



AN EFFICIENT METHOD FOR GENERATING A DISCRETE UNIFORM DISTRIBUTION USING A BIASED RANDOM SOURCE

XIAOYU LEI,* *The University of Chicago*

Abstract

We present an efficient algorithm to generate a discrete uniform distribution on a set of p elements using a biased random source for p prime. The algorithm generalizes Von Neumann's method and improves the computational efficiency of Dijkstra's method. In addition, the algorithm is extended to generate a discrete uniform distribution on any finite set based on the prime factorization of integers. The average running time of the proposed algorithm is overall sublinear: $O(n/\log n)$.

Keywords: Random numbers; probability theory

2020 Mathematics Subject Classification: Primary 68W20

Secondary 68Q87

1. Background

Sampling a target distribution from a random physical source has many applications. However, the random physical sources are often biased with unknown distribution, while we need a specific target distribution in applications. Therefore, an efficient algorithm to generate a target distribution from a random source is of great value. A simple method to generate a fair binary distribution from an unfair binary source with an unknown bias was first proposed in [7], and this method has served as a precursor of a series of algorithms to generate a target distribution from an unknown random source.

Von Neumann's method was improved in [3, 6] to generate a fair binary distribution from a biased random source. From the point of view of probability theory, [2] formally defined the kind of random procedure that can generate a target distribution. Elias also designed an infinite sequence of sampling schemes, with computational efficiency decreasing to the theoretical lower bound, though without providing an executable algorithm for the method. Elias' method needs to generate Elias' function first; such a preprocessing step needs an exponential space cost and at least a polynomial time cost [5], and thus Elias' method is computationally costly and inefficient.

Another method, for generating a uniform distribution on a set of p elements for p prime, was provided in [1], though Dijkstra's method is computationally inefficient. Indeed, in realizing this method we need a preprocessing step to generate and store a function which maps outcomes from the random source to some target values. However, such a preprocessing step needs an exponential time and space cost.

In this article we propose a new algorithm based on the idea of Dijkstra's method. The proposed algorithm does not need a preprocessing step, and is thus computationally efficient.

Received 2 May 2022; revision received 29 September 2022.

* Postal address: 5747 South Ellis Avenue, Chicago, Illinois, USA. Email: leixy@uchicago.edu

© The Author(s), 2023. Published by Cambridge University Press on behalf of Applied Probability Trust.

Algorithm 1 \mathcal{A}_1 : Von Neumann’s algorithm generating a fair binary random variable.

Input: A sequence of flips from a biased coin X

Output: Integer 0 or 1

- 1: Flip the coin twice.
- 2: If the result is either HH or TT, then discard the two coin flips and return to step 1.
- 3: If the result is HT, **return** $\mathcal{A}_1 = 0$. If the result is TH, **return** $\mathcal{A}_1 = 1$

The article is organized as follows: In Section 2, we briefly recast Von Neumann’s method as a starting point, as well as a special case of our algorithm. In Section 3, we heuristically construct and explain our algorithm. In Sections 4 and 5 we formally propose our algorithms and verify them theoretically. In Section 5, we prove that our algorithm has overall sublinear average running time. Another novel proof of Theorem 1 is given in Appendix A.

2. Introduction to Von Neumann’s method

Let $X \in \{H, T\}$ denote the outcome of a biased coin flip with probability $a = \mathbb{P}(X = H) \in (0, 1)$ of getting a head and probability $b = \mathbb{P}(X = T) = 1 - a$ of getting a tail. Let $\{X_i : i \geq 0\}$ be independent and identically distributed (i.i.d.) copies of X . Von Neumann proposed an algorithm \mathcal{A}_1 to generate a fair binary random variable with distribution $\mathbb{P}(\mathcal{A}_1 = 0) = \mathbb{P}(\mathcal{A}_1 = 1) = \frac{1}{2}$ as shown in Algorithm 1 [7].

Let $\{Y_i = (X_{2i}, X_{2i+1}) : i \geq 0\}$ be i.i.d. outcomes of pairs of flips, and τ be the first time that $Y_i \in \{HT, TH\}$; then

$$\mathbb{P}(\mathcal{A}_1 = 0) = \mathbb{P}(Y_\tau = HT) = \frac{\mathbb{P}(Y_0 = HT)}{\mathbb{P}(Y_0 \in \{HT, TH\})} = \frac{\mathbb{P}((X_0, X_1) = HT)}{\mathbb{P}((X_0, X_1) \in \{HT, TH\})} = \frac{1}{2}.$$

This derivation shows that \mathcal{A}_1 generates a fair binary distribution. Below, we propose an efficient algorithm to generate a uniform distribution on p elements for p prime. At each cycle we flip a coin p times; the algorithm returns a number in $\{0, \dots, p - 1\}$ except when the p flips are all heads or all tails, analogous to Von Neumann’s method.

