An Efficient Quantum Algorithm for the Traveling Salesman Problem

Anant Sharma,[1, 2, *] Nupur Deshpande,[3, †] Sanchita Ghosh,[4, 2, ‡] Sreetama Das,[5, 6, 7, 8, §] and Shibdas Roy[2, 9, ¶]

[1]*Department of Physics, Université de Bourgogne, Dijon 21000, France.*
[2]*Center for Quantum Engineering, Research and Education (CQuERE), TCG CREST, Kolkata 700091, India.*
[3]*Department of Physics, Indian Institute of Science Education and Research (IISER), Tirupati 517507, India.*
[4]*Faculty of Electrical Engineering, Mathematics and Computer Science,*
Delft University of Technology, Delft, 2628 CD, Netherlands.
[5]*Instituto de Física Interdisciplinar y Sistemas Complejos (IFISC),*
UIB-CSIC UIB Campus, Palma de Mallorca, E-07122, Spain.
[6]*Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO), 50125 Florence, Italy.*
[7]*European Laboratory for Non-Linear Spectroscopy (LENS), 50019 Sesto Fiorentino, Italy.*
[8]*Department of Physics and Astronomy, University of Florence, 50019 Sesto Fiorentino, Italy.*
[9]*Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India.*

The traveling salesman problem is the problem of finding out the shortest route in a network of cities, that a salesman needs to travel to cover all the cities, without visiting the same city more than once. This problem is known to be NP-hard with a brute-force complexity of $O(N^N)$ or $O(N^{2N})$ for N number of cities. This problem is equivalent to finding out the shortest Hamiltonian cycle in a given graph, if at least one Hamiltonian cycle exists in it. Quantum algorithms for this problem typically provide with a quadratic speedup only, using Grover's search, thereby having a complexity of $O(N^{N/2})$ or $O(N^N)$. We present a bounded-error quantum polynomial-time (BQP) algorithm for solving the problem, providing with an exponential speedup. The overall complexity of our algorithm is $O(N^3\log(N)\kappa/\epsilon + 1/\epsilon^3)$, where the errors ϵ are $O(1/\text{poly}(N))$, and κ is the not-too-large condition number of the matrix encoding all Hamiltonian cycles.

I. INTRODUCTION

Logistics and complex supply chain related problems that require optimization are challenging to solve. The Traveling Salesman Problem (TSP) is the most commonly explored use case of combinatorial optimization. The problem appears simple: find the shortest path in a graph that visits each node exactly once and returns to its origin. It is an NP-hard problem, where NP stands for nondeterministic polynomial-time [1]. The hardest of all problems in NP complexity class are NP-complete, while problems that are at least as hard as NP-complete problems and can lie outside NP are NP-hard. The real-world applications of TSP extend to domains, such as transportation, manufacturing, and network design.

A path in a weighted graph that visits each node, representing a city, exactly once, through edges weighted with distances (say, in kilometers) between the cities, before returning to the starting node, is called a Hamiltonian cycle. Clearly, there can be multiple distinct Hamiltonian cycles in a given graph, with different sums of the weights of the constituent edges, and the problem being considered here is then to find the Hamiltonian cycle with the smallest sum of the weights of the constituent edges. Besides undirected graphs, we mainly consider directed graphs here with only one edge between any two nodes in the same direction, since some or all roads in a city-network can be one-way, but the given graph itself need not be fully connected.

Classically, the problem has been tackled by exact as well as heuristic algorithms. Notably, seminal work in linear programming in Ref. [2] introduced cutting planes, laying the groundwork for branch and cut methods [3–5], and branch and bound algorithms [6, 7]. In particular, Ref. [8] discussed an implementation of the method from Ref. [2], suitable for TSP instances having a million or more cities. There are other approaches to solve TSP in the literature, such as a thermodynamic approach to find approximate solutions using a Monte Carlo algorithm [9].

With the advent of quantum computing and the possibility of solving combinatorial optimization problems faster than classical methods, TSP became a test bed for ample Noisy Intermediate Scale Quantum (NISQ)-era algorithms. Both gate-based approaches and annealer-based approaches have been tested extensively for TSP. Refs. [10–12] use annealer-based approaches for the problem. For example, while Ref. [10] explored the use of Quadratic Unconstrained Binary Optimization (QUBO) models in solving TSP through quantum annealing algorithms and Graph Neural Networks (GNN), Ref. [11] proposed a path-integral Monte Carlo quantum annealing scheme. By contrast, gate-based approaches include the use of Quantum Approximate Optimization Algorithm (QAOA) [13], and Variational Quantum Eigensolvers (VQE) [14]. Fault-Tolerant Quantum Computing (FTQC) algorithms use approaches based on Grover's search to solve the problem with a quadratic speedup [15–18]. In Ref. [16], the eigenstates corresponding to Hamiltonian cycles in the graph are treated as given, but

[*] anantsharma2410@gmail.com
[†] dnupur79@gmail.com
[‡] sanchita.ghosh14@gmail.com
[§] sreetama@ifisc.uib-csic.es
[¶] roy.shibdas@gmail.com

in practice, finding all the Hamiltonian cycles in a graph itself is an NP-complete problem.

In this work, we develop a gate-based FTQC BQP algorithm that solves TSP with exponential speedup. We achieved this by using a novel quantum circuit involving controlled swap gates to yield all candidate Hamiltonian cycles of a given graph of N nodes, quantum phase estimation to capture the sums of edges of these Hamiltonian cycles, and density matrix exponentiation to find the shortest Hamiltonian cycle. The algorithm complexity is $O(N^3 \log(N)\kappa/\epsilon + 1/\epsilon^3)$, that is polynomial in N, when the errors ϵ are polynomially but not exponentially small in N, as required for the overall error probability of our algorithm to be less than or equal to 1/3, and κ is the condition number of the matrix encoding all Hamiltonian cycles of the graph.

II. METHOD

Consider that an arbitrary directed graph is given, with N vertices that are connected by edges. Then, there can be a maximum of $2 \times {}^{N}C_2 = 2 \times N!/(2(N-2)!) = N(N-1)$ number of edges in the graph. A Hamiltonian cycle, if one exists in the graph, would have exactly N vertices and N edges. We here want to find a quantum algorithm that would take time, polynomial in N, to solve the problem of finding out the shortest Hamiltonian cycle, if one exists, in the graph.

