I. QUANTUM SEARCH

Classically, searching an unsorted database of N items requires $O(N)$ time. However, performing the search using the quantum algorithm of Grover's algorithm takes only $O(\sqrt{N})$ time [1], which is optimal for searching an unsorted database [2, 3]. The "time" here is defined as the number of queries to the oracle. Unlike other quantum algorithms, Grover's algorithm provides "only" a quadratic speedup instead of exponential speedup over classical counterparts. However, even quadratic speedup is important and considerable, since search algorithms are widely used in various fields.

Consider an unsorted database with N items, Grover's algorithm require $n = \log_2^N$ qubits to represent the database, the steps of the algorithm are given as follows:

1. Initialize the qubits to $|0\rangle^{\otimes n}$, and then apply the Hadamard transform $H^{\otimes n}$ on these qubits to obtain the uniform superposition state

$$|\varphi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

One can write the state $|\varphi\rangle$ to be,

$$|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = \sqrt{\frac{M}{N}}|T\rangle + \sqrt{\frac{N-M}{N}}|T_\perp\rangle$$

where $|T\rangle$ is the uniform superposition of M target elements, and $|T_\perp\rangle$ is the uniform superposition of $N - M$ non-target elements.

2. Apply the Grover iteration for about $\frac{\pi}{4}\sqrt{\frac{N}{M}}$ times. The iteration is described as below.

i. Apply the operator $U_T = I - 2|T\rangle\langle T|$.

ii. Apply the operator $U_\varphi = 2|\varphi\rangle\langle\varphi| - I$.

3. Measure these n qubits to obtain the target elements.

The algorithm is called single target Grover's algorithm if $M = 1$, and called multiple targets Grover's algorithm if $M > 1$. In addition, instead of searching the whole database for the target element, we could divide the whole database in several blocks, and then implement a variant of the algorithm, named partial search, to look for the block which contains the target element [4]. Subsequently, the partial search algorithm has been optimized [5–8] and further generalized to hierarchical quantum partial search algorithm [9, 10].

Beside the partial search, Grover's algorithm has been generalized to many other application scenarios, such as amplitude estimation [11], quantum counting [12–14], finding the minimum [15–17], applying for arbitrary initial complex amplitude distributions [18], fixed-point quantum search [19–21], and multi-phase search [22]. Furthermore, a closely related but more difficult problem, namely spatial search, has been also studied. The database of spatial search are some graph structure, and for some well-connected graphs, $O(\sqrt{N})$ time is still achievable [23–28]. So far, the Grover's algorithm has been demonstrated with NMR [29], trapped ion [30], photonic [31], and superconducting hardware [32].

II. INTEGER FACTORISATION

Today, the security of the most widely used public-key cryptography scheme, RSA scheme, is based the difficulty of factoring large numbers for classical computer. Thus, the problem of efficient factoring numbers has attracted widespread attention in the field of computer and information Science. However, there is still no no classical algorithm that can factoring numbers in polynomial time [33]. In 1994, a major breakthrough was made by Peter Shor that quantum computers could efficiently factor numbers in polynomial time [34, 35], posing a serious threat to information security in business transactions on the Internet, such as e-commerce.

Suppose we want to factor N using Shor's algorithm, for a randomly chose number a $(0 < a < N)$ that is co-prime to N, Shor's algorithm can output the minimum integer r that satisfies $a^r \bmod N = 1$. From this period r, the prime factors of N are given by the greatest common divisor (GCD) of $a^{r/2} \pm 1$ and N, which can be solved classically. The Shor's algorithm can be broken into the following simple steps:

1. Creat two registers, register 1 and register 2, which have $n = 2\left\lceil \log_2^N \right\rceil$ qubits and $m = \left\lceil \log_2^N \right\rceil$, respectively. Initialize the quantum register 2 to $|0, ..., 0, 1\rangle$, and the quantum register 1 to

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

2. Apply the modular exponential function $f(x) = a^x \bmod N$ on register 2 when register 1 is in state $|x\rangle$, and then obtain the state

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|a^x \bmod N\rangle$$

Due to the quantum parallelism, the quantum computer will calculate the function $a^x \bmod n$ for all $x \in \{0, ..., 2^n\}$ in parallel by only one step.

3. Apply quantum Fourier transformation (QFT) on the register 1, yielding

$$\frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} e^{2\pi i x y/2^n} |y\rangle|a^x \bmod N\rangle$$

4. Measure the registers. Due to the QFT in step 3, register 1 will output $|y\rangle$ with high probability, where

$y = c2^n/r$ (for integer c), which could enable to deduce the period r by some post processing on a classical computer. Then, the factor of n can be determined by taking $\gcd(x^{r/2}+1, n)$ and $\gcd(x^{r/2}-1, n)$.

Given an n-bit integer, the best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/4+o(1)})$ [33, 36]. However, the Shor's algorithm can solve this task in $O(n^3)$ time [34, 35], which achieves an exponential speedup over all classical algorithms. Furthermore, for some special case, such as factoring safe semiprimes (an important class of numbers used in cryptography), more feaster quantum algorithm is developed by changing the classical part of Shor's algorithm [37]. In the experimental side, some proof-of-principle experiments with a small number of qubits have already been achieved [38–46].