3. Heuristic explanation for the main idea

Let the random vector $X^n = (X_0, \dots, X_{n-1}) \in \{H, T\}^n$ be the outcome of n flips. Let $N_{\text{head}}(X^n)$ denote the head count in X^n , and $S_{\text{head}}(X^n)$ denote the rank sum of heads in X^n , with rank beginning from 0:

$$N_{\text{head}}(X^n) = \sum_{i=0}^{n-1} \mathbf{1}_{\{X_i=H\}}, \quad S_{\text{head}}(X^n) = \sum_{i=0}^{n-1} i \cdot \mathbf{1}_{\{X_i=H\}}. \tag{1}$$

For example, when $X^5 = (H, H, T, H, T)$, we have $N_{\text{head}}(X^5) = 3$ and $S_{\text{head}}(X^5) = 4$.

For a specific sequence of n flips $x^n = (x_0, \dots, x_{n-1}) \in \{H, T\}^n$ as an observation of X^n , if $N_{\text{head}}(x^n) = \sum_{i=0}^{n-1} \mathbf{1}_{\{x_i=H\}} = k$, then the probability of getting x^n in n flips is $\mathbb{P}(X^n = x^n) = \prod_{i=0}^{n-1} \mathbb{P}(X_i = x_i) = a^k b^{n-k}$, which only depends on the head count k . As a result, for $0 \leq k \leq n$, there are exactly $\binom{n}{k}$ outcomes of n flips containing k heads, each with the same probability $a^k b^{n-k}$. Let

$$S_k = \{A \subset \{0, 1, \dots, n-1\} : |A| = k\}, \tag{2}$$

Algorithm 2 \mathcal{A} : Generating a discrete distribution on the set $\{0, \dots, n-1\}$.

Input: A number n ; a sequence of flips from a biased coin X

Output: Integer in $\{0, \dots, n-1\}$

1: Flip the coin n times; denote the outcome by $X^n \in \{H, T\}^n$.

2: If the result is either all heads or all tails, discard the outcome and return to step 1.

3: Else **return** m when $X^n \in A_m$.

where $|A|$ means the cardinality of set A . Thus, S_k is the set of all subsets of $\{0, \dots, n-1\}$ containing k elements. Note that $|S_k| = \binom{n}{k}$, and each element in S_k corresponds to one and only one outcome of n flips with k heads in the following way:

$$\{i_1, \dots, i_k\} \in S_k \iff \dots H_{i_1} \dots H_{i_2} \dots H_{i_k} \dots, \tag{3}$$

where each i_t corresponds to the rank of an appearance of a head in the i_t th flip of n , $i_1 < i_2 < \dots < i_k$. As a result, we have the one-to-one correspondence

$$S_k \iff \{x^n \in \{H, T\}^n : N_{\text{head}}(x^n) = k\}, \tag{4}$$

and we also have $\mathbb{P}(X^n = x^n) = a^k b^{n-k}$ for all $x^n \in S_k$. Note that in the correspondences (3) and (4) we do not distinguish the left- and right-hand sides in the derivation below. And the equivalences are frequently used in the following proof.

Inspired by Von Neumann’s algorithm, we consider an algorithm generating a distribution on the set $\{0, \dots, n-1\}$. At each cycle we flip the coin n times, then the algorithm returns a number in $\{0, \dots, n-1\}$ except when the outcome is all heads or all tails. Define the sets $\{A_m : 0 \leq m \leq n-1\}$ to be a disjoint partition of $\bigsqcup_{1 \leq k \leq n-1} S_k$,

$$\bigsqcup_{k=1}^{n-1} S_k = \bigsqcup_{m=0}^{n-1} A_m,$$

where \bigsqcup means disjoint union. The algorithm, \mathcal{A} , is formally stated in Algorithm 2.

