The Classical Complexity of Boson Sampling

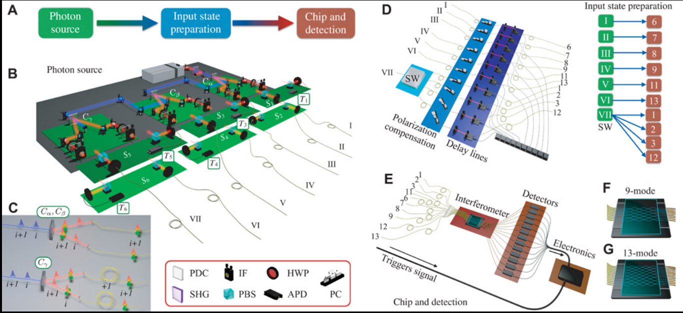
Peter Clifford[1] Raphaël Clifford[2]

[1]University of Oxford

[2]University of Bristol, UK

SODA 2018. Implementation: “Boson Sampling CRAN"

Boson Sampling



Science advances

- Introduced in 2011 as a route to establishing quantum computational supremacy.
- Classically intractable.
- Quantumly tractable.

Boson Sampling - mathematically

Consider an m by n matrix M as the first n columns of a Haar random m by m unitary.

- Sample $n \times n$ matrices A from M, each with probability.

$$\frac{|\operatorname{Per} A|^2}{\prod_{j=1}^{m} s_j!}, \; s_j \text{ is number of copies of the } j\text{th row of } M$$

Boson Sampling - mathematically

Consider an m by n matrix M as the first n columns of a Haar random m by m unitary.

- Sample $n \times n$ matrices A from M, each with probability.

$$\frac{|\operatorname{Per} A|^2}{\prod_{j=1}^{m} s_j!}, \; s_j \text{ is number of copies of the } j\text{th row of } M$$

$$M = \begin{pmatrix} M_{0,0} & M_{0,1} & M_{0,2} \\ M_{1,0} & M_{1,1} & M_{1,2} \\ M_{2,0} & M_{2,1} & M_{2,2} \\ M_{3,0} & M_{3,1} & M_{3,2} \\ M_{4,0} & M_{4,1} & M_{4,2} \\ M_{5,0} & M_{5,1} & M_{5,2} \\ M_{6,0} & M_{6,1} & M_{6,2} \\ M_{7,0} & M_{7,1} & M_{7,2} \\ M_{8,0} & M_{8,1} & M_{8,2} \end{pmatrix} \begin{matrix} \\ \times 2 \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

Boson Sampling - mathematically

Consider an m by n matrix M as the first n columns of a Haar random m by m unitary.

- Sample $n \times n$ matrices A from M, each with probability.

$$\frac{|\operatorname{Per} A|^2}{\prod_{j=1}^{m} s_j!}, \ s_j \text{ is number of copies of the } j\text{th row of } M$$

$$M = \begin{pmatrix} M_{0,0} & M_{0,1} & M_{0,2} \\ M_{1,0} & M_{1,1} & M_{1,2} \\ M_{2,0} & M_{2,1} & M_{2,2} \\ M_{3,0} & M_{3,1} & M_{3,2} \\ M_{4,0} & M_{4,1} & M_{4,2} \\ M_{5,0} & M_{5,1} & M_{5,2} \\ M_{6,0} & M_{6,1} & M_{6,2} \\ M_{7,0} & M_{7,1} & M_{7,2} \\ M_{8,0} & M_{8,1} & M_{8,2} \end{pmatrix} \begin{array}{l} \\ \times 2 \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \qquad A = \begin{pmatrix} M_{1,0} & M_{1,1} & M_{1,2} \\ M_{1,0} & M_{1,1} & M_{1,2} \\ M_{6,0} & M_{6,1} & M_{6,2} \end{pmatrix}$$

Boson Sampling - mathematically

Consider an m by n matrix M as the first n columns of a Haar random m by m unitary.

