Results in Control and Optimization 11 (2023) 100222

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

HUBO formulations for solving the eigenvalue problem

Kyungtaek Jun [a,*], Hyunju Lee [b]

[a] German Engineering Research and Development Center, LSTME Busan Branch, Busan, 46742, Republic of Korea
[b] Department of Radiology, Severance Hospital, Research Institute of Radiological Science, Yonsei University College of Medicine, Seoul, 03722, Republic of Korea

ARTICLE INFO

Keywords:
HUBO
Eigenvalue
Eigenvector
Quantum annealing
Quantum computing

ABSTRACT

Solving the eigenvalue problem is particularly important in almost all fields of science and engineering. With the development of quantum computers, multiple algorithms have been proposed for this purpose. However, such methods are usually only applicable to matrices of specific types, such as unitary or Hermitian matrices. The quantum annealer of the D-Wave, a quantum computer, returns the minimum value of the quadratic unconstrained binary optimization (QUBO) model. Thus, quantum annealers can be leveraged to solve arbitrary eigenvalue problems by formulating corresponding QUBO models. In this paper, we propose two higher-order unconstrained optimization (HUBO) formulations to solve eigenvalue problems involving $n \times n$ general matrices. In addition, we use a formula to reduce the order and convert the HUBO model into a QUBO model. Further, by using a quantum approximate optimization algorithm, this method can be extended to a gate-model quantum computer.

1. Introduction

Solving the eigenvalue problem is crucial to numerical analysis. Solutions for these problems play prominent roles in engineering, physics, chemistry, computer science, and economics. In classic computers, eigenvalues and eigenvectors are primarily used for singular value decomposition (SVD), pseudo-inverse calculation, and principal component analysis (PCA). Several studies have attempted to implement classical quantum algorithms to solve general eigenvalue problems. For example, quantum solution methods have been proposed to solve the wave equation [1], boundary-value problems [2], and linear initial-value differential equations [3]. In addition, a quantum phase estimation (QPE) method was proposed to identify the eigenvalues of a unitary matrix. QPE is based on Shor's algorithm [4], which can perform prime factorization in polynomial time. Likewise, several studies have demonstrated that quantum computing is incomparably faster than classical computing in certain fields [3,5–7]. Research on quantum algorithms has progressed naturally, even in the context of solving eigenvalue problems. Nevertheless, these algorithms are only applicable to eigenvalue problems under certain conditions. For example, the eigenvalues of unitary [8], Hermitian [5,9], and diagonalizable matrices [7] can be estimated using such methods. Several studies have yielded notable results regarding the identification of eigenvalues and eigenvectors of symmetric matrices. In [10,11], the ground state change problem was mapped to the Ising and quadratic unconstrained binary optimization (QUBO) problems, respectively. The authors also proposed a quantum annealing eigensolver for complex symmetric matrices using constrained real values in QUBOs. In [12], an algorithm was proposed that minimizes the corresponding Rayleigh quotient via iterative descent and computes the eigenpairs of a symmetric matrix. Moreover, multiple studies have attempted to compute the eigenvalues of non-Hermitian and non-unitary matrices [13,14]. However, to the best of our knowledge, there does not exist a method for computing the eigenvalues of a general matrix using a quantum computer.

* Corresponding author.
E-mail address: kyungtaek.jun@lstme.org (K. Jun).

https://doi.org/10.1016/j.rico.2023.100222
Received 14 April 2022; Received in revised form 30 November 2022; Accepted 5 March 2023
Available online 22 March 2023

The quantum computer, D-Wave quantum annealer, is capable of solving combinatorial optimization problems by identifying solutions that minimize the following cost (energy) function.

$$f(\vec{q}) = \vec{q}^T Q \vec{q} \tag{1}$$

where Q denotes an upper matrix, $\vec{q} = (q_1, \dots, q_N)^T$, and q_i denote binary variables. In the case of $q_i \in \{-1, 1\}$, the aforementioned equation represents the so-called Ising model; and in the case of $q_i \in \{0, 1\}$, represents the QUBO model. As the Ising and QUBO models are convertible into each other, we focus solely on the QUBO model. Because $q_i^2 = q_i$, the cost function can be reformulated as follows:

$$f(\vec{q}) = \sum_{i=1}^{N} Q_{i,i} q_i + \sum_{i<j}^{N} Q_{i,j} q_i q_j, \tag{2}$$

where $Q_{i,i}$ denotes diagonal terms and $Q_{i,j}$ denotes off-diagonal terms. In principle, if we can formulate the objective function as a QUBO model, its minimum value can be obtained and a solution to the objective function can be identified.