III. TOPOLOGICAL DATA ANALYSIS

Topological data analysis (TDA) [47] is a tool for extracting useful information from unstructured data by studying the shape of data. In particular, it can be used to estimate the certain topological features of data, such as the Betti numbers, which count the number of holes and voids of various dimensions in a scatterplot. In 2016, Lloyd, Garnerone, and Zanardi proposed a quantum algorithms for TDA, quantum TDA, which can provide an exponential speedup over the best currently known classical algorithms [48].

Generally, the k-th Betti number refers to the number of k-dimensional holes. Consider n data points, we will introduce the steps of the quantum TDA algorithm for calculating the k-th Betti number:

1. Simplicial complex construction. Implement the Grover's algorithm with a membership oracle function $\{f_k^\epsilon(s_k) = 1 \text{ if } s_k \in S_k^\epsilon\}$ to construct the simplicial complex state,

$$|\psi\rangle_k^\epsilon = \frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k \in S_k^\epsilon} |s_k\rangle.$$

where the k-simplex $|s_k\rangle$ is an n-qubit quantum state with $k+1$ 1s at positions $j_0, j_1, ..., j_k$ and 0s at the other remaining positions, which means a connected graph with points $j_0, j_1, ..., j_k$, and the Vietoris-Rips simplicial complex S_k^ϵ is the set of k-simplices where all points are within distance ϵ of each other.

2. Mixed state construction. Add an ancillary register consisted of n qubits and then perform controlled-NOT (CNOT) gates to copy $|\psi\rangle_k^\epsilon$ to construct $\frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k \in S_k^\epsilon} |s_k\rangle \otimes |s_k\rangle$, finally trace out the ancillary register to obtain mixed state ρ_k^ϵ.

$$\rho_k^\epsilon = \frac{1}{|S_k^\epsilon|} \sum_{s_k \in S_k^\epsilon} |s_k\rangle\langle s_k|.$$

3. Topological analysis. Define the boundary map ∂_k^ϵ as $\partial_k^\epsilon |s_k\rangle = \sum_l (-1)^l |s_{k-1}(l)\rangle$, where $|s_{k-1}(l)\rangle$ is obtained from s_k with vertices $j_0 ... j_l ... j_k$ by omitting the l-th point j_l from s_k. Then, transform the boundary map to a Hermitian matrix

$$B_k^\epsilon = \begin{pmatrix} 0 & \partial_k^\epsilon \\ \partial_k^{\epsilon\,\dagger} & 0 \end{pmatrix}.$$

Apply the phase-estimation algorithm to decompose ρ_k^ϵ in terms of the eigenvectors and eigenvalues of B_k^ϵ, and then measure the eigenvalue register. Employing the probability of measuring zero η_k^ϵ, the dimension of the kernel of ∂_k^ϵ could be calculated as $\dim(\mathrm{Ker}\,\partial_k^\epsilon) = \eta_k^\epsilon \cdot |S_k^\epsilon|$. Using the similar approach for calculating $\dim(\mathrm{Ker}\,\partial_{k+1}^\epsilon)$, then we can reconstruct the k-th Betti number by,

$\beta_k^\epsilon = \dim(\mathrm{Ker}\,\partial_k^\epsilon) - \dim(\mathrm{Im}\,\partial_{k+1}^\epsilon) = \dim(\mathrm{Ker}\,\partial_k^\epsilon) + \dim(\mathrm{Ker}\,\partial_{k+1}^\epsilon) - |S_{k+1}^\epsilon|$.

Above is the basic idea of calculating the k-th Betti number. In practical application, we can define the full Hermitian boundary map, Dirac operator, to be $B^\epsilon = B_1^\epsilon \oplus B_2^\epsilon \oplus ... \oplus B_n^\epsilon$, and then use the Dirac operator to estimate Betti numbers to all orders (see [48] for details).

In theory, the quantum TDA algorithm could estimate approximate values of Betti numbers to all orders and to accuracy δ in time $O(n^5/\delta)$ [48], by contrast, the best classical algorithms for estimating Betti numbers to all orders to accuracty δ takes time at least $O(2^n log(1/\delta))$ [49–51].

[1] L. K. Grover, Physical review letters **79**, 325 (1997).
[2] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM journal on Computing **26**, 1510 (1997).
[3] C. Zalka, Physical Review A **60**, 2746 (1999).
[4] L. Grover and J. Radhakrishnan, "Acm symp. on parallel algorithms and architectures," (2005).
[5] B.-S. Choi and V. E. Korepin, Quantum Information Processing **6**, 243 (2007).
[6] V. E. Korepin, Journal of Physics A: Mathematical and General **38**, L731 (2005).
[7] V. E. Korepin and J. Liao, Quantum Information Processing **5**, 209 (2006).
[8] V. E. Korepin and B. C. Vallilo, Progress of theoretical physics **116**, 783 (2006).
[9] V. E. Korepin and Y. Xu, International Journal of Modern Physics B **23**, 5727 (2009).
[10] V. E. Korepin and Y. Xu, International Journal of Modern Physics B **21**, 5187 (2007).
[11] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics **305**, 53 (2002).

