MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Polynomial Time Approximate/Perfect Samplers for Discretized Dirichlet Distribution

Tomomi MATSUI Mitsuo MOTOKI
Naoyuki KAMATANI Shuji KIJIMA

METR 2006–09 January 2006

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm

Polynomial Time Approximate/Perfect Samplers for Discretized Dirichlet Distribution[⋆]

Tomomi Matsui[⋆⋆], Mitsuo Motoki, Naoyuki Kamatani, and Shuji Kijima

[1] Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.
`http://www.simplex.t.u-tokyo.ac.jp/~tomomi/`

[2] Department of Information Processing,
School of Information Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan.
`mmotoki@jaist.ac.jp`

[3] Institute of Rheumatology, Tokyo Women's Medical University,
10-22 Kawada-cho, Shinjuku-ku, Tokyo 162-0054, Japan.

[4] Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.
`kijima@simplex.t.u-tokyo.ac.jp`

Abstract. The Dirichlet distribution often appears as prior and posterior distribution for the multinomial distribution in some statistical methods for bioinformatics. In this paper, we propose two Markov chains for sampling a random vector distributed according to a discretized Dirichlet distribution. In each transition of our chains, we need to sample a random variable according to a discretized beta distribution (2-dimensional Dirichlet distribution). The mixing rates of our chains are bounded by $\mathrm{O}(n^2 \log \Delta)$ and $\mathrm{O}(n^3 \log \Delta)$, respectively, where n is the dimension (the number of parameters), and $1/\Delta$ is the grid size for discretization. The obtained bounds do not depend on the magnitudes of parameters. To show the (weak) polynomiality, we apply the path coupling method carefully and show that our chains are rapidly mixing.

Our second chain gives a perfect (exact) sampling algorithm according to a discretized Dirichlet distribution. Our algorithm is a monotone coupling from the past (monotone CFTP) algorithm, which is a Las Vegas type randomized algorithm. Our perfect sampler simulates transitions of our second chain $\mathrm{O}(n^3 \ln \Delta)$ times in expectation.

We also report results of simulations, which indicate that our two chains have totally different features in practice.

[⋆] Some of the results in this paper appeared without detailed discussions in the extended abstract [16] in the Proceedings of ISAAC 2003 (The 14th Annual International Symposium on Algorithms and Computation).

[⋆⋆] Supported by Superrobust Computation Project of the 21st Century COE Program "Information Science and Technology Strategic Core."

1 Introduction

In this paper, we propose an approximate sampler and an exact sampler according to a discretized Dirichlet distribution. Both sampling algorithms are based on Markov chains and whose time complexities are bounded by a polynomial of the dimension (the number of parameters) and the logarithm of the grid size for discretization. We propose two chains whose stationary distributions are the discretized Dirichlet distribution. In each transition of our Markov chains, we need a random sample according to a discretized beta distribution (2-dimensional Dirichlet distribution). We show that our first chain is rapidly mixing, that is, the mixing time of our chain is bounded by $(1/2)n(n-1)\ln((\Delta-n)\varepsilon^{-1})$ where n is the dimension (the number of parameters), $1/\Delta$ is the grid size for discretization, and ε is the error bound. The mixing time of our second chain is bounded by $n(n-1)^2\ln(n(\Delta-n)/(2\varepsilon))$. To show the (weak) polynomiality of our chains, we employ the path coupling method. We also propose an exact sampling algorithm based on a monotone coupling from the past algorithm using our second chain. Our algorithm simulates transitions of our second chain $\mathrm{O}(n^3\ln\Delta)$ times in expectation.

We begin with a quick overview of some applications of our sampling algorithm appearing in bioinformatics. Statistical methods are widely studied in bioinformatics since they are powerful tools to discover genes causing a (common) disease from a number of observed data. These methods often use EM algorithm, Markov chain Monte Carlo method, Gibbs sampler, and so on. The Dirichlet distribution is a distribution over vectors of positive numbers in which the sum total is equal to 1. The distribution appears as prior and posterior distribution for the multinomial distribution in these methods since the Dirichlet distribution is the conjugate prior of parameters of the multinomial distribution [23]. For example, Niu, Qin, Xu, and Liu proposed a Bayesian haplotype inference method [18], which decides phased (paternal and maternal) individual genotypes (diplotype configuration for each subject) probabilistically. This method is based on Gibbs sampler. In their method, the Dirichlet distribution is used to update population haplotype frequencies, i.e., parameters of the multinomial distribution, for each iteration. That is to say, for each iteration starting from the Dirichlet distribution with some appropriate parameters, the parameters of the multinomial distribution are updated from the posterior distribution which is the Dirichlet distribution with updated parameters conditional on the "imputed" events.

Another example is a population structure inferring algorithm by Pritchard, Stephens, and Donnely [19]. Their algorithm is based on MCMC method. For each step of MCMC, the Dirichlet distribution with two distinct sets of parameters are used to sample allele frequencies in each population and admixture proportions for each individual. Similar to the first example, these two sets of parameters are updated at each iteration. The Dirichlet distribution was also used to estimate inbreeding coefficient and effective size from allele frequency changes [15], and to perform a meta-analysis of studies on the association of polymorphisms and risk of a disease [4]. In the paper [3], Burr used the distribution to examine the quasi-equilibrium theory for the distribution of rare alleles in a subdivided population. Kitada, Hayashi and Kishino used the Dirichlet distribution to estimate genetic distance between populations and effective population size [14]. To approximate the conditional genotypic probabilities for microsatellite loci for forensic medicine, Graham, Curran, and Weir used the distribution [11].

An application also appears in the area of econometrics. In [5], Chiang, Chib and Narasimhan dealt with set-brand choice model that is capable of accounting for the heterogeneity in consideration set and in the parameter of the brand choice model. They modeled consideration set heterogeneity by introducing a probability function on the possible subsets of given brands for each householder. In their consideration set model, the probability function (the vector of probabilities) is assumed to be distributed across the population according to a Dirichlet distribution. The model is estimated by Markov chain Monte Carlo sampling procedure and is applied to a scanner panel data. Their procedure generates 15-dimensional Dirichlet random vectors in each iteration and updates the Dirichlet parameters.

In the above examples, the Dirichlet distribution appears with various dimensions and various parameters. Thus we need an efficient algorithm for sampling from the Dirichlet distribution with arbitrary dimensions and parameters. One approach of sampling from a (continuous) Dirichlet distribution is by combining random variables distributed according to gamma distributions. More precisely, we obtain a Dirichlet random vector by normalizing the vector of obtained random variables so that the sum total is equal to 1 (see [8] for example). In this approach, the number of required samples according to gamma distributions is equal to n, the dimension of the Dirichlet distribution. Though we can sample from the gamma distribution by using rejection sampling, the ratio of rejection becomes higher as the parameter is smaller. Thus, it does not seem effective way for small parameters. Additionally, this approach has a serious problem. When we generate samples according to gamma distributions by using a digital computer, we need to discretize the domain and thus the set of vectors generated by this approach with positive probability is finite (and contained in an $n-1$ dimensional simplex). Then it is easy to see that the distribution of the set of vectors generated by the above procedure with positive probability is not uniform, but sparse around the center of the simplex.

This paper deals with the discretized Dirichlet distribution which is obtained by uniformly discretizing an $n-1$ dimensional simplex (a domain of a Dirichlet distribution). Discretization of the domain enables us to construct simple and natural Markov chains based on the Metropolis algorithm. As described later, the mixing times of our chains are linear to $\log \Delta$ (the logarithm of the inverse of a grid size $1/\Delta$). Thus, we can simulate the Dirichlet distribution by employing sufficiently small grid size.

We propose polynomial approximate sampler and polynomial time perfect (exact) sampling algorithm according to a discretized Dirichlet distribution. Our perfect sampling algorithm is based on a monotone coupling from the past

(CFTP) algorithm. The (monotone) CFTP algorithm was proposed by Propp and Wilson in 1996 [20, 21], which produces exact samples from the limit distribution of a monotone Markov chain. Monotone CFTP algorithm simulates infinite time transitions of a monotone Markov chain in (probabilistically) finite time. To employ their result, we propose our second chain whose mixing time is slower than our first chain but satisfies the monotonicity.

The organization of this paper is as follows. In the next section, we discuss the influence of discretization by showing the difference between the (original) Dirichlet distribution and its discretized version. We also give a brief review of path coupling method and (monotone) CFTP algorithm. In Section 3, we propose two Markov chains and describe our main results. Section 4 shows the mixing times of our chains. In Section 5, we discuss the correctness and the time complexity of our perfect sampling algorithm. Section 6 gives results of simulations. In the last section, we discuss results of simulations and describe related works. Appendix section gives proofs of key properties one of which is called *alternating inequalities.*

2 Definitions and Notations

2.1 Discretized Dirichlet Distribution

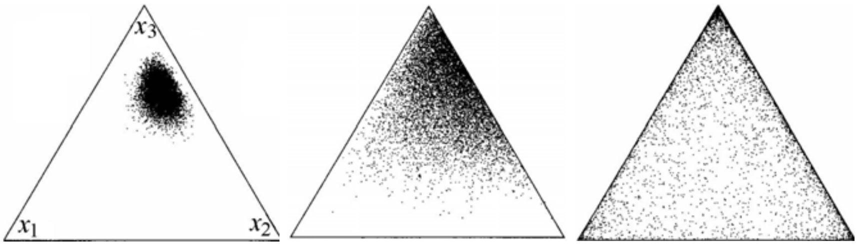
In this paper, we denote the set of integers (non-negative or positive integers) by Z (Z_+, Z_{++}) and the set of reals (non-negative or positive reals) by R (R_+, R_{++}). Dirichlet random vector $P = (P_1, P_2, \ldots, P_n)$ with non-negative Dirichlet parameters $u_1, \ldots, u_n \in \mathrm{R}_+$ is a vector of random variables that admits the probability density function

$$\frac{\Gamma(\sum_{i=1}^n u_i)}{\prod_{i=1}^n \Gamma(u_i)} \prod_{i=1}^n p_i^{u_i - 1}$$

defined on the set $\{(p_1, p_2, \ldots, p_n) \in \mathrm{R}_{++}^n \mid p_1 + \cdots + p_n = 1, p_1, p_2, \ldots, p_n > 0\}$ where $\Gamma(u)$ is the gamma function. Throughout this paper, we assume that $n \geq 2$. The statistics of the Dirichlet distribution with parameters $(u_1, \ldots, u_n)$ are given as follows. For each random variable P_i, $\mathrm{E}[P_i] = u_i/u_0$ and $\mathrm{Var}[P_i] = \frac{u_i(u_0 - u_i)}{u_0^2(u_0+1)}$ where $u_0 = \sum_{i=1}^n u_i$. Each pair of random variables P_i and P_j with $(i \neq j)$ satisfies that $\mathrm{Cov}[P_i, P_j] = \frac{-u_i u_j}{u_0^2(u_0+1)}$. Figure 1, derived from [8], illustrates 3-dimensional Dirichlet distributions by plotting 10^9 points (for each case) generated according to 3-dimensional Dirichlet distributions. As shown in Figure 1 (c), when values of all parameters are less than 1, the distribution function becomes convex and that causes a difficulty to construct a rapidly mixing chain whose stationary distribution is the Dirichlet distribution.

Given an integer $\Delta \geq n$, we discretize the domain with grid size $1/\Delta$ and obtain a discrete set of integer vectors Ω defined by

$$\Omega \stackrel{\text{def.}}{=} \{(x_1, x_2, \ldots, x_n) \in \mathrm{Z}_{++}^n \mid x_1 + \cdots + x_n = \Delta,\ x_1, x_2, \ldots, x_n > 0\}.$$



(a) $(u_1, u_2, u_3) = (10, 20, 50)$. (b) $(u_1, u_2, u_3) = (1, 2, 5)$. (c) $(u_1, u_2, u_3) = (0.1, 0.2, 0.5)$.

Fig. 1. 3-dimensional Dirichlet distributions [8].

A discretized Dirichlet random vector with non-negative Dirichlet parameters $u_1, \ldots, u_n$ is a random vector $X = (X_1, \ldots, X_n) \in \Omega$ with the distribution

$$\Pr[X = (x_1, \ldots, x_n)] = C_\Delta \prod_{i=1}^{n} (x_i/\Delta)^{u_i - 1}$$

where C_Δ is the partition function (normalizing constant) defined by

$$(C_\Delta)^{-1} \overset{\text{def.}}{=} \sum_{\boldsymbol{x} \in \Omega} \prod_{i=1}^{n} (x_i/\Delta)^{u_i - 1}.$$

Table 1 shows the influence of discretization. For some discretized Dirichlet distributions, we calculated the statistics, $\mathrm{E}_\Delta[P_i]$, $\mathrm{Var}_\Delta[P_i]$, and $\mathrm{Cov}_\Delta[P_i, P_j]$ by a brute force method.

2.2 Path Coupling Method

Given a pair of probabilistic distributions ν_1 and ν_2 on a finite state space Ω', the *total variation distance* between ν_1 and ν_2 is defined by $\mathrm{D_{TV}}(\nu_1, \nu_2) \overset{\text{def.}}{=} \frac{1}{2} \sum_{x \in \Omega'} |\nu_1(x) - \nu_2(x)|$. A *mixing time* $\tau(\varepsilon)$ of an ergodic Markov chain on a finite state space Ω' is defined by

$$\tau(\varepsilon) \overset{\text{def.}}{=} \max_{x \in \Omega'} \{\min\{t \mid \forall s \geq t,\ \mathrm{D_{TV}}(\pi, P_x^s) \leq \varepsilon\}\}$$

where π is the stationary distribution and P_x^s is the probabilistic distribution of the chain at time s with initial state x. The value $\tau(1/\mathrm{e})$ is called a *mixing rate* and denoted by τ when there is no ambiguity.