- Sample $n \times n$ matrices A from M, each with probability.

$$\frac{|\operatorname{Per} A|^2}{\prod_{j=1}^{m} s_j!}, \; s_j \text{ is number of copies of the } j\text{th row of } M$$

$$\binom{n+m-1}{n} \approx e^n (m/n)^n n^{-1/2})$$

different possibles matrices A.

$$A = \begin{pmatrix} M_{1,0} & M_{1,1} & M_{1,2} \\ M_{1,0} & M_{1,1} & M_{1,2} \\ M_{6,0} & M_{6,1} & M_{6,2} \end{pmatrix}$$

$$\binom{8^2+8-1}{8} \approx 2^{33}$$

Permanents are expensive.

The Classical Complexity of Boson Sampling

Aaronson and Arkhipov (2011):

- Exact sampling in poly time not possible unless the polynomial hierarchy collapses to the third level (not much more likely than proving NP = P).

- Approximate sampling in poly time conjectured to be hard for $m \geq n^5$ and "collision free". They further suggest $m \geq n^2$ should be hard classically.

The Classical Complexity of Boson Sampling

Aaronson and Arkhipov (2011):

- Exact sampling in poly time not possible unless the polynomial hierarchy collapses to the third level (not much more likely than proving NP = P).
- Approximate sampling in poly time conjectured to be hard for $m \geq n^5$ and “collision free". They further suggest $m \geq n^2$ should be hard classically.

“If one could implement our experiment with (say) $20 \leq n \leq 30$, then [..] such an experiment would arguably constitute the strongest evidence against the ECT [Extended Church-Turing Thesis] to date.”

— Aaronson and Arkhipov

The Classical Complexity of Boson Sampling

Aaronson and Arkhipov (2011):

- Exact sampling in poly time not possible unless the polynomial hierarchy collapses to the third level (not much more likely than proving NP = P).

- Approximate sampling in poly time conjectured to be hard for $m \geq n^5$ and “collision free". They further suggest $m \geq n^2$ should be hard classically.

~~“If one could implement our experiment with (say) $20 \leq n \leq 30$, then [..] such an experiment would arguably constitute the strongest evidence against the ECT [Extended Church-Turing Thesis] to date.”~~

— Aaronson and Arkhipov

Previous expectations

If one could implement our experiment with (say) $20 \leq n \leq 30$, then certainly a classical computer could verify the answers—but at the same time, one would be getting direct evidence that a quantum computer could efficiently solve an "interestingly difficult" problem, one for which the best-known classical algorithms require many millions of operations. While *disproving* the Extended Church-Turing Thesis is formally impossible, such an experiment would arguably constitute the strongest evidence against the ECT to date.

Aaronson & Arkhipov, arXiv:1011.3245 (2010)

the first steps. The eventual goal would be to demonstrate BosonSampling with (say) $n = 20$ or $n = 30$ photons: a regime where the quantum experiment probably *would* outperform its fastest classical simulation, if not by an astronomical amount. In our view, this would be an exciting proof-of-principle for quantum computation.

Aaronson & Arkhipov, arXiv:1309.7460 (2013)

though, this linear optics experiment is still not at all easy — to reach the regime where digital simulation is currently infeasible one should detect a coincidence of about 30 photons, whose paths through the interferometer can interfere. Further-

Preskill, arXiv:1203.5813 (2012)

extending to N of order 20 with currently available coherence times, clearly growing beyond the capabilities of modern classical supercomputing. We note that the fidelity will not

Goldstein et al., Phys. Rev. B 95 (2017)

input modes [3–11]. However, it remains a challenge to scale up the devices to 20–30 photons [1] traversing a correspondingly large network, a regime in which a quantum boson sampling machine is expected to outperform classical computers.

Barkhofen et al., Phys. Rev. Lett 118 (2017)

cavities decoherence time. The final theoretical result leads to a significant improvement in the efficiency and an additional step towards quantum supremacy which can be achieved with a 7 photons in 50 modes experiment.

Latmiral et al., New J. Phys 18 (2016)

Computing the permanent

Similar to the determinant but much slower to compute.

$$\mathrm{Per}(A) = \sum_{\sigma \in \pi_n} \prod_{i=1}^{n} A_{i,\sigma(i)}$$

where π_n is the set of all permutations of $1, \ldots, n$.

Problem is #P-hard (Valiant '79) and fastest algorithm takes $O(n2^n)$ time.

