then always be less or equal than 1, so that $\sum_{j\in\mathcal{N}_b(i)} V_{i,j} = \mathcal{W}_b(i) \le 1$.

We then have

$$\sum_{\substack{j=0\\ j\neq i}}^{k} P_{i,j} = \sum_{b=0}^{m} \sum_{j\in\mathcal{N}_b(i)} V_{i,j} p_b \le \sum_{b=0}^{m} p_b = 1,$$

as required. It is now enough to prove that $\pi(\boldsymbol{p})$ satisfies the detailed balance condition for all $\boldsymbol{p} \in \Delta^m$. If $j \notin \mathcal{N}(i)$, then $P_{i,j} = P_{j,i} = 0$ and the balance condition is trivially satisfied. For $j \in \mathcal{N}(i)$ we have

$$\frac{\pi_j(\boldsymbol{p})}{\pi_i(\boldsymbol{p})} = \frac{R_j \prod_{h=0}^m p_h^{n_{j,h}}}{R_i \prod_{h=0}^m p_h^{n_{i,h}}} = \frac{R_j p_b}{R_i p_c},$$

and by equation (9), $W_{i,j}/W_{j,i} = p_b/p_c$. The fact that $V_{i,j}/V_{j,i} = R_j/R_i$ follows directly from how these values are assigned in the algorithm for the pair i, j in lines 6 and 8. Given the connectedness condition, $\pi(\boldsymbol{p})$ is also the unique limiting distribution. $\square$

Proof of Proposition 3.6

By contradiction, assume that there exists a different reversible Markov chain with transition matrix Q that has the same adjacency structure and stationary distribution as the P-chain, and such that $Q \succeq_P P$. It follows that also Q has a similar decomposition as in equation (6) and the off-diagonal elements of Q will be the same as the entries of $\tilde{V} \circ W$, where $\circ$ denotes the entrywise product and with W as in equation (9), while $\tilde{V}$ is a matrix of real numbers. Since $Q \succeq_P P$ and $Q \neq P$, there must exist indices i, j such that $\tilde{V}_{i,j} > V_{i,j}$. We distinguish two cases:

- The value of $V_{i,j}$ is assigned in line 6 of Algorithm 5. Then, let $b \in \{0,\dots,m\}$ such that $j \in \mathcal{N}_b(i)$ and notice that by how the algorithm is designed we have $\sum_{j\in\mathcal{N}_b(i)} V_{i,j} = 1$ (cf. proof of Proposition 3.4). Therefore $\sum_{j\in\mathcal{N}_b(i)} \tilde{V}_{i,j} > 1$. We reach a contradiction by observing

$$\sum_{\substack{j=0\\ j\neq i}}^{k-1} Q_{i,j} = \sum_{c=0}^{m} \sum_{j\in\mathcal{N}_c(i)} \tilde{V}_{i,j} p_c \xrightarrow{p_b \to 1} \sum_{j\in\mathcal{N}_b(i)} \tilde{V}_{i,j} > 1.$$

- The value of $V_{i,j}$ is assigned in line 8 of Algorithm 5. Since the Q-chain is reversible, it follows that also $\tilde{V}_{j,i} > V_{j,i}$. However, the value of $V_{j,i}$ is assigned in line 6 of the algorithm and we reach the same contradiction as before.

$\square$

Proof of Proposition 3.7

Fix a state $i \in \Omega$ and notice that if $j \notin \mathcal{N}(i)$, then $\mathbb{P}(\phi(i,B,U)=j) = P_{i,j} = 0$. For any outcome $b \in \{0,\dots,m\}$ on the die, recall $\mathcal{N}_b(i) = \{j_0,\dots,j_w\}$, is the set of states accessible from i. It follows for any $j_l \in \mathcal{N}_b(i)$ that

$$\begin{aligned}\mathbb{P}(\phi(i,B,U)=j_l) &= \mathbb{P}\left(B=b, \sum_{h=0}^{l-1} V_{i,j_h} < U \le \sum_{h=0}^{l} V_{i,j_h}\right)\\ &= p_b \mathbb{P}\left(U \le V_{i,j_l}\right) = P_{i,j_l}.\end{aligned}$$

