<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>JIN-MIN LIANG ET AL: &quot;Variational Quantum Algorithms for Dimensionality Reduction and Classification&quot;, ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 20 March 2020 (2020-03-20), XP081625166, DOI: 10.1103/PHYSREVA.101.032323 abstract; Appendix A; Sections I-IV</td>
<td>1-10</td>
</tr>
</tbody>
</table>
**INTERNATIONAL SEARCH REPORT**

**A. CLASSIFICATION OF SUBJECT MATTER**

INV. G06N10/00  
ADD. G06N20/00

According to International Patent Classification (IPC) or to both national classification and IPC

---

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

---

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>DANIEL J EGGER ET AL: &quot;Quantum computing for Finance: state of the art and future prospects&quot;, ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 25 June 2020 (2020-06-25), XP081707073, abstract; Sections 4, 6, and 7 ---- /--</td>
<td>1-10</td>
</tr>
</tbody>
</table>

---

Further documents are listed in the continuation of Box C. See patent family annex.

**Date of the actual completion of the international search**

3 November 2021

**Date of mailing of the international search report**

12/11/2021

**Name and mailing address of the ISA**

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV RIJSWIJK  
Tel. (+31-70) 340-5040,  
Fax (+31-70) 340-5016

Authorized officer

Klasen, TJ

---

Form PCT/ISA/210 (second sheet) (April 2005)
FIG. 2
FIG. 1
6.- A method according to any of previous claims, wherein the quantum processor (QP) comprises:
   - \( N \) qubits;
   - An inlet configured for setting each \( N \) qubits as \( |0\rangle \);
   - A phase-shift gate at each \( N \) qubit wherein the phase-shift value of the \( j^{th} \) qubits is \( W_j \);
   - At least one quantum gate (QG) connecting at least two qubits;
   - Means for observing the quantum state of the \( N \) qubits.

7.- A method according to claim 6, wherein the quantum processor (QP) comprises \( N - 1 \) CNOT quantum gates, sequentially connecting a qubit and the next qubit.

8.- A method according to claim 6, wherein the quantum processor (QP) comprises a plurality of controlled-z operator, each controlled-z operator connecting two qubits.

9.- A computer system comprising:
   - a quantum processor (QP) comprising at least \( N \) qubits and at least one quantum gate (QG) parameterizable with at least one parameter \( (W_1, W_2, ..., W_p) \);
   - a classical binary processor (CP) in communication with the quantum processor (QP),
   the computer system adapted to carry out a method according to steps a) to f) of claim 1.

10.- A computer system as claim 9 wherein further adapted to carry out a method according to any of claims 2 to 8.
set a value in the at least one parameter of the quantum processor (QP); and, retrieving an observable (O) of the quantum state of each of the at least \( N \) qubits of the quantum processor (QP).

3.- A method according to claim 2, wherein the variational quantum optimization method comprises:

selecting a variational ansatz in the quantum processor (QP) comprising the at least one quantum gate (QG) and having parameters \( W_1, W_2, ..., W_p \);

defining a second scalar function \( G \) wherein

\[
G(W_1, W_2, ..., W_p) = \langle \Psi(W_1, W_2, ..., W_p) | H(s_1, ..., s_N) | \Psi(W_1, W_2, ..., W_p) \rangle
\]

wherein \( \Psi(W_1, W_2, ..., W_p) \) is a trial function responsive of the variational ansatz of the quantum processor (QP) according to the set of parameters \( W_1, W_2, ..., W_p \);

executing a classical global optimizer in the classical binary processor (CP) to find parameters \( W_1, W_2, ..., W_p \) providing the best approximation to the minimum of the second scalar function \( G(W_1, W_2, ..., W_p) \);

providing \( (s_1^0, ..., s_N^0) \) as the observable (O) of the quantum processor (QP) providing the minimum of the objective function \( G(W_1, W_2, ..., W_p) \).

