Sample caching Markov chain Monte Carlo approach to boson sampling simulation

Yong Liu 1, Min Xiong 1, Chunqing Wu 1, Dongyang Wang 1, Yingwen Liu 1, Jiangfang Ding 1, Anqi Huang 1, Xiang Fu 1, Xiaogang Qiang 1,2, Ping Xu 1, Mingtang Deng 1, Xuejun Yang 1 and Junjie Wu 1

1 Institute for Quantum Information & State Key Laboratory of High Performance Computing, College of Computer, National University of Defense Technology, Changsha 410073, People’s Republic of China
2 National Innovation Institute of Defense Technology, AMS, 100071 Beijing, People’s Republic of China

E-mail: junjiewu@nudt.edu.cn

Keywords: boson sampling, quantum supremacy, Markov chain Monte Carlo, autocorrelation, sample caching

Abstract
Boson sampling is a promising candidate for quantum supremacy. It requires to sample from a complicated distribution, and is trusted to be intractable on classical computers. Among the various classical sampling methods, the Markov chain Monte Carlo method is an important approach to the simulation and validation of boson sampling. This method however suffers from the severe sample loss issue caused by the autocorrelation of the sample sequence. Addressing this, we propose the sample caching Markov chain Monte Carlo method that eliminates the correlations among the samples, and prevents the sample loss at the meantime, allowing more efficient simulation of boson sampling. Moreover, our method can be used as a general sampling framework that can benefit a wide range of sampling tasks, and is particularly suitable for applications where a large number of samples are taken.

1. Introduction

Demonstrating quantum supremacy is a milestone in quantum computing, representing that quantum devices can outperform the fastest classical hardware on some task [1–4]. Evaluating the demonstration of quantum supremacy should base on the intense competition between the two sides: the developing quantum devices on some selective tasks, and the classical computers running the benchmark for the corresponding tasks to explore the supremacy threshold. The chosen task should be suitable for near-term implementation as well as able to be significantly accelerated by quantum computing.

Such a task is boson sampling [5]. On the one hand, its linear optical implementation merely requires identical bosons (typically photons), linear transformation, and passive detection. On the other hand, its classical simulation involves computing permanents of Gaussian complex matrices [6], which is likely to be classically intractable, even in approximation cases [7]. The classical hardness of boson sampling attracts enormous efforts to build large-scale physical devices to beat classical computers, and remarkable achievements have been made [8–18]. It is promising to show quantum advantage via boson sampling.

On the classical side, various sampling approaches have been raised for simulating boson sampling. The direct computation of the whole distribution of boson sampling is restricted within small scales because of the exponential growth of the state space. This stimulates a variety of methods for sampling from an exponentially large state space by calculating a few number of probabilities, such as the Clifford–Clifford (C&C) sampling method [19], the rejection sampling method [20] and the Metropolised independent sampling (MIS) method [21, 22]. Among the various sampling approaches, the MIS method allows to generate an effective sample by evaluating only a constant number of probabilities. More importantly, it can serve as a general sampling framework for far-ranging applications.
However, similar with the optical implementation of boson sampling, the sample loss is also a serious problem of classical samplers, as shown in figure 1(a). Current sampling methods generate massive candidate samples with each involving the evaluation of a probability, but only a small fraction of candidates are kept as effective samples, as shown in figure 1(b). For the MIS method, it has to discard massive candidates in order to tackle the autocorrelation issue inherited from the Markov chain Monte Carlo (MCMC) method. To this point, we define the computational hardness limit of the sampling tasks as the complexity of computing one probability. As for boson sampling, it is the complexity of computing one permanent [5], and to reach this limit, it has to eliminate the overhead of the sampling methods. In this work, we propose a method that could reach this limit.