In this study, we propose two higher-order unconstrained optimization (HUBO) models that can be converted into QUBO models. Both HUBO models are based on the following approach. First, a linear least-squares problem of the eigenvalue problem is used. Then, the binary representation of the solutions is set and it is input into the linear least-squares problem. Then, terms higher than the cubic expression are generated, and the model including these terms is called the HUBO model. Because the D-Wave quantum annealer can only solve QUBO or Ising models with linear and quadratic terms, its degree is reduced to be less than that of a quadratic expression using Eq. (15). Finally, the two representation methods for the solutions are presented and two HUBO models are created to solve the eigenvalue problem. Additionally, the Python code to create HUBO models that can be used in the D-Wave simulator is provided.

2. Methods

2.1. The linear least-squares problem

Given a matrix $A \in \mathbb{R}^{n\times n}$, a column vector of variables, $\vec{x} \in \mathbb{R}^n$, and a real number $\lambda \in \mathbb{R}$, the linear least-squares problem is to find λ and $\vec{x}$ satisfying $A\vec{x} = \lambda\vec{x}$. Thus, it can be formulated as follows:

$$\arg\min_{\lambda,\vec{x}} \|A\vec{x} - \lambda\vec{x}\| = 0 \tag{3}$$

To solve Eq. (3), let us begin by writing out $A\vec{x} - \lambda\vec{x}$:

$$A\vec{x} - \lambda\vec{x} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} - \lambda \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{4}$$

Taking the 2-norm square of the resultant vector in Eq. (4), we obtain the following:

$$\|A\vec{x} - \lambda\vec{x}\|_2^2 = \vec{x}^T A^T A\vec{x} - \lambda\vec{x}^T A\vec{x} - \lambda\vec{x}^T A^T \vec{x} + \lambda^2 \vec{x}^T \vec{x} \tag{5}$$

$$= \vec{x}^T A^T A\vec{x} - 2\lambda\vec{x}^T A\vec{x} + \lambda^2 \vec{x}^T \vec{x} \tag{6}$$

Because $\vec{x}^T A\vec{x}$ is scalar and $\vec{x}^T A\vec{x} = (\vec{x}^T A\vec{x})^T = \vec{x}^T A^T (\vec{x}^T)^T = \vec{x}^T A^T \vec{x}$, Eq. (5) is represented by Eq. (6).

2.2. HUBO model 1

While solving the binary least-squares problem, each x_i is represented by a combination of qubits $q_{i,l} \in \{0, 1\}$. As per the work by O'Malley and Vesselinov [15], the radix 2 representation of the positive real value, λ, is given by

$$\lambda \approx \sum_{l=-m}^{m} 2^l q_l^+ - \sum_{l=-m}^{m} 2^l q_l^- \tag{7}$$

where the positive integer, l, denotes the number of digits of λ and the negative integer, l, denotes the number of digits of fractional terms. We can now represent a large real value, x_i, as follows:

$$x_i \approx \sum_{l=0}^{m} 2^l q_{i,l}^+ - \sum_{l=0}^{m} 2^l q_{i,l}^-. \tag{8}$$

Both positive and negative numbers are represented using $q_{i,l}^+$ and $q_{i,l}^-$. Note that this representation can take the same value corresponding to different binary combinations. Since the eigenvector appears in the form of a straight line passing through the origin, we can express it in the form of an integer in which the minimum value appears in quantum annealers, as given by Eq. (8).

To derive a HUBO model, we insert Eqs. (7) and (8) into Eq. (6). This yields the summation terms of the first term in Eq. (6), as indicated below:

$$\vec{x}^T A^T A\vec{x} = \sum_{k=1}^{n} \left\{ \sum_{i=1}^{n} (a_{k,i}x_i)^2 + 2\sum_{i<j} a_{k,i}a_{k,j}x_ix_j \right\} \tag{9}$$

Eq. (9) can be calculated as follows:

$$\sum_{k=1}^{n}\sum_{i=1}^{n}(a_{k,i}x_i)^2 \approx \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{l=0}^{m} a_{k,i}^2 2^{2l}\left(q_{i,l}^+ + q_{i,l}^-\right) + \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{l_1<l_2} a_{k,i}^2 2^{l_1+l_2+1}\left(q_{i,l_1}^+ q_{i,l_2}^+ + q_{i,l_1}^- q_{i,l_2}^-\right) \tag{10}$$

$$\sum_{k=1}^{n}\sum_{i<j} 2a_{k,i}a_{k,j}x_ix_j \approx \sum_{k=1}^{n}\sum_{i<j}\sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{l_1+l_2+1}a_{k,i}a_{k,j}\left(q_{i,l_1}^+ q_{j,l_2}^+ + q_{i,l_1}^- q_{j,l_2}^- - q_{i,l_1}^+ q_{j,l_2}^- - q_{i,l_1}^- q_{j,l_2}^+\right) \tag{11}$$

