THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCA

Article

Pauli String Partitioning Algorithm with the Ising Model for Simultaneous Measurements

Published as part of The Journal of Physical Chemistry virtual special issue "Physical Chemistry of Quantum Information Science".

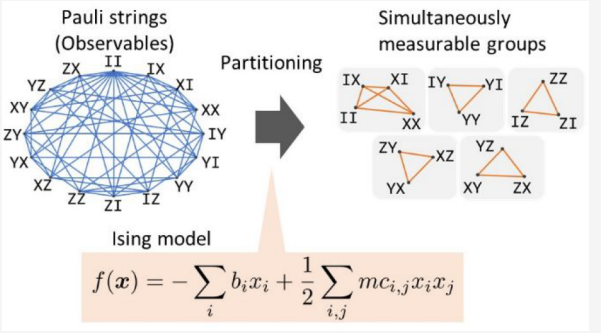
Tomochika Kurita,* Mikio Morita, Hirotaka Oshima, and Shintaro Sato

Cite This: *J. Phys. Chem. A* 2023, 127, 1068−1080

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We propose an efficient algorithm for partitioning Pauli strings into subgroups, which can be simultaneously measured in a single quantum circuit. Our partitioning algorithm drastically reduces the total number of measurements in a variational quantum eigensolver for a quantum chemistry, one of the most promising applications of quantum computing. The algorithm is based on the Ising model optimization problem, which can be quickly solved using an Ising machine. We develop an algorithm that is applicable to problems with sizes larger than the maximum number of variables that an Ising machine can handle (n_{bit}) through its iterative use. The algorithm has much better time complexity and solution optimality than other existing algorithms. We investigate the performance of the algorithm using the second-generation Digital Annealer, a high-performance Ising hardware, for up to 65535 Pauli strings using Hamiltonians of molecules and the full tomography of quantum states. We demonstrate a time complexity of $O(N)$ for $N \leq n_{bit}$ and $O(N^2)$ for $N > n_{bit}$ for the worst case, where N denotes the number of candidate Pauli strings and n_{bit} = 8,192 in this study. The reduction factor, which is the number of Pauli strings divided by the number of obtained partitions, can be 200 at maximum.



1. INTRODUCTION

Quantum computing has the potential to outperform classical computing in computational time.[1] In particular, among its several practical targets, there has been a major advancement in the areas of quantum chemistry.[2] In the current noisy intermediate-scale quantum computing,[3] variational quantum eigensolver (VQE) algorithms are extensively studied for quantum chemistry to calculate ground- and excited-state energies of chemicals,[4,5] including small molecules,[6,7] catalysts, and battery materials.[8,9]

VQE algorithms are designed to solve the Schrödinger equation,

$$H\psi = E\psi \tag{1}$$

using variational methods. To solve it using a quantum computer, the Hamiltonian H and wave function ψ are mapped to $\hat{H}$ and $\hat{\psi}$, respectively, through a second quantization:

$$\hat{H}\hat{\psi} = E\hat{\psi} \tag{2}$$

where $\hat{\psi}$ can be obtained by a quantum computer. To calculate the ground-state energy, a parametrized quantum state $\hat{\psi}(\boldsymbol{\theta})$ is created using a quantum computer. The parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots\}$ are iteratively optimized using a classical computer to minimize the expectation value of the given Hamiltonian $\hat{H}$:

$$E = \min_{\boldsymbol{\theta}} \frac{\langle\hat{\psi}(\boldsymbol{\theta})|\hat{H}|\hat{\psi}(\boldsymbol{\theta})\rangle}{\langle\hat{\psi}(\boldsymbol{\theta})|\hat{\psi}(\boldsymbol{\theta})\rangle} \tag{3}$$

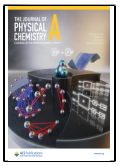
To estimate the expectation value $\langle\hat{\psi}(\boldsymbol{\theta})|\hat{H}|\hat{\psi}(\boldsymbol{\theta})\rangle$ using a quantum computer, the Hamiltonian is decomposed into some Pauli strings:

$$\hat{H} = \sum_i \lambda_i P_i \tag{4}$$

where P_i denotes the i-th Pauli string and $\lambda_i \in \mathbb{R}$ denotes the corresponding weight of P_i.

For the VQE algorithms, the number of Pauli strings scales as $O(n^4)$, where n denotes the number of qubits assigned to spin orbitals of a target molecule by a one-to-one correspondence, because the corresponding Hamiltonians only contain two-body interactions. A quite large n value (i.e., a large number of spin orbitals) leads to a large number of

Received: September 9, 2022
Revised: November 16, 2022
Published: January 18, 2023



https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

required measurements, which limits the ability of quantum computation in general. In principle, to obtain multidimensional information on a target quantum state, we have to prepare its several copies and obtain each one-dimensional information repeatedly through a basis-changing operation prior to each measurement. Furthermore, we need to measure a target quantum state multiple times for each one-dimensional information to obtain the expectation value with some desired precision because each measurement results in the projection of the state onto a measurement basis and we need its average value. This estimation process nature of the expectation value makes the overall algorithm time-consuming; this condition is especially true when the expectation values of Hamiltonians are subject to classical optimization as in the case of VQEs.

Some methods have been proposed to suppress an increase in the number of measurements.[10−18] One of the main methods is the partitioning method, where Pauli strings are partitioned so that their expectation values can be measured simultaneously.[10−14] In this method, a group of Pauli strings is divided into subgroups, and all the components of each subgroup are measured simultaneously using only one circuit (hereafter, such subgroups are called "partitions"). Notably, the partitioning method is also useful for simulating the time evolution of Hamiltonians in terms of reducing algorithmic errors induced via Trotter decomposition[19] and a quantum phase-estimation algorithm with ancilla qubits.[20] To maximize the effectiveness of the simultaneous measurement, the number of partitions should be minimized. However, minimizing the number of partitions is a NP-hard problem. To date, several algorithms have been proposed to address such problems.[10−14] They are mainly based on either the maximum clique searching method[11,14,21] or the graph coloring method.[11−14] For the maximum clique searching method, two algorithms are mainly used: Boppana−Halldórsson algorithm[22] and Bron−Kerbosch algorithm.[23,24] Their time complexity and solution optimality, however, have a trade-off relation. For the Boppana−Halldórsson algorithm, the time complexity along the number of Pauli strings is a polynomial but the optimality of solutions is not guaranteed. The Bron−Kerbosch algorithm can guarantee the optimality but exhibits exponential time complexity. Regarding the graph coloring method, Verteletskyi et al.[11,12] and Hamamura et al.[13] tested the performance of the largest-first method, which has been proven to afford the best performance among various heuristic orderings. This method takes polynomial time with the number of Pauli strings but provides less optimal partitioning results than maximum clique searching algorithms with the Bron−Kerbosch algorithm[14] when the number of Pauli strings exceeds ∼500. An algorithm-specific partitioning method proposes a partitioning scheme using the nature of the qubit-mapping methods of Hamiltonians in quantum chemistry.[14] This partitioning method may not require long time, but the resultant reduction factor, which is the number of Pauli strings divided by the number of obtained partitions, is only 8 at maximum (when the Jordan−Wigner qubit-mapping method is used).

Such classical algorithms with polynomial time scaling can reduce the number of measurements for estimating the expectation values of Hamiltonians to some extent. However, we need to further reduce the number of measurements as far as possible for the following reason. As mentioned above, the estimation of expectation value of the given Hamiltonian is subject to classical optimization (i.e., the expectation value has to be measured per optimization step) and the number of the optimization steps sharply increases with increasing number of parameters. Therefore, the factor of how the number of measurements is reduced using the partitioning method must be evaluated. However, although considerable effort has been made to improve the reduction factor, it cannot reach the factor obtained using the Bron−Kerbosch algorithm (which requires exponential time). Therefore, we believe that an application-specific computer that operates on a different calculation principle from a conventional computer would be necessary for drastic improvement in both time complexity and the resultant reduction factor.

Another method to reduce the total number of measurements is the shadowing method, which is based on classical shadowing.[15,16] The advantage of this method is that in some cases, a fewer number of required measurements can be realized compared with the partitioning method by efficiently determining the basis-changing operation per measurement.[16] However, in the worst case, the method requires the same number of quantum circuits as that of the measurements, which may become an additional and non-negligible cost for hardware experiment. Moreover, to determine per-measurement basis-changing operations, the computational cost of $\Omega(n_{\mathrm{meas}})$ is required, where n_{meas} denotes the total number of measurements. Yen et al.[17] discussed a combination of the shadowing and partitioning methods to further reduce the total number of measurements. From this point of view, it would also be of great help if the computational time for determining the necessary quantum circuits and the number of such circuits for the partitioning method could be considerably reduced.

In this paper, we propose a fast, effective, and versatile algorithm to address such partitioning problems with Ising machines. The proposed algorithm is based on the maximum clique searching method, and we transform maximum clique searching problems into quadratic unconstrained binary optimization (QUBO) problems, which are equivalent to the Ising model optimization problems.[25] This approach allows us to use Ising machines to quickly solve the problems. Using Fujitsu's second-generation Digital Annealer, a hardware architecture designed to efficiently solve QUBO problems,[26] as an Ising machine, we demonstrate that the performance of our algorithm is much better in terms of time complexity and solution optimality by comparing it with existing algorithms (Boppana−Halldórsson algorithm and Bron−Kerbosch algorithm). In addition to the partitioning problem for VQE Hamiltonians, we tested our algorithm on a full tomography of one n-qubit quantum state (where the number of Pauli strings to be measured is $4^n - 1$) to benchmark the results to the theoretical ones.[27] Our algorithm can be applied to the problems larger than the capacity of an Ising machine by using it repeatedly, as will be shown below.

This paper is organized as follows. Section 2 describes the theoretical background and general procedure of performing simultaneous measurements. Section 3 explains our new Ising model−based partitioning algorithm and how the Digital Annealer works. Section 4 describes the performance of the new algorithm in estimating the expectation values of multiple Pauli strings, comparing it with existing maximum clique searching algorithms. Section 5 summarizes this study and discusses future perspectives.

2. SIMULTANEOUS MEASUREMENTS

In this section, we explain the technical background of simultaneous measurements and partitioning. In section 2.1,

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

we describe the relation between the commutativity of Pauli strings and simultaneous measurements. In section 2.2, we present the general procedure of simultaneous measurements.