Our method, namely sample caching Markov chain Monte Carlo (SC-MCMC), makes contributions on two sides. (1) On the classical side, the SC-MCMC sampler provides a general sampling framework which allows to generate one effective sample from averagely one candidate sample. SC-MCMC deals with the autocorrelation issue of MCMC following a similar manner to MIS, but caches the discarded candidate samples for further reuse. In this way, the SC-MCMC sampler avoids the sample loss, and allows much more efficient sampling than a MIS sampler. Particularly, the SC-MCMC method can be used for quantum-enhanced applications based on boson sampling such as simulating the molecular vibronic spectra [23], finding dense subgraphs [24] or approximate optimization [25]. Moreover, Our method can also be directly applied on other quantum supremacy candidates, such as the Gaussian boson sampling [26], the IQP circuit sampling [27, 28] and random quantum circuit (RQC) sampling [29, 30], especially when a lot of samples are taken for validating the device [31]. (2) On the quantum side, our results suggest that the hardness of demonstrating quantum supremacy by boson sampling is increasing with the development of classical simulator. Based on the results of the simulation, we emphasize the importance of avoiding the photon loss in the optical implementation of boson sampling, showing that an improvement of 5% for the loss rate would spare hundreds of photons.

2. Boson sampling and Markov chain Monte Carlo

The task of boson sampling is to sample the output distributions of an \( m \)-mode interferometer network with \( n \) identical photons as input. Because of the interferences of photons, the probability of a certain output pattern of photons is related to the permanent of the transformation matrix decided by the interferometer network. Specifically, the output patterns are post-selected within the ‘collision-free’ regime where photons are non-bunching in each output port. The value of \( m \) is often chosen to be \( n^2 \) to meet the requirement for the hardness proof of boson sampling, leading to that the number of all the possible output patterns grows exponentially with \( n \). In summary, the task of classically simulating boson sampling is to generate samples from the probability distribution over the \( \binom{n}{m} \) output patterns.

Instead of calculating the probabilities of the whole distribution, a feasible method is Metropolis–Hastings algorithm, a kind of MCMC method [32]. The sampling is processed by constructing a Markov chain, with the state space corresponding to the set of the \( \binom{n}{m} \) possible output patterns of boson sampling, and the probabilities of all the patterns as the stationary probabilities. To expand the chain, a state \( s' \) is chosen as the candidate state (current state is \( s \)) following a easy-to-sample symmetric proposal probability distribution, and this candidate
state would be accepted and added on the chain with probability

\[
P_{\text{accept}} = \min(1, \frac{p(s')}{p(s)}),
\]

where \( p(s) \) is the boson sampling probability of state \( s \), or be rejected with the rest probability. If the candidate is rejected, the current state would duplicate on the end of the chain. Each time a node is added in the chain, the pattern corresponding to the added node is outputted as a candidate sample. In this way, it seems ideal that one may just need to evaluating one probability for one sample.

Unfortunately, the generated samples may be erroneous because the generated sample sequence suffers from severe autocorrelation, as shown in figure 2(a). The first-order autocorrelation of a sequence is often estimated using Durbin–Watson statistic \([33]\), while in this paper, we follow a more straightforward way to estimate the autocorrelation by

\[
\eta_1 = \frac{\sum_i (x_i - \bar{x})(x_{i+1} - \bar{x})}{\sum_i (x_i - \bar{x})^2},
\]

where \( x_i \) is the value of the \( i \)th sample, \( \bar{x} \) is the mean of the samples. Each \( x_i \) is assigned a value as the order of patterns after been sorted. The value of \( \eta_1 \) should be in \([-1, 1]\), and it reflects the autocorrelation in the sequence. The closer \( |\eta_1| \) is to 1, the stronger the samples are correlated, and the sign of \( \eta_1 \) indicates if the samples are positively or negatively correlated.

To overcome the autocorrelation of the sample sequence, in MIS, a method called ‘jump sampling’ (or ‘thinning procedure’) is applied to obtain independent samples, as shown in figure 2(b). In jump sampling, an effective sample is kept in every \( K \) candidates with the rest \( K - 1 \) candidates discarded. The remained samples are approximately independent. The value of \( K \) is determined by checking the autocorrelation of the obtained sequence. In MIS, the value of \( K \) is 100, which is claimed to be sufficient for simulating boson sampling of more than 30 photons \([21]\). However, this suggests that in MIS the ratio for keeping a candidate sample is \( \eta_k = \frac{1}{K} = 1\% \). Other methods such as the delayed rejection \([34–36]\) could be used to reduce the correlations among the samples at the cost of calculating more probabilities for one sample. If the autocorrelation issue could be tackled without abandoning any samples, the sampling process could be accelerated by 100 times.

