Linear-time exact sampling of sum-constrained random variables

Frédérique Bassino
frederique.bassino@univ-paris13.fr

Andrea Sportiello
andrea.sportiello@lipn.univ-paris13.fr

Université Paris 13
Sorbonne Paris Cité
LIPN, CNRS UMR 7030
93430 Villetaneuse, France

Abstract

In the problem of exactly-sampling from a probability distribution, one is often led to sample from a measure of the form $\mu(x_1, \dots, x_n) \propto \prod_i f_i(x_i) \times \delta_{x_1+\dots+x_n,m}$, with f_i's integer-valued distributions. When the f_i's are all equal, an algorithm of Devroye essentially solves this problem in linear time. However, in several important cases these f_i's are in the same family, but have distinct parameters. A typical example is the sampling of random set partitions, related to a set of f_i's which are geometric distributions with different averages. We describe here a simple algorithmic solution for a large family of n-tuples of functions. At this aim we provide the notion of "positive decomposition" of one probability distribution into another.

1 The problem

In this paper we address a problem in Algorithmics which is motivated by an application to the so-called "Boltzmann" Exact Sampling [DFLS04], but is most probably not confined to this realm of applications, and in fact can be formulated in terms of a rather fundamental problem in Probability. The algorithmic problem reads as follows:

Problem 1.1. *Let n be a positive integer, and $\boldsymbol{f} = (f_1, \dots, f_n)$ a n-tuple of integer-valued probability distributions taken in a family $\mathcal{F}$ of functions satisfying the hypotheses of the Central Limit Theorem. Let x_i be the random variable associated to the function f_i. Let $m = \lfloor \sum_i \mathbb{E}(x_i) \rfloor$. Assume that we can sample efficiently from the f_i's (i.e., we can extract a random x_i with complexity of order 1). We want an algorithm which samples from the constrained measure*

$$\mu_m(\boldsymbol{x}) \propto \prod_i f_i(x_i) \times \delta_{x_1+\dots+x_n,m} \tag{1}$$

and whose average complexity is linear in n, the constant depending only on $\mathcal{F}$ and on the ratio m/n.

This problem is essentially solved, by Devroye [Dev12], when the distributions f_i are all equal. However his ideas do not apply in the generic case, and it is often the case in real-life applications that these distributions are indeed in the same family, but have distinct underlying parameters (see the examples in Section 4).

In: L. Ferrari, M. Vamvakari (eds.): Proceedings of the GASCom 2018 Workshop, Athens, Greece, 18–20 June 2018, published at http://ceur-ws.org

The main idea of our algorithm is as follows: suppose that you can decompose [1]

$$f_i(x) = \sum_{s \in \mathbb{N}^d} q_i(s)\; g^{*s}(x) \tag{2}$$

where

- $g(x)$ is a 'simple' probability distribution (for example, $g = \mathrm{Bern}_{\frac{1}{2}}$);
- the $q_i(s)$ are non-negative, and, seen as probability distributions for the variable s, can be sampled efficiently;
- call $\sigma^2 = \sum_i \mathbb{V}\mathrm{ar}_{f_i}$; we need $\sqrt{m}/\sigma = \Theta(1)$.

In this case, as we will show in the following, we can sample in linear time the n-tuple $\boldsymbol{s} = (s_1, \ldots, s_n)$, from the distribution $\mu^{\boldsymbol{q}}(\boldsymbol{s}) = \prod_i q_i(s_i)$, apply a rejection scheme with average acceptance rate related to the quantity $\sqrt{m}/\sigma$ above (and thus of order 1), and finally sample $\boldsymbol{x}$ from the measure $\mu_m(\boldsymbol{x}|\boldsymbol{s}) = \prod_i g^{*s_i}(x_i) \times \delta_{x_1+\cdots+x_n,m}$, this latter operation can be performed efficiently because the function g is 'simple', and it enters this final stage just with an n-fold convolution. Thus, the resulting algorithm has linear average complexity.

The understanding of which functions f admit a decomposition as in equation (2) turns out to be an interesting question *per se*. We outline here the main results, for which more details will be given in a forthcoming paper. Then, we provide a detailed description of the algorithm, justify the claim on its complexity, and illustrate it on some relevant examples. In particular, we provide a linear-time algorithm for the uniform sampling of set partitions of a set of cardinality $n + m$ into exactly n parts. This marks an improvement w.r.t. previous results of the authors [BS15] (indeed this application has been the main motivation for this work), and, by previous results [BN07, BDS12], implies a linear exact sampling algorithm for uniform accessible deterministic complete automata (ACDA), and for minimal automata, in any given alphabet of finite size.

2 On positive decomposition

2.1 Positive decomposition of integer-valued probability distributions

In this section we introduce our 'fundamental' problem in Probability. As we have anticipated, for implementing our algorithm we are faced with the following basic construction. Let $\mathcal{F}$ and $\mathcal{G} = \{g_s(x)\}$ be two families of integer-valued probability distributions.

Definition 2.1. We say that $\mathcal{F}$ *decomposes in* $\mathcal{G}$ if all $f \in \mathcal{F}$ are linear combinations of functions in $\mathcal{G}$:

$$f(x) = \sum_s q(s)\; g_s(x) \qquad q(s) \in \mathbb{R}\,. \tag{3}$$

We say that $\mathcal{F}$ *decomposes positively in* $\mathcal{G}$ if the coefficients are non-negative, i.e. $q(s) \in \mathbb{R}^+$. In this case we write $\mathcal{F} \triangleleft \mathcal{G}$.

Remark 2.2. Decomposing positively is a transitive relation: if $\mathcal{F} \triangleleft \mathcal{G}$ and $\mathcal{G} \triangleleft \mathcal{H}$, then $\mathcal{F} \triangleleft \mathcal{H}$.

Remark 2.3. In case of positive decomposition, the coefficients $q(s)$ are themselves a probability distribution. Indeed, they are real-positive, and their sum is 1, because

$$1 = \sum_x f(x) = \sum_x \sum_s q(s)\; g_s(x) = \sum_s q(s) \sum_x g_s(x) = \sum_s q(s)\,. \tag{4}$$

Definition 2.4. We say that $\mathcal{G}$ is the *convolution family of* g if $\mathcal{G} = \{g_s\}_{s\in\mathbb{N}}$, and $g_s = g^{*s}$ (convolution power, $g^{*s}(x) = \sum_{x_1,\ldots,x_{s-1}} g(x_1)\cdots g(x_{s-1})g(x - x_1 - \cdots - x_{s-1})$). In this case we write $\mathcal{G} = \mathrm{Conv}[g]$.

We are now ready to state our problem.