2.1. Theoretical Background. A simultaneous measurement is based on the fact that the expectation values of two Pauli strings P_1 and P_2 can be simultaneously estimated by applying an appropriate basis-changing operation if and only if they commute each other (i.e., $P_1P_2 = P_2P_1$).[10] Generally, given a quantum state ρ, the expectation value of an observable M is $\mathrm{Tr}(M\rho)$. When the matrix B can diagonalize the matrix M so that $Z_{\{i_j\}} = BMB^{-1}$, this condition leads to

$$\mathrm{Tr}(M\rho) = \mathrm{Tr}[Z_{\{i_j\}}(B\rho B^{-1})] \tag{5}$$

where $Z_{\{i_j\}}$ denotes the tensor matrix of Z and I:

$$Z_{\{i_j\}} = \bigotimes_{j=1}^{n} Z^{i_j} \tag{6}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

where j denotes the qubit index beginning from 1 and $i_j \in \{0, 1\}$ for each j. Thus, we can estimate $\mathrm{Tr}(M\rho)$ by applying B to ρ as a basis-changing operation and performing a projective measurement of the qubits labeled with the sequence $\{j|i_j = 1\}$ along the computational basis (Z-basis).

Here, we suppose that ρ is an n-qubit state and B is a basis-changing operation. For any of $2^n - 1$ possible $\{i_j\}$ (excluding $i_j = 0$ for all the j values), $M_{\{i_j\}}$ exists, which satisfies

$$Z_{\{i_j\}} = BM_{\{i_j\}}B^{-1} \tag{8}$$

As shown in eq 5, eq 8 satisfies

$$\mathrm{Tr}(M_{\{i_j\}}\rho) = \mathrm{Tr}[Z_{\{i_j\}}(B\rho B^{-1})] \tag{9}$$

From eq 8, the commutation relation between $M_{\{i_j\}}$ and $M_{\{i_j\}}$, can be derived as

$$M_{\{i_j\}}M_{\{i_j\}'} = B^{-1}Z_{\{i_j\}}Z_{\{i_j\}'}B = B^{-1}Z_{\{i_j\}'}Z_{\{i_j\}}B = M_{\{i_j\}'}M_{\{i_j\}} \tag{10}$$

for any $\{i_j\},\{i_j\}'$. Therefore, for $M_{\{i_j\}}$ and $M_{\{i_j\}}$, commute for any $\{i_j\},\{i_j\}'$ is the requirement for simultaneous measurements. Moreover, for any partition $\{M_{\{i_j\}}\}$, the existence of a basis-changing gate B that satisfies eq 9 has been proven in a previous study.[10]

The commutativity of Pauli strings has two settings.[14] One is qubit-wise commutativity (QWC), which means that, for every qubit, corresponding Pauli operators commute each other. The other is general commutativity (GC), which means that Pauli strings commute as a whole, while each Pauli operator does not necessarily commute. In the GC setting, for some qubits, corresponding Pauli operators can be anticommuting. Two Pauli strings commute when the number of the anticommuting pairs of Pauli operators is even. When we apply QWC, the basis-changing gate B can be described as a tensor product of a single-qubit gate. Meanwhile, the resultant number of partitions is 3^n for an n-qubit full tomography (each of the partitions contains one Pauli string described as a tensor product of X, Y, and Z). For estimating the expectation values of Hamiltonians for VQE, a previous study[11] shows that the number of partitions is only three times less than the number of Pauli strings. By contrast, when we apply GC, B is described as entangled gates and the number of partitions is expected to be less than that for QWC. For n-qubit full tomography, when GC is applied, $4^n - 1$ Pauli strings can be divided into $2^n + 1$ partitions, each of which contains $2^n - 1$ Pauli strings.[27] In this study, we used GC to investigate the maximum effect of simultaneous measurements.

2.2. General Procedure and Its Time Complexity. Figure 1 shows a scheme for measuring the expectation values

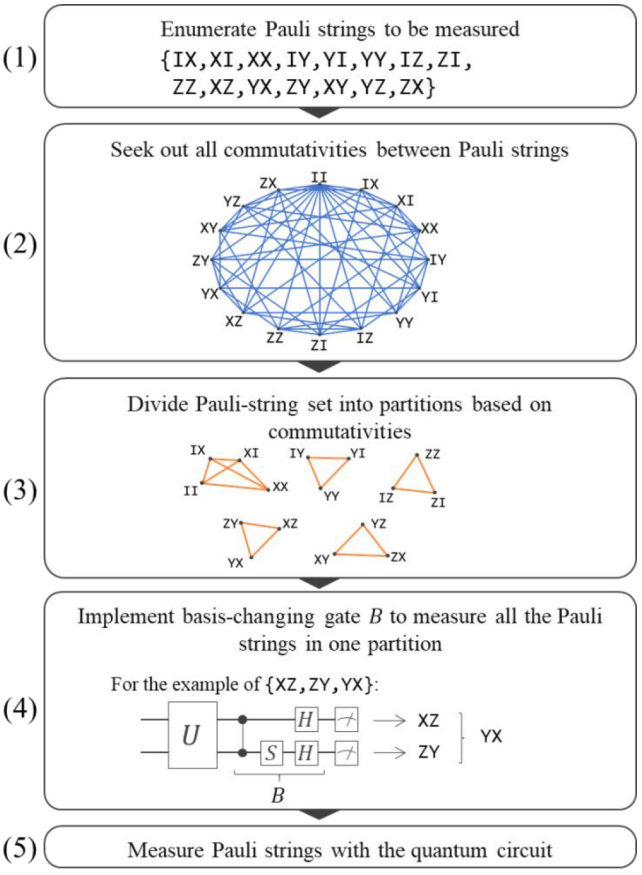


Figure 1. Schematic of measuring the expectation values of multiple Pauli strings using simultaneous measurements.

of multiple Pauli strings using simultaneous measurement and partitioning. After enumerating the Pauli strings that are required to estimate the expectation values (step (1)), the commutativity of each pair of Pauli strings was checked (step (2)). Then, we created partitions where all Pauli strings commute (step (3)). Based on this, we determined the basis-changing gate B for each partition (step (4)), and finally, all the expectation values of the Pauli strings were estimated (step (5)). Using steps (2) and (3) presented in Figure 1, the number of circuits was reduced to the number of partitions.

To evaluate the overall performance of partitioning, the time complexity and solution optimality should be examined. The time complexity of step (2) is $O(N^2n)$, where N denotes the number of Pauli strings and n denotes the number of qubits. Conversely, the time complexity of step (3) strongly depends on the algorithm used for the partitioning.

Two partitioning algorithms are generally used for solving the maximum clique searching problem. The Boppana–Halldórsson algorithm[22] uses a greedy method for creating each partition, which does not necessarily result in a maximum-size partition. It has a roughly quadratic time

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

complexity with no guarantee of optimality, although its worst-case time complexity is not well studied. Meanwhile, the Bron–Kerbosch algorithm[23,24] uses a rigorous method for creating each partition to obtain a maximum-size partition. It has an exponential time complexity of $O(3^{n/3})$ for creating one partition as the worst case,[24] but yields an optimal solution. More details of these algorithms are described in Appendix 1 in the Supporting Information.

3. METHODS

In this section, we describe our proposed Pauli string partitioning methods. Section 3.1 introduces the proposed Ising model-based partitioning algorithm. Sections 3.2 and 3.3 describe the specific partitioning problems we address in this study and the settings of the Digital Annealer, respectively.

3.1. Ising Model-Based Partitioning Algorithm. The proposed algorithm is based on maximum clique searching.[21] A partition with the maximum number of elements is created from the Pauli strings, and the process is repeated with the remaining Pauli strings until no other string remains, as shown in Figure 2. In this algorithm, we break down each partition-creating problem into a QUBO problem, which is equivalent to an Ising model problem[25] and can be solved efficiently using an Ising machine. In this QUBO problem, each Pauli string is assigned a binary variable, and the value of the variables distinguishes whether the corresponding Pauli strings are included in a target partition.

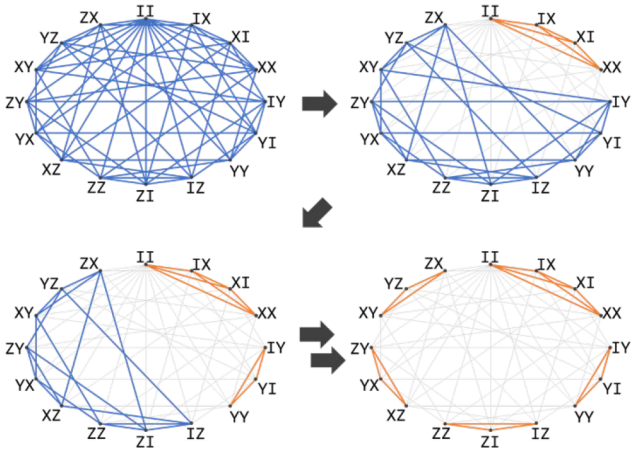


Figure 2. Illustration of the partitioning process for two-qubit full tomography. An edge between two nodes denotes that the corresponding two Pauli strings commute. One partition is described as the node groups, with thick edges colored in orange. In this example, the 16 Pauli strings (including the identity Pauli string) can be divided into five groups.

Suppose that we create one maximum-size partition from the candidate Pauli string group $\{P_1, ..., P_N\}$. To map this problem to a QUBO problem, the cost function should be determined so that it is minimized when the number of Pauli strings in the target partition is maximized. The cost function can be defined as follows:

$$f(x_1, ..., x_N) = -\sum_{1\le i\le N} b_i x_i + \frac{1}{2}\sum_{\substack{1\le i\le N\\ 1\le j\le N}} m c_{i,j} x_i x_j \tag{11}$$

where $x_1, ..., x_N \in \{0,1\}$ denotes the binary variables mapped to Pauli strings. $x_k = 1$ means that the Pauli string P_k is included in a target partition; $x_k = 0$ means otherwise. $b_1, ..., b_N$ denote positive constants, and we set $b_i = 1$ for all i. $c_{1,1}, ..., c_{1,N}, c_{2,1}, ..., c_{N,N}$ denote nonnegative constants, satisfying $c_{i,j} = 0$ if $P_iP_j = P_jP_i$; otherwise, $c_{i,j} = 1$. m denotes a positive constant. In the right-hand of eq 11, the first term means that the cost function decreases as the number of Pauli strings in the target clique increases. The second term means that if pairs of Pauli strings that do not commute exist in a partition, then the cost function increases. The value of m must be selected to satisfy the condition that the contribution of the second term is zero when the global minimum of eq 11 is realized. $m > 1$ satisfies this condition for any case (for details, see Appendix 2 in the Supporting Information), and we set $m = 2$ in this study. When the cost function reaches the global minimum, the obtained subgroup of Pauli strings $\{P_i|x_i = 1\}$ can be considered the partition that has the largest number of elements in a Pauli string set of interest. The pedagogical example of this Ising model mapping is presented in Figure 3.