Hence, ϕ is an update function for the Markov chain $(X_t)_{t\in\mathbb{N}}$. $\square$

Proof of Corollary 3.8

Given a fine and connected ladder $\pi : (0,1) \to \Delta^k$ as in equation (5), for $1 \le i \le k-1$, we have

$$\begin{aligned}&\mathcal{N}_0(i) = \{i-1\}, \qquad \mathcal{N}_1(i) = \{i+1\}, \qquad \mathcal{N}(i) = \{i-1, i+1\},\\ &\mathcal{S}_0(i) = R_{i-1}, \qquad \mathcal{S}_1(i) = R_{i+1}.\end{aligned}$$

Then, the matrix W defined in equation (9) and the off-diagonal entries of the matrix V output by Algorithm 5 are given by

$$W_{i,j} = \begin{cases} p & \text{if } j = i+1\\ (1-p) & \text{if } j = i-1\\ 0 & \text{otherwise}\end{cases}, \qquad V_{i,j} = \begin{cases} \frac{R_{i+1}}{R_i \vee R_{i+1}} & \text{if } j = i+1\\ \frac{R_{i-1}}{R_{i-1}\vee R_i} & \text{if } j = i-1\\ 0 & \text{otherwise}\end{cases}$$

Therefore, the transition matrix P defined in (6) is equivalent to (11) and the update function defined in (10) is the same as (12).

To see why ϕ is a monotonic update function, consider $i \le j$. It is trivial to check that $\phi(i,B,U) \le \phi(j,B,U)$ if $j \ne i+1$. If $j = i+1$, the monotonic condition would not be satisfied only if $\phi(i, B, U) = i+1$ and $\phi(i+1, B, U) = i$. However, this can not happen as it would require B to be equal to 0 and 1 simultaneously. □

Proof of Theorem 3.11

Proof. Let $w_0, \dots, w_{n_0}$ be the probabilities of W on $\Omega = \{0, \dots, n_0\}$. We shall consider generating function $P(x) = \sum_{i=0}^{n_0} w_i x^i$. This function is a product of linear and quadratic functions, that is

$$P(x) = c \prod_{j=1}^{k_0} ((x - a_j)^2 + b_j^2) \prod_{l=1}^{l_0} (x + c_l),$$

where $b_j \neq 0$ and $c_l > 0$ (the latter follows from the fact that a polynomial with positive coefficients cannot have positive roots). Now, it suffices to show that for big n the sequences of coefficients generated by

$$Q_n(x) = ((x-a)^2 + b^2)(1+x)^n, \quad L_n(x) = (x+c)(1+x)^n, \text{ where } b \neq 0, c > 0,$$

is positive and log concave. Indeed, since convolution preserves positivity and log-concavity, and corresponds to summing random variables, we can find suitable binomial $B(n, 1/2)$ (whose generating function is precisely $\frac{1}{2^n}(1+x)^n$) for each factor of P separately. Note that we ignore normalizing constants, as positivity and log-concavity are not affected.

The rest is just an attempt to verify this. In case of L_n there is nothing to prove since the sequence generated by $x + c$ with $c > 0$ is $(c, 1, 0, \dots)$ and it is positive and log-concave itself. Since $(x-a)^2 + b^2 = x^2 - 2ax + a^2 + b^2$, the sequence generated by Q_n is

$$a_k = (a^2 + b^2)\binom{n}{k} - 2a\binom{n}{k-1} + \binom{n}{k-2}, \qquad k \geq 0.$$