Let $\{Y_i = (X_{in}, \dots, X_{i(n-1)+n-1}) : i \geq 0\}$ be i.i.d. outcomes of n flips, and τ be the first time Y_i is neither all heads nor all tails. Then, for $0 \leq m \leq n-1$, we have

$$\begin{aligned} \mathbb{P}(\mathcal{A} = m) &= \mathbb{P}(Y_\tau \in A_m) \\ &= \frac{\mathbb{P}(X^n \in A_m)}{\mathbb{P}(X^n \in S_k \text{ for some } 1 \leq k \leq n-1)} \\ &= \frac{\sum_{k=1}^{n-1} \mathbb{P}(X^n \in A_m \cap S_k)}{\sum_{k=1}^{n-1} \mathbb{P}(X^n \in S_k)} = \frac{\sum_{k=1}^{n-1} |A_m \cap S_k| a^k b^{n-k}}{\sum_{k=1}^{n-1} |S_k| a^k b^{n-k}}. \end{aligned} \tag{5}$$

Let us consider a special case of algorithm \mathcal{A} where n is a prime, p . The reason for focusing on prime p comes from the fact in number theory that $p \mid \binom{p}{k} = |S_k|$ for all $1 \leq k \leq p-1$, where the symbol $|$ means ‘divides’. Then, for each k , we can partition S_k into disjoint p parts of equal size. For $1 \leq k \leq p-1$, assume that the choice of sets $\{A_m : 0 \leq m \leq p-1\}$ satisfies

$$|A_0 \cap S_k| = \dots = |A_{p-1} \cap S_k| = \frac{1}{p} |S_k|, \tag{6}$$

Algorithm 3 $\mathcal{A}_2(p)$: Generating a discrete uniform distribution on $\{0, \dots, p - 1\}$.

Input: A prime number p ; a sequence of flips from a biased coin X

Output: Integer in $\{0, \dots, p - 1\}$

1: Flip the coin p times, denote the outcome by $X^p \in \{H, T\}^p$.

2: If the result is either all heads or all tails, discard the outcome and return to step 1.

3: Else **return** $S_{\text{head}}(X^p) \pmod p$.

where the disjoint $\{A_m \cap S_k : 0 \leq m \leq p - 1\}$ partition S_k into p subsets of equal size. Based on (5) and (6), for $0 \leq m \leq p - 1$ we have

$$\mathbb{P}(\mathcal{A} = m) = \frac{\sum_{k=1}^{p-1} |A_m \cap S_k| a^k b^{n-k}}{\sum_{k=1}^{p-1} |S_k| a^k b^{n-k}} = \frac{\sum_{k=1}^{p-1} \frac{1}{p} |S_k| a^k b^{n-k}}{\sum_{k=1}^{p-1} |S_k| a^k b^{n-k}} = \frac{1}{p},$$

which means algorithm \mathcal{A} returns a uniform distribution on $\{0, \dots, p - 1\}$.

What remains is to find $\{A_m : 0 \leq m \leq p - 1\}$ satisfying (6). We can always first partition S_k into p subsets of equal size, and then define $\{A_m \cap S_k : 0 \leq m \leq p - 1\}$ to be these subsets, like the proposed method in [1]. However, this method has two disadvantages. First, there are many ways of partitioning S_k into subsets of equal size, and there is no widely accepted standard. Second, partitioning $\{S_k : 1 \leq k \leq p - 1\}$ and designing $\{A_m : 0 \leq m \leq p - 1\}$ need excessive time and storage cost, because there are 2^p different outcomes of p flips we need to handle, which grows exponentially as p increases. A preprocessing step of exponential time is unacceptable for an efficient algorithm.

With the help of the modulo p function, there is an ingenious way of designing $\{A_m : 0 \leq m \leq p - 1\}$ to satisfy (6). Based on the correspondence in (3), for $0 \leq m \leq p - 1$ we can indeed choose

$$A_m = \{X^p : S_{\text{head}}(X^p) = m \pmod p\}, \tag{7}$$

as we show in the next section.

4. Generating a uniform distribution on p (prime) elements

We provide an algorithm, $\mathcal{A}_2(p)$, that generates a discrete uniform distribution on the set $\{0, \dots, p - 1\}$, where p is a prime, as listed in Algorithm 3.

We need the following lemma before proving the main theory.

Lemma 1. *Let p be a prime number, and let $\{S_k : 1 \leq k \leq p - 1\}$ consist of all subsets of $\{0, \dots, p - 1\}$ having k elements. For fixed k , let $\{S_k^m : 0 \leq m \leq p - 1\}$ be defined by*

$$S_k^m = \left\{ \{i_1, \dots, i_k\} \in S_k : \sum_{j=1}^k i_j = m \pmod p \right\}. \tag{8}$$

Note that $S_k^m = A_m \cap S_k$, where A_m is defined in (7).

