Program for Simulating Boson Sampling

Jinlong Huang
Feb 15 2018

February 28, 2018

1 Transition Amplitude

In the model of boson sampling, the amplitude of transforming state $|t_1, t_2, \ldots, t_m\rangle$ to state $|s_1, s_2, \ldots, s_m\rangle$ is given by,

$$\langle s_1 \ldots s_m|U|t_1 \ldots t_m\rangle = \frac{G(U,s,t)}{\sqrt{(s_1! \cdots s_m!)(t_1! \cdots t_m!)}} \tag{1}$$

where

$$\begin{aligned} G(U,s,t) &= \frac{v_s^2}{n^m} \sum_{\{\boldsymbol{z}\}} (\bar{z}_1^{s_1} \bar{z}_2^{s_2} \cdots \bar{z}_m^{s_m}) \Pi_{k=1}^m (w_{k,1} z_1 + \ldots + w_{k,m} z_m)^{t_k} \\ &= \mathbb{E}\left(v_s^2 (\bar{z}_1^{s_1} \bar{z}_2^{s_2} \cdots \bar{z}_m^{s_m}) \Pi_{k=1}^m (w_{k,1} z_1 + \ldots + w_{k,m} z_m)^{t_k}\right) \\ &\equiv \mathbb{E}(\text{mGenGly}(z)) \end{aligned}$$

$$v_s \equiv \sqrt{(s_1!/s_1^{s_1})(s_2!/s_2^{s_2}) \cdots (s_m!/s_m^{s_m})} \tag{2}$$

$$z_i \in \left\{\sqrt{s_i} w^0, \sqrt{s_i} w^1, \ldots, \sqrt{s_i} w^{n-1}\right\} \tag{3}$$

$$w = e^{2\pi i/n} \tag{4}$$

2 Gurvits's Algorithm

Glynn's algorithm is an exact deterministic classical algorithm, while Gurvits's algorithm is an approximate sampling algorithm. In the following we analysized the success probability of Gurvits's algorithm.

From the Chernoff bound for complex variables, We have the following result. Let $X_i = \frac{\text{mGenGly}(z_i)}{\prod_{k=1}^m (s_k! t_k!)^{1/2}}, X = \sum_{i=1}^T X_i,\ \mu \equiv \mathbb{E}(X_i) = \langle s_1 \ldots s_m|U|t_1 \ldots t_m\rangle$, $b \equiv \min\{v_s/v_t, v_t/v_s\}$, then $|X_i| \leqslant v_s/v_t$. Considering the conjugate of X_i , we conclude $|X_i| \leqslant b$.

$$\boldsymbol{P}\left(\frac{|\frac{\boldsymbol{X}}{\boldsymbol{T}} - \boldsymbol{\mu}|}{|\boldsymbol{\mu}|} \geqslant \boldsymbol{\varepsilon}\right) \leqslant \boldsymbol{4e^{\frac{-T|\mu|^2\varepsilon^2}{4b^2}}} \tag{5}$$

To conclude, we have estimated $\langle s_1 \dots s_m | U | t_1 \dots t_m \rangle$ using $T = O(1/\varepsilon^2)$ samples from $\frac{\mathrm{mGenGly}(z_i)}{\prod_{k=1}^{m}(s_k! t_k!)^{1/2}}$ within additive error $\pm\varepsilon \times \min(v_s/v_t, v_t/v_s)$, and have a high probabilty $\left(1 - 4e^{\frac{-T|\mu|^2\varepsilon^2}{4b^2}}\right)$ success. For each mGenGly(z)we need mn operations, thus the polynomian-time sampling algorithm scales as $O(mn/\varepsilon^2)$.

Note that in order to success in a high probability, we also require the permanent of unitary, i.e. $|\mu|$ not too small.