Boson sampling distribution

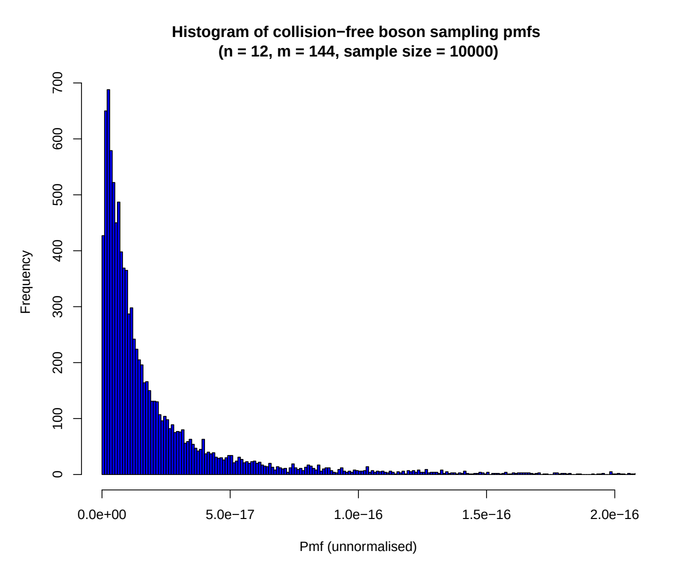


Figure: Boson Sampling probability mass function

Table of results for classical Boson Sampling

Algorithm	Permanents/sample	Max n	Approx/Exact
Rej S. (NSCJBML)	exponential	[15-20]	Approx/Heuristic
MCMC (NSCJBML)	~ 200	~ 30	Approx/Heuristic
Naive	$\binom{n+m-1}{n}$	~ 8	Exact
New result	~ 2	≥ 50	Exact

Table of results for classical Boson Sampling

Algorithm	Permanents/sample	Max n	Approx/Exact
Rej S. (NSCJBML)	exponential	[15-20]	Approx/Heuristic
MCMC (NSCJBML)	~ 200	~ 30	Approx/Heuristic
Naive	$\binom{n+m-1}{n}$	~ 8	Exact
New result	~ 2	≥ 50	Exact

Old complexity: $O\left(\binom{n+m-1}{n} n2^n\right)$ New: $O(n2^n)$

Table of results for classical Boson Sampling

Algorithm	Permanents/sample	Max n	Approx/Exact
Rej S. (NSCJBML)	exponential	[15-20]	Approx/Heuristic
MCMC (NSCJBML)	~ 200	~ 30	Approx/Heuristic
Naive	$\binom{n+m-1}{n}$	~ 8	Exact
New result	~ 2	≥ 50	Exact

Old complexity: $O\left(\binom{n+m-1}{n} n2^n\right)$ New: $O(n2^n)$

The largest boson sampling experiment to date has $n = 5$ photons.

Exact sampling - SODA 2018

Step one: equivalently sample from the pmf

$$p(\mathbf{r}) = \frac{1}{n!}\left|\mathrm{Per}\, A_{\mathbf{r}}\right|^2 = \frac{1}{n!}\left|\sum_{\sigma\in\pi[n]}\prod_{i=1}^{n} a_{r_i\sigma_i}\right|^2, \quad \mathbf{r}\in[m]^n.$$

Exact sampling - SODA 2018

Step one: equivalently sample from the pmf

$$p(\mathbf{r}) = \frac{1}{n!}\left|\operatorname{Per} A_{\mathbf{r}}\right|^2 = \frac{1}{n!}\left|\sum_{\sigma\in\pi[n]}\prod_{i=1}^{n} a_{r_i\sigma_i}\right|^2, \quad \mathbf{r}\in[m]^n.$$

For any ordered sequence of row ids $\mathbf{z}$ there are $n!/\prod_{j=1}^{m} s_j!$ equally likely values of $\mathbf{r}$ in the expanded sample space. So:

$$\frac{n!}{\prod_{j=1}^{m} s_j!}p(\mathbf{z}) = \frac{n!}{\prod_{j=1}^{m} s_j!}\frac{1}{n!}\left|\operatorname{Per} A_z\right|^2 = \frac{\left|\operatorname{Per} A_{\mathbf{z}}\right|^2}{\prod_{j=1}^{m} s_j!},$$

as claimed.

