US 20200057957A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2020/0057957 A1**

**Johnson et al.** (43) **Pub. Date: Feb. 20, 2020**

(54) **QUANTUM COMPUTER WITH IMPROVED QUANTUM OPTIMIZATION BY EXPLOITING MARGINAL DATA**

(71) Applicant: **Zapata Computing, Inc.**, Cambridge, MA (US)

(72) Inventors: **Peter D. Johnson**, Somerville, MA (US); **Maxwell D. Radin**, Cambridge, MA (US); **Jhonathan Romero**, Somerville, MA (US); **Yudong Cao**, Cambridge, MA (US); **Amara Katabarwa**, Cambridge, MA (US)

(21) Appl. No.: **16/543,165**

(22) Filed: **Aug. 16, 2019**

**Related U.S. Application Data**

(60) Provisional application No. 62/719,330, filed on Aug. 17, 2018.

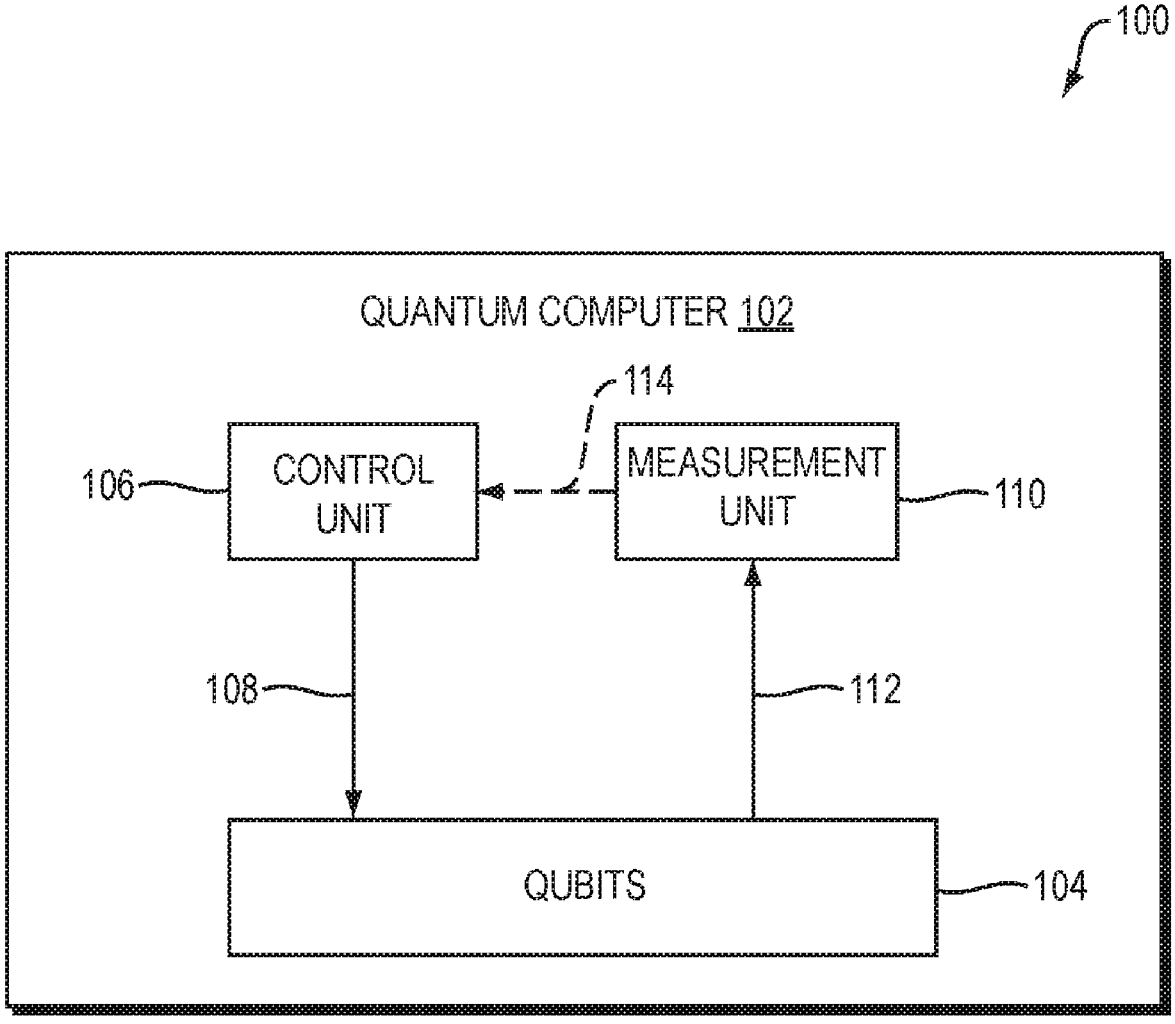
**Publication Classification**

(51) **Int. Cl.**
***G06N 10/00*** (2006.01)
***G06F 17/14*** (2006.01)

(52) **U.S. Cl.**
CPC ..... ***G06N 10/00*** (2019.01); ***G06F 17/14*** (2013.01)

(57) **ABSTRACT**

A quantum optimization system and method estimate, on a classical computer and for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables; and transform, on the classical computer, one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian.



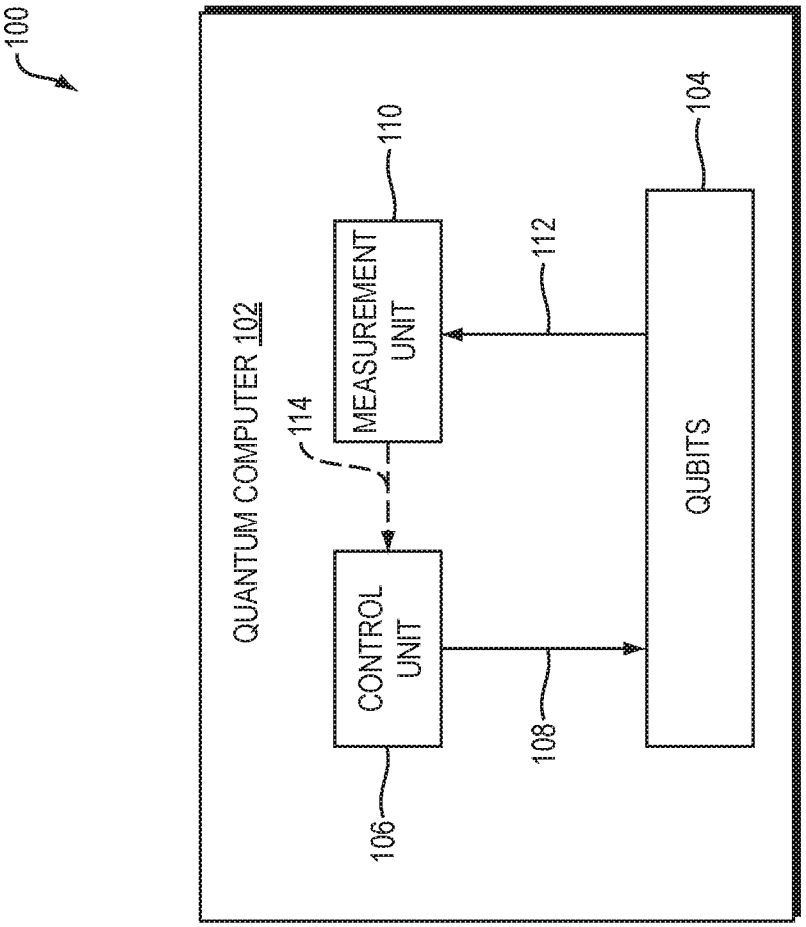

100
QUANTUM COMPUTER 102
MEASUREMENT UNIT
110
112
104
QUBITS
114
CONTROL UNIT
106
108

FIG. 1

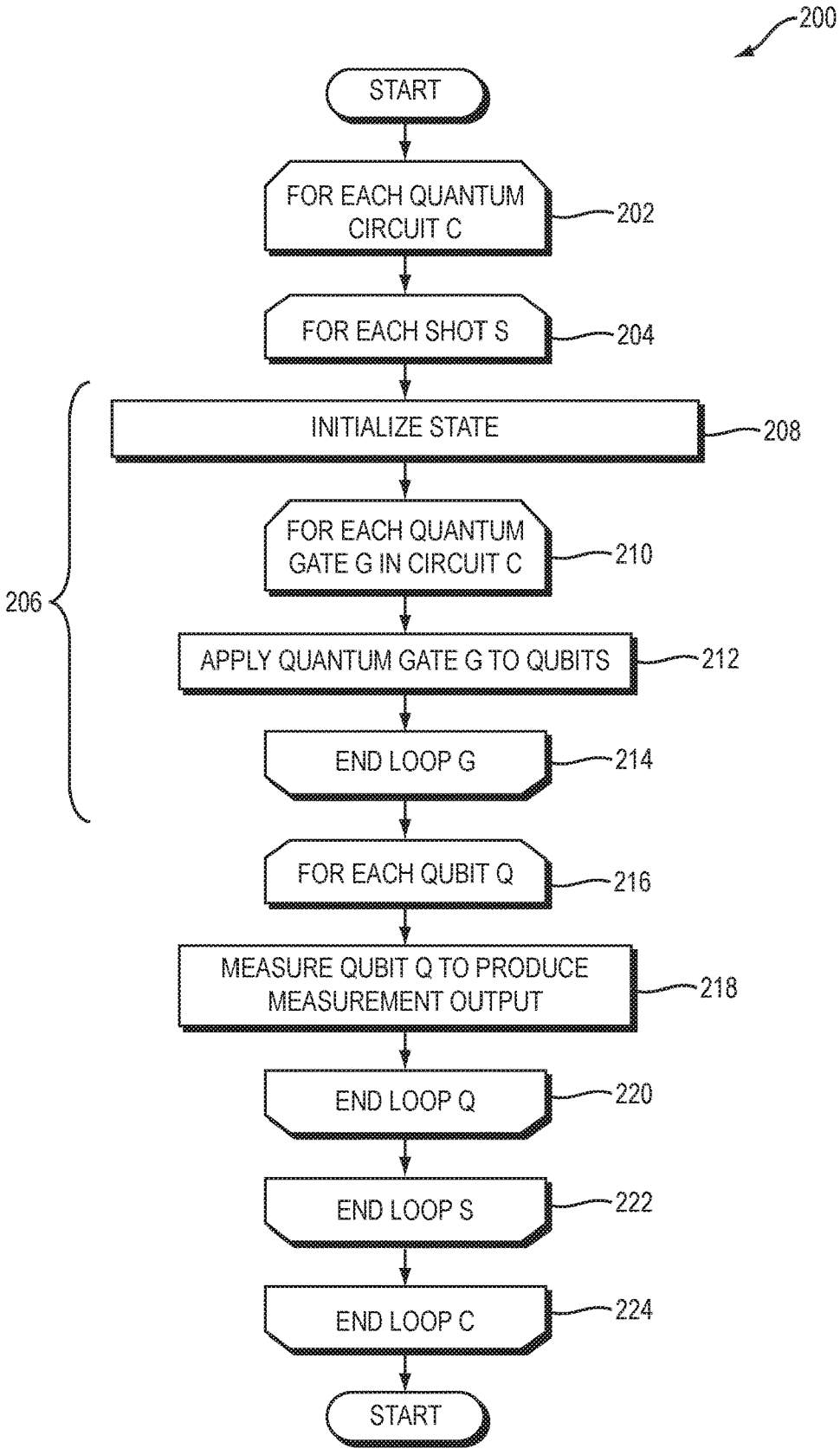

200
START
FOR EACH QUANTUM CIRCUIT C
202
FOR EACH SHOT S
204
206
INITIALIZE STATE
208
FOR EACH QUANTUM GATE G IN CIRCUIT C
210
APPLY QUANTUM GATE G TO QUBITS
212
END LOOP G
214
FOR EACH QUBIT Q
216
MEASURE QUBIT Q TO PRODUCE MEASUREMENT OUTPUT
218
END LOOP Q
220
END LOOP S
222
END LOOP C
224
START

FIG. 2A

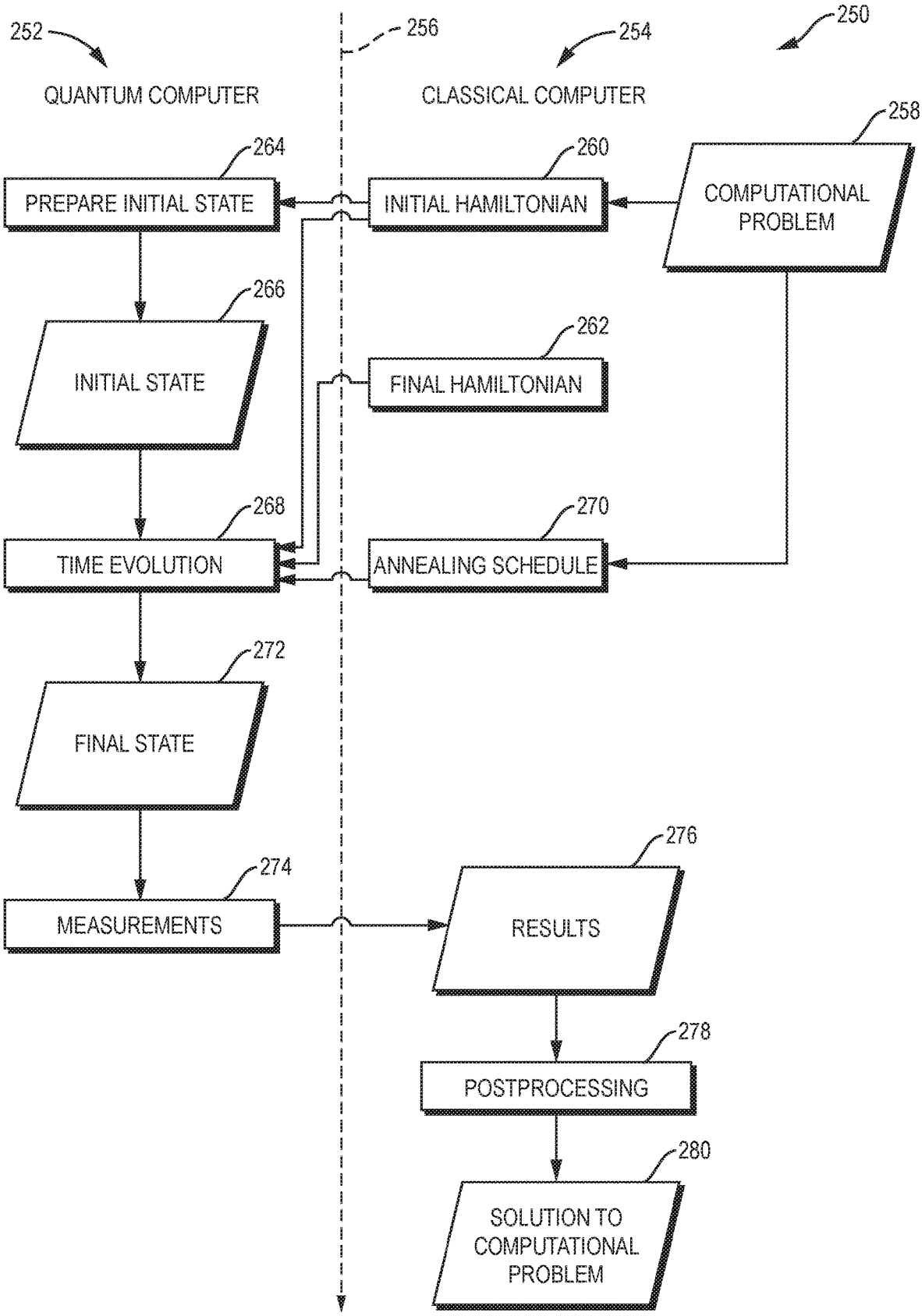

252
QUANTUM COMPUTER
256
254
CLASSICAL COMPUTER
250
258
COMPUTATIONAL PROBLEM
264
PREPARE INITIAL STATE
260
INITIAL HAMILTONIAN
266
INITIAL STATE
262
FINAL HAMILTONIAN
268
TIME EVOLUTION
270
ANNEALING SCHEDULE
272
FINAL STATE
276
RESULTS
274
MEASUREMENTS
278
POSTPROCESSING
280
SOLUTION TO COMPUTATIONAL PROBLEM