We start with $\log L := \lceil \log N \rceil$ number of qubits. For example, if we have 4 vertices in a graph, we would use 2 qubits that can have 4 levels: $|0\rangle = |00\rangle$, $|1\rangle = |01\rangle$, $|2\rangle = |10\rangle$, $|3\rangle = |11\rangle$. We create a unitary operator U, that encodes the distances of all the edges into the phase factors of their eigenstates:

$$U = \sum_{j,k=0}^{L-1} e^{i\phi_{jk}} |jk\rangle\langle jk|, \tag{1}$$

where ϕ_{jk} is the distance between city j to city k. The phase factor ϕ_{jk} is equal to the phase factor ϕ_{kj} in case of an undirected graph. For example, if we have $N = 4 (= L)$, i.e, cities represented by index values $j = 0, 1, 2, 3$, then we create the following diagonal unitary matrix:

$$\begin{aligned} U = {} & e^{i\phi_{00}}|00\rangle\langle 00| + e^{i\phi_{01}}|01\rangle\langle 01| + e^{i\phi_{02}}|02\rangle\langle 02| + e^{i\phi_{03}}|03\rangle\langle 03| \\ & + e^{i\phi_{10}}|10\rangle\langle 10| + e^{i\phi_{11}}|11\rangle\langle 11| + e^{i\phi_{12}}|12\rangle\langle 12| + e^{i\phi_{13}}|13\rangle\langle 13| \\ & + e^{i\phi_{20}}|20\rangle\langle 20| + e^{i\phi_{21}}|21\rangle\langle 21| + e^{i\phi_{22}}|22\rangle\langle 22| + e^{i\phi_{23}}|23\rangle\langle 23| \\ & + e^{i\phi_{30}}|30\rangle\langle 30| + e^{i\phi_{31}}|31\rangle\langle 31| + e^{i\phi_{32}}|32\rangle\langle 32| + e^{i\phi_{33}}|33\rangle\langle 33|. \end{aligned} \tag{2}$$

However, some edges from among all the possible $2\times {}^{N}C_2$ number of edges may not exist in the actual given graph. We would, then, precompute the sum of all the up to $2 \times {}^{N}C_2$ number of edges in the graph. Call this sum s, computing which is cheap and efficient even classically. Then, we would encode the phase factor $\phi_{jk} \forall j$ and $k < N$ for each edge that does not exist in the graph as equal to s, and $\phi_{jk} = 0$, j or $k \geq N$ in the unitary U. We will assume $L = N$ for simplicity, without loss of generality, in the rest of the paper.

If we now have $V = U^{\otimes N}$, then there are $N!$ number of eigenstates, that are possible Hamiltonian cycles from among a total of N^{2N} (L^{2N} if $L > N$) number of eigenstates of V, of which $(N-1)!$ Hamiltonian cycles are unique. For example, for $N = 4$, there are a total of 4^8 number of eigenstates of V, but we have the following $(4-1)! = 6$ eigenstates representing unique Hamiltonian cycles starting from city 0:

$$\begin{aligned} |\nu\rangle &= |01122330\rangle : \varphi_\nu = \phi_{01} + \phi_{12} + \phi_{23} + \phi_{30} : 0 \to 1 \to 2 \to 3 \to 0, \\ |\nu\rangle &= |03322110\rangle : \varphi_\nu = \phi_{03} + \phi_{32} + \phi_{21} + \phi_{10} : 0 \to 3 \to 2 \to 1 \to 0, \\ |\nu\rangle &= |02211330\rangle : \varphi_\nu = \phi_{02} + \phi_{21} + \phi_{13} + \phi_{30} : 0 \to 2 \to 1 \to 3 \to 0, \\ |\nu\rangle &= |03311220\rangle : \varphi_\nu = \phi_{03} + \phi_{31} + \phi_{12} + \phi_{20} : 0 \to 3 \to 1 \to 2 \to 0, \\ |\nu\rangle &= |01133220\rangle : \varphi_\nu = \phi_{01} + \phi_{13} + \phi_{32} + \phi_{20} : 0 \to 1 \to 3 \to 2 \to 0, \\ |\nu\rangle &= |02233110\rangle : \varphi_\nu = \phi_{02} + \phi_{23} + \phi_{31} + \phi_{10} : 0 \to 2 \to 3 \to 1 \to 0. \end{aligned} \tag{3}$$

Notice that there are ${}^{N}P_N = N! = 4! = 24$ number of possible permutations of four different vertices $|0\rangle$, $|1\rangle$, $|2\rangle$, $|3\rangle$. For example, the eigenstates $|01122330\rangle$ and $|23300112\rangle$ represent the same Hamiltonian cycle, but the starting points are different, i.e. cities 0 and 2, respectively. So, there are N number of same Hamiltonian

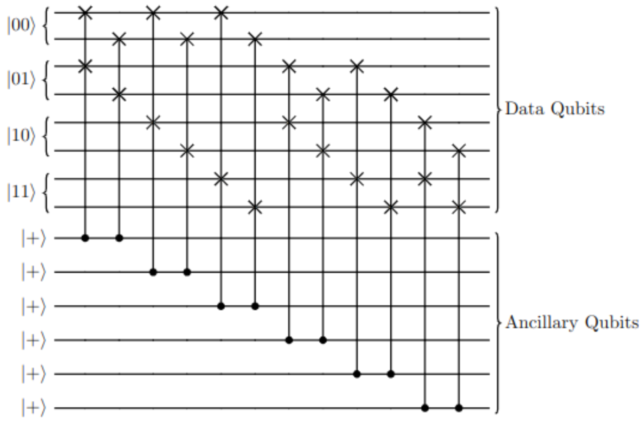

Figure 1. Quantum circuit to create a superposition of all permutations of $0, 1, 2, \ldots, N-1$ for $N = 4$.

cycles, but simply rotated with respect to each other, for each of the $(N-1)!$ unique Hamiltonian cycles, amounting to a total of $N!$ Hamiltonian cycle eigenstates of V for a fully-connected directed graph.

Now, we initialize N number of data registers, each of $\lceil \log N \rceil$ qubits, to $|0\rangle, |1\rangle, \ldots, |N-1\rangle$, respectively. Then, we create all possible permutations of $0, 1, \ldots, N-1$ by using ${}^N C_2$ number of single-qubit ancilla registers, each initialized to state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and applying $\lceil \log N \rceil$ number of controlled swap gates on each of every combination of two of the N data registers with one ancilla register as the control qubit. Please see the circuit in Figure 1.

We will then have the following state in the data registers, upon tracing out the ancilla registers, with $\sum_\mu \beta_\mu^2 = 2^{{}^N C_2}$:

$$\rho = \frac{1}{2^{{}^N C_2}} \sum_\mu \beta_\mu^2 |\mu\rangle\langle\mu|. \tag{4}$$

We, then, expand the state ρ from an N-register state to a $2N$-register state. For example, for $N = 4$, we augment a state $|0123\rangle$ to the state $|01122330\rangle$, so that it forms a valid eigenstate for the unitary V. We can augment the N-register state by adding N new registers, each of $\lceil \log N \rceil$ qubits initialised to $|0\rangle$, at appropriate places, and applying $CNOT$ gates to create adjacent copies of every register in ρ. This way, we will now have terms like $|01122330\rangle$ replacing $|0123\rangle$ (for $N = 4$) in ρ. Please see Figure 2. Call this new state σ, so that we have, with $\sum_\nu \beta_\nu^2 = 2^{{}^N C_2}$:

$$\sigma = \frac{1}{2^{{}^N C_2}} \sum_\nu \beta_\nu^2 |\nu\rangle\langle\nu|. \tag{5}$$

We use this state σ as the input eigenstates to perform quantum phase estimation of the unitary V, to obtain the following state at the output:

$$\gamma = \frac{1}{2^{{}^N C_2}} \sum_\nu \beta_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu|. \tag{6}$$

The state γ then has all the valid Hamiltonian cycle eigenstates $|\nu\rangle$, along with the estimates $\tilde{\varphi}_\nu$ of their corresponding (normalised-) sums φ_ν of weights of constituent edges ϕ_{jk}.

