Efficient Quantum Algorithm for SUBSET-SUM Problem

Sanchita Ghosh,[1,*] Anant Sharma,[1,†] Sreetama Das,[2,3,4,‡] and Shibdas Roy[1,5,4,§]

[1] *Center for Quantum Engineering, Research and Education (CQuERE), TCG CREST, Salt Lake, Sector 5, Kolkata 700091, India.*

[2] *Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO), Largo Enrico Fermi 6, 50125 Florence, Italy.*

[3] *European Laboratory for Non-Linear Spectroscopy (LENS), University of Florence, via Nello Carrara 1, 50019 Sesto Fiorentino, Italy.*

[4] *Department of Physics and Astronomy, University of Florence, via Sansone 1, 50019 Sesto Fiorentino, Italy.*

[5] *Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India.*

Problems in the complexity class NP are not all known to be solvable, but are verifiable given the solution, in polynomial time by a classical computer. The complexity class BQP includes all problems solvable in polynomial time by a quantum computer. Prime factorization is in NP class, and is also in BQP class, owing to Shor's algorithm. The hardest of all problems within the NP class are called NP-complete. If a quantum algorithm can solve an NP-complete problem in polynomial time, it would imply that a quantum computer can solve all problems in NP in polynomial time. Here, we present a polynomial-time quantum algorithm to solve an NP-complete variant of the $SUBSET - SUM$ problem, thereby, rendering $NP \subseteq BQP$. We illustrate that given a set of integers, which may be positive or negative, a quantum computer can decide in polynomial time whether there exists any subset that sums to zero. There are many real-world applications of our result, such as finding patterns efficiently in stock-market data, or in recordings of the weather or brain activity. As an example, the decision problem of matching two images in image processing is NP-complete, and can be solved in polynomial time, when amplitude amplification is not required.

I. INTRODUCTION

The complexity class P (Polynomial time) includes all computational problems that are known to be solvable in polynomial time by a classical computer [1]. Those that are not all known to be solvable, but verifiable given the solution, in polynomial time by a classical computer, constitute the complexity class NP (Non-deterministic Polynomial time) [1]. The complexity class BQP (Bounded-error Quantum Polynomial time) includes all computational problems that are known to be solvable in polynomial time by a quantum computer, where a bounded probability of error is allowed [2].

A problem that is in NP as well as BQP is the prime factorization problem, i.e. finding the prime factors of a given positive integer. This means that the prime factorization problem is not known to be solvable in polynomial time by a classical computer, but it is known to be solvable in polynomial time using Shor's algorithm [3] by a quantum computer.

The hardest of all problems in NP are called NP-complete. Specifically, an NP-complete problem is one, which any problem in the NP class can be reduced to in polynomial time. The prime factorization problem is in NP, but not known to be NP-complete. The $SUBSET - SUM$ problem is to decide, given a set S of integers, whether a subset of the integers sums to a target sum X [1]. A variant of this problem is to decide, given a set S of integers, which may be positive or negative, whether a subset sums to $X = 0$. This variant of the $SUBSET - SUM$ problem is known to be NP-complete [4], and is what we shall consider here.

There is no known BQP-algorithm for an NP-complete problem. If an NP-complete problem is shown to be solvable in polynomial time by a quantum computer, it would essentially mean that all problems in NP are solvable in polynomial time by a quantum computer. In other words, a BQP-algorithm for an NP-complete problem would imply $NP \subseteq BQP$, i.e. the complexity class NP lies in the complexity class BQP.

In this work, we present a polynomial-time quantum algorithm for the aforementioned NP-complete decision version of the $SUBSET-SUM$ problem. Existing quantum algorithms for the $SUBSET - SUM$ problem attain improvements in time complexity, that is still exponential in the size of the problem [5, 6]. There are polynomial-time quantum algorithms but with restrictive assumptions [7], pseudo-polynomial time classical algorithms using dynamic programming [8] or polynomial-time classical approximation algorithms [9] for the problem. "Whether $P = NP$" is literally a million-dollar problem [10]. We prove here that $NP \subseteq BQP$, by presenting a BQP-algorithm for the NP-complete variant of the $SUBSET - SUM$ problem, without any restriction or approximation or assumption.