Then we have

$$|S_k^m| = \frac{1}{p} \binom{p}{k} \quad \text{for all } 1 \leq k \leq p - 1 \text{ and } 0 \leq m \leq p - 1.$$

Proof. For fixed $1 \leq k \leq p - 1$, consider a permutation on S_k defined by

$$f(\{i_1, \dots, i_k\}) = \{(i_1 + 1) \pmod p, \dots, (i_k + 1) \pmod p\}.$$

Denote by f^0 the identity function id . Let $\langle f \rangle$ be the subgroup generated by f . We need to show that $\langle f \rangle = \{f^0 = \text{id}, f^1, \dots, f^{p-1}\}$. Since we know that $f^p = \text{id}$, we need to show $f^s \neq \text{id}$ for $1 \leq s \leq p - 1$.

If $f^s = \text{id}$ for some $1 \leq s \leq p - 1$, then we have

$$f^s(\{i_1, \dots, i_k\}) = \{(i_1 + s) \pmod p, \dots, (i_k + s) \pmod p\} = \{i_1, \dots, i_k\},$$

from which we have $\sum_{j=1}^k (i_j + s) = \sum_{j=1}^k i_j \pmod p$. The equality above shows $p \mid ks$, which implies $p \mid k$ or $p \mid s$, leading to a contradiction since $1 \leq k, s \leq p - 1$.

Let the group $\langle f \rangle$ act on S_k . For $\{i_1, \dots, i_k\} \in S_k$, let $O_{\{i_1, \dots, i_k\}}$ denote the orbit of $\{i_1, \dots, i_k\}$ under group action,

$$O_{\{i_1, \dots, i_k\}} = \{\{i_1^s, \dots, i_k^s\} := f^s(\{i_1, \dots, i_k\}) \text{ for } 0 \leq s \leq p - 1\}.$$

The theory of group action tells us that S_k can be divided to disjoint orbits with equal size p . In addition, for any $\{i_1, \dots, i_k\} \in S_k$, when s varies from 0 to $p - 1$, $\sum_{j=1}^k i_j^s \pmod p$ takes all values in $\{0, \dots, p - 1\}$.

If the claim above were not true, then there would exist $0 \leq s_1 < s_2 \leq p - 1$ such that

$$\sum_{j=1}^k i_j^{s_1} = \sum_{j=1}^k i_j^{s_2} \pmod p \Rightarrow \sum_{j=1}^k (i_j + s_1) = \sum_{j=1}^k (i_j + s_2) \pmod p.$$

The equality above shows that $p \mid k(s_2 - s_1)$, which implies $p \mid k$ or $p \mid (s_2 - s_1)$, leading to a contradiction since $1 \leq k, s_2 - s_1 \leq p - 1$.

The argument above shows that S_k is a union of disjoint orbits of equal size p . And in each orbit, for $0 \leq m \leq p - 1$, there exists one and only one element belonging to S_k^m , which means the $\{S_k^m : 0 \leq m \leq p - 1\}$ partition S_k into p subsets with equal size, and the proof is complete. \square

Figure 1 presents a special case to show the idea of the proof, with $p = 7$ and $k = 3$; the proof proceeds as shown.

Next, we prove the main theorem on algorithm $\mathcal{A}_2(p)$.

Theorem 1. *Let X denote a biased coin with probability $a \in (0, 1)$ of getting a head, and probability $b = 1 - a$ of getting a tail. For a prime p , $\mathcal{A}_2(p)$ has the following properties:*

- (i) $\mathcal{A}_2(p)$ terminates in a finite number of flips with probability 1. The algorithm returns a uniform distribution on $\{0, \dots, p - 1\} : \mathbb{P}(\mathcal{A}_2(p) = m) = 1/p$ for all $0 \leq m \leq p - 1$.
- (ii) The expected number of flips terminating $\mathcal{A}_2(p)$ is $p/(1 - a^p - b^p)$, which means that when p is large, the average running time approximates to the linear $O(p)$.
- (iii) By letting $p = 2$, $\mathcal{A}_2(2)$ is exactly Von Neumann's algorithm \mathcal{A}_1 .