Problem 2.5. *For a given function g, which functions f decompose positively in $\mathcal{G} = \mathrm{Conv}[g]$?*

A first useful property is the following

[1] For $s \geq 0$, here we use g^{*s} as a shortcut for the 'convolution power', i.e. the s-fold convolution of g.

Proposition 2.6. *If f_1 and f_2 decompose positively in the same $\mathcal{G} = \mathrm{Conv}[g]$, also $f_1 * f_2$ does, and the resulting set of coefficients, seen as a probability distribution, is the convolution of the two sets.*

$$(f_1 * f_2)(x) = \sum_{s_1,s_2} q_1(s_1)q_2(s_2)\ (g_{s_1} * g_{s_2})(x) = \sum_{s_1,s_2} q_1(s_1)q_2(s_2)\ g_{s_1+s_2}(x) = \sum_s (q_1 * q_2)(s)\ g_s(x)\,. \tag{5}$$

When $\mathcal{G}$ is a convolution family, the average and variance of g_s are linear in s, and we have

$$\mathbb{E}_f(x) = \sum_x x f(x) = \sum_s q(s) \sum_x x g_s(x) = \sum_s q(s) \sum_x x g^{*s}(x) = \sum_s q(s) s\ \mathbb{E}_g(x) = \mathbb{E}_q(s)\ \mathbb{E}_g(x)\,; \tag{6}$$

$$\begin{aligned}\mathbb{V}\mathrm{ar}_f(x) &= \sum_{x,x'} x(x-x') f(x) f(x') = \sum_{s,s'} q(s)q(s') \sum_{x,x'} x(x-x') g_s(x) g_{s'}(x') \\ &= \sum_{s,s'} q(s)q(s') \big(s\ \mathbb{V}\mathrm{ar}_g(x) + s(s-s')\ \mathbb{E}_g(x)^2\big) = \mathbb{E}_q(s)\ \mathbb{V}\mathrm{ar}_g(x) + \mathbb{V}\mathrm{ar}_q(s) \mathbb{E}_g(x)^2\,.\end{aligned} \tag{7}$$

Solving for $\mathbb{E}_q(s)$ and $\mathbb{V}\mathrm{ar}_q(s)$ gives

$$\mathbb{E}_q(s) = \frac{\mathbb{E}_f(x)}{\mathbb{E}_g(x)}\,; \qquad\qquad \mathbb{V}\mathrm{ar}_q(s) = \frac{\mathbb{V}\mathrm{ar}_f(x)}{\mathbb{E}_g(x)^2} - \frac{\mathbb{E}_f(x)\mathbb{V}\mathrm{ar}_g(x)}{\mathbb{E}_g(x)^3}\,; \tag{8}$$

which implies in particular that

$$\frac{\mathbb{V}\mathrm{ar}_f(x)}{\mathbb{E}_f(x)} \geq \frac{\mathbb{V}\mathrm{ar}_g(x)}{\mathbb{E}_g(x)}\,. \tag{9}$$

Equation (9) implies that, if $\mathcal{F} = \mathrm{Conv}[f]$, $\mathcal{G} = \mathrm{Conv}[g]$, $\mathcal{F} \triangleleft \mathcal{G}$ and $\mathcal{G} \triangleleft \mathcal{F}$, then $f = g$ and $\mathcal{F} = \mathcal{G}$. In other words, the symbol $\triangleleft$ is an order relation among convolution families, which thus form a semilattice, in which the top element is $\mathrm{Conv}[\delta_{x,1}]$.[2]

Write $\widetilde{f}(y) = \sum_x y^x f(x)$ for the generating function associated to f. It is well known that $\widetilde{g^{*s}} = (\widetilde{g})^s$, and thus

$$\widetilde{f}(y) = \sum_x y^x f(x) = \sum_x \sum_s q(s)\ y^x g_s(x) = \sum_s q(s) \widetilde{g^{*s}}(y) = \sum_s q(s) (\widetilde{g}(y))^s = \widetilde{q}(\widetilde{g}(y))\,. \tag{10}$$

Call $y(z)$ the function such that $\widetilde{g}(y(z)) = z$ (a solution to $\widetilde{g}(\bar{y}(z)) = 1 + z$ exists at least as a formal power series, $\bar{y}(z) = 1 + \sum_{n\geq 1} a_i z^i$, and we can set $y(z) = \bar{y}(z-1)$). As a result, we have the formal rule

$$q(s) = [z^s] \widetilde{f}(y(z)) = \oint \frac{\mathrm{d}z}{2\pi i z^{s+1}} \widetilde{f}(y(z))\,. \tag{11}$$

This formula is valid provided that $\widetilde{f}(y(z))$ is analytic in a neighbourhood of $z = 0$. This is a non-trivial condition, as analyticity is guaranteed under weak hypotheses only in a neighbourhood of $z = 1$.

Let now $\mathcal{F} = \mathrm{Conv}[f]$ and $\mathcal{G} = \mathrm{Conv}[g]$. In light of Proposition 2.6, a necessary and sufficient condition for $\mathcal{F} \triangleleft \mathcal{G}$ is that f decomposes positively in $\mathcal{G}$, that, from (11) above, means that $\widetilde{f}(y(z))$ shall be analytic in a neighbourhood of $z = 0$, and have all non-negative Taylor coefficients.

This is our best answer to the question posed by Problem 2.5.

2.2 Three fundamental distributions

There are three fundamental examples of basic functions g, to be used for constructing convolution families $\mathcal{G} = \mathrm{Conv}[g]$, corresponding to the three fundamental statistics of identically-distributed particles in statistical physics. We start with two of them:

[2] Because $\mathrm{Conv}[\delta_{x,1}] = \{\delta_{s,x}\}_{s\in\mathbb{N}}$ is just the canonical basis for distributions on $\mathbb{N}$, and any distribution decomposes positively in it, so we have proven that this poset is at least a semilattice, and we have identified its top element.

Fermionic statistics: For $b \in (0,1]$, the family $\mathcal{G}_b^{\mathrm{Fermi}}$ is defined by

$$g_s(r) = \mathrm{Bino}_{s,b}(r) = \mathrm{Bern}_b^{*s}(r) = b^r(1-b)^{s-r}\binom{s}{r}. \tag{12}$$

The functions g_s are Binomial distributions, that is, s-fold convolutions of Bernoulli random variables of parameter b. The average and variance of the Bernoulli distribution of parameter b are

$$\mathbb{E}_g(r) = b\,; \qquad\qquad \mathbb{V}\mathrm{ar}_g(r) = b(1-b)\,. \tag{13}$$

At $b = 1$ this set of functions reduces to the canonical basis $g_s(r) = \delta_{s,r}$.