There are many real-world applications of our result, such as finding efficiently whatever patterns exist in stock-market data, or in recordings of the weather or brain activity [11]. Some concrete examples of real-world NP-complete problems are listed in Ref. [10], such as finding a DNA sequence that best fits a collection of fragments of the sequence, finding a ground state in the

* sanchita.ghosh14@gmail.com
† anantsharma2410@gmail.com
‡ sreetama.das@ino.cnr.it
§ roy.shibdas@gmail.com

Ising model of phase transitions, finding optimal protein threading procedures, finding Nash equilibriums with specific properties in a number of environments, and determining if a mathematical statement has a short proof. All these problems can be solved by a quantum computer in polynomial time, owing to our result here.

II. METHOD

The decision problem has a given set S of integers, which may be positive or negative. The problem is to find whether there is a subset of these integers that sums to zero. For example, given a set $S = \{5, 9, -3, 450, -295, -2\}$, the answer is *yes*, since the subset $\{5, -3, -2\}$ sums to zero.

Consider that our given set S has N integers. Then, we would use $\log M := \lceil \log N \rceil$ number of qubits to index the N integers of the set S. For example, if the set S has 5 integers, then we start with $\log M = 3$ qubits, and use 5 levels: $|0\rangle = |000\rangle$, $|1\rangle = |001\rangle$, $|2\rangle = |010\rangle$, $|3\rangle = |011\rangle$, and $|4\rangle = |100\rangle$ to index the 5 integers of the set. We initialize N number of data registers, each of $\lceil \log N \rceil$ qubits, to $|0\rangle$, $|1\rangle$, $|2\rangle$, $\dots$, $|N-1\rangle$, respectively. Then, we generate all possible permutations of $0, 1, 2, \dots, N-1$ by using ${}^N C_2 = N(N-1)/2$ number of single-qubit ancilla registers, each initialized to the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and applying $\lceil \log N \rceil$ number of controlled swap gates on each combination of two data registers, with one ancilla register as control qubit. Please see circuit in Figure 1.

We then have the below state in the data registers,

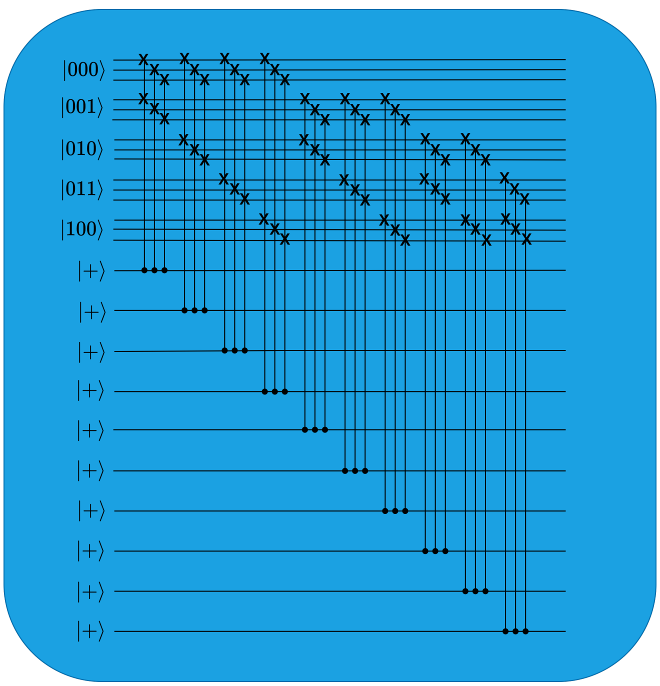

FIG. 1. Quantum circuit using controlled swaps to create a superposition of all permutations of $0, 1, \dots, N-1$ for $N = 5$.

upon tracing out the ancilla qubits, with $\sum_k \beta_k = 2^{{}^N C_2}$:

$$\xi_N = \frac{1}{2^{{}^N C_2}} \sum_k \beta_k |\zeta_k\rangle\langle\zeta_k|. \tag{1}$$

Box 1. Improved Quantum Phase Estimation (IQPE).