Proof. Let $\mathbf{X}^p = (X_0, \dots, X_{p-1})$ be the outcome of p flips of a biased coin, a random variable taking values in $\{H, T\}^p$. Based on the correspondences in (3) and (4), and the definition of S_k^m in (8), each $\mathbf{x}^p \in \{H, T\}^p$ corresponds to one and only one element in S_k^m by $N_{\text{head}}(\mathbf{x}^p) = k$ and $S_{\text{head}}(\mathbf{x}^p) = m \pmod p$ for some k and m , where $N_{\text{head}}(\mathbf{x}^p)$ and $S_{\text{head}}(\mathbf{x}^p)$ in (1) are the count and rank sum of heads respectively. Recall the definition of S_k in (2); then, by Lemma 1, the $\{S_k^m : 0 \leq m \leq p - 1\}$ partition S_k into p subsets with equal size.

P₁=7 k=3

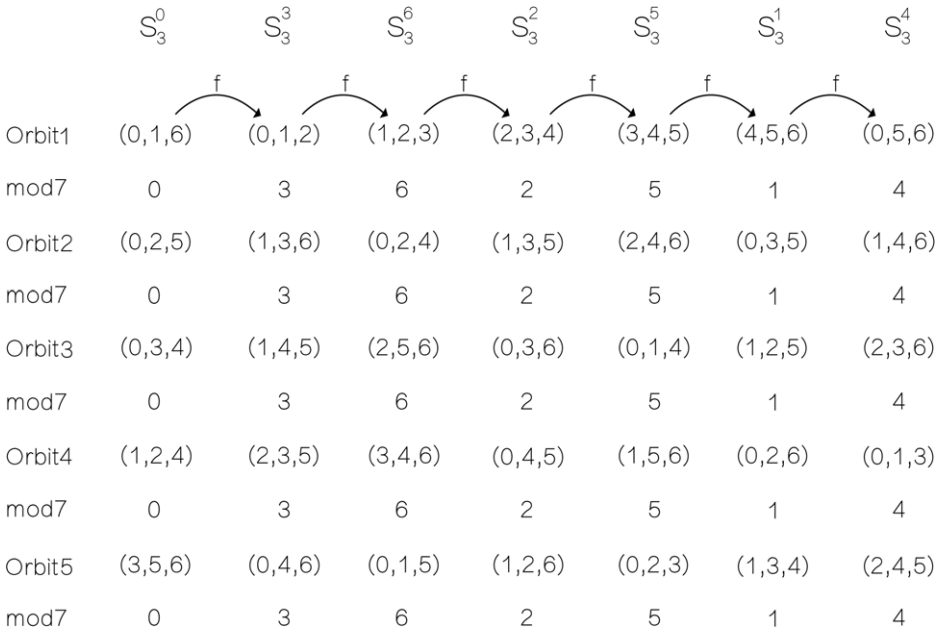


FIGURE 1. An example of the method in the proof.

Let $\{Y_i = (X_{ip}, \dots, X_{i(p-1)}) : i \geq 0\}$ be i.i.d. outcomes of p flips, and τ be the first time Y_i is neither all heads nor all tails. Then, for $0 \leq m \leq p - 1$, we have

$$\begin{aligned}
 \mathbb{P}(\mathcal{A}_2(p) = m) &= \mathbb{P}(S_{\text{head}}(Y_\tau) = m \pmod{p}) \\
 &= \mathbb{P}(S_{\text{head}}(X^p) = m \pmod{p} \mid X^p \text{ is neither all heads nor all tails}) \\
 &= \frac{\mathbb{P}(S_{\text{head}}(X^p) = m \pmod{p}, N_{\text{head}}(X^p) = k \text{ for some } 1 \leq k \leq p - 1)}{\mathbb{P}(N_{\text{head}}(X^p) = k \text{ for some } 1 \leq k \leq p - 1)} \\
 &= \frac{\sum_{k=1}^{p-1} \mathbb{P}(S_{\text{head}}(X^p) = m \pmod{p}, N_{\text{head}}(X^p) = k)}{\sum_{k=1}^{p-1} \mathbb{P}(N_{\text{head}}(X^p) = k)} \\
 &= \frac{\sum_{k=1}^{p-1} |S_k^m| a^k b^{p-k}}{\sum_{k=1}^{p-1} |S_k| a^k b^{p-k}} = \frac{1}{p},
 \end{aligned}$$

where the last identity is implied by the fact that $|S_k^m| = \frac{1}{p} \binom{p}{k} = \frac{1}{p} |S_k|$.