3 The Classical Simulation Program

In this simulation software we simulated boson sampling from 6 photons up to 10 photons. There are total 5 files from test6.m to test10.m. The number in the end of the file name stands for the number of photons in that simulation. In all files, the transition amplitude from initial state to final state is calculated using both Glynn's algorithm and Gurvits's algorithm.

For **Glynn's algorithms**, it has four parameters that can be tuned to a specific simulation instance: initial state, final state, # of photons n and # of modes m. In the following, specific instructions to adjust these parameters are given.

The initial state: In the definition of variable "mGenGly", the exponents of $Z(i)'$ specify how the initial state are arranged. For example, if the initial state is set to be [1,2,0,3], i.e. total 6 photons in 4 modes, mGenGly should be defined as mGenGly $= Z(i0)' * Z(i1)'^2 * Z(i3)'^3 * \cdots$. The prime in Matlab means complex conjugate. Also, according to the formula for transition amplitude, mGenGly should be further multiplied by a factor $(s_1!/\sqrt{s_1^{s_1}})(s_2!/\sqrt{s_2^{s_2}})\cdots(s_m!/\sqrt{s_m^{s_m}})$. In this example, mGenGly should be defined as mGenGly $= (1!/\sqrt{1^1}) * (2!/\sqrt{2^2}) * 1 * (3!/\sqrt{3^3}) * Z(i0)' * Z(i1)'^2 * Z(i2)'^0 * Z(i3)'^3 * \cdots$

The final state: There is only one place to be modified for final state: the exponents of dot(randU(i,:),ZVec), similarly as the initial state. If the final state is also [1,2,0,3], mGenGly should be defined as mGenGly $= \cdots * \mathrm{dot}(\mathrm{randU}(1,:),\mathrm{ZVec})^1 * \mathrm{dot}(\mathrm{randU}(2,:),\mathrm{ZVec})^2 * \mathrm{dot}(\mathrm{randU}(3,:),\mathrm{ZVec})^0 * \mathrm{dot}(\mathrm{randU}(4,:),\mathrm{ZVec})^3$.

# of photons n: Three places should be modified:
1) The set of Z that Z(i) can choose from, which is defined before While-loop
2) The end point for each While-loop
3) Denominator of TransAmp.

# of modes m: There are four places should be modified:
1) # of While-loop
2) length of ZVec
3) # of dot(randU(i,:)
4) Denominator of TransAmp.

For **Gurvits's algorithm**, besides above four parameters, there is one more parameter one can adjust: # of samples T. This variable is defined in the beginning of Gurvits's algorithm, and this is the only place should be changed.

To calculate the success probability of Gurvits's algorithm, two additional parameters should be specified: tolerance for error "TOL" and # of experiments "maxExp". In the case of 6 photons, TOL is set from 0.05 to 0.015, and maxExp = 1000. The figure for success probability vs Fidelity is plotted in the end of this report.

The executions of Glynn's algorithm in the case of 8 to 10 photons was not complete, for the time it requires in a standard laptop are 45.8 minutes, 20.6 hours and 27.7 days.

There are two ways to implement a random unitary according to Haar measure. The first one is import from file "RandomUnitary", which is based on file "opt_args.m". The other method is to firstly implement a random Hermitian matrix from rand(n) method provided in Matlab, and then exponentiate it to get a random unitary matrix.

The number of modes are supposed to be $O(n^2)$ to illustrate quantum supremacy, however in here for the purpose of demonstrating the efficiency of Gurvits's algorithm, we simplify the simulation by constraining $m = n$.

[1] Neville, Alex, et al. "No imminent quantum supremacy by boson sampling." *arXiv preprint arXiv:1705.00686* (2017).