Here we adapt the notation $\binom{n}{k} = 0$ for $k < 0$ and $k > n$. We first show that for big n this sequence is non-negative. The inequality $a_k \geq 0$ is equivalent to

$$(a^2 + b^2)(n-k+1)(n-k+2) - 2ak(n-k+2) + k(k-1) \geq 0.$$

Let us treat the left hand side as a polynomial in k. This is

$$k^2 \left(a^2 + 2a + b^2 + 1\right) + k\left((-2n-3)\left(a^2 + b^2\right) - 2an - 4a - 1\right) \\ + (n+1)(n+2)\left(a^2 + b^2\right).$$

Since the coefficient in front of k^2 is positive, we can hope to find n such that this polynomial is positive for all real k. For this the Δ of this quadratic form should be negative. We have

$$\begin{aligned}\Delta &= \left((-2n-3)\left(a^2+b^2\right) - 2an - 4a - 1\right)^2 \\ &\qquad - 4(n+1)(n+2)\left(a^2+b^2\right)\left(a^2+2a+b^2+1\right) \\ &= -4b^2n^2 + 4(a + 2a^2 + a^3 - 2b^2 + ab^2)n \\ &\qquad + (1 + 8a + 14a^2 + 8a^3 + a^4 - 2b^2 + 8ab^2 + 2a^2b^2 + b^4).\end{aligned}$$

As we can see the leading term is $-4b^2n^2$ and so for big n we get $\Delta < 0$.

We now show that for big n the sequence a_k is strictly log-concave; i.e., $a_k^2 > a_{k+1}a_{k-1}$. This is trivially true for $k = 0$ and $k = n+2$, but it is also easily verified for $k \in \{1, n-1, n, n+1\}$ by just substituting the value of k and letting $n \to \infty$.

To prove the result for $k \in \{2, \ldots, n-2\}$, rewrite the coefficients a_k as:

$$a_k = \binom{n}{k-1}\left[\frac{n-k+1}{k}(a^2+b^2) + \frac{k-1}{n-k+2} - 2a\right].$$

The inequality $a_k^2 > a_{k+1}a_{k-1}$ reduces to

$$\begin{aligned}\binom{n}{k-1}^2 &\left[\frac{n-k+1}{k}(a^2+b^2) + \frac{k-1}{n-k+2} - 2a\right]^2 \\ &> \binom{n}{k}\left[\frac{n-k}{k+1}(a^2+b^2) + \frac{k}{n-k+1} - 2a\right] \\ &\quad \times \binom{n}{k-2}\left[\frac{n-k+2}{k-1}(a^2+b^2) + \frac{k-2}{n-k+3} - 2a\right],\end{aligned}$$

This is

$$\begin{aligned}\frac{k}{k-1}\cdot\frac{n-k+2}{n-k+1} &\left[\frac{n-k+1}{k}(a^2+b^2) + \frac{k-1}{n-k+2} - 2a\right]^2 \\ &> \left[\frac{n-k}{k+1}(a^2+b^2) + \frac{k}{n-k+1} - 2a\right] \\ &\quad \times \left[\frac{n-k+2}{k-1}(a^2+b^2) + \frac{k-2}{n-k+3} - 2a\right].\end{aligned}$$

To deal with it we rewrite it slightly.

$$\begin{aligned}&\left[\frac{n-k+1}{k}(a^2+b^2) + \frac{k-1}{n-k+2} - 2a\right]^2 \\ &\quad > \left[\frac{(n-k)}{(k+1)}(a^2+b^2) + \frac{k}{n-k+1} - 2a\right] \\ &\quad\quad \times \left[\frac{n-k+1}{k}(a^2+b^2) + \frac{(k-2)(k-1)(n-k+1)}{(n-k+3)(n-k+2)k}\right. \\ &\quad\quad\quad \left. -2a\frac{(k-1)(n-k+1)}{(n-k+2)k}\right].\end{aligned}$$

