Reinforcement Learning-Aided Markov Chain Monte Carlo For Lattice Gaussian Sampling

Zheng Wang and Yili Xia
School of Information Science and Engineering
Southeast University
Nanjing, 210096, China
Email: z.wang@ieee.org and yili_xia@seu.edu.cn

Shanxiang Lyu
College of Cyber Security
Jinan University
Guangzhou, 510632, China
Email: shanxianglyu@gmail.com

Cong Ling
Department of EEE
Imperial College London
London, SW7 2AZ, United Kingdom
Email: cling@ieee.org

2021 IEEE Information Theory Workshop (ITW) | 978-1-6654-0312-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITW48936.2021.9611412

***Abstract*—Sampling from the lattice Gaussian distribution has emerged as a key problem in coding, decoding and cryptography. In this paper, the Gibbs sampling from Markov chain Monte Carlo (MCMC) methods is investigated for lattice Gaussian sampling. Firstly, the error function of random scan Gibbs sampling is derived, and we show that it is partially determined by the selection probabilities over the sampling components. Then, in order to minimize the error function for a better sampling performance, a reinforcement learning mechanism is proposed for random scan Gibbs sampling to adaptively update the selection probabilities by learning from the random samples generated along with the chain. Finally, simulation results based on MIMO detection are presented to confirm the performance gain at the expense of limited complexity cost.**

***Index Terms*—Lattice Gaussian sampling, reinforcement learning, Markov chain Monte Carlo, lattice coding and decoding, MIMO detection.**

I. Introduction

Recently, lattice Gaussian distribution has become a common theme in various research fields. In mathematics, Banaszczyk used it to prove the transference theorems for lattices [1]. In coding, it was applied to achieve the full shaping gain for lattice coding [2], and to achieve the capacity of the Gaussian channel and the secrecy capacity of the Gaussian wiretap channel, respectively [3]. In cryptography, lattice Gaussian distribution has already become a central tool in the construction of many primitives [4]. Meanwhile, it also has underpinned the fully-homomorphic encryption for cloud computing [5]. Algorithmically, lattice Gaussian sampling with a suitable variance allows to solve the shortest vector problem (SVP) and the closest vector problem (CVP); for example, it has led to efficient lattice decoding for multi-input multi-output (MIMO) systems [6].

Due to the central role of the lattice Gaussian distribution playing in these fields, its sampling algorithms become an important computational problem. However, it is rather difficult to perform the sampling even from a low-dimensional discrete Gaussian distribution. To this end, Markov chain Monte Carlo (MCMC) methods were introduced as an alternative way, which attempts to sample from lattice Gaussian distribution by building a Markov chain [7]. Typically, after a burn-in stage, which is measured by the *mixing time* in total variance distance, the Markov chain will step into a stationary distribution, where samples from the target distribution can be successfully obtained thereafter. Specifically, Gibbs sampling from MCMC was firstly adopted to lattice Gaussian sampling, which has an exponential convergence rate [8]. On the other hand, the independent Metropolis-Hastings-Klein (IMHK) sampling algorithm is proposed, which not only experiences uniform ergodicity but also entails an accessible convergence rate [9]. In addition, by introducing auxiliary variables into the sampling process, the sliced sampling is able to achieve a better convergence performance than IMHK [10].

In this paper, to improve the sampling performance of MCMC-based lattice Gaussian sampling, the random scan Gibbs sampling is studied. We firstly show that its error function is partially determined by the selection probabilities over sampling components. Then, we introduce reinforcement learning to random scan Gibbs sampling, which optimizes the selection probabilities in the way of learning from the generated samples. Typically, with respect to random scan Gibbs sampling, a whole framework of reinforcement learning is established with specific ingredients such as "state", "action", "policy" and "reward". Overall, our work is a good attempt for MCMC in cooperation with reinforcement learning, where considerable potential can be well exploited by means of learning.