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 initial state $|\Lambda_0\rangle$ can be prepared upto some error ϵ_Λ in time $\mathrm{poly} \log_2(T/\epsilon_\Lambda)$ (see Section A of Supplementary material of Ref. [12]). We apply the conditional Hamiltonian evolution $\sum_{\iota=0}^{T-1} |\iota\rangle\langle\iota| \otimes e^{i\Gamma\iota t_0/T}$ on the initial state in both registers, and then apply quantum Fourier transform (QFT) on the first register to obtain the state $\sum_{p=0}^{T-1} \nu_{p|j} |p\rangle|u_j\rangle$. Defining the estimate $\tilde{r}_p$ of the p-th eigenvalue r_p of Γ as $\tilde{r}_p := \frac{2\pi p}{t_0}$, we can relabel the Fourier basis states $|p\rangle$ to obtain $\sum_{p=0}^{T-1} \nu_{p|j} |\tilde{r}_p\rangle|u_j\rangle$. If the phase estimation is perfect, we have $\nu_{p|j} = 1$ if $\tilde{r}_p = r_j$, and 0 otherwise. So, we obtain the state $|\tilde{r}_j\rangle|u_j\rangle$, from which we get the estimate of r_j upon measuring the first register. This quantum phase estimation method errs by $\varepsilon = O(1/t_0)$ in estimating r_j [12], where $\varepsilon/2$ is the error in trace distance (see just before Section A and just before Theorem 6 in the Supplementary material of Ref. [12]).

In general, for $n = N-1, N-2, \dots, 1$, we can simply trace out from the state ξ_N, the last $N-n$ data registers of $\lceil \log N \rceil$ qubits each, to get:

$$\xi_n = \frac{1}{2^{{}^N C_2}} \sum_j \gamma_j |\zeta_j\rangle\langle\zeta_j|, \tag{2}$$

where now $\sum_j \gamma_j = 2^{{}^N C_2}$. We create N copies of unitary:

$$U = \begin{bmatrix} e^{2\pi i\phi_0} & 0 & \dots & 0 \\ 0 & e^{2\pi i\phi_1} & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & e^{2\pi i\phi_{M-1}} \end{bmatrix}, \tag{3}$$

where the phases $\phi_0, \dots, \phi_{N-1}$ are the N integers from the set S, divided by 2π, and $\phi_N, \phi_{N+1}, \dots, \phi_{M-1} = 0$, if $N < M$.

Phase estimation algorithm is known to obtain the phase of a given eigenstate of a unitary efficiently in polynomial time, depending on the desired accuracy of the phase estimate [2]. We will instead use what is called an improved quantum phase estimation (IQPE) method from Ref. [12], which is outlined in Box 1. We can then perform phase estimation on the unitary $U^{\otimes n}$ for the eigenstates ξ_n, since all the sums (denoted by φ_j's) of the possible combinations of the phase factors $\phi_0,\ldots,\phi_{N-1}$ are captured with $U^{\otimes n}$, $n=1,2,\ldots,N$. We do this, starting from the state $|\Lambda_0\rangle$ in a register and the state ξ_n in another register, to obtain the state:

$$\rho_n = \frac{1}{2^{{}^N C_2}} \sum_j \gamma_j |\tilde{\varphi}_j\rangle\langle\tilde{\varphi}_j| \otimes |\zeta_j\rangle\langle\zeta_j|, \qquad (4)$$

where $\tilde{\varphi}_j$ is the estimate of φ_j. We next exponentiate the state in the first register to get a unitary and estimate the phase of the eigenstate $|0\rangle$ of this unitary to know if there is any sum of zero. If this phase estimate is non-zero for any n, we output "Yes"; else, we output "No".

III. ALGORITHM

Our algorithm is as follows:

1. Given the set S of N integers, $s_0,s_1,\ldots,s_{N-1}$, create N copies of a diagonal unitary (3), where $\phi_q = s_q/(2\pi)$, $\forall\, q = 0,1,\ldots,N-1$, and $\phi_q = 0$, $\forall\, q = N,N+1,\ldots,M-1$ if $N<M$.

2. Initialize N number of data registers, each of $\lceil \log N\rceil$ qubits, to $|0\rangle$, $|1\rangle$, $\ldots$, $|N-1\rangle$, respectively, and generate all possible permutations of $0,1,\ldots,N-1$ by using ${}^N C_2 = N(N-1)/2$ number of ancilla qubits, each in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, and applying $\lceil \log N\rceil$ number of controlled swap gates on each combination of two data registers, with one ancilla as control qubit, as shown in Figure 1 for $N=5$. The resulting state in the data registers, denoted collectively as register B, upon tracing out the ancilla qubits, is (1). Start the next step with $n=N$.

