

# Quantum invariants for the graph isomorphism problem

Hernán I. de la Cruz,<sup>\*</sup> Fernando L. Pelayo,<sup>†</sup> Vicente Pascual,<sup>‡</sup> Jose J. Paulet,<sup>§</sup> and Fernando Cuartero<sup>¶</sup>  
*Universidad de Castilla - La Mancha, Spain*

Luis Llana<sup>\*\*</sup>  
*Universidad Complutense de Madrid, Spain*

Mauro Mezzini<sup>††</sup>  
*Roma Tre University, Italy*  
(Dated: October 7, 2022)

Graph Isomorphism is such an important problem in computer science, that it has been widely studied over the last decades. It is well known that it belongs to NP class, but is not NP-complete. It is thought to be of comparable difficulty to integer factorisation. The best known proved algorithm to solve this problem in general, was proposed by László Babai and Eugene Luks in 1983.

Recently, there has been some research in the topic by using quantum computing, that also leads the present piece of research. In fact, we present a quantum computing algorithm that defines an invariant over Graph Isomorphism characterisation. This quantum algorithm is able to distinguish more non-isomorphic graphs than most of the known invariants so far. The proof of correctness and some hints illustrating the extent and reason of the improvement are also included in this paper.

## I. INTRODUCTION

The problem of whether two graphs are isomorphic, that is, determining the existence of any bijection between the vertices set of both graphs that preserves the graph structure, has been a great challenge for the scientific community for a long time. Both theoretical and computational aspects related to this problem have been extensively discussed in the literature in the last decades. The problem of graph isomorphism is not only interesting from a theoretical point of view but also from the point of view of the computational complexity since the class of complexity to which this problem belongs remains uncertain. It is not yet determined whether the graph isomorphism problem can be considered as either a  $P$ -problem or a  $NP$ -complete problem or it belongs to a different category. It was even described as a disease in 1977 [1], due to the fact that every one who has studied it felt into an obsession about it.

As a consequence, many authors have addressed this problem so proposing different approaches to solve it but, so far, an efficient algorithm to conclude in any possible case, whether or not two given graphs are isomorphic is not known. Until now, the best known algorithm to generally solve the isomorphism problem was proposed by László Babai and Eugene Luks in [2]. It holds a computational complexity of  $O(2^{\sqrt{n \log n}})$  where  $n$  is the number of vertices in the graph.

It is well known the fact that two graphs having an identical characteristic polynomial is an invariant, i.e., it is a necessary (but no sufficient) condition for them to be isomorphic. Two graphs sharing the same characteristic polynomials are called *isospectral* graphs and they have been extensively studied in the literature. There is also an algorithm [3] that relies on this fact, for the case of eigenvalues holding some restrictions.

Although, as mentioned before, an efficient algorithm for the general case has not been found yet, there are some algorithms that can efficiently solve the problem in specific cases, like Nauty [4], the first efficient method that was able to deal with large graphs using branching techniques. Later, Saucy [5] proposed the use of disperse data structures. Other improvements were added by Bliss [6] and Conauto [7] introducing the use of a novel branching technique.

In 2015, László Babai reported the discovery of a quasi-polynomial time algorithm based on the one hand on a divide and conquer scheme and on the other hand on a smart use of both group theory and combinatorics, so decomposing the problem in the search of patterns based on *Johnson Graphs*, leading to an algorithm that solves the problem in time  $O(2^{c \log(n)})$  for some constant  $c$ . This result, published in [8] has not been confirmed yet.

Currently the lack of general efficient methods is not a hard problem since we can even say that the graph isomorphism problem has been solved from a practical point of view. There are two main reasons supporting this claim. First, there is a significant amount of graphs for which there is an efficient procedure and, second, there are some heuristic algorithms that provide suitable solutions.

Most of these efficient techniques, including Babai approach, are based on combinatorics, group theory and branching techniques. A good classification can be found in [9] where the authors rely on some other papers. In

---

\* HernanIndibil.Cruz@uclm.es

† FernandoL.Pelayo@uclm.es

‡ vpfuniversity@gmail.com

§ jpaulet@uclm.es

¶ Fernando.Cuartero@uclm.es

\*\* llana@sip.uclm.es

†† mauro.mezzini@uniroma3.it

[10] the authors provide the first general understanding of the Hidden Subgroup Problem, HSP over non Abelian groups. As a positive case when  $H$  is an arbitrary subgroup of  $G$  and  $H^G$  is the largest subgroup of  $H$  that is normal in  $G$ , it is true with high probability that  $H^G$  is uniquely determined in an efficient way, that is, within polynomial time. However, this generalisation to non Abelian case does not solve efficiently the graph isomorphism problem. In the paper [11] the authors move the GI problem into a combinatorial optimization problem to be solved by a Quantum Approximate Optimization Algorithm, QAOA. Nevertheless, there is still no full consensus over the complexity of a general quantum adiabatic algorithm on the related literature.

This paper is structured as follows: Section 2 presents the common pattern for codifying graphs. Section 3 presents one main contribution of this paper as a variation scheme for codifying graphs. Section 4 is devoted to present on detail the algorithm for characterizing graphs which represents the core of the paper. Conclusion and future work section finishes the paper.

## II. GRAPHS CODIFICATION

A graph is a collection of nodes, some pairs of them connected by an edge. Formally, an undirected/non directed graph is a pair  $G = (V, E)$ , where  $V = \{1, 2, \dots, n\}$  ( $|V| = n$ ) is the finite set of vertices/nodes, and  $E$  is a set of *edges* where each edge is a set of exactly two vertices belonging from  $V$ .  $E \subseteq V \times V$  is the set of edges where loops are not included, i.e., without edges in the form  $(x, x)$  for any node  $x$ .

Given two graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$ , we say that they are isomorphic if there exists a bijection  $b: V_1 \rightarrow V_2$  such that  $(x, y) \in E_1 \iff (b(x), b(y)) \in E_2$ .

Quantum states entanglement technique to model a non-directed graph was first conceived by Hein et al in [12, 13], where the entanglement states in an  $n$ -particles system are supporting a mathematical description of a graph where each vertex corresponds to a qubit and edges represent interactions between them.

In order to do this, the former system of  $n$  qubits must be in a superposition of base states  $|0\rangle$  and  $|1\rangle$ , i.e., in  $|+\rangle$  state. Afterwards, each edge  $u = (a, b) \in E$  is encoded by a controlled  $Z$  quantum gate over the qubits  $a$  and  $b$  no matter which acts as control and which is controlled.

Figure 1 shows the circuit modelling the  $C_4$  graph, i.e. graph  $G = (V, E)$ , where  $V = \{1, 2, 3, 4\}$  and the set of edges is  $E = \{(1, 2), (2, 3), (3, 4), (1, 4)\}$ . This corresponds to a circuit graph of 4 nodes (i.e.  $C_4$ ).

Right hand side of figure 2 shows graphically the amplitudes of the components of the Hilbert space basis on Quirk-QS (Quantum Simulator), which can also be explicitly represented by

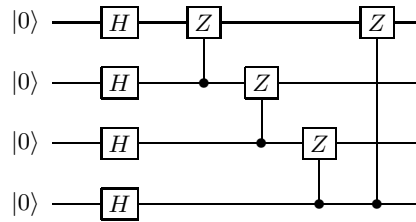


Figure 1. Example. Graph circuit  $C_4$

$$\begin{aligned}
 |C_1\rangle = & \frac{1}{4}(|0000\rangle + |0001\rangle + |0010\rangle - |0011\rangle \\
 & + |0100\rangle + |0101\rangle - |0110\rangle + |0111\rangle \\
 & + |1000\rangle - |1001\rangle + |1010\rangle + |1011\rangle \\
 & - |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle)
 \end{aligned}$$

According to this codification  $C_4$  graph is represented by the amplitudes of the entangled states, in fact the absolute value of their modules meet in all of them, where for the components corresponding to the 4 edges the sign is negative whereas for the remainder 12 the sign is positive.