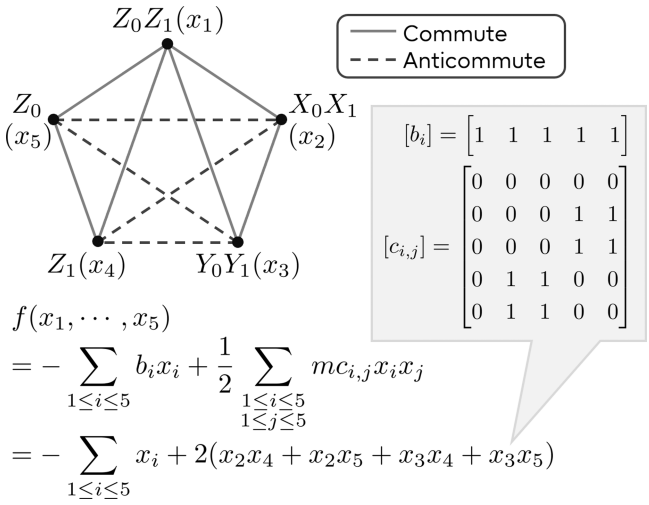


Figure 3. Illustration of the mapping of partitioning problems to the Ising model (eq 11), where we set $m = 2$. Here, we are creating a maximum-size partition from $\{Z_0Z_1, X_0X_1, Y_0Y_1, Z_1, Z_0\}$. The result of this equation is minimized when $\{x_1, x_2, x_3, x_4, x_5\} = \{1, 1, 1, 0, 0\}, \{1, 0, 0, 1, 1\}$, which reflect the maximum-size partitions.

When the number of variables $\{x_i\}$ is less than or equal to the number of variables that an Ising machine can handle (n_{bit}), we can directly determine $\{x_i\}$, which minimizes the target cost function using an Ising machine.[26,28,29] In that case, the number of variables necessary for the Ising machine is the number of Pauli strings. Ising machines are designed to solve such Ising-type optimization problems by setting the initial values of $\{x_i\}$ first, finding more optimal values that decrease the target cost function and updating them iteratively. Although the computational principle is different in each problem, Ising machines are generally designed to solve these problems faster than conventional computers.

When the number of Pauli strings exceeds n_{bit}, additional procedures are required. Given the Pauli string group $\mathcal{P} = \{P_1, ..., P_{n_{\mathrm{bit}}}, ..., P_N\}$, the partition C is created using the following procedures in our Ising model-based algorithm, as shown in Figure 4. (1) A subgroup $\mathcal{S}_{\mathcal{P}}$ that comprises the first n_{bit} elements of $\mathcal{P}$ is defined, and a partition $\mathcal{D} = \{P_{d_1}, ..., P_{d_p}\}$ is created using $\mathcal{S}_{\mathcal{P}}$ by solving the corresponding QUBO (eq

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

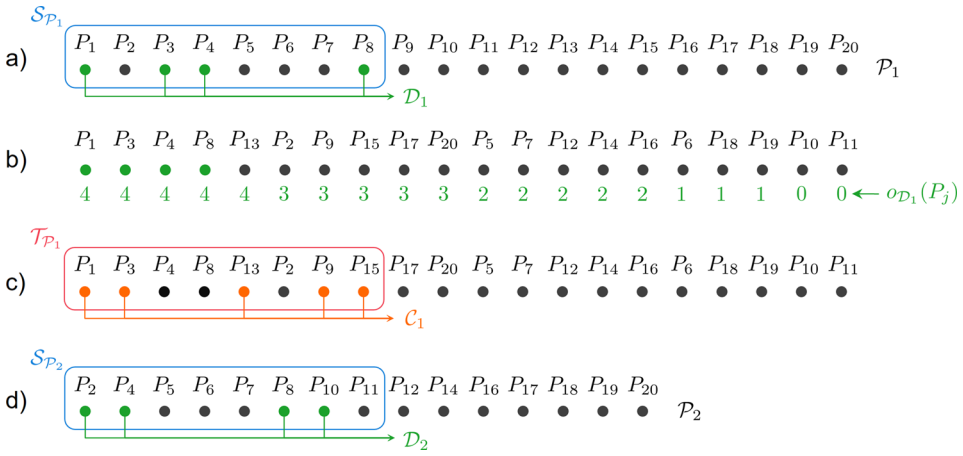


Figure 4. Illustration of the partitioning procedure when $N > n_{bit}$ ($N = 20$ and $n_{bit} = 8$). (a) Determine $\mathcal{S}_{\mathcal{P}_1}$ as the first n_{bit} elements of $\mathcal{P}_1$ and calculate $\mathcal{D}_1$ using the Digital Annealer. (b) Calculate $o_{\mathcal{D}_1}(P_j)$ for each $P_j \in \mathcal{P}_1$ and then sort $\mathcal{P}_1$ in a descending order of $o_{\mathcal{D}_1}(P_j)$. (c) Determine $\mathcal{T}_{\mathcal{P}_1}$ as the first n_{bit} elements of the sorted $\mathcal{P}_1$ and calculate C_1 using the Digital Annealer. (d) Determine $\mathcal{P}_2 = \mathcal{P}_1 - C_1$ and sort it in an ascending order of index j. Then, continue partitioning until $\mathcal{P}_k = \emptyset$.

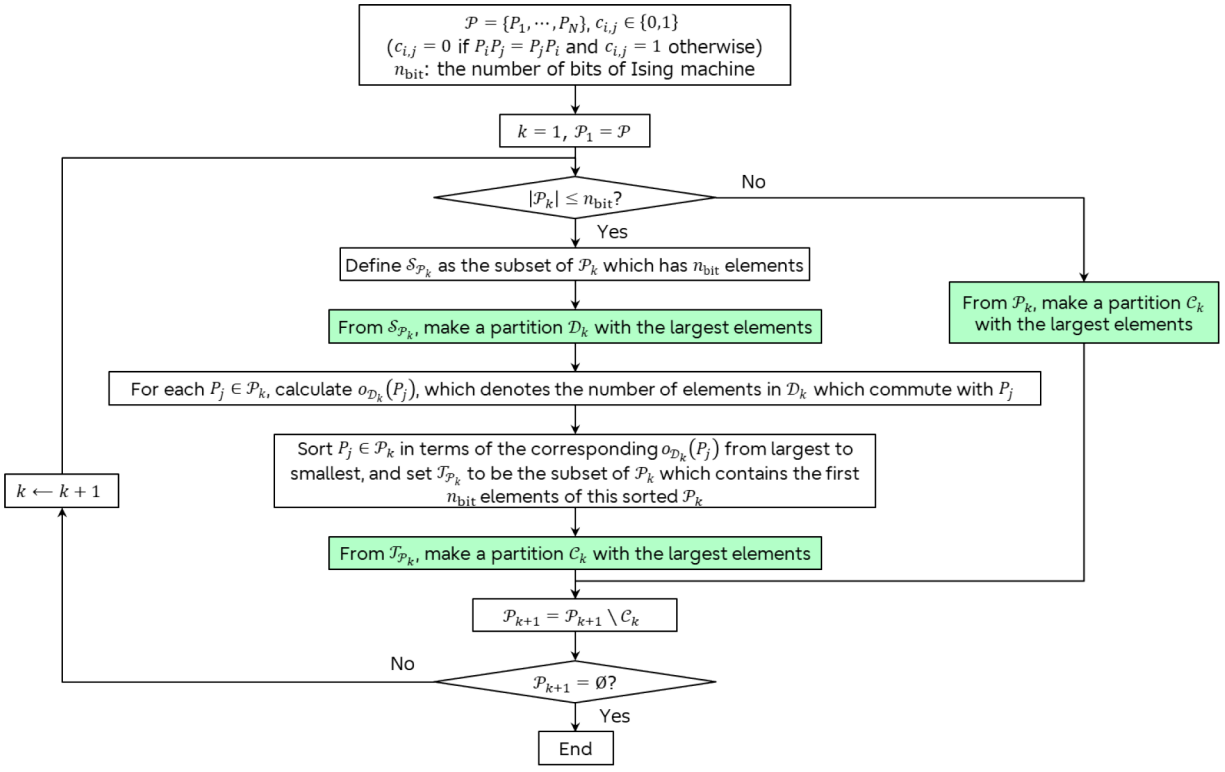


Figure 5. Flowchart of partitioning with the Ising model-based algorithm. Ising machines are employed in the green-colored steps.

11) problem with an Ising machine. (2) $o_{\mathcal{D}}(P_j)$ is defined for each $P_j \in \mathcal{P}$, which denotes the number of Pauli strings $P_i \in \mathcal{D}$ that satisfy $P_iP_j = P_jP_i$, and $\mathcal{P} = \{P_j\}$ is sorted in a descending order of $o_{\mathcal{D}}(P_j)$. (3) A subgroup $\mathcal{T}_{\mathcal{P}} \subset \mathcal{P}$ is defined, such that $\mathcal{T}_{\mathcal{P}}$ comprises the first n_{bit} elements of the sorted $\{P_j\}$. (4) A partition $C = \{P_{c_1}, ..., P_{c_q}\}$ is created using $\mathcal{T}_{\mathcal{P}}$ by solving the corresponding QUBO (eq 11) problem with the Ising machine. In this case, the Ising machine is used twice (in creating $\mathcal{D}$ and then C) for a single cycle of the partitioning process. Procedures from (2) to (4) can be repeated r times to create an optimal partition, with $\mathcal{D} \leftarrow C$ being updated between procedures (2) and (4). In this study, we set $r = 1$. The flowchart of this algorithm is presented in Figure 5, and the overall partitioning algorithm is shown in Chart 1.