II. Gibbs Sampling for Lattice Gaussian Distribution

Let matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{n \times n}$ consist of n linearly independent column vectors. The n-dimensional lattice Λ generated by $\mathbf{B}$ is defined by

$$\Lambda = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}, \tag{1}$$

where $\mathbf{B}$ is called the lattice basis. We define the Gaussian function centered at $\mathbf{c} \in \mathbb{R}^n$ for standard deviation $\sigma > 0$ as

$$\rho_{\sigma,\mathbf{c}}(\mathbf{z}) = e^{-\frac{\|\mathbf{z}-\mathbf{c}\|^2}{2\sigma^2}}, \tag{2}$$

for all $\mathbf{z} \in \mathbb{R}^n$. When $\mathbf{c}$ or σ are not specified, we assume that they are $\mathbf{0}$ and 1 respectively. Then, the *discrete Gaussian distribution* over Λ is defined as

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{Bx})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}}{\sum_{\mathbf{x}\in\mathbb{Z}^n} e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}} \tag{3}$$

Algorithm 1 Gibbs sampling for lattice Gaussian distribution

Input: $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{x}^0, t_{\text{mix}}(\epsilon)$
Output: $\mathbf{x} \sim D_{\Lambda,\sigma,\mathbf{c}}$

```
for t =1,2, ... do
    randomly choose the index i based on [α_1, ..., α_n]
    sample x_i from P_i(x_i|x_{[-i]}) shown in (4)
    update x with the sampled x_i and let X_t = x
    if t ≥ t_mix(ε) then
        output the state of X_t
    end if
end for
```

for all $\mathbf{x} \in \mathbb{Z}^n$, where $\rho_{\sigma,\mathbf{c}}(\Lambda) \triangleq \sum_{\mathbf{Bx}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{Bx})$ is just a scaling to obtain a probability distribution.

With respect to Gibbs sampling for lattice Gaussian distribution, each coordinate of $\mathbf{x}$ is sampled from the following 1-dimensional conditional distribution

$$P(x_i|\mathbf{x}_{[-i]}) = D_{\Lambda,\sigma,\mathbf{c}}(x_i|\mathbf{x}_{[-i]}) = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}}{\sum_{x_i\in\mathbb{Z}} e^{-\frac{1}{2\sigma^2}\|\mathbf{Bx}-\mathbf{c}\|^2}}, \quad (4)$$

where $1 \le i \le n$ denotes the coordinate index of $\mathbf{x}$, $\mathbf{x}_{[-i]} \triangleq [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]^T$. During this univariate sampling, the other $n-1$ variables contained in $\mathbf{x}_{[-i]}$ are leaving unchanged. There are various scan schemes to proceed the component updating. Among them, random scan is the basic one, which randomly chooses the coordinate index i from a set of selection probabilities $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]$ with $\sum_{i=1}^n \alpha_i = 1$ and $0 < \alpha_i < 1$. From it, the transition probability of the Markov chain in random scan Gibbs sampling is a weighted sum of the full conditional probabilities:

$$P(\mathbf{X}^t, \mathbf{X}^{t+1}) = \sum_{i=1}^{n} \alpha_i P_i(x_i|\mathbf{x}_{[-i]}), \quad (5)$$

where t is the index of Markov moves. Besides random scan, the extension to other scan strategies is possible. For example, systematic scan proceeds the component update in a fixed order (e.g., from x_n to x_1), thus completing a full iteration. In fact, it has been shown in [11] that systematic scan can be viewed as a special case of random scan with uniform selection probabilities $\alpha_i = \frac{1}{n}$, where the mixing times of these two scan schemes do not differ by more than a polynomial factor. Nevertheless, considerable potential does exist in random scan by the sophisticated selection probabilities $\boldsymbol{\alpha}$.

Theorem 1 ([12])**.** *Given the target lattice Gaussian distribution $\pi = D_{\Lambda,\sigma,\mathbf{c}}$, the Markov chain induced by random scan Gibbs sampling converges to the stationary distribution in total variation (TV) distance as $t \to \infty$:*

$$\lim_{t\to\infty} \|P^t(\mathbf{x};\cdot) - \pi\|_{TV} = 0. \quad (6)$$

Algorithm 1 illustrates the random scan Gibbs sampling for lattice Gaussian distribution. The initial Markov state $\mathbf{x}_{\text{initial}}$ can be chosen from the state space $\mathbb{Z}^n$ arbitrarily, while $t_{\text{mix}}(\epsilon)$ denotes the mixing time of the Markov chain.