3. Trace out from the state ξ_N, the last $N-n$ data registers of $\lceil \log N\rceil$ qubits each, to get the state (2). Initialize a register A to the state $|\Lambda_0\rangle$ of ℓ qubits, and identify ξ_n as register B now. Perform (improved) quantum phase estimation of the unitary $U^{\otimes n}$, with registers A and B as input, to obtain the state (4).

4. The effective state in the first register A only is:

$$\sigma = \frac{1}{2^{{}^N C_2}} \sum_{m=0}^{2^\ell-1} \alpha_m |\tilde{\varphi}_m\rangle\langle\tilde{\varphi}_m|, \qquad (5)$$

where $\sum_m \alpha_m = 2^{{}^N C_2}$. Notice that σ is a $2^\ell \times 2^\ell$ diagonal matrix in its eigenbasis $\{|\tilde{\varphi}_m\rangle\}$. It is required to know if α_0 is zero here. Thus, exponentiate the density matrix σ, by repeated application of the following to the unknown state σ in register A and a known state ς in a register C [13, 14]:

$$\mathrm{Tr}_A\left[e^{-i\mathcal{S}\Delta t}(\sigma\otimes\varsigma)e^{i\mathcal{S}\Delta t}\right] = \varsigma - i\Delta t[\sigma,\varsigma] + O(\Delta t^2), \qquad (6)$$

to obtain the unitary $e^{-i\sigma t}$, where $\mathcal{S}$ is the swap operator, which is sparse and so, $e^{-i\mathcal{S}\Delta t}$ can be performed efficiently [12, 15]. Then, perform phase estimation of the eigenstate $|\tilde{\varphi}_0\rangle = |0\rangle$ of the unitary. The phase estimation process requires controlled-$e^{-i\sigma t}$ operations, that can be performed by simply using conditional swap instead of swap operation above for varying times t (see Ref. [13] for details). If this phase estimate $\tilde{\alpha}_0$ is zero, proceed to the next step, else output the decision "Yes" as the solution, and terminate the algorithm.

5. Undo the steps 4 and 3 to revert back register B to ξ_n. Go back to step 3 for $n := n-1$, if $n>1$.

6. Output the decision "No" as the solution.

IV. ALGORITHM COMPLEXITY

Below, we analyse the complexity of our algorithm to demonstrate that it can be run on a quantum computer in polynomial time rather than exponential time:

1. Since the unitary U, and so, the Hamiltonian $\mathcal{A}$, is a sparse matrix, $U = e^{i\mathcal{A}\tau}$ can be implemented efficiently in $O(\log(M)s^2\tau) = O(\log(M)\tau)$ steps [12, 15], where

$$\mathcal{A} = \begin{bmatrix} 2\pi\phi_0 & 0 & \ldots & 0 \\ 0 & 2\pi\phi_1 & \ldots & 0 \\ & & \ddots & \\ 0 & 0 & \ldots & 2\pi\phi_{M-1} \end{bmatrix} \qquad (7)$$

is an $(s=1)$-sparse matrix. The N copies of the unitary U can be created in parallel.

2. The controlled swap operations on $\lceil \log N\rceil$ qubits of each data register can be performed parallelly, and there are ${}^N C_2 = N(N-1)/2 = O(N^2)$ such sets of controlled swaps, yielding a complexity of $O(N^2)$ for this step.

3. Since we use the improved phase estimation method from Ref. [12], and we perform this for upto the N number of unitaries $U^{\otimes n}$, $n=1,2,\ldots,N$, we have $\tau = O(1/\delta)$ in step 1 times N, where δ is the estimation precision error. Otherwise, the complexity of the phase estimation in this step is dominated by the quantum Fourier transform (QFT), that takes $O(\ell^2)$ steps. Since there are upto N iterations, the complexity of this step is $O(N\ell^2)$.

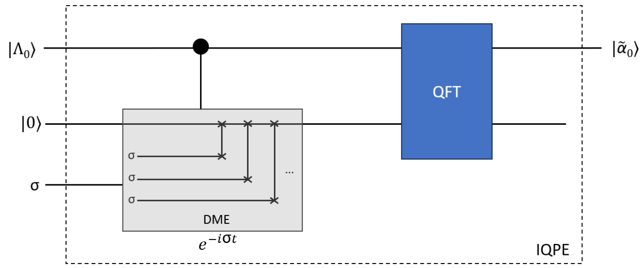

FIG. 2. Density matrix exponentiation (DME) [13, 14] and improved quantum phase estimation (IQPE) [12].