We investigated the performance of this Ising model–based algorithm in terms of time complexity and solution optimality compared with those of the Boppana–Halldórsson and Bron–Kerbosch algorithms, both of which are also based on maximum clique searching. For the Boppana–Halldórsson and Bron–Kerbosch algorithms, we used NetworkX[30] implemented in Python because both algorithms are implemented in it and it allows us to benchmark them easily.

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

Chart 1. Ising Model-Based Algorithm

Algorithm 1: Ising model–based algorithm

Input: a set of Pauli strings $\mathcal{P} = \{P_1, \cdots, P_N\}$, interaction coefficients $\{c_{i,j} | 1 \le i \le N, 1 \le j \le N\}$ ($c_{i,j} = 0$ if $P_iP_j = P_jP_i$, $c_{i,j} = 1$ otherwise), coefficients $\{b_i | 1 \le i \le N\}$ ($b_i = 1$ for any i), available number of variables of Ising machine n_{bit}, positive coefficient m, positive integer r
Set: $\mathcal{P}_1 = \{P_1, \cdots, P_N\}$
Set: $k = 1$
while $\mathcal{P}_k \neq \emptyset$ **do**
  if $|\mathcal{P}_k| > n_{\text{bit}}$ **then**
    Sort $P_j \in \mathcal{P}_k$ in ascending order of index j
    Set: $\mathcal{S}_{\mathcal{P}_k} \subset \mathcal{P}_k$ such that $\mathcal{S}_{\mathcal{P}_k}$ consists of the first n_{bit} elements of the sorted $\mathcal{P}_k$
    Determine binary variables $\{x_i | P_i \in \mathcal{S}_{\mathcal{P}_k}\}$ by Ising machine, which minimize eq 11 subject to $x_i = 0$ for all of $\{x_i | P_i \notin \mathcal{S}_{\mathcal{P}_k}\}$
    Set: $\mathcal{D}_k = \{P_i | P_i \in \mathcal{S}_{\mathcal{P}_k}, x_i = 1\}$
    for $l = 1, \cdots, r$ **do**
      Set: $o_{\mathcal{D}_k}(P_j) = |\{P_i | c_{i,j} = 0, P_i \in \mathcal{D}_k\}|$ for each $P_j \in \mathcal{P}_k$
      Sort $P_j \in \mathcal{P}_k$ in descending order of $o_{\mathcal{D}_k}(P_j)$
      Set: $\mathcal{T}_{\mathcal{P}_k} \subset \mathcal{P}_k$ such that $\mathcal{T}_{\mathcal{P}_k}$ consists of the first n_{bit} elements of the sorted $\mathcal{P}_k$
      Determine binary variables $\{x_i | P_i \in \mathcal{T}_{\mathcal{P}_k}\}$ by Ising machine, which minimize eq 11 subject to $x_i = 0$ for all of $\{x_i | P_i \notin \mathcal{T}_{\mathcal{P}_k}\}$
      $\mathcal{D}_k \leftarrow \{P_i | P_i \in \mathcal{T}_{\mathcal{P}_k}, x_i = 1\}$
      if $l = r$ **then**
        Set: $\mathcal{C}_k = \mathcal{D}_k$
      end if
    end for
  else
    Determine binary variables $x_1, \cdots, x_n$ which minimize eq 11 by Ising machine
    Set: $\mathcal{C}_k = \{P_i | x_i = 1\}$
  end if
  Set: $\mathcal{P}_{k+1} = \mathcal{P}_k - \mathcal{C}_k$
  $k \leftarrow k + 1$
end while
Output: $\mathcal{C}_1, \mathcal{C}_2, \cdots$

3.2. Overview of the Digital Annealer and Its Settings. In this study, we used the Digital Annealer as an Ising machine for the following reasons. First, all the variables in the Digital Annealer are fully connected, which is preferable for the problems under consideration because for any variable x_i, roughly $N/2$ variables x_j exist that satisfy $c_{i,j} \neq 0$. Second, the Digital Annealer can rapidly search the $2^{n_{\text{bit}}}$ space (where n_{bit} denotes the number of variables in the Digital Annealer) to obtain a (globally) minimum value of QUBO problems, such as eq 11. It is because of an efficient parallel trial scheme for a Markov chain Monte Carlo method combined with massive parallelization and a dynamic escaping function from local minima.[31] The Digital Annealer can generally solve such QUBO problems much faster than simulated annealing conducted on a classical computer.[31]

All calculations using Digital Annealer were conducted under a computational environment prepared for research use. The second-generation Digital Annealer[26] that we used herein allows us to tune the available number of variables up to 8,192. In this study, without further notice, all calculations using the Digital Annealer were conducted with n_{bit} = 8,192. For each calculation, the number of Monte Carlo steps was fixed to 10^8.

3.3. Partitioning Problems. In this study, we set the partitioning problems of n-qubit full tomography (n = 1, ..., 8) and estimating the expectation values of Hamiltonians for VQE. For the VQE problems, we set the target molecules as H_2, LiH, H_2O, and CH_4 in the STO-3G basis set and BeH_2, H_2O, N_2, and NH_3 in the 6-31G basis set. To create corresponding Hamiltonians, we set their molecular configurations by referring to ref 11 and used the Jordan–Wigner qubit-mapping method. For each molecule in the STO-3G basis set, we assumed a variable number of spatial orbitals in an active space, as summarized in Table 1. For each partitioning problem, we excluded the identity Pauli string $\otimes_{j=1}^{n} I$ because it commutes with all other Pauli strings and its expectation value is always 1.

For the full-tomography Pauli string set, we performed indexing of Pauli strings in the ascending order of $\sum_{i=1}^{n} p^{(i)} 4^{n+1-i}$, where $p^{(i)}$ = 0 if the Pauli operator of ith qubit is I and, similarly, 1 if X, 2 if Y, and 3 if Z. For estimating the expectation values of Hamiltonians for VQE, we performed indexing of Pauli strings along with the OpenFermion ordering.[11,32]

4. RESULTS AND DISCUSSIONS

4.1. Time Complexity and Solution Optimality when $N \le n_{\text{bit}}$. In this section, we discuss the time complexity and

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

Table 1. Pauli String Set for VQE (Excluding Identity)

molecules	basis set	qubit mapping	# of spatial orbitals	# of qubits	# of Pauli strings
H_2	STO-3G	Jordan−Wigner	1	2	3
H_2	STO-3G	Jordan−Wigner	2	4	14
LiH	STO-3G	Jordan−Wigner	3	6	117
LiH	STO-3G	Jordan−Wigner	4	8	192
LiH	STO-3G	Jordan−Wigner	5	10	275
LiH	SLO-3G	Jordan−Wigner	6	12	630
H_2O	STO-3G	Jordan−Wigner	4	8	220
H_2O	SLO-3G	Jordan−Wigner	5	10	311
H_2O	STO-3G	Jordan−Wigner	6	12	740
H_2O	STO-3G	Jordan−Wigner	7	14	1389
CH_4	STO-3G	Jordan−Wigner	4	8	240
CH_4	STO-3G	Jordan−Wigner	5	10	591
CH_4	STO-3G	Jordan−Wigner	6	12	1518
CH_4	STO-3G	Jordan−Wigner	7	14	3005
CH_4	STO-3G	Jordan−Wigner	8	16	5236
CH_4	STO-3G	Jordan−Wigner	9	18	8479
BeH_2	6-31G	Jordan−Wigner	13	26	9203
H_2O	6-31G	Jordan−Wigner	13	26	12731
N_2	6-31G	Jordan−Wigner	18	36	34622
NH_3	6-31G	Jordan−Wigner	15	30	52805

solution optimality for Pauli string partitioning problems when the number of Pauli strings N subject to partitioning does not exceed n_{bit}. Figure 6 shows the plots of algorithm runtime t against the number of Pauli strings N using the three algorithms, with their regression curves in the form of $t = aN^b$. The time complexity on N for the Ising model−based algorithm within the range of $N \leq n_{bit} = 8192$ was estimated as $N^{0.52}$. This condition contrasts with $N^{2.57}$ for the Boppana−Halldórsson algorithm and $N^{5.41}$ for the Bron−Kerbosch algorithm. These results suggest that the Ising model−based algorithm is the most scalable algorithm among the three in terms of time complexity. As shown in Figure 6, with the processing ability of our laptop resources, the Ising model−based algorithm showed better performance than the Bron−Kerbosch algorithm when $N \geq 300$ and the Boppana−Halldórsson algorithm when $N \geq 2000$. Although the runtime performances of the Boppana−Halldórsson and Bron−Kerbosch algorithms depend on how much computer resources one can utilize, the threshold number of Pauli strings N_{th} exists, such that the Ising model-based algorithm shows the best performance when $N > N_{th}$.

We investigated the performance guarantee for the time complexity for the Ising model-based algorithm. Each Monte Carlo step in a Digital−Annealer calculation takes the same amount of time.[31] Thus, the time for creating one partition is constant in our setting, where the number of Monte Carlo steps per Digital−Annealer calculation is constant. Figure 7 shows that the overall runtime t of Algorithm 1 in Chart 1 depends almost linearly on the resultant number of partitions. Here, creating one partition takes 4 to 5 s. With the introduction of τ, which denotes the average runtime for one

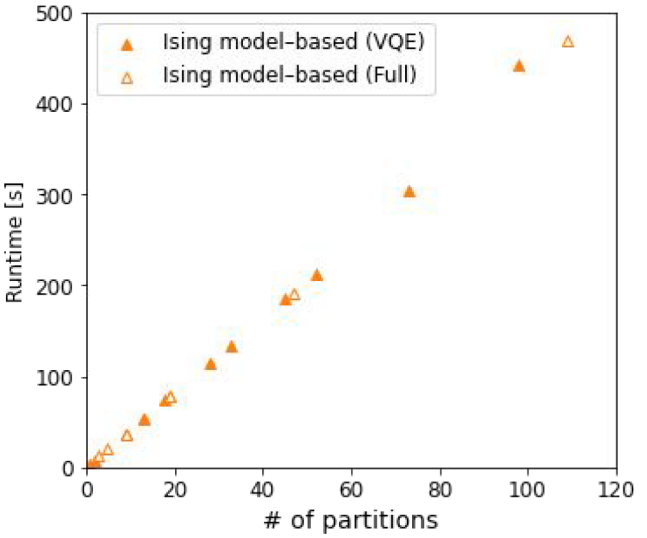


Figure 7. Plot of the partitioning runtime along the resultant number of partitions using the Ising model-based algorithm.