Path Coupling Theorem is a useful technique for bounding the mixing time.

Theorem 1. (Path Coupling [1, 2]) *Let MC be a finite ergodic Markov chain with state space Ω'. Let $G' = (\Omega', \mathcal{E}')$ be a connected undirected graph with vertex set Ω' and edge set $\mathcal{E}' \subseteq \binom{\Omega'}{2}$. Let $l : \mathcal{E}' \to \mathrm{R}$ be a positive length defined on*

Table 1. Influence of discretization.

(u_1, u_2, u_3, u_4)	maximum difference of statistic	$\varDelta$		
		10	50	100
$(1, 1, 1, 1)$	$\lvert \mathrm{E}_{\varDelta}[P_i] - \mathrm{E}[P_i] \rvert$	0	0	0
	$\lvert \mathrm{Var}_{\varDelta}[P_i] - \mathrm{Var}[P_i] \rvert$	0.015	0.003	0.0015
	$\lvert \mathrm{Cov}_{\varDelta}[P_i, P_j] - \mathrm{Cov}[P_i, P_j] \rvert$	0.005	0.001	0.0005
$(4, 3, 2, 1)$	$\max\left(\lvert \mathrm{E}_{\varDelta}[P_i] - \mathrm{E}[P_i] \rvert\right)$	0.051	0.0092	0.0046
	$\max\left(\lvert \mathrm{Var}_{\varDelta}[P_i] - \mathrm{Var}[P_i] \rvert\right)$	0.0036	0.00049	0.00023
	$\max\left(\lvert \mathrm{Cov}_{\varDelta}[P_i, P_j] - \mathrm{Cov}[P_i, P_j] \rvert\right)$	0.0080	0.0074	0.0073
$(0.1, 0.1, 0.1, 0.1)$	$\lvert \mathrm{E}_{\varDelta}[P_i] - \mathrm{E}[P_i] \rvert$	0	0	0
	$\lvert \mathrm{Var}_{\varDelta}[P_i] - \mathrm{Var}[P_i] \rvert$	0.11	0.071	0.061
	$\lvert \mathrm{Cov}_{\varDelta}[P_i, P_j] - \mathrm{Cov}[P_i, P_j] \rvert$	0.035	0.024	0.020
$(0.4, 0.3, 0.2, 0.1)$	$\max\left(\lvert \mathrm{E}_{\varDelta}[P_i] - \mathrm{E}[P_i] \rvert\right)$	0.13	0.10	0.092
	$\max\left(\lvert \mathrm{Var}_{\varDelta}[P_i] - \mathrm{Var}[P_i] \rvert\right)$	0.090	0.055	0.045
	$\max\left(\lvert \mathrm{Cov}_{\varDelta}[P_i, P_j] - \mathrm{Cov}[P_i, P_j] \rvert\right)$	0.051	0.042	0.040
$(2, 1.5, 1, 0.5)$	$\max\left(\lvert \mathrm{E}_{\varDelta}[P_i] - \mathrm{E}[P_i] \rvert\right)$	0.079	0.029	0.019
	$\max\left(\lvert \mathrm{Var}_{\varDelta}[P_i] - \mathrm{Var}[P_i] \rvert\right)$	0.014	0.0032	0.0019
	$\max\left(\lvert \mathrm{Cov}_{\varDelta}[P_i, P_j] - \mathrm{Cov}[P_i, P_j] \rvert\right)$	0.015	0.013	0.013

the edge set. For any pair of vertices $\{x, y\}$ of G', the distance between x and y, denoted by $d(x, y)$ and/or $d(y, x)$, is the length of a shortest path between x and y, where the length of a path is the sum of the lengths of edges in the path. For any $x \in \varOmega'$, we define $d(x, x) = 0$. (The positivity of edge lengths implies that $x \neq y \Leftrightarrow d(x, y) > 0$.) Suppose that there exists a joint process $(X, Y) \mapsto (X', Y')$ with respect to MC satisfying that whose marginals are a faithful copy of MC and

$$0 < \exists \beta < 1,\ \forall \{X, Y\} \in \mathcal{E}',\ \mathrm{E}[d(X', Y')] \leq \beta d(X, Y).$$

Then the mixing time $\tau(\varepsilon)$ of Markov chain MC satisfies

$$\tau(\varepsilon) \leq (1 - \beta)^{-1} \ln(D/(L\varepsilon))$$

where

$$D \overset{\text{def.}}{=} \max_{(x,y) \in \varOmega'^2} d(x, y) \quad \textit{and} \quad L \overset{\text{def.}}{=} \min\{d(x, y) \mid (x, y) \in \varOmega'^2, d(x, y) > 0\}.$$

The mixing rate satisfies that $\tau \leq (1 - \beta)^{-1}(1 + \ln(D/L))$.

The above theorem differs from the original theorem in [1] since the integrality of the edge length is not assumed. We drop the integrality and introduced the minimum distance L. This modification is not essential and we can show Theorem 1 similarly.

2.3 Coupling From the Past

In the rest of this section, we briefly review the coupling from the past (CFTP) algorithm. When we simulate an ergodic Markov chain for infinite time, we can gain a sample exactly according to the stationary distribution. Suppose that there exists a chain from infinite past, then a possible state at the present time of the chain for which we can have an evidence of the uniqueness without respect to an initial state of the chain, is a realization of a random sample exactly from the stationary distribution. This is the key idea of CFTP.

Suppose that we have an ergodic Markov chain MC with a finite state space Ω' and a transition matrix P. The transition rule of the Markov chain $X \mapsto X'$ can be described by a deterministic function $\phi : \Omega' \times [0,1) \to \Omega'$, called an *update function*, as follows. Given a random number λ uniformly distributed over $[0,1)$, the update function ϕ satisfies that $\Pr[\phi(x,\lambda) = y] = P(x,y)$ for any $x, y \in \Omega'$. We can realize the Markov chain by setting $X' = \phi(X,\lambda)$. Clearly, the update function corresponding to the given transition matrix P is not unique. The result of transitions of the chain from the time t_1 to t_2 ($t_1 < t_2$) with a sequence of random numbers $\boldsymbol{\lambda} = (\lambda[t_1], \lambda[t_1+1], \ldots, \lambda[t_2-1]) \in [0,1)^{t_2-t_1}$ is denoted by $\Phi_{t_1}^{t_2}(x,\boldsymbol{\lambda}) : \Omega' \times [0,1)^{t_2-t_1} \to \Omega'$ where

$$\Phi_{t_1}^{t_2}(x,\boldsymbol{\lambda}) \stackrel{\text{def.}}{=} \phi(\phi(\cdots(\phi(x,\lambda[t_1]),\ldots,\lambda[t_2-2]),\lambda[t_2-1]).$$

Given a negative integer T, we say that a sequence $\boldsymbol{\lambda} \in [0,1)^{|T|}$ satisfies the *coalescence condition*, when $\exists y \in \Omega'$, $\forall x \in \Omega'$, $y = \Phi_T^0(x,\boldsymbol{\lambda})$.

With these preparation, the standard CFTP algorithm is expressed as follows.

CFTP Algorithm [20]

```
Step 1. Set the starting time period $T := -1$ to go back, and set $\boldsymbol{\lambda}$ be the
    empty sequence.
Step 2. Generate random real numbers $\lambda[T], \lambda[T+1], \ldots, \lambda[\lceil T/2 \rceil - 1] \in [0,1)$,
    and insert them to the head of $\boldsymbol{\lambda}$ in order, i.e., put
        $\boldsymbol{\lambda} := (\lambda[T], \lambda[T+1], \ldots, \lambda[-1])$.
Step 3. Start a chain from each element $x \in \Omega'$ at time period $T$, and run each
    chain to time period 0 according to the update function $\phi$ with the sequence
    of numbers in $\boldsymbol{\lambda}$. (Note that every chain uses the common sequence $\boldsymbol{\lambda}$.)
Step 4. [Coalescence check]
    (a) If $\exists y \in \Omega'$, $\forall x \in \Omega'$, $y = \Phi_T^0(x,\boldsymbol{\lambda})$, then return $y$ and stop.
    (b) Else, update the starting time period $T := 2T$, and go to Step 2.
```

Theorem 2. (CFTP Theorem [20]) *Let MC be an ergodic finite Markov chain with a finite state space Ω', defined by an update function $\phi : \Omega' \times [0,1) \to \Omega'$. If the CFTP algorithm terminates with probability 1, then the obtained value is a realization of a random variable exactly distributed according to the stationary distribution.*

Theorem 2 gives a (probabilistically) finite time algorithm for infinite time simulation. However, simulations from all states executed in Step 3 is a hard requirement.

Suppose that there exists a partial order "$\succeq$" on the set of states Ω'. An update function ϕ is called *monotone* (with respect to "$\succeq$") if $\forall\lambda \in [0,1)$, $\forall x, \forall y \in \Omega'$, $x \succeq y \Rightarrow \phi(x,\lambda) \succeq \phi(y,\lambda)$. For ease, we also say that a chain is *monotone* if the chain has a *monotone* transition rule.

Theorem 3. (monotone CFTP [20]) *Suppose that a Markov chain defined by an update function ϕ is monotone with respect to a partially ordered set of states $(\Omega', \succeq)$, and $\exists x_{\max}, \exists x_{\min} \in \Omega'$, $\forall x \in \Omega'$, $x_{\max} \succeq x \succeq x_{\min}$. Then the CFTP algorithm terminates with probability 1, and a sequence $\boldsymbol{\lambda} \in [0,1)^{|T|}$ satisfies the coalescence condition, i.e., $\exists y \in \Omega'$, $\forall x \in \Omega'$, $y = \Phi_T^0(x, \boldsymbol{\lambda})$, if and only if $\Phi_T^0(x_{\max}, \boldsymbol{\lambda}) = \Phi_T^0(x_{\min}, \boldsymbol{\lambda})$.*

When a given Markov chain satisfies the conditions of Theorem 3, we can modify CFTP algorithm by substituting Step 4 (a) by

Step 4. (a)$'$ If $\exists y \in \Omega'$, $y = \Phi_T^0(x_{\max}, \boldsymbol{\lambda}) = \Phi_T^0(x_{\min}, \boldsymbol{\lambda})$, then return y and stop.

The algorithm obtained by the above modification is called a monotone CFTP algorithm.

3 Sampling Algorithms and Main Results

In this section, we propose two Markov chains and one perfect sampling algorithm. We also describe main results of this paper.

First, we introduce some notations. For any integer $b \geq 2$, we introduce a set of 2-dimensional integer vectors

$$\Omega(b) \overset{\text{def.}}{=} \{(Y_1, Y_2) \in \mathrm{Z}^2 \mid Y_1, Y_2 > 0,\ Y_1 + Y_2 = b\}$$

and a distribution function $f_b(Y_1, Y_2 \mid u_i, u_j) : \Omega(b) \to [0,1]$ with non-negative parameters u_i, u_j defined by

$$f_b(Y_1, Y_2 \mid u_i, u_j) \overset{\text{def.}}{=} C(u_i, u_j, b) Y_1^{u_i - 1} Y_2^{u_j - 1}$$

where

$$(C(u_i, u_j, b))^{-1} \overset{\text{def.}}{=} \sum_{(Y_1, Y_2) \in \Omega(b)} Y_1^{u_i-1} Y_2^{u_j-1} = \sum_{l=1}^{b-1} l^{u_i-1}(b-l)^{u_j-1}$$

is the partition function. We introduce $g_b(0|u_i,u_j), g_b(1|u_i,u_j), \ldots, g_b(b-1|u_i,u_j)$ defined by

$$g_b(k|u_i,u_j) \overset{\text{def.}}{=} \begin{cases} 0 & (k=0), \\ \sum_{y_1=1}^{k} f_b(y_1, b-y_1 \mid u_i, u_j) & (k \in \{1,2,\ldots,b-1\}). \end{cases}$$

It is clear that $0 = g_b(0|u_i,u_j) < g_b(1|u_i,u_j) < \cdots < g_b(b-1|u_i,u_j) = 1$ and

$$g_b(k|u_i,u_j) = \begin{cases} 0 & (k=0), \\ \sum_{l=1}^{k} C(u_i,u_j,b) l^{u_i-1}(b-l)^{u_j-1} & (k \in \{1,2,\ldots,b-1\}). \end{cases}$$

3.1 Markov Chain for Approximate Sampler

We describe our first chain $\mathcal{M}^{\mathrm{A}}$ with state space Ω (the discretized simplex). Given a current state $X \in \Omega$, we pick an ordered pair of mutually distinct indices $(i,j) \in \{1,2,\ldots,n\} \times \{1,2,\ldots,n\}$ uniformly at random and generate a random number $\lambda \in [0,1)$. Then transition $X \mapsto X'$ with respect to (i,j) and λ takes place as follows.

<u>**Markov chain $\mathcal{M}^{\mathrm{A}}$**</u>
Input: A current state $X \in \Omega$, a randomly chosen ordered pair of mutually distinct indices $(i,j) \in \{1,2,\ldots,n\}^2$ and a random real number $\lambda \in [0,1)$.
Step 1: Put $b = X_i + X_j$.
Step 2: Let $k \in \{1,2,\ldots,b-1\}$ be a unique value satisfying

$$g_b(k-1|u_i,u_j) \le \lambda < g_b(k|u_i,u_j).$$

Step 3: Put $X'_\ell := \begin{cases} k & (\ell = i), \\ b-k & (\ell = j), \\ X_\ell & (\text{otherwise}). \end{cases}$

Given a state $X \in \Omega$, an ordered pair of mutually distinct indices (i,j), and a real number $\lambda \in [0,1)$, we denote the state X' determined by the above procedure by $\psi(X,(i,j),\lambda) \stackrel{\text{def.}}{=} X'$. Clearly, this chain is irreducible and aperiodic. Since the detailed balance equations hold, the stationary distribution of chain $\mathcal{M}^{\mathrm{A}}$ is the discretized Dirichlet distribution.