This encoding scheme provides us with quite interesting features from which we are trying to get the most. In [14] the authors use this fact to study properties of the graphs leveraging on the performance increasing associated with quantum computing. Furthermore, this quantum encoding allows an efficient handling so as to overcome classical algorithms for common tasks in this field of research. In particular, first section of the given reference is focused on how it could be used for dealing with graph isomorphism problem, nevertheless they finally stated that this idea is not enough to properly characterize graph isomorphism, sic: *... it is tempting to believe that different graphs necessarily lead to states that are entangled in different ways. This is not the case, however, as graph states generated from different graphs may be equivalent up to local operations.*

In a quite similar way the problem is faced in [15] where this encoding scheme is taken to provide invariants for the graph isomorphism problem. Two main modifications over the proposal by Zhao et al. are now made. The first is just an aesthetic issue in the sense that it uses the adjacency matrix which under the restrictions of non-directed graph without any self loop, is equivalent to just one of the triangles out of the main diagonal (i.e. either row  $>$  column, or, row  $<$  column). Each column in the triangle is encoded in a column of quantum gates, the 0 of the diagonal of the adjacency matrix is encoded by a control. These quantum gates are  $Z$  when there is an edge and Identity when there is no edge.

For the previous example of  $C_4$  graph, its adjacency matrix can be represented by three ways where being one of them  $M_1$  is the following

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Figure 2 shows the corresponding codification of  $|M_1\rangle$  on Quirk-QS.

A noticeable contribution is made by Mills et al. in [15] where weighted graphs can be represented by means of the use of controlled *some root of Z* gates, instead of the former Pauli *Z* gates which mean a rotation of  $\pi$  radians on *Z* axe.

Making use of these adds-on that paper is focused (one more paper) on searching for invariants for graph isomorphism characterisation and some other related problems. There are many classical invariants for this purpose as the characteristic polynomial associated to the adjacency matrix for which two isomorphic graphs share the same invariant but the contrary is not true. Invariants searching is an interesting motto to finally find one which works both sides, i.e. a bijection between the quotient set of graphs isomorphisms-modulo and, the set of values for that invariant provided that, of course, this would be possible.

In the cited paper [15] they make use of Wigner's functions in this searching for invariants task. These functions allow characterising the set of reached states (see [16]). Experiments for this task has been carried out over the IBM Quantum Experience computers so showing that they are able to discriminate a number of invariants for graph isomorphism problem.

### III. MOTIVATION AND INVARIANTS

As previously stated, Mills et al. in [15] proposed using controlled *Z* quantum gates to model a graph. The *Z* gate performs a rotation of  $\pi$  radians on *Z* axe. But our proposal is to use gates with a smaller angle for rotations, as in that paper do when dealing with weighted graphs.

In the following example, we use the *T* gate, which corresponds to a rotation of  $\pi/4$  radians. In general, we will use the general parameterised  $R_z(\varphi)$  gate, when  $\varphi$  is the angle of rotation around the *Z* axis, with  $T = R_z(\frac{\pi}{4})$ . We make use of this gate in order to show the codification of the  $C_4$  graph, the circuit of 4 nodes.

As previously indicated, we believe that these gates can be used beyond the proposal of Mills et al. for weighted graphs. And first, we can see how using *T* gates instead of *Z* ones when encoding  $C_4$  graph, the new quantum circuit on Quirk-QS becomes as depicted in figure 3.

The detailed description of the corresponding quantum state is

$$\begin{aligned} |M_1\rangle = & \frac{1}{4}(|0000\rangle + |0001\rangle + |0010\rangle + e^{i\pi/4}|0011\rangle \\ & + |0100\rangle + |0101\rangle + e^{i\pi/4}|0110\rangle + e^{i\pi/2}|0111\rangle \\ & + |1000\rangle + e^{i\pi/4}|1001\rangle + |1010\rangle + e^{i\pi/2}|1011\rangle \\ & + e^{i\pi/4}|1100\rangle + e^{i\pi/2}|1101\rangle + e^{i\pi/2}|1110\rangle + e^{i\pi}|1111\rangle) \end{aligned}$$

Going deeper on the amplitudes graphically depicted in the referred Quirk-QS snapshot, it is easy being aware that variations between positive and negative values that appeared when using just *Z* gates become now into rotations on the phase component of each of the amplitudes. Therefore, 7 out of the 16 components of the whole quantum state keep a 0 radians phase component; 4 out of 16 hold a  $\pi/4$  radians phase component (the rotation provided by once operating *T* quantum gate); another 4 (out of 16) appear as rotated  $\pi/2$  radians (twice *T* applying) and the last of the 16 has a component of  $\pi$  radians which is equivalent to 4 times *T* operations.

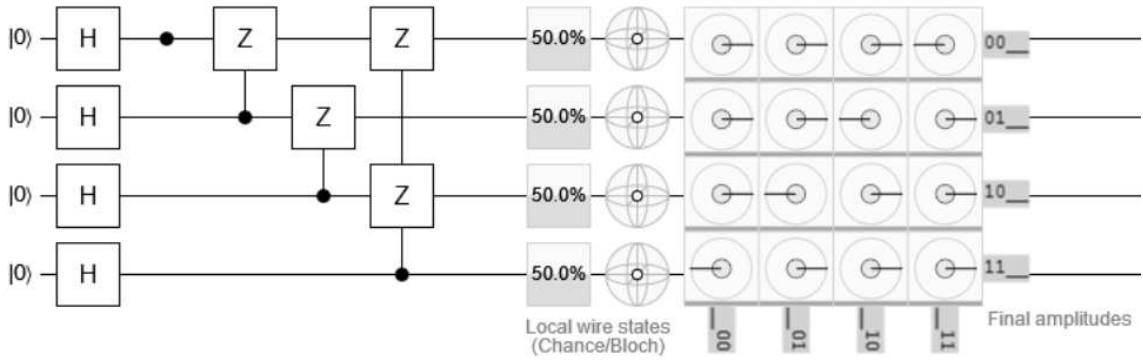
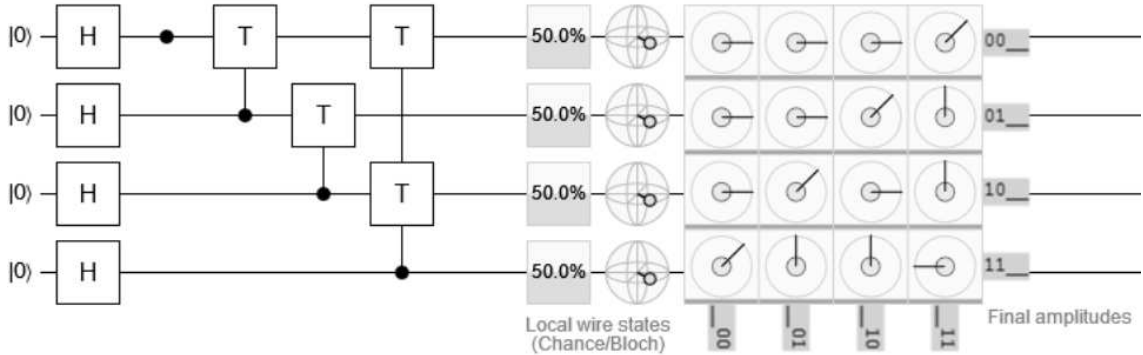
This distribution of amplitudes corresponding to the 16 elements of the base in the Hilbert Space for this state can be observed in detail in the table I.