Bosonic statistics: For $b \in \mathbb{R}^+$, the family $\mathcal{G}_b^{\mathrm{Bose}}$ is defined as

$$g_s(r) = \mathrm{Geom}_{s,b}(r) = \mathrm{Geom}_b^{*s}(r) = b^r(1+b)^{-s-r}\binom{s+r-1}{r}. \tag{14}$$

The functions g_s are s-fold convolutions of Geometric random variables of parameter b. The average and variance of the geometric distribution of parameter b are

$$\mathbb{E}_g(r) = b\,; \qquad\qquad \mathbb{V}\mathrm{ar}_g(r) = b(1+b)\,. \tag{15}$$

Due to the obvious fact $\binom{n}{k} = (-1)^k\binom{-n+k-1}{k}$, these two families are in fact the analytic continuation one of the other: $\mathrm{Geom}_{s,b}(r) = \mathrm{Bino}_{-s,-b}(r)$. As a result, the positive decomposition $f = \sum_s q(s)g^{*s}$, in the two cases of binomial and geometric distributions, have the same limit $b \to 0$, related to the Poissonian distribution $\mathrm{Poiss}_\rho(x) = e^{-\rho}\rho^x/x!$. In this case the discretisation is lost, as well as the parameter b, and we have a different notion of positive decomposition, namely one of the form f decomposes positively in Conv[Poiss] iff

$$f(x) = \int_0^\infty \mathrm{d}t\; q(t)\,\mathrm{Poiss}_t(x) \tag{16}$$

with $q(t) : \mathbb{R}^+ \to \mathbb{R}^+$.[3]

The possibility of having an integral, instead of a sum, is obviously related to the fact that the Poissonian is infinitely divisible, and would hold for all and only the infinitely divisible distributions. This leads us to our third fundamental statistics:

Classical statistics: The family $\mathcal{G}^{\mathrm{Class}}$ is defined as

$$g_t(r) = \mathrm{Poiss}_t(r) = e^{-t}\frac{t^r}{r!}\,. \tag{17}$$

The average and variance of g_t are

$$\mathbb{E}_{g_t}(r) = t\,; \qquad\qquad \mathbb{V}\mathrm{ar}_{g_t}(r) = t\,. \tag{18}$$

The analogue of (11) states in this case that $q(t)$ is the inverse Laplace transform of $\widetilde{f}(y+1)$, and the condition for f to be positively decomposable in Conv[Poiss] is that this function is both supported on and valued in $\mathbb{R}^+$.

By investigating the expressions $\widetilde{f}(y(\widetilde{g}))$, we can identify the restriction to these families of the poset induced by the order $\mathcal{F} \lhd \mathcal{G}$. In particular, for our three families of distributions we have the tables

	$\widetilde{g}(y)$	$y(\widetilde{g} = z)$
Bino_b	$1 + b(y-1)$	$\dfrac{b-1+z}{b}$
Geom_b	$(1-b(y-1))^{-1}$	$\dfrac{bz-1+z}{bz}$
Poiss	$\exp(y-1)$	$1+\ln(z)$

$\widetilde{f} \setminus \widetilde{g}$	Bino_b	Geom_b	Poiss
Bino_a	$\dfrac{b-a}{b} + z\dfrac{a}{b}$	$\cdot$	$\cdot$
Geom_a	$\dfrac{b}{(a+b)-az}$	$\dfrac{bz}{a-(a-b)z}$	$\dfrac{1}{1-a\ln z}$
Poiss	$\exp\left[\dfrac{1}{b}(z-1)\right]$	$\exp\left[\dfrac{1}{b}\left(1-\dfrac{1}{z}\right)\right]$	$\cdot$

[3] We will often write just Poiss, with no parameter, contrarily to Bino_b and Geom_b. When referring to a family $\mathcal{G}$, we mean a decomposition as in (16), while, when referring to a function, we mean $\mathrm{Poiss} \equiv \mathrm{Poiss}_1$.

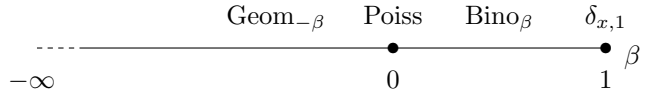
(obvious entries are omitted). The case of Poiss shall be treated separately, by explicit calculation, because of the variation in the criterium (11), which involves integrals instead of sums. In summary, we have,

$$\mathrm{Bino}_a(x) = \sum_{0\le s\le 1} \mathrm{Bino}_{\frac{a}{b}}(s)\ \mathrm{Bino}_{s,b}(x) \tag{19}$$

$$\mathrm{Poiss}_a(x) = \sum_s \mathrm{Poiss}_{\frac{a}{b}}(s)\ \mathrm{Bino}_{s,b}(x) = \int_0^\infty \mathrm{d}s\ \delta(s-a)\,\mathrm{Poiss}_s(x) \tag{20}$$

$$\mathrm{Geom}_a(x) = \sum_s \mathrm{Geom}_{\frac{a}{a+b}}(s)\ \mathrm{Bino}_{s,b}(x) = \int_0^\infty \frac{\mathrm{d}s}{a}\, e^{-\frac{s}{a}} \mathrm{Poiss}_s(x) = \sum_{s\ge 1} \mathrm{Geom}_{\frac{a-b}{a}}(s-1)\ \mathrm{Geom}_{s,b}(x) \tag{21}$$

As a result we have a rather simple picture of the associated portion of our semilattice: the restriction of the partial order to these families is indeed a total order, isomorphic to an interval $\beta \in (-\infty, 1]$, through the identification:



Recall that, as implied by equation (9), the order relation must be compatible with the total order given by $\mathbb{V}\mathrm{ar}_g/\mathbb{E}_g$. Referring back to the explicit expressions (13), (15) and (18), we see that in this parametrisation we just have $\mathbb{V}\mathrm{ar}_g/\mathbb{E}_g = 1-\beta$ for our three families of distributions.

Having identified this ordering is important for our algorithm. Indeed, in our problem we are faced with the question if a given function f is positively decomposable in some family $\mathrm{Conv}[g]$. The transitivity property of positive decomposition implies that, within our three families of distributions, the set of functions g for which f is positively decomposable in the family $\mathrm{Conv}[g]$ is an up-set of the lattice, and thus, under the identification above, an interval $[\beta_{\min}(f), 1]$ for some $\beta_{\min}(f) \le 1$.

Analogously, for a family $\mathcal{F}$ of functions we define

$$\beta_{\min}(\mathcal{F}) := \max_{f\in\mathcal{F}} \big(\beta_{\min}(f)\big)\,. \tag{22}$$

We have thus established that all the functions in the family $\mathcal{F}$ can be positively decomposed by any family $\mathcal{G} = \mathrm{Conv}[g]$, for g being one of our fundamental distributions, with parameter $\beta_{\min}(\mathcal{F}) \le \beta \le 1$, and under the identification above.