4.- A method according to claim 3, wherein the scalar value of the second scalar function \( G \) to be minimized in the classical global optimizer is obtained by executing \( K \) times, being \( K \) a predetermined positive integer number, the following steps:

sending instructions to the quantum computer to create the variational state with the at least one parameter \( W_1, W_2, ..., W_p \),

retrieving an observable (O) of the quantum state of each of the at least \( N \) qubits of the quantum processor (QP) and generating a sequence of binary numbers \( k_1, k_2, ..., k_N \) wherein \( k_j \) is 0 or 1 depending on the observed state of the corresponding \( j \)th qubit, \( |0 \rangle \) or \( |1 \rangle \) respectively;

computing in the classical binary processor (CP) an estimate \( F_k = H(k_1, k_2, ..., k_N) \);

returning the average value \( \left( \sum_{k=1}^{K} F_k \right) / K \).

5.- A method according to any of the preceding claims, wherein steps a) – f) are executed iteratively wherein in each iteration the scalar function \( F(x_1, x_2, ..., x_N) \) is the scalar function \( F_{new}(\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m) \) resulting from the previous step until the \( m \) value is less than a predetermined value.
CLAIMS

1. Method for optimizing a computing process in a classical binary processor (CP) wherein the computing process is optimized by reducing the number of parameters on which the process depends, using:

- a quantum processor (QP) comprising at least $N$ qubits and at least one quantum gate (QG) parameterizable with at least one parameter $(W_1, W_2, ..., W_p)$;
- a classical binary processor (CP) in communication with the quantum processor (QP),

wherein the method comprises the steps:

a) receiving in the classical binary processor (CP) an allocated scalar function $F$, the scalar function $F$ depending on $N$ parameters: $F(x_1, x_2, ..., x_N)$ and having at least a minimum value;

b) determining in the classical binary processor (CP) a set of values $(x_{1,0}^0, x_{2,0}^0, ..., x_{N,0}^0)$ minimizing the value of $F$;

c) defining an objective function $H(s_1, ..., s_N)$ depending on $s_i$, $i = 1, ..., N$, binary values taking the value 0 or 1, satisfying $H(s_1, ..., s_N) = F(s_1 x_{1,0}^0, s_2 x_{2,0}^0, ..., s_N x_{N,0}^0)$;

d) selecting $m < N$ and carrying out a variational quantum optimization method on the quantum processor (QP), being the objective function $H(s_1, ..., s_N)$ and being the binary variables $s_i$ under the constraints $\sum_{i=1}^{N} s_i = m$; and, providing $(s_{1,0}^0, ..., s_{N,0}^0)$ as the observable (O) of the quantum processor (QP) corresponding to the minimum of the objective function $H(s_1, ..., s_N)$, being the observable (O) $(s_{1,0}^0, ..., s_{N,0}^0)$ the direct result of a quantum measurement of the at least $N$ qubits;

e) allocating a new scalar function $F_{new}$ on the classical binary processor (CP) depending on $m$ parameters $x_{1,0}, x_{2,0}, ..., x_{m,0}$ wherein:

- the $m$ parameters $x_{1,0}, x_{2,0}, ..., x_{m,0}$ are selected among the $N$ parameters $x_1, x_2, ..., x_N$; that is, the $m$ indices $i_1, i_2, ..., i_m$ of $x_{i,j}, j = 1, ..., m$ are selected among those satisfying that $s_{ij}^0 = 1$, where $s_{ij}^0, i = 1, ..., N$ are the $N$ values provided by the variational quantum optimization method,

- $x_j = x_{i,j}, j = 1, ..., m$; and,

- $F_{new}(x_{1,0}, x_{2,0}, ..., x_{m,0}) = F(s_1 x_1, s_2 x_2, ..., s_N x_N)$;

f) returning the optimized new scalar function $F_{new}(x_{1,0}, x_{2,0}, ..., x_{m,0})$.

2. Method according to claim 1, wherein the classical binary processor (CP) is configured to:
Once the best bit-string $s^0$ has been obtained in the quantum circuit, it is used for generating the new module (MD) $F_{new}$ having a reduced number of arguments.

The classical binary processor (CP) allocates a new scalar the new scalar function as a module (MD) $F_{new}$ having $m = 5$ parameters.