Basis vector	Phase	N. of $\pi/4$ rotations
$ 0000\rangle$	0	0
$ 0001\rangle$	0	0
$ 0010\rangle$	0	0
$ 0011\rangle$	$\pi/4$	1
$ 0100\rangle$	0	0
$ 0101\rangle$	0	0
$ 0110\rangle$	$\pi/4$	1
$ 0111\rangle$	$\pi/2$	2
$ 1000\rangle$	0	0
$ 1001\rangle$	$\pi/4$	1
$ 1010\rangle$	0	0
$ 1011\rangle$	$\pi/2$	2
$ 1100\rangle$	$\pi/4$	1
$ 1101\rangle$	$\pi/2$	2
$ 1110\rangle$	$\pi/2$	2
$ 1111\rangle$	$\pi$	4

Table I. M1 state amplitude analysis

This way of representing graphs can be straightforward generalized to any graph such way that once the quantum gate has been chosen, the amplitude associated to each element of the base corresponds to a whole *multiple* of the rotation provided by the referred quantum gate.

As we have chosen *T* quantum gates  $T = Z^{1/4}$ , 8 different tentative rotations can appear at most (more than just 2 as provided by *Z* q-gate). We believe that having overcome this restriction release us from suffering the problem stated by Zhao et al., i.e. lacking from the discrimination power required for some pairs of non-isomorphic graphs that just looking at their entanglement properties seemed wrongly being so. We have taken the route of increasing the accuracy of the quantum gates used for this sake.

Figure 2. Example 2. Circuit for  $|M_1\rangle$ Figure 3. Example 2. Circuit for  $|M_1\rangle$  with  $T$  gates

Elaborating this requires knowing how the rotations on the amplitude phase are made, which comes from the fact that each element of the Hilbert's space base corresponds to a subgraph from the one under consideration. This subgraph is formed after taking the nodes having a 1 within the sequence of bits that defines it on the basis vector. On the example of  $C_4$  graph we are using, the element of the base  $|1011\rangle$  represents the subgraph including nodes  $\{1, 3, 4\}$ , which contains two edges  $((1, 4)$  and  $(3, 4)$ ). Finally each controlled  $T$  quantum gate acts over the two qubits related with the edge so generating an extra rotation of  $\pi/4$  radians on every amplitude in which both qubits meet on value 1. This way makes the amplitudes adding up all the rotations corresponding to the edges included in the given subgraph.

It is immediate to discover that the phase angle of the amplitude for the base vector is  $\pi/2$  coming from twice rotating  $\pi/4$ . This fact also shows that the phase angle of the amplitude can also provide the number of edges corresponding to the associated subgraph.

The above consideration is always true as coming straightforward from the effect of the controlled phase displacement quantum gate. This gate performs such displacement over all the amplitudes in which both qubits meet on 1, so adding up all these amplitudes therefore also computing the number of edges.

We are now on condition of stating, as conjecture, a

tentative invariant for the sake of addressing the graph isomorphism problem. We start with the following

**Proposition 1** Given two graphs  $G = (V, E)$  and  $G' = (V', E')$  with  $n$  nodes each one, they are isomorphic if and only if there exists a bijection  $f$

$$f : \{1 \dots n\} \longrightarrow \{1 \dots n\}$$

holding that for every nodes subset of  $V$ ,  $S = \{x_1, x_2, \dots, x_j\} \subseteq V$  there is a corresponding subset of  $V'$ ,  $S' = \{x_{f(1)}, x_{f(2)}, \dots, x_{f(j)}\} \subseteq V'$  such that the corresponding subgraphs over those subsets have the same number of edges.

**Proof:**

If there exists an isomorphism between  $G$  and  $G'$  then there is a permutation on nodes in  $V$  such that edges on  $E$  correspond to edges on  $E'$ . In fact, this permutation defines  $f$  bijection required to conform  $S'$  properly as it covers all the subsets of nodes. Therefore it is an invariant.

The other way round is also true, for this it is enough considering  $f$  bijection acting over sets  $S$  and  $S'$  with two nodes each, this defines the permutation characterizing the isomorphism since all the possible edges are defined on both  $S$  and  $S'$ . It is immediate that edges

exist together with the correspondent on the other set, i.e. isomorphism's definition is held.  $\square$

Proposition 1 states that from the phase amplitudes of our codification all the possible subgraphs are reachable, so this, the previously referred cite of Zhao et al. [14] is refuted, in fact assuming the use of quantum gates corresponding to small enough rotations on  $Z$  axis (with an appropriate value for the  $k$  parameter referred at the beginning of this section), the underlying quantum modeling of graphs that relies on the entanglement on the qubits phases is enough to characterize different graphs.

Nevertheless, the number of subgraphs for a graph of  $n$  nodes is  $2^n$  what could seem that we move from a given problem to another one with higher complexity than the former's one. Fortunately this is not this way in the end, since leveraging on the quantum algorithm allows us to classify all the  $2^n$  subgraphs by their number of edges, which also will conform a new invariant for our problem.

#### IV. NEW INVARIANTS

The presented codification provides a full characterization of different graphs by means of their amplitudes. Nodes permutation can order those amplitudes even when all of them share the same module because an order on their phases can be established. This order ranges from 0 to the total number of edges of the graph.

There are some drawbacks arising when the accuracy required for the quantum gates at modelling edges must guarantee that the last/biggest amplitude (corresponding to the whole graph, so including the maximum number of edges) is required being smaller than  $2\pi$ , i.e. for our example of  $C_4$  graph a quantum  $T$  gate is enough as  $T$  allows up to 7 rotations ( $\pi/4$  radians) before reaching  $2\pi$  radians. It is immediate that this 4-edges graph can easily be modelled. In fact, all the graphs with 4 or less nodes can be modelled. Notice that a 1000 nodes graph could have up to  $1000 \cdot 999/2$  edges for the complete graph, in which case a quantum gate like  $Z^{1/(1024 \cdot 512)}$  (for accuracy reason) is required for amplitudes not to reach  $2\pi$  in the phase component.

On practice, this imposes hard conditions for designing such very accurate quantum gates. Nevertheless, this is not new since e.g. Shor's algorithm also requires this sort of accuracy demands even when all the amplitudes being smaller than  $2\pi$  radians on the phase angle is not required (because modular arithmetic could be used).

Another drawback is the impossibility to direct access the amplitudes of the quantum state. Instead, measuring the system (making the quantum state to collapse) is needed. Quirk shows in many examples that the probability of measuring either  $|0\rangle$  or  $|1\rangle$  is 50% in all the qubits of the system which, of course, lacks of any practical information.

Therefore, a system transformation to make phase angles becoming amplitude modules in order to be aware of

these differences is required. The quantum phase estimation, QPE for short, algorithm [17] has been chosen for this purpose. This algorithm estimates the phase of the eigen-vector associated to a given quantum gate  $U$ . It is also present on the quantum computing part of Shor's algorithm [18]. This quantum algorithm moves the differences on the phase in the graph codification proposed in our model to a difference on the amplitudes that could be measured on practice. An sketched image of QPE into Shor algorithm can be seen in figure 4.