Notice that we use the improved quantum phase estimation method from Ref. [19] in the phase estimation of V, and so, we would have the time variable $t = O(\eta\kappa/\varepsilon)$ (the factor $\eta\kappa$ arises as a result of using $C = \eta\kappa$ in the controlled rotation later) when simulating the unitary U in the beginning by exponentiating a diagonal matrix Φ, such that $U = e^{i\Phi t}$. The matrix Φ to exponentiate to obtain unitary U of (2) is:

$$\begin{aligned}\Phi = &\phi_{00}|00\rangle\langle 00| + \phi_{01}|01\rangle\langle 01| + \phi_{02}|02\rangle\langle 02| + \phi_{03}|03\rangle\langle 03| \\ &+ \phi_{10}|10\rangle\langle 10| + \phi_{11}|11\rangle\langle 11| + \phi_{12}|12\rangle\langle 12| + \phi_{13}|13\rangle\langle 13| \\ &+ \phi_{20}|20\rangle\langle 20| + \phi_{21}|21\rangle\langle 21| + \phi_{22}|22\rangle\langle 22| + \phi_{23}|23\rangle\langle 23| \\ &+ \phi_{30}|30\rangle\langle 30| + \phi_{31}|31\rangle\langle 31| + \phi_{32}|32\rangle\langle 32| + \phi_{33}|33\rangle\langle 33|,\end{aligned}$$

which, being diagonal, and so, sparse, U can be simulated efficiently [19, 20]. Here, $\varepsilon/2$ is the estimation precision error in trace distance, and so, the maximum probability of estimation error. We use $\varepsilon = O(1/\text{poly}(N))$ to let time t to simulate U be polynomial, and not exponential, in N, as long as the error probability of our overall algorithm is below $1/3$.

In order to find the shortest Hamiltonian cycle, we add to γ an ancilla qubit, initialized in the state $(|0\rangle\langle 0| + |1\rangle\langle 1|)/2$, and rotate it, conditioned on $|\tilde{\varphi}_\nu\rangle$, to get, with $\alpha_\nu^2 := \beta_\nu^2/2^{{}^N C_2}$:

$$\begin{aligned}\xi = &\sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \\ &\otimes \left[(1 - C^2\tilde{\varphi}_\nu^2)|0\rangle\langle 0| + C^2\tilde{\varphi}_\nu^2|1\rangle\langle 1|\right].\end{aligned} \tag{7}$$

We ensure that all the eigenphases $\tilde{\varphi}_\nu$ are normalised to be between 0 and 1 by initially dividing all edges ϕ_{jk} by $\hat{\varphi}_{\max}$, that is taken as the sum of the N largest edges as an estimate of $\varphi_{\max}$. If $\hat{\varphi}_{\max}$ is equal to $\varphi_{\max}$ and if the phase estimation was perfect, then we have the minimum $\tilde{\varphi}_\nu$, which is $\tilde{\varphi}_{\min}$, equal to the inverse of the not-too-large condition number (of the matrix of which φ_ν's are the eigenvalues of only the valid Hamiltonian cycle eigenstates of V), $1/\kappa = \varphi_{\min}/\varphi_{\max}$. Then, the ancilla qubit above will be rotated from $|0\rangle$ only for $\tilde{\varphi}_{\min}$ that is equal to $1/\kappa$, if $C = \kappa$, since the probability $C^2\tilde{\varphi}_\nu^2$, attached to $|1\rangle$, will be equal to 1 only for $\tilde{\varphi}_{\min}$. That is, the ancilla qubit will not be rotated at all for any $\tilde{\varphi}_\nu$, other than $\tilde{\varphi}_{\min}$. However, since $\hat{\varphi}_{\max}$ would almost always be an overestimate of $\varphi_{\max}$, we would have $1/\kappa$ rather larger than $\tilde{\varphi}_{\min}$, more so when the variance in the given edges is large, or there are missing edges initially assigned a value of s. This is why we use an extra multiplicative factor η, that we discuss later, to ensure that only $\tilde{\varphi}_{\min}$ is likely going to be less than or equal to $1/(\eta\kappa)$. So, in order to find and separate out $\tilde{\varphi}_{\min}$, we choose above $C = \eta\kappa$, so that the ancilla qubit is rotated from $|0\rangle$ only for $\tilde{\varphi}_{\min}$ that is less than or equal to $1/(\eta\kappa)$. We will discuss shortly how to guess the value of $\eta\kappa$ to use.

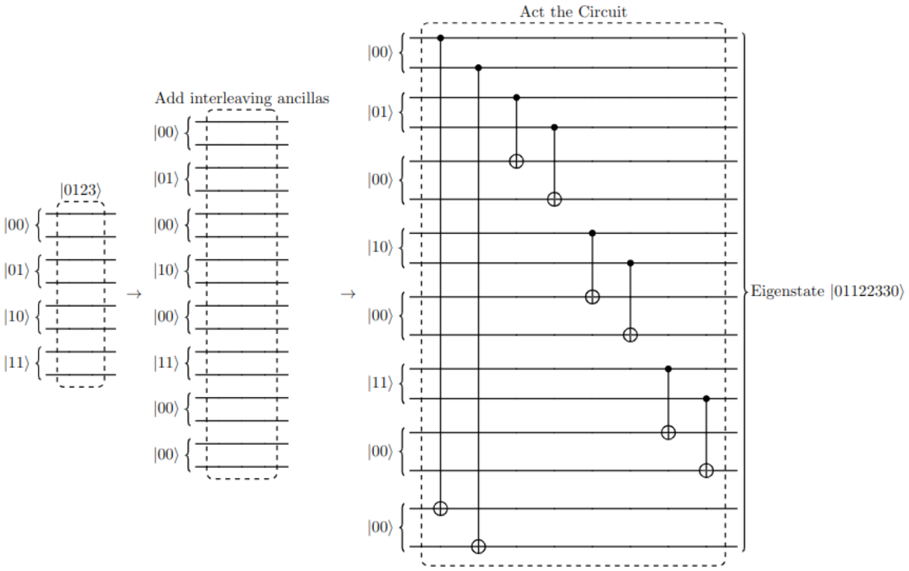

Figure 2. Quantum circuit to augment terms like $|0123\rangle$ in ρ to $|01122330\rangle$ in σ for $N = 4$.

We then need to find the Hamiltonian cycle $|\nu\rangle$, that corresponds to $\tilde{\varphi}_{\min}$. Note that if $C\tilde{\varphi}_{\min} = O(1)$, the probability of obtaining 1 as the outcome of measuring the ancilla qubit above can be as small as $O(1/(N-1)!) \geq O(1/2^{{}^{N}C_2}) = O(1/2^{O(N^2)})$, which can be more than exponentially small in N. So, we cannot efficiently perform a postselection on the outcome being 1 of measuring the ancilla like in Ref. [19], to obtain the below desired state from the state ξ in (7):

$$\zeta = \sum_{\tilde{\varphi}_{\min}} \frac{\alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2}{\lambda_{\min}} |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu|, \qquad (8)$$

where if only $\tilde{\varphi}_{\min}$ is less than $1/(\eta\kappa) = 1/C$, then we would have N terms in the summation, all of which are the same Hamiltonian cycles (with the same sum of weights of edges), just rotated with respect to each other, so that $\lambda_{\min} := \sum_{\tilde{\varphi}_{\min}} \alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2$. We would not use amplitude amplification either, as used in Ref. [19], since it gives only a quadratic speedup, while we want to get exponential speedup.