III. Error Function of Random Scan Gibbs Sampling

From (5), random scan Gibbs sampling is characterized by the selection probabilities $\boldsymbol{\alpha}$, which determines the percentage of visits to a sampling component of $\mathbf{x}$. In order to specify the optimal choice of $\boldsymbol{\alpha}$ for a better sampling performance, a measurement known as *error function* is studied in the following, which offers an intuition about selection probabilities from the point of view of statistic theory.

Suppose interest lies in estimating $\mu = E_\pi(h(\mathbf{x}))$, $\pi = D_{\Lambda,\sigma,\mathbf{c}}$, where $E(\cdot)$ stands for the expectation and $h(\cdot) \in L_0^2(\pi)$. Here, $L^2(\pi)$ is the Hilbert space of square integrable functions with respect to π so that $L_0^2(\pi) \triangleq \{h(\mathbf{x}) : E[h(\mathbf{x})] = 0, \text{Var}[h(\mathbf{x})] < \infty\}$ denotes the subspace of $L^2(\pi)$ consisting of functions with zero mean relative to π. Then the estimation by sampling should be $\widehat{\mu} = \frac{1}{T}\sum_{t=1}^T h(\mathbf{X}^t)$, which would be sufficiently close to μ via the law of large numbers. In order to measure the closeness between $\widehat{\mu}$ and μ, the error function $W(\boldsymbol{\alpha}, h)$ in terms of the mean squared error loss is defined as

$$W(\boldsymbol{\alpha}, h) = E_\pi[\{\widehat{\mu} - \mu\}^2]. \quad (7)$$

Therefore, the optimal selection probabilities $\boldsymbol{\alpha}$ will generate samples such that $\widehat{\mu}$ is as close to μ as possible on average, which results in a minimum value $W(\boldsymbol{\alpha}, h)$. In particular, given $h \in L_0^2(\pi)$, the optimal choice of $\boldsymbol{\alpha}_{\text{opt}}$ can be expressed as

$$\begin{aligned}\boldsymbol{\alpha}_{\text{opt}} &= \arg\min_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}, h) \\ &= \arg\min_{\boldsymbol{\alpha}} \left\{ \lim_{T\to\infty} \text{Var}_\pi\left(\frac{1}{T}\sum_{t=1}^T h(\mathbf{X}^t)\right)\right\} \\ &= \arg\min_{\boldsymbol{\alpha}} \left\{ \text{Var}_\pi(h(\mathbf{x})) + 2\sum_{t=1}^{\infty} \text{cov}_\pi\big(h(\mathbf{X}^0), h(\mathbf{X}^t)\big)\right\} \quad (8)\end{aligned}$$

where $\text{Var}(\cdot)$ and $\text{cov}(\cdot,\cdot)$ represent the variance and the covariance respectively.

However, it is difficult to figure out $\boldsymbol{\alpha}_{\text{opt}}$ who minimizes the error function shown above. Alternatively, we try to minimize the maximum error over the possible function $h \in L_0^2(\pi)$ by optimizing the selection probabilities $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha}_{\text{alt}} = \arg\min_{\boldsymbol{\alpha}} \left\{ \sup_{h\in L_0^2(\pi), \|h\|=1} W(\boldsymbol{\alpha}, h)\right\}, \quad (9)$$

where the requirement of $\|h\| = 1$ (i.e., $\text{Var}(h) = 1$) is added for standard normalization over $h(\cdot)$. Intuitively, compared to $\boldsymbol{\alpha}_{\text{opt}}$ in (8), $\boldsymbol{\alpha}_{\text{alt}}$ is actually a suboptimal solution. As shown in (8), $\text{Var}_\pi(h(\mathbf{X}))$ in $W(\boldsymbol{\alpha}, h)$ is independent of selection probabilities $\boldsymbol{\alpha}$. Besides $\text{Var}_\pi(h(\mathbf{X}))$, $W(\boldsymbol{\alpha}, h)$ is actually a sum of the first t-lag covariances. Hence, the formation of $\boldsymbol{\alpha}_{\text{alt}}$ in (9) can be simplified as

$$\boldsymbol{\alpha}_{\text{alt}} = \arg\min_{\boldsymbol{\alpha}} \left\{ \sup_{h\in L_0^2(\pi), \|h\|=1} \left[\sum_{t=1}^{\infty} \text{cov}_\pi\big(h(\mathbf{X}^0), h(\mathbf{X}^t)\big)\right]\right\}. \quad (10)$$

To further specify the covariance items in (10), we now

invoke the following Lemma in [13].