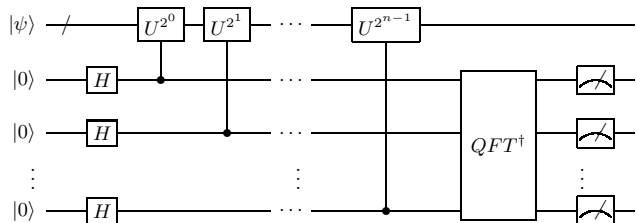


Figure 4. Quantum phase estimation circuit

#### A. Quantum Phase Estimation Algorithm applying

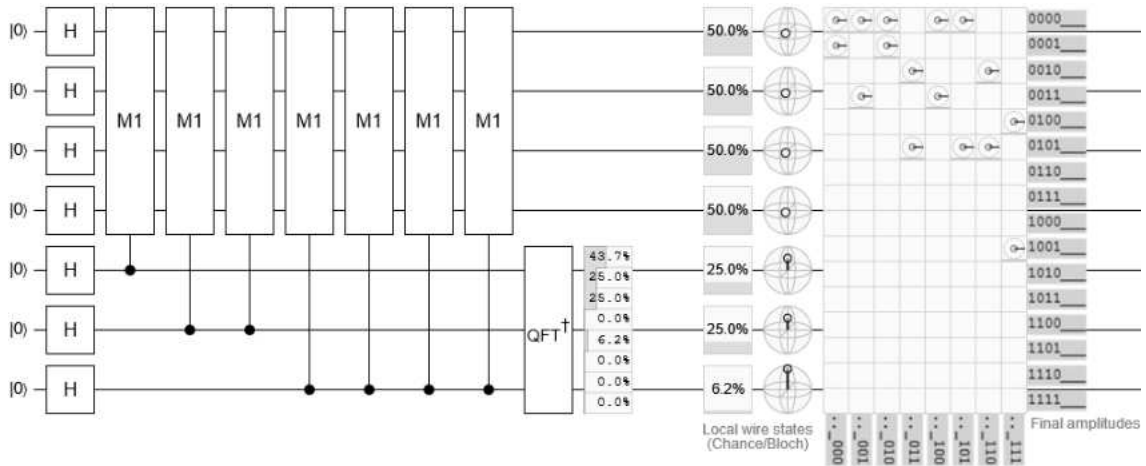
The following example is used in order to graphically show how does QPE Algorithm work. Let  $M_1$  be the piece of quantum algorithm corresponding to the  $C_4$  graph encoded on Figure 3 that is taken as oracle. The quantum phase algorithm over  $M_1$  circuit is shown in Figure 5.

Applying the quantum phase algorithm over  $M_1$  offers a set made of the 128 amplitudes of the Hilbert space. Only 16 among them are non-zero so corresponding to the 16 subgraphs. The non-zero components of the new quantum state are:

$$|M_1\rangle = \frac{1}{4}(|000000\rangle + |000001\rangle + |000010\rangle + |0000100\rangle + |000101\rangle + |0001000\rangle + |0001010\rangle + |0010011\rangle + |0010110\rangle + |0011001\rangle + |0011100\rangle + |0100111\rangle + |0101011\rangle + |0101101\rangle + |0101110\rangle + |1001111\rangle)$$

These non-zero amplitudes associated to base vectors should be understood as follows, qubits 1:3 binary encode the number of edges of the subgraph, besides remainder qubits 4:7 describe the presence/absence (1/0) of nodes within the corresponding subgraph. Next 3 paragraphs elaborate on detail, by means of our case study of  $C_4$ , how we are numbering qubits as well as how we encode and decode each digit of the *ket* in the Hilbert base.

Right hand side of figure 5 shows all these amplitudes where all of them but 16 lack of any phase (0 radians) whereas the rest of them (positive amplitudes) are those in which we are interested.

Figure 5. Evaluation of graph  $M_1$ 

Let us notice that first 3 qubits corresponds with control qubits (downside of Quirk circuit) which are computing the number of edges in the subgraphs. Top line in Quirk has 5 positive amplitudes out of 8 which corresponds with the base elements  $|0000000\rangle$ ,  $|0000001\rangle$ ,  $|0000010\rangle$ ,  $|0000100\rangle$  and  $|0000101\rangle$ , where first three bits (1:3) 000 mean 0 edges, and, the last four bits (4:7) those in charge of identifying the nodes involved, i.e.  $\emptyset$ ,  $\{1\}$ ,  $\{2\}$ ,  $\{3\}$ , and  $\{1,3\}$  respectively. The line behind the previous one stands for the remainder subgraphs without any edge but with nodes  $\{4\}$  and  $\{2,4\}$ .

For the rest it is easy to see that base element  $|0010011\rangle$  represents the single-edge subgraph  $\{1,2\}$ , the following single edge subgraphs are  $\{2,3\}$ ,  $\{1,4\}$  and  $\{3,4\}$ . The remainder non-zero amplitudes corresponds with 4 two edges subgraphs with three nodes each and the last refers the complete four-edges with four nodes subgraph  $C_4$ .

Going deeper on this, we state that just a single measurement on the control qubits provides us with so much information in the following way. One by one throws probabilities 25%, 25% and 6.25% of being  $|1\rangle$ . If we perform three measures at once we get the probability of reaching each of the values of the measurement (binary encoded) multiplied by the total number of subgraphs (16 in this case) representing the amount of subgraphs with such number of edges. In the example of  $C_4$  graph it includes 7 subgraphs with no edges, 4 subgraphs with one edge, 4 subgraphs with two edges and the only four edges subgraph, i.e., itself.

$\#(edges)$	$\%Probability$	$\#(subgraphs)$
0	43.75	7
1	25.0	4
2	25.0	4
3	0.0	0
4	6.25	1

For the sake of better illustrating the way it works, let us consider the following adjacency matrices of graphs:

$$M_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

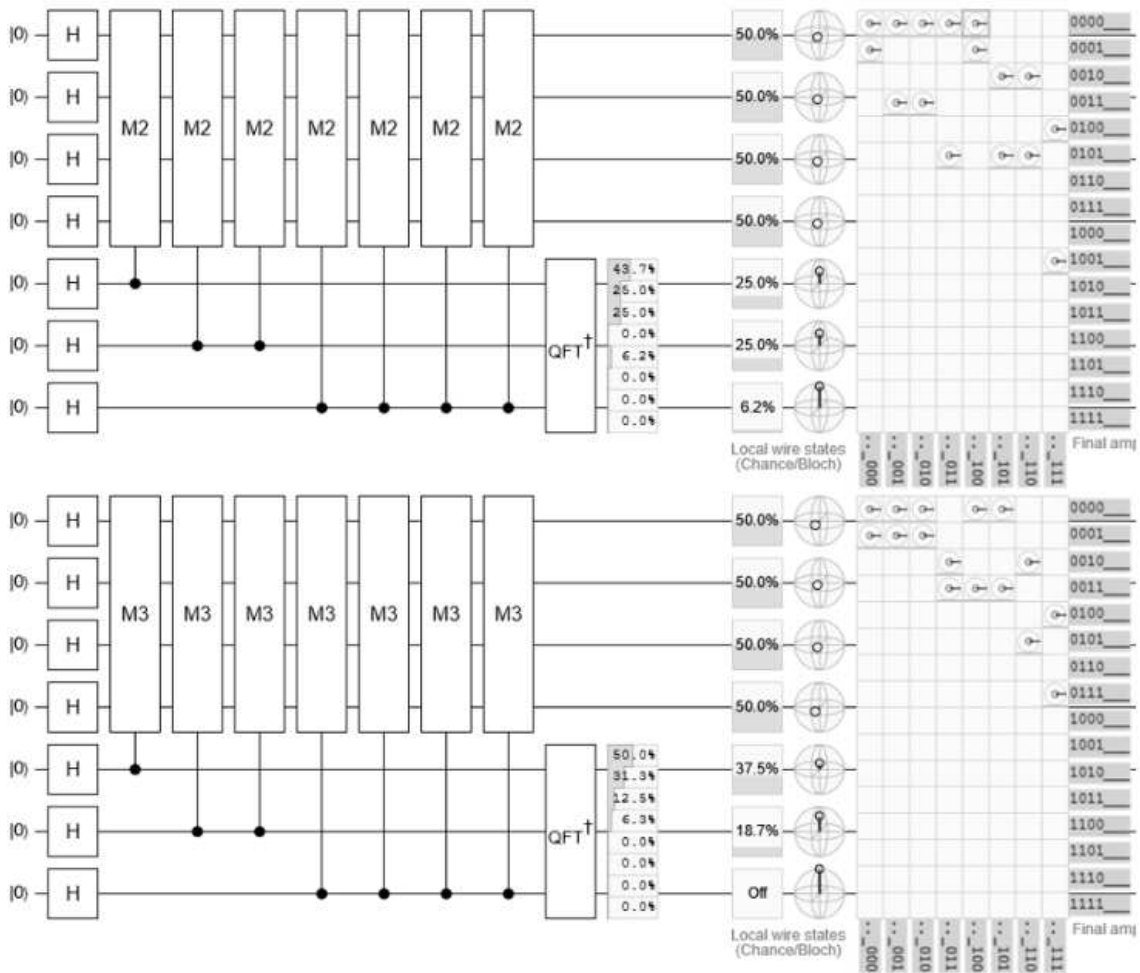
$$M_3 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Graph represented by  $M_1$  matrix has been evaluated by the quantum circuit on figure 5. Same way, figure 6 shows the respective evaluations of circuits  $M_2$  and  $M_3$ . All the values reached at measuring three bottom (control) qubits for  $M_1$  and  $M_2$  meet, so denoting they can be isomorphic graphs. On the contrary  $M_3$  evaluation differs from any of the previous as the corresponding graph is isomorphic with none of them.