3 The algorithm

3.1 The rejection paradigm

There exists a correspondence between measure decompositions and rejection schemes in exact-sampling algorithms, namely, given a decomposition of a measure

$$\mu(x) \propto \sum_y \mu_1(y)\ \mu_2(x|y)\ a(y) \tag{23}$$

with $a(y) \in [0,1]$, Algorithm 1 is an exact-sampling algorithm for μ.

Algorithm 1: Rejection paradigm.

```
begin
    repeat
        y ⟵ μ1;
        α ⟵ Bern_{a(y)};
    until α = 1;
    x ⟵ μ2( · |y);
    return x
```

Call $T_1(y)$ and $T_2(y)$ the average complexities for sampling from μ_1 when the outcome is y (plus sampling one Bernoulli with parameter $a(y)$), and for sampling from $\mu_2(\,\cdot\,|y)$, respectively. The number of failed runs

in the repeat-until loop is geometrically-distributed, and the complexity of each of these runs is an independent random variable. As a result, the average complexities of the failed runs of the loop, of the successful run of the loop, and of the unique run of the final instruction just add up, and the average complexity of the full algorithm is easily calculated via some surprising cancellations

$$\begin{aligned} T &= \frac{1-\sum_y \mu_1(y)a(y)}{\sum_y \mu_1(y)a(y)} \frac{\sum_y T_1(y)\mu_1(y)(1-a(y))}{\sum_y \mu_1(y)(1-a(y))} + \frac{\sum_y T_1(y)\mu_1(y)a(y)}{\sum_y \mu_1(y)a(y)} + \frac{\sum_y T_2(y)\mu_1(y)a(y)}{\sum_y \mu_1(y)a(y)} \\ &= \frac{\mathbb{E}(T_1 + a\, T_2)}{\mathbb{E}(a)} \end{aligned} \tag{24}$$

where averages $\mathbb{E}(\cdot)$ are taken w.r.t. μ_1. If the functions $T_1(y)$ and $T_2(y)$ are bounded by the constants T_1 and T_2, we get the bound

$$T \leq \frac{T_1}{\mathbb{E}(a)} + T_2\,. \tag{25}$$

Of course, the inequality above is an equality if the functions $T_1(y)$ and $T_2(y)$ are indeed constants.

3.2 Why the algorithm has linear complexity

First, let us address the question of why we shall hope that linear complexity can be achieved, without concentrating on the technical detail of how a single variable can be sampled with $\Theta(1)$ complexity.

Following the framework described in the introduction, n is a 'large' positive integer, and $\boldsymbol{f} = (f_1, \ldots, f_n)$ is a n-tuple of integer-valued probability distributions, taken in a family $\mathcal{F}$ of functions satisfying the hypotheses of the Central Limit Theorem, and such that $\beta_{\min}(\mathcal{F}) < 1$.

Let x_i be the random variable associated to the function f_i, and suppose that $m = \sum_i \mathbb{E}(x_i)$ is an integer. Under our hypotheses, it is trivial to sample in linear time from the unconstrained measure

$$\mu^{(f_1,\ldots,f_n)}(x_1, \ldots, x_n) = \mu^{(\boldsymbol{f})}(\boldsymbol{x}) = \prod_i f_i(x_i)\,; \tag{26}$$

however we want an algorithm which samples from the constrained measure

$$\mu_m^{(\boldsymbol{f})}(\boldsymbol{x}) \propto \prod_i f_i(x_i) \times \delta_{x_1+\cdots+x_n,m} \tag{27}$$

with average complexity which is linear in n, the constant depending only on $\mathcal{F}$ and on the ratio m/n.

Call $\mathbb{P}_{\boldsymbol{f}}(m')$ the probability, in the unconstrained measure, that $\sum_i x_i = m'$. By our CLT hypothesis, this probability is maximal around $m' = m$, has variance which is linear in n, and thus its value is of order $1/\sqrt{n}$ at $m' = m$. By definition,

$$\mathbb{P}_{\boldsymbol{f}}(m') = \frac{\sum_{\boldsymbol{x}} \prod_i f_i(x_i) \times \delta_{x_1+\cdots+x_n,m'}}{\sum_{\boldsymbol{x}} \prod_i f_i(x_i)}\,. \tag{28}$$

Now, choose $\beta \in [\beta_{\min}(\mathcal{F}), 1)$, and decompose the f_i's in the family $\mathcal{G} = \mathrm{Conv}[g]$, for g the fundamental distribution corresponding to b. Call $q_i(s)$ the corresponding expansion coefficients. We use a sum notation, with the disclaimer that for Poiss sums are replaced by integrals.

Thus we can write (call $\boldsymbol{g_s} = (g_{s_1}, \ldots, g_{s_n})$ and $\boldsymbol{q} = (q_1, \ldots, q_n)$)

$$\mu^{(\boldsymbol{f})}(\boldsymbol{x}) = \prod_i f_i(x_i) = \sum_{\boldsymbol{s}} \prod_i q_i(s_i) g_{s_i}(x_i) = \sum_{\boldsymbol{s}} \mu^{(\boldsymbol{q})}(\boldsymbol{s})\; \mu^{(\boldsymbol{g_s})}(\boldsymbol{x})\,; \tag{29}$$

and

$$\mu_{m'}^{(\boldsymbol{f})}(\boldsymbol{x})\mathbb{P}_{\boldsymbol{f}}(m') = \prod_i f_i(x_i)\; \delta_{\sum_i x_i,m'} = \sum_{\boldsymbol{s}} \prod_i q_i(s_i) g_{s_i}(x_i)\; \delta_{\sum_i x_i,m'} = \sum_{\boldsymbol{s}} \mu^{(\boldsymbol{q})}(\boldsymbol{s})\; \mu_{m'}^{(\boldsymbol{g_s})}(\boldsymbol{x}) \mathbb{P}_{\boldsymbol{g_s}}(m')\,. \tag{30}$$

Now, call $N = N(\boldsymbol{s}) = \sum_i s_i$. As $g_s = g^{*s}$, in fact $\mathbb{P}_{\boldsymbol{g_s}}(m')$ depends on $\boldsymbol{s}$ only through $N(\boldsymbol{s})$, and is

$$\mathbb{P}_{g_N}(m) = g^{*N}(m) = \begin{cases} \mathrm{Bino}_{N,b}(m) & \text{if } g = \mathrm{Bino}_b \\ \mathrm{Poiss}_N(m) & \text{if } g = \mathrm{Poiss} \\ \mathrm{Geom}_{N,b}(m) & \text{if } g = \mathrm{Geom}_b \end{cases} \tag{31}$$