In Eq. (10), the first summation represents linear terms and the second summation represents quadratic terms. Moreover, Eq. (11) is part of the quadratic terms in the HUBO model.

The HUBO form of the second term in Eq. (6) can be obtained as follows:

$$-2\lambda\vec{x}^T A\vec{x} = -2\lambda\sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}x_ix_j \tag{12}$$

$$\begin{aligned} &\approx \textstyle\sum_{k=1}^{n}\sum_{l=-m}^{m}\sum_{l_1=0}^{m} 2^{l+2l_1+1}a_{k,k}\left(q_l^- q_{k,l_1}^+ + q_l^- q_{k,l_1}^- - q_l^+ q_{k,l_1}^+ - q_l^+ q_{k,l_1}^-\right) \\ &+ \textstyle\sum_{k=1}^{n}\sum_{l=-m}^{m}\sum_{0\le l_1<l_2\le m} 2^{l+l_1+l_2+2}a_{k,k}\left(q_l^- q_{k,l_1}^+ q_{k,l_2}^+ + q_l^- q_{k,l_1}^- q_{k,l_2}^- - q_l^+ q_{k,l_1}^+ q_{k,l_2}^+ - q_l^+ q_{k,l_1}^- q_{k,l_2}^-\right) \\ &+ \textstyle\sum_{\substack{1\le k,i\le n,\\ k\ne i}}\sum_{l=-m}^{m}\sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{l+l_1+l_2+1}a_{k,i}\Big(q_l^+ q_{i,l_1}^+ q_{k,l_2}^- + q_l^+ q_{i,l_1}^- q_{k,l_2}^+ - q_l^+ q_{i,l_1}^+ q_{k,l_2}^+ - q_l^+ q_{i,l_1}^- q_{k,l_2}^- \\ &- q_l^- q_{i,l_1}^+ q_{k,l_2}^- - q_l^- q_{i,l_1}^- q_{k,l_2}^+ + q_l^- q_{i,l_1}^+ q_{k,l_2}^+ + q_l^- q_{i,l_1}^- q_{k,l_2}^-\Big) \end{aligned} \tag{13}$$

The expansion of Eq. (12) contains part of the quadratic and cubic terms when $k = i$, and comprises solely part of the cubic terms when $k \ne i$ in our HUBO model.

The HUBO form of the third term in Eq. (6) can be obtained as follows:

$$\begin{aligned} \lambda^2\vec{x}^T\vec{x} \approx \sum_{k=1}^{n}\Bigg\{ &\sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{2l_1+2l_2}\left(q_{l_1}^+ q_{k,l_2}^+ + q_{l_1}^+ q_{k,l_2}^- + q_{l_1}^- q_{k,l_2}^+ + q_{l_1}^- q_{k,l_2}^-\right) \\ &+ \sum_{l_1=-m}^{m}\sum_{0\le l_2<l_3\le m} 2^{2l_1+l_2+l_3+1}\left(q_{l_1}^+ q_{k,l_2}^+ q_{k,l_3}^+ + q_{l_1}^+ q_{k,l_2}^- q_{k,l_3}^- + q_{l_1}^- q_{k,l_2}^+ q_{k,l_3}^+ + q_{l_1}^- q_{k,l_2}^- q_{k,l_3}^-\right) \\ &+ \sum_{l_1=0}^{m}\sum_{-m\le l_2<l_3\le m} 2^{2l_1+l_2+l_3+1}\left(q_{k,l_1}^+ q_{l_2}^+ q_{l_3}^+ + q_{k,l_1}^+ q_{l_2}^- q_{l_3}^- + q_{k,l_1}^- q_{l_2}^+ q_{l_3}^+ + q_{k,l_1}^- q_{l_2}^- q_{l_3}^-\right) \\ &+ \sum_{-m\le l_1<l_2\le m}\sum_{0\le l_3<l_4\le m} 2^{l_1+l_2+l_3+l_4+2}\left(q_{l_1}^+ q_{l_2}^+ q_{k,l_3}^+ q_{k,l_4}^+ + q_{l_1}^+ q_{l_2}^+ q_{k,l_3}^- q_{k,l_4}^- + q_{l_1}^- q_{l_2}^- q_{k,l_3}^+ q_{k,l_4}^+ + q_{l_1}^- q_{l_2}^- q_{k,l_3}^- q_{k,l_4}^-\right)\Bigg\} \end{aligned} \tag{14}$$