$\#(edges)$	$\%Probability$	$\#(subgraphs)$
0	50.0	8
1	31.25	5
2	12.5	2
3	6.25	1

This quantum oracle 1 provides a powerful tool to be used onto graph isomorphism problem as identifying all the possible subgraphs of the one under consideration. This decomposition is indexed by the number of edges in each subgraph.

We have described the coding of a graph in a quantum circuit, and how to obtain its invariants by means of the table of subgraphs classified by probability, which offers us the number of edges. Now we must focus on

Figure 6. Evaluation of graphs  $M_2$  and  $M_3$ 


---

**Algorithm 1:** Algorithm to codify the quantum circuit
 

---

- 1 Let  $G = (V, E)$  be the graph, with  $|V| = n$  and  $|E| = m$
  - 2  $k = \lceil \log_2 m \rceil$
  - 3 Prepare the Oracle  $M$  in the following way
  - 4 **for** every  $(i, j) \in E, i < j$  **do**
  - 5     Add the gate Controlled  $Z^{1/2^k}$  with control in qubit  $i$  applied on qubit  $j$
  - 6 **end**
  - 7 Prepare  $n$  qubits initialized to  $|+\rangle$  for the Graph
  - 8 Prepare  $k$  qubits initialized to  $|+\rangle$  for the Estimation
  - 9 **for**  $i \in \{1, \dots, k\}$  **do**
  - 10     Apply  $2^{i-1}$  Controlled  $M$  gates on all Graph qubits, with control in Estimation qubit  $i$
  - 11 **end**
  - 12 Apply  $QFT^\dagger$  on Estimation qubits
  - 13 Measure Estimation qubits
- 

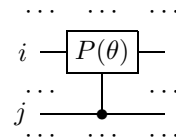
the measure of it that will refer to the control qubits in the application of the phase estimation algorithm. First,

we have that the observable to be measured does not depend on the order in which qubits are labelled, which is a necessary condition for an invariant. And second, in this case, we can use the tomography techniques described in [19] to obtain the measurement of the probabilities of the involved amplitudes.

### B. Correctness of the algorithm

This section is devoted to proof that the method above described is correct.

Let us consider a non directed graph  $G = (V, E)$ , let us assume that  $V = \{0, \dots, n-1\}$ . For each edge  $(i, j) \in E$  with  $0 \leq i < j < n$  let us consider the operator  $A_{ij}^\theta$ , with gates of precision  $\theta$ , defined as follows



This is formalised in the following definition

**Definition 1** Let  $G = (V, E)$  with  $V = \{0, \dots, n-1\}$  be a non directed graph,  $\theta \in \mathbb{R}$  s.t.  $0 \leq \theta < 2\pi$ , and  $(i, j) \in E$  with  $0 \leq i < j < n$ . Let us consider the operator

$$A_{ij}^\theta = I^{\otimes i} \otimes (P(\theta)) \otimes I^{\otimes j-i-1} \otimes |1\rangle\langle 1| + I^{\otimes j-i} \otimes |0\rangle\langle 0| \otimes I^{\otimes n-j}$$

Although we have not proved that the operators  $A_{ij}^\theta$  commute between them yet (this is consequence of Lemma 1), we have that the operator associated to the graph appearing in Figure 1 is given by the following

$$U_G^\theta = \prod_{\substack{0 \leq i < |V|, \\ i < j < |V|, \\ (i,j) \in E}} A_{ij}^\theta$$

Now let us consider a subgraph  $G' = (V', E')$  of  $G$ . Let us consider the vector

$$|\Psi_{G'}\rangle = x_0 \otimes \dots \otimes x_i \otimes \dots \otimes x_j \otimes \dots \otimes x_{n-1} \quad \text{where } x_i = \begin{cases} |1\rangle & \text{if } i \in V' \\ |0\rangle & \text{if } i \notin V' \end{cases}$$

**Lemma 1** Let  $G = (V, E)$  be a graph,  $(i, j) \in E$  with  $0 \leq i < j < n$ ,  $0 < \theta < 2\pi$ , and a subgraph  $G' = (V', E')$ . We have that

- if  $(i, j) \in E'$  then  $A_{ij}^\theta |\Psi_{G'}\rangle = e^{i\theta} |\Psi_{G'}\rangle$
- if  $(i, j) \notin E'$  then  $A_{ij}^\theta |\Psi_{G'}\rangle = |\Psi_{G'}\rangle$

**Proof:** First, let us recall that  $P(\theta)|0\rangle = |0\rangle$  and  $P(\theta)|1\rangle = e^{i\theta}|1\rangle$ .

If we apply the operator  $A_{ij}^\theta$  to  $|\Psi_{G'}\rangle$  we obtain

$$\begin{aligned} A_{ij}^\theta |\Psi_{G'}\rangle &= x_0 \otimes \dots \otimes x_{i-1} \otimes \\ &\left( (P(\theta)x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes |1\rangle\langle 1|x_j) \right. \\ &\quad + \\ &\quad \left. (x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes |0\rangle\langle 0|x_j) \right) \\ &\quad \otimes x_{j+1} \otimes \dots \otimes x_{n-1} \end{aligned}$$

Let us consider the following cases:

- $j \notin V'$ , then  $x_j = |0\rangle$ . In this case  $|1\rangle\langle 1|x_j = |1\rangle\langle 1|0\rangle = 0$  and  $|0\rangle\langle 0|x_j = |0\rangle\langle 0|0\rangle = |0\rangle$ . Therefore

$$\begin{aligned} &\overbrace{\left( (P(\theta)x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes |1\rangle\langle 1|x_j) \right)}^{=0} \\ &\quad + \\ &\quad \left( x_i \otimes x_{i+1} (\otimes x_{j-1} \otimes \overbrace{|0\rangle\langle 0|x_j}^{=x_j}) \right) \\ &= \\ &x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes x_j \end{aligned}$$

- $i \notin V'$  and  $j \in V'$ . In this case we have  $x_i = |0\rangle$ ,  $x_i = |1\rangle$ ,  $P(\theta)x_i = P(\theta)|0\rangle = |0\rangle = x_i$ ,  $|1\rangle\langle 1|x_j = |1\rangle\langle 1|1\rangle = |1\rangle = x_j$ , and  $|0\rangle\langle 0|x_j = |0\rangle\langle 0|1\rangle = 0$ . Therefore,

$$\begin{aligned} &\overbrace{\left( (P(\theta)x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes \overbrace{|1\rangle\langle 1|x_j}^{=x_j}) \right)}^{=x_i} \\ &\quad + \\ &\quad \overbrace{\left( x_i \otimes x_{i+1} (\otimes x_{j-1} \otimes \overbrace{|0\rangle\langle 0|x_j}^{=0}) \right)}^{=0} \\ &= \\ &x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes x_j \end{aligned}$$