We will show the following theorem, which gives an upper bound of the mixing time of our first chain $\mathcal{M}^{\mathrm{A}}$.

Theorem 4 *The mixing time $\tau^{\mathrm{A}}(\varepsilon)$ of the Markov chain $\mathcal{M}^{\mathrm{A}}$ satisfies*

$$\tau^{\mathrm{A}}(\varepsilon) \le (1/2)n(n-1)\ln((\Delta - n)\varepsilon^{-1}).$$

This theorem appears in [16] with an outline of the proof. In Section 4.1, we refine the proof in [16] and go into detail.

3.2 Perfect Sampling Algorithm

We describe our second chain $\mathcal{M}^{\mathrm{P}}$ for our perfect sampling algorithm. Given a current state $X \in \Omega$, we generate a random number $\lambda \in [1,n)$. Then transition $X \mapsto X'$ with respect to λ takes place as follows.

<u>**Markov chain $\mathcal{M}^{\mathrm{P}}$**</u>
Input: A current state $X \in \Omega$ and a random number $\lambda \in [1,n)$.
Step 1: Put $i := \lfloor \lambda \rfloor$ and $b := X_i + X_{i+1}$.
Step 2: Let $k \in \{1,2,\ldots,b-1\}$ be a unique value satisfying

$$g_b(k-1|u_i,u_{i+1}) \le (\lambda - \lfloor \lambda \rfloor) < g_b(k|u_i,u_{i+1}).$$

Step 3: Put $X'_\ell := \begin{cases} k & (\ell = i), \\ b-k & (\ell = i+1), \\ X_\ell & (\text{otherwise}). \end{cases}$

The update function $\phi : \Omega \times [1, n) \to \Omega$ of our chain is defined by $\phi(X, \lambda) \overset{\text{def.}}{=} X'$ where X' is determined by the above procedure. Clearly, this chain is irreducible and aperiodic. Since the detailed balance equations hold, the stationary distribution of chain $\mathcal{M}^{\mathrm{P}}$ is the discretized Dirichlet distribution.

When we discuss the mixing time of chain $\mathcal{M}^{\mathrm{P}}$, we assume the following condition.

Condition 1 *Parameters are arranged in non-increasing order, i.e.,*

$$u_1 \geq u_2 \geq \cdots \geq u_n.$$

We can assume Condition 1 by sorting parameters in $\mathrm{O}(n \ln n)$ time. Then we have the following result.

Theorem 5. *Under Condition 1, the mixing time $\tau^{\mathrm{P}}(\varepsilon)$ of $\mathcal{M}^{\mathrm{P}}$ satisfies*

$$\tau^{\mathrm{P}}(\varepsilon) \leq n(n-1)^2 \ln(n(\Delta - n)/(2\varepsilon)).$$

Next, we propose a perfect sampling algorithm based on the monotone CFTP algorithm. We introduce a specified pair of states $X_{\mathrm{U}}, X_{\mathrm{L}} \in \Omega$ defined by

$$X_{\mathrm{U}} \overset{\text{def.}}{=} (\Delta - n + 1, 1, 1, \ldots, 1), \quad X_{\mathrm{L}} \overset{\text{def.}}{=} (1, 1, \ldots, 1, \Delta - n + 1).$$

```
Algorithm 1
Step 1. Set the starting time period $T := -1$ to go back, and $\boldsymbol{\lambda}$ be the empty
    sequence.
Step 2. Generate random real numbers $\lambda[T], \lambda[T+1], \ldots, \lambda[\lceil T/2 \rceil - 1] \in [1, n)$
    and put $\boldsymbol{\lambda} = (\lambda[T], \lambda[T-1], \ldots, \lambda[-1])$.
Step 3. Start two chains from $X_{\mathrm{U}}$ and $X_{\mathrm{L}}$, respectively at time period $T$, and
    run them to time period 0 according to the update function $\phi$ with the
    sequence of numbers in $\boldsymbol{\lambda}$.
Step 4. [Coalescence check]
    (a) If $\exists Y \in \Omega$, $Y = \Phi_T^0(X_{\mathrm{U}}, \boldsymbol{\lambda}) = \Phi_T^0(X_{\mathrm{L}}, \boldsymbol{\lambda})$, then return $Y$ and stop.
    (b) Else, update the starting time period $T := 2T$, and go to Step 2.
```

The function $\Phi_{t_1}^{t_2}(x, \boldsymbol{\lambda})$, which has appeared in Section 2, is defined by

$$\Phi_{t_1}^{t_2}(x, \boldsymbol{\lambda}) \overset{\text{def.}}{=} \phi(\phi(\cdots(\phi(x, \lambda[t_1]), \ldots, \lambda[t_2 - 2]), \lambda[t_2 - 1]).$$

Here we note that we could also employ Wilson's read once algorithm [24] and Fill's interruptible algorithm [9, 10], each of which also gives a perfect sampler. We have the following results on the above sampling algorithm.

Theorem 6. *With probability 1, Algorithm 1 terminates and returns a state. The state obtained by Algorithm 1 is a realization of a sample exactly according to the discretized Dirichlet distribution.*

Theorem 7. *Under Condition 1, the expected number of transitions executed in Algorithm 1 is bounded by $\mathrm{O}(n^3 \ln \Delta)$.*

4 Analysis of Mixing Times

In this section, we introduce two joint processes and analyze the mixing times of our chains by using path coupling method. First, we introduce an undirected graph $G = (\Omega, \mathcal{E})$ with a vertex set Ω and an edge set $\mathcal{E}$ defined as follows. A pair of vertices (states) $\{X, Y\}$ is an edge of G if and only if $(1/2)\sum_{i=1}^{n} |X_i - Y_i| = 1$. Clearly, the graph G is connected. For each edge $e = \{X, Y\} \in \mathcal{E}$, there exists a unique pair of indices $j_1, j_2 \in \{1, \ldots, n\}$, called the *supporting pair* of e, satisfying

$$|X_i - Y_i| = \begin{cases} 1 \ (i \in \{j_1, j_2\}), \\ 0 \ (\text{otherwise}). \end{cases}$$

In the following subsections, we introduce two edge length functions.

4.1 Mixing Time of Markov Chain for Approximate Sampler

To estimate the mixing time of our first chain $\mathcal{M}^{\mathrm{A}}$, we put a weight of every edge in the graph $G = (\Omega, \mathcal{E})$ to 1. Then the distance between a pair of states $X, Y \in \Omega$ denoted by $d^{\mathrm{A}}(X, Y)$, is equal to the length of a shortest path on G from X to Y where the length of a path is equal to the number of edges contained in the path. For any state $X \in \Omega$, we define $d^{\mathrm{A}}(X, X) = 0$. It is clear that the *diameter* of graph G, the distance between a farthest pair of vertices, is equal to $\Delta - n$.

We define our first joint process by $(X, Y) \mapsto (\psi(X, (i, j), \lambda), \psi(Y, (i, j), \lambda))$ with a randomly chosen ordered pair of mutually distinct indices (i, j) and a uniform real random number $\lambda \in [0, 1)$, where ψ is the function defined by the chain $\mathcal{M}^{\mathrm{A}}$ in the previous section.

Proof of Theorem 4: For any pair of states $\{X, Y\} \in \mathcal{E}$, we estimate the expected distance between X' and Y' obtained by the above joint process i.e., $(X', Y') = (\psi(X, (i, j), \lambda), \psi(Y, (i, j), \lambda))$.

Clearly from the definition of our chain $\mathcal{M}^{\mathrm{A}}$, X'_i is a unique value k' satisfying

$$g_{b'}(k' - 1 | u_i, u_j) \leq \lambda < g_{b'}(k' | u_i, u_j)$$

where $b' \overset{\text{def.}}{=} X_i + X_j$. Similarly, Y'_i is a unique value k'' satisfying

$$g_{b''}(k'' - 1 | u_i, u_j) \leq \lambda < g_{b''}(k'' | u_i, u_j)$$

where $b'' \overset{\text{def.}}{=} Y_i + Y_j$. We need to consider following four cases.

Case 1: If the pair of indices $\{i, j\}$ is disjoint with the supporting pair of $\{X, Y\}$, then $b' = b''$ and thus we have $X'_i = k' = k'' = Y'_j$. It directly implies that $\{X', Y'\} \in \mathcal{E}$ and $d^{\mathrm{A}}(X', Y') = d^{\mathrm{A}}(X, Y) = 1$.

Case 2: If the pair $\{i, j\}$ is the supporting pair of $\{X, Y\}$, then $b' = b''$ and thus we have $X'_i = k' = k'' = Y'_j$. It directly implies that $X' = Y'$ and $d^{\mathrm{A}}(X', Y') = 0$.

Case 3: Consider the case that the ordered pair of indices (i, j) satisfies that i is in the supporting pair of $\{X, Y\}$ and j is not. Without loss of generality, we can assume that $X_i = Y_i + 1$ and thus we have that $X_i + X_j = b' = b'' + 1 = Y_i + Y_j + 1$ since $X_j = Y_j$.

Lemma 4 in Appendix section implies the following inequalities,

$$\begin{aligned}0 = g_{b''+1}(0|u_i, u_j) &= g_{b''}(0|u_i, u_j) \leq g_{b''+1}(1|u_i, u_j) \leq g_{b''}(1|u_i, u_j) \leq \cdots \\ &\leq g_{b''+1}(k-1|u_i, u_j) \leq g_{b''}(k-1|u_i, u_j) \leq g_{b''+1}(k|u_i, u_j) \leq \cdots \\ &\leq g_{b''+1}(b''-1|u_i, u_j) \leq g_{b''}(b''-1|u_i, u_j) = g_{b''+1}(b''|u_i, u_j) = 1,\end{aligned}$$

which we will call *alternating inequalities.* For example, if inequalities

$$g_{b''+1}(k-1|u_i, u_j) \leq \lambda < g_{b''}(k-1|u_i, u_j) \leq g_{b''+1}(k|u_i, u_j)$$

hold, then $X'_i = k > k - 1 = Y'_i$. And if

$$g_{b''+1}(k-1|u_i, u_j) \leq g_{b''}(k-1|u_i, u_j) \leq \lambda < g_{b''+1}(k|u_i, u_j)$$

hold, then $X'_i = k = Y'_i$. Thus we have

$$\begin{pmatrix} X'_i \\ Y'_i \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \dots, \begin{pmatrix} b''-1 \\ b''-1 \end{pmatrix}, \begin{pmatrix} b'' \\ b''-1 \end{pmatrix} \right\}$$

(see Figure 2). Since $X'_\ell = X_\ell$ and $Y'_\ell = Y_\ell$ for all $\ell \in \{1, 2, \dots, n\} \setminus \{i, j\}$, and $X'_i + X'_j = b' = b'' + 1 = Y'_i + Y'_j + 1$, we have that $\{X', Y'\} \in \mathcal{E}$ and thus $d^{\mathrm{A}}(X', Y') = d^{\mathrm{A}}(X, Y) = 1$.

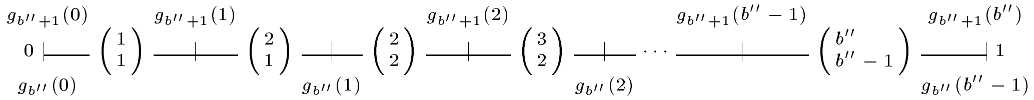

$g_{b''+1}(0)$
0
$g_{b''}(0)$
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$
$g_{b''+1}(1)$
$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$
$g_{b''}(1)$
$\begin{pmatrix} 2 \\ 2 \end{pmatrix}$
$g_{b''+1}(2)$
$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$
$g_{b''}(2)$
$\cdots$
$g_{b''+1}(b''-1)$
$\begin{pmatrix} b'' \\ b''-1 \end{pmatrix}$
$g_{b''+1}(b'')$
1
$g_{b''}(b''-1)$

Fig. 2. A figure of alternating inequalities. In the above, we denote $g_{b''}(k|u_i, u_j)$ and $g_{b''+1}(k|u_i, u_j)$ by $g_{b''}(k), g_{b''+1}(k)$, respectively.

Case 4: Consider the last case that the ordered pair of indices (i, j) satisfies that j is in the supporting pair of $\{X, Y\}$ and i is not. We can show that $d^{\mathrm{A}}(X', Y') = 1$ in a similar way to that of Case 3.

In Cases 1, 3 and 4, the distance between X' and Y' is equal to 1. When Case 2 occurred, the distance between X' and Y' decreases to 0. Since the probability of the event that Case 2 is selected is equal to $2/(n(n-1))$, the expectation of the distance $\mathrm{E}[d(X', Y')]$ becomes to $1 - 2/(n(n-1))$. Theorem 1 (Path Coupling Theorem), briefly reviewed in Section 2, shows that the mixing time $\tau^{\mathrm{A}}(\varepsilon)$ satisfies $\tau^{\mathrm{A}}(\varepsilon) \leq (1/2)n(n-1)\ln((\Delta - n)\varepsilon^{-1})$. □

4.2 Mixing Time of Markov Chain for Perfect Sampler

We define the length $l^{\mathrm{P}}(e)$ of an edge $e = \{X, Y\} \in \mathcal{E}$ by

$$l^{\mathrm{P}}(e) \overset{\text{def.}}{=} (1/(n-1)) \textstyle\sum_{i=1}^{j^*-1} (n-i)$$

where $j^* = \max\{j_1, j_2\} \geq 2$ and $\{j_1, j_2\}$ is the supporting pair of $e = \{X, Y\}$. Note that $1 \leq \min_{e \in \mathcal{E}} l^{\mathrm{P}}(e) \leq \max_{e \in \mathcal{E}} l^{\mathrm{P}}(e) \leq n/2$. For each pair $X, Y \in \Omega$, we define the distance $d^{\mathrm{P}}(X, Y)$ by the length of a shortest path between X and Y on G with respect to the edge length function l^{P}. Clearly, the diameter of G, i.e., $\max_{(X,Y) \in \Omega^2} d^{\mathrm{P}}(X, Y)$, is bounded by $n(\Delta - n)/2$, since $d^{\mathrm{P}}(X, Y) \leq (n/2) \sum_{i=1}^{n} (1/2)|X_i - Y_i| \leq (n/2)(\Delta - n)$ for any $(X, Y) \in \Omega^2$.