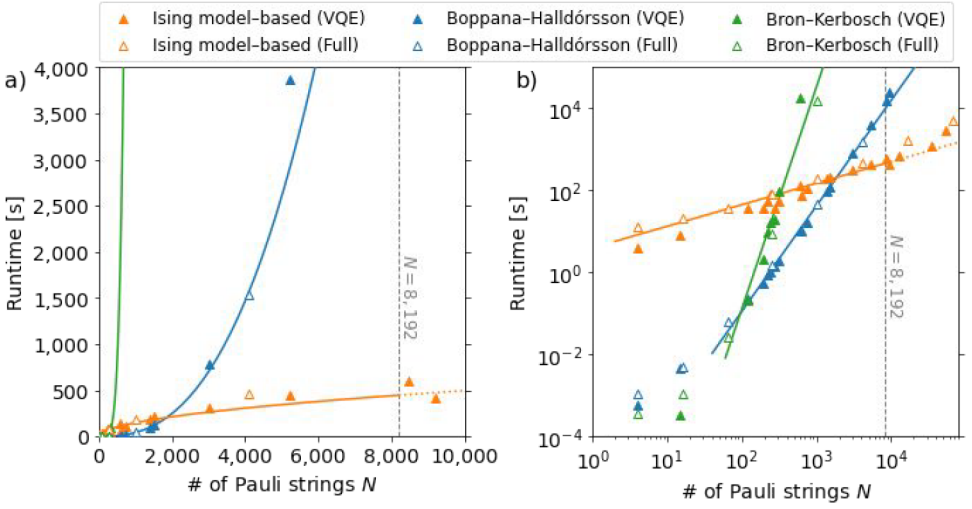


Figure 6. Plot of the runtime along the number of observables obtained using the Ising model-based algorithm (orange), Boppana−Halldórsson algorithm (blue), and Bron−Kerbosch algorithm (green). (a) Plotted in the linear axis, (b) plotted in the logarithmic axis. The regression curve for each algorithm is in the $t = aN^b$ form. The regression curve for the Ising model−based algorithm is determined based on the data plots for $N <$ 8192.

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

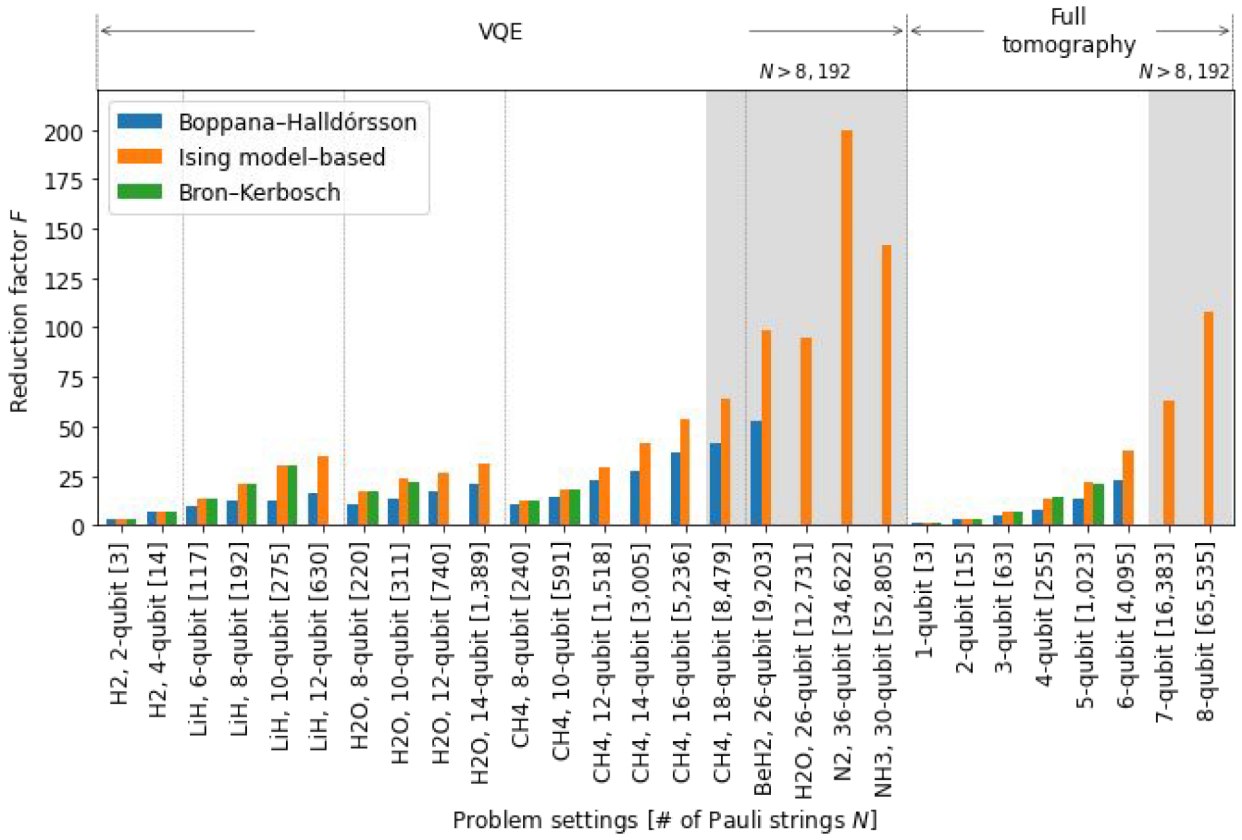


Figure 8. Bar plot of the reduction factor F (defined in the main text) using the Ising model-based algorithm (orange), Boppana−Halldórsson algorithm (blue), and Bron−Kerbosch algorithm (green). For the Ising model-based algorithm, we set n_{bit} = 8192. For the Boppana−Halldórsson and Bron−Kerbosch algorithms, some results are not shown because they are assumed to take more than 10 h.

Digital−Annealer calculation, the Ising model-based algorithm runtime is written as

$$t = \frac{\tau N}{F} \tag{12}$$

where F denotes the reduction factor,[14] which is the value defined as the number of Pauli strings N divided by the resultant number of partitions. The observed dependence $t \propto N^{0.52}$ reflects that the dependence of F on N is $F \propto N^{0.48}$. However, it also reflects the specific characteristic of partitioning problems, i.e., the Pauli strings for VQE observables and full tomography. This dependence of F cannot apply to all types of partitioning problems. The performance guarantee can be investigated by assuming that F = 1 as the worst case. With its application to eq 12, the worst-case time complexity can be confirmed as $O(N)$, which still has a lower dimension than those of the Bron−Kerbosch and Boppana−Halldórsson algorithms.

Next, we discuss the solution optimality of each algorithm. We evaluated it using the reduction factor F; a large reduction factor means a high solution optimality. Figure 8 shows the reduction factor for each partitioning problem. When the number of Pauli strings is small (<20), the reduction factors are similar among the algorithms. By contrast, when the number of Pauli strings increases, the resultant reduction factors of the Ising model-based algorithm and Bron−Kerbosch algorithm become larger than that of the Boppana−Halldórsson algorithm. This result is obtained because of the characteristics of the three algorithms. The Ising model-based algorithm and Bron−Kerbosch algorithm maximize the number of elements of a partition-per-partition creation process, whereas the Boppana−Halldórsson algorithm does not always maximize the number of elements of one partition because it incorporates the greedy approach. This point can be observed in Figure 9; for the Boppana−Halldórsson algorithm, the number of elements of each

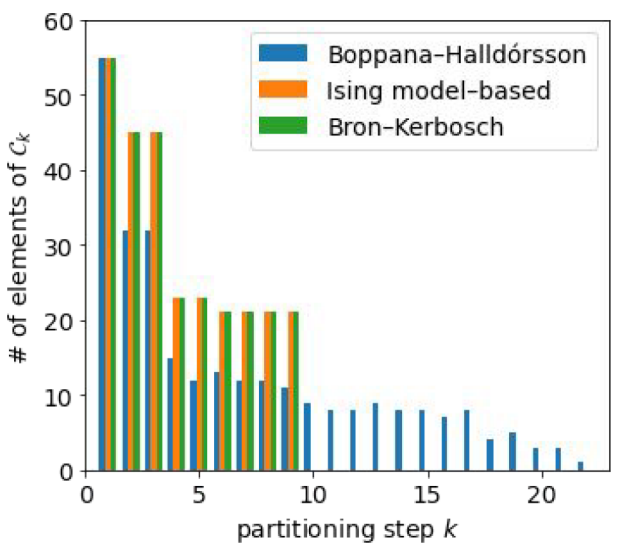


Figure 9. Number of Pauli strings of each partition as a result of partitioning 275 Pauli strings (LiH as a basis set of STO-3G and five spatial orbitals in the active space).

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

partition is smaller than that for the other algorithms. Notably, for n-qubit full tomography for $n \geq 4$, the resultant reduction factor F is less than $2^n + 1$, which is proven to be the maximum value.[27] The results show that the maximum reduction factor has not yet been obtained even via our method. This is probably because a partition that is chosen in the one partition-creating process is not always the ideal one that realizes the maximum effect. Each partition-creating process affects the subsequent partition-creating processes, and the remaining Pauli string groups may often have only partitions with less than $2^n - 1$ Pauli strings.

4.2. Time Complexity and Solution Optimality when $N > n_{bit}$. Section 4.1 presents the excellent partitioning performance of the Ising model-based algorithm for $N \leq n_{bit}$. However, in quantum chemical calculations, N may exceed n_{bit}. In this section, we examine the performance of the Ising model-based algorithm when $N > n_{bit}$.