Instead, in order to get the above state ζ in (8), we first exponentiate the ancilla qubit (call it ϱ) from (7) to get the unitary $Y = e^{i\varrho\tau}$, and perform phase estimation on Y for eigenstate $|1\rangle$. We create Y by repeatedly applying the below to state ϱ of ancilla "anc" and another state ς [21, 22]:

$$\mathrm{Tr}_{\mathrm{anc}}\left[e^{i\mathcal{S}\Delta\tau}(\varrho\otimes\varsigma)e^{-i\mathcal{S}\Delta\tau}\right] = \varsigma - i\Delta\tau[\varrho,\varsigma] + O(\Delta\tau^2). \qquad (9)$$

Here, $\mathcal{S}$ is the swap operator, which is sparse and so, $e^{i\mathcal{S}\Delta\tau}$ can be performed efficiently [19, 20]. Also, here, $\tau = n\Delta\tau$, where $n = O(\tau^2/\epsilon)$ is the required number of copies of ϱ, and so, the required number of times (9) needs to be repeated to simulate Y with an error of ϵ. The phase estimation process requires controlled-$e^{i\varrho\tau}$ operations $\sum_{\iota=0}^{T-1}|\iota\rangle\langle\iota| \otimes e^{i\varrho\tau\iota/T}$ (for some T), that is done by acting the conditional swap operator $|\iota\rangle\langle\iota| \otimes e^{i\mathcal{S}\tau\iota/T}$ on $|\iota\rangle\langle\iota| \otimes \varrho \otimes \varsigma$ instead, and tracing out ϱ. Note that the eigenvalues of $\mathcal{S}$ are ±1, so that $W^{1/T} = W^{2^1/T} = W^{2^2/T} = W^{2^3/T} \ldots$, where $W = e^{i\mathcal{S}\tau}$, if we use say $\tau = 2\pi$.

The phase estimate so obtained would be $\tilde{\lambda}_{\min}$, using which we further apply a rotation to the ancilla qubit in (7), to get:

$$\chi = \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \\ \otimes \left[\left(1 - \frac{C^2\tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}}\right)|0\rangle\langle 0| + \frac{C^2\tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}}|1\rangle\langle 1|\right], \qquad (10)$$

where since the eigenvalue of the eigenstate $|1\rangle$ of the ancilla qubit is evidently $\sum_\nu \frac{\alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}} = 1$, we get the eigenvalue of the eigenstate $|0\rangle$ of the ancilla qubit to be $\sum_\nu \alpha_\nu^2 \left(1 - \frac{C^2\tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}}\right) = \sum_\nu \alpha_\nu^2 - \sum_{\tilde{\varphi}_{\min}} \frac{\alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}} = 1-1 = 0$. Thus, the above state, upon tracing out the ancilla qubit, is just the desired state ζ from (8), where every $\tilde{\varphi}_\nu$ is equal to $\tilde{\varphi}_{\min}$, so that measuring the two registers yield the outputs, $\tilde{\varphi}_{\min}$ and an eigenstate ν, corresponding to the desired shortest Hamiltonian cycle in the graph. Clearly, if $\tilde{\varphi}_{\min}$ so obtained is larger than $s/\hat{\varphi}_{\max}$, the eigenstate ν so obtained would not be a Hamiltonian cycle, since it would have at least one missing edge assigned a normalised value of $s/\hat{\varphi}_{\max}$. So, we decide that there is no Hamiltonian cycle in the graph. Otherwise, we output the obtained ν and $\tilde{\varphi}_{\min} \times \hat{\varphi}_{\max}$ as the shortest Hamiltonian cycle and the sum of its edges, respectively.

Note that the controlled rotation in (7) is achieved by considering two ancilla qubits, denoted by A and B, initialised in the state $|0_A 0_B\rangle$, and then rotating this state, conditioned on phase estimates $|\tilde{\varphi}_\nu\rangle$, and ignoring one of the ancilla qubits:

$$\xi = \mathrm{Tr}_B \left\{ \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \otimes |\Phi_{AB}\rangle\langle\Phi_{AB}| \right\}, \quad (11)$$

where $|\Phi_{AB}\rangle := \sqrt{1 - C^2\tilde{\varphi}_\nu^2}|0_A 0_B\rangle + C\tilde{\varphi}_\nu |1_A 1_B\rangle$. In order to simulate the unitary Y, we perform density matrix exponentiation of the state ϱ of the ancilla qubit (as pointed out earlier), that requires $n = O(\tau^2/\epsilon)$ copies of ϱ. Clearly, we get two copies of ϱ from above, one by ignoring the qubit B, and the other by ignoring the qubit A, but undoing the action of $e^{i\varsigma\tau}$ on ϱ in between. Thus, we get n copies of ϱ, required for simulation of Y, by considering n ancilla qubits in the state $|0^{\otimes n}\rangle$, and then rotating this state, conditioned on $|\tilde{\varphi}_\nu\rangle$:

$$\vartheta = \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \otimes |\psi\rangle\langle\psi|, \quad (12)$$

where $|\psi\rangle := \sqrt{1 - C^2\tilde{\varphi}_\nu^2}|0^{\otimes n}\rangle + C\tilde{\varphi}_\nu |1^{\otimes n}\rangle$. With n ancilla qubits, the state corresponding to χ of (10) is:

$$\Theta = \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \otimes |\Psi\rangle\langle\Psi|, \quad (13)$$

where $|\Psi\rangle := \sqrt{1 - \frac{C^2\tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}}}|0^{\otimes n}\rangle + \frac{C\tilde{\varphi}_\nu}{\sqrt{\tilde{\lambda}_{\min}}}|1^{\otimes n}\rangle$.

We discuss an *empirical* way to guess $C = \eta\kappa$. The *mid-range* of the means (of N edges) of the Hamiltonian cycles is normally close or equal to the *mean* M of all edges. We use an extra factor $\omega/(2N)$ to offset any deviation of $2NM$ below $\hat{\varphi}_{\max}$, to get: $(\varphi_{\min}/N + \hat{\varphi}_{\max}/N)/2 = \omega M/(2N)$, with $\omega = \max(2N, (e! = 0)\theta)$, where $\theta = {}^N C_2$ for undirected graphs, $\theta/2 = {}^N C_2$ for directed graphs, e is number of too large (at least ten times the smallest edge) or missing edges. So, a guess of $1/\kappa$ is $1/\hat{\kappa} = \omega M/\hat{\varphi}_{\max} - 1$. We use $\eta = \Omega/\varpi$, if $\omega > 2N$, $\eta = (\Omega/\varpi)\cdot(\omega/(2N))$, if $\omega > 4N$, else $\eta = 1$, if $\omega = 2N$, where Ω is the sum of all edges including missing edges, $\varpi \leq s$ is the sum of all edges excluding missing or too large edges. Usually, $1/C = 1/(\eta\hat{\kappa})$ would be below, yet close to, *normalised* $\varphi_{\min}$, so that we repeat our algorithm a few times, slightly raising $1/C$ each time till we capture $\tilde{\varphi}_{\min}$ only. We find that this empirical method rarely fails [23].