The above equation consists of quadratic, cubic, and quartic terms. To reformulate a non-quadratic (higher-degree) polynomial into Ising/QUBO form, terms of the form, $axyz$, where a is a real number, are substituted with one of the following quadratic terms [16]:

$$axyz = \begin{cases} aw(x+y+z-2), & a<0 \\ a\{w(x+y+z-1)+(xy+yz+zx)-(x+y+z)+1\}, & a>0 \end{cases} \tag{15}$$

For all $x, y, z \in \{0, 1\}$, $axyz$ can be transformed into a combination of linear and quadratic terms by adding a new qubit w to every cubic term. Similarly, Eq. (15) can be applied twice to convert quartic terms into QUBO formulations. Eq. (6) takes the minimum value, 0, when the variables, scalar λ, and vector $\vec{x}$, satisfy the eigenvalue and eigenvector in the HUBO form. Therefore, the first HUBO model for the eigenvalues and eigenvectors is obtained by summing Eqs. (10), (11), (13), and (14), and the first QUBO model is obtained via polynomial reduction using the minimum selection given by Eq. (15).

2.3. HUBO model 2

To reduce the number of qubits used in Eqs. (7) and (8), the following new approximations of λ and x_i are introduced:

$$\lambda \approx -2^{m+1}q^- + \sum_{l=-m}^{m} 2^l q_l^+ \tag{16}$$

$$x_i \approx -2^{m+1}q_i^- + \sum_{l=0}^{m} 2^l q_{i,l}^+ \tag{17}$$

To derive the HUBO model, we insert Eqs. (16) and (17) into Eq. (6). Using the calculation rules utilized previously, we obtain the right-hand side term in Eq. (6). The first term in Eq. (6) can be obtained by adding the following two equations:

$$\sum_{k=1}^{n}\sum_{i=1}^{n}\left(a_{k,i}x_i\right)^2 \approx \sum_{k=1}^{n}\sum_{i=1}^{n} a_{k,i}^2 \left\{ 2^{2m+2}q_i^- + \sum_{l=0}^{m}\left(2^{2l} - 2^{l+m+2}q_i^-\right) q_{i,l}^+ + \sum_{0\le l_1<l_2\le m} 2^{l_1+l_2+1} q_{i,l_1}^+ q_{i,l_2}^+ \right\} \tag{18}$$

$$\sum_{k=1}^{n}\sum_{i<j} a_{k,i}a_{k,j}x_i x_j \approx \sum_{k=1}^{n}\sum_{i<j} a_{k,i}a_{k,j} \left\{ 2^{2m+3}q_i^- q_j^- - \sum_{l=0}^{m} 2^{l+m+2}\left(q_i^- q_{j,l}^+ + q_j^- q_{i,l}^+\right) + \sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{l_1+l_2+1} q_{i,l_1}^+ q_{j,l_2}^+ \right\} \tag{19}$$

The HUBO form of the second term in Eq. (6) can be obtained as follows:

$$-2\lambda \vec{x}^T A\vec{x} = -2\lambda \sum_{k=1}^{n}\sum_{i=1}^{n} a_{k,\mathrm{i}} x_i x_k \tag{20}$$

