Exponential Quantum Speedup for the Traveling Salesman Problem*

Anant Sharma  
Centre for Quantum Engineering, Research and Education (CQuERE)  
TCG Centres for Research and Education in Science and Technology  
Kolkata 700091, India.  
anantsharma2410@gmail.com

Nupur Deshpande  
Department of Physics  
Indian Institute of Science Education and Research  
Tirupati 517507, India.  
rajeshdeshpande@students.iisertirupati.ac.in

Sanchita Ghosh  
Centre for Quantum Engineering, Research and Education (CQuERE)  
TCG Centres for Research and Education in Science and Technology  
Kolkata 700091, India.  
sanchita.ghosh14@gmail.com

Sreetama Das  
Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO)  
European Lab. for Non-Linear Spectroscopy (LENS)  
Department of Physics and Astronomy  
University of Florence  
50019 Sesto Fiorentino, Italy.  
sreetama.das@ino.cnr.it

Shibdas Roy  
Centre for Quantum Engineering, Research and Education (CQuERE)  
TCG Centres for Research and Education in Science and Technology  
Kolkata 700091, India.  
Academy of Scientific and Innovative Research (AcSIR)  
Ghaziabad 201002, India.  
roy.shibdas@gmail.com

Abstract

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 an exponential speedup. The overall complexity of our algorithm is $O(N^3\log(N)\kappa/\epsilon + 1/\epsilon^3)$, where the errors $\epsilon$ are $O(1/poly(N))$, and $\kappa$ is the not-too-large condition number of the matrix encoding all Hamiltonian cycles.

Index Terms

Traveling Salesman Problem, Hamiltonian Cycle Problem, Quantum Algorithm, Exponential Speedup.

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 $N$ 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.

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 in Ref. [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 In-

*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 the Research and Innovation Action Grant Agreement No. 101070846 (MUQUABIS). A. S. thanks Palak Chawla for useful discussions.
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 Ref. [11] which proposed a path-integral Monte Carlo quantum algorithm that solves TSP with exponential speedup. We achieved this by using a novel quantum circuit involving controlled matrix exponentiation to find the shortest Hamiltonian cycle. We here want to find a quantum algorithm that would create the following diagonal unitary matrix:

\[
U = e^{i\phi_{00}}|00\rangle\langle 00| + e^{i\phi_{01}}|01\rangle\langle 01| + e^{i\phi_{02}}|02\rangle\langle 02| + e^{i\phi_{03}}|03\rangle\langle 03| + e^{i\phi_{10}}|10\rangle\langle 10| + e^{i\phi_{11}}|11\rangle\langle 11| + e^{i\phi_{12}}|12\rangle\langle 12| + e^{i\phi_{13}}|13\rangle\langle 13| + e^{i\phi_{20}}|20\rangle\langle 20| + e^{i\phi_{21}}|21\rangle\langle 21| + e^{i\phi_{22}}|22\rangle\langle 22| + e^{i\phi_{23}}|23\rangle\langle 23| + e^{i\phi_{30}}|30\rangle\langle 30| + e^{i\phi_{31}}|31\rangle\langle 31| + e^{i\phi_{32}}|32\rangle\langle 32| + e^{i\phi_{33}}|33\rangle\langle 33|.
\]

However, some edges from among all the possible 2×N C2 number of edges may not exist in the actual given graph. We would, then, precompute the sum of all the up to 2×N C2 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 \(\phi_{jk}\) and \(k < N\) for each edge that does not exist in the graph as equal to \(s\), and \(\phi_{jk} = 0\), \(j\) or \(k\) ≥ \(N\) in the unitary \(U\). We will assume \(L = 3\) 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 \(N^{2N} (L^{2N}) - L > N\) number of eigenstates of \(V\), of which \((N - 1)!\) Hamiltonian cycles are unique. For example, for \(N = 4\), there are a total of 48 number of eigenstates of \(V\), but we have the following \((4 - 1)! = 6\) eigenstates representing unique Hamiltonian cycles starting from city 0:

\[
|\nu\rangle = |01122330\rangle, \quad |\nu\rangle = |03322110\rangle, \quad |\nu\rangle = |02213330\rangle, \quad |\nu\rangle = |03311220\rangle, \quad |\nu\rangle = |01133220\rangle, \quad |\nu\rangle = |02233110\rangle.
\]