In all three cases, we can easily evaluate $\bar{N} := \mathrm{argmax}_N(\mathbb{P}_{g_N}(m))$ (the distributions above are easily seen to be log-concave in N, by direct calculation on (12), (14) and (17)), thus it suffices to analyse the ratios $\mathbb{P}_{g_N}(m)/\mathbb{P}_{g_{N-1}}(m)$). We get

$$\bar{N} = \begin{cases} \lfloor \frac{m}{b} \rfloor & \text{if } g = \mathrm{Bino}_b \\ \lfloor (1 - e^{-\frac{1}{m}})^{-1} \rfloor & \text{if } g = \mathrm{Poiss} \\ \lceil \frac{m}{b} \rceil & \text{if } g = \mathrm{Geom}_b \end{cases} \tag{32}$$

Note that, in all three cases, when m is large, this corresponds to $\bar{N} \simeq m/\mathbb{E}_g$. This is in agreement with (8), as indeed we have

$$\bar{N}_\alpha = \sum_i (\mathbb{E}_{q_i})_\alpha = \sum_i \frac{(\mathbb{E}_{f_i})_\alpha}{\mathbb{E}_{g_\alpha}} = \frac{m_\alpha}{\mathbb{E}_{g_\alpha}}\,, \tag{33}$$

and the fact that, as by CLT we converge to a normal distribution, average and mode are equal at leading order. Combining the previous results, we can write

$$\mu_m^{(f_1,\dots,f_n)}(\boldsymbol{x}) \propto \sum_{\boldsymbol{s}} \mu^{(q_1,\dots,q_n)}(\boldsymbol{s})\; \mu_m^{(g_{s_1},\dots,g_{s_n})}(\boldsymbol{x}) \frac{\mathbb{P}_{g_{N(\boldsymbol{s})}}(m)}{\mathbb{P}_{g_{\bar{N}}}(m)}\,. \tag{34}$$

By definition of $\bar{N}$, the ratio $a(N) := \frac{\mathbb{P}_{g_N}(m)}{\mathbb{P}_{g_{\bar{N}}}(m)}$ is in $[0,1]$, and thus can be used as the acceptance rate in a rejection algorithm as in Section 3.1. Indeed, at this point we can apply Algorithm 1 with the identifications

x	$\boldsymbol{x}$	$\mu(x)$	$\mu_m^{(f_1,\dots,f_n)}(\boldsymbol{x})$
y	$\boldsymbol{s}$	$\mu_1(y)$	$\mu^{(q_1,\dots,q_n)}(\boldsymbol{s})$
$a(y)$	$\mathbb{P}_{g_{N(\boldsymbol{s})}}(m)/\mathbb{P}_{g_{\bar{N}}}(m)$	$\mu_2(x\|y)$	$\mu_m^{(g_{s_1},\dots,g_{s_n})}(\boldsymbol{x})$

Let us neglect the dependence of the complexities $T_1(\boldsymbol{s})$ and $T_2(\boldsymbol{s})$ from $\boldsymbol{s}$, and use directly equation (25). Let us assume that we have determined that T_1 and T_2 are linear in n. We shall thus calculate the acceptance rate $\mathbb{E}(a)$ of that formula, and verifies that it scales as $\theta(1)$ for large n. In this setting, this reads

$$\mathbb{E}(a) = \sum_{\boldsymbol{s}} \mu^{(q_1,\dots,q_n)}(\boldsymbol{s}) \frac{\mathbb{P}_{g_{N(\boldsymbol{s})}}(m)}{\mathbb{P}_{g_{\bar{N}}}(m)}\,. \tag{35}$$

In other words, call $\hat{\mu}(N)$ the distribution on $N = N(\boldsymbol{s})$ induced by $\mu^{(q_1,\dots,q_n)}(\boldsymbol{s})$. Then we have

$$\mathbb{E}(a) = \sum_N \hat{\mu}(N) \frac{\mathbb{P}_{g_N}(m)}{\mathbb{P}_{g_{\bar{N}}}(m)} = \frac{\mathbb{E}_{\hat{\mu}}(\mathbb{P}_{g_N}(m))}{\mathbb{P}_{g_{\bar{N}}}(m)}\,. \tag{36}$$

This already makes clear why our strategy is successful. When n tends to infinity, because of our CLT hypothesis, $\hat{\mu}$ converges to a *normalised* Gaussian, with average $\bar{N}$ and some variance σ_1^2 of order n. On the other side, $\mathbb{P}_{g_N}(m)/\mathbb{P}_{g_{\bar{N}}}(m)$ converges to a *non-normalised* Gaussian, with average $\bar{N}$ and variance σ_2^2 of order n. This Gaussian has been rescaled as to be valued 1 when $N = \bar{N}$. As a result, we just have

$$\mathbb{E}(a) \simeq \int \mathrm{d}x\; \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x^2}{2\sigma_1^2} - \frac{x^2}{2\sigma_2^2}} = \frac{\sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}\,. \tag{37}$$

The resulting expression is homogeneous, so that it evaluates to a constant of order 1 for a whatever scaling in n of σ_1^2 and σ_2^2 (provided that the two variances have the same scaling). In this case the average complexity is

$$T \simeq T_1 \sqrt{1 + (\sigma_1/\sigma_2)^2} + T_2\,, \tag{38}$$

and thus, by our assumptions, is linear.

Note that we would have obtained an expression still of order 1, even if we did take a "wrong" tuning of m, in the sense that $\sum_i \mathbb{E}_{f_i} = m + \delta m$, with $\delta m \gg 1$, provided that δm is small w.r.t. σ_1 and σ_2 (which, under the hypotheses of the Central Limit Theorem, are of order $\sqrt{n}$). That is, in the language of Boltzmann samplers, we have a tolerance on the tuning of the Boltzmann parameter up to a relative error on the average outcome size of order $1/\sqrt{n}$.