Specifically, a new matrix $R_{new}$ is generated by selecting those columns of $R$ where $x$ has 1’s, similarly with the vector of capacities $u$, to create a new vector $u_{new}$ where the active components are given by $x$. Then the following problem is posed

$$
\arg\min_{x_{new}} R_{new}\|y - R_{new}\hat{x}\|^2
$$

$$
\hat{x}_j \geq 0, \forall j = 1, ..., m
$$

$$
\hat{x} \leq u_{new}
$$

The solution $\hat{x}$ gives the target vector of supplies $\hat{x}_j$. Therefore, the new module (MD) instantiating the scalar function $F_{new}$ has $m$ parameters parameters $\hat{x}_1, \hat{x}_2, ..., \hat{x}_m$ wherein:

1. the $m$ parameters $\hat{x}_1, \hat{x}_2, ..., \hat{x}_m$ are selected among the $N$ parameters $x_1, x_2, ..., x_N$; that is, the $m$ indices $i_1, i_2, ..., i_m$ of $\hat{x}_{i_j}$, $j = 1, ..., m$ are selected among those satisfying that $s_{i_j}^0 = 1$, where $s_{i_j}^0, i = 1, ..., N$ are the $N$ values provided by the variational quantum optimization method,

2. $\hat{x}_j = x_{i_j}$, $j = 1, ..., m$; and,

3. $F_{new}(\hat{x}_1, \hat{x}_2, ..., \hat{x}_m) = F(s_1 x_1, s_2 x_2, ..., s_N x_N)$.

The new allocated module (MD) instantiating the scalar function $F_{new}$ is represented in figure 1 as using less resources and having a lower number of parameters.
the value 0 or 1, satisfies \( H(s_1, ..., s_N) = F(s_1 x_1^0, s_2 x_2^0, ..., s_N x_N^0) \).

The lowest eigenvalue of \( H \) now encodes the global minimum of a second scalar function \( G \). With the help of the classical binary processor (CP) and a global optimization method, the parameters minimizing the expected value of

\[
G(W_1, W_2, ..., W_p) = \arg \min_W H(W) = \psi(W)^T H \psi(W)
\]

are determined which may also be expressed as:

\[
G(W_1, W_2, ..., W_m) = \langle \Psi(W_1, W_2, ..., W_m) | H(s_1, ..., s_N) | \Psi(W_1, W_2, ..., W_m) \rangle
\]

According to the ansatz shown in figure 2, the quantum circuit is parametrized with parameters \( W = W_1, W_2, ..., W_5 \) for \( m = 5 \). The result of the observable in each of the five qubits (represented at the right side of figure 2) is a bit-string \( s^0 \) that approximately minimizes the second scalar function \( G \) subject to the restriction \( \| s \|_0 = m \). As the measurement is stochastic, the procedure can be repeated a plurality of times to select the best bit-string from all of them.

Figure 3 shows a second quantum circuit parametrized with parameters \( W = W_1, W_2, ..., W_5 \) for \( m = 5 \).

Figure 2 shows a quantum circuit wherein \( |0\rangle \) is set for all qubits. This input is modified with phase-shift gate \( R_y(W_j) \), the phase shift around axis \( y \), wherein the phase-shift in the \( j \)th qubit is the parameter \( W_j \). Phase-shift gate \( R_y(W_j) \) may be modelled as

\[
R_y(W_j) = \begin{pmatrix} 1 & 0 \\ 0 & e^{iW_j} \end{pmatrix}
\]

Each qubit is connected with a CNOT gate with the following qubit in a sequential manner. After this elemental operation, the set of five qubits are observed.

Figure 3 shows the same part at the left wherein there are five phase-shift gate too after setting \( |0\rangle \) for all qubits. In this second quantum circuit the set of five qubits are connected through quantum gates using a predefined controlled-Z operator as shown in figure 3. That is, they are \( z \)-control binary gates. At the end, the set of five qubits are observed.
\[-11\-\]

\[x_j \geq 0, \forall j = 1, ..., N\]
\[x_j \leq u_j, \forall j = 1, ..., N\]

This solution is used for determining matrix \(D\) and now a new combinatorial optimization problem is solved by using the quantum processor (QP):

\[
\arg \min_{s} H = \|y - RDs\|^2
\]
\[
\|s\|_0 = m
\]

The optimal solution obtained in the classical binary processor (CP) is a set of values \((x_1^0, x_2^0, ..., x_N^0)\) minimizing the value of \(F\).