Lemma 1. *Let $\mathbf{X}^0, \mathbf{X}^1, \ldots$ be samples generated by random scan Gibbs sampling under stationary distribution and $\mathbf{i}$ the random variable representing the index randomly chosen from the selection probabilities $\boldsymbol{\alpha}$. For $h \in L_0^2(\pi)$, then it follows*

$$\mathrm{cov}_\pi(h(\mathbf{X}^0, h(\mathbf{X}^t)) = \mathrm{Var}[E(\cdots E(E(h(\mathbf{x})|\mathbf{i}, \mathbf{x}_{-\mathbf{i}})|\mathbf{x})|\cdots)] \tag{11}$$

with t conditional expectations taken alternatively on $\{\mathbf{i}, \mathbf{x}_{-\mathbf{i}}\}$ and $\mathbf{x}$.

It is noteworthy that the infinite sum asymptotic covariance in (10) is unnecessary in practice. Due to the convergence of Markov moves, the term $\mathrm{cov}_\pi(h(\mathbf{X}^0), h(\mathbf{X}^t))$ decays gradually along with t, where only a few order approximation is sufficient to offer a good metric estimation. Here, the first order is applied and we can easily arrive at the following result

$$\begin{aligned} W(\boldsymbol{\alpha}, h) \approx & \mathrm{Var}_\pi(h(\mathbf{x})) + 2\mathrm{cov}_\pi\left(h(\mathbf{X}^0), h(\mathbf{X}^1)\right) \\ = & \mathrm{Var}_\pi(h(\mathbf{x})) + 2\sum_{i=1}^{n} \alpha_i \mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i}))), \end{aligned} \tag{12}$$

which leads to an approximation of the desired selection probabilities $\boldsymbol{\alpha}$ in (10)

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \left\{ \sup_{h \in L_0^2(\pi), \|h\|=1} \left[\sum_{i=1}^{n} \alpha_i \mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i}))) \right] \right\}. \tag{13}$$

The expression $\boldsymbol{\alpha}^*$ in (13) provides a straightforward way to seek the desired choice of selection probabilities. Specifically, to compute the term $\mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ in (13), the generated samples along with the chain can be employed to obtain the approximations, i.e.,

$$\widehat{\boldsymbol{\alpha}}^* = \arg\min_{\boldsymbol{\alpha}} \left\{ \sup_{h \in L_0^2(\pi), \|h\|=1} \left[\sum_{i=1}^{n} \alpha_i \widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i}))) \right] \right\}. \tag{14}$$

Note that because of

$$\begin{aligned} \mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i}))) = & E_\pi[E^2(h(\mathbf{x})|\mathbf{x}_{-i})] - [E_\pi[E(h(\mathbf{x})|\mathbf{x}_{-i})]]^2 \\ = & E_\pi[E^2(h(\mathbf{x})|\mathbf{x}_{-i})] - [E(h(\mathbf{x}))]^2 \end{aligned} \tag{15}$$

where (15) holds according to the law of total probability for expectation (i.e., $E[E(A|B)] = E(A)$), the approximation $\widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ can be estimated by

$$\widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i}))) = \widehat{E}_\pi[E^2(h(\mathbf{x})\,|\mathbf{x}_{-i})] - [\widehat{E}(h(\mathbf{x}))]^2 \tag{16}$$

in practice.

Since the generation of samples along with the chain is also determined by the choice of selection probabilities $\boldsymbol{\alpha}$, the update of $\boldsymbol{\alpha}$ should be carried out alternately along with the samples rather than be performed only once. Typically, given (14), let $\boldsymbol{\alpha}$ be updated every M iterations while each iteration contains n-times univariate sampling to contribute the samples. Then, based on the samples, $\widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ can be calculated to optimize $\boldsymbol{\alpha}$ by (14). Here, the iteration of random scan Gibbs sampling is comparable to the Markov move of systematic scan Gibbs sampling, who updates all the n components of $\mathbf{x}$ by univariate sampling during a single Markov move.