We define our second joint process by $(X, Y) \mapsto (\phi(X, \lambda), \phi(Y, \lambda))$ with uniformly random real number $\lambda \in [1, n)$, where ϕ is the update function defined by the chain $\mathcal{M}^{\mathrm{P}}$ in Section 3.

<u>Proof of Theorem 5:</u> For any pair of states $\{X, Y\} \in \mathcal{E}$, we estimate the expectation of the distance between X' and Y' obtained by the above joint process i.e., $(X', Y') = (\phi(X, \lambda), \phi(Y, \lambda))$ where $\lambda \in [1, n)$ is a uniformly random real number. More precisely, we show that

$$\mathrm{E}[d^{\mathrm{P}}(X', Y')] \leq (1 - 1/(n(n-1)^2)) d^{\mathrm{P}}(X, Y). \tag{1}$$

In the following, we denote the supporting pair of $\{X, Y\}$ by $\{j_1, j_2\}$. Without loss of generality, we can assume that $j_1 < j_2$, and $X_{j_2} + 1 = Y_{j_2}$.

Case 1: When $\lfloor \lambda \rfloor = j_2 - 1$, we will show that

$$\mathrm{E}[d^{\mathrm{P}}(X', Y') | \lfloor \lambda \rfloor = j_2 - 1] \leq d^{\mathrm{P}}(X, Y) - (1/2)(n - j_2 + 1)/(n-1).$$

In case $j_1 = j_2 - 1$, we have $X' = Y'$ with conditional probability 1. Hence $d^{\mathrm{P}}(X', Y') = 0$. In the following, we consider the case $j_1 < j_2 - 1$. Put $b' = X_{j_2-1} + X_{j_2}$ and $b'' = Y_{j_2-1} + Y_{j_2}$. Since $X_{j_2} + 1 = Y_{j_2}$, $b' + 1 = b''$ holds. From the definition of the update function of our Markov chain, we have followings,

$$\begin{aligned} X'_{j_2-1} = k &\Leftrightarrow [g_{b'}(k-1|u_{j_2-1}, u_{j_2}) \leq \lambda - \lfloor \lambda \rfloor < g_{b'}(k|u_{j_2-1}, u_{j_2})] \\ Y'_{j_2-1} = k &\Leftrightarrow [g_{b'+1}(k-1|u_{j_2-1}, u_{j_2}) \leq \lambda - \lfloor \lambda \rfloor < g_{b'+1}(k|u_{j_2-1}, u_{j_2})]. \end{aligned}$$

Lemma 4 in Appendix section implies the following inequalities,

$$\begin{aligned} 0 &= g_{b'+1}(0|u_{j_2-1}, u_{j_2}) = g_{b'}(0|u_{j_2-1}, u_{j_2}) \\ &\leq g_{b'+1}(1|u_{j_2-1}, u_{j_2}) \leq g_{b'}(1|u_{j_2-1}, u_{j_2}) \leq \cdots \\ &\leq g_{b'+1}(b'-1|u_{j_2-1}, u_{j_2}) \leq g_{b'}(b'-1|u_{j_2-1}, u_{j_2}) = g_{b'+1}(b'|u_{j_2-1}, u_{j_2}) = 1. \end{aligned}$$

Thus we have

$$\begin{pmatrix} X'_{j_2-1} \\ Y'_{j_2-1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \ldots, \begin{pmatrix} b'-1 \\ b'-1 \end{pmatrix}, \begin{pmatrix} b'-1 \\ b' \end{pmatrix} \right\}$$

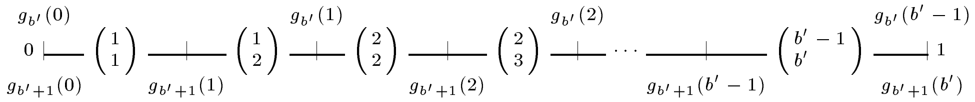


Fig. 3. A figure of alternating inequalities. In the above, we denote $g_{b'}(k|u_{j_2-1}, u_{j_2})$ and $g_{b'+1}(k|u_{j_2-1}, u_{j_2})$ by $g_{b'}(k), g_{b'+1}(k)$, respectively.

(see Figure 3). If $X'_{j_2-1} = Y'_{j_2-1}$, the supporting pair of $\{X', Y'\}$ is $\{j_1, j_2\}$ and thus $d^{\mathrm{P}}(X', Y') = d^{\mathrm{P}}(X, Y)$. If $X'_{j_2-1} \neq Y'_{j_2-1}$, the supporting pair of $\{X', Y'\}$ is $\{j_1, j_2 - 1\}$ and so $d^{\mathrm{P}}(X', Y') = d^{\mathrm{P}}(X, Y) - (n - j_2 + 1)/(n - 1)$.

Lemma 5 in Appendix section shows that the condition $u_{j_2-1} \geq u_{j_2}$ implies the inequality

$$\begin{aligned}
&\Pr[X'_{j_2-1} \neq Y'_{j_2-1} | \lfloor\lambda\rfloor = j_2 - 1] - \Pr[X'_{j_2-1} = Y'_{j_2-1} | \lfloor\lambda\rfloor = j_2 - 1] \\
&\quad = \textstyle\sum_{k=1}^{b'-1} [g_{b'}(k|u_{j_2-1}, u_{j_2}) - g_{b'+1}(k|u_{j_2-1}, u_{j_2})] \\
&\quad\quad - \textstyle\sum_{k=1}^{b'-1} [g_{b'+1}(k|u_{j_2-1}, u_{j_2}) - g_{b'}(k-1|u_{j_2-1}, u_{j_2})] \geq 0
\end{aligned}$$

and thus we have

$$\begin{aligned}
\Pr[X'_{j_2-1} = Y'_{j_2-1} | \lfloor\lambda\rfloor = j_2 - 1] \leq (1/2), \\
\Pr[X'_{j_2-1} \neq Y'_{j_2-1} | \lfloor\lambda\rfloor = j_2 - 1] \geq (1/2).
\end{aligned}$$

We need Condition 1 to show Lemma 5. From the above, we obtain that

$$\begin{aligned}
\mathrm{E}[d^{\mathrm{P}}(X', Y') | \lfloor\lambda\rfloor = j_2 - 1] &\leq (1/2) d^{\mathrm{P}}(X, Y) + (1/2)(d^{\mathrm{P}}(X, Y) - (n - j_2 + 1)/(n - 1)) \\
&= d^{\mathrm{P}}(X, Y) - (1/2)(n - j_2 + 1)/(n - 1).
\end{aligned}$$

Case 2: When $\lfloor\lambda\rfloor = j_2$, we can show that $\mathrm{E}[d^{\mathrm{P}}(X', Y') | \lfloor\lambda\rfloor = j_2] \leq d^{\mathrm{P}}(X, Y) + (1/2)(n - j_2)/(n - 1)$ in a similar way with Case 1.
Case 3: When $\lfloor\lambda\rfloor \neq j_2 - 1$ and $\lfloor\lambda\rfloor \neq j_2$, it is easy to see that the supporting pair $\{j'_1, j'_2\}$ of $\{X', Y'\}$ satisfies $j_2 = \max\{j'_1, j'_2\}$. Thus $d^{\mathrm{P}}(X, Y) = d^{\mathrm{P}}(X', Y')$.

The probability of appearance of Case 1 is equal to $1/(n - 1)$, and that of Case 2 is less than or equal to $1/(n - 1)$. From the above,

$$\begin{aligned}
&\mathrm{E}[d^{\mathrm{P}}(X', Y')] \leq d^{\mathrm{P}}(X, Y) - \frac{1}{n-1} \frac{1}{2} \frac{n - j_2 + 1}{n - 1} + \frac{1}{n-1} \frac{1}{2} \frac{n - j_2}{n - 1} \\
&= d^{\mathrm{P}}(X, Y) - \frac{1}{2(n-1)^2} \leq \left(1 - \frac{1}{2(n-1)^2} \frac{1}{\max_{\{X,Y\} \in \mathcal{E}} \{d^{\mathrm{P}}(X, Y)\}}\right) d^{\mathrm{P}}(X, Y) \\
&= \left(1 - \frac{1}{n(n-1)^2}\right) d^{\mathrm{P}}(X, Y).
\end{aligned}$$

Since the diameter of G is bounded by $n(\varDelta - n)/2$, Theorem 1 implies that the mixing time $\tau^{\mathrm{P}}(\varepsilon)$ satisfies $\tau^{\mathrm{P}}(\varepsilon) \leq n(n - 1)^2 \ln(n(\varDelta - n)/(2\varepsilon))$. □

5 Analysis of Perfect Sampling Algorithm

In this section, we discuss the correctness and the time complexity of Algorithm 1, and prove Theorems 6 and 7.

5.1 Correctness

In Subsection 2.3, we described monotone CFTP algorithm and Theorem 3 (theorem of monotone CFTP). Thus we only need to show the monotonicity of our chain $\mathcal{M}^{\mathrm{P}}$ defined by the update function ϕ.

First, we introduce a partial order on the state space Ω. For any vector $X \in \Omega$, we define the *cumulative sum vector* $\mathrm{c}_X \in \mathrm{Z}_+^{n+1}$ by

$$\mathrm{c}_X(i) \overset{\text{def.}}{=} \begin{cases} 0 & (i = 0), \\ X_1 + X_2 + \cdots + X_i & (i \in \{1, 2, \ldots, n\}), \end{cases}$$

where $\mathrm{c}_X = (\mathrm{c}_X(0), \mathrm{c}_X(1), \ldots, \mathrm{c}_X(n))$. Clearly, there exists a bijection between Ω and the set $\{\mathrm{c}_X \mid X \in \Omega\}$. For any pair of states $X, Y \in \Omega$, we say $X \succeq Y$ if and only if $\mathrm{c}_X \geq \mathrm{c}_Y$. It is clear that the relation "$\succeq$" is a partial order on Ω. We can see easily that $\forall X \in \Omega$, $X_{\mathrm{U}} \succeq X \succeq X_{\mathrm{L}}$.

We say that a state $X \in \Omega$ *covers* $Y \in \Omega$ (at j), denoted by $X \cdot\!\!\succ Y$ (or $X \cdot\!\!\succ_j Y$), when

$$\mathrm{c}_X(i) - \mathrm{c}_Y(i) = \begin{cases} 1 & (i = j), \\ 0 & (\text{otherwise}). \end{cases}$$

Note that $X \cdot\!\!\succ_j Y$ if and only if

$$X_i - Y_i = \begin{cases} +1 & (i = j), \\ -1 & (i = j + 1), \\ 0 & (\text{otherwise}). \end{cases}$$

Next, we show a key lemma for proving monotonicity.

Lemma 1. *$\forall X, \forall Y \in \Omega$, $\forall \lambda \in [1, n)$, $X \cdot\!\!\succ Y \Rightarrow \phi(X, \lambda) \succeq \phi(Y, \lambda)$.*

Proof: In the following proof, we assume that the index j satisfies that $X \cdot\!\!\succ_j Y$. We denote $\phi(X, \lambda)$ by X' and $\phi(Y, \lambda)$ by Y' for simplicity. For any index $i \neq \lfloor \lambda \rfloor$, it is easy to see that $\mathrm{c}_X(i) = \mathrm{c}_{X'}(i)$ and $\mathrm{c}_Y(i) = \mathrm{c}_{Y'}(i)$, and so $\mathrm{c}_{X'}(i) - \mathrm{c}_{Y'}(i) = \mathrm{c}_X(i) - \mathrm{c}_Y(i) \geq 0$ since $X \succeq Y$. In the following, we show that $\mathrm{c}_{X'}(\lfloor \lambda \rfloor) \geq \mathrm{c}_{Y'}(\lfloor \lambda \rfloor)$.

Clearly from the definition of our chain $\mathcal{M}^{\mathrm{P}}$, $X'_{\lfloor \lambda \rfloor}$ is a unique value k' satisfying

$$g_{b'}(k' - 1 | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1}) \leq (\lambda - \lfloor \lambda \rfloor) < g_{b'}(k' | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1})$$

where $b' \overset{\text{def.}}{=} X_{\lfloor \lambda \rfloor} + X_{\lfloor \lambda \rfloor + 1}$. Similarly, $Y'_{\lfloor \lambda \rfloor}$ is a unique value k'' satisfying

$$g_{b''}(k'' - 1 | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1}) \leq (\lambda - \lfloor \lambda \rfloor) < g_{b''}(k'' | u_{\lfloor \lambda \rfloor}, u_{\lfloor \lambda \rfloor + 1})$$

where $b'' \overset{\text{def.}}{=} Y_{\lfloor\lambda\rfloor} + Y_{\lfloor\lambda\rfloor+1}$. We need to consider following three cases.

Case 1: Consider the case that $\lfloor\lambda\rfloor \neq j-1$ and $\lfloor\lambda\rfloor \neq j+1$, where j is the index satisfying $X \succ_j Y$. Then $b' = b''$ holds and thus we have $X'_{\lfloor\lambda\rfloor} = k' = k'' = Y'_{\lfloor\lambda\rfloor}$ which implies $\mathrm{c}_{X'}(\lfloor\lambda\rfloor) \geq \mathrm{c}_{Y'}(\lfloor\lambda\rfloor)$.