To account for the restriction \(\|s\|_0 = m\) a quantum circuit that avoids unfeasible combinations is selected (simply because the quantum state it produces lives in a subspace that fulfills the restriction), or modify the function \(H\) or the function \(H\) is modified so that it penalizes unfeasible combinations. According to this embodiment, the second option is used by defining the function

\[
H = \|y - RDs\|^2 + \bar{P}\|1^T s - m\|^2
\]

where \(\bar{P}\) is a constant (Lagrange multiplier) and \(1^T\) the transpose of a vector of ones. The term \(\bar{P}\|1^T s - m\|^2\) penalizes those combinations not satisfying the cardinality constraint \(\|y\|_0 = m\) whenever the constant \(\bar{P}\) is high enough. \(\bar{P}\) may be of the same order than \(\|y - RD1\|\) wherein 1 is interpreted as a column vector replacing vector \(s\).

The specific manners to minimize the function \(H\) is as follows. The quantum state of the ansatz depends on \(W = W_1, W_2, ..., W_5\), the number parameters and the number of quantum bits of the quantum circuit chosen. In other embodiments the two numbers may be different. The quantum state is represented by a complex vector \(\Psi(W)\).

The mapping \(s_i \rightarrow 1/2(1 - \sigma_i^Z)\), from bits to Pauli \(Z\) matrix \(\sigma_i^Z\), converts the function \(H\) to a matrix \(\bar{H}\).

The objective function \(H(s_1, ..., s_N)\) depending on \(s_i, i = 1, ..., N\), binary values taking
\[
\sum_{i} R_{ij} = 1 \quad \forall j = 1, \ldots, N
\]

It is selected a number \( m < N \) of facilities in such a way that supplies \( x_j \) satisfy the demand \( y_j \) from customers. The logistic design determined by \( R_{ij} \) will be deemed fixed and therefore the amount of product \( x_j \) supplied by each facility is computed knowing that it cannot exceed the maximum capacity \( u_j \). This condition may be expressed as

\[
x_j \leq u_j \quad \forall j = 1, \ldots, N
\]

A new cost function \( F \) is defined as

\[
F = \|y - Rx\|^2
\]

Therefore, the problem comes down solving the following optimization problem for vector \( x \) of \( N \) components

\[
\underset{x}{\arg \min} F \\
\|x\|_0 = m \\
x_j \geq 0, \forall j = 1, \ldots, N \\
x_j \leq u_j \quad \forall j = 1, \ldots, N
\]

wherein \( \|x\|_0 = \sum_i |x_i|^0 \) stands for the 0-norm, which indicates the number of components different from 0.

The cost function can be expressed as \( F = \|y - RDs\|^2 \), where \( s \) is a vector of decision variables \( s_j \in \{0,1\} \) indicating whether facility \( j \) is selected or not, and \( D \) is a diagonal matrix \( D_{ij} = x_j \delta_{ij} \), being \( \delta_{ij} \) delta of Kroneker and \( x_j \) contains the supply of each facility.

The next step of the method solves the following optimization problem by means of the classical binary processor (CP), assuming there are no restrictions to the number of facilities selected (in other words, the vector \( s \) is a vector of ones),

\[
\underset{x}{\arg \min} F = \|y - RDs\|^2
\]
processor (CP) in communication with a quantum processor (QP). The classical binary processor (CP) comprises a central processing unit (CPU) in communication with a memory storage unit (M) where, when running a program, may instantiate objects like a function, a procedure or an object having implemented private and public variables and also private and public methods.

In the embodiment shown in figure 1, a method for determining the operative conditions in a supply chain wherein there are available \( N \) facilities to supply the demand \( o C \geq N \) custumers.

This task is high demanding from the computational cost point of view. Supplies and demands are determined by using vectors \( x_j \) and \( y_i \) respectively, with dimensions \( N \) and \( C \), since they may be implemented in an array in the classical binary processor. The optimal combination of transportation volumes between suppliers and demands requires a high computational cost assessing all the combinations such that the cost is minimum.

Figure 1 represents function \( F \) shown as a module (MD) depending on a high number of parameters, the plurality of arrows shown at the left side and the scalar value, the cost, with the arrow located at the right side. The purpose of the method is to generate a new instantiated function \( F'_{\text{new}} \) also shown as a module (MD), comprising a lower number \( m \) of parameters but providing the same scalar values under determined conditions. That is, the set of \( m \) reduced parameters is selected among the \( N \) original parameters. The new instantiated \( F'_{\text{new}} \) module (MD) requires a highly reduced computational cost when compared with the \( F \) module (MD); therefore, the computational resources associated to module (MD) \( F \) may be disposed.