4. The density matrix exponentiation can be done with $z = O(t^2\epsilon^{-1})$ copies of the density matrix, where $t = z\Delta t$ and ϵ is error determining the desired accuracy [13]. The circuit depth required for the density matrix exponentiation is $O(\log(2^\ell)z) = O(\ell t^2/\epsilon)$ [14]. As we use the improved phase estimation method from Ref. [12] with error μ, the complexity of this step is $O(\ell/(\epsilon\mu^2))$, since $t = O(1/\mu)$ [13]. Since we perform this step for up to N states, the overall complexity of this step is $O(N\ell/(\epsilon\mu^2))$. See Figure 2.

5. Undoing the previous 2 steps is trivial, since we need to just apply the conjugate transpose of the corresponding unitary operation in each case. We, therefore, ignore the complexity of this step.

6. This step does not contribute to overall complexity.

Thus, the dominant overall complexity of our algorithm is $O(\log(M)\tau\ell Nz) = O(\log(M)N\ell/(\delta\epsilon\mu^2))$, which is obtained by multiplying the complexities of steps 1 and 4, since steps 1-3 are required to be repeated to generate every copy of the density matrix required in step 4. Here, we must check how ℓ, δ, μ and ϵ scale with N.

Note that the use of the improved quantum phase estimation method from Box 1, as opposed to the conventional method from Ref. [2], makes the time variable τ in step 3 above not directly dependent on ℓ (although the quantity T is equal to 2^ℓ), as long as ℓ is at least as many qubits as required for the estimation precision error δ.

Now, it may appear that the required estimation error in step 4 is $\mu = O(1/t) = O((1/2^{{}^NC_2}) \times (2^{{}^NC_2}/2^\ell)) = O(1/2^\ell)$ for the algorithm to distinguish 0 and $1/2^\ell$ correctly (noting that we take $\ell \geq {}^NC_2$ later, which ensures $\alpha_0/2^{{}^NC_2}$, if non-zero, is always larger than $1/2^\ell$). This means that our algorithm will always output the correct decision, if we have $t = O(1/\mu) = O(2^\ell)$, which is exponential in ℓ. However, our algorithm need not output the correct decision all the time, but at least $2/3^{\text{rd}}$ of the time, to be BQP.

Now, as mentioned before, the estimation error in Ref. [12], and so μ here, is twice the error in trace distance. Then, $\mu/2$ gives the maximum probability of estimation error, since the trace distance between two states gives the maximum difference in probability of any measurement on the two states (see just above Eq. (3) in Ref. [16]). Also, if the error ϵ in simulating $e^{-i\sigma t}$ is an error in trace distance (as defined in Eq. (3) in Ref. [16]), it needs to be less than or equal to $1/6$ for the simulation to be successful with a probability of at least $2/3$ (see proof of Theorem 2 in Ref. [16]). The estimation error δ from step 3 is also twice the error in trace distance, so that $\delta/2$ determines the maximum probability of estimation error in step 3. Moreover, let υ be the error in simulating each copy of the unitary U in step 1. Since this error is in trace distance [15], it also determines the maximum probability of simulation error for each U in step 1. As the errors would accumulate, we must ensure:

$$N\upsilon + \delta/2 + 2\epsilon + \mu/2 \leq 1/3, \qquad (8)$$

where the factor N arises because there are N copies of the unitary U created. Taking $\upsilon = \delta = \mu = \epsilon$ for simplicity, the combined success probability would then be $1 - (N+3)\epsilon \geq 2/3$. Further, since we have upto N iterations, we need to have $(1-(N+3)\epsilon)^N \geq 2/3$ to ensure that our algorithm is BQP. Since for large N, the quantity $(N+3)\epsilon$ needs to be small for this to hold, we can effectively write that we must have $1 - N(N+3)\epsilon \geq 2/3$, which yields $\epsilon \leq 1/(N(N+3))$. In fact, this is also why the various errors have been *added* in (8). Since steps 1-3 are repeated z times, to obtain z copies of the state required for step 4, we should ideally have in (8):