Now let us relate the parameters σ_1 and σ_2 to the parameters of the problem. One case is rather easy: as the variances sum up under convolution,

$$\sigma_1^2 = \sum_i \mathbb{V}\mathrm{ar}_{q_i} \,. \tag{39}$$

For what concerns σ_2, we shall calculate the variance in the variable N of the function $g^{*N}(m)$. From the calculation on the explicit expressions (12), (14) and (17)) one easily derives that, in all three cases, $\sigma_2^2 = m\frac{\mathbb{V}\mathrm{ar}_g}{\mathbb{E}_g^3}$. Now, using (8) we get

$$\sigma_1^2 + \sigma_2^2 = \frac{1}{\mathbb{E}_g^2} \sum_i \mathbb{V}\mathrm{ar}_{f_i} \tag{40}$$

so that

$$\mathbb{E}(a) \simeq \frac{\sigma_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} = \sqrt{\frac{m\, \mathbb{V}\mathrm{ar}_g}{\mathbb{E}_g \sum_i \mathbb{V}\mathrm{ar}_{f_i}}} \,. \tag{41}$$

If we take as function g the one with parameter $\beta_{\min}(\mathcal{F})$ (which we have determined to be optimal), the asymptotic maximal acceptance rate for the set $\mathcal{F} = \{f_1, \ldots, f_n\}$ within our algorithm is given by

$$a(\mathcal{F}) := \sqrt{\frac{(1 - \beta_{\min}(\mathcal{F}))\; m}{\sum_i \mathbb{V}\mathrm{ar}_{f_i}}} \,. \tag{42}$$

Under our assumption that the f_i's satisfy the hypotheses of the Central Limit Theorem, we have that all $\mathbb{E}_{f_i}$'s and $\mathbb{V}\mathrm{ar}_{f_i}$'s are of order 1, so that $m = \sum_i \mathbb{E}_{f_i}$ and $\sum_i \mathbb{V}\mathrm{ar}_{f_i}$ have the same scaling behaviour. Once again we have obtained that, for n going to infinity, $a(\mathcal{F})$ converges to a finite quantity.

3.3 How the algorithm works in practice

At the level of generality of this paper, we cannot give an algorithm that works just out of the box. For a given problem, given by the n-tuple $(f_1, \ldots, f_n)$, a preliminary analytical study of the functions f_i is mandatory. One has to

- Determine that $\mathcal{F} \lhd \mathcal{G}$ for $\mathcal{G} = \mathrm{Conv}[g_\beta]$, for some parameter β smaller than 1.
- Devise an algorithm to sample from the corresponding distributions $q_i(s)$, in average time τ_i, so that $\sum_i \tau_i$ scales linearly with n.

Only at this point, one is ready to apply our general algorithm, which reads as in Algorithm 2.

Algorithm 2: Our new algorithm.

begin
  $\bar{N} = \texttt{round}(m/\mathbb{E}_g)$;
  repeat
    $N = 0$;
    for $i \leftarrow 1$ **to** n **do**
      $s_i \longleftarrow q_i$;
      N += s_i;
    $a = g^{*N}(m)/g^{*\bar{N}}(m)$;
    $\alpha \longleftarrow \mathrm{Bern}_a$;
  until $\alpha = 1$;
  Sample $\boldsymbol{x}$ with $\mu_m^{(\boldsymbol{g_s})}(\boldsymbol{x})$;
  return $\boldsymbol{x}$

Contrarily to the first part of the algorithm, where a preliminary analysis concerning the functions q_i is required, the final step of the algorithm, of sampling $\boldsymbol{x}$ with $\mu_m^{(\boldsymbol{g_s})}(\boldsymbol{x})$, is "universal", as it depends only on the family of basic distribution g_β, among binomial, geometric or Poissonian.

The Poissonian case is trivial in spirit (it suffices to sample m independent values $\xi_i \in [0, N]$, and then increment x_i if $0 < \xi_i - \sum_{j=1}^{i-1} s_j < s_i$), however, as now the intermediate variables s_i are real-valued, must be performed with float numbers and has complexity $\mathcal{O}(m \ln n)$. In the special case in which the s_i are integer-valued, that is the q_i's are supported on $b\mathbb{N}$ for some b, we have a genuinely linear alternate algorithm, which we will describe elsewhere.

The binomial and geometric case are in a sense 'dual', and are solved by the same algorithm, provided by Bacher, Bodini, Hollender and Lumbroso in [BBHL15, sec. 1.1] (see also [BBHT17]), which is genuinely linear. This algorithm, that we shall call "BBHL-shuffling(n, k)", may be seen as a black box which samples strings $\boldsymbol{\nu} = (\nu_1, \dots, \nu_n) \in \{0, 1\}^n$ with $\sum_i \nu_i = k$, uniformly, and that, for $n/k - 1 = \Theta(1)$, requires a time linear in n.[4] Then, in the two cases we have:

Case $g^{*N}(m) = \mathrm{Bino}_{N,b}(m)$: Use BBHL-shuffling$(N, m)$, then set

$$x_i = \nu_{j(i-1)+1} + \cdots + \nu_{j(i)}\,; \qquad\qquad j(i) = s_1 + s_2 + \cdots + s_i\,. \tag{43}$$

Case $g^{*N}(m) = \mathrm{Geom}_{N,b}(m)$: Use BBHL-shuffling$(m + N - 1, m)$. Call j_1, …, j_{N-1} the positions of the ν_i's which are valued 0. Let $j_0 = 0$. Then

$$x_i = j_i - j_{i-1}\,. \tag{44}$$

3.4 Bacher, Bodini, Hollender and Lumbroso algorithm

Here we describe an algorithm, given by Bacher, Bodini, Hollender and Lumbroso in [BBHL15, sec. 1.1], for the uniform sampling of strings $\boldsymbol{\nu} = (\nu_1, \dots, \nu_n) \in \{0, 1\}^n$ with $\sum_i \nu_i = k$. In Section 3.3 this algorithm is called "BBHL-shuffling(n, k)". Recall that it is 'optimal' for the randomness resource, in the sense that the random-bit complexity at leading order saturates the Shannon bound.

Call $\beta = k/n$. The idea is that you sample the ν_i's one by one, as if $\boldsymbol{\nu}$ were to be sampled with the measure $\mu(\boldsymbol{\nu}) = \mathrm{Bern}_\beta^n$, (this costs $\Theta(1)$ per variable), and, as soon as you have an excess of ν_i's equal to 0 or to 1, complete with a sequence of Fisher–Yates random swaps. These swaps cost $\sim \ln n$ each, but are performed on average only $\sim \sqrt{n}$ times, so the swap part of the algorithm has subleading complexity, and the overall complexity is genuinely linear, with no logarithmic factors. For completeness, we give in Algorithm 3 a summary of the main features. In this algorithm RndInt_j returns an uniform random integer in $\{1, \dots, j\}$.

Algorithm 3: BBHL-shuffling.

```
begin
    a = k, b = n − k, i = 0;
    repeat
        i ++;
        ν_i ⟵ Bern_β;
        if ν_i = 1 then a −−;
        else b −−;
    until a < 0 or b < 0;
    if a < 0 then ν̄ = 0;
    else ν̄ = 1;
    for j ← i to n do
        ν_j = ν̄;
        h ⟵ RndInt_j;
        swap ν_j and ν_h;
    return ν
```

Note that, if β is almost a 2-adic number, namely $\beta = k/n = a2^{-d} + \varepsilon$ for some integers a and d, and $\varepsilon = \mathcal{O}(1/\sqrt{n})$, it is convenient to just use the source of randomness $\mathrm{Bern}_{\beta'}$ for the value $\beta' = a2^{-d}$, which speeds up the main part of the algorithm, at the price of slowing down the subleading part.