The method for generating the new module (MD) \( F'_{\text{new}} \), instantiating the new scalar function, form the module (MD) instantiating the original scalar function \( F \) is according to the steps disclosed below.

As it has been indicated above, supplies and demands are represented by vectors \( x_j \) and \( y_i \) respectively, with dimensions \( N \) and \( C \). A matrix \( R_{ij} \) denotes normalized transportation volumes, i.e., the proportion of product that goes to customer \( i \) form facility \( j \). This matrix satisfies,
iterative process provides a sequence of scalar functions reproducing the behavior of the original scalar function but requiring a reduced number of parameters wherein the $m$ value is reduced in each iteration (in the first iteration $m < N$, in the second iteration $m' < m$ and so on) and therefore a highly reduced computational cost for evaluating it.

This process is stopped when a certain predetermined criteria is reached like the number of parameters of the scalar function.

**DESCRIPTION OF THE DRAWINGS**

These and other features and advantages of the invention will be seen more clearly from the following detailed description of a preferred embodiment provided only by way of illustrative and non-limiting example in reference to the attached drawings.

Figure 1  This figure shows a schematic representation of an hybrid computer according to an embodiment of the invention comprising a classical binary processor and a quantum processor.

Figure 2  This figure shows a specific ansatz of the quantum computer used for determining the best combination of facilities in order to supply the demand.

Figure 3  This figure shows a second specific ansatz of the quantum computer used for determining the best combination of facilities in order to supply the demand.

**DETAILED DESCRIPTION OF THE INVENTION**

As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product.

Figure 1 shows a schematic representation of a hybrid computer (1) according to a first embodiment of the invention. The hybrid computer (1) comprises a classical binary
According to another embodiment, the scalar value of the second scalar function $G$ to be minimized in the classical global optimizer is obtained by executing $K$ times, being $K$ a predetermined positive integer number, the following steps:

5. sending instructions to the quantum computer to create the variational state with the at least one parameter $W_1, W_2, \ldots, W_p$,

retrieving an observable of the quantum state of each of the at least $N$ qubits of the quantum processor and generating a sequence of binary numbers $c$ wherein $k_j$ is 0 or 1 depending on the observed state of the corresponding $j^\text{th}$ qubit, $|0\rangle$ or $|1\rangle$

10. respectively;

computing in the classical binary processor an estimate $F_k = H(k_1, k_2, \ldots, k_N)$;

returning the average value $(\sum_{k=1}^{K} F_k) / K$.

The classical global optimizer is executed in the classical binary processor in a search space defined by parameters $W_1, W_2, \ldots, W_p$. The classical global optimizer determine a sequence of points during the search process moving towards the coordinates where the second scalar function $G$ is minimum. In each point of the sequence of the iterative process the second scalar function $G$ has to be evaluated.

The ansatz is parametrized according to the parameters $W_1, W_2, \ldots, W_p$, the coordinates of a certain point of the iterative process of the classical global optimizer, creating the variational state. The observable of the quantum state is taken as a sample of the value of the second scalar function $G$. This process is repeated $K$ times and the result is the mean value of the $K$ sample values.

The observable of the quantum state provides values 0 or 1 depending on the observed state of the $N$ qubits corresponding to $|0\rangle$ or $|1\rangle$ respectively. The sequences of the binary values may be identified as $k_2, k_2, \ldots, k_N$. This sequence is taken as the argument of the objective function $F_k = H(k_1, k_2, \ldots, k_N)$ which is the scalar value used for computing the mean value.

According to another embodiment, the method departs from the scalar function $F(x_1, x_2, \ldots, x_N)$ reducing the set of parameters to $m$: $F_{\text{new}}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m)$. In the next iteration, the method is applied to $F_{\text{new}}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m)$ wherein now $F_{\text{new}}$ plays the roll of $F(x_1, x_2, \ldots, x_N)$ reducing the set of parameters to $m'$ wherein $m' < m$. This
information to the classical binary processor using 0 or 1 values depending on the state of each qubit, \([0]\) or \([1]\) respectively.