3. The sample caching method

The main scheme of the SC-MCMC protocol is shown in figure 2(c). It mainly contains two parts: one is a MCMC sampler providing candidate samples, with which the second part combined is a procedure that we call ‘sample caching’. In figure 2(d) we show the working processes of the SC-MCMC sampler, which can be divided into three phases:

\[\text{Figure 2. Scheme of jump sampling and sample caching. (a) The original sequence generated by MCMC sampler, where the samples are strongly correlated. (b) The jump sampling method. The samples within the leap are all discarded. The remained samples are approximately uncorrelated. (c) The sample caching method. Samples are filled in a cache first, and are afterwards randomly outputted from the cache. The output samples are approximately independent. (d) The three phases of SC-MCMC. After outputting the first sample, SC-MCMC enters the cache filling phase (CFP), in which it performs similar to MIS. After the cache is full, SC-MCMC enters the working phase (WP) and starts to output samples by every evaluation of a probability. Finally, in the cache cleaning phase (CCP), there can be a burst of \( L \) samples that can be the compensation for the delay in CFP. Averagely, only one probability is evaluated for one effective sample in the SC-MCMC sampler.} \]
1. The cache filling phase (CFP). In this phase, the SC-MCMC sampler follows the similar protocol to the MIS sampler: It outputs an effective sample in every $K$ candidates, where $K$ is the jump step used in the MIS sampler. The difference is that the unselected candidates are used to fill a sample cache till the cache is full. The sample cache can be completely filled after generating $\lceil L/(K-1) \rceil + L$ candidate samples, where $L$ is the size of the cache. In this stage it has to calculate $K$ probabilities for one effective sample;

2. The working phase (WP), in which the cache is full. For each time a candidate sample is generated by the MCMC process, randomly output a sample from the cache first and then store this new candidate into the cache. This procedure is repeated till the MCMC process generates $N$ candidates. Therefore in WP, the SC-MCMC approach drives efficient sample generation that an effective sample can be generated by calculating only one probability;

3. The cache cleaning phase (CCP). In this phase, the samples stored in the cache are outputted in random order without calculating any probabilities, and therefore there can be a burst of sample generation. This phase can be regarded as a compensation for the initial overhead of CFP, and result in that averagely the evaluation of every probability can support the generation of one effective sample.

Nevertheless, the first candidate sample can be directly outputted as an effective sample since there is no other sample that can be correlated with. As reflected from figure 2(d), if only a small number of samples are taken (i.e. $N < \lceil L/(K-1) \rceil + L$), the sampling process may stop in CFP, in this case the SC-MCMC sampler is actually executing the MIS protocol. Generally, in the final CCP, cleaning the cache provides a burst of sample generation. However, this step can be dangerous when the value of $N$ is not large enough because the burst output could result in sample correlation, and whether a sample can be kept is determined by how much it is correlated with the sample sequence. While if a large number of samples are taken, SC-MCMC can be $K$ times as fast as MIS.

It is worth noting that the generation of candidate samples discussed here is slightly different from that in [21] where the proposal probability distribution is chosen as the distribution of distinguishable particles corresponding to the boson sampling instance. This choice can reduce the correlations between samples, but introduces extra computational cost of a real permanent. Here we choose the uniform distribution as the proposal distribution so that only one permanent is required to calculate for every candidate sample (as we will show in the following discussion). In this way, compared with the C&C sampler which has an approximately equivalent cost of calculating two $n \times n$ permanents for one effective sample [19], the SC-MCMC sampler can modestly obtain a two-fold improvement over the C&C sampler when a large number of samples are taken.

Similar to MIS, SC-MCMC is also case independent. Besides boson sampling, SC-MCMC can be directly used as a general sampling framework for any sampling tasks as long as we can evaluate the probabilities for given events, for example the Gaussian boson sampling [26], IQP circuit sampling [27, 28] and random quantum circuit sampling [29, 30]. Meanwhile, SC-MCMC is particularly suitable for tasks where a large number of samples are taken.