For big n and fixed a, b, the right hand side is a product of two positive factors. We shall take the square root of both sides and use the inequality $2\sqrt{xy} \leq x+y$

to bound the right hand side. Then, it is enough to verify:

$$
\begin{aligned}
&2\left[\frac{n-k+1}{k}(a^2+b^2)+\frac{k-1}{n-k+2}-2a\right]\\
&\quad>\left[\frac{(n-k)}{(k+1)}(a^2+b^2)+\frac{k}{n-k+1}-2a\right]\\
&\qquad+\left[\frac{n-k+1}{k}(a^2+b^2)+\frac{(k-2)(k-1)(n-k+1)}{(n-k+3)(n-k+2)k}\right.\\
&\qquad\qquad\left.-2a\frac{(k-1)(n-k+1)}{(n-k+2)k}\right].
\end{aligned}
$$

Rewrite it by taking the RHS to the LHS and collecting common factors.

$$
\begin{aligned}
&\left[\frac{(n-k+1)}{k}-\frac{n-k}{k+1}\right](a^2+b^2)+\frac{2(k-1)}{n-k+2}-\frac{k}{n-k+1}\\
&\quad-\frac{(k-2)(k-1)(n-k+1)}{(n-k+3)(n-k+2)k}-\left[1-\frac{(k-1)(n-k+1)}{(n-k+2)k}\right]2a>0.
\end{aligned}
$$

Notice:

- $\displaystyle \frac{(n-k+1)}{k}-\frac{n-k}{k+1}=\frac{n+1}{k(k+1)}$,
- $\displaystyle \frac{2(k-1)}{n-k+2}-\frac{k}{n-k+1}-\frac{(k-2)(k-1)(n-k+1)}{(n-k+3)(n-k+2)k}= -\frac{(1+k+(k-1)^2+n-(k-1)n)(n+1)}{(n-k+1)(n-k+2)(n-k+3)k}$,
- $\displaystyle 1-\frac{(k-1)(n-k+1)}{(n-k+2)k}=\frac{n+1}{(n-k+2)k}$.

Thus, by taking the common denominator, it is enough to verify $P_{a,b}(k,n)>0$, where

$$
\begin{aligned}
P_{a,b}(k,n)&=(a^2+b^2)(n-k+3)(n-k+2)(n-k+1)\\
&\quad-\left(1+k+(k-1)^2+n-(k-1)n\right)(k+1)\\
&\quad-2a(k+1)(n-k+3)(n-k+1).
\end{aligned}
$$

This is a polynomial of degree three in k. The discriminant of a cubic polynomial Ak^3+Bk^2+Ck+D is given by

$$\Delta=B^2C^2-4AC^3-4B^3D-27A^2D^2+18ABCD,$$

and is negative if there are two conjugate complex and one real roots.

In our case the discriminant of $k\to P_{a,b}(k,n)$ is

$$\Delta(n,a,b)=-4b^2n^6+O(n^5),$$

and so for big n it is negative (recall that $b \neq 0$). We conclude that there is only one real root. Notice that

$$P_{a,b}(2,n) = (a^2+b^2)n^3 + O(n^2),$$
$$P_{a,b}(n-2,n) = n^2 + O(n),$$

so that for n big enough, $P_{a,b}(k,n) > 0$ for all $k \in [2, n-2]$ as desired.

□

Proof of Proposition 3.12

Proof. Augment the ladder d times to construct a new ladder $\pi' : \Delta^m \to \Delta^w$, where $w < \min\{(k+1)(m+1)^d, \binom{2d+m}{m}\}$. We showed in Proposition 2.14 that π' is a fine and connected ladder and that we can define sets $A_0, \dots, A_k$ as in equation (19). We now show that for any state $a \in \Omega'$, it is always possible to move to a different state if $b \in E$ is rolled (except from the state proportional to $p_b^{2d}/C(\boldsymbol{p})$). Fix a state $a \in \Omega'$ in the set A_i, therefore of the form

$$C_a \pi_i(\boldsymbol{p}) \binom{d}{\boldsymbol{n}} \boldsymbol{p}^{\boldsymbol{n}}, \quad \text{for some } \boldsymbol{n} \in \Lambda_d^m.$$