According to an embodiment, the variational quantum optimization method comprises:

- selecting a variational ansatz in the quantum processor comprising the at least one quantum gate and having parameters \(W_1, W_2, ..., W_p\);
- defining a second scalar function \(G\) wherein
  \[G(W_1, W_2, ..., W_p) = \langle \Psi(W_1, W_2, ..., W_p) | H(s_1, ..., s_N) | \Psi(W_1, W_2, ..., W_p) \rangle\]
- wherein \(\Psi(W_1, W_2, ..., W_p)\) is a trial function responsive of the variational ansatz of the quantum processor according to the set of parameters \(W_1, W_2, ..., W_p\);
- executing a classical global optimizer in the classical binary processor to find parameters \(W_1, W_2, ..., W_p\) providing the best approximation to the minimum of the second scalar function \(G(W_1, W_2, ..., W_p)\);
- providing \((s_1^0, ..., s_N^0)\) as the observable of the quantum processor providing the minimum of the objective function \(G(W_1, W_2, ..., W_p)\).

The selection of the ansatz has influence on the convergence of the method but it has been proven that any ansatz may be used for this purpose. The ansatz may provide at list \(P\) parameters \(W_1, W_2, ..., W_p\) which may be different to the number of qubits of the quantum processor.

A second scalar function \(G\) is defined within the quantum processor, defined from the objective function \(H(s_1, ..., s_N)\) satisfying

\[G(W_1, W_2, ..., W_p) = \langle \Psi(W_1, W_2, ..., W_p) | H(s_1, ..., s_N) | \Psi(W_1, W_2, ..., W_p) \rangle\]

wherein the trial function is adjusted by varying parameters \(W_1, W_2, ..., W_p\) from the classical binary processor. The value of the second scalar function \(G\) depends on the values of the \(P\) parameters; therefore, the search space defined by the set of variables \(W_1, W_2, ..., W_p\) is explored using a classical optimization method such as a Newton-Raphson, a Gradient descent or a GMRES.

Once the classical method obtains the best approximation to the minimum of the second scalar function \(G\) the observable corresponding to said minimum is determined: \((s_1^0, ..., s_N^0)\) and said observable is the one providing the minimum of the objective function \(G(W_1, W_2, ..., W_p)\).
The method also further comprises:

\( e) \) \textit{allocating a new scalar function} \( F_{\text{new}} \) \textit{on the classical binary processor depending on} \( m \) \textit{parameters} \( \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m \) \textit{wherein:}

5

- \( \text{the} \ m \ \text{parameters} \ \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m \ \text{are selected among the} \ \text{N} \ \text{parameters} \ x_1, x_2, \ldots, x_N; \ \text{that is, the} \ m \ \text{indices} \ i_1, i_2, \ldots, i_m \ \text{of} \ \tilde{x}_i, j = 1, \ldots, m \ \text{are selected among those satisfying} \ s^0_{i_j} = 1, \ \text{where} \ s^0_i, i = 1, \ldots, N \ \text{are the} \ N \ \text{values provided by the variational quantum optimization method,}

- \( \tilde{x}_j = x_{i_j}, \ j = 1, \ldots, m; \ \text{and,} \)

10

- \( F_{\text{new}}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m) = F(s_1 x_1, s_2 x_2, \ldots, s_N x_N); \)

\( f) \) \textit{returning the optimized new scalar function} \( F_{\text{new}}(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m). \)

Step \( e) \) provides the specific selection of \( m \) variables among the original \( N \) variables allowing to allocate the new function \( F_{\text{new}} \) having a smaller number of variables. The new allocated function substitutes the original \( F \) function but provides the same result with a highly reduced computational cost. That is, the original \( F \) function may be disposed freeing the allocated computational resources.

According to a specific embodiment, the classical binary processor is configured to:

20

\textit{set a value in the at least one parameter of the quantum processor; and,}

\textit{retrieving an observable of the quantum state of each of the at least} \( N \) \textit{qubits of the quantum processor.}

A specific configuration of the classical binary processor allows a communication with the quantum processor allowing to set a value in said quantum processor. In some embodiments, the execution of specific code in the classical binary processor must have access to the hardware of the quantum processor allowing for instance to set parameters of the ansatz that comprises one or more quantum gates modifying the wave representing the states of the qubits. The connection between a bus of data or a register with the parameter value may require an interface for adapting the binary values of the classical binary processor to the parameters of the qubit depending on the physical properties used to configure the quantum gate.