Essentially, the correctness of SC-MCMC is guaranteed by the MCMC process. Some methods can be applied to validate the sampling results [37–39], but a more straightforward way is to compare the frequency graph with the probability distribution. We found an empirical result that the number of samples may has to be in the order of $100 \cdot {\binom{n}{m}}$ to construct a frequency graph with a similarity of 99% with the probability distribution.

More importantly, the sample caching process eliminates the autocorrelations within the sample sequence without losing any candidate samples. Under the asymptotic condition where the size of the cache is large enough, the number of samples stored in the cache follows the probability of the samples, and the uniformly random choice makes it independent among the draws. Practically, the correlations among the samples can be eliminated with a sample cache in a limited size. In figure 3(a) we plot the first-order autocorrelation against the size of sample cache for different scales of boson sampling. In figure 3(b) we plot the autocorrelation of the sample sequences obtained from different approaches in simulating an instance of boson sampling with 30 photons and 900 modes, where the ‘lag’ indicates the orders of the autocorrelations. In the sample sequence from the original MCMC approach, we can still find remained correlation between two samples even in lag 100 owing to the improper choice of the proposal probability distribution (see supplementary material for details, available online at stacks.iop.org/NJP/22/033022/mmedia). In comparison, the SC-MCMC sampler with $L = 1000$ can reduce the autocorrelation at every order to a quite low level, and a cache with $L = 4000$ can approximately generate independent samples, even though the proposal distributions in the SC-MCMC and the MCMC samplers are the same.

To understand how sample cache works, we analyze the jump sampling method first. For the MCMC sampler with state space $S = \{s_1, \ldots, s_N\}$, the transition probabilities can be described by a matrix $P = \{p_{ij}\}_{N \times N}$.
with \( p_{ij} \) equals the probability to transit from state \( s_i \) to \( s_j \) in one step. The \( k \)-step transition probability matrix is \( P^{(k)} = P^k \), and \( p^{(k)}_{ij} \) is the probability to transit from \( s_i \) to \( s_j \) in \( k \) steps. It satisfies that \( \lim_{k \to \infty} p^{(k)}_{ij} = p(s_i) \) for arbitrary \( i \) and \( j \), where \( p(s_i) \) is the probability of state \( s_i \) \([22, 40, 41]\). Thus a sample is independent from another that is infinite steps away. Actually, in a certain number of steps (e.g. \( K \) steps), the \( K \)-step transition probability \( p^{(K)}_{ij} \) approximately equals to \( p(s_i) \) for arbitrary \( i \) and \( j \), which means that a sample is approximately independent from the samples with a distance of more than \( K \) steps. The value of \( K \) depends on how fast the Markov chain converges \([42]\). Empirically, we can choose a sufficiently large value. If the samples between the \( K \) steps are discarded, the correlations among the remained samples are negligible.

In SC-MCMC, the result can be regarded as a reordered sequence from the original one. The probability that the two adjacent samples are at a distance of \( k (k \geq 1) \) steps in the original sequence is

\[
p(k, L) = \left( \frac{L - 1}{L} \right)^{k-1} \cdot \frac{1}{L},
\]

where \( L \) is the size of the sample cache. Then we can obtain the probability that two adjacent samples are still correlated (the distance between the two samples in the original sequence does not exceed \( K \), the corresponding jumping step in jump sampling) by

\[
P_{cr} = \sum_{k=1}^{K} p(k, L) = 1 - \left( \frac{L - 1}{L} \right)^K \approx \frac{K}{L}.
\]

Clearly, \( \lim_{L \to \infty} P_{cr} = 0 \), and the first-order autocorrelation is thus eliminated if the sample cache is sufficiently large. The autocorrelations of higher orders are eliminated in the same way (see supplementary information).

The next question is that practically what size should the cache be? The exact size of the sample cache required varies case-by-case, however we can choose a sufficiently big cache. This answer is easy to obtain from Eq.(4) by ensuring \( P_{cr} < \epsilon \), then \( L > \frac{K}{\epsilon} \). Setting \( K = 200 \) and \( \epsilon = 0.05 \), resulting in \( L = 4000 \), is sufficient for the case of \( n = 30 \) and \( m = 900 \). It is supposed to be sufficient for boson sampling with larger scale, and the size of cache could be further enlarged if needed. Most significant of all, no sample is lost no matter what size the cache is, which suggests that our method reaches a time complexity of simulating boson sampling to \( O(n2^n) \).