Exact sampling. Compute the joint pmf

Lemma (Marginal probabilities)

The joint pmf of the subsequence $(r_1, \ldots, r_k)$ *is given by*

$$p(r_1, \ldots, r_k) = \frac{(n-k)!}{n!} \sum_{c \in \mathcal{C}_k} \left| \operatorname{Per} A^c_{r_1,\ldots,r_k} \right|^2, \quad k = 1, \ldots, n,$$

where $\mathcal{C}_k$ *is the set of* k*-combinations taken without replacement from* $[n]$ *and* $A^c_{r_1,\ldots,r_k}$ *is the matrix formed from rows* $(r_1, \ldots, r_k)$ *of the columns* c *of* A*.*

Exact sampling. Compute the joint pmf

Lemma (Marginal probabilities)

The joint pmf of the subsequence $(r_1, \ldots, r_k)$ is given by

$$p(r_1, \ldots, r_k) = \frac{(n-k)!}{n!} \sum_{c \in \mathcal{C}_k} \left| \mathrm{Per}\, A^c_{r_1,\ldots,r_k} \right|^2, \quad k = 1, \ldots, n,$$

where $\mathcal{C}_k$ is the set of k-combinations taken without replacement from $[n]$ and $A^c_{r_1,\ldots,r_k}$ is the matrix formed from rows $(r_1, \ldots, r_k)$ of the columns c of A.

Algorithm A samples a chain of conditional pmfs,

$$p(\mathbf{r}) = p(r_1)p(r_2|r_1)p(r_3|r_1, r_2) \ldots p(r_n|r_1, r_2, \ldots, r_{n-1}).$$

Exact sampling. Algorithm A

Algorithm A Boson Sampler: single sample in $\mathcal{O}(mn3^n)$ time

Require: m and n positive integers; A first n columns of $m \times m$ Haar random unitary matrix

1: $\mathbf{r} \leftarrow \varnothing$ ▷ Empty array
2: **for** $k \leftarrow 1$ to n **do**
3: $w_i \leftarrow \sum_{c \in \mathcal{C}_k} |\operatorname{Per} A^c_{(\mathbf{r},i)}|^2,\ i \in [m]$ ▷ Make array w
4: $x \leftarrow \text{Sample}(w)$ ▷ Sample index x from w
5: $\mathbf{r} \leftarrow (\mathbf{r}, x)$ ▷ Append x to $\mathbf{r}$
6: **end for**

Exact sampling. Algorithm A

Algorithm A Boson Sampler: single sample in $\mathcal{O}(mn3^n)$ time

Require: m and n positive integers; A first n columns of $m \times m$ Haar random unitary matrix

$\mathbf{r} \leftarrow \varnothing$ ▷ Empty array
for $k \leftarrow 1$ to n **do**
  $w_i \leftarrow \sum_{c \in \mathcal{C}_k} |\operatorname{Per} A^{c}_{(\mathbf{r},i)}|^2,\ i \in [m]$ ▷ Make array w
  $x \leftarrow \operatorname{Sample}(w)$ ▷ Sample index x from w
  $\mathbf{r} \leftarrow (\mathbf{r}, x)$ ▷ Append x to $\mathbf{r}$
end for

Running time:

$$m \sum_{k=1}^{n} k 2^k \binom{n}{k} = m \frac{2}{3} n 3^n = O(mn3^n)$$

Faster exact sampling. Expand the sample space again

We introduce an auxiliary array $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, where $\boldsymbol{\alpha} \in \pi[n]$. Define the pmf:

$$\phi(r_1, \ldots, r_k|\boldsymbol{\alpha}) = \frac{1}{k!} \left| \operatorname{Per} A_{r_1,\ldots,r_k}^{\{\alpha_1,\ldots,\alpha_k\}} \right|^2, \quad k = 1, \ldots, n-1.$$

Faster exact sampling. Expand the sample space again

We introduce an auxiliary array $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, where $\boldsymbol{\alpha} \in \pi[n]$. Define the pmf:

$$\phi(r_1, \ldots, r_k|\boldsymbol{\alpha}) = \frac{1}{k!}\left|\mathrm{Per}\, A^{\{\alpha_1,\ldots,\alpha_k\}}_{r_1,\ldots,r_k}\right|^2, \quad k = 1, \ldots, n-1.$$

Let $e_k = \phi(r_1, \ldots, r_k|\boldsymbol{\alpha})$ and $d_k = \sum_{r_k} e_k$, $k = 1, \ldots, n-1$ with $e_n = p(r_1, \ldots, r_n)$ and $d_n = p(r_1, \ldots, r_{n-1})$.