FIG. 2B

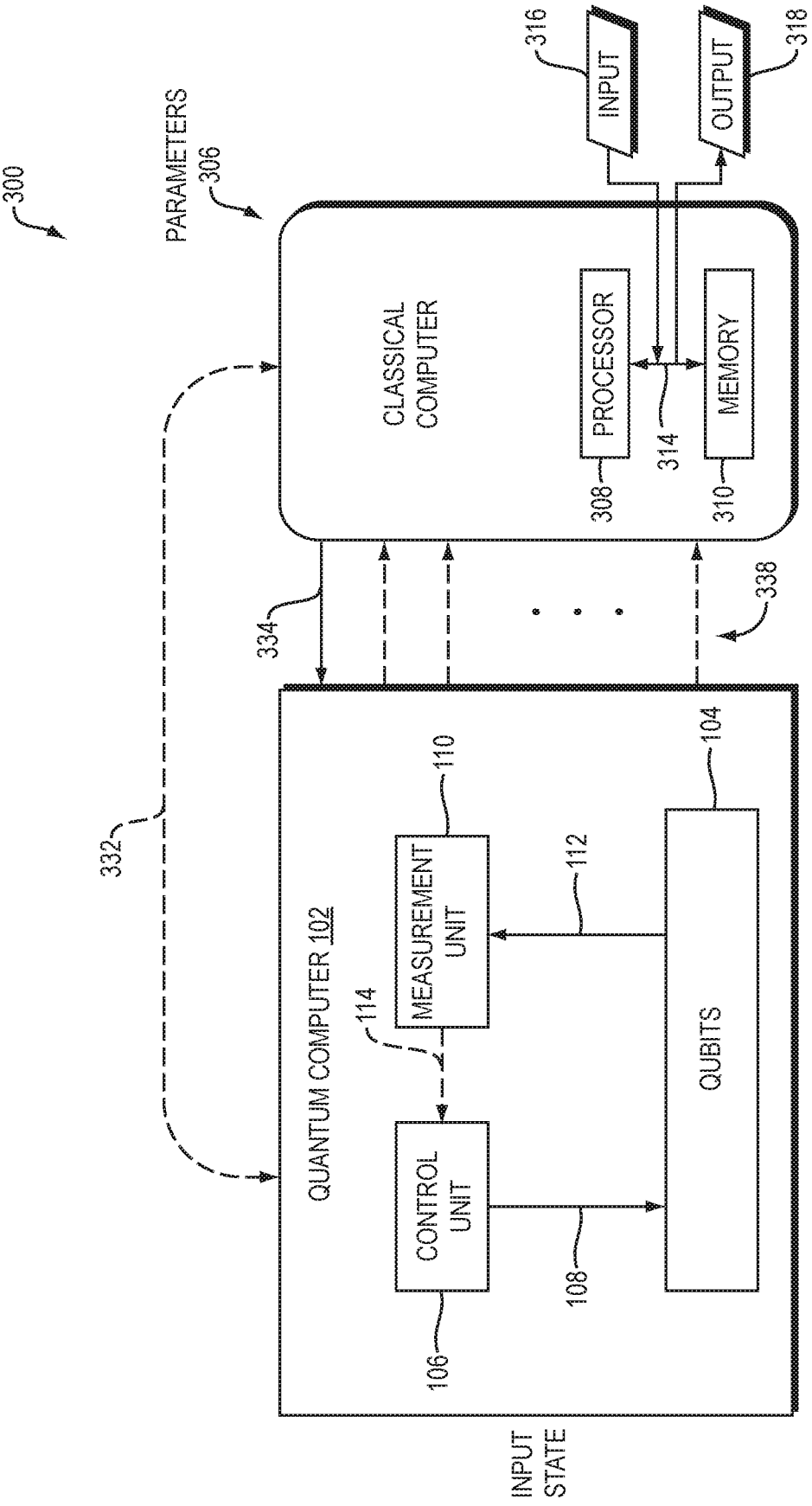

300
PARAMETERS
306
316
INPUT
318
OUTPUT
CLASSICAL COMPUTER
PROCESSOR
308
314
MEMORY
310
334
338
332
QUANTUM COMPUTER 102
MEASUREMENT UNIT
110
112
114
CONTROL UNIT
106
108
QUBITS
104
INPUT STATE
MEASUREMENT READOUT

FIG. 3

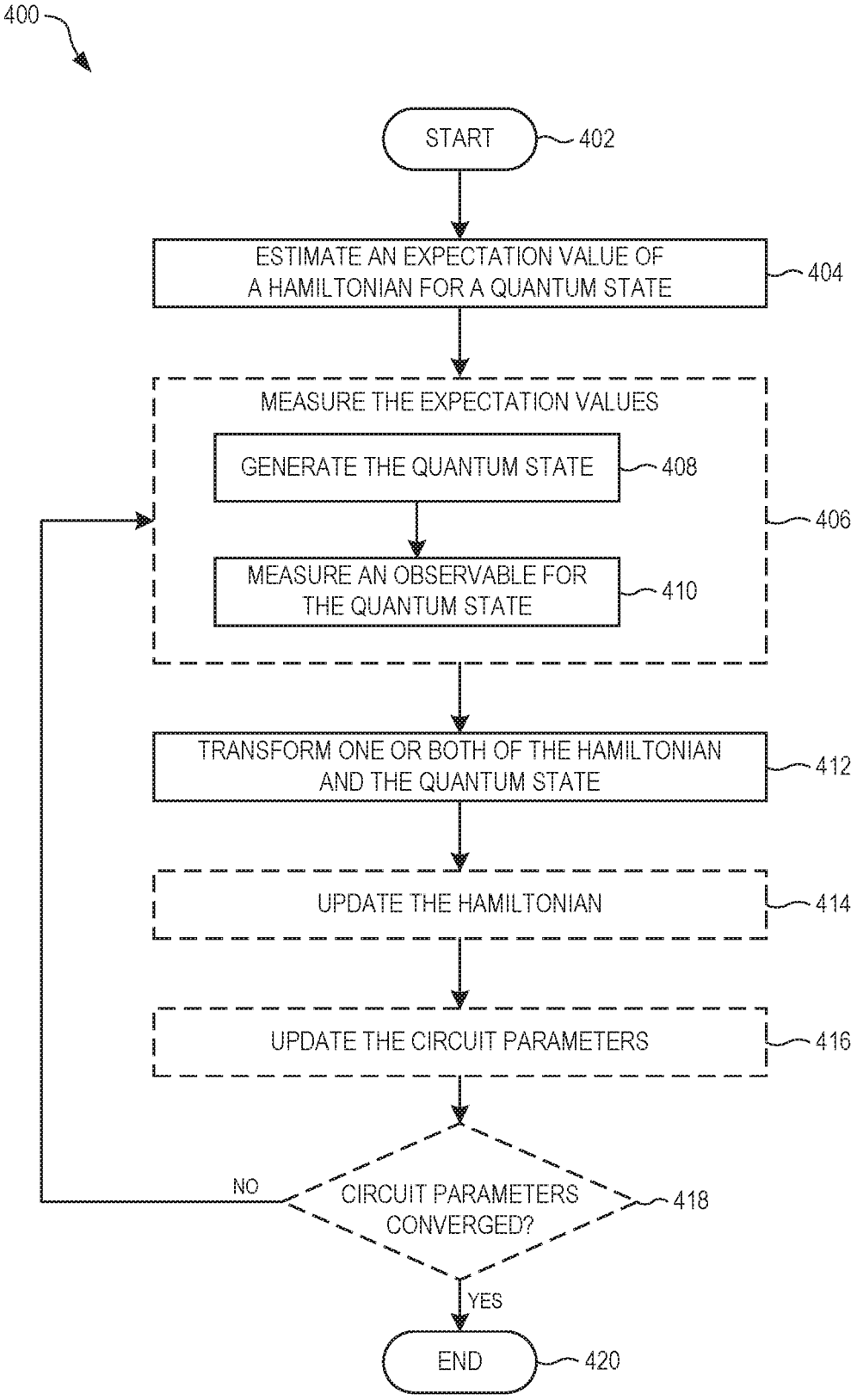

400
START
402
ESTIMATE AN EXPECTATION VALUE OF A HAMILTONIAN FOR A QUANTUM STATE
404
MEASURE THE EXPECTATION VALUES
GENERATE THE QUANTUM STATE
408
MEASURE AN OBSERVABLE FOR THE QUANTUM STATE
410
406
TRANSFORM ONE OR BOTH OF THE HAMILTONIAN AND THE QUANTUM STATE
412
UPDATE THE HAMILTONIAN
414
UPDATE THE CIRCUIT PARAMETERS
416
NO
CIRCUIT PARAMETERS CONVERGED?
418
YES
END
420

FIG. 4

# QUANTUM COMPUTER WITH IMPROVED QUANTUM OPTIMIZATION BY EXPLOITING MARGINAL DATA

## BACKGROUND

**[0001]** Quantum computers promise to solve industry-critical problems which are otherwise unsolvable. Key application areas include chemistry and materials, bioscience and bioinformatics, logistics, and finance. Interest in quantum computing has recently surged, in part, due to a wave of advances in the performance of ready-to-use quantum computers. Although these machines are not yet able to solve useful industry problems, the precipice of utility seems to be rapidly approaching.

**[0002]** In addition to the improvements in the quantum hardware, recent algorithmic developments have created new opportunities for achieving quantum utility. In particular, the variational quantum eigensolver (VQE) algorithm and the quantum approximate optimization algorithm (QAOA) do not demand the same degree of quantum resources as do the well-known Shor's factoring and Grover's search algorithm. These new algorithms are used to solve the ground-state energy problem and the well-known, NP-hard combinatorial optimization problem MAXCUT, respectively. Therefore, these algorithms give hope for solving useful problems on near-term quantum hardware.

**[0003]** There are a number of limiting factors that must be overcome to implement useful instances of these algorithms on near-term quantum devices. First, the device needs to bear a sufficiently large number of qubits. Otherwise, classical methods for solving the problem will outperform the quantum computer. Second, the processes, or quantum gates, on the quantum computer need to be of sufficiently high fidelity. Low fidelity quantum gates will output quantum states with degraded coherence, and coherent output states are necessary for many of these applications. The performance of algorithms such as VQE and QAOA can, in principle, can be systematically improved by including more and more layers of quantum gates. The number of layers of gates is referred to as the circuit depth. By increasing the circuit depth, one increases the expressibility of the output state. For example, the maximal degree of quantum entanglement in the output state is related to the depth of the circuit preparing the state. Preparing a variety of entangled states in a coherent fashion is necessary for the functioning of many algorithms. Therefore, when executing a quantum algorithm such as VQE or QAOA, one seeks to maximize the expressibility of the circuit, while at the same time minimizing the decoherence of the output state.

**[0004]** Ideally, a quantum computer would implement perfect gates, so that one would not need to decrease depth to minimize decoherence. This is not, however, possible in practice.

## SUMMARY

**[0005]** A quantum optimization system and method estimate, on a classical computer and for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables; and transform, on the classical computer, one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian.

**[0006]** Other features and advantages of various aspects and embodiments of the present invention will become apparent from the following description and from the claims.

**[0007]** In a first aspect, a quantum optimization method includes estimating, on a classical computer and for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables. The quantum optimization method also includes transforming, on the classical computer, one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian.

**[0008]** In certain embodiments of the first aspect, the method further includes measuring the expectation value of each of the observables on a quantum computer by generating the quantum state on the quantum computer, and measuring, on the quantum computer, each of the observables for the quantum state.

**[0009]** In certain embodiments of the first aspect, generating the quantum state includes generating the quantum state with a parametrized quantum circuit programmable via one or more circuit parameters.

**[0010]** In certain embodiments of the first aspect, the method further includes updating the one or more circuit parameters such that the parametrized quantum circuit outputs an updated quantum state that better approximates a ground state of the Hamiltonian.

**[0011]** In certain embodiments of the first aspect, the method further includes repeating (i) generating the quantum state with the parametrized quantum circuit, (ii) measuring each of the observables for the quantum state, (iii) transforming one or both of the Hamiltonian and the quantum state, (iv) updating the Hamiltonian based on the transforming, and (v) updating the one or more circuit parameters, until the one or more circuit parameters have converged.