We discuss the solution optimality first. We have shown the performance of the Ising model-based algorithm when n_{bit} = 8192 and N > 8192 in Figures 6 and 8. The runtimes required using the Ising model-based algorithm are within 1.5 h and remain much shorter than those required in the Boppana–Halldórsson algorithm even when N > 8,192 (Figure 6), and a much better solution optimality (i.e., larger reduction factor F) is realized. We demonstrated that $F \approx 200$ when solving the partitioning problem of N = 34,622 (Figure 8). This result strongly suggests that the extension of the Ising model-based algorithm to the cases of $N > n_{bit}$ is effective.

However, solution optimality and time complexity for the case of $N > n_{bit}$ are anticipated to be quantitatively different from those for the case of $N \leq n_{bit}$. In the following subsection, we will discuss their differences in detail.

4.2.1. Solution Optimality. When $N > n_{bit}$, a natural assumption is that the performance of the Ising model-based algorithm degrades as N increases with respect to n_{bit}. To confirm this, we compared the Ising model-based algorithm performance in the cases of n_{bit} = 8192 and n_{bit} = 1024, as shown in Table 2. The number of partitions is larger when n_{bit} = 1024, which leads to the decrease in the reduction factor F. This finding suggests that the size of n_{bit} indeed affects the performance.

Figure 10 shows the number of elements of each partition C_k (k = 1, ..., 100) for some of the partitioning results for N > 8192 with the conditions of n_{bit} = 8192 and n_{bit} = 1024. For n_{bit} = 1024, the resultant number of elements of each partition C_k is substantially smaller than that for n_{bit} = 8192, especially when a partition with >200 elements. This condition can be a direct cause of the increasing number of partitions. However, executing two Digital–Annealer calculations (one to determine $\mathcal{D}_k$ and another to determine C_k) is more effective than executing one Digital–Annealer calculation to determine $\mathcal{D}_k$ and then regarding it as C_k because the number of elements of C_k is larger than that of $\mathcal{D}_k$ for each k (Figure 10). Furthermore, the ratio of the number of elements of C_k to that of $\mathcal{D}_k$ is greater in the case of n_{bit} = 1024 than in the case of n_{bit} = 8192 (Figure 11). This finding suggests that the effect of two Digital–Annealer calculations is greater when the size of N with respect to n_{bit} is larger.

To further investigate how the relation between N and n_{bit} influences its performance, a convenient step is to introduce a parameter D, which denotes the relative dimension of the partitioning problem against n_{bit}. We define D as

Table 2. Partitioning Performance of the Ising Model-Based Algorithm (n_{bit} = 8192 Mode and n_{bit} = 1024 Mode) and Boppana–Halldórsson Algorithm in the Case of N > 8192[a]

						Ising model-based algorithm (n_{bit} = 8192)				Ising model-based algorithm (n_{bit} = 1024)				Boppana-Halldórsson algorithm		
problem type	molecule	# of spatial orbitals	basis set	# of qubits	# of Pauli strings	D	# of partitions	reduction factor	# of DA calculations	calculation time	D	# of partitions	reduction factor	# of DA calculations	calculation time	# of partitions	reduction factor	calculation time
VQE	CH_4	9	STO-3G	18	8479	1.04	132	64.2	134	598.7	8.28	136	62.3	222	903.6	203	41.8	15795.2
VQE	BeH_2	13	6-31G	26	9203	1.12	93	99.0	96	421.0	8.99	97	94.9	156	637.1	176	52.3	24525.9
VQE	H_2O	13	6-31G	26	12731	1.56	134	95.0	153	711.3	12.43	139	91.6	234	953.4	N/A	N/A	N/A
VQE	N_2	18	6-31G	36	34622	4.23	173	200.1	245	1153.3	33.81	198	174.9	356	1450.2	N/A	N/A	N/A
VQE	NH_3	15	6-31G	30	52805	6.45	372	141.9	586	2915.0	51.57	471	112.1	892	3635.9	N/A	N/A	N/A
full tomography				7	16383	2.00	261	62.8	338	1601.0	16.00	268	61.1	470	1906.0	N/A	N/A	N/A
full tomography				8	65335	8.00	609	107.6	985	5027.1	64.00	661	99.1	1246	5073.7	N/A	N/A	N/A

[a]For each problem and each Ising model-based algorithm mode, the relative problem dimension D (defined in the main text), number of resultant partitions, reduction factor, number of Digital–Annealer calculations, and calculation time are shown.

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

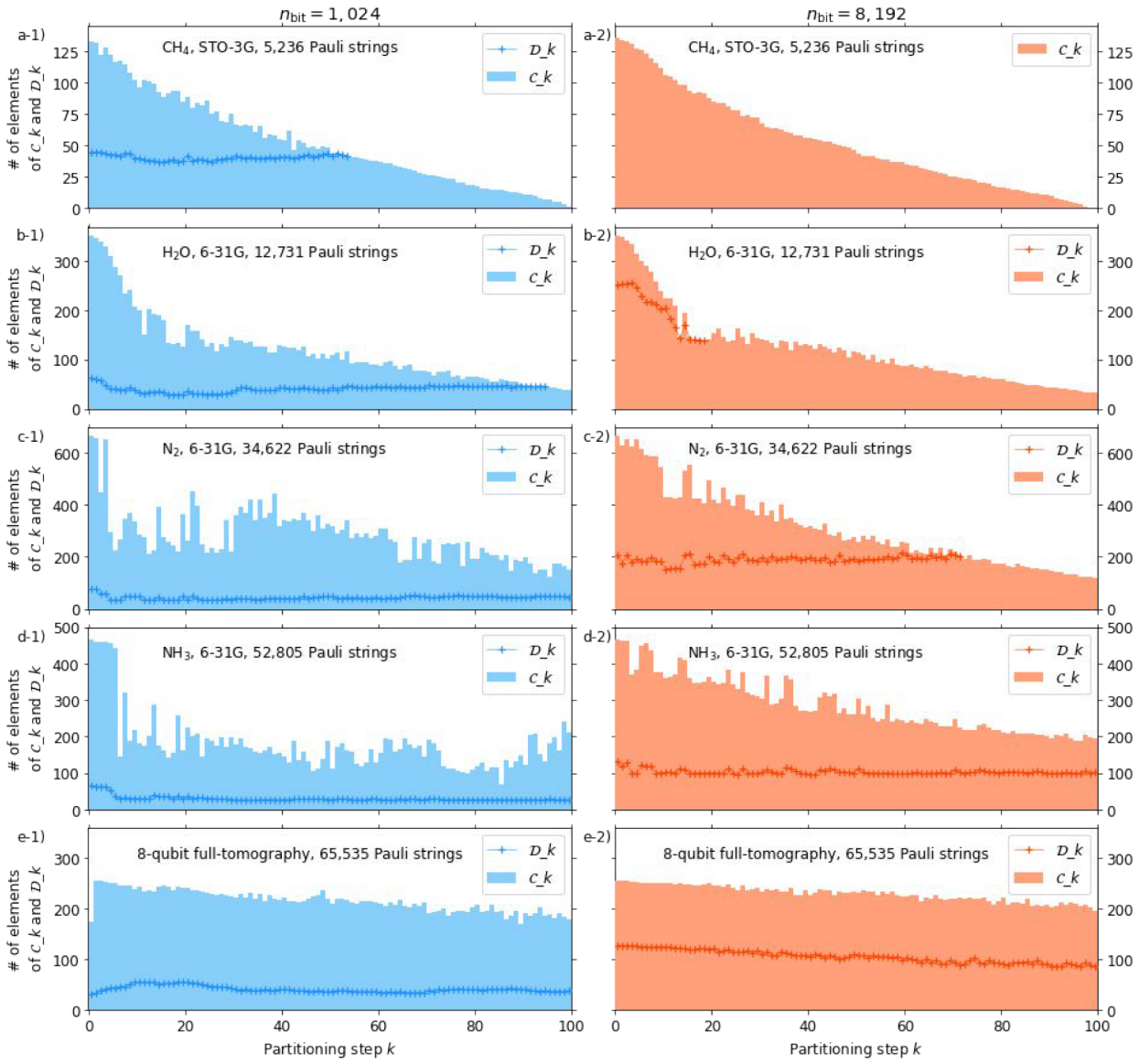


Figure 10. Bar plots of the number of elements of C_k and the number of elements of $\mathcal{D}_k$ (as defined in Algorithm 1 in Chart 1) per partitioning step k (in a range of $1 \leq k \leq 100$) for partitioning problems of (a) CH_4 in STO-3G basis set (8 spatial orbitals), (b) H_2O in 6-31G basis set (13 spatial orbitals), (c) N_2 in 6-31G basis set (18 spatial orbitals), (d) NH_3 in 6-31G basis set (15 spatial orbitals), and (e) eight-qubit full tomography.

$$D \equiv \begin{cases} N/n_{bit} & (\text{if} \quad N > n_{bit}) \\ 1 & (\text{if} \quad N \leq n_{bit}) \end{cases} \tag{13}$$

We first investigated the performance of the Ising model-based algorithm for the partitioning problems of $N > 8192$ with various D (by varying $n_{bit} \in \{8192, 4096, 2048, 1536, 1024, 768, 512, 384, 256, 192\}$ on solving the same partitioning problem). Figure 12 shows the logarithmic plots of the resultant reduction factor F_D along D. The reduction factor F_D slightly decreases with increasing D for all the N values. Therefore, we can expect that the reduction factor is maximum under the $D = 1$ ($N \leq n_{bit}$) condition.

Then, we investigated the performance of the Ising model-based algorithm when $N \leq 8192$, with $D \geq 1$, including $D = 1$. For these calculations, we set $n_{bit} \in \{8192, 4096, 2048, 1536, 1024, 512, 256, 192, 128, 96, 64, 48\}$. The reduction factor F_D decreases as D increases, but its ratio to F_1, that is,

$$p_D \equiv \frac{F_D}{F_1} \tag{14}$$

is not less than 0.9 until $D \approx 10$. As d increases above ~10, we observed that p_D substantially decreases. When D is sufficiently large, the dependence of p_D on d is denoted by $O(D^{-1})$ (see Appendix 3 in the Supporting Information).

These results show that although the Ising model-based algorithm is effective even when $N > n_{bit}$, the larger n_{bit} provides better optimality for partitioning the problems with $N > n_{bit}$.