Lemma (Sampling from expectation)

With the preceding notation, let $\phi(\mathbf{r}|\boldsymbol{\alpha}) = \prod_{k=1}^{n} e_k/d_k$ *then* $p(\mathbf{r}) = \mathbb{E}_{\boldsymbol{\alpha}}\{\phi(\mathbf{r}|\boldsymbol{\alpha})\}$ *where the expectation is taken over* $\boldsymbol{\alpha}$*, uniformly distributed on* $\pi[n]$ *for fixed* $\mathbf{r}$.

Algorithm B - exact sampling

Sample from chain of conditional probabilities:

$$\phi(\mathbf{r}|\boldsymbol{\alpha}) = \phi(r_1|\boldsymbol{\alpha})\phi(r_2|r_1,\boldsymbol{\alpha})\phi(r_3|r_1,r_2,\boldsymbol{\alpha})\ldots\phi(r_n|r_1,r_2,\ldots,r_{n-1},\boldsymbol{\alpha})$$

Algorithm B - exact sampling

Sample from chain of conditional probabilities:

$$\phi(\mathbf{r}|\boldsymbol{\alpha}) = \phi(r_1|\boldsymbol{\alpha})\phi(r_2|r_1,\boldsymbol{\alpha})\phi(r_3|r_1,r_2,\boldsymbol{\alpha})\ldots\phi(r_n|r_1,r_2,\ldots,r_{n-1},\boldsymbol{\alpha})$$

- $\phi(r_1|\boldsymbol{\alpha}) = |\operatorname{Per} A_{r_1}^{\alpha_1}|^2 = |a_{r_1,\alpha_1}|^2$.
- Takes $O(m)$ time to sample the first row.

Algorithm B - exact sampling

Sample from chain of conditional probabilities:

$$\phi(\mathbf{r}|\boldsymbol{\alpha}) = \phi(r_1|\boldsymbol{\alpha})\phi(r_2|r_1,\boldsymbol{\alpha})\phi(r_3|r_1,r_2,\boldsymbol{\alpha})\ldots\phi(r_n|r_1,r_2,\ldots,r_{n-1},\boldsymbol{\alpha})$$

- $\phi(r_1|\boldsymbol{\alpha}) = |\,\mathrm{Per}\,A_{r_1}^{\alpha_1}|^2 = |a_{r_1,\alpha_1}|^2$.
- Takes $O(m)$ time to sample the first row.
- $\phi(r_2|r_1,\boldsymbol{\alpha}) = e_2/d_2$ but d_2 does not involve r_2.

Algorithm B - exact sampling

Sample from chain of conditional probabilities:

$$\phi(\mathbf{r}|\boldsymbol{\alpha}) = \phi(r_1|\boldsymbol{\alpha})\phi(r_2|r_1,\boldsymbol{\alpha})\phi(r_3|r_1,r_2,\boldsymbol{\alpha})\dots\phi(r_n|r_1,r_2,\dots,r_{n-1},\boldsymbol{\alpha})$$

- $\phi(r_1|\boldsymbol{\alpha}) = |\operatorname{Per} A_{r_1}^{\alpha_1}|^2 = |a_{r_1,\alpha_1}|^2$.
- Takes $O(m)$ time to sample the first row.
- $\phi(r_2|r_1,\boldsymbol{\alpha}) = e_2/d_2$ but d_2 does not involve r_2.
- Since e_2 is proportional to $|\operatorname{Per} A_{r_1,r_2}^{\alpha_1,\alpha_2}|^2$, calculate m permanents of 2×2 matrices; a further $O(m)$ operations.