**[0012]** In certain embodiments of the first aspect, transforming one or both of the Hamiltonian and the quantum state includes applying a unitary transformation to said one or both of the Hamiltonian and the quantum state.

**[0013]** In certain embodiments of the first aspect, the method further includes generating, on the classical computer, the expectation value of each of the observables.

**[0014]** In certain embodiments of the first aspect, the method further includes updating, on the classical computer, a first representation of the quantum state based on the expectation value of the Hamiltonian to better approximate a ground state of the Hamiltonian.

**[0015]** In certain embodiments of the first aspect, the method further includes repeating (i) generating the expectation value of each of the observables, (ii) transforming one or both of the Hamiltonian and the quantum state, and (iii) updating the first representation of the quantum state, until the first representation of the quantum state has converged.

**[0016]** In certain embodiments of the first aspect, the linear combination of the observables includes at least one observable with a zero weight that becomes non-zero due to said transforming the Hamiltonian. Furthermore, the expectation values of the observables include an expectation value for the at least one observable with a zero weight.

**[0017]** In certain embodiments of the first aspect, transforming one or both of the Hamiltonian and the quantum state includes applying a fermionic transformation to the one or both of the Hamiltonian and the quantum state.

**[0018]** In certain embodiments of the first aspect, the fermionic transformation includes rotations of active orbitals.

**[0019]** In certain embodiments of the first aspect, the fermionic transformation includes transformations out of an active space to incorporate at least one of a core orbital and a virtual orbital.

**[0020]** In certain embodiments of the first aspect, the fermionic transformation includes rotations that respect one or more of an open-shell spin symmetry, a closed-shell spin symmetry, and a geometric symmetry.

**[0021]** In certain embodiments of the first aspect, the method further includes implementing a quantum subspace expansion technique.

**[0022]** In certain embodiments of the first aspect, the method further includes implementing a marginal projection technique.

**[0023]** In certain embodiments of the first aspect, the method further includes obtaining any of the expectation values the observables via orbital frames.

**[0024]** In certain embodiments of the first aspect, transforming one or both of the Hamiltonian and the quantum state includes applying a Majorana fermionic transformation to the one or both of the Hamiltonian and the quantum state.

**[0025]** In certain embodiments of the first aspect, the method further includes minimizing the expectation value of the Hamiltonian using a Givens parameterization.

**[0026]** In certain embodiments of the first aspect, the method further includes minimizing the expectation value of the Hamiltonian using semidefinite programming.

**[0027]** In certain embodiments of the first aspect, transforming one or both of the Hamiltonian and the quantum state includes applying a spin transformation to the one or both of the Hamiltonian and the quantum state.

**[0028]** In certain embodiments of the first aspect, the method further includes minimizing the expectation value of the Hamiltonian using semidefinite programming.

**[0029]** In certain embodiments of the first aspect, the Hamiltonian is an Ising Hamiltonian configured for solving a combinatorial optimization problem.

**[0030]** In certain embodiments of the first aspect, the method further includes minimizing the expectation value of the Hamiltonian using semidefinite programming.

**[0031]** In certain embodiments of the first aspect, transforming one or both of the Hamiltonian and the quantum state includes minimizing the expectation value of the Hamiltonian estimated for the quantum state.

**[0032]** In certain embodiments of the first aspect, minimizing the expectation value of the Hamiltonian includes minimizing the expectation value of the Hamiltonian using semidefinite programming.

**[0033]** In a second aspect, a computer system configured for quantum optimization includes a processor and a memory communicably coupled with the processor and storing machine-readable instructions. Wen executed by the processor, the machine-readable instructions control the computing system to estimate, for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables. The machine-readable instructions also control the computing system to transform one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian estimated for the quantum state.

**[0034]** In certain embodiments of the second aspect, the computing system includes a quantum computer that is communicably coupled with the processor and configured to measure the expectation value of each of the observables.

## BRIEF DESCRIPTION OF THE FIGURES

**[0035]** FIG. **1** is a diagram of a quantum computer according to one embodiment of the present invention;

**[0036]** FIG. **2**A is a flowchart of a method performed by the quantum computer of FIG. **1** according to one embodiment of the present invention;

**[0037]** FIG. **2**B is a diagram of a hybrid quantum-classical computer which performs quantum annealing according to one embodiment of the present invention;

**[0038]** FIG. **3** is a diagram of a hybrid quantum-classical computer according to one embodiment of the present invention; and

**[0039]** FIG. **4** is a flow chart of a quantum optimization method, in embodiments.

## DETAILED DESCRIPTION

**[0040]** Embodiments of the present invention are directed to a quantum computer which effectively appends quantum gates to a variational quantum circuit to artificially extend the depth of the circuit. When the quantum computer executes variational quantum algorithms, the resulting estimates that are output by the quantum computer are of higher quality than the estimates produced by quantum computers without the appended quantum gates.

**[0041]** Embodiments of the present invention extend the capabilities of quantum computers for solving problems such as, but not limited to, ground state energy calculations and combinatorial optimization. Quantum computers implemented according to embodiments of the present invention require little to no additional quantum computational overhead compared to existing quantum computers because the same effect as appending quantum gates, or increasing circuit depth, is achieved by postprocessing data on a classical computer, without adding any actual quantum gates to the quantum computer.

**[0042]** Embodiments of the present invention may include performing some computations using a classical computer and other computations using a quantum computer. The computations (e.g., optimization routines) executed by the classical computer may be performed in parallel with the computations performed by the quantum computer. In general, the classical and quantum processors may work in tandem to provide an approximately optimal answer. In general, the role of the quantum computer is to generate statistically sampled bit strings, and the role of the classical processor is to analyze these sampled bit strings and to adapt the quantum processor accordingly.

**[0043]** Embodiments of the present invention use quantum computation data that is typically not used, to improve the utility of the quantum computer. In particular, typically when a quantum computer executes a variational quantum algorithm to produce a plurality of output values (e.g., Pauli product expectation value estimates), only a single fixed linear combination of such values is typically used. In contrast, embodiments of the present invention use a plurality of linear combinations of the individual output values, not merely the fixed linear combination of those values, which results in a better estimate of the optimal value being

targeted. Quantum computers implemented according to embodiments of the present invention exploit algebraic structure in the problem instance (e.g., the quantum Hamiltonian) to process this data and to output a lower estimate of the ground state energy.

**[0044]** As stated above, ideally a quantum computer would implement perfect gates, so that one would not need to decrease circuit depth to minimize decoherence. This is not, however, possible in practice, because an increased circuit depth results in increased decoherence. Embodiments of the present invention implement a compromise by using additional classical processing to achieve the same effect as appending ideal quantum gates to the end of the sequence of imperfect quantum gates implemented on the quantum computer. This improves the expressibility of the parameterized quantum circuit, while not detracting from the coherence of the output state. More specifically, in embodiments of the present invention, a quantum computer, having imperfect gates, repeatedly executes a quantum circuit followed by quantum measurement to produce initial output data. A classical computer then processes the initial output data. This classical post-processing effectively appends a sequence of perfect quantum gates to the imperfect gates of the quantum computer. The term "expressibility" refers to the ability of the quantum computer to generate a variety of highly coherent entangled quantum states as the output of the quantum circuit.

**[0045]** Embodiments of the present invention can be used to extend the capabilities of quantum computers when implementing variational quantum algorithms such as VQE or QAOA by producing more accurate results and/or tackling larger problem instances. Such quantum algorithms are used to efficiently generate good approximate solutions to ground state energy problems and combinatorial optimization problems, which have application to drug and materials discovery, as well as route optimization and artificial intelligence.

**[0046]** Some of the embodiments described herein generate, measure, or utilize quantum states that approximate a target quantum state (e.g., a ground state of a Hamiltonian). As will be appreciated by those skilled in the art, there are many ways to quantify how well a first quantum state "approximates" a second quantum state. In this description, any concept or definition of approximation known in the art may be used without departing from the scope hereof. For example, when the first and second quantum states are represented as first and second vectors, respectively, the first quantum state approximates the second quantum state when an inner product between the first and second vectors (called the "fidelity" between the two quantum states) is greater than a predefined amount (typically labeled E). In this example, the fidelity quantifies how "close" or "similar" the first and second quantum states are to each other. The fidelity represents a probability that a measurement of the first quantum state will give the same result as if the measurement were performed on the second quantum state. Proximity between quantum states can also be quantified with a distance measure, such as a Euclidean norm, a Hamming distance, or another type of norm known in the art. Proximity between quantum states can also be defined in computational terms. For example, the first quantum state approximates the second quantum state when a polynomial time-sampling of the first quantum state gives some desired information or property that it shares with the second quantum state.

**[0047]** In the prior art variational quantum algorithm, the problem Hamiltonian may be defined and mapped to a sum of Pauli product terms. The quantum state may then be prepared, and the expectation value of each Pauli term may be measured on the quantum computer. The energy expectation value may then be estimated, and a classical optimization routine may be used to suggest new state preparation parameters based on the energy expectation value estimate. The algorithm may then return to the beginning and prepare the updated quantum state based on the updated circuit parameters.

**[0048]** A quantum computer implemented according to an embodiment of the present invention (which may operate under the control of a classical computer) may also begin by defining the problem Hamiltonian and mapping the problem Hamiltonian to a sum of Pauli product terms. A quantum computer implemented according to an embodiment of the present invention may prepare the quantum state and measure the expectation value of each Pauli term. The quantum computer implemented according to an embodiment of the present invention, or a classical computer which receives output of the previous steps from the quantum computer, may then perform a marginals optimization procedure (MOP), such as by optimizing energy and updating the Hamiltonian. The classical computer may then optimize the energy in any of a variety of ways by exploiting structure in the marginal expectation values to carry out additional minimization steps towards obtaining the minimum energy. The outputs of this energy optimization are the optimized energy value and the optimal transformation, both of which may be used to update the Hamiltonian.

**[0049]** Once the energy has been optimized and the Hamiltonian has been updated, the quantum computer implemented according to an embodiment of the present invention may update the circuit parameters based on the estimated energy of the updated Hamiltonian, as is typically done in, for example, the variational quantum eigensolver algorithm, and then return to step 1 and prepare the updated quantum state based on the updated circuit parameters. The quantum computer implemented according to an embodiment of the present invention may repeat the process just described any number of times.

**[0050]** A general method of the marginals optimization procedure will now be described in more detail. The marginals optimization procedure makes use of quantum marginal data during the course of a variational quantum algorithm to accelerate the optimization of the algorithm. Such algorithms are examples of hybrid quantum-classical algorithms, in which a quantum processor and classical processor work in tandem to execute an algorithm. Towards this end, the method introduced herein aims to ramp up the effort of the classical processor so as to extract as much utility from the quantum computer's data output as possible.

**[0051]** Variational quantum algorithms such as VQE and QAOA work as follows:

**[0052]** 1. The problem Hamiltonian is defined and mapped to a sum of Pauli product terms $H=\Sigma_i h_i P_i$ on N qubits, where $P_i=\sigma_1^{\mu_{i,1}} \otimes \ldots \otimes \sigma_N^{\mu_{i,N}}$.

**[0053]** 2. A variational quantum circuit $U_T(\theta_T) \ldots U_1(\theta_1)$ prepares quantum states, $|\psi(\vec{\theta})\rangle = U_T(\theta_T) \ldots U_1(\theta_1)|0\rangle^{\otimes N}$.

**[0054]** 3. The expectation value of each Pauli term, $\langle P_i \rangle$, is measured on the quantum computer, with respect to the prepared state $|\psi(\vec{\theta})\rangle$. This is achieved by repeatedly preparing $|\psi(\vec{\theta})\rangle$ and measuring it in the Pauli basis of $P_i$ to record a binary outcome ±1 in each shot.

**[0055]** 4. The energy expectation value is estimated as $\langle H \rangle = \Sigma_i h_i \langle P_i \rangle$.

**[0056]** 5. A classical optimization routine is used to suggest new state preparation parameters based on the energy expectation value estimate. (Note: depending on the classical optimization routine, multiple loss function evaluations may be executed before new circuit parameters are suggested.)

**[0057]** 6. Return to Step 1, preparing the updated state.

**[0058]** MOP may, for example, replace Step 3 above as follows. An observation of MOP is that there is structure in the marginal expectation values $\langle P_i \rangle$ which may be exploited to carry out an additional minimization step towards obtaining the minimum energy. Generally, depending on the structure of the Hamiltonian, there may be a group of unitary operations $U_R$ which preserves the space of marginals:

$$U_R^\dagger P_i U_R = \sum_j R_i^j P_j.$$

**[0059]** This enables a free depth increase of the variational quantum circuit, boosting the opportunity to drive the state preparation towards the ground state. By carrying out the following optimization routine, MOP may search about a manifold of quantum states to find a better approximation to the ground state energy in each loop of the variational quantum algorithm:

$$\min_R \sum_i h_i R_i^j \langle P_j \rangle$$

**[0060]** subject to "certain conditions on R"

**[0061]** There are several possible ways to carry out this optimization routine. We will describe examples of concrete methods for carrying out the optimization for each of the example applications.

**[0062]** The outputs of this marginals optimization procedure are the optimized energy value and the optimal transformation R*. It is useful to view this transformation as a Heisenberg transformation of the target Hamiltonian:

$$H_n \to H_{n+1} \equiv \sum_{i,j} h_i^{(n)} R_i^j P_j = \sum_j h_j^{(n+1)} P_j,$$

where n indexes the Hamiltonian coefficients in the nth loop of the variational quantum algorithm. Such a transformation does not change the spectrum of the Hamiltonian, and thus the ground state energy of $H_n$ is equal to the ground state energy of H. The value of transforming the Hamiltonian is that the variational circuit may be better able to prepare an approximation to the ground state of $H_n$ than to the ground state of the initial Hamiltonian H.

**[0063]** The MOP-enhanced variational quantum algorithm may then work as follows:

**[0064]** 1. The problem Hamiltonian is defined and mapped to a sum of Pauli product terms $H_0 = \Sigma_i h_i P_i$ on N qubits, where $P_i = \sigma_1^{\mu_{i,1}} \otimes \ldots \otimes \sigma_N^{\mu_{i,N}}$.

**[0065]** 2. A variational quantum circuit $U_T(\theta_T) \ldots U_1(\theta_1)$ prepares quantum states, $|\psi(\vec{\theta})\rangle = U_T(\theta_T) \ldots U_1(\theta_1)|0\rangle^{\otimes N}$.