[4] For completeness, this algorithm is quickly described in the following subsection.

4 Examples

The drawback of the algorithm outlined in Section 3.3 is that the user needs to perform a preliminary analytic study of its given probability distributions f_i. There are obvious cases in which this work is not required: when all of the f_i's are distributions in our three basic families, i.e. binomial, geometric or Poissonian. In this case $\beta_{\min}(\mathcal{F})$ is evaluated directly, the functions $q_i(s)$ are given explicitly in Section 2.2, and are just, yet again, binomial, geometric or Poissonian (we provide the generating function for the basic cases, $\mathrm{Bern}_b = \mathrm{Bino}_{1,b}$ and $\mathrm{Geom}_b = \mathrm{Geom}_{1,b}$, however the generic case is deduced immediately via Proposition 2.6). It is well known that one can sample from these distributions in constant complexity, so that we know that we have a linear-time algorithm as soon as we verify that $\mathbb{V}\mathrm{ar}_{f_i}/(\mathbb{E}_{f_i}(1-\beta)) = \Theta(1)$, which is straightforward in most cases.

The paradigm of sum-constrained geometric random variables with distinct parameters applies, for example, to set partitions, enumerated by the Stirling numbers of the second kind. The same paradigm with Bernoulli variables applies, for example, to permutations with a given number of cycles, enumerated by the Stirling numbers of the first kind. In this section we describe in detail these two examples.

4.1 Set partitions, Stirling numbers of the second kind and Automata

Each of the P_n partitions B of a set of cardinality n has a number $k(B)$ of parts in the range $\{1, 2, \ldots, n\}$. The Stirling numbers of the second kind $S_2(n,k)$ are the corresponding enumeration, and a classical algorithmic problem is to sample B uniformly from the corresponding ensemble $\mathcal{S}_2(n,k)$ [FS09].

This problem has acquired further interest since it has been shown, in [BN07], that accessible deterministic complete automata (ACDA) are in bijection with a certain subset of set partitions in $\mathcal{S}_2(kn+1, n)$ (where $k = \Theta(1)$ is the size of the alphabet, and n is the number of states), characterised by a property easy to test (the 'backbone' should stay above the diagonal), and that this subset is asymptotically a finite fraction of the ensemble. As a result, up to a rejection scheme, a uniform sampling algorithm for set partitions provides a uniform sampling algorithm for accessible deterministic complete automata.

Furthermore, in [BDS12] it is shown that minimal automata, which are at sight a subset of ACDA's, are asymptotically dense in this ensemble (i.e., the ratio of the cardinalities of the sets converges to 1), so, yet again, up to a rejection scheme, a uniform sampling algorithm for set partitions provides a uniform sampling algorithm for minimal automata.

Sampling from $\mathcal{S}_2(n+m, n)$, when m/n is $\theta(1)$, can be done by the Boltzmann Method [DFLS04] with complexity $\mathcal{O}(n^{\frac{3}{2}})$, and we show here that our new method gives a linear complexity. The recursive description of $\mathcal{S}_2(n,k)$ provides a natural bijection with certain tableaux, which are described in [FS09, pag. 63], and, in more detail, in [BDS12], for which the so-called 'backbone' relates to a list $\boldsymbol{x} = (x_1, \ldots, x_n)$ as in our setting, with $\mu^{(\boldsymbol{f})}(\boldsymbol{x}) \propto \prod_i \mathrm{Geom}_{b_i}(x_i) \times \delta_{x_1+\cdots+x_n, n-k}$. As a result, a uniform set partition in $\mathcal{S}_2(n+m, n)$ can be sampled in two steps: first we sample the backbone $\boldsymbol{x} = (x_1, \ldots, x_n)$ with the measure above, then, for every $1 \le i \le n$, we sample x_i independent random integers in $\{1, \ldots, i\}$, which describe the so-called 'wiring part' of the tableau. The latter step has a complexity $\mathcal{O}(m \ln n)$, which is intrinsic to this problem and is in fact implied by the Shannon bound (i.e., the asymptotics of $\ln S_2(\alpha n, n)$, as evinced by well-known saddle-point calculations, see e.g. [BN07]). As this latter step is invoked only once in the algorithm, it turns out that this part of the algorithm is optimal for the randomness resource. As, in our algorithm, the first part of sampling the backbone has complexity linear in n and off from optimal only by a multiplicative factor, we deduce that the whole algorithm is asymptotically optimal (although, admittedly, the convergence to optimality is slow, as it goes as $1/\ln n$).

The parameters b_i of the geometric variables are tuned as to have $b_i/(b_i+1) \propto i$ and $\sum_i \mathbb{E}(x_i) = m + o(\sqrt{m})$, which is solved by

$$b_i = \frac{\omega i}{n - \omega i}\,; \qquad\qquad 1 + \frac{m}{n} = -\frac{\ln(1-\omega)}{\omega}\,. \tag{45}$$

Then, using the fact, seen in Section 2.2, that $\mathrm{Geom}_a(x) = \sum_s \mathrm{Geom}_{\frac{a}{a+b}}(s)\ \mathrm{Bino}_{s,b}(x)$, and choosing for simplicity $\beta = \frac{1}{2}$, we have

$$\mathrm{Geom}_{b_i}(x) = \sum_s \mathrm{Geom}_{\frac{1}{1+2b_i}}(s)\ \mathrm{Bino}_{s,\frac{1}{2}}(x) = \sum_s \mathrm{Geom}_{\frac{n-\omega i}{n+\omega i}}(s)\ \mathrm{Bino}_{s,\frac{1}{2}}(x) \tag{46}$$

Also, $\bar{N} = 2m$. At this point we are ready to use our Algorithm 2, with this value of $\bar{N}$, functions q_i given by $\mathrm{Geom}_{\frac{n-\omega i}{n+\omega i}}$, and acceptance factor

$$a(N) = \frac{g^{*N}(m)}{g^{*\bar{N}}(m)} = 2^{2m-N} \frac{N!\, m!}{(N-m)!\, (2m)!} \tag{47}$$

and the part "Sample $\boldsymbol{x}$ with $\mu_m^{(\boldsymbol{g_s})}(\boldsymbol{x})$" is performed via the BBHL-shuffling(N, m) with parameter $\beta = \frac{1}{2}$. The acceptance rate $a = \mathbb{E}(a(N))$ is given by (42), just with $\beta_{\min} = 0$ traded for $\beta = \frac{1}{2}$, which we have chosen as a good compromise between efficiency and complexity of coding, and gives

$$a = \sqrt{\frac{m}{2\sum_i b_i(1+b_i)}}\,. \tag{48}$$

However,

$$\sum_i b_i(1+b_i) \simeq n \int_0^1 \mathrm{d}x\, \frac{\omega x}{(1-\omega x)^2} = n\left(\frac{\ln(1-\omega)}{\omega} + \frac{1}{1-\omega}\right). \tag{49}$$