4.2.2. Time Complexity. Next, we discuss the time complexity. As described in Figure 6, the partitioning runtime t for $N > 8192$ shows some deviations from the extrapolated regression curve ($t = aN^{0.52}$), which is determined from the runtime data for $N \leq 8192$. When $N > n_{bit}$, the partitioning runtimes would be longer owing to the following factors: (i) a greater number of resultant partitions leads to more Digital−

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

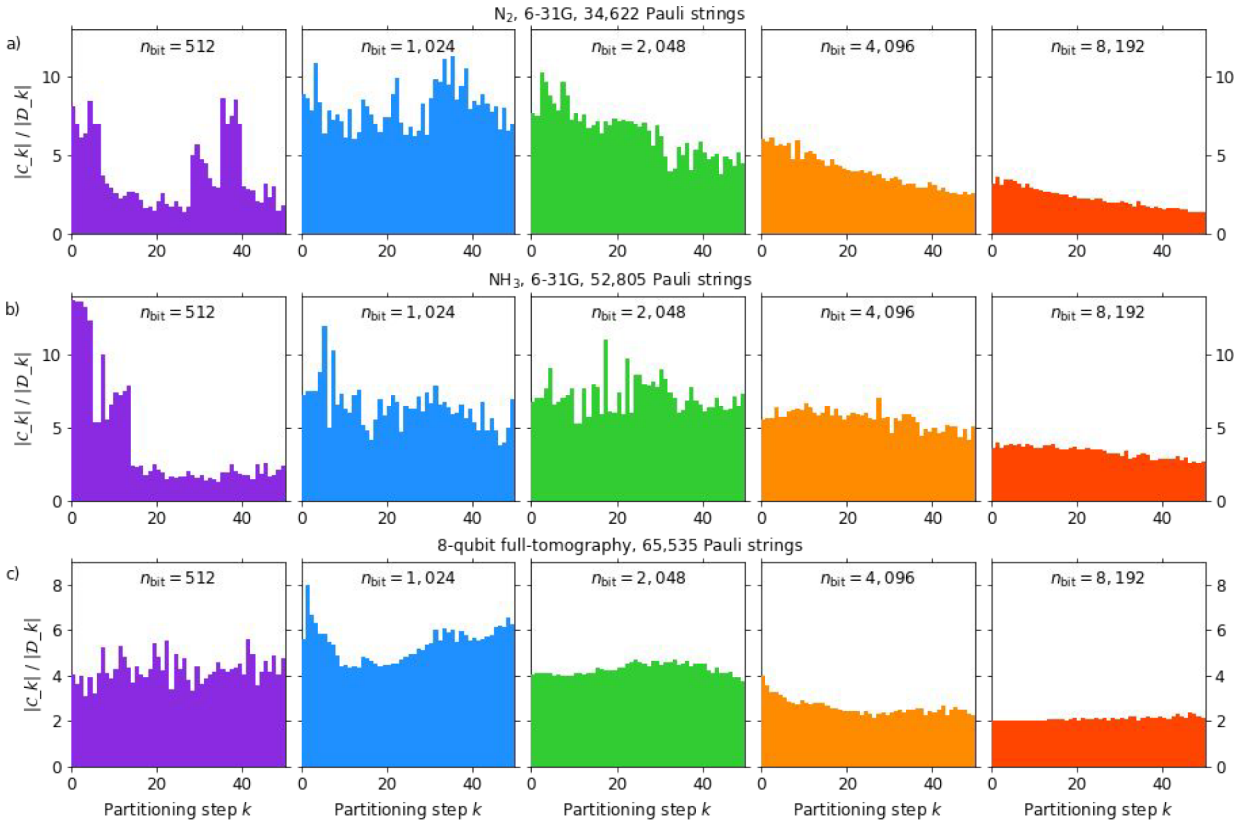


Figure 11. Bar plots of the number of elements of C_k divided by that of $\mathcal{D}_k$ (as defined in Algorithm 1 in Chart 1) per partitioning step k (in a range of $1 \leq k \leq 50$) for partitioning problems of (a) N_2 in the 6-31G basis set (18 spatial orbitals), (b) NH_3 in the 6-31G basis set (15 spatial orbitals), and (c) eight-qubit full tomography.

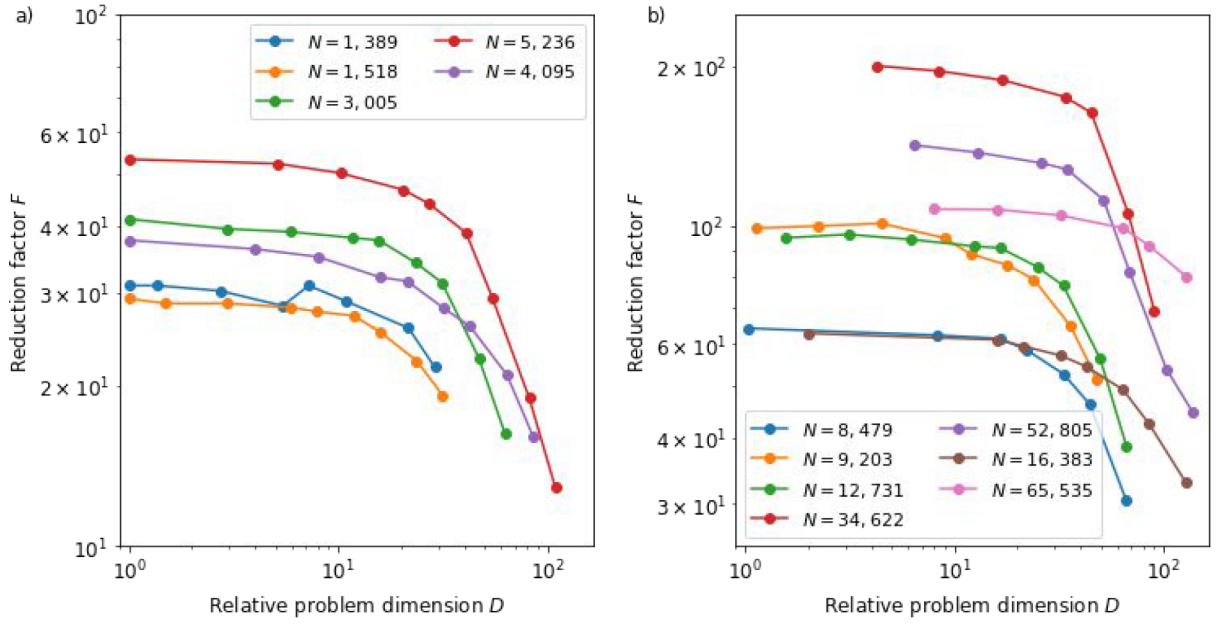


Figure 12. Plots of the reduction factor F as a function of the relative problem dimension D for the partitioning problems of (a) $N \leq 8192$ and (b) $N > 8192$.

Annealer calculation step and (ii) two Digital−Annealer calculations are required to create a single partition as long as the number of remaining Pauli strings exceeds n_{bit}. The algorithm runtime t_D for a relative dimension D is described as

$$t_D = \frac{\tau_D(1+s)N}{F_D} \tag{15}$$

where s denotes the ratio of the number of partitions whose creations require two Digital−Annealer calculations to that of all partitions and τ_D denotes the average time for one Digital−

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068−1080

Annealer calculation. From eqs 12, 14 and 15, t_D can be rewritten as

$$t_D = \frac{\tau_D}{\tau_1}(1 + s)p_D^{-1}t_1 \tag{16}$$

As described in eq 16, three contributions (τ_D/τ_1, $1 + s$, and p_D^{-1}) are made to the deviation of t_D from t_1. Among them, within the range of our simulation ($N \leq 65535$), the difference between the actual and estimated runtimes from the extrapolated regression curve (in the case of $n_{bit} = 8192$) is mainly contributed by $1 + s$, that is, up to 1.6. p_D^{-1}, which is expected to be $\leq$1.1 because $D \leq 8$ and τ_D/τ_1 is ~1. However, when a partitioning problem with an even larger number of Pauli strings N is assumed to be solved, the contribution of $1 + s$ is $O(1)$ because $s \leq 1$ by definition. Similarly, $\tau_D/\tau_1 = O(1)$. The contribution of p_D^{-1} becomes the main factor of time complexity, which evolves as $O(N)$ (Figure 13 and Appendix 3). Therefore, with the assumption of $t_{D=1} \propto N^{0.52}$, even when $N > 8192$, the time complexity for $N > 8192$ is estimated as $N^{1.52}$.

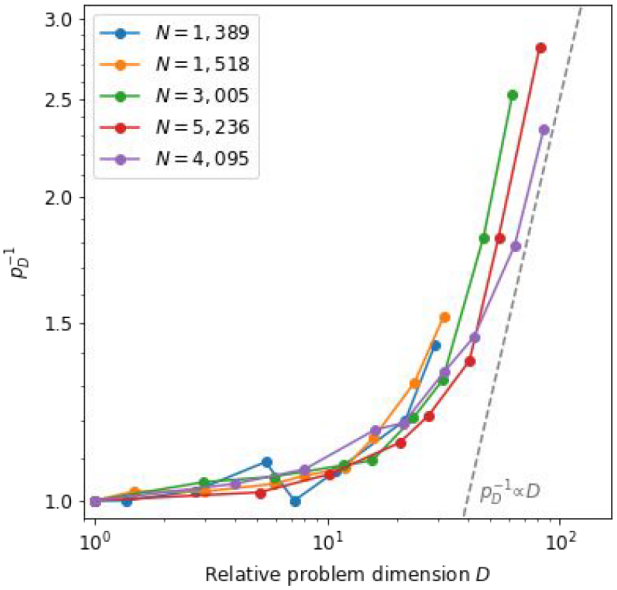


Figure 13. Plots of p_D^{-1} as a function of the relative problem dimension D for the partitioning problems of $N \leq 8192$.

Next, we discuss the worst-case time complexity. Even when assuming $t_1 = O(N)$ as the worst case, we can confirm that $t_D = O(N^2)$, which still has a lower complexity than the observed time complexity of the Boppana–Halldórsson algorithm. Moreover, when the overall partitioning procedure depicted in Figure 1 is considered, a notable detail is observed that if a sufficiently large n_{bit} is available for large N, then the partitioning procedure (3) may not be the most rate-limiting step in the overall procedure of measuring the expectation values of multiple Pauli strings (Figure 1) because the commutativity evaluation step (2) of Pauli strings (Figure 1) has a higher dimension of the time complexity of $O(N^2n)$.