**[0066]** 3. The expectation value of each Pauli term, $\langle P_i \rangle$, is measured on the quantum computer, with respect to the prepared state $|\psi(\vec{\theta})\rangle$. This is achieved by repeatedly preparing $|\psi(\vec{\theta})\rangle$ and measuring it in the Pauli basis of $P_i$ to record a binary outcome ±1 in each shot.

**[0067]** 4. The classical optimization routine

$$\min_R \sum_{i,j} h_j R_i^j \langle P_i \rangle$$

is carried out, which outputs the updated energy expectation value $\Sigma_{i,j} h_j [R^*]_i^j \langle (P_i) \rangle$ and updated Hamiltonian $H_n \to H_{n+1} = \Sigma_{i,j} h_j [R^*]_i^j P_i$, where R* is the optimal transformation which acts unitarily on the Hamiltonian.

**[0068]** 5. An additional classical optimization routine is used to suggest new state preparation parameters based on the energy expectation value estimate. (Note: depending on the classical optimization routine, multiple loss function evaluations may be executed before new circuit parameters are suggested.)

**[0069]** 6. Return to Step 1, preparing the updated state.

**[0070]** More generally, it may be the case that a finite number of additional marginals $Q_j$ should be added in order that the space of marginals, now $\{P_i, Q_j\}$, is preserved by $U_R$. When the size of the set $\{P_i, Q_j\}$ grows no more than polynomially in the number of qubits, for a given application, MOP remains computationally efficient.

**[0071]** It is valuable to view MOP in light of the N-representability problem, or, more generally, the quantum marginal problem. Literature on these problems highlights the fact that the energy expectation value of a k-body Hamiltonian on N qubits is a function of at most

$$O\left(\binom{N}{K} 4^k\right)$$

parameters, as opposed to the $O(4^N)$ needed to describe the state. Furthermore, the space of

$$O\left(\binom{N}{K} 4^k\right)$$

parameters forms a convex set. At first glance, it seems, then, that the problem of energy minimization should be efficiently solvable by carrying out a convex optimization in this few-parameter space. The catch, however, is the fact that the convex set of valid parameter values does not admit a

tractable characterization. Moreover, the N-representability problem and quantum marginal problem have been shown to be QMA-complete.

**[0072]** Through MOP, we gain a more general perspective on the role of the quantum computer in variational quantum algorithms. The quantum computer supplies valid quantum marginal data, while the classical processor calculates energy expectation values and solicits new marginal data through new state preparations. While the standard approach to VQE carries out the optimization based only on the energy expectation value, MOP carries out an additional optimization exploiting the available valid quantum marginal data. Thus, rather than just energy expectation values, the commodity supplied by the quantum computer appears to be the valid quantum marginal data.

**[0073]** We describe MOP as applied, for example, to the case of a Fermionic two-body Hamiltonian. Consider the general fermionic Hamiltonian with Coulomb repulsion in second-quantization form,

$$H = \sum_{i,j} S^{ij} a_i^\dagger a_j + \sum_{i,j,k,l} D^{ijkl} a_i^\dagger a_j^\dagger a_k a_l,$$

and where the $S_{ij}$ account for a "number penalty" term to force the output to be of the proper particle number. When implementing the standard variational quantum eigensolver, the algorithm concludes with an optimized state preparation using a variational circuit outputting the state $\rho(\vec{\theta}^*)$, for which the energy is estimated, up to sampling error, according to

$$\langle E(\vec{\theta}^*)\rangle = Tr(H\rho(\vec{\theta}^*)),$$

where $\rho(\vec{\theta}^*)$ accounts for implementation and readout error. In the final round of VQE, the expectation value of, in general, each term $V_{ij}=\mathrm{Tr}(a_i^\dagger a_j \rho(\vec{\theta}^*))$ and $W_{ijkl}=\mathrm{Tr}(a_i^\dagger a_j^\dagger a_k a_l \rho(\vec{\theta}^*))$ has been computed. These values are known as the 1- and 2-particle reduced density matrices (RDMs). We show how, with just classical post-processing, these RDMs can be used to improve the ground state energy estimation.

**[0074]** First, consider the case that a sequence of error-free, tunable gates $U_{\vec{\lambda}}$ could be applied at the end of the variational circuit, producing $\rho(\vec{\lambda}, \vec{\theta}^*)=U(\vec{\lambda})\rho(\vec{\theta})U(\vec{\lambda})^\dagger$. Then, minimizing the energy with respect to tuning the parameters $\vec{\lambda}$ would, in the worst case, maintain the energy expectation estimation, while in many cases it would lower it. Of course, in practice, introducing such gates is costly because the coherence of the output state is compromised by the additional error from these gates. However, we can achieve the effect of implementing these gates by a particular post-processing technique that we call the marginals optimization procedure (MOP).

**[0075]** Consider re-expressing the energy expectation with these additional gates in the Heisenberg picture,

$$\langle \mathrm{H}\rangle = Tr(HU(\vec{\lambda})\rho(\vec{\theta})U(\vec{\lambda})^\dagger) = Tr(U(\vec{\lambda})^\dagger HU(\vec{\lambda})\rho(\vec{\theta})).$$

**[0076]** In the example of the fermionic Hamiltonian, if we choose the gates $U(\vec{\lambda})$ to correspond to fermionic orbital rotations, then their action on the creation or annihilation operators of the Hamiltonian is to output a linear combination of creation or annihilation operators, respectively,

$$\langle H\rangle = \sum_{i,j} S^{ij} Tr\left(U_R^\dagger a_i^\dagger a_j U_R \rho(\vec{\theta})\right) + \sum_{i,j,k,l} D^{ijkl} Tr\left(U_R^\dagger a_i^\dagger a_j^\dagger a_k a_l U_R \rho(\vec{\theta})\right) =$$
$$S^{ij} R_i^I * R_j^J * Tr\left(a_I^\dagger a_J \rho(\vec{\theta})\right) + D^{ijkl} R_i^I * R_j^J * R_k^K R_l^L Tr\left(a_I^\dagger a_J^\dagger a_K a_L \rho(\vec{\theta})\right),$$

where the summation is inferred by Einstein notation and the R tensors are orbital rotation matrices characterized by an element of SU(N), where N is the number of spin orbitals. Noticing that the traces are simply the 1-RDMs V and 2-RDMs W of $\rho(\vec{\theta})$, as estimated on the quantum computer, we find that we can efficiently (in the number of spin orbitals) compute the expected energy of the states $U_R\rho(\vec{\theta})U_R^\dagger$,

$$\langle H\rangle = Tr(SRVR^\dagger) + Tr(D(R\otimes R)W(R^\dagger\otimes R^\dagger)),$$

where we have reshaped S,V and D,W into N-by-N and $N^2$-by-$N^2$ matrices, respectively. Finally, by carrying out the following optimization problem we can, in general, obtain an improved estimate of the ground state energy,

$$\min_R Tr(SRVR^\dagger) + Tr(D(R\otimes R)W(R^\dagger\otimes R^\dagger))$$
$$\text{subject to } RR^\dagger = I \text{ and } R^\dagger R = I$$

where the two conditions on R ensure unitarity. The energy expectation value obtained from this optimization is ensured to be at most that estimated in the initial VQE experiment. As motivated by the orbital-optimized coupled cluster method and the Brueckner coupled cluster method such orbital rotations stand to improve the energy estimates in practice. The minimization problem above can be approached in a number of ways. While black-box optimization may be used, it may be useful to employ a method which exploits the structure in the optimization problem. We emphasize that the marginals optimization procedure is independent of the particular optimization routine chosen, and we will discuss alternative optimization routines later on.

**[0077]** Finally, we note that, for the Fermionic case, the marginals optimization procedure can be implemented in both a restricted and extended regime. We will discuss these variants in more detail later on. In the first case, it may be that the Hamiltonian does not depend on certain marginals. This may happen, for example, if the Hamiltonian possesses certain symmetries. In such cases, it may be useful to consider just a subgroup of all possible orbital rotations. Indeed, it is possible to restrict the orbital optimization of the equation above to just a subgroup of orbital optimizations. An explicit example of this would be the case where the Hamiltonian possessed spin-symmetry. Then, one may want to restrict the optimization to rotations of just the spatial orbitals. For the second case, consider that the initial Hamiltonian accounted only for a subspace of active orbitals. Following a VQE calculation one may want to extend the

orbital rotations in order to involve orbitals from the inactive space. Such restrictions and extensions are analogous to standard techniques in quantum chemistry.

**[0078]** The application of the marginals optimization procedure to VQE with a unitary coupled-cluster (UCC) ansatz may be viewed as a novel adaption of the classical orbital-optimized coupled cluster (OCC) and Breuckner coupled cluster (BCC) algorithms to quantum computers. The OCC and BCC approaches are based on an ansatz comprised of an orbital rotation and a coupled cluster operator which are simultaneously optimized using a classical computer.

**[0079]** The notion of restricting the orbital rotation to have the same effect on spin-up and spin-down orbitals is analogous to the treatment of spin in other methods based on orbital rotations, such as Hartree-Fock theory and Kohn-Sham density-functional theory. MOP without the spin restriction is analogous to unrestricted Hartree-Fock (UHF), while MOP with spin restrictions is analogous to restricted Hartree-Fock (RHF) in the case of closed-shell systems and restricted open-shell Hartree-Fock (ROHF) in the case of open-shell systems.

**[0080]** The general marginal optimization procedure includes, but is not limited to, using Hamiltonians decomposed into a linear combination of Pauli strings. The only requisite is that estimates for certain quantum marginal data be obtained. An example of using the marginals optimization procedure where the Hamiltonian is not decomposed into Pauli strings is as follows. The orbital frames method [Motta, Mario, et al. "Low rank representations for quantum simulation of electronic structure." arXiv preprint arXiv: 1808.02625 (2018)] provides a decomposition of the Hamiltonian in terms of products of fermionic number operators, each rotated by an orbital transformation:

$$H=\sum_{\ell=1}^{L}\left[\sum_{i_1}^{N} g_{i_1}^{(\ell,1)} n_{1}^{(\ell)}+\sum_{i_1,i_2=1}^{N} g_{i_1 i_2}^{(\ell,2)} n_{1}^{(\ell)} n_{i_2}^{(\ell)}+\right.$$
$$\left.\sum_{i_1,i_2,i_3=1}^{N} g_{i_1 i_2 i_3}^{(\ell,3)} n_{i_1}^{(\ell)} n_{i_2}^{(\ell)} n_{i_3}^{(\ell)}+\square\sum_{i_1,i_2,\square i_k=1}^{N} g_{i_1,i_2\square}^{(\ell,3)} n_{i_1}^{(\ell)} n_{i_2}^{(\ell)}\square n_{K}^{(\ell)}\right],$$

**[0081]** where

**[0082]** $g_i^{(l,k)}$ represent coefficients for the k-body terms obtained from a decomposition of the Hamiltonian,

**[0083]** L is the number of frames in the decomposition,

**[0084]** K is the highest order term appearing in the Hamiltonian and is equal to 2 when the Hamiltonian is the electronic structure Hamiltonian,

**[0085]** $i_m$ indexes the spin orbitals, and

**[0086]** the number operators are

$$n_i^{(l)}=a_i^{(l)\dagger}a_i^{(l)} \tag{1}$$

with $a_i^{(l)*}$ and $a_i^{(l)}$ being the creation and annihilation operators corresponding to the single-particle orbital

$$\psi_i^{(\ell)}=\sum_j U_{ji}^{(\ell)}\phi_j$$

where $U^{(l)}$ is an N×N matrix obtained from the decomposition,

**[0087]** l is the index of single-particle orbital bases $\{\psi_i^{(l)}\}$ (where i runs from 1 to N) obtained by the decomposition, and by which each value of l indexes a different, so-called, orbital frame.

**[0088]** For a k-body Hamiltonian decomposed in terms of orbital frames, the fermionic marginals up to k-body may be determined by the expectation values of the

$$n_{i_1}^{(\ell)}\ldots n_{i_m}^{(\ell)}.$$

Specifically, the fermionic marginals

$$\square a_{i_1}\ldots a_{i_m}\square$$

may be reconstructed as appropriate linear combinations of the estimated expectation values

$$\square n_{i_1}^{(\ell)}\ldots n_{i_m}^{(\ell)}\square.$$

With the reconstructed fermionic marginals, the marginals optimization procedure may be performed.

**[0089]** To carry out MOP, the transformation on the marginals must be parameterized. One method for parameterizing a unitary or an orthogonal matrix is with an exponential generator. In the unitary case, any anti-Hermitian matrix A generates a corresponding unitary $U_A$ via the matrix exponential $U_A=\exp(A)$. Similarly, for the real orthogonal transformation case using in the Majorana fermionic transformations, any real anti-symmetric matrix B generates a corresponding orthogonal matrix $O_B=\exp(B)$. The coordinates of these matrices A and B constitute the parameters that are varied in a black box optimization used to carry out the marginals optimization.

**[0090]** In carrying out the optimization subroutine of the marginals optimization procedure, it may be beneficial to choose a parameterization of the rotation matrix R which yields analytic gradients. One way to achieve this in the case of fermionic or Majorana fermionic orbital rotations is to employ a, so-called, Givens decompositions of the rotation matrix. Givens decompositions are closely related to "match gate circuits" which are able to be efficiently simulated and give a decomposition of Majorana fermionic orbital transformations. The Givens decomposition of the orbital transformation gives a parameterization of the rotation where the gradient of the energy with respect to these parameters is efficiently computable.

**[0091]** The marginals optimization procedure may be used in conjunction with a number of existing techniques for improving the performance of variational quantum algorithms. One example is the quantum subspace expansion technique introduced in [McClean, Jarrod R., et al. "Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states." Physical Review A 95 (2017): 042308]. In the quantum subspace expansion, a set of operators $\{O_i\}$ is used to generate a subspace of states spanned by $\{O_i\rho O_j^\dagger\}$. The expectation values $H_{ij}=\mathrm{Tr}(\hat{H}O_i\rho O_j^\dagger)$ and $S_{ij}=\mathrm{Tr}(O_i\rho O_i^\dagger)$ are measured and then used to solve the generalized eigenvalue problem $Hv=\lambda Sv$. The

minimum eigenvalue gives the minimum energy of the Hamiltonian $\hat{H}$ in the subspace. This technique has been shown to mitigate error and to improve the performance of optimization. The marginals optimization procedure may be used in conjunction with quantum subspace expansion as follows. After carrying out a round of marginals optimization to produce an updated Hamiltonian H→H', quantum subspace expansion may be performed with respect to the updated Hamiltonian.