$$\begin{aligned}
&\approx \textstyle\sum_{k=1}^{n} a_{k,k}\left(2^{3m+4}q^- q_k^- + \sum_{l_1=0}^{m} 2^{2l_1+m+2} q^- q_{k,l_1}^+ - \sum_{l=-m}^{m} 2^{l+2m+3} q_l^+ q_k^- - \sum_{l=-m}^{m}\sum_{l_1=0}^{m} 2^{l+2l_1+1} q_l^+ q_{k,l_1}^+\right)\\
&+ \textstyle\sum_{k=1}^{n} a_{k,k}\Big(-\sum_{l_1=0}^{m} 2^{l_1+2m+4} q^- q_k^- q_{k,l_1}^+ + \sum_{l=-m}^{m}\sum_{l_1=0}^{m} 2^{l+l_1+m+3} q_l^+ q_k^- q_{k,l_1}^+ + \sum_{0\le l_1<l_2\le m} 2^{l_1+l_2+m+2} q^- q_{k,l_1}^+ q_{k,l_2}^+\\
&- \textstyle\sum_{l=-m}^{m}\sum_{0\le l_1<l_2\le m} 2^{l+l_1+l_2+1} q_l^+ q_{k,l_1}^+ q_{k,l_2}^+\Big)\\
&+ \textstyle\sum_{1\le k,i\le n,k\ne i} a_{k,i}\Big[\Big\{2^{3m+4}q^- q_i^- q_k^- - \sum_{l=0}^{m} 2^{l+2m+3}\left(q^- q_i^- q_{k,l}^+ + q^- q_k^- q_{i,l}^+\right) + \sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{l_1+l_2+m+2} q^- q_{i,l_1}^+ q_{k,l_2}^+\Big\}\\
&+ \textstyle\sum_{l_3=-\mathrm{m}}^{m}\Big\{-2^{l_3+2m+3} q_{l_3}^+ q_i^- q_k^- + \sum_{l=0}^{m} 2^{l+m+l_3+2}\left(q_i^- q_{l_3}^+ q_{k,l}^+ + q_k^- q_{l_3}^+ q_{i,l}^+\right) - \sum_{l_1=0}^{m}\sum_{l_2=0}^{m} 2^{l_1+l_2+l_3+1} q_{l_3}^+ q_{i,l_1}^+ q_{k,l_2}^+\Big\}\Big]
\end{aligned} \tag{21}$$

The HUBO form of the third term in Eq. (6) can be obtained as follows:

$$\begin{aligned}
\lambda^2 \vec{x}^T\vec{x} \approx &\sum_{k=1}^{n}\left(2^{4m+4}q^- q_k^- + \sum_{l=-m}^{m} 2^{2l+2m+2} q_l^+ q_k^- + \sum_{l=0}^{m} 2^{2l+2m+2} q^- q_{k,l}^+ + \sum_{l=-m}^{m}\sum_{l_3=0}^{m} 2^{2l+2l_3} q_l^+ q_{k,l_3}^+\right)\\
&+ \sum_{k=1}^{n}\left[\sum_{-m\le l_1<l_2\le m}\left(2^{l_1+l_2+2m+3} q_k^- q_{l_1}^+ q_{l_2}^+ + \sum_{l=0}^{m} 2^{2l+l_1+l_2+1} q_{k,l}^+ q_{l_1}^+ q_{l_2}^+\right) - \sum_{l=-m}^{m} 2^{l+3m+4} q_k^- q^- q_l^+\right]\\
&+ \sum_{k=1}^{n}\left[\sum_{0\le l_1<l_2\le m}\left(2^{l_1+l_2+2m+3} q^- q_{k,l_1}^+ q_{k,l_2}^+ + \sum_{l=-m}^{m} 2^{2l+l_1+l_2+1} q_l^+ q_{k,l_1}^+ q_{k,l_2}^+\right) - \sum_{l=0}^{m} 2^{l+3m+4} q_k^- q^- q_{k,l}^+\right]\\
&- \sum_{k=1}^{n}\sum_{l=-m}^{m}\sum_{l_3=0}^{m}\left(2^{l+2l_3+m+2} q_{k,l_3}^+ q^- q_l^+ + 2^{2l+l_3+m+2} q_l^+ q_k^- q_{k,l_3}^+ - 2^{l+l_3+2m+4} q^- q_l^+ q_k^- q_{k,l_3}^+\right)\\
&- \sum_{k=1}^{n}\left[\sum_{-m\le l_1<l_2\le m}\left(\sum_{l=0}^{m} 2^{l+l_1+l_2+m+3} q_k^- q_{k,l}^+ q_{l_1}^+ q_{l_2}^+\right) + \sum_{0\le l_1<l_2\le m}\left(\sum_{l=-m}^{m} 2^{l+l_1+l_2+m+3} q^- q_l^+ q_{k,l_1}^+ q_{k,l_2}^+\right)\right.\\
&\left. + \sum_{-m\le l_1<l_2\le m}\left(\sum_{0\le l_4<l_5\le m} 2^{l_1+l_2+l_4+l_5+2} q_{l_1}^+ q_{l_2}^+ q_{k,l_4}^+ q_{k,l_5}^+\right)\right]
\end{aligned} \tag{22}$$

Therefore, our second HUBO model for the eigenvalues and eigenvectors is obtained by summing Eqs. (18), (19), (21), and (22), and our second QUBO model can be obtained via polynomial reduction using the minimum selection given by Eq. (15).