III. ALGORITHM

1. We create the unitary $U = e^{i\Phi t}$ encoding all weights of edges ϕ_{jk}, where $\Phi = \sum_{j,k=0}^{L-1} \frac{\phi_{jk}}{\hat{\varphi}_{\max}} |jk\rangle\langle jk|$, $\hat{\varphi}_{\max}$ is sum of N largest edges upon replacing missing edges $\forall j,k < N$ by s, s is sum of all given edges, $\log L = \lceil \log N \rceil$. We create N copies of U for $V = U^{\otimes N}$, since the number of edges in every Hamiltonian cycle is exactly N.

2. We initialize N number of data registers of $\lceil \log N \rceil$ qubits each to $|0\rangle, |1\rangle, |2\rangle, \ldots, |N-1\rangle$, respectively. We generate all permutations of $0, 1, 2, \ldots, N-1$ using ${}^N C_2$ number of single-qubit ancilla registers, each initialized to $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and applying $\lceil \log N \rceil$ number of controlled swap gates on each combination of two data registers with one ancilla as control qubit.

3. The first register has the state $\rho = \frac{1}{2^{{}^N C_2}} \sum_\mu \beta_\mu^2 |\mu\rangle\langle\mu|$, where $\sum_\nu \beta_\mu^2 = 2^{{}^N C_2}$. We augment this N-register state by adding N ancilla registers each of $\lceil \log(N) \rceil$ qubits in state $|0\rangle$ and acting $CNOT$ gates to get a $2N$-register state $\sigma = \frac{1}{2^{{}^N C_2}} \sum_\nu \beta_\nu^2 |\nu\rangle\langle\nu|$, where $\sum_\nu \beta_\nu^2 = 2^{{}^N C_2}$.

4. We perform quantum phase estimation [19] on V using σ as input to get $\gamma = \frac{1}{2^{{}^N C_2}} \sum_\nu \beta_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu|$.

5. We add to the state γ an ancilla register of n qubits, initialized in the state $|0^{\otimes n}\rangle$, and rotate it, conditioned on $|\tilde{\varphi}_\nu\rangle$, to get the state $\vartheta = \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \otimes |\psi\rangle\langle\psi|$, where $\alpha_\nu^2 := \beta_\nu^2 / 2^{{}^N C_2}$, $|\psi\rangle := \left[\sqrt{1 - C^2\tilde{\varphi}_\nu^2}|0^{\otimes n}\rangle + C\tilde{\varphi}_\nu |1^{\otimes n}\rangle\right]$ and C is set to $\eta\kappa$.

6. We exponentiate the effective state ϱ of each ancilla qubit, using n copies of ϱ from n ancilla qubits, to get a unitary $Y := e^{i\varrho\tau}$, and perform phase estimation on Y for the eigenstate $|1\rangle$. If the phase estimate for $|1\rangle$ is 0, we repeat step 5 with a slightly smaller value of C, and then repeat this step, until we can capture $\tilde{\varphi}_{\min}$ in step 5 and get $\tilde{\lambda}_{\min}$ as the phase estimate in this step.

7. We further apply a rotation to the ancilla register in the state ϑ from step 5, using $\tilde{\lambda}_{\min}$ obtained in step 6, to get $\Theta = \sum_\nu \alpha_\nu^2 |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu| \otimes |\Psi\rangle\langle\Psi|$, where $|\Psi\rangle := \left[\sqrt{1 - \frac{C^2\tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}}}|0^{\otimes n}\rangle + \frac{C\tilde{\varphi}_\nu}{\sqrt{\tilde{\lambda}_{\min}}}|1^{\otimes n}\rangle\right]$. So, upon tracing out the ancilla register $|\Psi\rangle$ from Θ, we get $\zeta = \sum_{\tilde{\varphi}_{\min}} \frac{\alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2}{\lambda_{\min}} |\tilde{\varphi}_\nu\rangle\langle\tilde{\varphi}_\nu| \otimes |\nu\rangle\langle\nu|$, as $\sum_\nu \frac{\alpha_\nu^2 C^2 \tilde{\varphi}_\nu^2}{\tilde{\lambda}_{\min}} = 1$.

8. Measure the two registers in the state ζ from step 7 to get $\tilde{\varphi}_{\min}$ and a ν. If $\tilde{\varphi}_{\min}$ is larger than $s/\hat{\varphi}_{\max}$, output the decision that there is no Hamiltonian cycle in the graph. Otherwise, output the obtained values of ν and $\tilde{\varphi}_{\min} \times \hat{\varphi}_{\max}$ as the desired shortest Hamiltonian cycle of the graph, and the sum of its edges, respectively.

The improved quantum phase estimation method from Ref. [19], that we use above, is as follows. We start with an initial state $|\Lambda_0\rangle|u_j\rangle$, where $|u_j\rangle$ is the j-th eigenstate of the Hermitian matrix Γ, that we exponentiate, and

$|\Lambda_0\rangle := \sqrt{\frac{2}{T}} \sum_{\iota=0}^{T-1} \sin \frac{\pi(\iota+\frac{1}{2})}{T} |\iota\rangle$ for some large T. The state $|\Lambda_0\rangle$ can be prepared upto some error ϵ_Λ in time $\mathrm{poly}\log(T/\epsilon_\Lambda)$ (see Ref. [19]). We apply the conditional Hamiltonian evolution $\sum_{\iota=0}^{T-1} |\iota\rangle\langle\iota| \otimes e^{i\Gamma\iota t/T}$ on the initial state in both registers, and then apply inverse quantum Fourier transform on the first register to get the state $\sum_{q=0}^{T-1} \upsilon_{q|j} |q\rangle |u_j\rangle$. Defining the estimate $\tilde{r}_q$ of the q-th eigenvalue r_q of Γ as $\tilde{r}_q := \frac{2\pi q}{t}$, we relabel the Fourier basis states $|q\rangle$ to get $\sum_{q=0}^{T-1} \upsilon_{q|j} |\tilde{r}_q\rangle |u_j\rangle$. If the phase estimation is perfect, we have $\upsilon_{q|j} = 1$ if $\tilde{r}_q = r_j$, and 0 otherwise. So, we get the state $|\tilde{r}_j\rangle|u_j\rangle$, from which we obtain the estimate of r_j upon measuring the first register.