Notice that there are \(N!\) permutations of four different vertices \(0, 1, 2, 3\). 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 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 cycles.
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 \( \ket{0}, \ket{1}, \ldots, \ket{N-1} \), respectively. Then, we create all possible permutations of \( 0,1,\ldots,N-1 \) by using \( N \) qubits. Each initialized state to state \( \ket{+} = \frac{1}{\sqrt{2}} (\ket{0} + \ket{1}) \), 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 \( \sum_\mu \beta_\mu^2 = 2^N c^2 \):

\[
\rho = \frac{1}{2^N c^2} \sum_\mu \beta_\mu^2 \ket{\mu}
\]

We then, expand the state \( \rho \) from an \( N \)-register state to a \( 2N \)-register state. For example, for \( N = 4 \), we augment a state \( \ket{0123} \) to the state \( \ket{0112\ldots30} \), so that it forms a valid eigenstate for the unitary \( U \). We can augment the \( N \)-register state by adding \( N \) new registers, each of \( \lceil \log N \rceil \) qubits initialized to \( \ket{0} \), at appropriate places, and applying \( \text{CNOT} \) gates to create adjacent copies of every register in \( \rho \). This way, we will now have terms like \( \ket{0112\ldots30} \) (for \( N = 4 \)) in \( \rho \). Please see Figure 2. Call this new state \( \sigma \), so that we have, with \( \sum_\nu \beta_\nu^2 = 2^N c^2 \):

\[
\sigma = \frac{1}{2^N c^2} \sum_\nu \beta_\nu^2 \ket{\nu}
\]

We use this state \( \sigma \) as the input eigenstates to perform quantum phase estimation of the unitary \( U \), to obtain the following state at the output:

\[
\gamma = \frac{1}{2^N c^2} \sum_\nu \beta_\nu^2 \ket{\varphi_\nu} \otimes \ket{\nu}
\]

The state \( \gamma \) then has all the valid Hamiltonian cycle eigenstates \( \ket{\nu} \), along with the estimates \( \varphi_\nu \) of their corresponding (normalised-) sums \( \varphi_\nu \) of weights of constituent edges \( \phi_{jk} \).

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 \( t = O(\eta \kappa / \epsilon) \) (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 \( \Phi \), such that \( U = e^{\Phi t} \). The matrix \( \Phi \) to exponentiate to obtain unitary \( U \) of (2) is:

\[
\Phi = \phi_{00} \ket{00}\bra{00} + \phi_{01} \ket{01}\bra{01} + \phi_{02} \ket{02}\bra{02} + \phi_{03} \ket{03}\bra{03} + \phi_{10} \ket{10}\bra{10} + \phi_{11} \ket{11}\bra{11} + \phi_{12} \ket{12}\bra{12} + \phi_{13} \ket{13}\bra{13} + \phi_{20} \ket{20}\bra{20} + \phi_{21} \ket{21}\bra{21} + \phi_{22} \ket{22}\bra{22} + \phi_{23} \ket{23}\bra{23} + \phi_{30} \ket{30}\bra{30} + \phi_{31} \ket{31}\bra{31} + \phi_{32} \ket{32}\bra{32} + \phi_{33} \ket{33}\bra{33},
\]

which, being diagonal, and so, sparse, \( U \) can be simulated efficiently [19], [20]. Here, \( \epsilon / 2 \) is the estimation precision error in trace distance, and so, the maximum probability of estimation error. We use \( \epsilon = O(1/poly(N)) \) 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 \( \gamma \) an ancilla qubit, initialized in the state \( \ket{0112\ldots30} \) replacing \( \ket{0} \) (for \( N = 4 \)), and apply it, conditioned on \( \ket{\varphi_\nu} \), to get, with \( \alpha_\nu^2 := \beta_\nu^2 / 2^N c^2 \):

\[
\xi = \sum_\nu \alpha_\nu^2 \ket{\varphi_\nu} \otimes \ket{\nu} \otimes \left[ (1 - C^2 \varphi_\nu^2) \ket{0} \bra{0} + C^2 \varphi_\nu^2 \ket{1} \bra{1} \right]
\]