Theoretically, such a Markov mixing with dynamical updating $\boldsymbol{\alpha}$ corresponds to adaptive MCMC [14], where $\boldsymbol{\alpha}$ gradually converges to $\widehat{\boldsymbol{\alpha}}^*$. Although the underlying chain interrupted by the update of $\boldsymbol{\alpha}$ is not Markov, convergence in total variation norm is still obtained as the adaptive chain is approaching a chain induced by the random scan Gibbs sampling characterized by selection probabilities $\boldsymbol{\alpha}$ [15]. In practice, one can update $\boldsymbol{\alpha}$ for a few times (which can be viewed as a burn-in stage while the convergence to the target distribution is still maintained) and then continue the Markov chain with a fixed $\boldsymbol{\alpha}$.

IV. Reinforcement Learning-Aided Random Scan Gibbs Sampling

In order to effectively determine the optimized $\boldsymbol{\alpha}$, we introduce reinforcement learning into random scan Gibbs sampling, which dynamically updates $\boldsymbol{\alpha}$ in a learning way. As one of the basic machine learning paradigms, reinforcement learning has been widely applied in various research fields, which enables an agent to learn an optimal or near-optimal policy that maximizes the defined "reward function" [16]. Typically, it resorts to a framework which defines the interaction between the agent and the environment in terms of "states", "actions", "rewards", "policy" while the agent learns to achieve the goal in the uncertain and complex environment.

From the reinforcement learning point of view, the samples along with the chain, i.e., $\{\mathbf{X}^{(j-1)M+1}, \ldots, \mathbf{X}^{jM}\}$, $1 \le j \le J$ account for the agent "state", where J is the number of times to update $\boldsymbol{\alpha}$. The sampling operations to generate these samples based on (4) correspond to the "action". Given the state $\{\mathbf{X}^{(j-1)M+1}, \ldots, \mathbf{X}^{jM}\}$, the summation $\sum_{i=1}^{n} \alpha_i \mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ in (13) can be used to describe the "reward" as

$$R = -\sum_{i=1}^{n} \alpha_i \mathrm{Var}_\pi(E(h(\mathbf{x}|\mathbf{x}_{-i}))), \tag{17}$$

which is calculated through the approximation way as

$$\widehat{R} = -\sum_{i=1}^{n} \alpha_i r(i) \tag{18}$$

with $r(i) = \widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$. Clearly, a large summation $\sum_{i=1}^{n} \alpha_i \widehat{\mathrm{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ naturally results in a small reward $\widehat{R}(\boldsymbol{\alpha})$. Finally, the selection probabilities $\boldsymbol{\alpha}$ serve as the "policy", which determines the learning agent's way of behaving at a given time. Over all, this establishes a framework of reinforcement learning for random scan Gibb sampling while the policy $\boldsymbol{\alpha}$ is the learning target of the agent.

On the other hand, given the reward $\widehat{R}$ in (18), the learning mechanism has a strong motivation to assign a large enough value (i.e., $\alpha_i \approx 1$) to the smallest value $r(i)$, so as to make the reward as large as possible. Unfortunately, this breaks the Markov mixing of Gibbs sampling as the univariate sampling

Algorithm 2 Reinforcement learning-aided MCMC algorithm for lattice Gaussian sampling

Input: $\mathbf{B}, \sigma, \mathbf{c}, \mathbf{x}_{\text{initial}}, \tau, M, J, \boldsymbol{\alpha}^0, T$
Output: $\mathbf{x} \sim D_{\Lambda,\sigma,\mathbf{c}}$