Case 2: Consider the case that $\lfloor\lambda\rfloor = j-1$. Since $X \succ_j Y$, we have $b' = b'' + 1$. From the definition of cumulative sum vector,

$$\begin{aligned}
&\mathrm{c}_{X'}(\lfloor\lambda\rfloor) - \mathrm{c}_{Y'}(\lfloor\lambda\rfloor) = \mathrm{c}_{X'}(j-1) - \mathrm{c}_{Y'}(j-1) \\
&= \mathrm{c}_{X'}(j-2) + X'_{j-1} - \mathrm{c}_{Y'}(j-2) - Y'_{j-1} \\
&= \mathrm{c}_{X}(j-2) + X'_{j-1} - \mathrm{c}_{Y}(j-2) - Y'_{j-1} = X'_{j-1} - Y'_{j-1}.
\end{aligned}$$

Thus, it is enough to show that $X'_{j-1} \geq Y'_{j-1}$.

Lemma 4 in Appendix section implies the following alternating inequalities

$$\begin{aligned}
0 &= g_{b''+1}(0|u_{j-1},u_j) = g_{b''}(0|u_{j-1},u_j) \leq g_{b''+1}(1|u_{j-1},u_j) \leq g_{b''}(1|u_{j-1},u_j) \leq \cdots \\
&\leq g_{b''+1}(k-1|u_{j-1},u_j) \leq g_{b''}(k-1|u_{j-1},u_j) \leq g_{b''+1}(k|u_{j-1},u_j) \leq \cdots \\
&\leq g_{b''+1}(b''-1|u_{j-1},u_j) \leq g_{b''}(b''-1|u_{j-1},u_j) = g_{b''+1}(b''|u_{j-1},u_j) = 1.
\end{aligned}$$

If inequalities

$$g_{b''+1}(k-1|u_{j-1},u_j) \leq (\lambda - \lfloor\lambda\rfloor) < g_{b''}(k-1|u_{j-1},u_j) \leq g_{b''+1}(k|u_{j-1},u_j)$$

hold, then $X'_{\lfloor\lambda\rfloor} = k > k-1 = Y'_{\lfloor\lambda\rfloor}$. And if

$$g_{b''+1}(k-1|u_{j-1},u_j) \leq g_{b''}(k-1|u_{j-1},u_j) \leq (\lambda - \lfloor\lambda\rfloor) < g_{b''+1}(k|u_{j-1},u_j)$$

hold, then $X'_{\lfloor\lambda\rfloor} = k = Y'_{\lfloor\lambda\rfloor}$. Thus we have

$$\begin{pmatrix} X'_{j-1} \\ Y'_{j-1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \ldots, \begin{pmatrix} b''-1 \\ b''-1 \end{pmatrix}, \begin{pmatrix} b'' \\ b''-1 \end{pmatrix} \right\}$$

(see Figure 4). From the above, we have that $X'_{j-1} \geq Y'_{j-1}$.

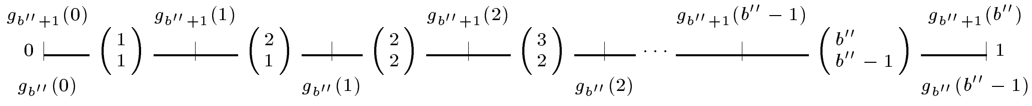


Fig. 4. A figure of alternating inequalities. In the above, we denote $g_{b''}(k|u_{j-1},u_j)$ and $g_{b''+1}(k|u_{j-1},u_j)$ by $g_{b''}(k), g_{b''+1}(k)$, respectively.

Case 3: Consider the case that $\lfloor\lambda\rfloor = j+1$. Since $X \succ_j Y$, we have $b' + 1 = b''$. From the definition of cumulative sum vector,

$$\begin{aligned}
&\mathrm{c}_{X'}(\lfloor\lambda\rfloor) - \mathrm{c}_{Y'}(\lfloor\lambda\rfloor) = \mathrm{c}_{X'}(j+1) - \mathrm{c}_{Y'}(j+1) \\
&= \mathrm{c}_{X'}(j) + X'_{j+1} - \mathrm{c}_{Y'}(j) - Y'_{j+1} \\
&= \mathrm{c}_{X}(j) + X'_{j+1} - \mathrm{c}_{Y}(j) - Y'_{j+1} = 1 + X'_{j+1} - Y'_{j+1}.
\end{aligned}$$

Thus, it is enough to show that $1 + X'_{j+1} \geq Y'_{j+1}$.

Then Lemma 4 also implies the following alternating inequalities

$$\begin{aligned} 0 &= g_{b'+1}(0|u_{j+1}, u_{j+2}) = g_{b'}(0|u_{j+1}, u_{j+2}) \\ &\leq g_{b'+1}(1|u_{j+1}, u_{j+2}) \leq g_{b'}(1|u_{j+1}, u_{j+2}) \leq \cdots \\ &\leq g_{b'+1}(k-1|u_{j+1}, u_{j+2}) \leq g_{b'}(k-1|u_{j+1}, u_{j+2}) \leq g_{b'+1}(k|u_{j+1}, u_{j+2}) \leq \cdots \\ &\leq g_{b'+1}(b'-1|u_{j+1}, u_{j+2}) \leq g_{b'}(b'-1|u_{j+1}, u_{j+2}) = g_{b'+1}(b'|u_{j+1}, u_{j+2}) = 1. \end{aligned}$$

Then it is easy to see that

$$\begin{pmatrix} X'_{j+1} \\ Y'_{j+1} \end{pmatrix} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \ldots, \begin{pmatrix} b'-1 \\ b'-1 \end{pmatrix}, \begin{pmatrix} b'-1 \\ b' \end{pmatrix} \right\}$$

(see Figure 5). From the above, we obtain the inequality $1 + X'_{j+1} \geq Y'_{j+1}$. □

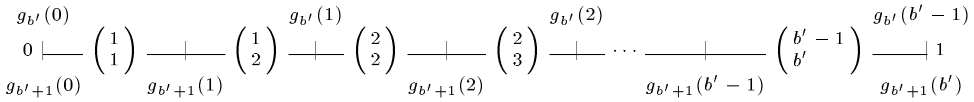


Fig. 5. A figure of alternating inequalities. In the above, we denote $g_{b'}(k|u_{j+1}, u_{j+2})$ and $g_{b'+1}(k|u_{j+1}, u_{j+2})$ by $g_{b'}(k), g_{b'+1}(k)$, respectively.

Lemma 2. *The Markov chain $\mathcal{M}^{\mathrm{P}}$ defined by the update function ϕ is monotone with respect to "$\succeq$", i.e., $\forall \lambda \in [1, n)$, $\forall X, \forall Y \in \Omega$, $X \succeq Y \Rightarrow \phi(X, \lambda) \succeq \phi(Y, \lambda)$.*

Proof: It is easy to see that there exists a sequence of states $Z_1, Z_2, \ldots, Z_r$ with appropriate length satisfying $X = Z_1 \cdot\!\succ Z_2 \cdot\!\succ \cdots \cdot\!\succ Z_r = Y$. Then applying Lemma 1 repeatedly, we can show that $\phi(X, \lambda) = \phi(Z_1, \lambda) \succeq \phi(Z_2, \lambda) \succeq \cdots \succeq \phi(Z_r, \lambda) = \phi(Y, \lambda)$. □

Lastly, we show the correctness of our algorithm.

Proof of Theorem 6: From Lemma 2, the Markov chain defined by the update function ϕ is monotone with respect to the poset $(\Omega, \succeq)$. It is clear that $(X_{\mathrm{U}}, X_{\mathrm{L}})$ is a unique pair of maximum and minimum states. Thus, Algorithm 1 is a monotone CFTP algorithm and Theorems 2 and 3 implies Theorem 6. □

5.2 Expected Running Time

In the rest of this section, we prove Theorem 7 by estimating the expectation of *coalescence time* $T_* \in \mathrm{Z}_{++}$ defined by

$$T_* \overset{\text{def.}}{=} \min\{t > 0 \mid \exists y \in \Omega,\ \forall x \in \Omega,\ y = \Phi^0_{-t}(x, \boldsymbol{\lambda})\}.$$

Note that $\boldsymbol{\lambda}$ is a sequence of random numbers and thus T_* is a random variable.

In [20], Propp and Wilson showed an inequality which gives an upper bound of coalescence time by using mixing rate of corresponding chain. When we employ their result straightforwardly with the (upper bound of) mixing time obtained in Theorem 5, the obtained coalescence time is not tight. In the following, we use their technique carefully and derive a better bound of the coalescence time.

Lemma 3. *Under Condition 1, the coalescence time T_* of $\mathcal{M}$ satisfies* $\mathrm{E}[T_*] = \mathrm{O}(n^3 \ln \Delta)$.

Proof. Let $G = (\Omega, \mathcal{E})$ be the undirected graph and $d^{\mathrm{P}}(X, Y)$, $\forall X, \forall Y \in \Omega$, be the metric on G, both of which are defined in Section 4.

We put $D^* \overset{\text{def.}}{=} d^{\mathrm{P}}(X_{\mathrm{U}}, X_{\mathrm{L}})$ and $\tau_0 \overset{\text{def.}}{=} n(n-1)^2(1 + \ln D^*)$. By using the property $[X \neq Y \Leftrightarrow 1 \leq d^{\mathrm{P}}(X, Y)]$ and the inequality (1) obtained in the proof of Theorem 5, we have that

$$\begin{aligned}
\Pr[T_* > \tau_0] &= \Pr\left[\Phi^0_{-\tau_0}(X_{\mathrm{U}}, \boldsymbol{\lambda}) \neq \Phi^0_{-\tau_0}(X_{\mathrm{L}}, \boldsymbol{\lambda})\right] = \Pr\left[\Phi^{\tau_0}_0(X_{\mathrm{U}}, \boldsymbol{\lambda}) \neq \Phi^{\tau_0}_0(X_{\mathrm{L}}, \boldsymbol{\lambda})\right] \\
&\leq \textstyle\sum_{(X,Y)\in\Omega^2} d^{\mathrm{P}}(X, Y) \Pr\left[X = \Phi^{\tau_0}_0(X_{\mathrm{U}}, \boldsymbol{\lambda}), Y = \Phi^{\tau_0}_0(X_{\mathrm{L}}, \boldsymbol{\lambda})\right] \\
&= \mathrm{E}\left[d^{\mathrm{P}}\left(\Phi^{\tau_0}_0(X_{\mathrm{U}}, \boldsymbol{\lambda}), \Phi^{\tau_0}_0(X_{\mathrm{L}}, \boldsymbol{\lambda})\right)\right] \leq \left(1 - \frac{1}{n(n-1)^2}\right)^{\tau_0} d(X_{\mathrm{U}}, X_{\mathrm{L}}) \\
&= \left(1 - \frac{1}{n(n-1)^2}\right)^{n(n-1)^2(1+\ln D^*)} D^* \leq \mathrm{e}^{-1}\mathrm{e}^{-\ln D^*} D^* \leq \frac{1}{\mathrm{e}}.
\end{aligned}$$

By *submultiplicativity* of coalescence time (see [20] for example), for any $k \in \mathrm{Z}_+$, $\Pr[T_* > k\tau_0] \leq (\Pr[T_* > \tau_0])^k \leq (1/\mathrm{e})^k$. Thus

$$\begin{aligned}
\mathrm{E}[T_*] &= \textstyle\sum_{t=0}^{\infty} t \Pr[T_* = t] \leq \tau_0 + \tau_0 \Pr[T_* > \tau_0] + \tau_0 \Pr[T_* > 2\tau_0] + \cdots \\
&\leq \tau_0 + \tau_0/\mathrm{e} + \tau_0/\mathrm{e}^2 + \cdots = \tau_0/(1 - 1/\mathrm{e}) \leq 2\tau_0 = 2n(n-1)^2(1 + \ln D^*).
\end{aligned}$$

Clearly, $D^* \leq n(\Delta - n)/2 \leq \Delta^2$ because $n \leq \Delta$. Then we obtain the result that $\mathrm{E}[T_*] = \mathrm{O}(n^3 \ln \Delta)$. □

Lastly, we bound the expected number of transitions executed in Algorithm 1.

Proof of Theorem 7: We denote T_* be the coalescence time of our chain $\mathcal{M}^{\mathrm{P}}$. Put $K = \lceil \log_2 T_* \rceil$. Algorithm 1 terminates when we set the starting time period $T = -2^K$ at $(K+1)$st iteration. Then the total number of simulated transitions is bounded by $2(2^0 + 2^1 + 2^2 + \cdots + 2^K) < 2 \cdot 2 \cdot 2^K \leq 8T_*$, since we need to execute two chains from both X_{U} and X_{L}. Thus the expectation of total number of transitions of $\mathcal{M}^{\mathrm{P}}$ is bounded by $\mathrm{O}(\mathrm{E}[8T_*]) = \mathrm{O}(n^3 \ln \Delta)$. □

6 Experimental Study

In this section, we report simulation results. Through all simulations, we use Mersenne Twister [25] as a pseudo-random generator. We ran (almost all) simulations on the PC Linux machine with following specifications.

Machine: Dell Precision 450
CPU: Intel Xeon 2.8GHz (FSB 533MHz) × 2
OS: RedHat Linux 8.0 (Kernel 2.4.18-14smp)
Memory: Dual channel PC2100 DDR SDRAM 2GByte
Compiler: Intel C++ Compiler 7.0

The difference of our two Markov chains are the choice of indices and thus the computational effort required in each transition is (almost) irrelevant to chains. The real running time of 10^{10} transitions by the above machine is between 2∼6 hours.