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

We then need to find the Hamiltonian cycle $|\nu\rangle$, that corresponds to $\tilde{\varphi}_{\text{min}}$. Note that if $C\tilde{\varphi}_{\text{min}} = O(1)$, the probability of obtaining 1 as the outcome of measuring the ancilla qubit above can be as small as $O(1/(N-1)!)$, 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 $\chi$ in (7):

$$\zeta = \sum_{\tilde{\varphi}_{\text{min}}} \frac{\alpha^2 N^2 \tilde{\varphi}^2_{\text{max}}}{\lambda_{\text{min}}} |\tilde{\varphi}_{\nu}\rangle \langle \tilde{\varphi}_{\nu}| \otimes |\nu\rangle \langle \nu|,$$

where if only $\tilde{\varphi}_{\text{min}}$ 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 $\lambda_{\text{min}} := \sum_{\tilde{\varphi}_{\text{min}}} \alpha^2 N^2 \tilde{\varphi}^2_{\text{max}}$. We would not use amplitude amplification, 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 $\zeta$ in (8), we first exponentiate the ancilla qubit (call it $\rho$) from (7) to get the unitary $Y = e^{i\sigma}$. We create $Y$ by repeatedly applying the below circuit to augment terms like $|0123\rangle$ in $\rho$ to $|0112330\rangle$ in $\sigma$ for $N = 4$.

![Quantum circuit to augment terms like $|0123\rangle$ in $\rho$ to $|0112330\rangle$ in $\sigma$ for $N = 4$.](image)

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

$$\chi = \sum_{\nu} \alpha^2 N^2 \tilde{\varphi}^2_{\nu} |\tilde{\varphi}_{\nu}\rangle \langle \tilde{\varphi}_{\nu}| \otimes |\nu\rangle \langle \nu|,$$

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

Note that the controlled rotation in (12) is achieved by considering two ancilla qubits, denoted by $A$ and $B$, initialized in the state $|0_A0_B\rangle$, and then rotating this state, conditioned on phase estimates $|\hat{\phi}_\nu\rangle $, and ignoring one of the ancilla qubits:

$$
\xi = \text{Tr}_B \left\{ \sum_\nu \alpha_\nu^2 |\hat{\phi}_\nu\rangle \langle \hat{\phi}_\nu| \otimes |\nu\rangle \otimes |\Phi_{A\nu}\rangle \langle \Phi_{A\nu}| \right\},
$$

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

$$
\hat{\vartheta} = \sum_\nu \alpha_\nu^2 |\hat{\phi}_\nu\rangle \langle \hat{\phi}_\nu| \otimes |\nu\rangle \otimes |\psi\rangle \langle \psi|, 
$$

where $|\nu\rangle := \sqrt{1-C^2|\nu\rangle \langle \nu| + C^2|\nu\rangle \langle \nu|}$. With $n$ ancilla qubits, the state corresponding to $\chi$ of (10) is:

$$
\Theta = \sum_\nu \alpha_\nu^2 |\hat{\phi}_\nu\rangle \langle \hat{\phi}_\nu| \otimes |\nu\rangle \otimes |\Psi\rangle \langle \Psi|,
$$

where $|\Psi\rangle := \sqrt{1-C^2|\nu\rangle \langle \nu| + C^2|\nu\rangle \langle \nu|}$.

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: $(\hat{\varphi}_{\min}/N + \hat{\varphi}_{\max}/N)/2 = \omega M/(2N)$, with $\omega = \max(2N, (e!) \theta)$, where $\theta = N^2 C_2$ for undirected graphs, $\theta/2 = N^2 C_2$ for directed graphs, $e$ is number of too large (at least ten times the smallest) or missing edges. So, a guess of $1/\kappa = 1/\kappa = \omega M/(\hat{\varphi}_{\max} - 1)$. We use $\eta = \Omega/\omega$, if $\omega > 2N$, $\eta = (\Omega/\omega) \times \omega/(2N)$, if $\omega > 4N$, else $\eta = 1$, if $\omega = 2N$, where $\Omega$ is the sum of all edges including missing edges, $s \leq s$ is the sum of all edges excluding missing or too large edges. $1/\kappa = 1/(\eta \kappa)$ would be below, yet close to, normalised $\varphi_{\min}$, so that we repeat our algorithm a few times, slightly raising $1/\kappa$ each time till we capture $\varphi_{\min}$ only. We find that this empirical method rarely fails [23].