2.4. HUBO model for complex variables

To solve the eigenvalue problem when λ and x_j are complex numbers, the following parameters are used:

$$\lambda \approx \sum_{l_1=-m}^{m} 2^{l_1} q_{l_1}^+ - 2^{m+1} q_1^- + i\left(\sum_{l_2=-m}^{m} 2^{l_2} q_{l_2}^+ - 2^{m+1} q_2^-\right) \tag{23}$$

$$x_j \approx \sum_{l_1=-m}^{m} 2^{l_1} q_{j,l_1}^+ - 2^{m+1} q_{j,1}^- + i\left(\sum_{l_2=-m}^{m} 2^{l_2} q_{j,l_2}^+ - 2^{m+1} q_{j,2}^-\right) \tag{24}$$

Complex numbers comprise real and imaginary parts. By substituting Eqs. (23) and (24) into Eq. (6), the QUBO formulations corresponding to the real and imaginary parts, respectively, are obtained. The final QUBO model can then be obtained by adding the two QUBO formulations. However, when an imaginary number is used, twice as many qubits are required compared to the case of a real number—therefore, qubit variables of as small a size as possible should be used.

3. Implementation and results

This section outlines the implementation process and presents the simulator results corresponding to certain examples.

3.1. Steps of the implementation process

The following list describes the process of determining variables that satisfy $A\vec{x} = \lambda\vec{x}$ for a 2×2 matrix A with $x_i, \lambda \in \{-3, -2, \dots, 3\}$.

(1) The number of qubits and the combination form of the qubits are set to represent one variable. Because all variables, x_1, x_2, and λ, take integral values between −3 and 3, each variable can be represented using two positive and negative qubits, as follows:

$$A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Longrightarrow A \begin{bmatrix} q_1 + 2q_2 - q_3 - 2q_4 \\ q_5 + 2q_6 - q_7 - 2q_8 \end{bmatrix} = (q_9 + 2q_{10} - q_{11} - 2q_{12}) \begin{bmatrix} q_1 + 2q_2 - q_3 - 2q_4 \\ q_5 + 2q_6 - q_7 - 2q_8 \end{bmatrix} \tag{25}$$

(2) $\|A\vec{x} - \lambda\vec{x}\|_2^2$ is calculated and the QUBO matrix, QM, satisfying the following condition is identified.

$$\|A\vec{x} - \lambda\vec{x}\|_2^2 = \begin{bmatrix} q_1 q_2 \cdots q_{12} \end{bmatrix} QM \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{12} \end{bmatrix} \tag{26}$$

(3) The global minimum energy of the QUBO matrix, QM, is estimated using a D-Wave's qpu solver.
(4) The solution pairs $(q_1, q_2, \dots, q_{12})$ obtained from the simulator are converted into the variables x_1, x_2, and λ.

The code for calculating the QUBO matrix is provided in Appendix A, and the pseudocode can be easily obtained from our code.

3.2. Results

This section presents the eigenvalues and eigenvectors of five matrices calculated using D-Wave's qpu solver by following the proposed method. Table 1 presents a comparison between the mathematically calculated results and those obtained using the D-Wave's qpu solver. The "mathematical results" column presents the mathematically calculated eigenvalues and eigenvectors of a given matrix. Following the aforementioned steps, we determine the number of qubits used to represent one variable. Each number of logical qubits and ranges are presented in the column "Used qubit number and range". If we set the qubit number used for a variable to be 4 and x_1 is expressed as $q_1 + 2q_2 - q_3 - 2q_4$, the range of x_1 is $[-3, 3]$. Therefore, this variable expression method yields only the eigenvalue λ and elements of the eigenvector $\vec{x}$ lying in the range $[-3, 3]$. By increasing the number of qubits, the range can be expanded. For instance, if x_1 is expressed as $q_1 + 2q_2 + 4q_3 - q_4 - 2q_5 - 4q_6$, x_1 is a value in the range $[-7, 7]$. For this reason, considering higher numbers of qubits yields larger eigenvalues and eigenvectors using the D-Wave's qpu solver. In this study, five HUBO models are proposed to solve eigenvalue problems. Different form of polynomial reduction is performed on the cubic and quartic terms of Eqs. (13), (14), (21), and (22) depending on the sign of the coefficients [17]. Because the eigenvalues and $\vec{x}_{\bar{i}}$ may be positive, negative, or 0, we can assume that q^+q^- of each variable is zero in Eqs. (7) and (8), respectively. The eigenvalue and eigenvector of the actual matrix can be obtained based on the qubit variable that evaluates the minimum value of the HUBO model to 0.

