A quantum walk-assisted approximate algorithm for bounded NP optimisation problems

S. Marsh · J. B. Wang

Received: 27 June 2018 / Accepted: 2 January 2019 / Published online: 16 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
This paper describes an application of the quantum approximate optimisation algorithm (QAOA) to efficiently find approximate solutions for computational problems contained in the polynomially bounded NP optimisation complexity class (NPO PB). We consider a generalisation of the QAOA state evolution to alternating quantum walks and solution-quality-dependent phase shifts and use the quantum walks to integrate the problem constraints of NPO problems. We apply the concept of a hybrid quantum-classical variational scheme to attempt finding the highest expectation value, which contains a high-quality solution. We synthesise an efficient quantum circuit for the constrained optimisation algorithm, and we numerically demonstrate the behaviour of the circuit with respect to an illustrative NP optimisation problem with constraints, minimum vertex cover. With examples, this paper demonstrates that the degree of accuracy to which the quantum walks are simulated can be treated as an additional optimisation parameter, leading to improved results.

Keywords Quantum optimisation · Quantum walks · QAOA · Minimum vertex cover

1 Introduction

Quantum computers exploit the properties of quantum mechanics such as superposition and entanglement, providing the ability to solve certain computational problems [1–3] far more efficiently than any classical computer. However, the power of quantum computation does not apply indiscriminately to all computational problems. It is an active area of study as to whether a quantum advantage applies to the class of ‘NP optimisation problems.’
In 2014, Farhi et al. [4] published a new algorithmic framework called the quantum approximate optimisation algorithm (QAOA) for finding approximate solutions to combinatorial optimisation problems using quantum computation. This framework was applied to some example optimisation problems, returning ‘good’ solutions according to the relevant metric. For a combinatorial optimisation problem with integer objective function \( c \) where solutions can be encoded using \( n \) bits, they define a diagonal quantum operator \( \hat{C} \) by its action on the \( n \)-dimensional computational basis states \( \ket{x} \) such that \( \hat{C} \ket{x} = c(x) \ket{x} \). The authors choose the canonical mixing operator \( \hat{B} = \sum_{i=1}^{n} \sigma_{i}^{x} \), where \( \sigma_{i}^{x} \) is the Pauli-X operator acting on the \( i \)th qubit of the register. The authors then make use of the quantum adiabatic theorem [5]. Since \( \hat{B} \) satisfies the Perron–Frobenius requirements, by evolving a system initially in the highest-eigenvalue eigenstate of \( \hat{B} \) under the influence of a Hamiltonian which slowly interpolates from \( \hat{B} \) to \( \hat{C} \) over a large time \( T \), the final state of the system will be the highest-eigenvalue eigenstate of \( \hat{C} \). Taking the linear interpolation

\[
\hat{H}(t) = \frac{t}{T} \hat{C} + \left(1 - \frac{t}{T}\right) \hat{B}, \quad t \in [0, T]
\]

and performing Trotterisation on the time evolution into \( p \) time steps followed by a further Trotterisation on each of the resultant terms lead to the state evolution

\[
\ket{\beta, \gamma} = e^{-i\beta_{p} \hat{B}} e^{-i\gamma_{p} \hat{C}} \ldots e^{-i\beta_{1} \hat{B}} e^{-i\gamma_{1} \hat{C}} \ket{s}.
\]

The state \( \ket{s} \) is the \( n \)-dimensional equal superposition, corresponding to the highest-eigenvalue eigenstate of this particular \( \hat{B} \). The \( 2p \) unknowns \( \vec{\beta} = (\beta_{1}, \ldots, \beta_{p}) \) and \( \vec{\gamma} = (\gamma_{1}, \ldots, \gamma_{p}) \) are treated as optimisation parameters, with the optimal values corresponding to an evolution path that replicates that of \( \hat{H}(t) \) as closely as the parameter space allows. The search space can be restricted to \( \gamma \in [0, 2\pi)^{p} \) and \( \beta \in [0, \pi)^{p} \) because both \( \hat{C} \) and \( \hat{B} \) have integer eigenvalues. The QAOA takes the optimal parameter values which maximise the expectation value \( F_{p}(\vec{\beta}, \vec{\gamma}) = \left\langle \beta, \gamma \right| \hat{C} \right| \beta, \gamma \rangle \), since a high expectation value with respect to \( \hat{C} \) means a solution \( x \) with a high value of \( c(x) \) on average. The QAOA has the critical properties that \( \lim_{p \to \infty} \max_{\vec{\beta}, \vec{\gamma}} F_{p}(\vec{\beta}, \vec{\gamma}) = \max_{x} c(x) \) and \( \max_{\vec{\beta}, \vec{\gamma}} F_{p}(\vec{\beta}, \vec{\gamma}) \geq \max_{\vec{\beta}, \vec{\gamma}} F_{p-1}(\vec{\beta}, \vec{\gamma}) \). Consequently, the algorithm’s performance improves with \( p \), guaranteeing the optimal solution in the limit. Farhi et al. then restricted to very low \( p \), choosing to study \( p = 1 \) for the NP optimisation (NPO) problem of maximum cut.

In this paper, we consider a generalisation of the QAOA state evolution as a series of quantum walks interleaved with solution-quality-dependent phase shifts and use the quantum walks to integrate the problem constraints of NPO problems. A continuous time random walk on a graph \( G \) models the flow of probability between neighbouring vertices on the graph. This concept was extended to the quantum domain by [6]. Consider a graph \( G = (V, E) \) with adjacency matrix \( A \). For our purposes, it is convenient to assume that \( G \) has \( 2^{n} \) vertices, so the vertices can be identified with the \( 2^{n} \)-dimensional computational basis states. Then, the continuous-time quantum
A quantum walk-assisted approximate algorithm for bounded… Page 3 of 18

walk on $G$ can be defined by the propagator $\hat{U}(t) = e^{-it\hat{A}}$ with respect to a $n$-qubit quantum register, where $\hat{A}$ is the $2^n$-dimensional quantum operator defined on the computational basis states by the adjacency matrix $A$. The probability distribution over the graph after time $t$ is held in the probability of measuring each of the basis states after the action of operator $\hat{U}(t)$ on the initial quantum state. In contrast to the classical random walk, interference and other quantum phenomena can come into effect. This leads to markedly different behaviour between the two cases. Continuous-time quantum walks provide a versatile platform for universal quantum computation [7]. They have been used extensively in graph theoretical applications [8–11] and are the basis of many other quantum algorithms [12–15].