IV. ALGORITHM COMPLEXITY

In Figure 1 for step 2, the swap gates on $\lceil \log N \rceil$ number of qubits of each data register are applied parallelly. Since there are ${}^{N}C_2$ number of such sets of swap gates, the complexity of this step is $O(N^2)$. The complexity is independent of N in Figure 2 for step 3, since all the $CNOT$ gates can be applied in parallel. In step 1, creating each copy of U has a complexity of $O(2\log(N)t)$ [19, 20], assuming $L = N$, where each eigenstate has $2\log L$ qubits. Further, N copies of U, required for V, can be created in parallel. The use of improved quantum phase estimation from Ref. [19] in step 4 along with the controlled rotation with $C = \eta\kappa$ in step 5 require the time variable t in step 1 to be $O(\eta\kappa/\varepsilon)$ (see Ref. [19]). In step 5, the complexity is polynomial for the controlled rotation of the ancilla register state, as in Ref. [19]. The circuit depth of the density matrix exponentiation in step 6 of the (single) ancilla qubit is $O(\log(2)n) = O(\tau^2/\epsilon) = O(1/\epsilon^3)$ [21, 22], where ϵ is the simulation error for Y, as also the precision error of improved phase estimation of Y for which $\tau = O(1/\epsilon)$. Here, ϵ or $\epsilon/2$ is an error in trace distance, and so, determines the maximum error probability in simulation or estimation of Y [24]. The eigenvalue of ϱ for eigenstate $|1\rangle$ can be as low as $O(1/(N-1)!)$, that is indistinguishable from 0 with less than $O(\log((N-1)!))$ phase estimate qubits, that otherwise leads to $\tau \geq O((N-1)!)$ in usual phase estimation [1]. Since $\epsilon/2$ is maximum error probability, we use $\epsilon = O(1/\mathrm{poly}(N))$, not $\epsilon = O(1/(N-1)!)$, but with $T = O((N-1)!) \leq O(2^{O(N^2)})$ in *improved* phase estimation, so that $\tau = O(\mathrm{poly}(N))$, as long as the cumulative error probability of our algorithm is 1/3 or lower. The complexities of steps 7 and 8 can be ignored. So, the overall complexity of our algorithm is $O(2\log(N)t)$, which yields $O(\log(N)\eta\kappa/\epsilon) \leq O(N^3 \log(N)\kappa/\epsilon)$, taking $\varepsilon = \epsilon$ for simplicity, with $\epsilon = O(1/\mathrm{poly}(N))$, and $\eta \leq O(N^3)$ (since $\Omega/\varpi \leq O(N^2 s/s) = O(N^2)$ and $\omega/(2N) \leq O(N^2/N) = O(N)$). This is because the complexity of step 1 is dominant amongst all steps, if it suffices to have $\epsilon \geq O(1/N)$. Also, $\kappa = O(\mathrm{poly}(N))$, since it is geometrically unlikely, if not impossible, for a Hamiltonian cycle of N edges to be exponentially larger than another Hamiltonian cycle of N edges in a city-network graph, even with too-large edges, as each Hamiltonian cycle must visit all cities, including too-far ones. When there are $e \leq O(N^2)$ missing edges, we have $\hat{\kappa} = \hat{\varphi}_{\max}/((e+1)s - \hat{\varphi}_{\max})$, where $\hat{\varphi}_{\max} < ps + O(10(N-p)/\theta)s$ with $p = \min(N, e)$ (since an edge at least 10 times the smallest edge is treated too-large), so that with $\theta = O(N^2)$, we get $\hat{\kappa} < O((N^2 p + 10(N-p))/(N^2(e+1-p) - 10(N-p))) = O(\mathrm{poly}(N))$. Note that we need to classically precompute $\hat{\varphi}_{\max}$ by summing the N largest edges, for which we sort the data, that is expensive. If we use merge sort to sort the ${}^{N}C_2$ number of edges, the complexity is $O({}^{N}C_2 \log({}^{N}C_2)) = O(N^2 \log(N))$, which is less than $O(N^3 \log(N)\kappa/\epsilon)$. But if $\epsilon \leq O(1/N^2)$, the complexity of step 6 is dominant, so the overall complexity of our algorithm is at least $O(1/\epsilon^3) = O(N^6)$. This is an exponential speedup over the brute-force complexity of $O(N^N)$ for undirected [1], or $O(N^{2N})$ for directed graphs.

V. DISCUSSION

Note that if there is a city that is very far from the rest of the cities in the network, every Hamiltonian cycle will need to traverse to that far-off city and back, since a Hamiltonian cycle must visit all cities. This is why it is geometrically unlikely for the ratio κ of the longest Hamiltonian cycle to the shortest Hamiltonian cycle to be exponential in N. However, it is not absolutely impossible. It is possible that one of the Hamiltonian cycles has a path to that far-off city via a highly zig-zag route or requires having to cross a mountain, for example. In this case, it is indeed possible for that Hamiltonian cycle to be exponentially larger than the shortest Hamiltonian cycle. Nonetheless, this is practically irrelevant, even if not absolutely unlikely, since the salesman is anyway not going to be crossing the mountain, as he would normally be aware of such an unusually long path that exists and would naturally avoid it anyway. Under these unusual or unlikely circumstances, our empirical method to guess C would yield a very poor estimate of C, and the algorithm complexity can become exponential in N.

We also mentioned that our empirical method rarely fails. Please see Ref. [23], where we indicate in red the rare cases that fail from among the many examples we consider, while in most cases, indicated in green, it succeeds. These rare cases are when there can be many eigenvalues $\tilde{\varphi}_\nu$, and not just the smallest one $\tilde{\varphi}_{\min}$, that are lower than $1/C$, and therefore, can get picked up by our controlled rotation. Usually, such a situation may arise in cases of highly skewed data. Otherwise, as can be seen from the many successful cases, indicated in green, our empirical method usually yields an estimate of $1/C$ that is lower than, but very close to, the $\tilde{\varphi}_{\min}$ we are after. This means that if we are unable to capture $\tilde{\varphi}_{\min}$, we need to repeat our algorithm upon slightly raising the estimated value of $1/C$, until we are able to capture $\tilde{\varphi}_{\min}$.

Since our empirical method normally yields an estimate of $1/C$ very close to $\tilde{\varphi}_{\min}$, it would usually take only a few repetitions of our algorithm, until the value of $1/C$ used exceeds $\tilde{\varphi}_{\min}$, so that we are able to capture it. Evidently, the number of repetitions required would depend on the size of the chosen increment in the value of $1/C$ in every iteration, compared to the value of $\tilde{\varphi}_{\min}$. The size of the increment needs to be chosen carefully, so that in trying to minimize the number of repetitions, we do not end up raising $1/C$ above multiple (unequal) $\tilde{\varphi}_{\nu}$'s and not just $\tilde{\varphi}_{\min}$, in which case the algorithm would not be able to capture the shortest Hamiltonian cycle only, and may, therefore, not yield the right answer in the end.

Appendix A gives an example of our entire algorithm.

VI. CONCLUSION

We presented a bounded-error quantum polynomial-time (BQP) algorithm for solving the NP-hard traveling salesman problem. The worst-case brute-force time complexity for this problem is $O(N^N)$ for undirected graphs, and $O(N^{2N})$ for directed graphs, where N is the number of vertices in the graph. The quantum speedup known to be achievable for this problem is quadratic only, using Grover's search, yielding a complexity of $O(N^{N/2})$ for undirected graphs or $O(N^N)$ for directed graphs, which are still exponential in N. On the contrary, our algorithm presented here provides with an exponential quantum speedup, by solving the problem in quantum polynomial time. The overall time complexity of our algorithm is $O(N^3 \log(N)\kappa/\epsilon + 1/\epsilon^3)$, which is indeed polynomial, and not exponential, in N, where the errors ϵ need not be exponentially small, and can be polynomially small, as long as the overall cumulative probability of error of our algorithm does not exceed 1/3.