Approximate Sampler: For each setting of parameters, we ran 10^9 processes of chain $\mathcal{M}^{\mathrm{A}}$. For each Markov chain process, we chose a random seed deterministically and transitions are executed 50 steps. The initial state is an integer vector in Ω obtained by rounding $(\Delta/n, \ldots, \Delta/n)$. (In Section 6.3, we slightly change the above settings.)

Perfect Sampler (Algorithm 1): For each setting of parameters, we executed Algorithm 1 (perfect sampling algorithm) 10^4 times. For each execution, we chose a random seed deterministically and check the coalescence time T_* exactly.

6.1 Influence of Dirichlet Parameters

First, we show results on the relation between Dirichlet parameters and mixing time. We fixed the dimension n to 4 and the discretizing grid size $1/\Delta$ to $1/100$. We selected the vector of Dirichlet parameters from $(1,1,1,1)$, $(4,3,2,1)$, $(2,1.5,1,0.5)$, $(0.1,0.1,0.1,0.1)$, $(0.4,0.3,0.2,0.1)$ and $(10^{-5},10^{-5},10^{-5},10^{-5})$. We note that the case $(1,1,1,1)$ corresponds to the uniform distribution over Ω.

Figure 6 shows results of our approximate sampler based on $\mathcal{M}^{\mathrm{A}}$. Along the vertical axis we give the total variation distance ε between the true distribution and the probability distribution obtained by 10^9 processes. The horizontal axis means the number of transitions of chains from the initial state. As Figure 6 shows, the decrease of total variation distance are saturated at about 0.005, though it must descend constantly. This is caused by the limitation of the number of samples (10^9), that is, the total variation distance has a positive lower bound. Figure 6 shows that the larger number of transitions we execute, the smaller the difference will be. Aside from this saturation, we can see that if the value of a Dirichlet parameter is greater than or equal to 1, the mixing time is less than the case that values of all Dirichlet parameters are less than 1. The above discussions are limited, because we only dealt with the case that $n = 4$.

Table 2 show the distributions of coalescence times of our CFTP algorithm. Contrary to the approximate sampler, the coalescence time becomes smaller when the values of all Dirichlet parameters are less than 1.

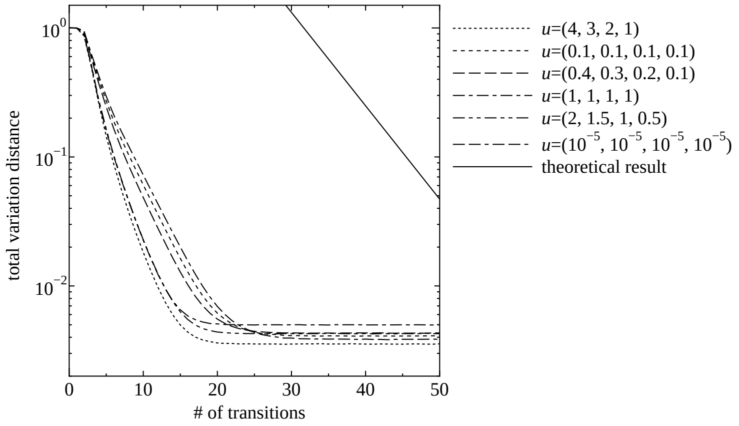


Fig. 6. Approximate Sampler: Influence of Dirichlet parameters (u_1, u_2, u_3, u_4).

Table 2. Perfect Sampler: Influence of Dirichlet parameters (u_1, u_2, u_3, u_4).

			coalescence time		
n	$1/\Delta$	(u_1, u_2, u_3, u_4)	ave. (s. d.)	max.	min.
4	1/100	(1,1,1,1)	45.5 (16.4)	129	4
4	1/100	(4,3,2,1)	41.3 (13.1)	114	9
4	1/100	(2,1.5,1,0.5)	39.6 (14.5)	127	3
4	1/100	(0.1,0.1,0.1,0.1)	34.9 (17.9)	141	3
4	1/100	(0.4, 0.3, 0.2, 0.1)	36.4 (17.5)	140	3
4	1/100	$(10^{-5}, 10^{-5}, 10^{-5}, 10^{-5})$	33.2 (17.7)	125	3

ave.: average, s. d.: standard deviation

6.2 Influence of Grid Size

Next, we confirm how the grid size $1/\Delta$ contribute to the mixing time. We fixed the dimension n to 4 again and the parameter to $(1, 1, 1, 1)$. We chose Δ from 10, 20, 50, 100, and 200.

Figure 7 gives results on our approximate sampler. We plotted the total variation distance ε for each Δ. This figure shows that Δ will have little contribution to the mixing time. More specifically, until the decrease of ε is saturated, the ratios of decreasing have little difference for each Δ. In the proof of Theorem 4, the term $(\Delta - n)$ is artificially introduced as the diameter of the graph $G = (\Omega, A)$. These experimental results, however, suggest that the mixing time of $\mathcal{M}^{\mathrm{A}}$ does not depend on Δ and thus $\mathcal{M}^{\mathrm{A}}$ gives a strongly polynomial time approximate sampler. This property is substantiated by the fact that the diameter of our chain is bounded by n and independent of Δ.

For perfect sampler, we added the case that $\Delta = 400$ and 16-dimensional cases. Table 3 shows the distributions of coalescence times. The average coalescence time increases with respect to Δ. Thus, it seems that the coalescence time of chain $\mathcal{M}^{\mathrm{P}}$ includes a linear term of $\ln \Delta$ and thus $\mathcal{M}^{\mathrm{P}}$ gives not strongly but weakly polynomial time sampler.

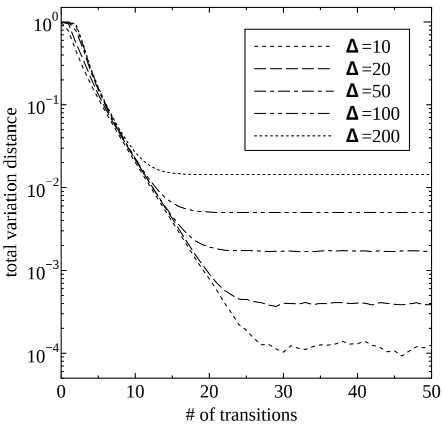


Fig. 7. Approximate Sampler: Influence of grid size $1/\Delta$.

6.3 Influence of Dimension

Finally, we checked the influence of the dimension n. For approximate sampler, we fixed the discretizing grid size $1/\Delta$ to $1/26$ and chose the dimension n between 3 and 13, because of restriction of the memory. We also fixed each parameter to 1. For each setting of the dimension, we ran 2×10^9 processes of chain $\mathcal{M}^{\mathrm{A}}$ with the initial state $(\Delta - n + 1, 1, \ldots, 1)$. We show all results in Figure 8. Since our purpose is to compare the mixing time and dimension, we picked up the first time instance that the total variation distance ε falls short of 0.5, 0.2, 0.1, and/or 0.05. These picked time instances are marked by •, ○, ■ and □ respectively in Figure 8. In Figure 8 (b), we show the results for each ε. Though accurate consideration cannot be made because of the insufficient range of dimension, our results indicate that the mixing time of $\mathcal{M}^{\mathrm{A}}$ is $\mathrm{o}(n^2)$.

For perfect sampler, we added the cases that $\Delta = 400$ and $(u_1, \ldots, u_n) = (0.5, \ldots, 0.5), (2, \ldots, 2)$. Table 4 shows the distributions of coalescence times. The results convince the tightness of our bounds, i.e., both the mixing rate and the coalescence time of Algorithm 1 are $\Theta(n^3)$ for fixed Δ.

7 Discussion

In this paper, we proposed two Markov chains $\mathcal{M}^{\mathrm{A}}$ and $\mathcal{M}^{\mathrm{P}}$ for sampling a random vector distributed according to a discretized Dirichlet distribution. We also showed that the mixing rates of $\mathcal{M}^{\mathrm{A}}$ and $\mathcal{M}^{\mathrm{P}}$ are bounded by $\mathrm{O}(n^2 \log \Delta)$

Table 3. Perfect Sampler: Influence of grid size $1/\varDelta$.

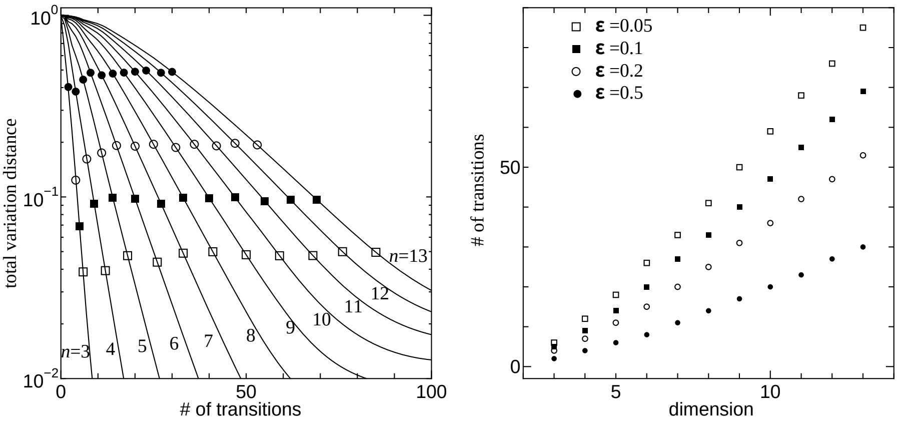
			coalescence time				
n	$1/\varDelta$	$(u_1, u_2, \ldots, u_n)$	ave.	(s. d.)	max.	min.	$\frac{\text{ave.}}{\ln \varDelta}$
4	1/10	(1,1,1,1)	24.9	(13.1)	107	3	10.8
4	1/20	(1,1,1,1)	32.2	(14.4)	121	3	10.7
4	1/50	(1,1,1,1)	40.1	(15.8)	150	3	10.3
4	1/100	(1,1,1,1)	45.5	(16.4)	129	4	9.9
4	1/200	(1,1,1,1)	50.5	(17.3)	153	6	9.5
4	1/400	(1,1,1,1)	55.6	(18.0)	149	8	9.0
16	1/20	$(1, 1, \ldots, 1)$	2,030.7	(940.1)	7,617	308	469.9
16	1/50	$(1, 1, \ldots, 1)$	3,612.4	(978.0)	10,967	1,236	640.1
16	1/100	$(1, 1, \ldots, 1)$	4,303.7	(1,000.2)	11,612	1,942	647.8
16	1/200	$(1, 1, \ldots, 1)$	4,888.2	(1,017.4)	10,530	2,235	639.5
16	1/500	$(1, 1, \ldots, 1)$	5,624.8	(1,028.9)	12,334	3,001	627.4
16	$1/10^3$	$(1, 1, \ldots, 1)$	6,152.6	(1,048.9)	15,050	3,376	617.4
16	$1/10^4$	$(1, 1, \ldots, 1)$	7,899.4	(1,113.5)	15,388	4,862	594.5
16	$1/10^5$	$(1, 1, \ldots, 1)$	9,626.1	(1,133.3)	15,961	6,341	579.5
16	$1/10^6$	$(1, 1, \ldots, 1)$	11,368.5	(1,180.5)	17,790	7,768	570.4
16	$1/10^7$	$(1, 1, \ldots, 1)$	13,086.4	(1,216.0)	19,576	9,346	562.8
16	$1/10^8$	$(1, 1, \ldots, 1)$	14,826.5	(1,271.0)	22,178	10,984	557.9
16	$1/10^9$	$(1, 1, \ldots, 1)$	16,559.6	(1,308.0)	23,744	12,631	553.9

ave.: average, s. d.: standard deviation

and $\mathrm{O}(n^3 \log \varDelta)$ respectively, by using path coupling method where n is the dimension and $1/\varDelta$ is the discretizing grid size. The obtained bounds of mixing rates do not depend on the magnitudes of parameters. By employing Propp and Wilson's results on monotone CFTP algorithm, we can construct a perfect sampling algorithm based on $\mathcal{M}^{\mathrm{P}}$. The expected number of transitions required in our perfect sampling algorithm is bounded by $\mathrm{O}(n^3 \log \varDelta)$

Our second chain $\mathcal{M}^{\mathrm{P}}$ is obtained from our first chain $\mathcal{M}^{\mathrm{A}}$ by restricting possible pairs of indices (to be updated) from $n(n-1)/2$ to $n-1$. Thus, it seems that $\mathcal{M}^{\mathrm{P}}$ is $\Theta(n)$ times slower than $\mathcal{M}^{\mathrm{A}}$. However, our computational experience indicates that our two chains have totally different features. Our results of simulations say that the mixing rate of $\mathcal{M}^{\mathrm{A}}$ is independent of the grid size $1/\varDelta$ and bounded by $\mathrm{o}(n^2)$. However, strong polynomiality of the mixing rate of $\mathcal{M}^{\mathrm{A}}$ remains open even in the case that all the parameters are greater than or equal to 1. On the contrary, the results of simulations convince that the coalescence time of chain $\mathcal{M}^{\mathrm{P}}$ is $\Theta(n^3 \log \varDelta)$. Thus, the time complexity of our perfect sampling algorithm is weakly polynomial of the input size. To construct a strongly polynomial time perfect sampler, it seems that we need a different Markov chain. In case of uniform distribution, Randall and Winkler [22] proposed a strongly polynomial time 'approximate' sampler which mixes in $\Theta(n^3 \log n)$.

The key property for constructing our chains and perfect sampling algorithm is the *alternating inequalities* shown in Lemma 4. The property appears implicitely in Dyer and Greenhil's paper [6], which proposed an approximately



(a): Dimension and total variation distance. (b): Dimension and mixing time.

