# The Classical Complexity of Boson Sampling : A multi-core CPU implementation

**Manan Vaswani**

Level I
40cp project

Supervisor: Dr. Raphaël Clifford
Date: 6th May, 2019

## Acknowledgement of Sources

# Contents

# 1 Introduction

# 2 Background

# 3 Preliminaries

## 3.1 Permanent of a matrix

Computing the permanent of large matrices is one of the key parts of the Boson Sampling problem, as will be discussed in detail while explaining the problem in the subsequent sections of the paper.

### 3.1.1 Definition

**Definition 3.1.** *[1] The permanent of an $n \times n$ matrix $A = (a_{ij})$ is defined as*

$$\operatorname{Per} A = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)} \tag{1}$$

*where the sum is over all elements of the symmetric group $S_n$ i.e. over all permutations of the numbers in $[n] = \{1, 2, ..., n\}$*

**Example 3.2.** For a $2 \times 2$ matrix, the permanent is calculated as follows

$$\operatorname{Per} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad + bc \tag{2}$$

**Example 3.3.** For a $3 \times 3$ matrix, the permanent is calculated as follows

$$\operatorname{Per} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = aei + bfg + cdh + ceg + bdi + afh \tag{3}$$

One can observe that the definition of the permanent is similar to the more commonly used determinant function, differing in the fact that the permanent definition lacks the alternating signs. An important property to note about the permanent function is that it is invariant to transposition i.e. $\operatorname{Per} A = \operatorname{Per} A^T$ [2].

### 3.1.2 Computing the permanent

Valiant showed that the problem of computing the permanent of a matrix is in the class #P-complete which implies that it is unlikely to have a polynomial time algorithm which implies that it is unlikely to have a polynomial-time algorithm [3]. The naive algorithm obtained by directly translating the formula into an algorithm would run in $\mathcal{O}(n!n)$ time.

A significant improvement on the naive approach, Rysers algorithm uses a variant of the inclusion-exclusion principle and can be evaluated in $\mathcal{O}(2^n n^2)$ time [2]. Nijenhuis and Wilf sped this up to $\mathcal{O}(2^n n)$ time by iterating over the sum in Gray Code order [4].

Another formula that is as fast as Rysers was independently derived by Balasubramanian[5], Bax[6], Franklin and Bax[7], and Glynn[8], all using different methods. We shall henceforth refer to this as Glynn's formula and it is described as follows.

Let $M = (m_{ij})$ be an $n \times n$ matrix with $m_{ij} \in \mathbb{C}$, then

$$\operatorname{Per} M = \frac{1}{2^{n-1}} \sum_{\delta} \left( \prod_{k=1}^{m} \delta_k \right) \prod_{j=1}^{m} \sum_{i=1}^{m} \delta_i m_{ij} \tag{4}$$

where $\delta \in \{-1, 1\}^n$ with $\delta_1 = 1$. Hence there are $2^{n-1}$ such values for $\delta$.

Implementing the formula as is would require $\mathcal{O}(2^n n^2)$ time. However, iterating over the $\delta$ arrays in Gray code order reduces it to $\mathcal{O}(2^n n)$ time.

# 4 Describing the paper and specifying the problem

Cliiford and Clifford [9] gave a study and analysis of the classical complexity of the exact Boson Sampling problem and proposed an algorithm that is significantly faster than previous algorithms for the problem. The algorithm is simple to implement, and is able to solve the Boson Sampling problem for system sizes much greater than quantum computing systems currently available, which reduces the likelihood of achieving quantum supremacy in the context of Boson Sampling in the near future rephrase last bit.

## 4.1 Explaining the problem

A summary of the problem in purely mathematical terms is as follows.

Let $m$ and $n$ be positive integers. Consider all possible multisets[1] of size $n$ with elements in $[m]$, where $[m] = \{1, ..., m\}$. Let $z = [z_1, z_2, ..., z_n]$ be an array representation of such a multiset, with its elements in non-decreasing order. In other words, $z$ is an array of $n$ integers taken from $[m]$ (with repetition) and arranged in non-decreasing order. Define $\Phi_{m,n}$ to be the set of all distinct values that $z$ can take. Maybe explain size of phi m,n with stars and bars thing. Define $\mu(z) = \prod_{j=1}^{m} s_j!$ where $s_j$ is the multiplicity of $j$ in the array $z$ i.e. the number of times it appears in $z$.

$A = (a_{ij})$ is a complex-valued $m \times n$ matrix constructed by taking the first $n$ columns of a given $m \times m$ Haar random unitary matrix. For each $z$, build an $n \times n$ matrix $A_z$ where the $k^{\text{th}}$ row of $A_z$ is the $z_k^{\text{th}}$ row in $A$, for $k = 1, ..., n$. Finally, define a probability mass function over $\Phi_{m,n}$ as

$$q(z) = \frac{1}{\mu(z)} |\operatorname{Per} A_z|^2 = \frac{1}{\mu(z)} |\sum_{\sigma} \prod_{k=1}^{n} a_{z_k \sigma_k}|^2, \quad z \in \Phi_{m,n}$$

[1] A multiset is a special kind of set in which elements can be repeated

where $\operatorname{Per} A_z$ is the permanent of $A_z$ and $\pi[n]$ is the set of all permutations of $[n]$.

The computing task is to simulate random samples from the above pmf $q(z)$.

# 5 Implementation

# 6 Results

# 7 Conclusion

# References


[1] Henryk Minc Marvin Marcus. Permanents. *The American Mathematical Monthly*, vol. 72:577–591, 1965.

[2] Herbert John Ryser. *Combinatorial Mathematics*. Mathematical Association of America, 1963.

[3] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.

[4] Albert Nijenhuis and Herbert S. Wilf. 23 - the permanent function (perman). In ALBERT NIJENHUIS and HERBERT S. WILF, editors, *Combinatorial Algorithms (Second Edition)*, pages 217 – 225. Academic Press, second edition edition, 1978.

[5] K Balasubramanian. *Combinatorics and diagonals of matrices*. PhD thesis, Indian Statistical Institute, Calcutta, Dec 1980.

[6] Eric Bax. *Finite-difference algorithms for counting problems*. PhD thesis, California Institute of Technology, February 1998.

[7] Eric Bax and Joel Franklin. A finite-difference sieve to count paths and cycles by length. *Information Processing Letters*, 60(4):171 – 176, 1996.

[8] David G. Glynn. The permanent of a square matrix. *European Journal of Combinatorics*, 31(7):1887 – 1891, 2010.

[9] Peter Clifford and Raphaël Clifford. The classical complexity of boson sampling. *CoRR*, abs/1706.01260, 2017.