If $\boldsymbol{p}^{\boldsymbol{n}} \neq p_b^d$, then there exists another state $a' \in A_i$ connected to a and such that $n'_{a',b} = n'_{a,b} + 1$ and the chain may move to it. We showed in the proof of Proposition 2.14 that $A_i \cap A_j \neq \emptyset, \forall j \neq i$. Therefore, if $\boldsymbol{p}^{\boldsymbol{n}} = p_b^d$ there exists a connected state a' in $A_j \neq A_i$ such that $n'_{a',b} = n'_{a,b}+1$, unless $\pi_i(\boldsymbol{p}) \propto p_b^{2d}/C(\boldsymbol{p})$.

Now, consider applying CFTP on the ladder π' using the transition matrix of Proposition 3.4 and the update function of Proposition 3.7. We prove the bound by considering sets of moves that, regardless of the starting point, end up in a singleton. Let a be the minimum of the entries of the matrix V, as produced by Algorithm 5. This choice of a allows us to conclude that whenever we draw $U < a$ in the CFTP algorithm and $B \in E$, then all the tracked particles move, except the particles in the state proportional to $p_b^{2d}/C(\boldsymbol{p})$. Therefore if such event happens on $2d$ consecutive iterations, then the algorithm necessarily ends as all the particles must have coalesced in the state proportional to $p_b^{2d}/C(\boldsymbol{p})$. That is, if $u_1 \le a, \dots, u_{2d} \le a$ we can write

$$\phi_{2d}(i, (b, \dots, b), (u_1, \dots, u_{2d})) = \{a\}, \qquad \forall i \in \{0, \dots, w\},$$

where $a \in \Omega'$ is the state of the ladder proportional to $p_b^{2d}/C(\boldsymbol{p})$. Let τ_b be the number of iterations required for this event to happen for the first time. The probability generating function of τ_b is given by

$$\begin{aligned} f_{\tau_b}(x) &= \sum_{j=0}^{\infty} (ap_b)^{2d} x^{2d} \left[(1-ap_b)x + \ldots + (ap_b)^{2d-1}(1-ap_b)x^{2d} \right]^j \\ &= \frac{x^{2d}(ap_b)^{2d}(ap_b x - 1)}{ap_b x (ap_b x)^{2d} - x(ap_b x)^{2d} + x - 1}, \end{aligned}$$

so that

$$\mathbb{E}[\tau_b] = f'_{\tau_b}(1) = \frac{(ap_b)^{-2d} - 1}{1 - ap_b}.$$

Since the number of required rolls N equals the number of iterations of the algorithm, it follows that $N \leq \tau_b$. The same reasoning holds for all $b \in E$, so that we conclude:

$$\mathbb{E}[N] \leq \min_{b \in E} \mathbb{E}[\tau_b] = \min_{b \in E} \frac{(ap_b)^{-2d} - 1}{1 - ap_b}.$$

□

Proof of Corollary 3.14

Proof. Follows by Proposition 3.12 by noticing that in the case $m = 1$, we necessarily have $E = \{0, 1\}$. □

Proof of Proposition 3.15

Proof. Requiring π to be strictly log-concave is equivalent to have $R_i^2 > R_{i-1}R_{i+1}$ for all $i \in \{1, \dots, k-1\}$ by equation (5). In turn, this implies

$$\frac{R_i}{R_{i-1} \vee R_i} \geq \frac{R_{i+1}}{R_i \vee R_{i+1}}, \quad \frac{R_i}{R_i \vee R_{i+1}} \geq \frac{R_{i-1}}{R_{i-1} \vee R_i}, \tag{22}$$

so that $\rho \leq 1$ since $P_{i,i+1} \geq P_{i+1,i+2}$ and $P_{i+1,i} \geq P_{i,i-1}$. However, given $p \in (0, 1)$, it cannot be that $\rho = 1$. Indeed, this could happen only if $P_{i,i+1} = P_{i+1,i+2}$ and $P_{i+1,i} = P_{i,i-1}$. However, this would imply either $R_i^2 = R_{i-1}R_{i+1}$ or $R_{i+1}^2 = R_i R_{i+2}$ thus contradicting strict log-concavity. We then conclude that $\rho \in (0, 1)$ for all $p \in (0, 1)$.