**[0092]** The marginal estimates obtained during a variational quantum algorithm will incur a degree of error due to statistical sampling error and device error. This can lead to the set of estimated marginals being invalid, as defined by the well-known quantum marginals problem or, in the fermionic case, the N-representability problem. The marginals projection technique introduced in [Rubin, Nicholas C., Ryan Babbush, and Jarrod McClean. "Application of fermionic marginal constraints to hybrid quantum algorithms." New Journal of Physics 20 (2018): 053020] gives a method for adjusting the estimated marginal values to bring them closer to the set of valid marginals in the case of fermionic problem instances. This technique may be used in conjunction with the marginals optimization procedure by first applying the marginals projection technique to obtain a less-errored set of marginal estimates and then using these improved marginal estimates as input to the marginals optimization procedure.

**[0093]** MOP may be used in a number of different applications. The choice of unitary transformations that are optimized over depend on the structure of the Hamiltonian and may vary from problem to problem. We describe three general classes of transformations: orbital rotations, Majorana rotations, and local unitary transformations. For each class we discuss example applications.

**[0094]** In quantum chemistry and materials science, a basic subroutine is the determination of the ground state energy. The cost of running classical algorithms for accurately estimating the ground state energy of a molecule or material grows exponentially in the number of electron orbitals considered. Several quantum algorithms have been proposed for estimating ground state energies. Recently, the variational quantum eigensolver has attracted significant attention from the quantum computing community because of its prospects for implementation on near-term devices.

**[0095]** In many quantum chemistry and materials settings, the system of interest is described by the general fermionic two-body Hamiltonian as previously described. During the course of a VQE implementation, which attempts to determine the ground state energy of this Hamiltonian, the Hamiltonian can be adapted so as to systematically lower the ground state energy estimates. During the marginals optimization procedure, the following optimization problem may be solved:

$$\min_{R} Tr(SRVR^{\dagger}) + Tr(D(R \otimes R)W(R^{\dagger} \otimes R^{\dagger}))$$

$$\text{subject to } RR^{\dagger} = I \text{ and } R^{\dagger}R = I$$

**[0096]** Blackbox non-linear programming techniques, such as MATLAB's built-in fmincon function or the scipy-optimize optimization module can be used to carry out this optimization. We describe a more theoretically-motivated alternative with performance guarantees that is based on semidefinite programming relaxation methods.

**[0097]** In typical quantum chemistry problems, many of the two-body fermionic Hamiltonian coefficients, or integrals, will be zero. For example, if $a_i$ and $a_j$ correspond to states with different spin, then $S^{ij}$ will be zero and the corresponding marginal $V_{ij}$ need not be measured. Restricting R to be block diagonal, with one block corresponding to spin up orbitals and another block corresponding to spin down orbitals, will prevent the rotation from affecting terms in the Hamiltonian that are zero due to spin considerations.

**[0098]** In some cases, it may be advantageous to consider only orbital rotations that have the same effect on both spin-up and spin-down orbitals (i.e., the two blocks of R are identical). For example, such a restriction could be useful for constraining the variational minimization to closed-shell wavefunctions (i.e., states that are invariant with respect to spin rotations). Provided that the state $\rho(\vec{\theta}^{*})$ is a closed-shell state and that the spatial orbitals corresponding to the spin-up and spin-down spin orbitals are identical, restricting the rotation to have the same effect on spin-up and spin-down orbitals ensures that the transformed state is closed-shell and reduces the number of parameters that must be optimized via MOP. It may also be advantageous to restrict the blocks of R to be identical for some open-shell cases. For example, restricting R may be required to ensure that the transformed state is an eigenstate of the total spin operator. Such constraints manifest as homogeneous linear constraints on R, which will translate into homogeneous linear constraints on $R \otimes R^{\dagger}$ that can be easily incorporated into the semidefinite programming method as described below.

**[0099]** We introduce an optimization method which is tailored to the fermionic Hamiltonian. We adapt the previous optimization problem into a more-tractable form. First, we reshape the unitaries R into $N^2$ dimensional vectors $|R\rangle$, allowing us to rewrite the minimization problem as

$$\min_{|R\rangle} \langle R \mid V \otimes S^{T} \mid R\rangle + \langle RR \mid (I \otimes X_{23} \otimes I)W \otimes D^{T}(I \otimes X_{23} \otimes I) \mid RR\rangle$$

$$\text{subject to}$$

$$\langle R \mid (I \otimes |i\rangle\langle j|)|R\rangle = \delta_{ij} \text{ and } \langle R \mid (|i\rangle\langle j| \otimes I) \mid R\rangle = \delta_{ij}$$

where $X_{23}$ is the swap operator on the second and third systems. Next, noting that $\langle R|V \otimes S^{T}|R\rangle = \langle RR|V \otimes S^{T} \otimes I \otimes I|RR\rangle$, and defining

$$M = V \otimes S^{T} \otimes I \otimes I + (ity \otimes X_{23} \otimes I)W \otimes D^{T}(I \otimes X_{23} \otimes I),$$

we simplify the optimization problem to

$$\min_{|R\rangle} \langle RR \mid M \mid RR\rangle$$

$$\text{subject to}$$

$$\langle R \mid (I \otimes |i\rangle\langle j|) \mid R\rangle = \delta_{ij} \text{ and } \langle R \mid (|i\rangle\langle j| \otimes I)|R\rangle = \delta_{ij}$$

**[0100]** We introduce the self-consistency approach, where we iteratively solve restricted versions of the above optimization problem. To summarize, we fix one of the rotation matrices, optimize with respect to the other, and then, in the next iteration, set the fixed rotation matrix to the optimal rotation matrix of the previous iteration. In this approach,

the optimization of each iteration may be carried out using the following semidefinite relaxation technique. Defining $M_i=(I\otimes\langle R_i|)M(I\otimes|R_i\rangle)$, where $R_i$ is the fixed rotation, the optimization problem becomes,

$$\min_{|R\rangle}\langle R|M_i|R\rangle$$

$$\text{subject to } \langle R|(I\otimes|i\rangle\langle j|)|R\rangle=\delta_{ij} \text{ and}$$

$$\langle R|(|i\rangle\langle j|\otimes I)|R\rangle=\delta_{ij}$$

**[0101]** The above minimization problem has the form of a quadratically-constrained quadratic programming problem. Such problems are, in general, NP-hard. A standard technique for obtaining approximate solutions is known as semidefinite relaxation. We propose a type of semidefinite relaxation method for generating approximate solutions to the above optimization problem. The first step is to express the optimization problem as nearly a semidefinite programming problem:

$$\min_{\rho} Tr(M_i\rho)$$

$$\text{subject to } Tr((I\otimes|i\rangle\langle j|)\rho)=\delta_{ij} \text{ and}$$

$$Tr((|i\rangle\langle j|\otimes I)\rho)=\delta_{ij}$$

$$\rho \text{ is rank}-\text{one}$$

**[0102]** The problem is not a semidefinite program, however, because of the non-convex rank-one constraint. The semidefinite relaxation method then relaxes the rank-one constraint and solves the resulting semidefinite program, which only takes polynomial-time in the dimension of ρ. The optimal value ρ* is then used to generate rank-one candidate solutions, of which the optimal one is taken as the approximate solution. The candidates $|K\rangle$ are obtained by sampling from the multivariate normal distribution $\mathcal{N}(0,\rho^*)$, where ρ* serves as the covariance matrix. The sampled vectors $|K\rangle$ do not necessarily satisfy the quadratic constraints of unitarity given in the minimization problems above, and are likely infeasible. To generate feasible candidates these samples must be rounded, which is a standard technique in semidefinite relaxation methods. We choose to round each sample $|K\rangle$ to the closest unitary in distance given by the Frobenius norm. If $K=USV^\dagger$ is the singular value decomposition of K, the closest unitary to K is given by $W=UV^\dagger$. Thus, we round $|K\rangle$ by vectorizing to obtain matrix K, computing the singular value decomposition $K=USV^\dagger$, and then taking the rounded $R^*=UV_\dagger$ to be the candidate rotation. By generating many samples and choosing the best one, we boost the chances of generating a good candidate rotation. From the best sample, we obtain our approximate solution to the optimization problem and plug this in to the fixed rotation to the next round of iteration: $R_{i+1}=R^*$. This procedure is continued until the improvement in energy from one round to the next falls below some threshold.

**[0103]** We also introduce a direct semidefinite relaxation approach. For any fermionic Hamiltonian, we may implement the marginal optimization procedure using a more general group of unitary transformations than just orbital rotations. This group is known as the Bogoliubov transformations. Such transformations map linear combinations of fermionic creation to linear combinations of creation and annihilation operators. Bogoliubov transformations have a simple characterization in terms of Majorana operators $\gamma_{2i}=(a_i+a_i^\dagger)/t2$ and $\gamma_{2i+1}=(a_i-a_i^\dagger)/\sqrt{2}i$. These operators satisfy the elegant, single commutation relation $\{\gamma_\nu,\gamma_\mu\}=\delta_{\nu\mu}I$. $U_S=\Sigma_\mu S_\nu{}^\mu\gamma_\mu$, where S is a real orthogonal transformation. The group of Bogoliubov transformations are simply the real rotations of the 2N vectors $\gamma_i$. This group contains, as a subgroup, the orbital rotation group SU(N) that was considered previously. Let $U_S$ be the unitary representation of a Majorana rotation and let $S_\mu{}^\nu$ be the matrix entries of the corresponding rotation in $\mathbb{R}^{2N}$, then each Majorana operator transforms as $U_S^\dagger\gamma_\nu U_S=\Sigma_\mu S_\nu{}^\mu\gamma_\mu$. Such transformations alter the Hamiltonian as follows. In terms of Majorana operators, the original Hamiltonian is

$$H=\sum_{\nu,\mu}\Sigma_{\nu\mu}\gamma_\nu\gamma_\mu+\sum_{\nu,\mu,\alpha,\beta}\Delta_{\nu\mu\alpha\beta}\gamma_\nu\gamma_\mu\gamma_\alpha\gamma_\beta,$$

**[0104]** where $\Sigma_{\nu\mu}$ and $\Delta_{\nu\mu\alpha\beta}$ are obtained from $S_{\nu\mu}$ and $D_{\nu\mu\alpha\beta}$. Making the replacement, $U_S^\dagger\gamma_\nu U_S=\Sigma_\mu S_\nu{}^\mu\gamma_\mu$, we write the transformed Hamiltonian as

$$\begin{aligned}H(S)&=U_S^\dagger HU_S\\&=\sum_{\nu,\mu}\Sigma_{\nu\mu}\sum_{n,m}S_\nu^nS_\mu^m\gamma_n\gamma_m+\\&\sum_{\nu,\mu,\alpha,\beta}\sum_{n,m,a,b}\Delta_{\nu\mu\alpha\beta}S_\nu^nS_\mu^mS_\alpha^aS_\beta^b\\&\gamma_n\gamma_m\gamma_a\gamma_b.\end{aligned}$$

**[0105]** The marginals which may be computed for the marginals optimization procedure are of the form,

$$\Lambda_{nm}=Tr(\gamma_n\gamma_m\rho(\vec{\theta^*}))$$

$$\Omega_{nmab}=Tr(\gamma_n\gamma_m\gamma_a\gamma_b\rho(\vec{\theta^*})).$$

**[0106]** Through either the Jordan-Wigner or Bravyi-Kitaev transformations, which map fermionic operators to qubit operators, the Majorana operators may be mapped to Pauli product operators. In computing the reduced density matrices $Tr(a_i^\dagger a_j^\dagger a_k a_l\rho(\vec{\theta^*}))$ for a standard VQE problem, one may compute the expectations of the Pauli product operators which comprise the creation and annihilation operators. These Pauli product operators are precisely products of the Majorana operators. Consequently, in performing the marginals optimization procedure over Bogoliubov transformations, the relevant marginals $\Lambda_{nm}$ and $\Omega_{nmab}$, are simply those Pauli product expectation values which have been already computed. In terms of these Majorana operator marginals, we can write the Bogoliubov-transformed energy expectation value as

$$\langle H(S)\rangle=Tr(\Sigma S\Lambda S^T)+Tr(\Delta(S\otimes S)\Omega(S^T\otimes S^T)),$$

where we have reshaped S,V and D,W into 2N-by-2N and $(2N)^2$-by-$(2N)^2$ matrices, respectively. Finally, by carrying

out the following optimization problem we can, in general, obtain an improved estimate of the ground state energy,

$$\min_{S} Tr(\Sigma S \Lambda S^T) + Tr(\Delta(S \otimes S)\Omega(S^T \otimes S^T))$$

$$\text{subject to } SS^T = I \text{ and } S^T S = I$$

**[0107]** We can approach this optimization problem using the tools introduced above for the case of orbital rotation. The essential difference is that the matrices have all real entries, leading to solving a real semidefinite program.