The QAOA also requires an efficient method for determining a specific problem-dependent expectation value. We adopt the recent concept of a hybrid quantum-classical variational scheme [16] for this purpose and prove its efficiency for all problems in NPO PB. These results are aggregated to present a final quantum algorithm for finding approximate solutions to any problem in the NPO PB class, using the minimum vertex cover problem as a representative example. A vertex cover of a graph $G = (V, E)$ is a subset of the vertices such that for every $\langle u, v \rangle \in E$, either $u$ or $v$ is in the set. That is, every edge has at least one end in the set. Out of all vertex covers existing for $G$, the minimum vertex cover is the one with the fewest vertices. This is an NPO PB problem [17], with the goal to maximise the number of vertices not in the vertex cover. As well as being a useful problem to state and study in terms of computational complexity, minimum vertex cover has wide applicability to real-world problems [18–22]. The algorithm is shown to produce high-quality solutions efficiently for various classes of minimum vertex cover problem instances.

In the following section, we demonstrate how the constraints associated with NPO problems can be encoded into the QAOA framework. Next, we give an efficient strategy for finding the optimal QAOA parameters for NPO problems with polynomially bounded measure. We provide an efficient quantum circuit to carry out an approximation to the desired state evolution. Finally, we numerically study the behaviour of the circuit using the minimum vertex cover problem and show that a lower quantum walk simulation accuracy can lead to improved results for some instances. Thus, we propose that the degree of quantum walk simulation accuracy can be treated as a further optimisation parameter.

2 Encoding NPO problems

An NP optimisation problem [23] is a combinatorial optimisation problem constrained by the following conditions:

- It must be efficient to determine whether a solution is feasible, or valid.
- The size of each feasible solution $x$ must be bounded from above by some polynomial function of the size of the problem instance, where the size can be represented as the length of its binary string encoding.
- The measure $c(x)$, also called the objective function, is efficiently computable.

Then, without loss of generality, given a problem instance, the aim of a NP optimisation problem is to find $x$ such that

 Springer
\[ c(x) = \max(\{c(x') : x' \text{ is feasible}\}). \]

The complexity class NPO labels the set of all NP optimisation problems. The polynomially bounded NP optimisation problem class (NPO PB) adds the further restriction of \( c \) being bounded by some polynomial function in the size of the problem instance \([23]\). Many optimisation problems such as minimum vertex cover, graph partitioning and maximal clique are contained in NPO PB \([17]\). Optimisation problems such as integer programming, number partitioning and travelling salesman are not contained in NPO PB \([24]\). For real-world application of NPO problems, a heuristic can suffice, which aims to find a good solution efficiently.

Consider the maximum-size feasible solution \( x \) for a given problem instance of an NPO problem. It must be possible to encode \( x \) in some unique binary string of length \( n \), with \( n \) growing at most polynomially in the size of the problem instance. Given that \( x \) is the maximum-size solution by definition, all other feasible solutions \( x' \) can also be represented by unique length-\( n \) binary strings. Consequently, it suffices to consider the integers in the range \( 0 \ldots 2^n - 1 \). Some of these integers will correspond to feasible solutions, while others may not. Thus, there is a natural way to encode the measure of any NPO problem into the QAOA, by defining \( \hat{C} \ket{x} = c(x) \ket{x} \). Without modification, the QAOA will work to produce solutions \( x \) with a high value of \( c(x) \). However, these produced solutions need to be feasible.

In order to enforce this, the structure of \( \hat{B} \) needs to be considered. The transverse field operator \( \hat{B} \) can also be defined equivalently as

\[ \hat{B} \ket{x} = \sum_{i=0}^{n-1} |n_i(x)) \]

where \( n_i(x) = x \oplus (1 \ll i) \) is the \( i \)th neighbour of \( x \), produced by toggling the \( i \)th bit of the binary string \( x \), \( i = 0 \ldots (n-1) \). The matrix form of \( \hat{B} \) is the adjacency matrix of a hypercube, as illustrated in Fig. 1.

![Fig. 1](image_url)  
**Fig. 1** Representation of the transverse field operator \( \hat{B} \) as an \( n \)-dimensional hypercube in three (left) and four (right) dimensions
For any NPO problem, it is efficient by definition to compute whether \( x \) is feasible. Hence, a validation function \( v \) can be defined, such that

\[
v(x) = \begin{cases} 
1 & \text{if } x \text{ is a feasible solution}, \\
0 & \text{otherwise}.
\end{cases}
\] (5)

The function value \( v(x) \) is efficiently computable for all \( x \in \{0, \ldots, 2^n - 1\} \). Then, defining a modified \( \hat{B} \) operator, we can incorporate problem constraints,

\[
\hat{B} |x\rangle = \sum_{i=0}^{n-1} v_i(x) |n_i(x)\rangle,
\] (6)

where \( v_i(x) = v(x)v(n_i(x)) \). This function is 1 if \( x \) and its \( i \)th neighbour are both feasible, and it is easily checked that \( \hat{B} \) maintains hermiticity with this modification.

This acts as a modification of the hypercube into a subgraph containing the feasible solutions. The infeasible solutions which do not satisfy problem constraints have degree 0. \( \hat{B} \) still satisfies the Perron–Frobenius requirements, and thus \( \hat{H}(t) \) will continue to satisfy the adiabatic conditions [5]. Figure 2 presents a vertex cover-specific example of the \( \hat{B} \) operator.