1: **for** $j = 1, \ldots, J$ **do**
2: **for** $m = 1 \ldots, M$ **do**
3: **for** $k = 1, \ldots, n$ **do**
4: randomly choose the index i based on $\boldsymbol{\alpha}^{j-1}$
5: sample x_i from $P(x_i|\mathbf{x}_{[-i]})$ shown in (4)
6: **end for**
7: **end for**
8: **for** $i = 1, \ldots, n$ **do**
9: compute $r(i) = \widehat{\text{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ in (16)
10: update $Q^j(i)$ based on (20)
11: **end for**
12: update $\boldsymbol{\alpha}^j$ based on (19)
13: **end for**
14: **for** $t = 1, \ldots, T$ **do**
15: **for** $k = 1, \ldots, n$ **do**
16: randomly choose the index i based on $\boldsymbol{\alpha}^J$
17: sample x_i from $P(x_i|\mathbf{x}_{[-i]})$ shown in (4)
18: **end for**
19: output the state of $\mathbf{X}_t = \mathbf{x}$
20: **end for**

concerns only one coordinate of $\mathbf{x}$, which is harmful to the realization of lattice Gaussian sampling. To this end, although a few choices of i are highly preferred with large values α_i, the other choices of i also should be visited reasonably. Actually, this is similar to the *exploration-exploitation dilemma* in reinforcement learning: the agent has to exploit what it already knows in order to obtain the reward, but it also has to explore to make better action selections in the future.

In order to flexibly adjust the trade-off within $\boldsymbol{\alpha}$, following the *softmax* algorithm in the classic scenario of multi-arm bandit, we use Boltzmann distribution to characterize the policy $\boldsymbol{\alpha}$, i.e.,

$$\alpha_i = \frac{e^{-\frac{Q(i)}{\tau}}}{\sum_{i=1}^{n} e^{-\frac{Q(i)}{\tau}}} \tag{19}$$

with initial selection probabilities $\boldsymbol{\alpha}^0 = [\frac{1}{n}, \ldots, \frac{1}{n}]$. Here, $\tau > 0$ is known as the temperature and a larger τ means a more uniform distribution of $\boldsymbol{\alpha}$, $Q(i)$ is the average reward with respect to coordinate index i. For j-th time to update $\boldsymbol{\alpha}$ (i.e., $\boldsymbol{\alpha}^j$), $Q^j(i)$ updates itself by

$$Q^j(i) = \frac{Q^{j-1}(i) \cdot (j-1) + r(i)}{j} \tag{20}$$

with $Q^0(i) = 0$ for $1 \le i \le n$. According to (19), a small α_i^j is assigned regarding to a large value $\widehat{\text{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$, so as to improve the reward $\widehat{R}$. Then, based on the updated $\boldsymbol{\alpha}^j$, the underlying chain continues and another M iterations are carried out to get samples for updating $Q^{j+1}(i)$. Other allocation schemes about $\boldsymbol{\alpha}$ also work, and the reason we use Boltzmann distribution is due to its flexibility by tuning τ.

To summarize, the proposed reinforcement learning-aided MCMC algorithm for lattice Gaussian sampling is outlined in Algorithm 2. As can be seen, the optimization of $\boldsymbol{\alpha}$ by reinforcement learning from steps 1 to 13 works as a burn-in process for the Markov mixing. After that, from step 14 to 20, random scan Gibbs sampling continues but with a fixed selection probabilities $\boldsymbol{\alpha}^J$. We point out the convergence in total variation norm to the target lattice Gaussian distribution is guaranteed in both these two stages. Overall, the total number of iterations of the proposed reinforcement learning-aided MCMC algorithm is $K = J \cdot M + T$, and it is flexible in choosing M as well as the number of times to update $\boldsymbol{\alpha}$ (i.e., J). The larger M, the more precise of the approximation $\widehat{\text{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$. Accordingly, a large M will lead few changes in the update of $\boldsymbol{\alpha}$, thus leading to a small size of J. Additionaly, restricted classes of function h is also helpful to the efficient computation and here we apply the linear function $h(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i x_i$ with respect to Gaussian variates by following the setup in [17].

V. Simulations

In this section, the performance of the proposed reinforcement learning-aided MCMC algorithm for lattice Gaussian sampling is exemplified in the scenario of the uplink signal detection in MIMO communications.