4. Discussion

Fifteen couplers are used for each qubit of D-Wave's Advantage, while the number is only two or three for IBM Quantum. Thus, the D-Wave system is selected to test the proposed QUBO model for the solution of the eigenvalue problem. D-Wave's Advantage quantum annealer involves over 5000 qubits and over 35,000 couplers; thus, up to 180 logical qubits can be used. In turn, the size of the QUBO matrix can be as large as 180×180. The size of the QUBO matrix is expressed as the product of the representable range for the variables, x_i and λ, and the size of matrix, A, as given by Eq. (26). For example, if x_i is expressed at a 30-bit level using qubits, the corresponding maximum size of a computable matrix is 5×5. This problem originates from the lack of couplers because connectivity is required between all qubits when all coefficients of the QUBO matrix, which is an upper triangular matrix, are not 0. Therefore, solution of the eigenvalue problem for large matrices remains difficult even with quantum annealers.

The proposed algorithm estimates the number of cases where the variables for $A\vec{x} = \lambda\vec{x}$ can be expressed in qubits. However, the actual obtainment of these useless cases does not matter. The QUBO model for the eigenvalue problem includes the case in which all x_i and λ are zero. Because quantum annealers take several microseconds to calculate the optimization problem once, the total number of cases can be obtained via multiple annealing. However, when the number of logical qubits used exceeds 100, identification of the optimization solution becomes difficult, even after 500,000 shots. In particular, we focus on D-Wave's hybrid solver, which can utilize one million logical qubits. It can identify an exact solution in just a single annealing step, even thousands of QUBO matrices are required to be calculated. As the constraint conditions can be set by the user, we can eliminate useless solutions in which all variables are zero. A hybrid solver uses a combination of a quantum annealer and classical computer to determine the minimum for a given QUBO matrix. Computation using a hybrid solver takes longer than that using a quantum annealer alone; however, the hybrid solver provided by the D-Wave system is sufficiently accurate to be used with general matrices. Quantum annealers and classical computers can be used together to solve eigenvalue problems for large matrices.

Table 1
Comparison between eigenvalues and eigenvectors obtained using the D-Wave's qpu solver and those obtained via mathematical calculation.

Matrix	Mathematical result: Eigenvalue	Mathematical result: Eigenvector	Used qubit number and range	D-Wave's qpu solver result (x_1, x_2, λ) or (x_1, x_2, x_3, λ)
$A = \begin{bmatrix} 1 & 0 \\ -4 & 3 \end{bmatrix}$	$\lambda_1 = 1$	$x = \begin{bmatrix} t \\ 2t \end{bmatrix}$ $(t \in R)$	4 qubits	$(1, 2, 1), (-1, -2, 1)$
	$\lambda_2 = 3$	$x = \begin{bmatrix} 0 \\ s \end{bmatrix}$ $(s \in R)$	$[-3, 3]$	$(0, \pm 1, 3), (0, \pm 2, 3), (0, \pm 3, 3)$
$A = \begin{bmatrix} 1 & -2 \\ 1 & 4 \end{bmatrix}$	$\lambda_1 = 2$	$x = \begin{bmatrix} -2t \\ t \end{bmatrix}$ $(t \in R)$	4 qubits	$(2, -1, 2), (-2, 1, 2)$
	$\lambda_2 = 3$	$x = \begin{bmatrix} -s \\ s \end{bmatrix}$ $(s \in R)$	$[-3, 3]$	$(\pm 1, \mp 1, 3), (\pm 2, \mp 2, 3), (\pm 3, \mp 3, 3)$
$A = \begin{bmatrix} -8 & -5 \\ 5 & 2 \end{bmatrix}$	$\lambda_1 = \lambda_2 = -3$	$x = \begin{bmatrix} -t \\ t \end{bmatrix}$ $(t \in R)$	6 qubits $[-7, 7]$	$(\pm k, \mp k, 3)$ for $k = \{1, 2, \dots, 7\}$
$A = \begin{bmatrix} 5 & 2 & 0 \\ 2 & 5 & 0 \\ -3 & 4 & 6 \end{bmatrix}$	$\lambda_1 = 3$	$x = \begin{bmatrix} 3t \\ -3t \\ 7t \end{bmatrix}$ $(t \in R)$	6 qubits	$(3, -3, 7, 3), (-3, 3, -7, 3)$
	$\lambda_2 = 6$	$x = \begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix}$ $(s \in R)$	$[-7, 7]$	$(0, 0, \pm k, 6)$, $(\pm k, \pm k, \pm k, 7)$
	$\lambda_3 = 7$	$x = \begin{bmatrix} w \\ w \\ w \end{bmatrix}$ $(w \in R)$		for $k = \{1, 2, \dots, 7\}$
$A = \begin{bmatrix} 1 & 3 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$	$\lambda_1 = \lambda_2 = -2$	$x = \begin{bmatrix} t \\ -t \\ 0 \end{bmatrix}$ $(t \in R)$	6 qubits	$(\pm k, \mp k, 0, -2)$, $(0, 0, \pm l, -2)$
	$\lambda_1 = \lambda_2 = -2$	$x = \begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix}$ $(s \in R)$	$[-7, 7]$	$(\pm k, \pm k, l, -2)$, $(\pm k, \pm k, -l, -2)$
	$\lambda_3 = 4$	$x = \begin{bmatrix} w \\ w \\ 0 \end{bmatrix}$ $(w \in R)$		$(\pm k, \pm k, 0, 4)$ for $k, l = \{1, 2, \dots, 7\}$