The unitary operator \( e^{-i\beta\hat{B}} \) represents a continuous-time quantum walk [6] over the feasible states of the adjacency matrix \( \hat{B} \). This adjacency matrix represents a modified hypercube, where vertices correspond to unique solutions to the combinatorial optimisation problem in question. By modifying the hypercube operator \( \hat{B} \) to disconnect the feasible solutions from the infeasible solutions, and modifying the initial state \( |s\rangle \) to have nonzero probability only for states representing feasible solutions, the quantum walk via \( e^{-i\gamma\hat{C}} \) will never ‘enter’ an infeasible state. In combination with the fact that \( e^{-i\gamma\hat{C}} \) is a diagonal unitary, this means that the state \( |\vec{\beta}, \vec{\gamma}\rangle \) (with an appropriately modified initial state) will guarantee a feasible solution when measured.

![Fig. 2](image-url) a An example four-vertex input graph, for which the aim is to find the minimum vertex cover. b Illustration of the connectivity of the modified hypercube operator, for the given vertex cover problem instance. The solid green subgraph is the feasible region, and each dashed line touches an infeasible solution and so is deleted from our mixing operator (Color figure online)
It is required that the highest-energy state of $\hat{B}$ be known and efficiently preparable, in order to perform an adiabatic evolution starting from this state [5]. The original transverse field (or hypercube) operator $\hat{B}$ satisfied this requirement, with the highest-energy state being the equal superposition state. However, incorporation of constraints via $\hat{B}$ means that the highest-eigenvalue state is no longer the equal superposition. Rather, it will be some non-trivial superposition over the states dependent on the structure of the constraints and the problem instance—not at all easy to find, let alone prepare efficiently. To circumvent this issue, a ‘prior’ adiabatic evolution can be performed. This is done in [4] for the specific NPO problem maximum independent set. The first evolution is from the highest-eigenvalue state of $-\hat{C}$ to the highest-eigenvalue state of $\hat{B}$. The highest-eigenvalue state of $-\hat{C}$ is equivalent to the ground state of $\hat{C}$, corresponding to the lowest-quality solution to the NPO problem. For many NPO problems, this lowest-quality solution can be found efficiently. For example, the lowest-quality vertex cover of a graph corresponds to the cover using every one of the $n$ available vertices. However, if the lowest-quality solution cannot be found efficiently, the algorithm supports a generalisation to the use of any feasible solution as the initial state. This inspires a transition from an adiabatic perspective (requiring the initial state to be the lowest-quality feasible solution) to a quantum walk perspective (supporting any feasible state as the initial state). The modified QAOA state evolution is described below.

Given a particular NP optimisation problem, assume that an initial feasible solution $|s\rangle$ can be efficiently found and prepared. Then, for level-$p$ QAOA, define $2p - 1$ parameters $\vec{\beta} = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$ and $\vec{\gamma} = (\gamma_1, \ldots, \gamma_{p-1}) \in (0, 2\pi)^{p-1}$. With these parameters, the state evolution is defined as the alternating series of operators

$$
|\vec{\beta}, \vec{\gamma}\rangle = e^{-i\beta_p \hat{B}} e^{-i\gamma_{p-1} \hat{C}} \cdots e^{-i\gamma_1 \hat{C}} e^{-i\beta_1 \hat{B}} |s\rangle.
$$

(7)

The $e^{-i\beta \hat{B}}$ operator encodes the problem constraints through the modified hypercube operator $\hat{B}$ and performs a continuous-time quantum walk over the valid states dependent on the parameter $\beta$. The $e^{-i\gamma \hat{C}}$ operator encodes the NPO measure through the diagonal operator $\hat{C}$, modifying the relative phase of the computational basis states $|x\rangle$ depending on the quality of the solution $c(x)$. The walk operators are the components of the evolution that performs the amplification. Since the high-quality states have a uniquely distinguished phase due to applications of $e^{-i\gamma \hat{C}}$, the amplitude of these states will be amplified relative to the lower-quality solutions at some point during the quantum walk. Note that the $\vec{\beta}$ parameters can no longer be restricted to $[0, \pi)$ because $\hat{B}$ does not necessarily have integer eigenvalues. Specifically, it is straightforward to find modified hypercubes such that the ratio between two eigenvalues is irrational, and therefore in general $\hat{B}$ is not periodic [25] so we cannot put any bound on the range of $\vec{\beta}$.

There is a subtlety involved in this method for integrating problem constraints, which forces an additional limit on the computational problems that fit into the algorithmic framework. The assumption made is that by disconnecting the valid solutions from the invalid ones, the subgraph containing the valid solutions is connected. In fact, the restriction is slightly weaker—there must be a path from the initial state of
the algorithm to the solution state (or at least to sufficiently high-quality solutions), for any problem instance. A wide range of NPO problems fit this description. Take for example the NP optimisation problem of minimum vertex cover. Consider any arbitrary vertex cover represented by bit string \(x\) on a graph \(G = (V, E)\). By adding another unused vertex to the cover \(x\), the resulting set represented by \(x'\) is still a vertex cover: all edges \((ij) \in E\) are still covered. In addition, there is an edge in \(B\) connecting \(x\) and \(x'\), since they differ in a single bit and both represent vertex covers. The same argument applies to \(x'\), creating a path of edges from \(x\) up to \(11\ldots1\) (the vertex cover using every vertex). Hence, there is a path along the modified hypercube from every vertex cover to the solution \(11\ldots1\), and by extension every other vertex cover. So the subgraph of the hypercube operator representing valid vertex covers is connected.