Denote by X_t^i the chain at time t given that it started in state i. Monotonic CFTP (cf. Algorithm 4) tracks backwards in time the trajectories of the coupled chains X_t^0 and X_t^k and stops when the two coalesce. Following the notation of [35], let $T_\star$ be the time this happens and, to ease the analysis, define $T^\star$ as the smallest time such that $X_t^0 = X_t^k$, where the chains are now tracked forwards in time. Notice that $T_\star$ and $T^\star$ have the same distribution and that the number of tosses N required by the algorithm equals $T_\star$.

Define $D_t^{i,j} = |X_t^i - X_t^j|$ as the distance between two coupled particles started at states i and j after t steps. In particular, focus on the distance $D_t^{i,i+1}$ between two particles started at consecutive states. At each step a p-coin is tossed and a uniform random variable is drawn so that the trajectories of the two chains can be tracked in a coupled fashion. In particular, given equation (22), we have that the two particles started at states i and $(i+1)$ can in one step either stay still, coalesce in state i or state $(i+1)$, move to states $(i+1)$ and $(i+2)$ or move to states $(i-1)$ and i respectively. Therefore, after one step the distance between the two coupled and consecutive particles can either decrease by 1 or remain the same:

$$D_1^{i,i+1} = \begin{cases} 0 & \text{with probability } (P_{i,i+1} - P_{i+1,i+2}) + (P_{i+1,i} - P_{i,i-1}) \\ 1 & \text{with probability } 1 - (P_{i,i+1} - P_{i+1,i+2}) - (P_{i+1,i} - P_{i,i-1}) \end{cases}$$

where the transition probabilities $P_{i,j}$ are given in equation (11). Denote by

$$\rho_{i,i+1} = 1 - (P_{i,i+1} - P_{i+1,i+2}) - (P_{i+1,i} - P_{i,i-1}),$$

so that $\mathbb{E}[D_1^{i,i+1}] = \rho_{i,i+1}$. Let $\rho = \max_i \rho_{i,i+1}$ and notice that by conditioning on how the particles move on the first step and by the Markov property, it follows

$$\begin{aligned}\mathbb{E}[D_t^{i,i+1}] &= P_{i+1,i+2}\mathbb{E}[D_{t-1}^{i+1,i+2}] + P_{i,i-1}\mathbb{E}[D_{t-1}^{i-1,i}] + (1 - P_{i,i+1} - P_{i+1,i})\,\mathbb{E}[D_{t-1}^{i,i+1}] \\ &\leq (\mathbb{E}[D_{t-1}^{i-1,i}] \vee \mathbb{E}[D_{t-1}^{i,i+1}] \vee \mathbb{E}[D_{t-1}^{i+1,i+2}])\rho_{i,i+1} \\ &\leq \rho^t,\end{aligned}$$

where $\vee$ denotes the maximum between two numbers.

To conclude, notice that $\mathbb{P}(T^\star \geq t) = \mathbb{P}(D_t^{0,k} \geq 1)$. It then follows by Markov's inequality and the result above that

$$\mathbb{P}(T^\star \geq t) = \mathbb{P}(D_t^{0,k} \geq 1) \leq \mathbb{E}[D_t^{0,k}] = \sum_{i=0}^{k-2} \mathbb{E}[D_t^{i,i+1}] \leq (k-1)\rho^t,$$