Additionally, the classical binary processor comprises means for carrying out an observation on the quantum state of each of the at least \( N \) qubits and transferring the
A new objective function $H(s_1, \ldots, s_N)$ is defined wherein its parameters are binary values taking the value 0 or 1. These values will used like flags that indicates those variables among the $N$ variables that are kept and, those variables among the $N$ variables that are discarded. The objective function $H$ has matching values with function $F$, specifically at each coordinate determined by the binary combination $s_1, \ldots, s_N$, the value of the objective function is the value of $F(s_1 x_1^0, s_2 x_2^0, \ldots, s_N x_N^0)$; that is, the value of the scalar function $F$ when the coordinates are the set of values where $F$ is minimum except those coordinates where $s_i = 0$ that the coordinate is also zero. The objective function $H$ is therefore construed from the scalar function $F(x_1, x_2, \ldots, x_N)$ by determining the coordinates where $F$ reaches the minimum, that is $(x_1^0, x_2^0, \ldots, x_N^0)$, and for all binary combinations $s_1, \ldots, s_N$ letting $H(s_1, \ldots, s_N) = F(s_1 x_1^0, s_2 x_2^0, \ldots, s_N x_N^0)$.

The method further comprises the step d):

d) selecting $m < N$ and carrying out a variational quantum optimization method on the quantum processor, being the objective function $H(s_1, \ldots, s_N)$ and being the binary variables $s_i$ under the constraints $\sum_{i=1}^{N} s_i = m$; and, providing $(s_1^0, \ldots, s_N^0)$ as the observable of the quantum processor corresponding to the minimum of the objective function $H(s_1, \ldots, s_N)$.

Step d) provides the specific objective function to the quantum processor $H(s_1, \ldots, s_N)$ allowing to use a variational quantum optimization method wherein the quantum processor is adjusted by varying the wave function until the minimum of objective function is obtained and the binary variables $s_i$ satisfy the constraint $\sum_{i=1}^{N} s_i = m$. In such conditions, the observable of the quantum qubits are taken as the coordinates of the objective function $H$ corresponding to the minimum, i.e., $s_1^0, \ldots, s_N^0$. These parameters $(s_1^0, \ldots, s_N^0)$ are the direct result of a quantum measurement of the at least $N$ qubits.

According to an embodiment, the experiment is repeated $K$ times assessing the minimum value as the mean value of all the observations. The result is a set of $N$ binary variables $s_1^0, \ldots, s_N^0$, fulfilling the condition $\sum_{i=1}^{N} s_i = m$ where $H(s_1, \ldots, s_N)$ is minimum and therefore makes the function $F(s_1 x_1^0, s_2 x_2^0, \ldots, s_N x_N^0)$ minimum. Actually expressed as $F(s_1^0 x_1^0, s_2^0 x_2^0, \ldots, s_N^0 x_N^0)$. 
combination in order to reduce the computational cost of the original scalar function.

DESCRIPTION OF THE INVENTION

The present invention is a method for optimizing a process in a classical binary processor combining:

- a quantum processor comprising at least N qubits and at least one quantum gate parameterizable with at least one parameter \( W_1, W_2, ..., W_p \);
- a classical binary processor in communication with the quantum processor.

The classical binary processor is the processor where the computational cost is optimized. The quantum processor is an auxiliary unit performing some specific tasks being controlled by the classical binary processor. Specifically, the quantum processor comprises at least a quantum gate with at least one parameter being set by the classical binary processor.

According to some embodiments, the quantum processor comprises a plurality of quantum gates forming a specific ansatz, that is, a specific combination of quantum gates.

The method comprises at least the steps:

a) receiving in the classical binary processor an allocated scalar function \( F \), the scalar function \( F \) depending on \( N \) parameters: \( F(x_1, x_2, ..., x_N) \);

b) determining in the classical binary processor a set of values \( (x_1^0, x_2^0, ..., x_N^0) \) minimizing de value of \( F \);

c) defining an objective function \( H(s_1, ..., s_N) \) depending on \( s_i \), \( i = 1, ..., N \), binary values taking the value 0 or 1, satisfying \( H(s_1, ..., s_N) = F(s_1 x_1^0, s_2 x_2^0, ..., s_N x_N^0) \).