Specifically, simulation results for an $n \times n$ MIMO system with a square channel matrix containing i.i.d. Gaussian entries are presented. The i-th entry of the transmitted signal $\mathbf{x}$, denoted as x_i, is a modulation symbol taken independently from a Q^2-QAM constellation $\mathcal{X} \in \mathbb{Z}$ with Gray mapping. Meanwhile, it is assumed a flat fading environment, where the channel matrix $\mathbf{H}$ contains uncorrelated complex Gaussian fading gains with unit variance and remains constant over each frame duration. Let E_b represent the average power per bit at the receiver, then $E_b/N_0 = n/(\log_2(M)\sigma_w^2)$ holds where M is the modulation level and σ_w^2 is the noise power. Then, we can construct the system model as

$$\mathbf{c} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{21}$$

and this decoding problem of $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathcal{X}^n} \|\mathbf{c} - \mathbf{H}\mathbf{x}\|^2$ can be solved by sampling over the discrete Gaussian distribution

$$P_{\Lambda(\mathbf{H}),\sigma,\mathbf{c}}(\mathbf{x}) = \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{H}\mathbf{x}-\mathbf{c}\|^2}}{\sum_{\mathbf{x} \in \mathcal{X}^n} e^{-\frac{1}{2\sigma^2}\|\mathbf{H}\mathbf{x}-\mathbf{c}\|^2}} \tag{22}$$

because the optimal solution has the largest probability making it most likely be encountered by sampling. For this reason, we examine the decoding error probabilities to compare the sampling performance of Markov chains.

In Fig. 1, the bit error rates (BERs) of the proposed reinforcement learning-aided MCMC sampling detectors are evaluated against the number of iterations in a 8×8 uncoded MIMO system with 16-QAM. Here, we use LLL reduction-aided SIC decoding serves as a performance baseline for a better comparison, where the trade-off coefficient $1/4 < \eta < 1$

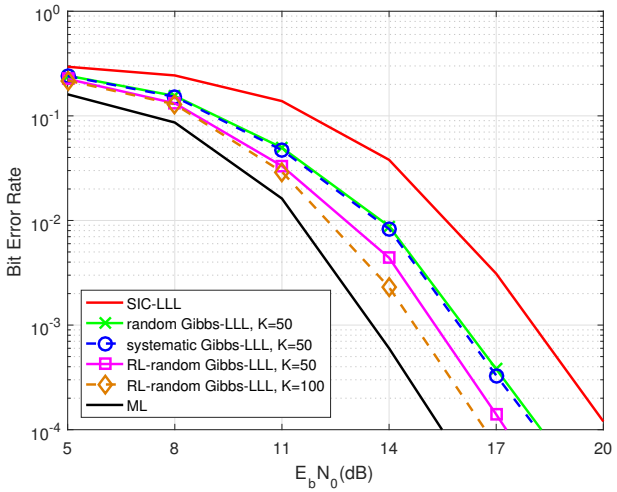

Fig. 1. Bit error rate versus average SNR per bit for the uncoded 8×8 MIMO system using 16-QAM.

in Lovász condition is set as 0.99 for a relatively orthogonal lattice basis. Based on it, the decoding result of SIC-LLL i.e., $\mathbf{x}_{\text{SIC-LLL}}$ serves as the starting point of the following MCMC sampling schemes. On the other hand, the performance of ML decoding, which is realized by sphere decoding (SD) to solve the closest vector problem (CVP) in (21) by enumerations is also given as a baseline. Clearly, there is a substantial performance gap between lattice reduction-aided decoding scheme and sampling decoding schemes. Specifically, with the standard deviation $\sigma = \min_i \|\widehat{\mathbf{h}}_i\| /(2\sqrt{\pi})$, under the same number of iterations $K = 50$, systematic scan Gibbs sampling is comparable to the random scan Gibbs sampling with equal selection probabilities $\boldsymbol{\alpha} = [\frac{1}{n}, \ldots, \frac{1}{n}]$. This is in line with the fact that systematic scan can be viewed as a special case of random scan with even choices of being visited. Although systematic scan is preferred due to easy implementation in hardware, random scan has more potential to be exploited by optimizing the selection probabilities $\boldsymbol{\alpha}$.