$$z(N\upsilon + \delta/2) + 2\epsilon + \mu/2 \leq 1/3. \qquad (9)$$

However, using a value of $z = O(t^2/\epsilon) = O(1/\epsilon^3)$ raises the upper bound to ϵ, required to satisfy the above inequality, and so, we use the worst-case $z = 1$ to obtain the upper bound to ϵ to consider. Also, note that we must have $t/(6\epsilon) \geq \pi$ in step 4 (see Theorem 2 in Ref. [16]), which implies $\epsilon \leq t/(6\pi) = O(1/(6\pi\mu))$, that, in turn, yields $\epsilon \leq O(1/\sqrt{6\pi})$ with $\mu = \epsilon$. Clearly, an $\epsilon \leq 1/(N(N+3))$ satisfies this requirement. Thus, the phase estimation error probability in step 4 needs to be $\mu/2 = \epsilon/2 \leq 1/(2N(N+3))$, regardless of ℓ, and we need not have $\mu = O(1/2^\ell)$, for our algorithm to be BQP.

Besides, the density matrix exponentiation method used in step 4 is efficient, providing exponential speedup to our algorithm when the matrix being exponentiated is of low rank [13]. We use the same number of qubits ℓ in register A for all values of n upto N. A suitable choice of ℓ is $\ell \geq {}^NC_2$. Although after step 3, the number of non-zero entries in register B is ${}^NP_n = N!/(N-n)!$, that in register A is ${}^NC_n = N!/(n!(N-n)!)$, which is always much less than $2^{{}^NC_2}$. Clearly, for all values of $n = 1, 2, \ldots, N$, the density matrix σ in step 4 is then of low rank. Taking $\ell \geq {}^NC_2$ also ensures $\alpha_0/2^{{}^NC_2}$ (that if non-zero, is at least $n!(N-n)!/2^{{}^NC_2}$) is always larger than $1/2^\ell$, as mentioned earlier. This is because each γ_j in (4) is at least $(N-n)!$, and there are ${}^nP_n = n!$ number of ζ_j's, that, representing the same subset of the set S, have the same value of φ_j.

Thus, if we take $M = N$ (for when $\lceil \log N \rceil = \log N$), $\ell = {}^N C_2 = O(N^2)$ and $\epsilon = \delta = \mu = O(1/N^2)$, our algorithm has a complexity of $O(\log(M) N \ell/(\delta\epsilon\mu^2)) = O(N^{11}\log(N))$, which is indeed polynomial, and not exponential, in N.

V. DISCUSSION

While classical algorithms for NP-complete problems take at least $O(2^N)$ steps in the worst case, the common quantum algorithms achieve quadratic speedup over classical algorithms [17], using a technique called amplitude amplification [18, 19], that is based on Grover's search algorithm [20]. By contrast, we achieve exponential quantum speedup, using the density matrix exponentiation method in step 4. Our overall algorithm takes only $O(N^{11}\log(N))$ steps. For example, in quantum imaging, the decision problem of matching two images is known to be NP-complete [21]. It is known to have a quadratic quantum speedup, again using amplitude amplification based on Grover's search [22, 23]. However, this problem can be solved by a quantum computer in polynomial time using density matrix exponentiation, without needing amplitude amplification.

Note that ${}^N C_n \leq {}^N C_{N/2} \, \forall n \in [1, N]$ and ${}^N C_{N/2} \geq 2^{N/2}$. This reveals that if we do not use density matrix exponentiation in step 4, and rather use conventional quantum state tomography to estimate the density matrix σ, we would obtain a best of quadratic quantum speedup, scaling as $2^{N/2}$, as is obtained with Grover's search for NP-complete problems. However, our use of density matrix exponentiation technique in step 4 to know α_0 for eigenstate $|0\rangle$ of the low rank density matrix σ allows for achieving an exponential quantum speedup to solve the NP-complete decision problem.

Note that in step 1, there are infinitely many possible integer sums that are mapped to the fixed phase interval $(-2\pi, 2\pi)$ through $U^{\otimes n}$. Thus, there can be many sums that are too close to zero as phases to be distinguishable in step 3. So, α_0 in step 4 can be incorrectly non-zero, when it was supposed to be zero, affecting the decision of the algorithm. However, α_0' needs to be larger than $O(2^\ell/N^2) = O(2^{{}^N C_2}/N^2)$ in step 4 for μ to exceed $O(1/N^2)$, where $\alpha_0'/2^\ell = \alpha_0/2^{{}^N C_2}$. This is because we have $\alpha_0'/2^\ell \leq O(1/N^2)$ for $\mu \leq O(1/N^2)$. If the maximum estimation error probability in step 3 is $\delta/2$, where $\delta = O(1/\tau) \leq O(1/N^2)$, we have $\alpha_0' \leq O({}^N C_{N/2}/N^2) < O(2^{{}^N C_2}/N^2)$, since the total number of sums for a given n is maximum for $n = N/2$. So, our algorithm cannot output a wrong decision with a probability exceeding $1/3$.