This type of connectivity is a general property of a large number of NP optimisation problems, including set packing, maximum cut, maximum independent set, maximum clique and hitting set. A typical NP optimisation problem aims to minimise/maximise the number of elements in the set, under some constraint. By adding/removing elements to/from the set, respectively, the solution is worsened but still satisfies the constraint. Problems with this property will satisfy the connectivity requirement. The reader is invited to confirm that the examples given, from Karp’s [26] original list of 21 NP-complete problems, are some of the problems that fall into this category. It may also be possible to choose \(B\) differently such that the subgraph remains connected for any problem instance. Recent research has explored the use of various mixing operators in the QAOA [27].

3 Finding the optimal parameters

To estimate the expectation value \(F_p(\vec{\beta}, \vec{\gamma})\), the state \(\lvert \vec{\beta}, \vec{\gamma} \rangle\) is prepared and sampled using the illustrative quantum circuit depicted in Fig. 3. The composition of the quantum circuit is given in more detail in Sect. 4. We repeatedly set up the state \(\lvert \vec{\beta}, \vec{\gamma} \rangle\), measure the state to obtain a solution bit string \(x\) and then evaluate \(c(x)\). The average of these \(c(x)\) values will converge to the expectation value \(F_p(\vec{\beta}, \vec{\gamma})\). This estimate is fed back to the classical optimiser, and the parameters \(\vec{\beta}, \vec{\gamma}\) in the quantum state evolution can then be adjusted accordingly. This is the so-called hybrid quantum-classical approach as adapted in [16].

**Theorem 1**  \(F_p(\vec{\beta}, \vec{\gamma})\) can be efficiently determined using the variational method for any NPO PB problem. An NPO PB problem of size \(n\) has measure \(c\) which is bounded by \([0, c_{\text{max}}(n)]\), such that \(c_{\text{max}}(n)\) grows at most polynomially in \(n\). The expectation value \(F_p(\vec{\beta}, \vec{\gamma}) = \langle \vec{\beta}, \vec{\gamma} \rvert \hat{C} \rvert \vec{\beta}, \vec{\gamma} \rangle\) also lies in this range. According to the central limit theorem, the number of samples required to estimate the mean of a population with variance \(\sigma^2\) to within \(\epsilon\) is \(z^2\sigma^2/\epsilon^2\), where \(z\) is the \(z\)-score associated with the required confidence interval (CI) [28]. Using Popoviciu’s inequality on variances [29], for a bounded distribution in \([0, c_{\text{max}}(n)]\) the variance is at most \(\frac{1}{4}c_{\text{max}}(n)^2\). Hence, for a fixed confidence interval, the number of samples required grows like \(\mathcal{O}\left(c_{\text{max}}(n)^2/\epsilon^2\right)\). This is polynomial with respect to \(n\).  \(\Box\)
Fig. 3 Illustration of the hybrid quantum-classical variational method for finding the optimal QAOA parameters $\vec{\beta}^*$ and $\vec{\gamma}^*$. The dashed regions are the parameterised components of the phase and quantum walk operators, respectively.

Hence, any problem which fits into the NPO PB class is a suitable candidate for this algorithm. With this in mind, the overall variational QAOA process is as follows. Start with some arbitrary initial $\vec{\beta}$ and $\vec{\gamma}$. Repeatedly construct and measure the state $\ket{\vec{\beta}, \vec{\gamma}}$ to get a bit string $x$, and evaluate $c(x)$. Enough repetitions will give a satisfactory estimate of $F_p(\vec{\beta}, \vec{\gamma})$. Return this value to the optimiser, and obtain a new updated set of parameters $\vec{\beta}$ and $\vec{\gamma}$. This process repeats until a maximum is found, and the optimiser terminates. Throughout this process, keep track of the highest seen value of $c(x)$ and the corresponding bit string $x$. At the end of the algorithm, this $x$ is taken as the solution.

4 Explicit quantum circuit

In this section, we provide an explicit quantum circuit for the implementation of the QAOA state evolution. We also prove the efficiency of the circuit and give an expression for the circuit complexity.

4.1 The measure operator

There always exists an efficient quantum circuit for the implementation of $e^{-i\gamma \hat{C}}$. Welch et al. [30] provide a strategy for generation of a quantum circuit to implement this operator without use of ancilla qubits for diagonal $\hat{C}$ having efficiently computable elements. This builds from work by Childs [31], who proved that if $\hat{C}$ is diagonal and has efficiently computable elements, then an efficient quantum circuit for $e^{-i\gamma \hat{C}}$ can be found. This circuit is given in Fig. 4. For any NP optimisation problem, it is efficient to compute the measure value $c(x)$ for any input $x$ by definition, and thus an efficient quantum circuit can be found.

In addition, we note that for NP optimisation problems with a simple measure, it is straightforward to find a circuit for $e^{-i\gamma \hat{C}}$ directly. For example, in the case of minimum vertex cover, $e^{-i\gamma \hat{C}}$ can be implemented up to a global phase via applying $R_Z(\gamma)$ to each qubit. This follows from $\hat{C} = \frac{1}{2} \sum_{i=1}^{n} (1 - \hat{\sigma}_i^z)$, which counts the number of qubits in the $|1\rangle$ state.
Fig. 4 Quantum circuit for implementation of $e^{-i\gamma\hat{C}}$ where $c(x)$ is now an arbitrary objective function, and $\hat{c}$ the corresponding quantum operator to compute $c(x)$ in binary [31]. That is, $\hat{c} |x\rangle = |c(x)\rangle$. The integer $k$ is the number of bits of precision required to represent $c(x)$. Explicitly, $k = \lceil \log \epsilon_{\text{max}}(n) \rceil$.

4.2 The modified hypercube operator

An efficient quantum circuit for $e^{-i\beta\hat{B}}$ also exists. In 2003, Aharonov and Ta-Shma [32] proposed a method for efficient implementation of $e^{-i\beta\hat{B}}$ as long as $\hat{B}$ is efficiently row-computable. A Hamiltonian $\hat{H}$ is efficiently row-computable if for every computational basis state $|b\rangle$, all the nonzero matrix elements $\langle a | \hat{H} | b \rangle$ can be efficiently found.