ACKNOWLEDGMENTS

A. S. thanks Palak Chawla for discussions. S. D. acknowledges financial support from the European Union's Horizon 2020 research and innovation programme under FET-OPEN Grant Agreement No. 828946 (PATHOS) and from the European Commission's Horizon Europe Framework Programme under Research and Innovation Action Grant Agreement No. 101070546 (MUQUABIS).

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2002).

[2] G. Dantzig, R. Fulkerson, and S. Johnson, Solution of a large-scale traveling-salesman problem, Operations Research **2**, 393 (1954).

[3] M. Grötschel and O. Holland, Solution of large-scale symmetric travelling salesman problems, Mathematical Programming **51**, 141 (1991).

[4] M. Padberg and G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, SIAM Review **33**, 60 (1991).

[5] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, On the solution of traveling salesman problems, Documenta Mathematica **3**, 645 (1998).

[6] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, An algorithm for the traveling salesman problem, Operations Research **11**, 972 (1963).

[7] M. Held and R. M. Karp, The traveling-salesman problem and minimum spanning trees: Part II, Mathematical Programming **1**, 6 (1971).

[8] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems, Mathematical Programming **97**, 91 (2003).

[9] V. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications **45**, 41 (1985).

[10] H. He, Quantum annealing and graph neural networks for solving TSP with QUBO (2024), arXiv:2402.14036.

[11] R. Martoňák, G. E. Santoro, and E. Tosatti, Quantum annealing of the traveling-salesman problem, Physical Review E **70**, 057701 (2004).

[12] E. Rodriguez, E. Osaba, and I. Oregi, Analyzing the behaviour of D-Wave quantum annealer: fine-tuning parameterization and tests with restrictive Hamiltonian formulations, in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)* (2022) pp. 938–946.

[13] W. Qian, R. A. M. Basili, M. M. Eshaghian-Wilner, A. Khokhar, G. Luecke, and J. P. Vary, Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem, Entropy **25**, 1238 (2023).

[14] A. Matsuo, Y. Suzuki, I. Hamamura, and S. Yamashita, Enhancing VQE convergence for optimization problems with problem-specific parameterized quantum circuits, IEICE Transactions on Information and Systems **E106.D**, 1772 (2023).

[15] J. Bang, J. Ryu, C. Lee, S. Yoo, J. Lim, and J. Lee, A quantum heuristic algorithm for the traveling salesman problem, Journal of the Korean Physical Society **61**, 1944 (2013).

[16] K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience (2018), arXiv:1805.10928.

[17] A. Moylett, N. Linden, and A. Montanaro, Quantum speedup of the traveling-salesman problem for bounded-degree graphs, Physical Review A **95**, 032323 (2017).

[18] C. Tszyunsi and I. I. Beterov, A quantum algorithm for solving the travelling salesman problem by quantum phase estimation and quantum search, Journal of Experimental and Theoretical Physics **137**, 210 (2023).

[19] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Physical Review

Letters **103**, 150502 (2009).

[20] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse Hamiltonians, Communications in Mathematical Physics **270**, 359 (2006).

[21] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nature Physics **10**, 631 (2014).

[22] M. Kjaergaard, M. E. Schwartz, A. Greene, G. O. Samach, A. Bengtsson, M. O'Keeffe, *et al.*, Demonstration of density matrix exponentiation using a superconducting quantum processor, Physical Review X **12**, 011005 (2022).

[23] TSP-Data, Tests of our empirical method for TSP: https://shorturl.at/npLN7.

[24] S. Kimmel, C. Y.-Y. Lin, G. H. Low, M. Ozols, and T. J. Yoder, Hamiltonian simulation with optimal sample complexity, npj Quantum Information **3**, 10.1038/s41534-017-0013-7 (2017).

Appendix A: Example of Algorithm for $N = 4$

Let us now run our entire algorithm through an example directed graph with $N = 4$. Then, the nodes of the graph are denoted by 0, 1, 2, 3, respectively. Let the given edges be $\phi_{01} = 10$, $\phi_{02} = 12$, $\phi_{03} = 15$, $\phi_{12} = 29$, $\phi_{13} = 10$, $\phi_{23} = 25$, $\phi_{10} = 28$, $\phi_{30} = 20$, $\phi_{21} = 42$, $\phi_{32} = 73$, and let the edges 20 and 31 be missing, so that we have $\phi_{20} = \phi_{31} = s$, where s is the sum of the given edges $s = \phi_{01} + \phi_{02} + \phi_{03} + \phi_{12} + \phi_{13} + \phi_{23} + \phi_{10} + \phi_{30} + \phi_{21} + \phi_{32} = 264$. Then, the lengths of the $(N-1)! = 6$ (unique) Hamiltonian cycle (HC) eigenstates from (3) are: $\varphi_{01122330} = 84$, $\varphi_{03322110} = 158$, $\varphi_{02211330} = 84$, $\varphi_{03311220} = 572$, $\varphi_{01133220} = 357$, $\varphi_{02233110} = 329$, out of which $\varphi_{03311220}$, $\varphi_{01133220}$, $\varphi_{02233110}$ are valid HC eigenstates, but not of interest to us, since they contain edges that are missing in the given graph. Moreover, we can see that there are two shortest HCs of the same length, viz. $\varphi_{01122330} = \varphi_{02211330} = 84 = \varphi_{\min}$, while the largest HC, even if not of interest to us, is $\varphi_{03311220} = 572 = \varphi_{\max}$. Our estimate of $\varphi_{\max}$ is $\hat{\varphi}_{\max} = 42 + 73 + 264 + 264 = 643$. Thus, in step 1 of our algorithm, all the edge-weights are normalized by dividing by this value of $\hat{\varphi}_{\max}$. Then, we get in (5):

$$\sigma = \frac{1}{64}\left[|\nu_1\rangle\langle\nu_1| + |\nu_2\rangle\langle\nu_2| + |\nu_3\rangle\langle\nu_3| + |\nu_4\rangle\langle\nu_4|\right.$$
$$+ 2|\nu_5\rangle\langle\nu_5| + 2|\nu_6\rangle\langle\nu_6| + |\nu_7\rangle\langle\nu_7| + |\nu_8\rangle\langle\nu_8|$$
$$+ |\nu_9\rangle\langle\nu_9| + |\nu_{10}\rangle\langle\nu_{10}| + 2|\nu_{11}\rangle\langle\nu_{11}| + 2|\nu_{12}\rangle\langle\nu_{12}|$$
$$+ 2|\nu_{13}\rangle\langle\nu_{13}| + 2|\nu_{14}\rangle\langle\nu_{14}| + 2|\nu_{15}\rangle\langle\nu_{15}| + 2|\nu_{16}\rangle\langle\nu_{16}|$$
$$+ 4|\nu_{17}\rangle\langle\nu_{17}| + 4|\nu_{18}\rangle\langle\nu_{18}| + 5|\nu_{19}\rangle\langle\nu_{19}| + 5|\nu_{20}\rangle\langle\nu_{20}|$$
$$\left. + 5|\nu_{21}\rangle\langle\nu_{21}| + 5|\nu_{22}\rangle\langle\nu_{22}| + 6|\nu_{23}\rangle\langle\nu_{23}| + 6|\nu_{24}\rangle\langle\nu_{24}|\right],$$