Using (45), and calling $\omega = 1 - e^{-\theta}$ (it turns out that $\omega \in (0,1)$), we have

$$a = \sqrt{\frac{e^{-\theta} - 1 + \theta}{2(e^\theta - 1 - \theta)}}\,, \tag{50}$$

which behaves as $\frac{1}{\sqrt{2}}\left(1 - \frac{1}{3}\theta + \cdots\right)$ for θ small, and as $\sqrt{\frac{\theta}{2}} e^{-\frac{\theta}{2}}$ for θ large. On the other side, the ratio m/n is parametrised by θ as

$$\frac{m}{n} = \frac{e^{-\theta} - 1 + \theta}{1 - e^{-\theta}} \tag{51}$$

which is a monotone function, valued $\sim \theta/2$ near $\theta = 0$ and $\sim \theta$ for $\theta \to \infty$, so that $a \sim \frac{1}{\sqrt{2}}\left(1 - \frac{2}{3}\frac{m}{n} + \cdots\right)$ for m/n small, and as $\sqrt{\frac{m}{2n}} e^{-\frac{m}{2n}}$ for m/n large.

4.2 Cycles in permutations and Stirling numbers of the first kind

Each of the $n!$ permutations σ of $\mathfrak{S}_n$ has a number $k(\sigma)$ of cycles in the range $\{1, 2, \ldots, n\}$. The Stirling numbers of the first kind $S_1(n,k)$ are the corresponding enumeration, and a classical algorithmic problem is to sample uniformly from the corresponding ensemble $\mathcal{S}_1(n,k)$ [FS09, pag. 121]. When $n/k - 1$ is $\theta(1)$, this problem is solved by Boltzmann sampling [DFLS04] with complexity $\mathcal{O}(n^{\frac{3}{2}})$, and we show here that our new method gives a linear complexity.

The recursive description of $\mathcal{S}_1(n,k)$, namely $\mathcal{S}_1(n,k) \simeq \mathcal{S}_1(n-1,k-1) \cup \mathcal{S}_1(n-1,k) \times \{1, \ldots, n-1\}$, provides a natural bijection with certain 0-1 matrices M_{ij}, for which the diagonal relates to a list $\boldsymbol{x} = (x_1, \ldots, x_n)$ as in our setting, with $x_i = 1 - M_{ii}$ and $\mu^{(\boldsymbol{f})}(\boldsymbol{x}) \propto \prod_i \mathrm{Bern}_{b_i}(x_i) \times \delta_{x_1 + \cdots + x_n, n-k}$. The parameters b_i are tuned as to have $b_i/(1-b_i) \propto i - 1$ and $m := n - k = \sum_i \mathbb{E}(x_i) + o(\sqrt{n})$, with solution

$$b_i = \frac{\omega(i-1)}{n + \omega(i-1)}\,; \qquad\qquad 1 - \frac{m}{n} = \frac{\ln(1+\omega)}{\omega}\,. \tag{52}$$

Then, using the fact, seen in Section 2.2, that $\mathrm{Bino}_a(x) = \sum_s \mathrm{Bino}_{\frac{a}{b}}(s)\, \mathrm{Bino}_{s,b}(x)$, this expansion being positive for $a \le b$, and choosing $\beta = \frac{\omega}{1+\omega} \ge \max_i b_i$, we have

$$\mathrm{Bino}_{b_i}(x) = \sum_s \mathrm{Bino}_{\frac{(i-1)(1+\omega)}{n+(i-1)\omega}}(s)\, \mathrm{Bino}_{s,\beta}(x) \tag{53}$$

Also, $\bar{N} = \lfloor m/\beta \rfloor$. At this point we are ready to use our Algorithm 2, with this value of $\bar{N}$, functions q_i given by $\mathrm{Bino}_{\frac{(i-1)(1+\omega)}{n+(i-1)\omega}}$, and acceptance factor

$$a(N) = \frac{g^{*N}(m)}{g^{*\bar{N}}(m)} = (1-\beta)^{N-\bar{N}} \frac{N!\, (\bar{N}-m)!}{(N-m)!\, \bar{N}!} \tag{54}$$

and the part "Sample $\boldsymbol{x}$ with $\mu_m^{(\boldsymbol{g_s})}(\boldsymbol{x})$" is performed via the BBHL-shuffling(N, m) with parameter β. From this point on, the analysis can be performed in a similar way as for set partitions. We finally get for the acceptance rate, given implicitly as a function of m/n,

$$\frac{m}{n} = 1 - \frac{\theta}{e^\theta - 1}\,; \qquad\qquad a = \sqrt{\frac{e^\theta - 1 - \theta}{1 + e^\theta(\theta - 1)}}\,. \tag{55}$$

Acknowledgements

A.S. is supported by ANRMetaConc (ANR-15-CE40-0014, France).

References

[BBHL15] A. Bacher, O. Bodini, A. Hollender and J. Lumbroso. MergeShuffle: A Very Fast, Parallel Random Permutation Algorithm. At *https://arxiv.org/abs/1508.03167*, 2015.

[BBHT17] A. Bacher, O. Bodini, H.-K. Hwang and T.-H. Tsai. Generating random permutations by coin tossing: classical algorithms, new analysis, and modern implementation. *ACM Transactions on Algorithms*, 13:#24, 2017.

[BDS12] F. Bassino, J. David and A. Sportiello. Asymptotic enumeration of minimal automata. In: *STACS 2012, Paris, France, LIPIcs – Leibniz International Proceedings in Informatics*, 14:88–99, 2012.

[BN07] F. Bassino and C. Nicaud. Enumeration and Random Generation of Accessible Automata. *Theoretical Computer Science*, 381:86–104, 2007.

[BS15] F. Bassino and A. Sportiello. Linear-time generation of inhomogenous random directed walks. In: *ANALCO'15, San Diego, California*, pp. 51–65, 2015.

[Dev12] L. Devroye. Simulating Size-constrained Galton-Watson Trees. *SIAM Journal on Computing*, 41:1–11, 2012.

[DFLS04] P. Duchon, P. Flajolet, G. Louchard and G. Schaeffer. Boltzmann Samplers for the Random Generation of Combinatorial Structures. *Combinatorics Probability and Computing*, 13:577-625, 2004.

[FS09] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.