Let E denote the expected number of flips terminating $\mathcal{A}_2(p)$. Hence, E satisfies

$$E = p\mathbb{P}(N_{\text{head}}(X^p) = k \text{ for some } 1 \leq k \leq p - 1) + (p + E)\mathbb{P}(X^p \text{ is all heads or all tails}),$$

Algorithm 4 $\mathcal{A}_3(n)$: Generating a discrete uniform distribution on the set $\{0, \dots, n-1\}$.

Input: A sequence of flips; an integer n ; a set \mathcal{M} containing all prime factors of n , where each prime repeats as many times as its multiplicity in the decomposition of n

Output: Integer in $\{0, \dots, n-1\}$

- 1: Set $r = 0$.
 - 2: **while** $\mathcal{M} \neq \emptyset$ **do**
 - 3: Take a prime p' out of \mathcal{M} .
 - 4: $n = n/p'$.
 - 5: Run $\mathcal{A}_2(p')$, and let t denote the return value.
 - 6: $r = r + t \cdot n$.
 - 7: **return** r .
-

from which we have

$$E = \frac{p}{1 - \mathbb{P}(X^p \text{ is all heads or all tails})} = \frac{p}{1 - a^p - b^p}. \quad \square$$

We have also come up with a creative and short proof for Theorem 1(i) using random variables in the residue class \mathbb{Z}_p ; see Appendix A.

5. Generating a uniform distribution on n elements

Let n be any positive integer with prime factorization $n = \prod_{i=1}^s p_i^{t_i}$. Let \mathcal{M} be the set of all prime factors of n considering multiplicity, which means p_i appears t_i times in \mathcal{M} . The algorithm $\mathcal{A}_3(n)$ shown in Algorithm 4 generates a discrete uniform distribution on the set $\{0, \dots, n-1\}$ in an iterative way.

The following theorem shows the validity of algorithm $\mathcal{A}_3(n)$.

Theorem 2. For any integer n , $\mathcal{A}_3(n)$ has the following properties:

- (i) $\mathcal{A}_3(n)$ terminates in a finite number of flips with probability 1. It returns a uniform distribution on $\{0, \dots, n-1\}$: $\mathbb{P}(\mathcal{A}_3(n) = m) = 1/n$ for all $0 \leq m \leq n-1$.
- (ii) When n has prime factorization $\prod_{i=1}^s p_i^{t_i}$, the expected number of flips terminating $\mathcal{A}_3(n)$ is

$$\sum_{i=1}^s \frac{t_i p_i}{1 - a^{p_i} - b^{p_i}}.$$

Therefore, the average running time is approximately $\sum_{i=1}^s t_i p_i$ for large n .

- (iii) The overall order of the average running time is $O(n/\log(n))$.

Proof. To show (i), note that each outcome of $\mathcal{A}_3(n)$ corresponds to one and only one sequence of outcomes of $\mathcal{A}_2(p_i)$. For this fact, first we consider a simplified case where $n = p_1 p_2$ is a product of two prime numbers p_1 and p_2 , and p_1 may equal p_2 .

Given $n = p_1 p_2$, then $\mathcal{M} = \{p_1, p_2\}$. Suppose we first get p_1 from \mathcal{M} and then p_2 . Then the outcomes $\mathcal{A}_2(p_1) = m_1$ and $\mathcal{A}_2(p_2) = m_2$ correspond to the outcome $\mathcal{A}_3(n) = m_1 p_2 + m_2$. Since $0 \leq m_1 \leq p_1 - 1$ and $0 \leq m_2 \leq p_2 - 1$, we have the range for $\mathcal{A}_3(n)$:

$$0 \leq \mathcal{A}_3(n) \leq (p_1 - 1)p_2 + p_2 - 1 = n - 1,$$

which shows that $\mathcal{A}_3(n) \in \{0, \dots, n-1\}$. Note that, for $0 \leq m \leq n-1$, there exists one and only one pair of (m_1, m_2) as

$$\left(\left\lfloor \frac{m}{p_2} \right\rfloor, m - \left\lfloor \frac{m}{p_2} \right\rfloor p_2 \right)$$

satisfying the equation $m = m_1 p_2 + m_2$ ($0 \leq m_1 \leq p_1 - 1$, $0 \leq m_2 \leq p_2 - 1$). So, the outcome $\mathcal{A}_3(n) = m$ corresponds to the outcomes $\mathcal{A}_2(p_1) = m_1$ and $\mathcal{A}_2(p_2) = m_2$.