We can consider several additional strategies to improve the partitioning method for future studies. For example, setting different $\mathcal{S}_{\mathcal{P}_k}$ by sorting $\mathcal{P}_k$ along different orders[11] potentially results in a more optimal $\mathcal{T}_{\mathcal{P}_k}$ than along the OpenFermion order, for obtaining C_k with more elements. Moreover, repeating the partitioning steps from (2) to (4) r times can increase the number of elements of C_k. However, the number of elements in a partition cannot exceed n_{bit} when using the Ising model-based algorithm, which would be the theoretical limit of the algorithm. In addition, enabling a larger n_{bit} than 8192 with the future development of annealing machines would ensure a shorter algorithm runtime by reducing the contributions of $1 + s$ and p_D^{-1}, whereas it potentially increases t_1 by increasing τ_1.

5. CONCLUSIONS AND FUTURE PERSPECTIVE

Herein, we propose a method for the partitioning of Pauli strings. We transfer the portioning to Ising optimization and solve it using an Ising machine. Compared with conventional algorithms (the Boppana–Halldórsson and Bron–Kerbosch algorithms), our Ising model-based algorithm shows advantages in terms of solution optimality and time complexity as the number of Pauli strings increases. Therefore, we believe the proposed method to be one of the most useful methods for solving the partitioning problem, especially when large quantum systems become available in the future. Moreover, the proposed method is versatile for any sort of quantum algorithms.

To effectively perform such simultaneous measurements, the implementation of basis-changing gate sets B (as denoted in step (4) in Figure 1) is necessary. When GC is applied for partitioning, constituting B is known to be a nontrivial problem.[14] In addition, such gate sets B include two-qubit entangling gates (e.g., CNOT), which is likely to be noisier than single-qubit gates. In this context, the reduction of the number of two-qubit gates in B is an important problem. A universal method for constituting B is available in ref 14. Another strategy is to tune the partitioning process to suppress the number of entangling gates.[13] To make this strategy compatible with the Ising model-based algorithm, we may need to tune the coefficients $c_{i,j}$ to circumvent too many entangling gates, which will be the subject of future works.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.2c06453.

Appendix text: Details of the Boppana–Halldórsson and Bron–Kerbosch algorithms, proof that $m > 1$ guarantees that the global minimum of $f(x_1, ..., x_N)$ in eq 11 is realized when the second term is zero and proof of the dependence of p_D on d is represented by $O(D^{-1})$ (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Tomochika Kurita – *Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Atsugi, Kanagawa 243-0197, Japan;* orcid.org/0000-0002-2097-9127; Email: kurita.tomo@fujitsu.com

Authors

Mikio Morita – *Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Kawasaki, Kanagawa 211-8588, Japan*

Hirotaka Oshima – *Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Atsugi, Kanagawa 243-0197, Japan*

Shintaro Sato – *Quantum Laboratory, Fujitsu Research, Fujitsu Limited, Atsugi, Kanagawa 243-0197, Japan*

https://doi.org/10.1021/acs.jpca.2c06453
J. Phys. Chem. A 2023, 127, 1068–1080

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jpca.2c06453

Author Contributions

T.K. developed the algorithm and performed the numerical calculations, analysis, and writing of the manuscript. All the members contributed to the discussions, analysis, and writing of the manuscript. H.O. and S.S. supervised the project.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Yoshinori Tomita, Toshiyuki Miyazawa, and Kazuya Takemoto for support for the utilization of the Digital Annealer and fruitful discussions.

■ REFERENCES

(1) Nielsen, M. A.; Chuang, I.*Quantum computation and quantum information*; Cambridge University Press: Cambridge, MA, U.S.A., 2010.
(2) McArdle, S.; Endo, S.; Aspuru-Guzik, A.; Benjamin, S. C.; Yuan, X. Quantum Computational Chemistry. *Rev. Mod. Phys.* **2020**, *92*, 015003.
(3) Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79.
(4) Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P. J.; Aspuru-Guzik, A.; O'Brien, J. L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213.
(5) McClean, J. R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of Variational Hybrid Quantum-Classical Algorithms. *New J. Phys.* **2016**, *18*, 023023.
(6) O'Malley, P. J. J.; Babbush, R.; Kivlichan, I. D.; Romero, J.; McClean, J. R.; Barends, R.; Kelly, J.; Roushan, P.; Tranter, A.; Ding, N.; et al. Scalable Quantum Simulation of Molecular Energies. *Phys. Rev. X* **2016**, *6*, 031007.
(7) Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J. M.; Gambetta, J. M. Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature* **2017**, *549*, 242.
(8) Paudel, H. P.; Syamlal, M.; Crawford, S. E.; Lee, Y.-L.; Shugayev, R. A.; Lu, P.; Ohodnicki, P. R.; Mollot, D.; Duan, Y. Quantum Computing and Simulations for Energy Applications: Review and Perspective. *ACS Eng. Au* **2022**, *2*, 151.
(9) Rice, J. E.; Gujarati, T. P.; Motta, M.; Takeshita, T. Y.; Lee, E.; Latone, J. A.; Garcia, J. M. Quantum Computation of Dominant Products in Lithium-Sulfur Batteries. *J. Chem. Phys.* **2021**, *154*, 134115.
(10) Jena, A.; Genin, S.; Mosca, M. Pauli Partitioning with Respect to Gate Sets. *arXiv:1907.07859 [quant-ph]* **2019**.
(11) Verteletskyi, V.; Yen, T.-C.; Izmaylov, A. F. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *J. Chem. Phys.* **2020**, *152*, 124114.
(12) Yen, T.-C.; Verteletskyi, V.; Izmaylov, A. F. Measuring All Compatible Operators in One Series of Single-Qubit Measurements Using Unitary Transformations. *J. Chem. Theory Comput.* **2020**, *16*, 2400−2409.
(13) Hamamura, I.; Imamichi, T. Efficient evaluation of quantum observables using entangled measurements. *npj Quantum Inf.* **2020**, *6*, 56.
(14) Gokhale, P.; Angiuli, O.; Ding, Y.; Gui, K.; Tomesh, T.; Suchara, M.; Martonosi, M.; Chong, F. T. $O(N^3)$ Measurement Cost for Variational Quantum Eigensolver on Molecular Hamiltonians. *IEEE Trans. Quantum Eng.* **2020**, *1*, 1−24.
(15) Huang, H.-Y.; Kueng, R.; Preskill, J. Predicting Many Properties of a Quantum System from Very Few Measurements. *Nat. Phys.* **2020**, *16*, 1050−1057.
(16) Huang, H.-Y.; Kueng, R.; Preskill, J. Efficient Estimation of Pauli Observables by Derandomization. *Phys. Rev. Lett.* **2021**, *127*, 030503.
(17) Yen, T.-C.; Ganeshram, A.; Izmaylov, A.-F Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates. *arXiv:2201.01471 [quant-ph]* **2022**.
(18) Kübler, J. M.; Arrasmith, A.; Cincio, L.; Coles, P. J. An Adaptive Optimizer for Measurement-Frugal Variational Algorithms. *Quantum* **2020**, *4*, 263.
(19) Gui, K.; Tomesh, T.; Gokhale, P.; Shi, Y.; Chong, F. T.; Marnotosi, M.; Suchara, M. Term Grouping and Travelling Salesperson for Digital Quantum Simulation. *arXiv:2001.05983 [quant-ph]* **2021**.
(20) Izmaylov, A.-F.; Yen, T.-C.; Lang, R. A.; Verteletskyi, V. Unitary Partitioning Approach to the Measurement Problem in the Variational Quantum Eigensolver Method. *J. Chem. Theory Comput.* **2020**, *16*, 190−195.
(21) Östergård, P. R. J. A fast algorithm for the maximum clique problem. *Discrete Appl. Math.* **2002**, *120*, 197−207.
(22) Boppana, R.; Halldórsson, M. Approximating Maximum Independent Sets by Excluding Subgraphs. *BIT Numerical Math.* **1992**, *32*, 180−196.
(23) Bron, C.; Kerbosch, J. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* **1973**, *16*, 575−577.
(24) Tomita, E.; Tanaka, A.; Takahashi, H. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Comput. Sci.* **2006**, *363*, 28−42.
(25) Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2014**, *2*, 5.
(26) Matsubara, S.; Takatsu, M.; Miyazawa, T.; Shibasaki, T.; Watanabe, Y.; Takemoto, K.; Tamura, H.Digital Annealer for High-Speed Solving of Combinatorial optimization Problems and Its Applications. *25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Beijing, China, January 13−16, 2020; ASP-DAC, 2020; pp 667−672. DOI: 10.1109/ASP-DAC47756.2020.904510.
(27) Lawrence, J.; Brukner, C.; Zeilinger, A. Mutually unbiased binary observable sets on *N* qubits. *Phys. Rev. A* **2002**, *65*, 032320.
(28) Johnson, M. W.; Amin, M. H. S.; Gildert, S.; Lanting, T.; Hamze, F.; Dickson, N.; Harris, R.; Berkley, A. J.; Johansson, J.; Bunyk, P.; et al. Quantum annealing with manufactured spins. *Nature* **2011**, *473*, 194−198.
(29) Yamaoka, M.; Yoshimura, C.; Hayashi, M.; Okuyama, T.; Aoki, H.; Mizuno, H. A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing. *IEEE J. Solid-State Circuits* **2016**, *51* (1), 303−309.
(30) NetworkX Home Page. https://networkx.org/ (accessed 03-03-2022).
(31) Aramon, M.; Rosenberg, G.; Valiante, E.; Miyazawa, T.; Tamura, H.; Katzgraber, H. G. Physics-inspired optimization for quadratic unconstrained problems using a Digital Annealer. *Front. Phys.* **2019**, *7*, 48.
(32) OpenFermion Home Page. https://quantumai.google/openfermion/ (accessed 03-03-2022).

■ NOTE ADDED AFTER ASAP PUBLICATION

This paper was published ASAP on January 18, 2023, with data missing from Table 2. The corrected version was reposted on February 2, 2023.