CRediT authorship contribution statement

Kyungtaek Jun: Conceived and designed the research problem, Coded for simulations using Python Numpy, Solved theoretical solution, Performed the hardware testing, Developed theoretical results, Writing – original draft. **Hyunju Lee:** Solved theoretical solution, Developed theoretical results, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research was supported by the quantum computing technology development program of the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT (MSIT)) (No. 2020M3H3A111036513)

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.rico.2023.100222.

References

[1] Costa PC, Jordan S, Ostrander A. Quantum algorithm for simulating the wave equation. Phys Rev A 2019;99(1):012323.
[2] Montanaro A, Pallister S. Quantum algorithms and the finite element method. Phys Rev A 2016;93(3):032324.
[3] Berry DW. High-order quantum algorithm for solving linear differential equations. J Phys A 2014;47(10):105301.
[4] Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. IEEE; 1994, p. 124–34.
[5] Abrams DS, Lloyd S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. Phys Rev Lett 1999;83(24):5162.
[6] Berry DW, Childs AM, Ostrander A, Wang G. Quantum algorithm for linear differential equations with exponentially improved dependence on precision. Comm Math Phys 2017;356(3):1057–81.
[7] Shao C. Computing eigenvalues of diagonalizable matrices in a quantum computer. 2019, arXiv preprint arXiv:1912.08015.
[8] Kitaev AY. Quantum measurements and the abelian stabilizer problem. 1995, arXiv preprint quant-ph/9511026.
[9] Parker JB, Joseph I. Quantum phase estimation for a class of generalized eigenvalue problems. Phys Rev A 2020;102(2):022422.
[10] Teplukhin A, Kendrick BK, Babikov D. Calculation of molecular vibrational spectra on a quantum annealer. J Chem Theory Comput 2019;15(8):4555–63.
[11] Teplukhin A, Kendrick BK, Babikov D. Solving complex eigenvalue problems on a quantum annealer with applications to quantum scattering resonances. Phys Chem Chem Phys 2020;22(45):26136–44.
[12] Krakoff B, Mniszewski SM, Negre CF. Controlled precision QUBO-based algorithm to compute eigenvectors of symmetric matrices. PLoS One 2022;17(5):e0267954.
[13] Daskin A, Grama A, Kais S. A universal quantum circuit scheme for finding complex eigenvalues. Quantum Inf Process 2014;13(2):333–53.
[14] Wang H, Wu L-A, Liu Y-x, Nori F. Measurement-based quantum phase estimation algorithm for finding eigenvalues of non-unitary matrices. Phys Rev A 2010;82(6):062303.
[15] O'Malley D, Vesselinov VV. ToQ. jl: A high-level programming language for d-wave machines based on Julia. In: 2016 IEEE high performance extreme computing conference. IEEE; 2016, p. 1–7.
[16] Jiang S, Britt KA, McCaskey AJ, Humble TS, Kais S. Quantum annealing for prime factorization. Sci Rep 2018;8(1):1–9.
[17] Jun K, Conley R, Huang Y, Lim H, Yu K. Solving linear systems by quadratic unconstrained binary optimization on D-wave quantum annealing device. In: Quantum information science, sensing, and computation XIII, vol. 11726. International Society for Optics and Photonics; 2021, p. 117260C.