4. Numerical simulation

We demonstrate our method for boson sampling simulation by implementing the parallelized Glynn’s algorithm to exactly compute the permanents of arbitrary square matrices \([43]\). The numerical simulations were done on 64 nodes of Tianhe-2 supercomputer \([44, 45]\) or another local cluster, as shown in table 1. The sampling rate reached 1.01 Hz on Tianhe-2 nodes when simulating boson sampling with 30 photons, and the 32 cluster...
nodes afforded the simulation for 21 photons with a sampling rate of 18.09 Hz. The percentage of the time used for calculating permanents exceeds 99% when \( n \) is sufficiently large.

Combined with the performance estimation on Tianhe-2 supercomputer [46], the average time estimated (in seconds) for obtaining a \( n \)-photon candidate sample is

\[
\text{time} = \frac{-1.9925}{n^2 10^{-11}} \cdot \text{if we are allowed to simulate the generation of a large number of samples.}
\]

This scaling result indicates that the average time estimated for a 50-photon sample can be reduced from about 10 d to within 100 min. To compare the performance of the simulator with the physical setups, we use the function defined in [21]

\[
QA(n, \eta) = log (t_c / t_q),
\]

where \( t_c \) is the estimated classical runtime for a boson sampling instance with \( n \) photons and \( m = n^2 \) modes. \( t_q \propto (R_q \eta^n)^{-1} \) is the quantum runtime for the corresponding instance in which \( R_q \) is the \( n \)-photon repetition rate of the photon source, and \( \eta \) is the transmission probability for a photon (see supplementary information). \( QA > 0 \) suggests that the performance of a quantum boson sampling setup surpasses that of Tianhe-2 supercomputer.

In figure 4 we plot four lines corresponding to \( QA(n, \eta) = 0 \) under different combinations of classical approaches and physical setups. Associated with the transmission probability realized in recent experiments (<0.4 [8–11, 13, 16, 18]), the increment of photon number required to reach a corresponding performance is vast. For example, when \( \eta = 0.5 \) and \( R_q = 10 \text{ GHz} \) (probably beyond the reach of near-term experiments) the increment exceeds 300 photons, while it reduces to 30 when \( \eta \) is improved to 0.55. Further, for a network with given shape, there exist the minimum request for \( \eta \) even when we have sufficiently many photons (see supplementary information). Our results suggest that currently, enhancing the transmission probability can greatly reduce the required photon number for quantum advantage.

### Table 1

| Scale     | \( r_5 \) | \( r_{200} \) | \( r_{500} \) | \( r_{1000} \) | \( r_{2000} \) | \( r_{4000} \) |
|-----------|-----------|--------------|-------------|-------------|-------------|-------------|
| 9p.81m    | 0.9115    | 0.0102       | 0.0261      | 0.0119      | 0.0067      | 0.0035      |
| 15p.225m  | 0.9490    | 0.0128       | 0.0451      | 0.0205      | 0.0102      | 0.0057      |
| 21p.441m  | 0.9633    | 0.0030       | 0.0622      | 0.0311      | 0.0162      | 0.0081      |
| 25p.625m  | 0.9673    | -0.0183      | 0.0642      | 0.0328      | 0.0167      | 0.0089      |
| 30p.900m  | 0.9702    | -0.0832      | 0.0684      | 0.0309      | 0.0095      | -0.0162     |

![Figure 4](image-url)
5. Conclusion

In summary, we proposed a classical sampling approach which can be used to simulate boson sampling. Our method tackles the autocorrelation issue which leads to the sample loss in MIS, and can be 100 times as fast as MIS if a large number of samples are taken. For cases where only a small number of samples are required, our method can perform as well as MIS. Therefore our method enables more efficient simulation of boson sampling, especially in the cases where boson sampling is applied for quantum-enhanced problems. Above all, our method is a general sampling framework and could support customized optimization for specific tasks to gain further advanced efficiency.

The progress in classical simulators can be challenges to physical implementations, which however could provide some instructive feedback on physical experiments. The comparison between classical computing and quantum boson sampling suggests that currently, reducing photon loss may approach closer to quantum advantage, and is perhaps more important than merely enlarging the scale of boson sampling devices.