We can verify that $\hat{B}$ is efficiently row-computable for any NPO problem. Given basis state $|b\rangle$, set $x \leftarrow v(b)$. Then, for $i \leftarrow 1, 2, \ldots, n$ toggle bit $i$ of $b$ to produce $a_i$. Set $y_i \leftarrow v(a_i)$. If $y_i \land x$, then $\langle a_i | \hat{B} | b \rangle = 1$. This produces all the nonzero elements of row $a$ of $\hat{B}$ as per Eq. (6) and makes only $(n + 1)$ calls to $v(x)$ which is known to run in polynomial time. Hence, $\hat{B}$ is efficiently row-computable. Consequently, $e^{-i\beta\hat{B}}$ always has an efficient quantum circuit. Here, we give an explicit circuit to carry out this unitary, reminiscent of the method given in [31] for sparse matrix simulation. We assume the existence of an efficient ‘validation’ oracle

$$\hat{U}_v |x\rangle |y\rangle = |x\rangle |y \oplus v(x)\rangle.$$  

We can safely make the assumption that $\hat{U}_v$ is efficient, since there is an efficient classical circuit to compute the validation function. The classical circuit can be used to construct an efficient reversible circuit, which in turn can be translated into an efficient quantum circuit. We next construct an operator $\hat{V}_j$ such that

$$\hat{V}_j |x, y, z\rangle = |x, y \oplus n_j(x), z \oplus v_j(x)\rangle.$$  

The operator acts on a Hilbert space $\mathcal{H}^{2^n} \otimes \mathcal{H}^{2^n} \otimes \mathcal{H}^2$ and is unitary with $\hat{V}_j^\dagger = \hat{V}_j$.

Next, on the same space as $\hat{V}_j$, we define a Hermitian (but not unitary) operator $\hat{S} = \hat{S} \otimes |1\rangle \langle 1|$, where the $\hat{S}$ operator performs $n$ pairwise qubit swaps between the $|x\rangle$ and $|y\rangle$ registers. Using these two operators $\hat{V}_j$ and $\hat{S}$, we can replicate the action of $\hat{B}$. We consider the operator
\( \hat{B} = \sum_{j=0}^{n-1} \hat{V}_j \hat{S} \hat{V}_j \). \hspace{1cm} (10)\)

**Theorem 2** When restricted to the subspace of states \( |x, 0, 0\rangle \), \( \hat{B} = \hat{B} \).

\[
\hat{B} |x, 0, 0\rangle = \sum_{j=0}^{n-1} \hat{V}_j \hat{S} \hat{V}_j |x, 0, 0\rangle = \sum_{j=0}^{n-1} \hat{V}_j |n_j(x), x, v_j(x)\rangle = \sum_{j=0}^{n-1} v_j(a) |n_j(x), x \oplus n_j(n_j(x)), v_j(x) \oplus v_j(n_j(x))\rangle = \sum_{j=0}^{n-1} v_j(x) |n_j(x), 0, 0\rangle = \hat{B} |x, 0, 0\rangle
\]

where we use \( n_j(n_j(x)) = x \) and \( v_j(n_j(x)) = v_j(x) \).

The terms in Eq. (10) do not commute in general—they only commute when \( v(n_j(a)) = v(n_k(a)) \) for all \( j, k \). In other words, commutativity only occurs in the two trivial cases where either all solutions are feasible (original hypercube) or none are (empty graph). Hence, we find a circuit for \( e^{-i\beta \hat{B}} \) via finding a circuit for each \( e^{-i\beta \hat{V}_j \hat{S} \hat{V}_j} = \hat{V}_j e^{-i\beta \hat{S} \hat{V}_j} \), and then combining them to approximate the correct evolution to the desired level of accuracy. For example, using a first-order approximation [33] where \( \hat{V}_j \hat{S} \hat{V}_j \) has unit norm, we have

\[
e^{-i\beta \hat{B}} = \left( V_0 e^{-i\beta \hat{S}/m} V_1 e^{-i\beta \hat{S}/m} V_2 \ldots V_{n-1} e^{-i\beta \hat{S}/m} V_{n-1} \right)^m + O\left( n\beta^2/m \right).
\] \hspace{1cm} (11)\)

**Theorem 3** The operator \( e^{-i\beta \hat{S}} \) can be expressed as \( \mathbb{I} \otimes |0\rangle \langle 0| + e^{-i\beta \hat{S} \otimes |1\rangle \langle 1|}. \)

\[
e^{-i\beta \hat{S}} = \sum_{k=0}^{\infty} \frac{(-i\beta)^k}{k!} \hat{S}^k \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I} \otimes \mathbb{I} + \sum_{k=1}^{\infty} \frac{(-i\beta)^k}{k!} \hat{S}^k \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \mathbb{I} \otimes |0\rangle \langle 0| + e^{-i\beta \hat{S} \otimes |1\rangle \langle 1|}.
\]

This is exactly the exponential swap operator [34] controlled by \( |z\rangle \). Figure 5 illustrates the quantum circuit for \( \hat{V}_j e^{-i\beta \hat{S}} \hat{V}_j \). Each application of this circuit has an intuitive interpretation, having the following effect on a computational basis state:
Fig. 5 $\hat{V} e^{-i\beta \hat{S}} \hat{V}$ circuit, using $2n + 3$ total qubits, where $\oplus_j$ is a not gate applied to the $j$th qubit of the $x$ register.

$$|x\rangle \mapsto \begin{cases} 
\cos \beta |x\rangle - i \sin \beta |n_j(x)\rangle & \text{if } x \text{ and } n_j(x) \text{ are both feasible solutions,} \\
|x\rangle & \text{otherwise.}
\end{cases}$$

Hence, one application of $\hat{V} e^{-i\beta \hat{S}} \hat{V}$ mixes $x$ with its $j$th neighbour, if and only if they are both feasible solutions.