Fig. 8. Approximate Sampler: Influence of dimension n.

uniform sampler for $2 \times J$ contingency tables. Matsui, Matsui and Ono [17] extended their result to $2 \times 2 \times \cdots \times 2 \times J$ contingency tables with conditional multinomial distributions. Recently, Kijima and Matsui [13] proposed a class of logarithmic separable concave distributions defined on a discretized simplex and showed that every probability function in the class satisfies the alternating inequalities. The result implies that there exist a perfect sampler and a weakly polynomial time approximate sampler with respect to every probability function in the class. The class includes discretized Dirichlet distributions whose Dirichlet parameters are greater than or equal to 1. The difficulty of proving Lemma 4 results from the cases that all the Dirichlet parameters are less than 1.

To show the polynomiality of our perfect sampling algorithm, we need a specified edge length function defined in Subsection 4.2, and the property shown in Lemma 5. The edge length function originated from Kijima and Matsui's recent paper [12] which proposes a polynomial time perfect sampling algorithm for $2 \times J$ contingency tables. The main difference is that when we deal with the uniform distribution on $2 \times J$ contingency tables, the transition probabilities of the Markov chain are predetermined. However, in case of discretized Dirichlet distribution, the transition probabilities vary with respect to the magnitude of parameters. Thus, we need complicate discussions in the proof of Lemma 5, especially in cases that all the Dirichlet parameters are less than 1. Showing the inequality appearing in Lemma 5 is much easier in case of the uniform distribution on $2 \times J$ contingency tables.

Table 4. Perfect Sampler: Influence of dimension n.

			coalescence time				
n	$1/\Delta$	$(u_1, u_2, \ldots, u_n)$	ave.	(s. d.)	max.	min.	$\frac{\ln(\text{ave.})}{\ln n}$
3	1/20	$(1, 1, 1)$	10.2	(5.4)	42	2	1.54
4	1/20	$(1, 1, 1, 1)$	32.2	(14.4)	121	3	1.92
5	1/20	$(1, 1, \ldots, 1)$	71.3	(29.1)	238	6	2.09
6	1/20	$(1, 1, \ldots, 1)$	129.4	(48.8)	454	14	2.16
7	1/20	$(1, 1, \ldots, 1)$	213.3	(79.6)	754	36	2.24
4	1/400	$(1, 1, 1, 1)$	55.6	(18.0)	149	8	2.90
8	1/400	$(1, 1, \ldots, 1)$	611.1	(134.4)	1,502	218	3.09
16	1/400	$(1, 1, \ldots, 1)$	5,431.5	(1,034.4)	12,896	2,550	3.10
32	1/400	$(1, 1, \ldots, 1)$	45,324.8	(8,146.6)	103,033	27,448	3.09
64	1/400	$(1, 1, \ldots, 1)$	364,470.2	(64,404.3)	860,360	226,763	3.08
128	1/400	$(1, 1, \ldots, 1)$	2,865,607.6	(525,024.8)	6,589,297	1,778,885	3.06
4	1/400	$(0.5, 0.5, \ldots, 0.5)$	48.6	(19.6)	162	3	2.80
8	1/400	$(0.5, 0.5, \ldots, 0.5)$	577.2	(156.4)	1,466	152	3.06
16	1/400	$(0.5, 0.5, \ldots, 0.5)$	5,313.5	(1,119.6)	15,007	2,438	3.09
32	1/400	$(0.5, 0.5, \ldots, 0.5)$	44,859.7	(8,458.7)	117,760	24,232	3.09
64	1/400	$(0.5, 0.5, \ldots, 0.5)$	362,574.2	(65,575.4)	843,410	207,957	3.08
4	1/400	$(2, 2, \ldots, 2)$	60.9	(16.2)	172	20	2.96
8	1/400	$(2, 2, \ldots, 2)$	635.0	(123.3)	1,741	316	3.10
16	1/400	$(2, 2, \ldots, 2)$	5,543.1	(972.4)	13,273	3,218	3.11
32	1/400	$(2, 2, \ldots, 2)$	45,525.0	(7,876.1)	98,410	28,206	3.09
64	1/400	$(2, 2, \ldots, 2)$	366,243.8	(63,508.6)	830,563	229,377	3.08

ave.: average, s. d.: standard deviation

Appendix

Lemma 4. *The pair of functions g_b and g_{b+1} satisfies that*

$$\forall b \in \{2, 3, \ldots\},\ \forall u_i, \forall u_j \geq 0,\ \forall k \in \{1, 2, \ldots, b\},$$
$$g_{b+1}(k-1|u_i, u_j) \leq g_b(k-1|u_i, u_j) \leq g_{b+1}(k|u_i, u_j).$$

This lemma is essentially equivalent to a lemma appearing in Appendix section of the paper [16] without proof. Figure 9 illustrates an image of alternating inequalities.

Proof: In the following, we show the second inequality. We can show the first inequality in a similar way.

We denote $C(u_i, u_j, b+1) = C_{b+1}$ and $C(u_i, u_j, b) = C_b$ for simplicity. From the definition of $g_b(k|u_i, u_j)$, we obtain that

$$\begin{aligned} H(k) &\overset{\text{def.}}{=} g_{b+1}(k|u_i, u_j) - g_b(k-1|u_i, u_j) \\ &= \textstyle\sum_{l=1}^{k} C_{b+1} l^{u_i-1}(b-l+1)^{u_j-1} - \sum_{l=1}^{k-1} C_b l^{u_i-1}(b-l)^{u_j-1} \\ &= (1 - C_{b+1} \textstyle\sum_{l=k+1}^{b} l^{u_i-1}(b-l+1)^{u_j-1}) - (1 - C_b \sum_{l=k}^{b-1} l^{u_i-1}(b-l)^{u_j-1}) \end{aligned}$$

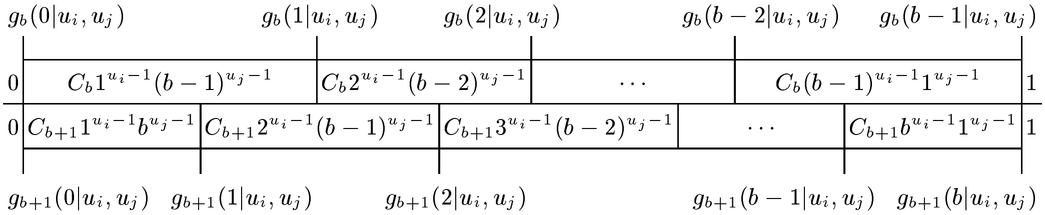


Fig. 9. A figure of alternating inequalities, where $C_{b+1} = C(u_i, u_j, b+1)$ and $C_b = C(u_i, u_j, b)$.

$$\begin{aligned}
&= C_b \textstyle\sum_{l=k+1}^{b}(l-1)^{u_i-1}(b-l+1)^{u_j-1} - C_{b+1}\sum_{l=k+1}^{b} l^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle\sum_{l=k+1}^{b}(C_b(l-1)^{u_i-1}(b-l+1)^{u_j-1} - C_{b+1}l^{u_i-1}(b-l+1)^{u_j-1}) \\
&= \textstyle\sum_{l=k+1}^{b} C_b l^{u_i-1}(b-l+1)^{u_j-1}\left(\left(1-\frac{1}{l}\right)^{u_i-1} - \frac{C_{b+1}}{C_b}\right).
\end{aligned}$$

Similarly, we can also show that

$$\begin{aligned}
H(k) &= C_{b+1}\textstyle\sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1} - C_b\sum_{l=1}^{k-1} l^{u_i-1}(b-l)^{u_j-1} \\
&\geq C_{b+1}\textstyle\sum_{l=2}^{k} l^{u_i-1}(b-l+1)^{u_j-1} - C_b\sum_{l=2}^{k}(l-1)^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle\sum_{l=2}^{k}(C_{b+1}l^{u_i-1}(b-l+1)^{u_j-1} - C_b(l-1)^{u_i-1}(b-l+1)^{u_j-1}) \\
&= \textstyle\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}\left(\frac{C_{b+1}}{C_b} - \left(1-\frac{1}{l}\right)^{u_i-1}\right).
\end{aligned}$$

By introducing the function $h : \{2, 3, \ldots, b\} \to \mathrm{R}$ defined by $h(l) \overset{\text{def.}}{=} \left(1-\frac{1}{l}\right)^{u_i-1} - \frac{C_{b+1}}{C_b}$, we have the following equality and inequality

$$\begin{aligned}
H(k) &= \textstyle\sum_{l=k+1}^{b} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l) &(2)\\
&\geq -\textstyle\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l). &(3)
\end{aligned}$$

(a) Consider the case that $u_i \geq 1$.

Since $u_i - 1 \geq 0$, the function $h(l)$ is monotone non-decreasing. When $h(k) \geq 0$ holds, we have $0 \leq h(k) \leq h(k+1) \leq \cdots \leq h(b)$, and so (2) implies the non-negativity $H(k) \geq 0$. If $h(k) < 0$, then inequalities $h(2) \leq h(3) \leq \cdots \leq h(k) < 0$ hold, and so (3) implies that $H(k) \geq -\sum_{l=2}^{k} C_b l^{u_i-1}(b-l+1)^{u_j-1}h(l) \geq 0$.

(b) Consider the case that $0 \leq u_i \leq 1$.

Since $u_i - 1 \leq 0$, the function $h(l)$ is monotone non-increasing. If the inequality $h(b) \geq 0$ hold, we have $h(2) \geq h(3) \geq \cdots \geq h(b) \geq 0$ and inequality (2) implies the non-negativity $H(k) \geq 0$. Thus, we only need to show that $h(b) = (\frac{b-1}{b})^{u_i-1} - \frac{C_{b+1}}{C_b} \geq 0$.

In the rest of this proof, we substitute $u_{i'} - 1$ by $\alpha_{i'}$ for all i'. We define a function $H_0(b, \alpha_i, \alpha_j)$ by $H_0(b, \alpha_i, \alpha_j) \overset{\text{def.}}{=} (b-1)^{\alpha_i}C_{b+1}^{-1} - b^{\alpha_i}C_b^{-1}$. It is clear that if the condition $[-1 \leq \forall\alpha_i \leq 0,\ -1 \leq \forall\alpha_j,\ \forall b \in \{2,3,4,\ldots\},\quad H_0(b,\alpha_i,\alpha_j) \geq 0]$ holds, we obtain the required result that $h(b) \geq 0$ for each $b \in \{2,3,4,\ldots\}$. Now we transform the function $H_0(b,\alpha_i,\alpha_j)$ and obtain another expression as follows;

$$
\begin{aligned}
&H_0(b,\alpha_i,\alpha_j) = (b-1)^{\alpha_i}\sum_{k=1}^{b} k^{\alpha_i}(b-k+1)^{\alpha_j} - b^{\alpha_i}\sum_{k=1}^{b-1} k^{\alpha_i}(b-k)^{\alpha_j} \\
&= \textstyle\sum_{k=1}^{b}(b-1)^{\alpha_i}k^{\alpha_i}(b-k+1)^{\alpha_j}\frac{(b-k)+(k-1)}{b-1} - b^{\alpha_i}\sum_{k=1}^{b-1}k^{\alpha_i}(b-k)^{\alpha_j} \\
&= \textstyle\sum_{k=1}^{b-1}\Big[(b-1)^{\alpha_i}k^{\alpha_i}(b-k+1)^{\alpha_j}\left(\frac{b-k}{b-1}\right) + (b-1)^{\alpha_i}(k+1)^{\alpha_i}(b-k)^{\alpha_j}\left(\frac{k}{b-1}\right) \\
&\qquad\qquad -b^{\alpha_i}k^{\alpha_i}(b-k)^{\alpha_j}] \\
&= \textstyle\sum_{k=1}^{b-1}\frac{(b-1)^{\alpha_i}k^{\alpha_i}(b-k)^{\alpha_j}}{b-1}\left[\left(1+\frac{1}{b-k}\right)^{\alpha_j}(b-k) + \left(1+\frac{1}{k}\right)^{\alpha_i}k - \left(\frac{b}{b-1}\right)^{\alpha_i}(b-1)\right].
\end{aligned}
$$

Then it is enough to show that the function

$$
H_1(b,\alpha_i,\alpha_j,k) \overset{\text{def.}}{=} \left(1+\tfrac{1}{b-k}\right)^{\alpha_j}(b-k) + \left(1+\tfrac{1}{k}\right)^{\alpha_i}k - \left(\tfrac{b}{b-1}\right)^{\alpha_i}(b-1)
$$

is nonnegative for any $k \in \{1,2,\ldots,b-1\}$. Since $1+1/(b-k) > 1$ and $\alpha_j \geq -1$, we have

$$
H_1(b,\alpha_i,\alpha_j,k) \geq H_1(b,\alpha_i,-1,k) = \tfrac{(b-k)^2}{b-k+1} + \left(1+\tfrac{1}{k}\right)^{\alpha_i}k - \left(\tfrac{b}{b-1}\right)^{\alpha_i}(b-1).
$$

We differentiate the function H_1 by α_i, and obtain the following

$$
\begin{aligned}
\frac{\partial}{\partial\alpha_i}H_1(b,\alpha_i,-1,k) &= \left(1+\tfrac{1}{k}\right)^{\alpha_i}k\log\left(1+\tfrac{1}{k}\right) - \left(\tfrac{b}{b-1}\right)^{\alpha_i}(b-1)\log\left(\tfrac{b}{b-1}\right) \\
&= \left(1+\tfrac{1}{k}\right)^{\alpha_i}\log\left(1+\tfrac{1}{k}\right)^{k} - \left(1+\tfrac{1}{b-1}\right)^{\alpha_i}\log\left(1+\tfrac{1}{b-1}\right)^{(b-1)}.
\end{aligned}
$$