Algorithm B - exact sampling

Sample from chain of conditional probabilities:

$$\phi(\mathbf{r}|\boldsymbol{\alpha}) = \phi(r_1|\boldsymbol{\alpha})\phi(r_2|r_1, \boldsymbol{\alpha})\phi(r_3|r_1, r_2, \boldsymbol{\alpha}) \ldots \phi(r_n|r_1, r_2, \ldots, r_{n-1}, \boldsymbol{\alpha})$$

- $\phi(r_1|\boldsymbol{\alpha}) = |\operatorname{Per} A_{r_1}^{\alpha_1}|^2 = |a_{r_1,\alpha_1}|^2$.
- Takes $O(m)$ time to sample the first row.
- $\phi(r_2|r_1, \boldsymbol{\alpha}) = e_2/d_2$ but d_2 does not involve r_2.
- Since e_2 is proportional to $|\operatorname{Per} A_{r_1,r_2}^{\alpha_1,\alpha_2}|^2$, calculate m permanents of 2×2 matrices; a further $O(m)$ operations.
- At stage k we need to sample r_k from the pmf proportional to $|\operatorname{Per} A_{r_1,\ldots,r_k}^{\alpha_1,\ldots,\alpha_k}|^2$ considered simply as a function of r_k.

Exploiting the Laplace expansion to speed up stage k

We exploit the Laplace expansion:

$$\mathrm{Per}\, B = \sum_{\ell=1}^{k} b_{k,\ell}\, \mathrm{Per}\, B^{\diamond}_{k,\ell},$$

where $B^{\diamond}_{k,\ell}$ is the submatrix with row k and column ℓ removed.

Exploiting the Laplace expansion to speed up stage k

We exploit the Laplace expansion:

$$\operatorname{Per} B = \sum_{\ell=1}^{k} b_{k,\ell} \operatorname{Per} B^{\diamond}_{k,\ell},$$

where $B^{\diamond}_{k,\ell}$ is the submatrix with row k and column ℓ removed.

Lemma (Amortised permanent computation)

Let B be a $k \times k$ complex matrix and let $\{B^{\diamond}_{k,\ell}\}$ be submatrices of B with row k and column ℓ removed, $\ell \in [k]$. The collection $\{\operatorname{Per} B^{\diamond}_{k,\ell}, \ell \in [k]\}$ can be evaluated jointly in $\mathcal{O}(k2^k)$ time and $\mathcal{O}(k)$ additional space.

The complexity of Boson Sampling

The total time for stage k is $O(k2^k + mk)$. Therefore the total running time is:

$$O(n2^n + mn^2)$$

The same complexity as computing a single permanent.

The complexity of Boson Sampling

The total time for stage k is $O(k2^k + mk)$. Therefore the total running time is:

$$O(n2^n + mn^2)$$

The same complexity as computing a single permanent.

In practice taking one sample takes roughly twice as long as computing one permanent. This pushes the threshold for quantum computational supremacy to at least $n = 50$.

Running times

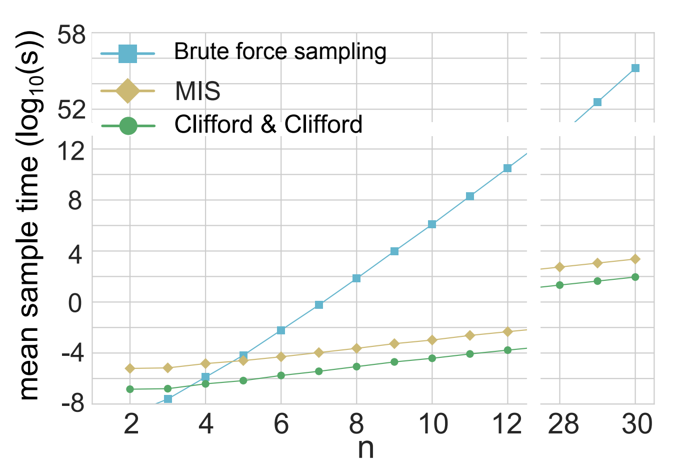


Figure: Running times

What next?

- Classical statistical tests for experimental Boson Samplers.
- Exact sampling takes $O(n2^n)$ time. How much faster is approximate sampling?
- What other quantum sampling problems could have faster classical algorithms?

What next?

- Classical statistical tests for experimental Boson Samplers.
- Exact sampling takes $O(n2^n)$ time. How much faster is approximate sampling?
- What other quantum sampling problems could have faster classical algorithms?

Thank you for listening