- $(i, j) \in E'$ . This case is similar to the previous one. But  $x_i = |1\rangle$  and  $P(\theta)|1\rangle = e^{i\theta}|1\rangle$ . Therefore,

$$\begin{aligned} &\overbrace{\left( (P(\theta)x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes \overbrace{|1\rangle\langle 1|x_j}^{=x_j}) \right)}^{=e^{i\theta}x_i} \\ &\quad + \\ &\quad \overbrace{\left( x_i \otimes x_{i+1} (\otimes x_{j-1} \otimes \overbrace{|0\rangle\langle 0|x_j}^{=0}) \right)}^{=0} \\ &= \\ &e^{i\theta}x_i \otimes x_{i+1} \cdots \otimes x_{j-1} \otimes x_j \end{aligned}$$

Since  $e^{i\theta}$  is a scalar, we obtain  $A_{ij}^\theta |\Psi_{G'}\rangle = e^{i\theta} |\Psi_{G'}\rangle$

□

A clear consequence of the previous Proposition is that the operators  $A_{ij}^\theta$  commute, so the following definition is correct,

**Definition 2** Let  $G = (V, E)$  be a non directed graph and  $0 < \theta < 2\pi$ . We define

$$U_G^\theta = \prod_{\substack{0 \leq i < j < |V| \\ (i,j) \in E}} A_{ij}^\theta$$

**Proposition 2** Let us consider  $G = (V, E)$  a graph, and  $(i, j) \in E$  with  $0 \leq i < j < n$ ,  $1 < \theta < 2\pi$ , and a subgraph  $G' = (V', E')$ . Then

$$U_G^\theta |\Psi_{G'}\rangle = e^{i\theta|E'|} |\Psi_{G'}\rangle \quad (1)$$

**Proof:** This is a direct consequence of the previous Lemma 1. □

In order to obtain  $|E'|$  we can apply the Quantum Phase Estimation (QPE in short) circuit. But first, we have to rewrite Equation 1. We have not given a proper value of  $\theta$  yet but applying the QPE circuit is required, therefore, the value of  $\theta$  is restricted. First, we are going to apply the QPE circuit to estimate the number of

subgraphs that a given graph  $G = (V, E)$  has. Let us consider  $e = \lceil \log_2 |E| \rceil$  and

$$e' = \begin{cases} e & \text{if } |E| \text{ is not a power of } 2 \\ e + 1 & \text{if } |E| \text{ is a power of } 2 \end{cases}$$

Therefore for  $\theta = \frac{2\pi}{2^{e'}}$ , we consider the QPE circuit for  $U_G^\theta$  and  $e$  qubits for estimating the number of edges of a subgraph  $G' \subseteq G$ . Let us start on Equation (1).

$$\begin{aligned} U_G^\theta |\Psi_{G'}\rangle &= e^{i\theta|E'|} |\Psi_{G'}\rangle \\ &= e^{2\pi i \frac{\theta|E'|}{2\pi}} |\Psi_{G'}\rangle = e^{2\pi i \frac{|E'|}{2^{e'}}} |\Psi_{G'}\rangle \end{aligned} \quad (2)$$

Now we can use the QPE estimation circuit to estimate  $\frac{|E'|}{2^{e'}}$ .

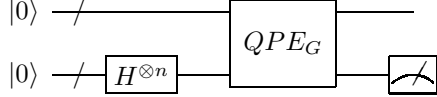
Let us consider the Quantum Phase Estimation  $QPE_G$  operator built from the operator  $U_G^\theta$ . As we have seen, all the elements of the canonical base correspond to a subgraph and vice-versa. If we apply the operator  $QPE_G$  to  $\Psi_{G'}$  we obtain:

$$QPE_G |\Psi_{G'}\rangle = |\chi_{\Psi_{G'}} \otimes \Psi_{G'}\rangle \quad (3)$$

Where  $\chi_{\Psi_{G'}}$  is the binary representation of  $\frac{|E'|}{2^{e'}}$ . Since  $\frac{|E'|}{2^{e'}} < 1$  and  $|E'| = 2^{e'} \frac{|E'|}{2^{e'}}$ .

Besides,  $\chi_{\Psi_{G'}}$  is also the binary representation of  $|E'|$  as a natural number.

Our algorithm can be summarised as



Mathematically it is  $QPE_G(I^e \oplus H^{\otimes |V|})$ , where  $e$  is the number of bits of the estimation (as indicated before).

**Theorem 1** *Let  $G = (V, E)$  be a non directed graph. Then*

$$\begin{aligned} QPE_G(I^e \oplus H^{\otimes |V|}) |0^{e+|V|}\rangle &= \sum_{\substack{x \in \{0,1\}^e, \\ G'=(V',E'), \\ x=|E'|}} 2^{-\frac{n}{2}} |\tilde{x} \otimes \Psi_{G'}\rangle \end{aligned}$$

**Proof:**

$$\begin{aligned} QPE_G(I^e \otimes H^{\otimes n}) |0^{\otimes e}\rangle \otimes |0^{\otimes n}\rangle &= \\ QPE_G |0^{\otimes e}\rangle \otimes \left| \left( 2^{-\frac{n}{2}} \sum_{G' \subseteq G} \Psi_{G'} \right) \right\rangle &= \\ 2^{-\frac{n}{2}} \sum_{G' \subseteq G} QPE_G |0^{\otimes e}\rangle \otimes |\Psi_{G'}\rangle &= \\ 2^{-\frac{n}{2}} \sum_{G' \subseteq G} |\chi_{\Psi_{G'}} \otimes \Psi_{G'}\rangle &= \\ = \sum_{x \in \{0,1\}^e} \sum_{x=\chi_{\Psi_{G'}}} 2^{-\frac{n}{2}} |\tilde{x} \otimes \Psi_{G'}\rangle & \end{aligned} \quad (4)$$

First of all, if  $x = \chi_{\Psi_{G'}}$  is the  $l$ -bit binary representation of a number between 0 and 1. So  $2^l x$  is an integer whose representation in base 2 is also  $x$ , but now interpreted as an integer. Let us recall that  $\chi_{\Psi_{G'}} = \frac{\theta|E'|}{2\pi}$  and  $\theta = \frac{2\pi}{2^{e'}}$ . So  $\chi_{\Psi_{G'}} = \frac{|E'|}{2^{e'}}$ , so  $2^l x = |E'|$ . If we interpret  $x$  as an integer,  $x = |E'|$ . Therefore, the last line of the previous equation can be rewritten as

$$\begin{aligned} \sum_{x \in \{0,1\}^e} \sum_{\substack{G'=(V',E'), \\ x=|E'|}} 2^{-\frac{n}{2}} |\tilde{x} \otimes \Psi_{G'}\rangle &= \\ \sum_{\substack{x \in \{0,1\}^e, \\ G'=(V',E'), \\ x=|E'|}} 2^{-\frac{n}{2}} |\tilde{x} \otimes \Psi_{G'}\rangle & \end{aligned}$$

□

Finally we can measure the first  $e$  qubits of the final state

$$\sum_{\substack{x \in \{0,1\}^e, \\ G'=(V',E'), \\ x=|E'|}} 2^{-\frac{n}{2}} |\tilde{x} \otimes \Psi_{G'}\rangle$$

All elements in the last term share the coefficient  $2^{-\frac{n}{2}}$ , thus the probability of computing  $x$  is

$$\begin{aligned} p(x) &= (2^{-\frac{n}{2}})^2 \left| \{ \Psi_{G'} \mid G' = (V', E'), x = |E'| \} \right| = \\ &= 2^{-n} \left| \{ \Psi_{G'} \mid G' = (V', E'), x = |E'| \} \right| \end{aligned}$$

### C. Discussion

Although there is no exhaustivity on the presented codification, this is a very strong and useful invariant as the number of subgraphs in a  $n$ -nodes graph is  $2^n$  and we provide a way to classify all of them in polynomial time over the number of edges of that graph.