Acknowledgments

We gratefully acknowledge the help from China Greatwall Technology, and the National Supercomputer Center in Guangzhou. We appreciate the helpful discussion with other members of QUANTA group. JW acknowledges the support from National Natural Science Foundation of China under Grants No. 11621091 and No. 11690031. XQ acknowledges support from National Natural Science Foundation of China under Grants No. 11804389.

ORCID iDs

Yong Liu  https://orcid.org/0000-0003-0523-6816
Xiang Fu  https://orcid.org/0000-0002-1365-8501
Junjie Wu  https://orcid.org/0000-0001-5951-8988

References

[1] Preskill J 2012 Quantum computing and the entanglement frontier arXiv:1203.5813
[2] Arute F et al 2019 Quantum supremacy using a programmable superconducting processor Nature 574 505–10
[3] Spagnolo N and Sciarrino F 2018 The race for quantum supremacy: pushing the classical limit for photonic hardware Natl Sci. Rev. 6 2–3
[4] Nielsen M A and Chuang I 2002 Quantum computation and quantum information Am. J. Phys. 70 558–9
[5] Aaronson S and Arkhipov A 2011 The computational complexity of linear optics Proc. 43rd Annual ACM Symposium on Theory of Computing (New York: ACM) pp 333–42
[6] Schelle S and Buhmann S Y 2008 1–133 macroscopic quantum electrodynamics-concepts and applications
[7] Valiant L G 1979 The complexity of computing the permanent Theor. Comput. Sci. 8 189–201
[8] Spring J B et al 2013 Boson sampling on a photonic chip Science 339 798–801
[9] Broome M A, Fedrizzi A, Rahimi-Keshari S, Dove J, Aaronson S, Ralph T C and White A G 2013 Photonic boson sampling in a tunable circuit Science 339 794–8
[10] Tillmann M, Dukic B, Heilmann R, Nolte S, Szameit A and Walther P 2013 Experimental boson sampling Nat. Photon. 7 540–4
[11] Spagnolo N et al 2014 Experimental validation of photonic boson sampling Nat. Photon. 8 615–20
[12] Lund A P, Laing A, Rahimi-Keshari S, Rudolph T, O’Brien J L and Ralph T C 2014 Boson sampling from a gaussian state Phys. Rev. Lett. 113 100502
[13] Bentivegna M et al 2013 Experimental scattershot boson sampling Sci. Adv. 1 e1400255
[14] Seshadreesan K P, Olson J P, Motes K R, Rohde P P and Dowling J P 2015 Boson sampling with displaced single-photon fock states versus single-photon-added coherent states: the quantum–classical divide and computational-complexity transitions in linear optics Phys. Rev. A 91 022334
[15] Latmiral L, Spagnolo N and Sciarrino F 2016 Towards quantum supremacy with lossy scattershot boson sampling New J. Phys. 18 113008
[16] Zhong H et al 2018 12–photon entanglement and scalable scattershot boson sampling with optimal entangled–photon pairs from parametric down–conversion Phys. Rev. Lett. 121 250505
[17] Paesani S, Ding Y, Santagati R, Chakhmakhyhyan L, Vigliar C, Rottwitt K, Oxenløwe L K, Wang J, Thompson M G and Laing A 2019 Generation and sampling of quantum states of light in a silicon chip Nat. Phys. 15 925–9
[18] Wang H et al 2019 Boson sampling with 20 input photons and a 60–mode interferometer in a 1014-dimensional hilbert space Phys. Rev. Lett. 123 250503
[19] Clifford P and Clifford R 2018 The classical complexity of boson sampling Proc. 29th Annual ACM–SIAM Symp. on Discrete Algorithms, SODA 2018 (New Orleans, LA, USA, 7–10 January 2018) pp 146–55
[20] Villalonga B, Lyakh D, Boixo S, Neven H, Humble T S, Biswas R, Rieffel E G, Ho A and Mandrà S 2019 Establishing the quantum supremacy frontier with a 281 p p/s simulation arXiv:1905.00444