### III. Algorithm

1) We create the unitary $U = e^{i\kappa t}$ encoding all weights of edges $\phi_{jk}$, where $\Phi = \sum_{j,k=0}^{L-1} \phi_{max} |jk\rangle \langle jk|$, $\phi_{max}$ is sum of $N$ largest edges upon replacing missing edges $\forall j,k < N$ by $N$, $s$ is sum of all given edges, $\log L = \log N$. We create $N$ copies of $U$ for $\nu = U^{\otimes N}$.

2) We initialize $N$ number of data registers of $[\log N]$ 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 $N^2 C_2$ number of single-qubit ancilla registers, each initialized to $|+\rangle = 1/{\sqrt{2}} (|0\rangle + |1\rangle)$, and applying $[\log N]$ 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 $\rho = 1/{2^N} \sum_{\mu} \beta_{\mu}^2 |\mu\rangle \langle \mu|$, where $\sum_{\mu} \beta_{\mu}^2 = 2^N C_2$. We augment this $N$-register state by adding $N$ ancilla registers each of $[\log(N)]$ qubits in state $|0\rangle$ and acting $CNOT$ gates to get a $2N$-register state $\sigma = 1/{2^N} \sum_{\mu} \beta_{\mu}^2 |\mu\rangle \langle \mu|$, where $\sum_{\mu} \beta_{\mu}^2 = 2^N C_2$.

4) We perform quantum phase estimation [19] on $V$ using $\sigma$ as input to get $\gamma = 2^{N/2} \sum_{\nu} \beta_{\nu}^2 |\nu\rangle \langle \nu|$, see step 5.

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

6) We exponentiate the effective state $\phi$ of each ancilla qubit, using $n$ copies of $\gamma$ from ancilla qubits, to get a unitary $Y := e^{i\eta \kappa}$, and perform phase estimation on $Y$ for the eigenstate $|1\rangle$. If the phase estimate for $|1\rangle$ is 0, we repeat step 3 with a slightly smaller value of $C$, and then repeat this step, until we can capture $\varphi_{\min}$ in step 5 and get $\varphi_{\min}$ as the phase estimate in this step.

7) We further apply a rotation to the ancilla register in the state $\vartheta$ from step 5 using $\varphi_{\min}$ obtained in step 6 to get $\Theta = \sum_{\nu} \alpha_\nu^2 |\hat{\phi}_\nu\rangle \langle \hat{\phi}_\nu| \otimes |\nu\rangle \otimes |\Psi\rangle \langle \Psi|$, where $|\Psi\rangle := \sqrt{1-C^2|\nu\rangle \langle \nu| + C^2|\nu\rangle \langle \nu|}$. So, upon tracing out the ancilla register $|\Psi\rangle$ from $\Theta$, we get $\zeta := \sum_{\nu} \alpha_\nu^2 C^2 |\nu\rangle \langle \nu|$, as $\sum_{\nu} \alpha_\nu^2 C^2 |\nu\rangle \langle \nu| = 1$.