### 4.3 QAOA state evolution complexity

We require the validation oracle complexity to be $O(\text{poly}(n))$. Then, the circuit $\hat{V} e^{-i\beta \hat{S}} \hat{V}$ in Fig. 5 has complexity $O(\text{poly}(n))$, since the other sections have only linear complexity. Using Eq. (11), we need to take $m = O(n\beta^2/\epsilon)$ repetitions of the circuit to simulate the desired quantum walk with accuracy $\epsilon$. With higher-order Lie product formulae, the complexity of each quantum walk becomes $O(\text{poly}(n)\beta^{1+\delta}/\epsilon^{\delta})$ for any choice of $\delta > 0$. Overall, we interleave the circuits given in Sects. 4.1 and 4.2 to get the overall level-$p$ QAOA state evolution complexity,

$$O(\text{poly}(n) \beta^m),$$

where $m = n\beta_{\max}^{1+\delta}/\epsilon^{\delta}$ for a choice of $\delta > 0$ and $\beta_{\max} = \max(\beta_1, \ldots, \beta_p)$.

Note that in order to accurately simulate each quantum walk, the circuit complexity scales at least linearly with the values considered for the $\vec{\beta}$ optimisation parameters. For a low-depth circuit, we would prefer to keep each $\beta$ small (or within a constant range). Note that as $p \to \infty$, we reproduce the original adiabatic evolution, with the optimal parameter values approaching 0. Thus, as $p$ increases, the optimal $\beta$ values decrease. In Sect. 5, it is shown that similarly to $p$, choosing low $m$ still leads to high-quality results and in some cases exceeds the performance of choosing higher $m$. Thus, $m$ can be treated as an additional (discrete) optimisation parameter for the algorithm.

Once a value for $m$ is chosen, the full state evolution circuit can be constructed. At each step, the classical optimiser produces new values for the gate parameters $\vec{\gamma}$ and $\vec{\beta}$ as shown in Figs. 4 and 5. The gate parameters are adjusted, and the quantum state evolution is carried out. This is repeated until a local optimum is found.
5 Numerical demonstration

In this section, we present some numerical results to demonstrate the behaviour of our state evolution circuit on the minimum vertex cover problem. Google’s Cirq library [35] is used to classically simulate the circuit. A Nelder–Mead nonlinear optimiser [36] is used to minimise the expectation value \( F_p(\vec{\beta}, \vec{\gamma}) \), which in the case of vertex cover represents the mean number of vertices used in the superposition of covers.

In the case of vertex cover, the validation function uses \( O(n^2) \) operations (looping through each of the graph edges to ensure all have at least one end in the cover) and the measure function uses \( O(n) \) operations (counting the number of on bits in the solution binary string). Hence, the state evolution complexity is

\[
O\left( pn^2 m \right) = O\left( pn^3 \beta_{\text{max}}^2 / \epsilon \right), \tag{14}
\]

where \( \delta = 1 \) (first-order Lie product formula approximation) is chosen for simplicity.

The following sections provide results for some four-vertex graphs using \( p = 1 \), so that the algorithm optimises the time of a single quantum walk. The QAOA state evolution takes the form

\[
|\beta\rangle = \left( \prod_{j=0}^{3} \hat{V}_j e^{-i \beta \hat{S}_j / m} \right)^m |1111\rangle. \tag{15}
\]

We study the quality of the solutions produced by the circuit for different choices of \( m \). In addition, we include in “Appendix A” some preliminary results examining the average solution quality in the exact simulation case, where \( m \to \infty \).

5.1 Example 1

First, consider the example graph given in Fig. 2, which has two minimum vertex covers 0110 and 1010. A validation circuit for this graph is obtained by converting the edge-covering constraint \( v(x) = (x_1 \lor x_2) \land (x_2 \lor x_3) \land (x_3 \lor x_4) \land (x_4 \lor x_2) \) into exclusive sum of products (ESOP) form, \( v(x) = (x_2 \land x_4) \oplus (x_2 \land x_3 \land \neg x_4) \oplus (x_1 \land \neg x_2 \land x_3 \land x_4) \).

This ESOP expression leads naturally to a quantum circuit, as shown in Fig. 6.

Figure 7 shows the final probability distribution of the \( |\beta^*\rangle \) state for \( m = 1, 10 \) and 100, where \( \beta^* \) is the optimised walk time parameter computed by the variational method. Consider first the \( m = 1 \) case. All of the probability is condensed into the 0110 optimal solution with negligible probability in any other state, giving the optimal

---

*Fig. 6* Validation oracle \( U_v \) for the problem instance in Fig. 2.

---

\[
|\beta\rangle = \left( \prod_{j=0}^{3} \hat{V}_j e^{-i \beta \hat{S}_j / m} \right)^m |1111\rangle.
\]
expectation value of 2 with respect to $\tilde{C}$. The optimised walk time is $\beta^* = \frac{\pi}{2}$, which according to Eq. (12) performs $|x\rangle \mapsto -i |n_j(x)\rangle$ when the $j$th neighbour of $x$ is feasible. Thus, the state evolution can be interpreted as a direct path through the hypercube $1111 \mapsto 0111 \mapsto 0111 \mapsto 0101 \mapsto 0101$, where the second and fourth steps remain in the same solution state since 0011 and 0100 are not feasible solutions.

When $m = 10$, as per Fig. 7b, the minimised expectation value is 2.26, with $\beta^* = -0.9$. Approximately 78% of the probability is condensed into the two optimal solutions, with a relatively low chance of measuring a sub-optimal solution. As would be expected, there is no probability of measuring an infeasible solution. There is negligible improvement in the $m = 100$ case, with a final expectation value of 2.25 where $\beta^* = -1.08$. For this example graph, the $m = 1$ performs best since it is able to perform a direct walk through the solution space to an optimal solution without the use of quantum interference. The following example demonstrates that this is not always the case.