[21] Neville A, Sparrow C, Clifford R, Johnston E, Birchall P M, Montanaro A and Laing A 2017 Classical boson sampling algorithms with superior performance to near-term experiments Nat. Phys. 13 1153–7
[22] Liu J S 1996 Metropolized independent sampling with comparisons to rejection sampling and importance sampling Stat. Comput. 6 113–9
[23] Huh J, Guerreschi G G, Peropadre B, McClean J R and Aspuru-Guzik A 2015 Boson sampling for molecular vibronic spectra Nat. Photon. 9 615–20
[24] Arrazola J M and Bromley T R 2018 Using gaussian boson sampling to find dense subgraphs Phys. Rev. Lett. 121 030503
[25] Arrazola J M, Bromley T R and Rebentrost P 2018 Quantum approximate optimization with gaussian boson sampling Phys. Rev. A 98 012322
[26] Hamilton C S, Kruse R, Sansoni L, Barkhofen S, Silberhorn C and Jex I 2017 Gaussian boson sampling Phys. Rev. Lett. 119 170501
[27] Bremner M J, Montanaro A and Shepherd D J 2016 Average case complexity versus approximate simulation of commuting quantum computations Phys. Rev. Lett. 117 080501
[28] Shepherd D and Bremner M J 2009 Temporally unstructured quantum computation Proc. R. Soc. A 465 1413–39
[29] Boixo S, Isakov S V, Smelyanskiy V N, Babbush R, Ding N, Jiang Z, Bremner M J, Martinis J M and Neven H 2018 Characterizing quantum supremacy in near-term devices Nat. Phys. 14 595–600
[30] Guo C et al 2019 General-purpose quantum circuit simulator with projected entangled-pair states and the quantum supremacy frontier Phys. Rev. Lett. 123 190501
[31] Hangleiter D, Kliesch M, Eisert J and Gogolin C 2019 Sample complexity of device-independently certified ‘quantum supremacy’ Phys. Rev. Lett. 122 210502
[32] Chib S and Greenberg E 1995 Understanding the metropolis-hastings algorithm Am. Stat. 49 327–35
[33] Durbin J and Watson G S 1971 Testing for serial correlation in least squares regression: III Biometrika 58 1–19
[34] Au S-K and Beck J L 2001 Estimation of small failure probabilities in high dimensions by subset simulation Probab. Eng. Mech. 16 263–77
[35] Tierney L and Mira A 1999 Some adaptive monte carlo methods for bayesian inference Stat. Med. 18 2507–15
[36] Zuev K M and Katafygiotis L S 2011 Modified metropolis-hastings algorithm with delayed rejection Probab. Eng. Mech. 26 405–12
[37] Aaronson S and Arkhipov A 2014 Bosonsampling is far from uniform Quantum Inf. Comput. 14 1383–423
[38] Baireviogna M et al 2014 Bayesian approach to boson sampling validation Int. J. Quantum Inf. 12 1560028
[39] Agresti I, Viggianiello N, Flamini F, Spagnolo N, Crespi A, Osellame R, Wiebe N and Sciarrino F 2019 Pattern recognition techniques for boson sampling validation Phys. Rev. X 9 011013
[40] Paul A 2017 Markov Chains: From Theory to Implementation and Experimentation 1st edn (New York: Wiley)
[41] Wan L, Lou W, Abner E and Krysio R J 2016 A comparison of time-homogeneous markov chain and markov process multi-state models Commun. Stat.: Case Stud., Data Anal. Appl. 292–100
[42] Schmitt F, Schmitt F, Rothlauf F and Rothlauf F 2001 On the importance of the second largest eigenvalue on the convergence rate of genetic algorithms Proc. 14th Symp. on Reliable Distributed Systems
[43] Glynn D G 2010 The permanent of a square matrix Eur. J. Comb. 31 1887–91
[44] Liao X, Xiao L, Yang C and Lu Y 2014 MilkyWay-2 supercomputer: system and application Frontiers Comput. Sci. 8 345–56
[45] Liao X 2014 MilkyWay-2: back to the world Top 1 Frontiers Comput. Sci. 8 343–4
[46] Wu J, Liu Y, Zhang B, Jin X, Wang Y, Wang H and Yang X 2018 A benchmark test of boson sampling on Tianhe-2 supercomputer Natl Sci. Rev. 5 715–20