8) Measure the two registers in the state $\zeta$ from step 7 to get $\varphi_{\min}$ and a $\nu$. If $\varphi_{\min}$ is larger than $s/\varphi_{max}$, output the decision that there is no Hamiltonian cycle in the graph. Otherwise, output the obtained values of $\nu$ and $\varphi_{\min} \times \varphi_{\max}$ 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 $|A_0\rangle \langle u_j|$, where $|u_j\rangle$ is the $j$-th eigenstate of the Hermitian matrix $\Gamma$, that we exponentiate, and $|A_0\rangle := \sqrt{\frac{1}{T} \sum_{t=0}^{T-1} \sin \frac{\pi (t+1/2)}{T} |u_j\rangle}$ for some large $T$. The state $|A_0\rangle$ can be prepared up to some error $\epsilon_k$ in time $\text{poly} \log(T/\epsilon_k)$ (see Ref. [19]). We apply the conditional Hamiltonian evolution $\sum_{t=0}^{T-1} |t\rangle \langle t| \otimes e^{t^2 \Gamma T}$ on the initial state in both registers, and then apply inverse quantum Fourier transform on the first register to get the state $\sum_{t=1}^{T-1} v(t\langle q | |u_j\rangle$. Defining the estimate $\tilde{r}_q$ of the $q$-th eigenvalue of $\Gamma$ as $\tilde{r}_q := \frac{2\pi q}{T}$, we relabel the Fourier basis states $|q\rangle$ to get $\sum_{t=1}^{T-1} v(t\langle q | |u_j\rangle$. If the phase estimation is perfect, we have $v(t\langle q | = 1$ if $r_q = r_j$, and 0 otherwise. So, we get the state $|\tilde{r}_j\rangle \langle u_j|$, 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 $[\log N]$ 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 4 creating each copy of $U$ has a complexity of $O(2 \log(N))$ [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 $C = \eta e^{i\varphi}$ in step 5 require the time variable $\ell$ in step 2 to be $O(\eta \ell e^{i\varphi})$ (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 $O(\log(2n)) = O(\tau^{\ell}/e) = O(1/e^3)$ [21], [22], where $\epsilon$ is the simulation error for $Y$, as also the precision error of improved phase estimation of $Y$ for which $\tau = O(1/e)$. Here, $\epsilon$ or $\ell/2$ is an error in trace distance, and so, determines the maximum error probability in simulation or estimation of $Y$ [23]. The eigenvalue of $\eta$ 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 $\tau \geq O(1/(N-1))$ in usual phase estimation [1]. Since $\ell/2$ is maximum error probability, we use $\epsilon = O(1/poly(N))$, not $\epsilon = O(1/(N-1))$, but with $T = O((N-1)!)$ \leq $O(\eta \ell e^{i\varphi})$ in improved phase estimation, so that $\tau = O(poly(N))$, 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 $O(2 \log(N)\ell t)$, which yields $O(\log(N)\eta \ell e^{i\varphi}) \leq O(\eta N^{3} \log(N)\kappa /\ell e^{i\varphi})$, taking $\epsilon = \ell e^{i\varphi}$ for simplicity, with $\epsilon = O(1/poly(N))$, and $\eta \leq O(3^{\ell})$ (since $\ell \pi /2 \leq O(N^{2} s / s) = O(N^{2})$ and $\kappa /\ell e^{i\varphi} = O(N^{2}/N) = O(N)$. This is because the complexity of step 1 is dominant amongst all steps, if it suffices to have $\ell \geq O(1/N)$. Also, $\kappa = O(poly(N))$, 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 too-far Hamiltonian cycles visits all cities, including too-far ones. When there are $\ell \leq O(N^{2})$ missing edges, we have $\kappa = \varphi_{\max} /((e+1)s - \varphi_{\max})$, where $\varphi_{\max} < ps + O(10(N - p)/\ell) s$ 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 $\kappa < O(\eta N^{2} p + 10(N - p)(N^{2}(e+1) - p) - (N(p - 10(N - p))) / O(poly(N))$. Note that we need to classically precompute $\varphi_{\max}$ by summing the $N$ largest edges, for which we sort the data, that is expensive. If we use merge sort to sort the $NC_2$ number of edges, the complexity is $O(N^{3} \log(N) C_2) = O(N^{3} \log(N))$, which is less than $O(N^{3} \log(N) \kappa /\ell e^{i\varphi})$. But if $\ell \leq O(1/N^{2})$, the complexity of step 6 is dominant, so the overall complexity of our algorithm is at least $O(1/e^3) = O(N^{-3})$. Clearly, this is an exponential speedup over the brute-force complexity of $O(N^N)$ for undirected [1], or $O(N^{2N})$ for directed graphs.

REFERENCES