From the computational complexity point of view we wonder whether it can perform within polynomial time order. We start focusing on the amount of control qubits required as well as the number of times we need invoking the oracle which models the graph under investigation. Both issues lay on the accuracy of the quantum gate  $Z^k$  appropriate for the case, where  $k = 1/2^m$ ,  $m \in \mathbb{N}$  which implies the need of  $m - 1$  control qubits besides a number of oracle calls performing of  $2^{m-1} - 1$  times.

For the particular case of the  $C_4$  graph we are using as example all along the present piece of research, the corresponding quantum gate is  $T = Z^{1/4}$ , thus 3 control qubits are used, and the oracle is being called  $2^3 - 1 = 7$  times.

It is immediate that the value for  $k$ , following the pattern  $1/2^m$ , fully determines the previous two key elements in complexity estimation. Previously, we have elaborated how the number of edges in the graph imposes

a lower bound for  $k$  so that the angle  $\pi/k$  multiplied by the number of edges must be smaller than  $2\pi$

It is straightforward that the amount of control qubits belongs to the order of the logarithm (base 2) of the total number of edges in the graph, besides, the number of oracle requests meets the order of the number of edges. As in the worst case the number of edges is  $n \cdot (n - 1)$ , where  $n$  is the number of nodes of the graph, we can state that this algorithm takes polynomial time on the number of nodes of the graph.

Of course, this cannot be interpreted as full complexity of algorithm was polynomial, because computing the computational cost of measuring is still pending.

We want to know the probabilities of each control qubit. For this sake, as this operation is just performed once over a register set to  $|H^{\otimes n} \otimes H^{\otimes k}\rangle$  ( $n + k$  qubits) thus the algorithm will output

$$Alg(|H^{\otimes n} \otimes H^{\otimes k}\rangle) = QPE_G |\Psi_{G'}\rangle = |\chi_{\Psi_{G'}} \otimes \Psi_{G'}\rangle$$

Then, we need the *partial trace* on the second component from the register so obtained.

$$\text{tr}_{\chi_{\Psi_{G'}}} |\chi_{\Psi_{G'}} \otimes \Psi_{G'}\rangle \langle \chi_{\Psi_{G'}} \otimes \Psi_{G'}|$$

But the calculation of the partial trace from the state of the system requires the evaluation of the amplitudes of it that, according with Postulate 3 of Quantum Mechanics [20], cannot be performed. Therefore, on practice, estimating these values necessarily involves either repeating executions and measurements several times (Montecarlo fashioned) or, some other tomography techniques. Both options carry a need of exponential time when looking for  $m$  precision digits for the correct value.

## V. EMPIRICAL ACCURACY AND LIMITS

For the sake of briefly, and roughly, illustrating the accuracy of our invariant proposal the table II summarises the figures of accuracy for all graphs up to seven nodes.

### A. Petersen and Pentagonal Prism graphs

We will work over a couple of well known 10 nodes, 3 edges per node, regular graphs as Petersen and Pentagonal Prism graph. Their adjacency matrices are placed over their respective graphs as shown in figures 7 and 8. We number the nodes in Petersen graph by inner pentagon with 1 on the top and clockwise, then 6 to the one on top of the outer pentagon and clockwise too. Same way of numbering for Pentagonal Prism one, no matter which is consider as inner/outer.

Let us codify each of the graphs following the way previously described for being taken as oracle for further analysis. The quantum gate required for this is  $Z^{1/8}$

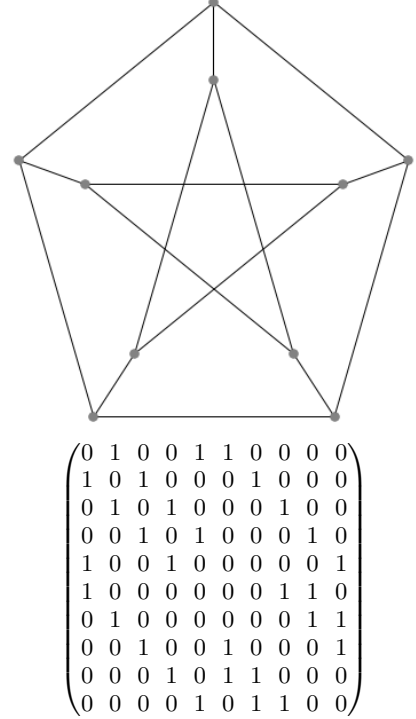


Figure 7. Petersen Prism graph

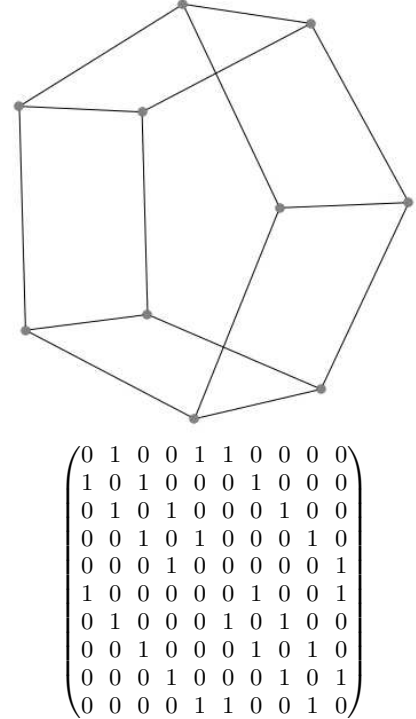


Figure 8. Pentagonal Prism graph

Number of nodes	1	2	3	4	5	6	7
Number of graphs	1	2	8	64	1,024	32,768	2,097,152
Number of non isomorphic graphs	1	2	4	11	34	156	1,044
Number of quantum invariant different graphs	1	2	4	11	34	156	1,021
Number of different eigenspectra	1	2	4	11	33	151	988

Table II. Invariant accuracy up to 7 nodes

which holds the required accuracy as far as rotations amplitudes of  $\pi/8$  allows up to 15 times before reaching  $2\pi$  so corresponding to the base vector  $|11111111\rangle$  which represents the whole graph. Its amplitude equals  $1/32 \cdot e^{i 15/8\pi}$ .

We require 4 ( $= \text{Roof}(\log_2(15))$ ) control qubits and 15 times oracle calls in order to phase estimating. Once all this stuff has been charged on Quirk quantum simulator the outcome is summarized in the table III so classifying all their subgraphs.

This table shows that both graphs are not isomorphic. If we only took under consideration those subgraphs having 8 or more edges they would seem isomorphic graphs, but taking a deep enough sight of the table it is easy to identify their differences for subgraphs of smaller number of edges. In fact, there are 76 independent subgraphs (with no edges) for Petersen graph meanwhile for Pentagonal Prism one there are 81 of these sort of subgraphs (empty and 10 single node graphs included in both cases).

Even more, if we will be able to measure the amplitude of  $|00001010001010\rangle$  in the quantum circuit for Pentagonal Prism graph we will get 0 even when it is one of those included within the 81 previously mentioned (those without any edge), and, as it is the largest number of nodes for this case, we have that subgraph  $\{2, 4, 8, 10\}$  corresponds to the maximal independent set.