**[0108]** Another common Hamiltonian model of interest is the spin Hamiltonian. This is used to describe the behavior of certain magnetic materials. In the case of spin-½ systems, the two-body Hamiltonian takes the general form

$$H = \sum_{i,\mu} S_{i,\mu} \sigma_i^\mu + \sum_{i,\mu} D_{i,j,\mu,\nu} \sigma_i^\mu \otimes \sigma_j^\nu,$$

where $\sigma_i^\mu$ of represent the X, Y, and Z, Pauli operators on the ith qubit. While the orbital rotations preserve the form of the Hamiltonian in the fermionic case, local unitary rotations $U_R = R_1 \otimes \ldots \hat{\times} R_N$ preserve the form of the spin Hamiltonian,

$$U_R^\dagger H U_R = \sum_{i,\mu} S_{i,\mu} U_R^\dagger \sigma_i^\mu U_R + \sum_{i,\mu} D_{i,j,\mu,\nu} U_R^\dagger \sigma_i^\mu \otimes \sigma_j^\nu U_R$$

$$= \sum_{i,\mu} S_{i,\mu} R_i^\dagger \sigma_i^\mu R_i + \sum_{i,j,\mu,\nu} D_{i,j,\mu,\nu} R_i^\dagger \sigma_i^\mu R_i \otimes R_j^\dagger \sigma_j^\nu R_j.$$

**[0109]** Using the fact that the adjoint representation $R \otimes R^\dagger$ of the SU(2) transformations are SO(3) transformations O, we write

$$U_R^\dagger H U_R = \sum_{i,\mu,m} S_{i,\mu} O_{m,i}^\mu \sigma_i^m + \sum_{i,j,\mu,\nu,m,n} D_{i,j,\mu,\nu} O_{m,i}^\mu O_{n,j}^\nu \sigma_i^m \otimes \sigma_j^n,$$

where the operators $O_i$ are determinant one, orthogonal, three-by-three matrices (i.e. SO(3)). Let $L_i^m = \mathrm{Tr}(\sigma_i^m \rho(\vec{\theta}))$ and $M_{ij}^{mn} = \mathrm{Tr}(\sigma_i^m \otimes \sigma_j^n \rho(\vec{\theta}))$ be the one- and two-body RDMs, respectively, of different Pauli products. Then the local-transformed energy expectation value is

$$\langle H(R) \rangle = \sum_{i,\mu,m} S_{i,\mu} O_{m,i}^\mu L_i^m + \sum_{i,j,\mu,\nu,m,n} D_{i,j,\mu,\nu} O_{m,i}^\mu O_{n,j}^\nu M_{ij}^{mn}$$

$$= \sum_i \vec{S}_i^T O_i \vec{L}_i + \sum_{i,j} Tr(D_{ij} O_i M_{ij} O_j^T).$$

**[0110]** Many combinatorial optimization problems can be recast as an Ising Hamiltonian energy minimization problem. The quantum approximate optimization algorithm [?] was proposed for generating approximate solutions to such optimization problems, in particular, focusing on the graph-theoretic problem of MAXCUT. The Hamiltonian considered in the QAOA algorithm is a standard classical Ising Hamiltonian

$$H = \sum_{\langle i,j \rangle \in E} Z_i \otimes Z_j,$$

where E is the set of edges of the graph defining the MAXCUT problem. This Hamiltonian is a specific instance of the spin Hamiltonians considered above.

**[0111]** FIG. **4** is a flow chart of a quantum optimization method **400**. Method **400** may be performed on either a classical computer, or a hybrid quantum-classical computer. Method **400** starts at a block **402**. In a block **404**, an expectation value of a Hamiltonian is estimated for a quantum state. The Hamiltonian is expressed as a linear combination of observables, and the Hamiltonian is estimated based on expectation values of the observables. In a block **412**, one or both of the Hamiltonian and the quantum state are transformed to reduce the expectation value of the Hamiltonian. In one embodiment, the expectation value of the Hamiltonian is minimized in block **412**. The minimization may be implemented with semidefinite programming techniques.

**[0112]** In some embodiments, method **400** includes a block **406** in which the expectation value of each of the observables is measured on a quantum computer. Block **406** may contain sub-blocks **408** and **410**. In sub-block **408**, the quantum state is generated on the quantum computer. In sub-block **410**, an observable is measured with the quantum state to obtain the expectation value of the observable. For any one observable, blocks **408** and **410** may be repeated to obtain sufficient statistics of the measurements to accurately determine the expectation value of the one observable. Blocks **408** and **410** may also be repeated for all the observables so that all the expectation values are obtained via measurements on the quantum computer.

**[0113]** In some embodiments, method **400** includes a block **414** in which the Hamiltonian is updated based on the transforming in block **412**. Block **414** may be implemented on a classical computer. For example, in block **412** a transformation (e.g., a unitary transformation) may be identified that, when applied either to the Hamiltonian or the quantum state, lowers the estimated energy (i.e., the expectation value of the Hamiltonian). In block **414**, the Hamiltonian is updated by applying the identified transformation to the Hamiltonian to generate an updated Hamiltonian. The quantum state better approximates the ground state of the updated Hamiltonian, as compared to the Hamiltonian prior to updating.

**[0114]** In some embodiments, a parametrized quantum circuit, programmable via one or more circuit parameters, is used in sub-block **408** to generate the quantum state. In some of these embodiments, method **400** further includes a block **416** in which the circuit parameters are updated so that the parametrized quantum circuit outputs an updated quantum state that better approximates the ground state of the Ham-

iltonian (either before or after transforming in block **412**). Block **416** may be implemented on a classical computer. For example, a classical optimization algorithm may be used to select the new circuit parameters to minimize a cost function that quantifies a distance between the quantum state and a target state (e.g., a ground state). The cost function may be based on the Hamiltonian prior to updating in block **416**, or on the updated Hamiltonian generated in block **416**.

**[0115]** In some embodiments, method **400** includes a decision **418** that checks for convergence of the circuit parameters. If, in decision **418**, the circuit parameters are updated by an amount that is below a threshold, then the circuit parameters have converged and method **400** ends at block **420**. If, in decision **418**, the circuit parameters are updated by an amount that is above the threshold, then the circuit parameters have not converged and method **400** repeats blocks **406**, **412**, **414**, and **416** to obtain a better approximation of the ground state and the corresponding ground-state energy. In sub-block **408**, the parameterized quantum circuit receives the updated circuit parameters determined in block **416** to generate the updated quantum state. Blocks **406**, **412**, **414**, and **416** may continue to repeat until it is determined in decision **418** that the circuit parameters have converged.

**[0116]** In other embodiments, block **406** (including sub-blocks **408** and **410**) may be implemented on a classical computer rather than a quantum computer. In these embodiments, all of method **400** is implemented on the classical computer to simulate operation of the quantum computer. More specifically, in block **406**, the expectation value of each of the observables may be determined on the classical computer. For example, the expectation value may be calculated deterministically via an equation or deterministic model. Alternatively, the expectation value may be determined stochastically (e.g., to simulate the randomness inherent to measurements performed on the quantum computer).

**[0117]** In some of the embodiments where all of method **400** is implemented on a classical computer, the quantum state may be represented on the classical computer as a first representation, in which case the first representation of the quantum state may be updated in block **416** instead of the circuit parameters. Blocks **406**, **412**, **414**, and **416** may then be repeated until the first representation of the quantum state converges.

**[0118]** In some embodiments of method **400**, the linear combination of the observables includes at least one observable with a zero weight that becomes non-zero when the Hamiltonian is transformed in block **412**. In these embodiments, the expectation value of an observable with a zero weight will not contribute to the expectation value of the Hamiltonian. However, after the Hamiltonian is transformed in block **414**, the zero weight for the observable may become non-zero, in which the expectation value of the observable will contribute to the expectation value of the Hamiltonian. Thus, in these embodiments, the expectation values of the observables include an expectation value for the at least one observable with a zero weight.

**[0119]** In other embodiments of method **400**, a fermionic transformation is applied, in block **412**, to one or both of the Hamiltonian and the quantum state. In some of these embodiments, the fermionic transformation include rotations of active orbitals. The fermionic transformation may include transformations out of an active space, of the active orbitals, to incorporate at least one of a core orbital and a virtual orbital. The fermionic transformation may include rotations that respect one or more of an open-shell spin symmetry, a closed-shell spin symmetry, and a geometric symmetry. In some of these embodiments, method **400** is implemented with a quantum subspace expansion technique, or a marginal projection technique, as described above. In other of these embodiments, the expectation values of the observables are obtained via orbital frames.

**[0120]** In other embodiments of method **400**, a Majorana fermionic transformation is applied, in block **412**, to one or both of the Hamiltonian and the quantum state. In some of these embodiments, the expectation value of the Hamiltonian is minimized using a Givens parameterization. In other of these embodiments, the expectation value of the Hamiltonian is minimized using semidefinite programming.

**[0121]** In other embodiments of method **400**, a spin transformation is applied, in block **412**, to one or both of the Hamiltonian and the quantum state. In some of these embodiments, the expectation value of the Hamiltonian is minimized using semidefinite programming.

**[0122]** In other embodiments of method **400**, the Hamiltonian is an Ising Hamiltonian configured for solving a combinatorial optimization problem. In some of these embodiments, the expectation value of the Hamiltonian is minimized using semidefinite programming.

**[0123]** It is to be understood that although the invention has been described above in terms of particular embodiments, the foregoing embodiments are provided as illustrative only, and do not limit or define the scope of the invention. Various other embodiments, including but not limited to the following, are also within the scope of the claims. For example, elements and components described herein may be further divided into additional components or joined together to form fewer components for performing the same functions.

**[0124]** Various physical embodiments of a quantum computer are suitable for use according to the present disclosure. In general, the fundamental data storage unit in quantum computing is the quantum bit, or qubit. The qubit is a quantum-computing analog of a classical digital computer system bit. A classical bit is considered to occupy, at any given point in time, one of two possible states corresponding to the binary digits (bits) 0 or 1. By contrast, a qubit is implemented in hardware by a physical medium with quantum-mechanical characteristics. Such a medium, which physically instantiates a qubit, may be referred to herein as a "physical instantiation of a qubit," a "physical embodiment of a qubit," a "medium embodying a qubit," or similar terms, or simply as a "qubit," for ease of explanation. It should be understood, therefore, that references herein to "qubits" within descriptions of embodiments of the present invention refer to physical media which embody qubits.

**[0125]** Each qubit has an infinite number of different potential quantum-mechanical states. When the state of a qubit is physically measured, the measurement produces one of two different basis states resolved from the state of the qubit. Thus, a single qubit can represent a one, a zero, or any quantum superposition of those two qubit states; a pair of qubits can be in any quantum superposition of 4 orthogonal basis states; and three qubits can be in any superposition of 8 orthogonal basis states. The function that defines the quantum-mechanical states of a qubit is known as its wavefunction. The wavefunction also specifies the probability distribution of outcomes for a given measurement. A qubit,

which has a quantum state of dimension two (i.e., has two orthogonal basis states), may be generalized to a d-dimensional "qudit," where d may be any integral value, such as 2, 3, 4, or higher. In the general case of a qudit, measurement of the qudit produces one of d different basis states resolved from the state of the qudit. Any reference herein to a qubit should be understood to refer more generally to an d-dimensional qudit with any value of d.

**[0126]** Although certain descriptions of qubits herein may describe such qubits in terms of their mathematical properties, each such qubit may be implemented in a physical medium in any of a variety of different ways. Examples of such physical media include superconducting material, trapped ions, photons, optical cavities, individual electrons trapped within quantum dots, point defects in solids (e.g., phosphorus donors in silicon or nitrogen-vacancy centers in diamond), molecules (e.g., alanine, vanadium complexes), or aggregations of any of the foregoing that exhibit qubit behavior, that is, comprising quantum states and transitions therebetween that can be controllably induced or detected.

**[0127]** For any given medium that implements a qubit, any of a variety of properties of that medium may be chosen to implement the qubit. For example, if electrons are chosen to implement qubits, then the x component of its spin degree of freedom may be chosen as the property of such electrons to represent the states of such qubits. Alternatively, the y component, or the z component of the spin degree of freedom may be chosen as the property of such electrons to represent the state of such qubits. This is merely a specific example of the general feature that for any physical medium that is chosen to implement qubits, there may be multiple physical degrees of freedom (e.g., the x, y, and z components in the electron spin example) that may be chosen to represent 0 and 1. For any particular degree of freedom, the physical medium may controllably be put in a state of superposition, and measurements may then be taken in the chosen degree of freedom to obtain readouts of qubit values.

**[0128]** Certain implementations of quantum computers, referred as gate model quantum computers, comprise quantum gates. In contrast to classical gates, there is an infinite number of possible single-qubit quantum gates that change the state vector of a qubit. Changing the state of a qubit state vector typically is referred to as a single-qubit rotation, and may also be referred to herein as a state change or a single-qubit quantum-gate operation. A rotation, state change, or single-qubit quantum-gate operation may be represented mathematically by a unitary 2×2 matrix with complex elements. A rotation corresponds to a rotation of a qubit state within its Hilbert space, which may be conceptualized as a rotation of the Bloch sphere. (As is well-known to those having ordinary skill in the art, the Bloch sphere is a geometrical representation of the space of pure states of a qubit.) Multi-qubit gates alter the quantum state of a set of qubits. For example, two-qubit gates rotate the state of two qubits as a rotation in the four-dimensional Hilbert space of the two qubits. (As is well-known to those having ordinary skill in the art, a Hilbert space is an abstract vector space possessing the structure of an inner product that allows length and angle to be measured. Furthermore, Hilbert spaces are complete: there are enough limits in the space to allow the techniques of calculus to be used.)

**[0129]** A quantum circuit may be specified as a sequence of quantum gates. As described in more detail below, the term "quantum gate," as used herein, refers to the application of a gate control signal (defined below) to one or more qubits to cause those qubits to undergo certain physical transformations and thereby to implement a logical gate operation. To conceptualize a quantum circuit, the matrices corresponding to the component quantum gates may be multiplied together in the order specified by the gate sequence to produce a $2^n \times 2^n$ complex matrix representing the same overall state change on n qubits. A quantum circuit may thus be expressed as a single resultant operator. However, designing a quantum circuit in terms of constituent gates allows the design to conform to a standard set of gates, and thus enable greater ease of deployment. A quantum circuit thus corresponds to a design for actions taken upon the physical components of a quantum computer.

**[0130]** A given variational quantum circuit may be parameterized in a suitable device-specific manner. More generally, the quantum gates making up a quantum circuit may have an associated plurality of tuning parameters. For example, in embodiments based on optical switching, tuning parameters may correspond to the angles of individual optical elements.

**[0131]** In certain embodiments of quantum circuits, the quantum circuit includes both one or more gates and one or more measurement operations. Quantum computers implemented using such quantum circuits are referred to herein as implementing "measurement feedback." For example, a quantum computer implementing measurement feedback may execute the gates in a quantum circuit and then measure only a subset (i.e., fewer than all) of the qubits in the quantum computer, and then decide which gate(s) to execute next based on the outcome(s) of the measurement(s). In particular, the measurement(s) may indicate a degree of error in the gate operation(s), and the quantum computer may decide which gate(s) to execute next based on the degree of error. The quantum computer may then execute the gate(s) indicated by the decision. This process of executing gates, measuring a subset of the qubits, and then deciding which gate(s) to execute next may be repeated any number of times. Measurement feedback may be useful for performing quantum error correction, but is not limited to use in performing quantum error correction. For every quantum circuit, there is an error-corrected implementation of the circuit with or without measurement feedback.