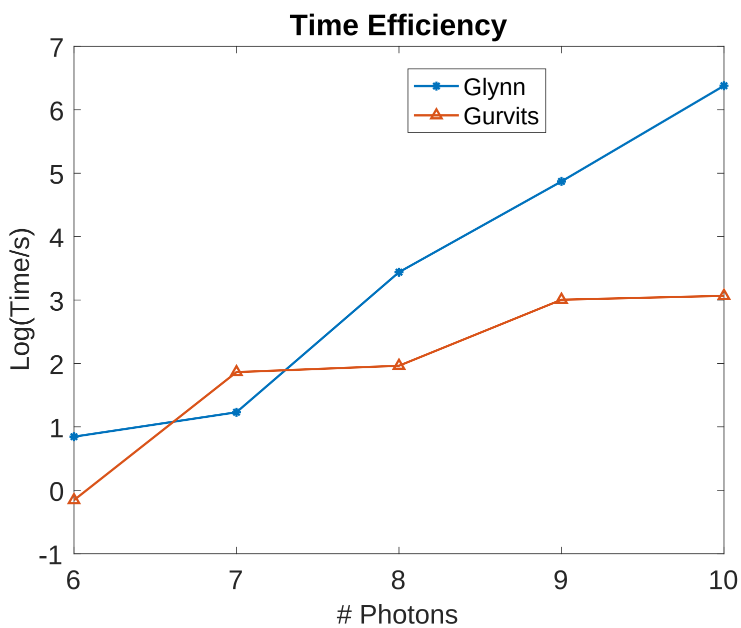

Figure 1: The time efficiency for both Glynn and Gurvits are analyzed in all cases from 6 photons to 10 photons. Due to the capability of classical hardware, however, the successful probability for Gurvits's algorithm is analyzed only in the case with only 6 photons. As seen from the plot for time efficiency, the time required for Glynn's algorithm grows exponentially, while the time for Gurvits's method grows much slower. This result is consistent with the result in [1].

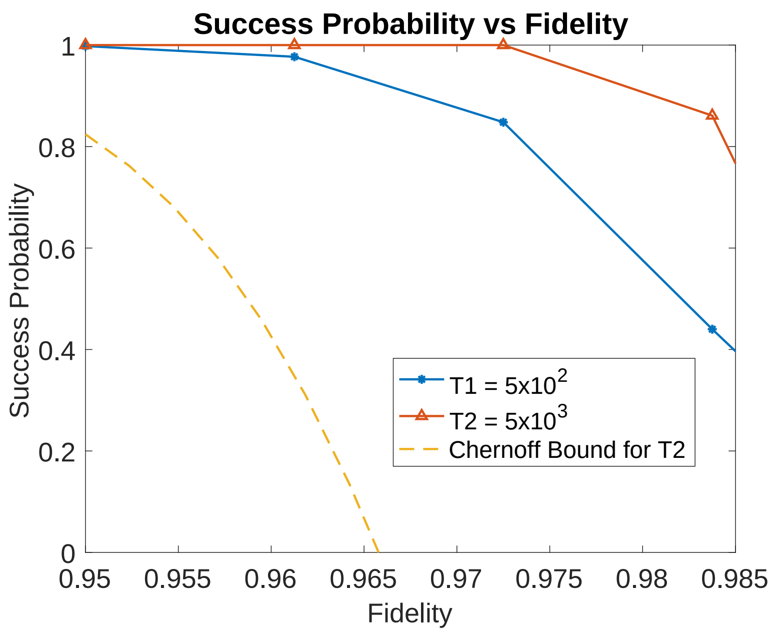

Figure 2: In the case of 6 photons, the results for success probability vs fidelity is plotted in file "6photons.pdf". To compute the success probability, Gurvits's method was run by 1000 times. The dashed line stands for the theoretical prediction from Chernoff bound. From this theoretical bound, to achieve fidelity 90%, success probability 97%, the number of samples should be $T = 5 \times 10^6$. However, in practice, the actual samples we need in Gurvits are much less than we expected. In this simulation, even if the number of samples is only T=5×10^2, the accuracy of Gurvits's algorithm goes beyond our expectation. When fidelity is set as 95%, the success probability is 100%.When fidelity is set as 96%, the success probability is still 97%.