For the general case $n = \prod_{i=1}^s p_i^{t_i}$, based on the same method as above, we conclude that, for each m , there exists a unique set $\{m_{p'} : p' \in \mathcal{M}\}$ such that the outcome $\mathcal{A}_3(n) = m$ corresponds to the outcomes $\mathcal{A}_2(p') = m_{p'}$ ($p' \in \mathcal{M}$). Therefore, the probability of $\mathcal{A}_3(n) = m$ is

$$\mathbb{P}(\mathcal{A}_3(n) = m) = \prod_{p' \in \mathcal{M}} \mathbb{P}(\mathcal{A}_2(p') = m_{p'}) = \prod_{i=1}^s \left(\frac{1}{p_i} \right)^{t_i} = \frac{1}{n} \quad \text{for all } 0 \leq m \leq n-1.$$

To prove (ii), note that, for $n = \prod_{i=1}^s p_i^{t_i}$, the set \mathcal{M} contains each prime factor p_i t_i times. By the iterative construction of $\mathcal{A}_3(n)$, we need to run $\mathcal{A}_3(p_i)$ once every time we pick p_i from \mathcal{M} . Based on Theorem 1(ii), the expected number of flips for $\mathcal{A}_2(p_i)$ is $p_i / (1 - a^{p_i} - b^{p_i})$, from which we conclude that the expected number of flips terminating $\mathcal{A}_3(n)$ is

$$\sum_{i=1}^s \frac{t_i p_i}{1 - a^{p_i} - b^{p_i}}.$$

To analyze the average running time of the algorithm $\mathcal{A}_3(n)$, define the function $c(n) = \sum_{i=1}^s t_i p_i$ to be the sum of prime factors of n multiplied by their multiplicity, which is a good approximation to the average running time of $\mathcal{A}_3(n)$ according to Theorem 2(ii). We see that, for prime numbers, the complexity is linear. For composite numbers, the complexity is sub-linear. For $n = p_1^{t_1}$, since $c(n) = t_1 p_1$, the average running time is almost $\log(n)$. We have the following theorem from number theory [4, Corollary 2.11]:

$$\lim_{N \rightarrow \infty} \left| \left\{ 2 \leq n \leq N : c(n) < \frac{n}{\log^{1-\varepsilon}(n)} \right\} / N = 1 \quad \text{for all } 0 < \varepsilon < 1. \right.$$

So, we have an overall sublinear $O(n / \log(n))$ complexity for the algorithm $\mathcal{A}_3(n)$. □

Remark 1. In [2], another method for generating a discrete uniform distribution on the set $\{0, \dots, n-1\}$ was proposed. Elias' method needs Elias' function mapping outcomes of the random source to target values. However, unlike Theorem 2(iii), the efficiency of Elias' method is defined by complicated mathematical formulas without analytic and concise form, and is thus hard to analyze theoretically. Besides, Elias' method suffers the same problem as Dijkstra's method mentioned in Section 3. The computation of Elias' function, an essential preprocessing step of Elias' method, is computationally inefficient, and the storage of Elias' function is also an excessive space cost.

Appendix A. A new proof for Theorem 1(i)

Consider random variables taking values in $\mathbb{Z}_p = \{\bar{0}, \dots, \overline{p-1}\}$, where \bar{i} represents the residual class of i modulo p . Regard $\bar{0}$ as a tail and $\bar{1}$ as a head. Let X denote the outcome of a flip satisfying $\mathbb{P}(X = \bar{0}) = a$ and $\mathbb{P}(X = \bar{1}) = b$. Let X_0, \dots, X_{p-1} be independent copies

of X . Define $X^p = (X_0, \dots, X_{p-1})$ to be the outcome of p flips. We then have the following equivalences:

$$\begin{aligned} X^p \text{ is all heads or all tails} &\iff X_i = \bar{0} \text{ for all } 0 \leq i \leq p-1 \\ &\text{or } X_i = \bar{1} \text{ for all } 0 \leq i \leq p-1 \\ &\iff \sum_{i=0}^{p-1} X_i = \bar{0}, \\ S_{\text{head}}(X^p) = m(\text{mod } p) &\iff \sum_{i=0}^{p-1} \bar{i} \cdot X_i = \bar{m}. \end{aligned}$$

Also note that, for any permutation σ , we have $(X_0, \dots, X_{p-1}) \stackrel{d}{=} (X_{\sigma(0)}, \dots, X_{\sigma(p-1)})$, since all the X_i are i.i.d. In the following, we let σ denote the special permutation

$$\sigma = \begin{pmatrix} 0 & 1 & \dots & p-2 & p-1 \\ 1 & 2 & \dots & p-1 & 0 \end{pmatrix}.$$