**[0132]** Some embodiments described herein generate, measure, or utilize quantum states that approximate a target quantum state (e.g., a ground state of a Hamiltonian). As will be appreciated by those trained in the art, there are many ways to quantify how well a first quantum state "approximates" a second quantum state. In the following description, any concept or definition of approximation known in the art may be used without departing from the scope hereof. For example, when the first and second quantum states are represented as first and second vectors, respectively, the first quantum state approximates the second quantum state when an inner product between the first and second vectors (called the "fidelity" between the two quantum states) is greater than a predefined amount (typically labeled E). In this example, the fidelity quantifies how "close" or "similar" the first and second quantum states are to each other. The fidelity represents a probability that a measurement of the first quantum state will give the same result as if the measurement were performed on the second quantum state. Proximity between quantum states can also be quantified with a distance measure, such as a Euclidean norm, a Hamming

distance, or another type of norm known in the art. Proximity between quantum states can also be defined in computational terms. For example, the first quantum state approximates the second quantum state when a polynomial time-sampling of the first quantum state gives some desired information or property that it shares with the second quantum state.

**[0133]** Not all quantum computers are gate model quantum computers. Embodiments of the present invention are not limited to being implemented using gate model quantum computers. As an alternative example, embodiments of the present invention may be implemented, in whole or in part, using a quantum computer that is implemented using a quantum annealing architecture, which is an alternative to the gate model quantum computing architecture. More specifically, quantum annealing (QA) is a metaheuristic for finding the global minimum of a given objective function over a given set of candidate solutions (candidate states), by a process using quantum fluctuations.

**[0134]** FIG. **2**B shows a diagram illustrating operations typically performed by a computer system **250** which implements quantum annealing. The system **250** includes both a quantum computer **252** and a classical computer **254**. Operations shown on the left of the dashed vertical line **256** typically are performed by the quantum computer **252**, while operations shown on the right of the dashed vertical line **256** typically are performed by the classical computer **254**.

**[0135]** Quantum annealing starts with the classical computer **254** generating an initial Hamiltonian **260** and a final Hamiltonian **262** based on a computational problem **258** to be solved, and providing the initial Hamiltonian **260**, the final Hamiltonian **262** and an annealing schedule **270** as input to the quantum computer **252**. The quantum computer **252** prepares a well-known initial state **266** (FIG. **2**B, operation **264**), such as a quantum-mechanical superposition of all possible states (candidate states) with equal weights, based on the initial Hamiltonian **260**. The classical computer **254** provides the initial Hamiltonian **260**, a final Hamiltonian **262**, and an annealing schedule **270** to the quantum computer **252**. The quantum computer **252** starts in the initial state **266**, and evolves its state according to the annealing schedule **270** following the time-dependent Schrödinger equation, a natural quantum-mechanical evolution of physical systems (FIG. **2**B, operation **268**). More specifically, the state of the quantum computer **252** undergoes time evolution under a time-dependent Hamiltonian, which starts from the initial Hamiltonian **260** and terminates at the final Hamiltonian **262**. If the rate of change of the system Hamiltonian is slow enough, the system stays close to the ground state of the instantaneous Hamiltonian. If the rate of change of the system Hamiltonian is accelerated, the system may leave the ground state temporarily but produce a higher likelihood of concluding in the ground state of the final problem Hamiltonian, i.e., diabatic quantum computation. At the end of the time evolution, the set of qubits on the quantum annealer is in a final state **272**, which is expected to be close to the ground state of the classical Ising model that corresponds to the solution to the original optimization problem **258**. An experimental demonstration of the success of quantum annealing for random magnets was reported immediately after the initial theoretical proposal.

**[0136]** The final state **272** of the quantum computer **254** is measured, thereby producing results **276** (i.e., measurements) (FIG. **2**B, operation **274**). The measurement operation **274** may be performed, for example, in any of the ways disclosed herein, such as in any of the ways disclosed herein in connection with the measurement unit **110** in FIG. **1**. The classical computer **254** performs postprocessing on the measurement results **276** to produce output **280** representing a solution to the original computational problem **258** (FIG. **2**B, operation **278**).

**[0137]** As yet another alternative example, embodiments of the present invention may be implemented, in whole or in part, using a quantum computer that is implemented using a one-way quantum computing architecture, also referred to as a measurement-based quantum computing architecture, which is another alternative to the gate model quantum computing architecture. More specifically, the one-way or measurement based quantum computer (MBQC) is a method of quantum computing that first prepares an entangled resource state, usually a cluster state or graph state, then performs single qubit measurements on it. It is "one-way" because the resource state is destroyed by the measurements.

**[0138]** The outcome of each individual measurement is random, but they are related in such a way that the computation always succeeds. In general, the choices of basis for later measurements need to depend on the results of earlier measurements, and hence the measurements cannot all be performed at the same time.

**[0139]** Any of the functions disclosed herein may be implemented using means for performing those functions. Such means include, but are not limited to, any of the components disclosed herein, such as the computer-related components described below.

**[0140]** Referring to FIG. **1**, a diagram is shown of a system **100** implemented according to one embodiment of the present invention. Referring to FIG. **2**A, a flowchart is shown of a method **200** performed by the system **100** of FIG. **1** according to one embodiment of the present invention. The system **100** includes a quantum computer **102**. The quantum computer **102** includes a plurality of qubits **104**, which may be implemented in any of the ways disclosed herein. There may be any number of qubits **104** in the quantum computer **104**. For example, the qubits **104** may include or consist of no more than 2 qubits, no more than 4 qubits, no more than 8 qubits, no more than 16 qubits, no more than 32 qubits, no more than 64 qubits, no more than 128 qubits, no more than 256 qubits, no more than 512 qubits, no more than 1024 qubits, no more than 2048 qubits, no more than 4096 qubits, or no more than 8192 qubits. These are merely examples, in practice there may be any number of qubits **104** in the quantum computer **102**.

**[0141]** There may be any number of gates in a quantum circuit. However, in some embodiments the number of gates may be at least proportional to the number of qubits **104** in the quantum computer **102**. In some embodiments the gate depth may be no greater than the number of qubits **104** in the quantum computer **102**, or no greater than some linear multiple of the number of qubits **104** in the quantum computer **102** (e.g., 2, 3, 4, 5, 6, or 7).

**[0142]** The qubits **104** may be interconnected in any graph pattern. For example, they be connected in a linear chain, a two-dimensional grid, an all-to-all connection, any combination thereof, or any subgraph of any of the preceding.

**[0143]** As will become clear from the description below, although element **102** is referred to herein as a "quantum computer," this does not imply that all components of the quantum computer **102** leverage quantum phenomena. One

or more components of the quantum computer **102** may, for example, be classical (i.e., non-quantum components) components which do not leverage quantum phenomena.

**[0144]** The quantum computer **102** includes a control unit **106**, which may include any of a variety of circuitry and/or other machinery for performing the functions disclosed herein. The control unit **106** may, for example, consist entirely of classical components. The control unit **106** generates and provides as output one or more control signals **108** to the qubits **104**. The control signals **108** may take any of a variety of forms, such as any kind of electromagnetic signals, such as electrical signals, magnetic signals, optical signals (e.g., laser pulses), or any combination thereof.

**[0145]** For example:

**[0146]** In embodiments in which some or all of the qubits **104** are implemented as photons (also referred to as a "quantum optical" implementation) that travel along waveguides, the control unit **106** may be a beam splitter (e.g., a heater or a mirror), the control signals **108** may be signals that control the heater or the rotation of the mirror, the measurement unit **110** may be a photodetector, and the measurement signals **112** may be photons.

**[0147]** In embodiments in which some or all of the qubits **104** are implemented as charge type qubits (e.g., transmon, X-mon, G-mon) or flux-type qubits (e.g., flux qubits, capacitively shunted flux qubits) (also referred to as a "circuit quantum electrodynamic" (circuit QED) implementation), the control unit **106** may be a bus resonator activated by a drive, the control signals **108** may be cavity modes, the measurement unit **110** may be a second resonator (e.g., a low-Q resonator), and the measurement signals **112** may be voltages measured from the second resonator using dispersive readout techniques.

**[0148]** In embodiments in which some or all of the qubits **104** are implemented as superconducting circuits, the control unit **106** may be a circuit QED-assisted control unit or a direct capacitive coupling control unit or an inductive capacitive coupling control unit, the control signals **108** may be cavity modes, the measurement unit **110** may be a second resonator (e.g., a low-Q resonator), and the measurement signals **112** may be voltages measured from the second resonator using dispersive readout techniques.

**[0149]** In embodiments in which some or all of the qubits **104** are implemented as trapped ions (e.g., electronic states of, e.g., magnesium ions), the control unit **106** may be a laser, the control signals **108** may be laser pulses, the measurement unit **110** may be a laser and either a CCD or a photodetector (e.g., a photomultiplier tube), and the measurement signals **112** may be photons.

**[0150]** In embodiments in which some or all of the qubits **104** are implemented using nuclear magnetic resonance (NMR) (in which case the qubits may be molecules, e.g., in liquid or solid form), the control unit **106** may be a radio frequency (RF) antenna, the control signals **108** may be RF fields emitted by the RF antenna, the measurement unit **110** may be another RF antenna, and the measurement signals **112** may be RF fields measured by the second RF antenna.

**[0151]** In embodiments in which some or all of the qubits **104** are implemented as nitrogen-vacancy centers (NV centers), the control unit **106** may, for example, be a laser, a microwave antenna, or a coil, the control signals **108** may be visible light, a microwave signal, or a constant electromagnetic field, the measurement unit **110** may be a photodetector, and the measurement signals **112** may be photons.

**[0152]** In embodiments in which some or all of the qubits **104** are implemented as two-dimensional quasiparticles called "anyons" (also referred to as a "topological quantum computer" implementation), the control unit **106** may be nanowires, the control signals **108** may be local electrical fields or microwave pulses, the measurement unit **110** may be superconducting circuits, and the measurement signals **112** may be voltages.

**[0153]** In embodiments in which some or all of the qubits **104** are implemented as semiconducting material (e.g., nanowires), the control unit **106** may be microfabricated gates, the control signals **108** may be RF or microwave signals, the measurement unit **110** may be microfabricated gates, and the measurement signals **112** may be RF or microwave signals.

**[0154]** Although not shown explicitly in FIG. **1** and not required, the measurement unit **110** may provide one or more feedback signals **114** to the control unit **106** based on the measurement signals **112**. For example, quantum computers referred to as "one-way quantum computers" or "measurement-based quantum computers" utilize such feedback **114** from the measurement unit **110** to the control unit **106**. Such feedback **114** is also necessary for the operation of fault-tolerant quantum computing and error correction.

**[0155]** The control signals **108** may, for example, include one or more state preparation signals which, when received by the qubits **104**, cause some or all of the qubits **104** to change their states. Such state preparation signals constitute a quantum circuit also referred to as an "ansatz circuit." The resulting state of the qubits **104** is referred to herein as an "initial state" or an "ansatz state." The process of outputting the state preparation signal(s) to cause the qubits **104** to be in their initial state is referred to herein as "state preparation" (FIG. **2**A, section **206**). A special case of state preparation is "initialization," also referred to as a "reset operation," in which the initial state is one in which some or all of the qubits **104** are in the "zero" state i.e. the default single-qubit state. More generally, state preparation may involve using the state preparation signals to cause some or all of the qubits **104** to be in any distribution of desired states. In some embodiments, the control unit **106** may first perform initialization on the qubits **104** and then perform preparation on the qubits **104**, by first outputting a first set of state preparation signals to initialize the qubits **104**, and by then outputting a second set of state preparation signals to put the qubits **104** partially or entirely into non-zero states.

**[0156]** Another example of control signals **108** that may be output by the control unit **106** and received by the qubits **104** are gate control signals. The control unit **106** may output such gate control signals, thereby applying one or more gates to the qubits **104**. Applying a gate to one or more qubits causes the set of qubits to undergo a physical state change which embodies a corresponding logical gate operation (e.g., single-qubit rotation, two-qubit entangling gate or multi-qubit operation) specified by the received gate control signal. As this implies, in response to receiving the gate control signals, the qubits **104** undergo physical transformations which cause the qubits **104** to change state in such

a way that the states of the qubits **104**, when measured (see below), represent the results of performing logical gate operations specified by the gate control signals. The term "quantum gate," as used herein, refers to the application of a gate control signal to one or more qubits to cause those qubits to undergo the physical transformations described above and thereby to implement a logical gate operation.

**[0157]** It should be understood that the dividing line between state preparation (and the corresponding state preparation signals) and the application of gates (and the corresponding gate control signals) may be chosen arbitrarily. For example, some or all the components and operations that are illustrated in FIGS. W and X as elements of "state preparation" may instead be characterized as elements of gate application. Conversely, for example, some or all of the components and operations that are illustrated in FIGS. W and X as elements of "gate application" may instead be characterized as elements of state preparation. As one particular example, the system and method of FIGS. W and X may be characterized as solely performing state preparation followed by measurement, without any gate application, where the elements that are described herein as being part of gate application are instead considered to be part of state preparation. Conversely, for example, the system and method of FIGS. W and X may be characterized as solely performing gate application followed by measurement, without any state preparation, and where the elements that are described herein as being part of state preparation are instead considered to be part of gate application.

**[0158]** The quantum computer **102** also includes a measurement unit **110**, which performs one or more measurement operations on the qubits **104** to read out measurement signals **112** (also referred to herein as "measurement results") from the qubits **104**, where the measurement results **112** are signals representing the states of some or all of the qubits **104**. In practice, the control unit **106** and the measurement unit **110** may be entirely distinct from each other, or contain some components in common with each other, or be implemented using a single unit (i.e., a single unit may implement both the control unit **106** and the measurement unit **110**). For example, a laser unit may be used both to generate the control signals **108** and to provide stimulus (e.g., one or more laser beams) to the qubits **104** to cause the measurement signals **112** to be generated.