**B. Counterexample**

This proposal is just an invariant, that does not characterise the isomorphism over graphs.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

**VI. CONCLUSIONS AND FUTURE WORK**

In this work we have studied some interesting questions related to invariants on the graph isomorphism problem from the quantum information perspective.

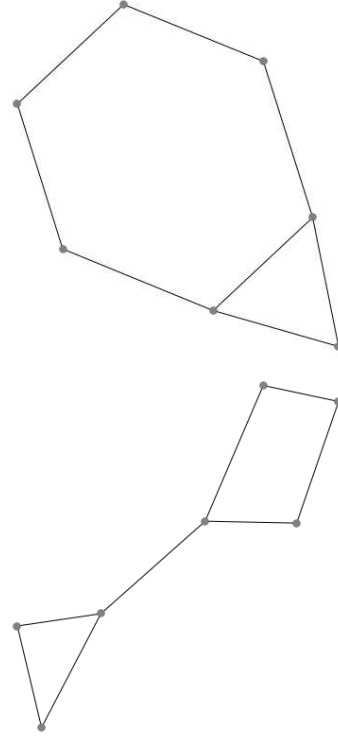


Figure 9. Two 7 nodes non isomorphic graphs with same invariant

To begin with, we have presented a new way for encoding non directed graphs of  $n$  nodes, by means of quantum gates on  $n$  qubits, in such a way that we can establish a relationship between the adjacency characteristics of the nodes of the graph, and the angles of each amplitude of the quantum state of the system. The latter corresponds to the subset of nodes of the graph identified by 1 in the sequence of bits that defines each amplitude.

Once a graph has been encoded in the described fashion, applying the phase estimation algorithm outputs valuable information about all subgraphs of the original one, classified by the number of edges they contain. We can note that there is an exponential number of subgraphs, so we have provided an algorithm that can sort out that exponential number of graphs within polynomial time (apart from the task of measuring the system).

In fact, such piece of information provides us with a powerful invariant on graph isomorphism. Some examples are provided with the aim of illustrating its distinguishing power. However, it does not discriminate all

#(edges)	Petersen		Pentagonal Prism	
	% Probability	#(subgraphs)	% Probability	#(subgraphs)
0	7.42	76	7.91	81
1	13.18	135	12.21	125
2	16.13	165	15.14	155
3	13.18	135	17.58	180
4	17.58	180	12.21	125
5	8.50	87	12.40	127
6	9.77	100	7.81	80
7	5.86	60	6.35	65
8	2.93	30	2.93	30
9	2.93	30	2.93	30
10	1.46	15	1.46	15
11	0.00	0	0.00	0
12	0.97	10	0.97	10
13	0.00	0	0.00	0
14	0.00	0	0.00	0
15	0.10	1	0.10	1

Table III. Petersen and Pentagonal Prism results

#(edges)	G1		G2	
	% Probability	#(subgraphs)	% Probability	#(subgraphs)
0	20.31	26	20.31	26
1	25.78	33	25.78	33
2	21.09	27	21.09	27
3	14.06	18	14.06	18
4	10.16	13	10.16	13
5	3.91	5	3.91	5
6	3.91	5	3.91	5
7	0.00	0	0.00	0
8	2.93	1	2.93	1

Table IV. G1 and G2 graph results

graphs as in counterexample section is shown.

We still know neither under what circumstances non-isomorphic graphs can share the same classification table of subgraphs by number of edges, nor, whether graphs considered hard to be classified by classical algorithms could be properly discriminated (or not) by our algorithm, e.g., strong regular graphs.

122215NB-C31) and the Region of Madrid project FORTE-CM (S2018/TCS-4314) co-funded by EIE Funds of the European Union and the Qsimov Quantum Computing project ‘Elaboracion de una plataforma de computacion cuantica, y actividades formativas’ (220426UCTR)

## ACKNOWLEDGEMENTS

This work has been supported by the Spanish MINECO/FEDER project AwESOMe (PID2021-

- 
- [1] R. C. Read and D. G. Corneil, The graph isomorphism disease, *Journal of Graph Theory* **1**, 339 (1977).
  - [2] L. Babai and E. M. Luks, Canonical labeling of graphs, in *Proc. of the ACM STOC Conference* (IOS Press Amsterdam, 1983) pp. 171–183.
  - [3] L. Babai, D. Y. Grigoryev, and D. M. Mount, Isomorphism of graphs with bounded eigenvalue multiplicity, in *Proc. of the 14th Annual ACM Symposium on Theory of Computing* (1984) pp. 310–324.
  - [4] B. D. McKay, Practical graph isomorphism, in *10th. Manitoba Conference on Numerical Mathematics and Computing* (1981) pp. 45–87.
  - [5] P. Darga, K. Sakallah, and I. Markov, Faster symmetry discovery using sparsity of symmetries, in *Proc. of the 45th Design Automation Conference* (2004) pp. 149–154.
  - [6] T. Junttila and P. Kaski, Engineering an efficient canonical labeling tool for large and sparse graphs, in *Proc. of the 9th Workshop on Algorithm Engineering and Exper-*

- iments (2007) pp. 135–149.
- [7] J. López-Presa and A. Fernández Anta, Fast algorithm for graph isomorphism testing, in *Proc. of the 8th International Symposium on Experimental Algorithms* (2009) pp. 221–232.
- [8] L. Babai, Graph isomorphism in quasipolynomial time [extended abstract], in *Proc. of the 48th Annual Symposium on Theory of Computing* (2016).
- [9] D. Conte, P. Foggia, C. Sansone, and M. Vento, Thirty years of graph matching in pattern recognition, *Intl. Journal of Pattern Recognition and AI* **18**, 265 (2004).
- [10] S. Hallgren, A. Russell, and A. Ta-Shma, The hidden subgroup problem and quantum computation using group representations, *SIAM Journal on Computing* **32**, 916 (2003).
- [11] F. Gaitan and L. Clark, Graph isomorphism and adiabatic quantum computing, *Phys. Rev. A* **89**, 022342 (2014).
- [12] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. V. d. Nest, and H. Briegel, Entanglement in graph states and its applications, in *2006 International School of Physics “Enrico Fermi”*, Vol. 162 (IOS Press Amsterdam, 2006) pp. 115–218.
- [13] M. Hein, J. Eisert, and H. J. Briegel, Multiparty entanglement in graph states, *Phys. Rev. A* **69**, 062311 (2004).
- [14] L. Zhao, C. A. Pérez-Delgado, and J. F. Fitzsimons, Fast graph operations in quantum computation, *Phys. Rev. A* **93**, 032314 (2016).
- [15] P. W. Mills, R. P. Rundle, J. H. Samson, S. J. Devitt, T. Tilma, V. M. Dwyer, and M. J. Everitt, Quantum invariants and the graph isomorphism problem, *Phys. Rev. A* **100**, 052317 (2019).
- [16] R. P. Rundle, P. W. Mills, T. Tilma, J. H. Samson, and M. J. Everitt, Simple procedure for phase-space measurement and entanglement validation, *Phys. Rev. A* **96**, 022117 (2017).
- [17] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, arXiv 10.48550/ARXIV.QUANT-PH/9511026 (1995).
- [18] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review* **41**, 303 (1999).
- [19] G. Tóth, W. Wieczorek, D. Gross, R. Krischek, C. Schwemmer, and H. Weinfurter, Permutationally invariant quantum tomography, *Phys. Rev. Lett.* **105**, 250403 (2010).
- [20] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
- [21] F. Harary, *Graph Theory* (Addison-Wesley, Reading, MA, 1969).
- [22] C. Gidney, *Quirk Quantum Simulator* (<https://github.com/Strilanc/Quirk>, cited October 2021).
- [23] N. Biggs, *Algebraic Graph Theory* (Cambridge University Press, 1974).