For fixed $t \neq \bar{0} \in \mathbb{Z}_p$, we have

$$\begin{aligned} \mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = \bar{0}, \sum_{i=0}^{p-1} X_i = t\right) &= \mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i + \sum_{i=0}^{p-1} X_i = t, \sum_{i=0}^{p-1} X_i = t\right) \\ &= \mathbb{P}\left(\sum_{i=0}^{p-1} \overline{i+1} \cdot X_i = t, \sum_{i=0}^{p-1} X_i = t\right) \\ &= \mathbb{P}\left(\sum_{i=0}^{p-1} \overline{i+1} \cdot X_{\sigma(i)} = t, \sum_{i=0}^{p-1} X_{\sigma(i)} = t\right) \\ &= \mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = t, \sum_{i=0}^{p-1} X_i = t\right). \end{aligned}$$

Note that any $t \neq \bar{0}$ can generate \mathbb{Z}_p . By iterating the derivation above, we have

$$\mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = k, \sum_{i=0}^{p-1} X_i = t\right) = \mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = s, \sum_{i=0}^{p-1} X_i = t\right) \text{ for all } k, s \in \mathbb{Z}_p.$$

Summing over $t \neq \bar{0}$ on both sides of the above equation, we have, for $k, s \in \mathbb{Z}_p$,

$$\mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = k, \sum_{i=0}^{p-1} X_i \neq \bar{0}\right) = \sum_{t \neq \bar{0}} \mathbb{P}\left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = k, \sum_{i=0}^{p-1} X_i = t\right)$$

$$\begin{aligned}
 &= \sum_{t \neq \bar{0}} \mathbb{P} \left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = s, \sum_{i=0}^{p-1} X_i = t \right) \\
 &= \mathbb{P} \left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = s, \sum_{i=0}^{p-1} X_i \neq \bar{0} \right),
 \end{aligned}$$

which implies, for $k, s \in \mathbb{Z}_p$,

$$\mathbb{P} \left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = k \mid \sum_{i=0}^{p-1} X_i \neq \bar{0} \right) = \mathbb{P} \left(\sum_{i=0}^{p-1} \bar{i} \cdot X_i = s \mid \sum_{i=0}^{p-1} X_i \neq \bar{0} \right).$$

This equality is equivalent to the statement

$$\begin{aligned}
 &\mathbb{P}(S_{\text{head}}(X^p) = k \pmod{p} \mid X^p \text{ is neither all heads nor all tails}) \\
 &= \mathbb{P}(S_{\text{head}}(X^p) = s \pmod{p} \mid X^p \text{ is neither all heads nor all tails})
 \end{aligned}$$

for all $0 \leq k, s \leq p - 1$, as desired.

Acknowledgements

The author acknowledges Prof. Mei Wang at UChicago for helpful discussions and advice, and thanks PhD candidate Haoyu Wei at UCSD for useful suggestions and kind support. The author also acknowledges the editor of *Journal of Applied Probability* and the two anonymous referees for their valuable comments and remarks.

Funding information

There are no funding bodies to thank relating to the creation of this article.

Competing interests

There were no competing interests to declare which arose during the preparation or publication process of this article.

References

- [1] DIJKSTRA, E. W. (1990). Making a fair roulette from a possibly biased coin. *Inf. Process. Lett.* **36**, 193.
- [2] ELIAS, P. (1972). The efficient construction of an unbiased random sequence. *Ann. Math. Statist.* **43**, 865–870.
- [3] HOEFFDING, W. AND SIMONS, G. (1994). Unbiased coin tossing with a biased coin. *Ann. Math. Statist.* **41**, 341–352.
- [4] JAKIMCZUK, R. (2012). Sum of prime factors in the prime factorization of an integer. *Int. Math. Forum* **7**, 2617–2621.
- [5] PAE, S. (2005). Random number generation using a biased source. Doctoral thesis, University of Illinois Urbana-Champaign.
- [6] STOUT, Q. F. AND WARREN, B. (1984). Tree algorithms for unbiased coin tossing with a biased coin. *Ann. Prob.* **12**, 212–222.
- [7] VON NEUMANN, J. (1951). Various techniques used in connection with random digits. *J. Res. Nat. Bureau Standards Appl. Math.* **12**, 36–38.