as desired. □

Proof of Lemma 3.17

Proof. Note that a univariate ladder is log-concave if its coefficients R_i define a log-concave sequence. Then, let R be a random variable on $\{0, \ldots, k\}$ having p.m.f. proportional to the coefficients R_i of the ladder π, that is such that $\mathbb{P}(R = i) \propto R_i$. Moreover, let n be such that $Z = R+B_n$ is strictly log-concave, as stated in Theorem 3.11. Consider $\pi' : (0,1) \to \Delta^{k+n}$, an n-fold augmentation of π. As noticed in Remark 2.13, $Y \sim \pi'(p)$ has the same distribution as $\pi + \mathrm{Bin}(n,p)$ and the coefficients R_i's of the ladder π' are proportional to $\mathbb{P}(Z = i)$. The desired result holds by noticing that multiplication by a constant preserves log-concavity. □

References

[1] Asmussen, S., Glynn, P. W. and Thorisson, H. (1992). Stationarity detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **2** 130–157.

[2] Blanchet, J. and Meng, X. (2005). Exact sampling, regeneration and minorization conditions Technical Report, Tech. rep., Columbia University. URL http://www. columbia. edu/˜ b2814/papers/JSMsent. pdf.

[3] Blanchet, J. and Zhang, F. (2017). Exact Simulation for Multivariate Itô Diffusions. *ArXiv e-prints*.

[4] Bubley, R. and Dyer, M. (1997). Path coupling: A technique for proving rapid mixing in Markov chains. In *Proceedings 38th Annual Symposium on Foundations of Computer Science* 223-231.

[5] Cai, Y., Oikonomou, A., Velegkas, G. and Zhao, M. (2019). An Efficient ε-BIC to BIC Transformation and Its Application to Black-Box Reduction in Revenue Maximization. *arXiv preprint arXiv:1911.10172*.

[6] Dale, H., Jennings, D. and Rudolph, T. (2015). Provable quantum advantage in randomness processing. *Nature communications* **6** 8203.

[7] Dughmi, S., Hartline, J. D., Kleinberg, R. and Niazadeh, R. (2017). Bernoulli factories and black-box reductions in mechanism design. In *STOC'17—Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* 158–169. ACM, New York. MR3678179

[8] Fill, J. A. (1998). An interruptible algorithm for perfect sampling via Markov chains. *Ann. Appl. Probab.* **8** 131–162. MR1620346

[9] Flajolet, P., Pelletier, M. and Soria, M. (2011). On Buffon machines and numbers. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms* 172–183. Society for Industrial and Applied Mathematics.

[10] Flegal, J. M. and Herbei, R. (2012). Exact sampling for intractable probability distributions via a Bernoulli factory. *Electron. J. Stat.* **6** 10–37. MR2879671

[11] Gonçalves, F. B., Łatuszyński, K. G. and Roberts, G. O. (2017). Exact Monte Carlo likelihood-based inference for jump-diffusion processes. *ArXiv e-prints*.

[12] Gonçalves, F. B., Łatuszyński, K. and Roberts, G. O. (2017). Barker's algorithm for Bayesian inference with intractable likelihoods. *Braz. J. Probab. Stat.* **31** 732–745. MR3738176

[13] Goyal, V. and Sigman, K. (2012). On simulating a class of Bernstein polynomials. *ACM Trans. Model. Comput. Simul.* **22** Art. 12, 5. MR2958668

[14] Henderson, S. G. and Glynn, P. W. (2003). Nonexistence of a class of variate generation schemes. *Oper. Res. Lett.* **31** 83–89. MR1957197

[15] Herbei, R. and Berliner, L. M. (2014). Estimating ocean circulation: an MCMC approach with approximated likelihoods via the Bernoulli factory. *J. Amer. Statist. Assoc.* **109** 944–954. MR3265667

[16] Hoggar, S. G. (1974). Chromatic polynomials and logarithmic concavity. *J. Combinatorial Theory Ser. B* **16** 248–254. MR0342424

[17] Holtz, O., Nazarov, F. and Peres, Y. (2011). New coins from old, smoothly. *Constr. Approx.* **33** 331–363. MR2784483