The allocated scalar function \( F \) depends on \( N \) parameters and comprises the code that, when executed over the \( N \) parameters \( (x_1, x_2, ..., x_N) \), provide the scalar value \( F(x_1, x_2, ..., x_N) \). This scalar function is the process to be optimized by the method. The scalar function \( F \) shows at least a minimum. For convex functions, the minimum value is unique. The set of values \( (x_1^0, x_2^0, ..., x_N^0) \) minimizing de value of \( F \) may be chosen among those well-known in the prior art such as Newton-Raphson, Gradient descent or GMRES.
reserved. Even if a function has been identified with an object, this interpretation is not limited to implementations based on the use of object-oriented programming languages.

When a computing process depends on multiple evaluations of such function, the processor must prepare the set of variables to be transferred to the instantiated function, to evaluate with the code implementing the evaluation algorithm and preparing the result to return the scalar value. When the computational cost resulting from executing the code implementing the evaluation algorithm is highly dependent on the number of variables, then only classical code optimization methods may be applied.

According to this invention, the method provides an automatic method for reducing the number of variables by selecting a subset of variables in such a manner that the resulting subset of variables reproduces the same behavior on the function when removing the dependency with the rest of the variables.

This is also a problem with a high computational cost and that in practice requires prohibitive computational resources. Such problem involves choosing \( m \) variables out of \( N \) original variables requiring brute force to try combinatorial \( (N,m) \), which can become a huge number. For example, if \( N=100 \) and \( m=20 \) is the limited number of variables selected, the result is 535,983,370,403,809,600,000 possible combinations not being affordable in a classical computer.

The use of quantum processors is very incipient and they have very limited use since they have their specific form of programming and it is not easy to combine a classic binary computer with a quantum processor.

It is known the patent application No. US20190095811A1 for solving a variational quantum eigenvalue in a quantum computing machine.

There are other publications wherein the quantum processor is specifically programmed for solving a determined problem but it none of them discloses the combination of a classical binary processor and a quantum processor in such a way that allows determining specific combinations of variables among the huge of possible
METHOD FOR OPTIMIZING A COMPUTING PROCESS IN A CLASSICAL BINARY PROCESSOR

DESCRIPTION

FIELD OF THE INVENTION

The present invention is related to a method for optimizing a computing process in a classical binary processor wherein the computing process is optimized by reducing the number of parameters on which the process depends. The invention is a computer implemented method that combines a classical binary processor and a quantum processor wherein the method receives a high-cost instantiated function and instantiates a new function requiring a much lower computational cost. The method, according to a specific embodiment, may be applied iteratively providing a sequence of functions wherein each function requires a lower computational cost with respect to the previous one converging to a very efficient instantiated function.

PRIOR ART

One of the technical fields with more intensive development is the optimization of computational processes having very high computational requirements. Some of the strategies are based on the splitting of the whole process into a plurality of tasks that may be processed in parallel using parallel architectures.

In many cases the main process may not be implemented in a parallel computer or, even if it is parallelized, the individual tasks still have a very high computational cost or the new parallel paradigm has little impact on the whole process.

In many problems in Engineering uses a process being modeled by a scalar function depending on a plurality of variables. In the context of this invention, an allocated or an instantiated function must be interpreted as an object allocating memory resources and having a plurality of input parameters for transferring data into the object, the specific code for computing the value of the function according to a specific expression or even a more complex algorithm, and an output parameter. Once this function is not used, it can be unallocated, allowing to reuse of memory resources being previously
Published:
— with international search report (Art. 21(3))
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))
Title: METHOD FOR OPTIMIZING A COMPUTING PROCESS IN A CLASSICAL BINARY PROCESSOR

Abstract: The present invention is related to a method for optimizing a process in a classical binary processor. The invention is a computer-implemented method that combines a classical binary processor and a quantum processor wherein the method receives a high-cost instantiated function and instantiates a new function requiring a much lower computational cost. The method, according to a specific embodiment, may be applied iteratively providing a sequence of functions wherein each function requires a lower computational cost in respect to the previous one converging to a very efficient instantiated function.