where $\nu_1 = 01122330$, $\nu_2 = 01133220$, $\nu_3 = 02211330$, $\nu_4 = 02233110$, $\nu_5 = 03311220$, $\nu_6 = 03322110$, $\nu_7 = 10022331$, $\nu_8 = 10033221$, $\nu_9 = 12200331$, $\nu_{10} = 12233001$, $\nu_{11} = 13300221$, $\nu_{12} = 13322001$, $\nu_{13} = 20011332$, $\nu_{14} = 20033112$, $\nu_{15} = 21100332$, $\nu_{16} = 21133002$, $\nu_{17} = 23300112$, $\nu_{18} = 23311002$, $\nu_{19} = 30011223$, $\nu_{20} = 30022113$, $\nu_{21} = 31100223$, $\nu_{22} = 31122003$, $\nu_{23} = 32200113$, and $\nu_{24} = 32211003$. Then, we get (in decimal) in (6) for the 6 valid HCs, assuming perfect phase estimation: $\tilde{\varphi}_{\nu_1} = \tilde{\varphi}_{\nu_{10}} = \tilde{\varphi}_{\nu_{17}} = \tilde{\varphi}_{\nu_{19}} = \frac{84}{643} = 0.131$, $\tilde{\varphi}_{\nu_2} = \tilde{\varphi}_{\nu_{12}} = \tilde{\varphi}_{\nu_{13}} = \tilde{\varphi}_{\nu_{23}} = \frac{357}{643} = 0.555$, $\tilde{\varphi}_{\nu_3} = \tilde{\varphi}_{\nu_{11}} = \tilde{\varphi}_{\nu_{16}} = \tilde{\varphi}_{\nu_{20}} = \frac{84}{643} = 0.131$, $\tilde{\varphi}_{\nu_4} = \tilde{\varphi}_{\nu_7} = \tilde{\varphi}_{\nu_{18}} = \tilde{\varphi}_{\nu_{21}} = \frac{329}{643} = 0.512$, $\tilde{\varphi}_{\nu_5} = \tilde{\varphi}_{\nu_9} = \tilde{\varphi}_{\nu_{14}} = \tilde{\varphi}_{\nu_{22}} = \frac{572}{643} = 0.890$, $\tilde{\varphi}_{\nu_6} = \tilde{\varphi}_{\nu_8} = \tilde{\varphi}_{\nu_{15}} = \tilde{\varphi}_{\nu_{24}} = \frac{158}{643} = 0.246$. So, there are 2 unique HCs of shortest length 0.131.

Now, using our empirical method, since we have $M = 792/12$, $e = 2$, $\theta = 12 > 2N = 8$, $\omega = 12$, $\eta = \Omega/\varpi = 3$, and $1/\hat{\kappa} = 149/643$, we get an estimate of $1/C$ to use as $\frac{149}{3\times 643} = 0.077$, which is less than but close to $\tilde{\varphi}_{\min} = 0.131$. So, we would need to repeat the algorithm a few times to be able to capture $\tilde{\varphi}_{\min}$; for example, if we use an increment size of 0.03 in raising the value of $1/C$ to use in each iteration, we would need to repeat our algorithm only twice to be able to successfully capture $\tilde{\varphi}_{\min}$. Clearly, the algorithm would fail on this occasion, only if the increment size chosen for $1/C$ is $0.246 - 0.077 = 0.169$ or larger. Considering that the chosen increment size is 0.03, the value of $1/C$ to use would become $0.077 + 0.06 = 0.137$ after 2 repetitions of the algorithm, so that we would have $C\tilde{\varphi}_{\min} = 0.956$ and $\lambda_{\min} = \left[\left(\frac{1}{64} + \frac{1}{64} + \frac{4}{64} + \frac{5}{64}\right) + \left(\frac{1}{64} + \frac{2}{64} + \frac{2}{64} + \frac{5}{64}\right)\right] \times (0.956)^2 = \left(\frac{11}{64} + \frac{10}{64}\right) \times 0.914 = 0.3$. Assuming that our estimate $\tilde{\lambda}_{\min}$ is equal to $\lambda_{\min}$, we get in (10):

$$\chi = \frac{C^2\tilde{\varphi}^2_{\min}}{\tilde{\lambda}_{\min}}|\tilde{\varphi}_{\min}\rangle\langle\tilde{\varphi}_{\min}| \otimes \left[\left(\frac{1}{64}|\nu_1\rangle\langle\nu_1| + \frac{1}{64}|\nu_{10}\rangle\langle\nu_{10}|\right.\right.$$
$$\left. + \frac{4}{64}|\nu_{17}\rangle\langle\nu_{17}| + \frac{5}{64}|\nu_{19}\rangle\langle\nu_{19}|\right) + \left(\frac{1}{64}|\nu_3\rangle\langle\nu_3|\right.$$
$$\left.\left. + \frac{2}{64}|\nu_{11}\rangle\langle\nu_{11}| + \frac{2}{64}|\nu_{16}\rangle\langle\nu_{16}| + \frac{5}{64}|\nu_{20}\rangle\langle\nu_{20}|\right)\right]$$
$$= \frac{0.914}{0.3}|0.131\rangle\langle 0.131| \otimes \left[\left(\frac{1}{64}|\nu_1\rangle\langle\nu_1| + \frac{1}{64}|\nu_{10}\rangle\langle\nu_{10}|\right.\right.$$
$$\left. + \frac{4}{64}|\nu_{17}\rangle\langle\nu_{17}| + \frac{5}{64}|\nu_{19}\rangle\langle\nu_{19}|\right) + \left(\frac{1}{64}|\nu_3\rangle\langle\nu_3|\right.$$
$$\left.\left. + \frac{2}{64}|\nu_{11}\rangle\langle\nu_{11}| + \frac{2}{64}|\nu_{16}\rangle\langle\nu_{16}| + \frac{5}{64}|\nu_{20}\rangle\langle\nu_{20}|\right)\right]$$
$$= |0.131\rangle\langle 0.131| \otimes [(0.048|\nu_1\rangle\langle\nu_1| + 0.048|\nu_{10}\rangle\langle\nu_{10}|$$
$$+ 0.190|\nu_{17}\rangle\langle\nu_{17}| + 0.238|\nu_{19}\rangle\langle\nu_{19}|) + (0.048|\nu_3\rangle\langle\nu_3|$$
$$+ 0.095|\nu_{11}\rangle\langle\nu_{11}| + 0.095|\nu_{16}\rangle\langle\nu_{16}| + 0.238|\nu_{20}\rangle\langle\nu_{20}|)]$$

Thus, upon measuring the first register above, we would get $\tilde{\varphi}_{\min} = 0.131$, so that we can calculate the length of the shortest Hamiltonian cycle(s) in the given graph to be $\tilde{\varphi}_{\min} \times \hat{\varphi}_{\max} = 0.131 \times 643 = 84.23$, which is close to the actual value of 84. Upon measuring the second register, we would get one of the eigenstates ν_1, ν_{10}, ν_{17}, ν_{19}, depicting one (same) shortest HC, or one of the eigenstates ν_3, ν_{11}, ν_{16}, ν_{20}, depicting the other (same) shortest HC. Clearly, we can output only either of the 2 distinct HCs, and not both, since the measurement collapses the superposition state in the second register to either of them.