[18] Huber, M. (2016). Nearly optimal Bernoulli factories for linear functions. *Combin. Probab. Comput.* **25** 577–591. MR3506427

[19] Huber, M. (2017). Optimal linear Bernoulli factories for small mean problems. *Methodol. Comput. Appl. Probab.* **19** 631–645. MR3649562

[20] Huber, M. L. (2016). *Perfect simulation. Monographs on Statistics and Applied Probability* **148**. CRC Press, Boca Raton, FL. MR3443710

[21] Jacob, P. E. and Thiery, A. H. (2015). On nonnegative unbiased estimators. *Ann. Statist.* **43** 769–784. MR3319143

[22] Johnson, O. and Goldschmidt, C. (2006). Preservation of log-concavity on summation. *ESAIM Probab. Stat.* **10** 206–215. MR2219340

[23] Keane, M. and O'Brien, G. L. (1994). A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **4** 213–219.

[24] Łatuszyński, K., Kosmidis, I., Papaspiliopoulos, O. and Roberts, G. O. (2011). Simulating events of unknown probabilities via reverse time martingales. *Random Structures Algorithms* **38** 441–452. MR2829311

[25] Lee, A., Doucet, A. and Łatuszyński, K. (2014). Perfect simulation using atomic regeneration with application to Sequential Monte Carlo. *ArXiv e-prints*.

[26] Mendo, L. (2019). An asymptotically optimal Bernoulli factory for certain functions that can be expressed as power series. *Stochastic Process. Appl.* **129** 4366–4384. MR4013865

[27] Mira, A. (2001). Ordering and improving the performance of Monte Carlo Markov chains. *Statist. Sci.* **16** 340–350. MR1888449

[28] Mossel, E. and Peres, Y. (2005). New coins from old: computing with unknown bias. *Combinatorica* **25** 707–724. With an appendix by Christopher Hillar. MR2199432

[29] Nacu, Ş. and Peres, Y. (2005). Fast simulation of new coins from old. *Ann. Appl. Probab.* **15** 93–115. MR2115037

[30] Niazadeh, R. (2017). PhD Thesis: Algorithms Vs. Mechanisms.: Mechanism Design for Complex Environments, PhD thesis, Cornell University.

[31] Patel, R. B., Rudolph, T. and Pryde, G. J. (2019). An experimental quantum Bernoulli factory. *Science advances* **5** eaau6668.

[32] Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika* **60** 607–612. MR0362823

[33] Pólya, G. (1928). Über positive darstellung von polynomen. *Vierteljschr. Naturforsch. Ges. Zürich* **73** 141–145.

[34] Powers, V. and Reznick, B. (2001). A new bound for Pólya's Theorem with applications to polynomials positive on polyhedra. *Journal of Pure and Applied Algebra*.

[35] Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995)* **9** 223–252. MR1611693

[36] Saumard, A. and Wellner, J. A. (2014). Log-concavity and strong log-concavity: a review. *Stat. Surv.* **8** 45–114. MR3290441

[37] Schmon, S. M., Doucet, A. and Deligiannidis, G. (2019). Bernoulli Race Particle Filters. *arXiv preprint arXiv:1903.00939*.

[38] Vats, D., Roberts, G. O., Łatuszyński, K. and Gonçalves, F. B. (2019). Efficient Bernoulli factory MCMC for intractable likelihoods Technical Report.

[39] von Neumann, J. (1951). Various techniques used in connection with random digits. In *Monte Carlo Method* (A. S. Householder, G. E. Forsythe and H. H. Germond, eds.) 36–38. National Bureau of Standards Applied Mathematics Series, 12, Washington, D.C.: U.S. Government Printing Office.

[40] Yuan, X., Liu, K., Xu, Y., Wang, W., Ma, Y., Zhang, F., Yan, Z.,

Vijay, R., Sun, L. and Ma, X. (2016). Experimental quantum randomness processing using superconducting qubits. *Physical review letters* **117** 010502.