Clearly, the proposed reinforcement learning-aided MCMC sampling achieves a better decoding performance than the standard random scan Gibbs sampling under the same iterations. This is because the selection probabilities $\boldsymbol{\alpha}$ are optimized via the criterion of error function, which is implemented according to reinforcement learning by learning from the random samples generated along with the chain. Here, we set $M = 10$, $J = 2$, $T = 30$ for the case of $K = 50$, and $M = 20$, $J = 3$, $T = 40$ for the case of $K = 100$, and $\tau = 1$ for both cases. To calculate $\widehat{\text{Var}}(E(h(\mathbf{x}|\mathbf{x}_{-i})))$ in (16), all the samples generated by the univariate sampling at each Markov moves are employed. As expected, with the increase of Markov moves, the decoding performance improves gradually as the chain is more and more approaching the target distribution. On the other hand, we also observe that the reinforcement learning-aided MCMC sampling is not significantly slower than the standard random scan. This is easy to understand because the former only performs the update of $\boldsymbol{\alpha}$ for J times while other operations except the update are the same with random scan Gibbs sampling. In particular, with $E_bN_0 = 14$dB, the reinforcement learning-aided MCMC sampling takes 0.186 second when selection probabilities $\boldsymbol{\alpha}$ are updated while 0.0151 second are used by the operations of each standard iteration. Considering the limited cost of update, the extra complexity within it turns out to be negligible.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China under Grants No. 61801216, Natural Science Foundation of Jiangsu Province under Grant No. BK20180420, State Key Laboratory of Integrated Services Networks (Xidian University) under Grant No. ISN21-31.

REFERENCES

[1] W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Math. Ann.*, vol. 296, pp. 625–635, 1993.
[2] G. Forney and L.-F. Wei, "Multidimensional constellations–Part II: Voronoi constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 941–958, Aug. 1989.
[3] C. Ling and J.-C. Belfiore, "Achieiving the AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inform. Theory*, vol. 60, no. 10, pp. 5918–5929, Oct. 2014.
[4] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," in *Proc. Ann. Symp. Found. Computer Science*, Rome, Italy, Oct. 2004, pp. 372–381.
[5] C. Gentry, "Fully homomorphic encryption using ideal lattices," *STOC*, pp. 169–178, 2009.
[6] Z. Wang and C. Ling, "Lattice Gaussian sampling by Markov chain Monte Carlo: Bounded distance decoding and trapdoor sampling," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3630–3645, 2019.
[7] Z. Wang, "Markov chain Monte Carlo methods for lattice Gaussian sampling: Convergence analysis and enhancement," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 6711–6724, 2019.
[8] Z. Wang, Y. Huang, and S. Lyu, "Lattice-reduction-aided Gibbs algorithm for lattice Gaussian sampling: Convergence enhancement and decoding optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4342–4356, 2019.
[9] Z. Wang and C. Ling, "On the geometric ergodicity of Metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 738–751, Feb. 2018.
[10] Z. Wang, L. Liu, and C. Ling, "Sliced lattice Gaussian sampling: Convergence improvement and decoding optimization," *IEEE Transactions on Communications*, pp. 1–1, 2020.
[11] B. He, C. D. Sa, I. Mitliagkas, and C. Re, "Scan order in Gibbs sampling: Models in which it matters and bounds on how much," *Neural Information Processing Systems (NIPS)*, pp. 1–9, 2016.
[12] Z. Wang, C. Ling, and G. Hanrot, "Markov chain Monte Carlo algorithms for lattice Gaussian sampling," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, USA, Jun. 2014, pp. 1489–1493.
[13] J. S. Liu, W. H. Wong, and A. Kong, "Covariance structure and convergence rate of the Gibbs sampler with various scans," *J. Roy. Statist. Soc. Series B*, **57**(1): 157-169, 1995.
[14] K. Latuszynski, G. O. Roberts, and J. S. Rosenthal, "Adaptive Gibbs samplers and related MCMC methods," *The Annals of Applied Probability*, vol. 23, no. 1, pp. 66–98, Feb. 2013.
[15] R. A. Levine and G. Casella, "Optimizing random scan Gibbs samplers," *Journal of Multivariate Analysis*, vol. 97, pp. 2071–2100, 2006.
[16] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," The MIT Press, 2015.
[17] R. A. Levine, Z. Yu, W. G. Hanley, and J. J. Nitao, "Implementing random scan Gibbs samplers," in *Proc. Computational Statistics*, 2005, pp. 177–196.