[12] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, arXiv preprint quant-ph/9605034 (1996).

[13] G. Brassard, P. Høyer, and A. Tapp, Automata, languages and programming , 820 (1998).

[14] M. Mosca *et al.*, in *MFCS98 workshop on Randomized Algorithms* (1998) pp. 90–100.

[15] C. Durr and P. Hoyer, arXiv preprint quant-ph/9607014 (1996).

[16] A. Nayak and F. Wu, in *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (ACM, 1999) pp. 384–393.

[17] L. A. B. Kowada, C. Lavor, R. Portugal, and C. M. De Figueiredo, International Journal of Quantum Information **6**, 427 (2008).

[18] E. Biham, O. Biham, D. Biron, M. Grassl, and D. A. Lidar, Physical Review A **60**, 2742 (1999).

[19] L. K. Grover, Physical Review Letters **95**, 150501 (2005).

[20] T. Tulsi, L. Grover, and A. Patel, arXiv preprint quant-ph/0505007 (2005).

[21] T. J. Yoder, G. H. Low, and I. L. Chuang, Physical review letters **113**, 210501 (2014).

[22] L. Tan, B. Wan-Su, L. Wen-Qian, Z. Hou, and F. Xiang-Qun, Chinese Physics Letters **31**, 050301 (2014).

[23] A. M. Childs and J. Goldstone, Physical Review A **70**, 022314 (2004).

[24] S. Chakraborty, L. Novo, A. Ambainis, and Y. Omar, Physical review letters **116**, 100501 (2016).

[25] T. G. Wong, Journal of Physics A: Mathematical and Theoretical **49**, 195303 (2016).

[26] J. Janmark, D. A. Meyer, and T. G. Wong, Physical Review Letters **112**, 210502 (2014).

[27] D. A. Meyer and T. G. Wong, Physical review letters **114**, 110503 (2015).

[28] T. G. Wong, Quantum Information Processing **15**, 1411 (2016).

[29] I. L. Chuang, N. Gershenfeld, and M. Kubinec, Physical review letters **80**, 3408 (1998).

[30] K.-A. Brickman, P. Haljan, P. Lee, M. Acton, L. Deslauriers, and C. Monroe, Physical Review A **72**, 050306 (2005).

[31] P. Walther, K. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger, Nature **434**, 169 (2005).

[32] L. DiCarlo, J. Chow, J. Gambetta, L. S. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin, *et al.*, Nature **460**, 240 (2009).

[33] J. M. Pollard, in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 76 (Cambridge University Press, 1974) pp. 521–528.

[34] P. W. Shor, SIAM Journal on Computing **26**, 1484 (1997).

[35] P. W. Shor, in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on* (Ieee, 1994) pp. 124–134.

[36] V. Strassen, Jahresbericht der Deutschen Mathematiker-Vereinigung **78**, 1 (1976).

[37] F. Grosshans, T. Lawson, F. Morain, and B. Smith, arXiv preprint arXiv:1511.04385 (2015).

[38] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, Science **351**, 1068 (2016).

[39] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'malley, D. Sank, A. Vainsencher, J. Wenner, *et al.*, Nature Physics **8**, 719 (2012).

[40] E. Martin-Lopez, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, and J. L. O'brien, Nature Photonics **6**, 773 (2012).

[41] A. Politi, J. C. Matthews, and J. L. O'brien, Science **325**, 1221 (2009).

[42] C.-Y. Lu, D. E. Browne, T. Yang, and J.-W. Pan, Physical Review Letters **99**, 250504 (2007).

[43] B. Lanyon, T. Weinhold, N. K. Langford, M. Barbieri, D. James, A. Gilchrist, and A. White, Physical Review Letters **99**, 250505 (2007).

[44] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature **414**, 883 (2001).

[45] N. Johansson and J.-Å. Larsson, arXiv preprint arXiv:1706.03215 (2017).

[46] H.-L. Huang, Q. Zhao, X. Ma, C. Liu, Z.-E. Su, X.-L. Wang, L. Li, N.-L. Liu, B. C. Sanders, C.-Y. Lu, *et al.*, Physical Review Letters **119**, 050503 (2017).

[47] G. Carlsson, Bulletin of the American Mathematical Society **46**, 255 (2009).

[48] S. Lloyd, S. Garnerone, and P. Zanardi, Nature communications **7**, 10138 (2016).

[49] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, Discrete & Computational Geometry **37**, 103 (2007).

[50] S. Basu, Discret. Comput. Geom. **22**, 1 (1999); **30**, 65 (2003); Found. Comput. Math. **8**, 45 (2008); arXiv:1409.1534.

[51] J. Friedman, Algorithmica **21**, 331 (1998).