VI. CONCLUSION

To summarise, we presented here the first BQP-algorithm for an NP-complete variant of the $SUBSET-SUM$ problem, thereby, proving $NP \subseteq BQP$. There are existing BQP-algorithms, such as Shor's algorithm, for problems, such as prime factorization, that are known to be in NP but not NP-complete. For NP-complete problems, the usual approaches achieve a quadratic quantum speedup over classical algorithms, using amplitude amplification, based on Grover's search. In our algorithm, we achieve an exponential quantum speedup by using density matrix exponentiation, without requiring amplitude amplification. Our work ensures that many real-world computationally difficult problems can be solved efficiently in polynomial time by quantum computers, including but not limited to finding patterns in stock-market data, matching two images in image processing, or finding optimal protein threading procedures.

ACKNOWLEDGMENTS

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

[1] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. (Cengage Learning, 2013).

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

[3] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.

[4] J. Kleinberg and É. Tardos, *Algorithm Design*, 2nd ed. (Pearson/Addison-Wesley, 2006).

[5] D. J. Bernstein, S. Jeffery, T. Lange, and A. Meurer, Quantum algorithms for the Subset-Sum problem, in *Post-Quantum Cryptography – PQCrypto 2013*, Lecture Notes in Computer Science, Vol. 7932, edited by P. Gaborit (Springer Berlin Heidelberg, 2013) pp. 16–33.

[6] X. Bonnetain, R. Bricout, A. Schrottenloher, and Y. Shen, Improved classical and quantum algorithms for Subset-Sum, in *Advances in Cryptology – ASIACRYPT 2020*, Lecture Notes in Computer Science, Vol. 12492, edited by S. Moriai and H. Wang (Springer Cham, 2020) pp. 633–666.

[7] A. Daskin, A quantum approach to Subset-Sum and similar problems (2017), arXiv:1707.08730.

[8] K. Bringmann, A near-linear pseudopolynomial time algorithm for Subset Sum, in *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2017) pp. 1073–1084.

[9] H. Kellerer, R. Mansini, U. Pferschy, and M. G. Speranza, An efficient fully polynomial approximation scheme for the Subset-Sum problem, Journal of Computer and System Sciences **66**, 349 (2003).

[10] L. Fortnow, The status of the P versus NP problem, Communications of the ACM **52**, 78 (2009).

[11] S. Aaronson, The limits of quantum computers (2008), Scientific American, March 2008, pp. 62-69.

[12] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Physical Review Letters **103**, 150502 (2009).

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

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

[15] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Efficient quantum algorithms for simulating sparse hamiltonians, Communications in Mathematical Physics **270**, 359 (2007).

[16] 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**, 13 (2017).

[17] M. Fürer, Solving NP-complete problems with quantum search, in *LATIN 2008: Theoretical Informatics*, Vol. 106, edited by E. S. Laber, C. Bornstein, L. T. Nogueira, and L. Faria (Springer Berlin Heidelberg, 2008) pp. 784–792.

[18] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, in *Contemporary Mathematics*, Vol. 305 (American Mathematical Society, 2002) Chap. Quantum Computation and Information, pp. 53–74.

[19] H. Kwon and J. Bae, Quantum amplitude-amplification operators, Physical Review A **104**, 062438 (2021).

[20] L. K. Grover, A fast quantum mechanical algorithm for database search, in *STOC'96: Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing* (Association for Computing Machinery, 1996) pp. 212–219.

[21] D. Keysers and W. Unger, Elastic image matching is NP-complete, Pattern Recognition Letters **24**, 445 (2003).

[22] N. Jiang, Y. Dang, and J. Wang, Quantum image matching, Quantum Information Processing **15**, 3543 (2016).

[23] H. Tezuka, K. Nakaji, T. Satoh, and N. Yamamoto, Grover search revisited: Application to image pattern matching, Physical Review A **105**, 032440 (2022).