Since k, b is a pair of positive integers satisfying $1 \leq k \leq b-1$, the non-positivity of α_i implies $0 \leq (1+1/k)^{\alpha_i} \leq (1+1/(b-1))^{\alpha_i}$ and $0 \leq \log(1+1/k)^k \leq \log(1+1/(b-1))^{b-1}$. Thus the function $H_1(b,\alpha_i,-1,k)$ is monotone non-increasing with respect to $\alpha_i \leq 0$. Thus we have

$$
\begin{aligned}
H_1(b,\alpha_i,-1,k) \geq H_1(b,0,-1,k) &= \tfrac{(b-k)^2}{b-k+1} + \left(1+\tfrac{1}{k}\right)^0 k - \left(\tfrac{b}{b-1}\right)^0(b-1) \\
&= \tfrac{(b-k)^2}{b-k+1} + k - b + 1 = \tfrac{(b-k)^2+1^2-(b-k)^2}{b-k+1} = \tfrac{1}{b-k+1} \geq 0.
\end{aligned}
$$

□

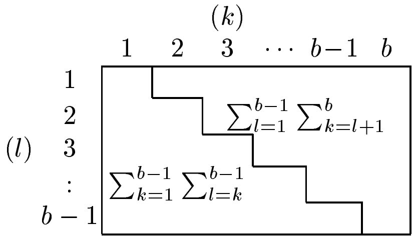
Lemma 5. *The pair of functions g_b and g_{b+1} satisfies that*

$$
\forall b \in \{2,3,\ldots\},\quad \forall u_i \geq \forall u_j,
$$

$$
\textstyle\sum_{k=1}^{b-1}[g_b(k|u_i,u_j) - g_{b+1}(k|u_i,u_j)] - \sum_{k=1}^{b-1}[g_{b+1}(k|u_i,u_j) - g_b(k-1|u_i,u_j)] \geq 0.
$$

Proof: We denote $C(u_i, u_j, b+1) = C_{b+1}$ and $C(u_i, u_j, b) = C_b$ for simplicity. It is not difficult to show the following equalities,

$$
\begin{aligned}
G &\overset{\text{def.}}{=} \textstyle\sum_{k=1}^{b-1}[g_b(k|u_i,u_j) - g_{b+1}(k|u_i,u_j)] - \sum_{k=1}^{b-1}[g_{b+1}(k|u_i,u_j) - g_b(k-1|u_i,u_j)] \\
&= \textstyle\sum_{k=1}^{b-1} g_b(k|u_i,u_j) - \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) \\
&\quad \textstyle -\sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) + \sum_{k=1}^{b-1} g_b(k-1|u_i,u_j) \\
&= \textstyle\sum_{k=1}^{b-1} g_b(k|u_i,u_j) - \sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) \\
&\quad \textstyle -\sum_{k=1}^{b-1} g_{b+1}(k|u_i,u_j) + \sum_{k=2}^{b-1} g_b(k-1|u_i,u_j) \\
&= \textstyle\sum_{k=1}^{b-1} \Bigl(C_b \sum_{l=1}^{k} l^{u_i-1}(b-l)^{u_j-1}\Bigr) - \sum_{k=1}^{b-1} \Bigl(C_{b+1} \sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1}\Bigr) \\
&\quad \textstyle -\sum_{k=1}^{b-1} \Bigl(C_{b+1} \sum_{l=1}^{k} l^{u_i-1}(b-l+1)^{u_j-1}\Bigr) + \sum_{k=2}^{b-1} \Bigl(C_b \sum_{l=1}^{k-1} l^{u_i-1}(b-l)^{u_j-1}\Bigr) \\
&= \textstyle C_b \sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l)^{u_j-1} - C_{b+1} \sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l+1)^{u_j-1} \\
&\quad \textstyle -C_{b+1} \sum_{l=1}^{b-1}(b-l)l^{u_i-1}(b-l+1)^{u_j-1} + C_b \sum_{l=1}^{b-1}(b-l-1)l^{u_i-1}(b-l)^{u_j-1} \\
&= \textstyle C_b \sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1} - C_{b+1} \sum_{l=1}^{b-1}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1} \\
&= \textstyle C_b C_{b+1} \Bigl(\ C_{b+1}^{-1} \sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1} \\
&\qquad\qquad \textstyle -C_b^{-1} \sum_{l=1}^{b}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1}\Bigr) \\
&= \textstyle C_b C_{b+1} \Bigl(\ \Bigl(\sum_{k=1}^{b} k^{u_i-1}(b-k+1)^{u_j-1}\Bigr)\Bigl(\sum_{l=1}^{b-1}(2b-2l-1)l^{u_i-1}(b-l)^{u_j-1}\Bigr) \\
&\qquad\qquad \textstyle -\Bigl(\sum_{k=1}^{b-1} k^{u_i-1}(b-k)^{u_j-1}\Bigr)\Bigl(\sum_{l=1}^{b}(2b-2l)l^{u_i-1}(b-l+1)^{u_j-1}\Bigr)\Bigr) \\
&= \textstyle C_b C_{b+1} \Bigl(\ \sum_{k=1}^{b} \sum_{l=1}^{b-1}(2b-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{k=1}^{b-1} \sum_{l=1}^{b}(2b-2l)(kl)^{u_i-1}((b-k)(b-l+1))^{u_j-1}\Bigr) \\
&= \textstyle C_b C_{b+1} \Bigl(\ \sum_{k=1}^{b} \sum_{l=1}^{b-1}(2b-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{l=1}^{b-1} \sum_{k=1}^{b}(2b-2k)(lk)^{u_i-1}((b-l)(b-k+1))^{u_j-1}\Bigr) \\
&= \textstyle C_b C_{b+1} \sum_{k=1}^{b} \sum_{l=1}^{b-1} \bigl((2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1}\bigr) \\
&= \tfrac{C_b C_{b+1}}{2} \textstyle \Bigl(\ \sum_{k=1}^{b} \sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle +\sum_{k=1}^{b} \sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1}\Bigr) \\
&= \tfrac{C_b C_{b+1}}{2} \textstyle \Bigl(\sum_{k=1}^{b} \sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\quad \textstyle +\sum_{k=1}^{b} \sum_{l=1}^{b-1}(2(b-k+1)-2(b-l)-1) \\
&\quad\ \ ((b-k+1)(b-l))^{u_i-1}((b-(b-k+1)+1)(b-(b-l)))^{u_j-1}\Bigr) \\
&= \tfrac{C_b C_{b+1}}{2} \textstyle \Bigl(\ \sum_{k=1}^{b} \sum_{l=1}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
&\qquad\qquad \textstyle -\sum_{k=1}^{b} \sum_{l=1}^{b-1}(2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1}\Bigr)
\end{aligned}
$$



(k)
1 2 3 $\cdots$ $b-1$ b
(l)
1
2
3
:
$b-1$
$\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}$
$\sum_{k=1}^{b-1}\sum_{l=k}^{b-1}$

$$
\begin{aligned}
= \tfrac{C_bC_{b+1}}{2}\Big(& \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& + \textstyle\sum_{k=1}^{b-1}\sum_{l=k}^{b-1}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& - \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& - \textstyle\sum_{k=1}^{b-1}\sum_{l=k}^{b-1}(2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1}\Big) \\
= \tfrac{C_bC_{b+1}}{2}\Big(& \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)(kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& - \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\
& - \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& + \textstyle\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1}\Big) \\
= \frac{C_bC_{b+1}}{2}\sum_{l=1}^{b-1}\sum_{k=l+1}^{b}(2k-2l-1)\Big(& (kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& -(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\
& -((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& +((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1}\Big).
\end{aligned}
$$

We define a function $G_0(k,l,u_i,u_j)$ by

$$
\begin{aligned}
G_0(k,l,u_i,u_j) \overset{\text{def.}}{=} & (kl)^{u_i-1}((b-k+1)(b-l))^{u_j-1} \\
& -(l(k-1))^{u_i-1}((b-l+1)(b-k+1))^{u_j-1} \\
& -((b-k+1)(b-l))^{u_i-1}(kl)^{u_j-1} \\
& +((b-l+1)(b-k+1))^{u_i-1}(l(k-1))^{u_j-1}.
\end{aligned}
$$

Since $1 \le l < l+1 \le k \le b$, it is clear that $(2k-2l-1) > 0$. Thus, we only need to show that $\forall l \in \{1,2,\ldots,b-1\}$, $\forall k \in \{2,3,\ldots,b\}$, $\forall u_i \ge \forall u_j$, $G_0(l,k,u_i,u_j) \ge 0$. It is easy to see that

$$
\begin{aligned}
& G_0(k,l,u_i,u_j) = \\
& (l(k-1))^{u_i-1}(b-k+1)^{u_j-1}\left((1+\tfrac{1}{k-1})^{u_i-1}(b-l)^{u_j-1}-(b-l+1)^{u_j-1}\right) \\
& +((b-k+1)(b-l))^{u_i-1}l^{u_j-1}\left(-k^{u_j-1}+(1+\tfrac{1}{b-l})^{u_i-1}(k-1)^{u_j-1}\right).
\end{aligned}
$$

Then it is clear that $G_0(k,l,u_i,u_j)$ is non-decreasing with respect to u_i and so $G_0(k,l,u_i,u_j) \ge G_0(k,l,u_j,u_j)$. By substituting u_i by u_j in the definition of $G_0(k,l,u_i,u_j)$, it is easy to see that $G_0(k,l,u_j,u_j) = 0$. Thus we have the desired result. □

References

1. Bubley, R. and Dyer, M.: Path coupling: A technique for proving rapid mixing in Markov chains, 38th Annual Symposium on Foundations of Computer Science, IEEE, San Alimitos, 1997, 223–231.
2. Bubley, R.: Randomized Algorithms : Approximation, Generation, and Counting, Springer, New York, 2001.
3. Burr, T. L.: Quasi-equilibrium theory for the distribution of rare alleles in a subdivided population: justification and implications, *Theoretical Population Biology*, **57** (2000), 297–306.
4. Burr, D., Doss, H., Cooke, G. E. and Goldschmidt-Clermont, P. J.: A meta-analysis of studies on the association of the platelet PlA polymorphism of glycoprotein IIIa and risk of coronary heart disease, *Statistics in Medicine*, **22** (2003), 1741–1760.
5. Chiang, J., Chib, S. and Narasimhan, C.: Markov chain Monte Carlo and models of consideration set and parameter heterogeneity, *Journal of Econometrics*, **89** (1999), 223–248.
6. M. Dyer and C. Greenhill, Polynomial-time counting and sampling of two-rowed contingency tables, *Theoretical Computer Science*, **246** (2000), 265–278.
7. Dimakos, X. K.: A guide to exact simulation, *International Statistical Review*, **69** (2001), 27–48.
8. Durbin, R., Eddy, R., Krogh, A. and Mitchison, G.: Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge Univ. Press, 1998.
9. Fill, J.: An interruptible algorithm for perfect sampling via Markov chains. *The Annals of Applied Probability*, **8** (1988), 131–162.
10. Fill, J., Machida, M., Murdoch, D. and Rosenthal, J.: Extension of Fill's perfect rejection sampling algorithm to general chains. *Random Structures and Algorithms*, **17** (2000), 290–316.
11. Graham, J., Curran, J. and Weir, B. S.: Conditional genotypic probabilities for microsatellite loci, *Genetics*, **155** (2000), 1973–1980.
12. Kijima, S. and Matsui, T.: Polynomial time perfect sampling algorithm for two-rowed contingency tables, *Random Structures and Algorithms* (to appear).
13. Kijima, S. and Matsui, T.: Rapidly mixing chain and perfect sampler for logarithmic separable concave distributions on simplex, Proceedings of the 2005 International Conference on the Analysis of Algorithms (AofA), Discrete Mathematics and Computer Science, DMTCS Proceedings Series, Volume AD, 2005, 369–380.
14. Kitada, S., Hayashi, T. and Kishino, H.: Empirical Bayes procedure for estimating genetic distance between populations and effective population size, *Genetics*, **156** (2000), 2063–2079.
15. Laval, G., SanCristobal, M. and Chevalet C.: Maximum-likelihood and Markov chain Monte Carlo approaches to estimate inbreeding and effective size form allele frequency changes, *Genetics*, **164** (2003), 1189–1204.
16. Matsui, T., Motoki, M. and Kamatani, N.: Polynomial time approximate sampler for discretized Dirichlet distribution, Proceedings of 14th International Symposium on Algorithms and Computation (ISAAC 2003), Lecture Notes in Computer Science, Springer, **2906** (2003), 676–685.
17. Matsui, T., Matsui, Y. and Ono, Y.: Random generation of $2 \times 2 \times \cdots \times 2 \times J$ contingency tables, *Theoretical Computer Science*, **326** (2004), 117–135.
18. Niu, T., Qin, Z. S., Xu, X. and Liu, J. S.: Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms, *American Journal of Human Genetics*, **70** (2002), 157–169.

19. Pritchard, J. K., Stephens, M. and Donnely, P.: Inference of population structure using multilocus genotype data, *Genetics*, **155** (2000) 945–959.
20. Propp, J. and Wilson, D.: Exact sampling with coupled Markov chains and applications to statistical mechanics, *Random Structures and Algorithms*, **9** (1996), 223–252.
21. Propp, J. and Wilson, D.: How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph, *Journal of Algorithms*, **27** (1998), 170–217.
22. Randall, D. and Winkler, P.: Mixing Points on an Interval, Proceedings of the Second Workshop on Analytic Algorithms and Combinatorics, Vancouver, 2005, 216–221.
23. Robert, C. P.: The Bayesian Choice, Springer, New York, 2001.
24. Wilson, D.: How to couple from the past using a read-once source of randomness, *Random Structures and Algorithms*, **16** (2000), 85–113.
25. Mersenne Twister Home Page, http://www.math.keio.ac.jp/˜matumoto/mt.html