### 5.2 Example 2

The four-vertex example graph shown in Fig. 8 has a unique minimum vertex cover $x = 0001$. The final probability distribution using the optimised $\beta$ parameter is also shown for the $m = 1, 10$ and 100 cases. Choosing $m = 1$ is insufficient to obtain a high-quality result for this graph, with the majority of probability held in the $|0111\rangle$ state and leading to an expectation value of 2.67 with $\beta^* = 0.96$. The direct walk method as in the previous example does not work in this instance, since the structure

![Fig. 7](image-url) Final probability distributions for $p = 1$ constrained QAOA on the graph in Fig. 2
Fig. 8 Final probability distributions for $p = 1$ constrained QAOA on the graph in (a).

Setting $m = 10$ improves the quality markedly, with over 60% chance of measuring the optimal solution and an expectation value of 1.70 using $\beta^* = -0.86$. Increasing
to $m = 100$ allows the algorithm to hone in completely to the optimal 0001 solution state with $\beta^* = -1.67$.

5.3 Example 3

The final example, a four-vertex line graph, is shown in Fig. 9. The line graph has three minimum vertex covers 0101, 0110 and 1010. This is another case where $m = 1$ suffices with $\beta^* = \frac{\pi}{2}$, since the sequence of steps $|1111\rangle \mapsto |0111\rangle \mapsto |0111\rangle \mapsto |0101\rangle \mapsto |0101\rangle$ leads to an optimal solution. Taking $m = 10$ and $m = 100$ also leads to produce high-quality final states, with expectation values 2.07 and 2.11, respectively. Rather than the direct walking approach leading to a single solutions, the final state involves a superposition over more than one optimal solution.

In the above examples, it can be seen that in addition to emulating the continuous quantum walk behaviour for higher-order approximations, the optimiser can also find direct paths through the hypercube to high-quality solutions. In addition, in some instances, lower $m$ leads to improved solution quality. Hence, $m$ can be optionally be treated as an additional optimisation parameter.

6 Conclusion

In this paper, we have presented an algorithm for finding approximate solutions to NP optimisation problems with polynomially bounded measure (NPO PB) using the quantum approximate optimisation algorithm (QAOA). We have shown that the constraints involved with NP optimisation problems can be incorporated into the QAOA state evolution. This is done by interpreting the state evolution as a series of quantum walks and then restricting the quantum walks to the region of feasible solutions. The QAOA also requires a method for efficiently finding the value of a certain expectation value. We have demonstrated that the recent concept of a hybrid quantum-classical variational algorithm suits for this purpose and is efficient for NP optimisation problems that have polynomially bounded measure. Finally, we provide an explicit quantum circuit to carry out the constrained QAOA evolution to the desired level of accuracy. Finally, we numerically study its behaviour and performance on instances of minimum vertex cover. With examples, it is shown that choosing a lower-depth circuit that less accurately simulates the desired quantum walk can lead to improved results in some cases. We propose that the degree of quantum walk simulation accuracy can be treated as an additional optimisation parameter.

There is significant potential for future work on various aspects of this QAOA-based algorithm. The algorithm supports any efficiently preparable feasible solution as an initial state. Further work could investigate the impact of the choice of initial state, or even a superposition over multiple starting states—in particular whether the ‘worst-case’ solution (corresponding to the ground state of $\hat{C}$) is the best option for the initial state.

The graph used for quantum walks was modified from the transverse field (hypercube) mixer, which is the conventional choice for an adiabatic evolution. However,
other graphs could also be a possibility, and a corresponding quantum circuit could be obtained in a similar way to this work, if the graph is sparse and the neighbours of each vertex are efficiently computable. In terms of the continuous-time quantum walk perspective, a quantum walk over any graph which connects the feasible states is a valid choice. Future work could investigate the impact of different choices for mixing operators in the constrained case.

Finally, applications to many other NP optimisation problems could be explored. Particular problems may produce symmetries in the quantum state evolution, leading to an expectation value which can be evaluated efficiently. Achieving this would remove the requirement for the variational sampling technique and could potentially lead to a guarantee on the approximation factor. This has been done in the original QAOA paper for \( p = 1 \) on the maximum-cut problem, but not for the modified version incorporating constraints. Analysing other problems in the NPO PB class and their corresponding state evolutions is a pathway for further research.

Acknowledgements This research was supported by a Hackett Postgraduate Research Scholarship and an Australian Government Research Training Program Scholarship at the University of Western Australia. We thank Lyle Noakes for his helpful comments and discussions.

Appendix A: Numerical results for random graphs

Classical simulations of the quantum state evolution were performed to evaluate the quality of approximate solutions in the context of minimum vertex cover. We define the approximation quality for a particular problem instance as the ratio of the number of vertices in the minimum vertex cover to the approximate cover. Since for a \( n \)-qubit quantum register the classical computer must store all \( 2^n \) quantum amplitudes in memory, results were obtained for only low-\( n \) simulations (\( \lesssim 20 \)). To reduce the computational cost, the state evolution is carried out by direct computation and multiplication of the matrices \( e^{-i\gamma \hat{C}} \) and \( e^{-i\beta \hat{B}} \), rather than simulating the quantum circuit. As in Sect. 5, the Nelder–Mead nonlinear optimiser [36] is used to optimise the expectation value \( F_p(\vec{\beta}, \vec{\gamma}) \).

The performance of the \( p = 2 \) algorithm was tested on a random sample of \( G(n, 0.5) \) Erdős–Rényi graphs. The \( G(n, 0.5) \) Erdős–Rényi random graph model [37] has equal probability to select each of the \( 2^{n(n-1)/2} \) \( n \)-vertex graphs, so it gives a good impression of the ‘average case’ performance of the heuristic. The solution qualities for each random graph are shown in Fig. 10, with 20 instances considered per \( n \). Taking \( n = 5 \) as an example, the optimal vertex cover is found for all but two random instances tested. The solution quality decreases reasonably slowly, and for all trialled graphs the produced solution used at most 1.6 times the number of vertices as the optimal solution.