**[0159]** In general, the quantum computer **102** may perform various operations described above any number of times. For example, the control unit **106** may generate one or more control signals **108**, thereby causing the qubits **104** to perform one or more quantum gate operations. The measurement unit **110** may then perform one or more measurement operations on the qubits **104** to read out a set of one or more measurement signals **112**. The measurement unit **110** may repeat such measurement operations on the qubits **104** before the control unit **106** generates additional control signals **108**, thereby causing the measurement unit **110** to read out additional measurement signals **112** resulting from the same gate operations that were performed before reading out the previous measurement signals **112**. The measurement unit **110** may repeat this process any number of times to generate any number of measurement signals **112** corresponding to the same gate operations. The quantum computer **102** may then aggregate such multiple measurements of the same gate operations in any of a variety of ways.

**[0160]** After the measurement unit **110** has performed one or more measurement operations on the qubits **104** after they have performed one set of gate operations, the control unit **106** may generate one or more additional control signals **108**, which may differ from the previous control signals **108**, thereby causing the qubits **104** to perform one or more additional quantum gate operations, which may differ from the previous set of quantum gate operations. The process described above may then be repeated, with the measurement unit **110** performing one or more measurement operations on the qubits **104** in their new states (resulting from the most recently-performed gate operations).

**[0161]** In general, the system **100** may implement a plurality of quantum circuits as follows. For each quantum circuit C in the plurality of quantum circuits (FIG. **2**A, operation **202**), the system **100** performs a plurality of "shots" on the qubits **104**. The meaning of a shot will become clear from the description that follows. For each shot S in the plurality of shots (FIG. **2**A, operation **204**), the system **100** prepares the state of the qubits **104** (FIG. **2**A, section **206**). More specifically, for each quantum gate G in quantum circuit C (FIG. **2**A, operation **210**), the system **100** applies quantum gate G to the qubits **104** (FIG. **2**A, operations **212** and **214**).

**[0162]** Then, for each of the qubits Q **104** (FIG. **2**A, operation **216**), the system **100** measures the qubit Q to produce measurement output representing a current state of qubit Q (FIG. **2**A, operations **218** and **220**).

**[0163]** The operations described above are repeated for each shot S (FIG. **2**A, operation **222**), and circuit C (FIG. **2**A, operation **224**). As the description above implies, a single "shot" involves preparing the state of the qubits **104** and applying all of the quantum gates in a circuit to the qubits **104** and then measuring the states of the qubits **104**; and the system **100** may perform multiple shots for one or more circuits.

**[0164]** Referring to FIG. **3**, a diagram is shown of a hybrid classical quantum computer (HQC) **300** implemented according to one embodiment of the present invention. The HQC **300** includes a quantum computer component **102** (which may, for example, be implemented in the manner shown and described in connection with FIG. **1**) and a classical computer component **306**. The classical computer component may be a machine implemented according to the general computing model established by John Von Neumann, in which programs are written in the form of ordered lists of instructions and stored within a classical (e.g., digital) memory **310** and executed by a classical (e.g., digital) processor **308** of the classical computer. The memory **310** is classical in the sense that it stores data in a storage medium in the form of bits, which have a single definite binary state at any point in time. The bits stored in the memory **310** may, for example, represent a computer program. The classical computer component **304** typically includes a bus **314**. The processor **308** may read bits from and write bits to the memory **310** over the bus **314**. For example, the processor **308** may read instructions from the computer program in the memory **310**, and may optionally receive input data **316** from a source external to the computer **302**, such as from a user input device such as a mouse, keyboard, or any other input device. The processor **308** may use instructions that have been read from the memory **310** to perform computations on data read from the memory **310** and/or the input **316**, and generate output from those instruc-

tions. The processor **308** may store that output back into the memory **310** and/or provide the output externally as output data **318** via an output device, such as a monitor, speaker, or network device.

**[0165]** The quantum computer component **102** may include a plurality of qubits **104**, as described above in connection with FIG. **1**. A single qubit may represent a one, a zero, or any quantum superposition of those two qubit states. The classical computer component **304** may provide classical state preparation signals Y**32** to the quantum computer **102**, in response to which the quantum computer **102** may prepare the states of the qubits **104** in any of the ways disclosed herein, such as in any of the ways disclosed in connection with FIGS. **1** and **2**A-**2**B.

**[0166]** Once the qubits **104** have been prepared, the classical processor **308** may provide classical control signals **334** to the quantum computer **102**, in response to which the quantum computer **102** may apply the gate operations specified by the control signals **332** to the qubits **104**, as a result of which the qubits **104** arrive at a final state. The measurement unit **110** in the quantum computer **102** (which may be implemented as described above in connection with FIGS. **1** and **2**A-**2**B) may measure the states of the qubits **104** and produce measurement output **338** representing the collapse of the states of the qubits **104** into one of their eigenstates. As a result, the measurement output **338** includes or consists of bits and therefore represents a classical state. The quantum computer **102** provides the measurement output **338** to the classical processor **308**. The classical processor **308** may store data representing the measurement output **338** and/or data derived therefrom in the classical memory **310**.

**[0167]** The steps described above may be repeated any number of times, with what is described above as the final state of the qubits **104** serving as the initial state of the next iteration. In this way, the classical computer **304** and the quantum computer **102** may cooperate as co-processors to perform joint computations as a single computer system.

**[0168]** Although certain functions may be described herein as being performed by a classical computer and other functions may be described herein as being performed by a quantum computer, these are merely examples and do not constitute limitations of the present invention. A subset of the functions which are disclosed herein as being performed by a quantum computer may instead be performed by a classical computer. For example, a classical computer may execute functionality for emulating a quantum computer and provide a subset of the functionality described herein, albeit with functionality limited by the exponential scaling of the simulation. Functions which are disclosed herein as being performed by a classical computer may instead be performed by a quantum computer.

**[0169]** The techniques described above may be implemented, for example, in hardware, in one or more computer programs tangibly stored on one or more computer-readable media, firmware, or any combination thereof, such as solely on a quantum computer, solely on a classical computer, or on a hybrid classical quantum (HQC) computer. The techniques disclosed herein may, for example, be implemented solely on a classical computer, in which the classical computer emulates the quantum computer functions disclosed herein.

**[0170]** The techniques described above may be implemented in one or more computer programs executing on (or executable by) a programmable computer (such as a classical computer, a quantum computer, or an HQC) including any combination of any number of the following: a processor, a storage medium readable and/or writable by the processor (including, for example, volatile and non-volatile memory and/or storage elements), an input device, and an output device. Program code may be applied to input entered using the input device to perform the functions described and to generate output using the output device.

**[0171]** Embodiments of the present invention include features which are only possible and/or feasible to implement with the use of one or more computers, computer processors, and/or other elements of a computer system. Such features are either impossible or impractical to implement mentally and/or manually. For example, in any practical use of embodiments of the present invention, carrying out the optimization of the energy expectation value will be computationally demanding and impossible to perform manually, or mentally. Even with a conservative estimate, millions of individual computational steps would be needed. Even solely using a classical computer, the routine is, in general, likely inefficient because generating a variety of valid marginal data is challenging due to the QMA-completeness of the quantum marginal problem.

**[0172]** Any claims herein which affirmatively require a computer, a processor, a memory, or similar computer-related elements, are intended to require such elements, and should not be interpreted as if such elements are not present in or required by such claims. Such claims are not intended, and should not be interpreted, to cover methods and/or systems which lack the recited computer-related elements. For example, any method claim herein which recites that the claimed method is performed by a computer, a processor, a memory, and/or similar computer-related element, is intended to, and should only be interpreted to, encompass methods which are performed by the recited computer-related element(s). Such a method claim should not be interpreted, for example, to encompass a method that is performed mentally or by hand (e.g., using pencil and paper). Similarly, any product claim herein which recites that the claimed product includes a computer, a processor, a memory, and/or similar computer-related element, is intended to, and should only be interpreted to, encompass products which include the recited computer-related element (s). Such a product claim should not be interpreted, for example, to encompass a product that does not include the recited computer-related element(s).

**[0173]** In embodiments in which a classical computing component executes a computer program providing any subset of the functionality within the scope of the claims below, the computer program may be implemented in any programming language, such as assembly language, machine language, a high-level procedural programming language, or an object-oriented programming language. The programming language may, for example, be a compiled or interpreted programming language.

**[0174]** Each such computer program may be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a computer processor, which may be either a classical processor or a quantum processor. Method steps of the invention may be performed by one or more computer processors executing a program tangibly embodied on a computer-readable medium to perform functions of the invention by operating on input and generating output. Suitable processors include, by way of example, both general and special purpose microproces-

sors. Generally, the processor receives (reads) instructions and data from a memory (such as a read-only memory and/or a random access memory) and writes (stores) instructions and data to the memory. Storage devices suitable for tangibly embodying computer program instructions and data include, for example, all forms of non-volatile memory, such as semiconductor memory devices, including EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROMs. Any of the foregoing may be supplemented by, or incorporated in, specially-designed ASICs (application-specific integrated circuits) or FPGAs (Field-Programmable Gate Arrays). A classical computer can generally also receive (read) programs and data from, and write (store) programs and data to, a non-transitory computer-readable storage medium such as an internal disk (not shown) or a removable disk. These elements will also be found in a conventional desktop or workstation computer as well as other computers suitable for executing computer programs implementing the methods described herein, which may be used in conjunction with any digital print engine or marking engine, display monitor, or other raster output device capable of producing color or gray scale pixels on paper, film, display screen, or other output medium.

**[0175]** Any data disclosed herein may be implemented, for example, in one or more data structures tangibly stored on a non-transitory computer-readable medium (such as a classical computer-readable medium, a quantum computer-readable medium, or an HQC computer-readable medium). Embodiments of the invention may store such data in such data structure(s) and read such data from such data structure(s).

What is claimed is:

**1**. A quantum optimization method, comprising:

estimating, on a classical computer and for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables; and

transforming, on the classical computer, one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian.

**2**. The quantum optimization method of claim **1**, further comprising measuring the expectation value of each of the observables on a quantum computer by:

generating the quantum state on the quantum computer; and

measuring, on the quantum computer, said each of the observables for the quantum state.

**3**. The quantum optimization method of claim **2**, wherein said generating the quantum state includes generating the quantum state with a parametrized quantum circuit programmable via one or more circuit parameters.

**4**. The quantum optimization method of claim **3**, further comprising updating the one or more circuit parameters such that the parametrized quantum circuit outputs an updated quantum state that better approximates a ground state of the Hamiltonian.

**5**. The quantum optimization method of claim **4**, further comprising repeating:

said generating the quantum state with the parametrized quantum circuit;

said measuring each of the observables for the quantum state;

said transforming one or both of the Hamiltonian and the quantum state;

updating the Hamiltonian based on said transforming; and

said updating the one or more circuit parameters;

until the one or more circuit parameters have converged.

**6**. The quantum optimization method of claim **1**, wherein said transforming one or both of the Hamiltonian and the quantum state includes applying a unitary transformation to said one or both of the Hamiltonian and the quantum state.

**7**. The quantum optimization method of claim **1**, further comprising generating, on the classical computer, the expectation value of each of the observables.

**8**. The quantum optimization method of claim **7**, further comprising updating, on the classical computer, a first representation of the quantum state based on the expectation value of the Hamiltonian to better approximate a ground state of the Hamiltonian.

**9**. The quantum optimization method of claim **8**, further comprising repeating:

said generating the expectation value of each of the observables;

said transforming one or both of the Hamiltonian and the quantum state; and

said updating the first representation of the quantum state;

until the first representation of the quantum state has converged.

**10**. The quantum optimization method of claim **1**, wherein:

the linear combination of the observables includes at least one observable with a zero weight that becomes non-zero due to said transforming the Hamiltonian; and

the expectation values of the observables include an expectation value for the at least one observable with a zero weight.

**11**. The quantum optimization method of claim **1**, wherein said transforming one or both of the Hamiltonian and the quantum state includes applying a fermionic transformation to said one or both of the Hamiltonian and the quantum state.

**12**. The quantum optimization method of claim **11**, the fermionic transformation including rotations of active orbitals.

**13**. The quantum optimization method of claim **11**, the fermionic transformation including transformations out of an active space to incorporate at least one of a core orbital and a virtual orbital.

**14**. The quantum optimization method of claim **11**, the fermionic transformation including rotations that respect one or more of an open-shell spin symmetry, a closed-shell spin symmetry, and a geometric symmetry.

**15**. The quantum optimization method of claim **11**, further comprising implementing a quantum subspace expansion technique.

**16**. The quantum optimization method of claim **11**, further comprising implementing a marginal projection technique.

**17**. The quantum optimization method of claim **11**, further comprising obtaining any of the expectation values the observables via orbital frames.

**18**. The quantum optimization method of claim **1**, wherein said transforming one or both of the Hamiltonian and the quantum state includes applying a Majorana fermionic transformation to said one or both of the Hamiltonian and the quantum state.

**19**. The quantum optimization method of claim **18**, further comprising minimizing the expectation value of the Hamiltonian using a Givens parameterization.

**20**. The quantum optimization method of claim **18**, further comprising minimizing the expectation value of the Hamiltonian using semidefinite programming.

**21**. The quantum optimization method of claim **1**, wherein said transforming one or both of the Hamiltonian and the quantum state includes applying a spin transformation to said one or both of the Hamiltonian and the quantum state.

**22**. The quantum optimization method of claim **1**, wherein the Hamiltonian is an Ising Hamiltonian configured for solving a combinatorial optimization problem.

**23**. The quantum optimization method of claim **1**, wherein said transforming one or both of the Hamiltonian and the quantum state includes minimizing the expectation value of the Hamiltonian estimated for the quantum state.

**24**. The quantum optimization method of claim **23**, wherein said minimizing the expectation value of the Hamiltonian includes minimizing the expectation value of the Hamiltonian using semidefinite programming.

**25**. A computing system configured for quantum optimization, comprising:

a processor;

a memory communicably coupled with the processor and storing machine-readable instructions that, when executed by the processor, control the computing system to:

estimate, for a quantum state, an expectation value of a Hamiltonian, expressible as a linear combination of observables, based on expectation values of the observables, and

transform one or both of the Hamiltonian and the quantum state to reduce the expectation value of the Hamiltonian estimated for the quantum state.

**26**. The computing system of claim **25**, further comprising a quantum computer that is communicably coupled with the processor and configured to measure the expectation value of each of the observables.

* * * * *