Note that the results in this ‘Appendix’ are an exact simulation of the desired QAOA state evolution, rather than a simulation of the quantum circuit. Hence, the results here are specific to the case \( m \to \infty \), where \( m \) is the ‘walk simulation accuracy’ parameter introduced in Sect. 4.3. The solution qualities would be further improved by optimising over this parameter.
Fig. 10  Ratio of the number of vertices in the approximate cover to the minimum vertex cover. The grey line is the mean, and the shaded region is the 95% CI. The size of each data point is proportional to the number of tested instances with the same approximation quality

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26, 1484 (1997)
2. Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: STOC, p. 59. ACM (2003)
3. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: STOC, p. 212. ACM (1996)
4. Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm (2014). arXiv:1411.4028 [quant-ph]
5. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum Computation by Adiabatic Evolution (2000). arXiv:quant-ph/0001106
6. Farhi, E., Gutmann, S.: Quantum computation and decision trees. Phys. Rev. A 58, 915 (1998)
7. Childs, A.M., Gosset, D., Webb, Z.: Universal computation by multiparticle quantum walk. Science 339, 791 (2013)
8. Izaac, J.A., Zhan, X., Bian, Z., Wang, K., Li, J., Wang, J.B., Xue, P.: Centrality measure based on continuous-time quantum walks and experimental realization. Phys. Rev. A 95, 032318 (2017)
9. Loke, T., Tang, J.W., Rodriguez, J., Small, M., Wang, J.B.: Comparing classical and quantum PageRanks. Quantum Inf. Process. 16, 25 (2017)
10. Gamble, J.K., Friesen, M., Zhou, D., Joynt, R., Coppersmith, S.N.: Two-particle quantum walks applied to the graph isomorphism problem. Phys. Rev. A 81, 052313 (2010)
11. Xu, S., Sun, X., Wu, J., Zhang, W.-W., Arshed, N., Sanders, B.C.: Quantum walk on a chimera graph. N. J. Phys. 20, 053039 (2018)
12. Childs, A.M., Goldstone, J.: Spatial search by quantum walk. Phys. Rev. A 70, 022314 (2004)
13. Li, Z.-J., Wang, J.B.: An analytical study of quantum walk through glued-tree graphs. J. Phys. A Math. Theor. 48, 355301 (2015)
14. Manouchehri, K., Wang, J.B.: Physical Implementation of Quantum Walks. Springer, Berlin (2014)
15. Qiang, X., Loke, T., Montanaro, A., Aungskunsiri, K., Zhou, X., O’Brien, J.L., Wang, J.B., Matthews, J.C.F.: Efficient quantum walk on a quantum processor. Nat. Commun. 7, 11511 (2016)
16. 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. 5, 4213 (2014)
17. Lucas, A.: Ising formulations of many np problems. Front. Phys. 2, 5 (2014)
18. Filiol, E., Franc, E., Gubbioni, A., Moquet, B., Roblot, G.: Combinatorial optimisation of worm propagation on an unknown network. World Acad. Sci. Eng. Technol. 23, 373 (2007)
19. Lancia, G., Bafna, V., Istrail, S., Lippert, R., Schwartz, R.: SNP’s, problems, complexity, and algorithms. In: Computers and Games, p. 182. Springer (2001)
20. O’Callahan, R., Choi, Jong Deok: Hybrid dynamic data race detection. ACM SIGPLAN Not. 38, 167 (2003)
21. Hou, Aimin: A theory of measurement in diagnosis from first principles. Artif. Intell. 65, 281 (1994)
22. Domingo, C., Mishra, N., Pitt, L.: Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries. Mach. Learn. 37, 89 (1999)
23. Kann, V.: Strong lower bounds on the approximability of some NPO PB-complete maximization problems. In: Wiedermann, J., Hájek, P. (eds.) Mathematical Foundations of Computer Science, p. 227. Springer (1995)
24. Kann, V.: Polynomially bounded minimization problems which are hard to approximate. In: Lingas, A., Karlsson, R., Carlsson, S. (eds.) Automata, Languages and Programming, p. 52. Springer (1993)
25. Godsil, C.: State transfer on graphs. Discrete Math. 312, 129 (2012)
26. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds.) Complexity of Computer Computations, p. 85. Springer (1972)
27. Hadfield, S., Wang, Z., O’Gorman, B., Rieffel, E.G., Venturelli, D., Biswas, R.: From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz (2017). arXiv:1709.03489 [quant-ph]
28. Feller, W.: The fundamental limit theorems in probability. Bull. Am. Math. Soc 51, 800 (1945)
29. Sharma, R., Gupta, M., Kapoor, G.: Some better bounds on the variance with applications. J. Math. Inequal. 4, 355 (2010)
30. Welch, J., Greenbaum, D., Mostame, S., Aspuru-Guzik, A.: Efficient quantum circuits for diagonal unitaries without ancillas. N. J. Phys. 16, 033040 (2014)
31. Childs, A.M.: Quantum information processing in continuous time. Ph.D. thesis, Massachusetts Institute of Technology (2004)
32. Aharonov, D., Ta-Shma, A.: Adiabatic quantum state generation and statistical zero knowledge. In: STOC, p. 20. ACM (2003)
33. Suzuki, M.: General theory of higher-order decomposition of exponential operators and symplectic integrators. Phys. Lett. A 165, 387 (1992)
34. Lau, H.-K., Pooser, R., Siopsis, G., Weedbrook, C.: Quantum machine learning over infinite dimensions. Phys. Rev. Lett. 118, 080501 (2017)
35. Google. Cirq. https://github.com/quantumlib/Cirq (2018). Accessed 30 Oct 2018
36. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. 7, 308 (1965)
37. Erdős, P., Rényi, A.: On random graphs. I. Publ. Math. 6, 290 (1959)