Classical and Quantum Graph Isomorphisms

Ravinder Rai

March 6, 2019

Quantum computers are continually coming closer to becoming a reality, but the question of what we can do with them is still a mystery. In this work we look at graph isomorphism as it pertains to quantum computers. Graph isomorphism itself is a well known concept, but when looking at two graphs being isomorphic on a quantum computer, a new phenomenon occurs, which we define as quantum graph isomorphism. To see how this works, we introduce a two player game that redefines graph isomorphism, and then use quantum resources to paint the picture of quantum graph isomorphism.

1 Preliminaries

Some important basic concepts are presented here. First, a graph is defined as a collection of nodes, with line-connections between them. These nodes are called vertices, and connections are called edges. A bijection is a function that maps vertices from one graph to another, that is one-to-one and onto.

The path between any two vertices x and y in a graph is a sequence of edges which connect a list of vertices starting with either x or y and ending with the other. The edges in a graph may be associated with weights, and the sum of the weights of the edges in a path is called the distance of that path. The graphs that we consider here have no specific weights, so we say the weight of each edge is 1, so distance of a path would give a positive integer equal to the number of edges in the path. The shortest path between two vertices in a graph is a path with the least amount of edges possible between those two vertices, and so we denote the shortest distance, δ, as the distance of the shortest path.

Now we define a permutation matrix. A permutation is simply a rearrangement of some number of objects. Using a given permutation, one can construct a permutation matrix, which is constructed as follows: Given a permutation, π, of size n, (for simplicity, let the objects being permuted be the natural numbers, i.e. $1, 2, ..., n$), the permutation matrix will be an $n \times n$ matrix, where each column will be a standard basis vector. The standard basis vector in each column will be given by $e_{\pi(i)}$ where i is the $i'th$ row, and there will be a 1 in the $j'th$ column if and only if $\pi(i) = j$. Note that $P^T = P^{-1}$ and $PP^T = I$.

Another important definition is the adjacency matrix of a graph. Consider some graph G, then A_G is the adjacency matrix of G, and is constructed as follows: the $i'th$ rows and $j'th$ columns both represent the vertices and the element $A_{G_{i,j}}$ is a one if the $i'th$ vertex shares and edge with the $j'th$ vertex, and zero otherwise. Note that adjacency matrices only apply to finite graphs.

Now, a constraint satisfaction problem (csp) is a something that computers can solve, and is modelled by a three element vector, (X, D, C). In a csp, X is defined to be a finite set of indeterminate variables, and in the case for graphs, it is the set of in determinates associated with the set of vertices. D is a set of domains, such that the elements in D would be domains like the natural numbers, complex numbers, and others. C is the constraint, which is usually some relation defined between the variables in X, so naturally in the case of graph theory, this constraint/relation could be whether there is an

edge or not between any two vertices in X. A csp is said to have a solution is there exists a function that takes a subset of X and maps it to some domain in D such that no constraints in C are violated, and all variables in X are included.

There is an important game that is discussed in this paper, and it relates to some two graphs G and H. The game works as follows: consider two players, Alice and Bob, and two graphs G and H with the same number of vertices. Now consider a referee who gives Alice and Bob a vertex from the two different graphs G and H. Then Alice and Bob both respond with another vertex from either of the two graphs, and are said to win if they meet the following conditions:

- $x_A \in V(G) \Leftrightarrow y_A \in V(H)$ and $x_B \in V(G) \Leftrightarrow y_B \in V(H)$, where $x_A, x_B, y_A, y_B \in V(G) \cup V(H)$
- $rel(g_A, g_B) = rel(h_A, h_B)$

Vertices g_A and g_B are the vertices among x_A, x_B, y_A, y_B that belong to graph G, and similarly for h_A, and h_B. If there is a strategy to this game such that Alice and Bob always win, then the graphs are said to be isomorphic (more below), and the strategy is a winning strategy.

Also note: iff mean if and only if.

2 Classical Graph Isomorphism

Graph isomorphism is said to occur when two graphs are the essentially the same, aside from labels and manipulations, like rotations or rearrangements of vertices. Intuitively, this means that if two graphs are isomorphic, but perhaps it does not immediately seem obvious, then one will be able to manipulate one of the graphs via rotations or rearrangements of vertices so that it is identical to the other one (again, aside from the vertices' labels). There are multiple formal definitions of graph isomorphism, presented below.

The first and most well known definition of graph isomorphism is defined by a function that maps vertices in one graph to the other, while preserving the edge connections. This function is an edge-preserving bijective function.

Definition 2.1 (Graph Isomorphism - Edge-preserving Bijection)**.** Two graphs G and H are said to be *isomorphic* if there exists a bijection $f : V(G) \rightarrow$

$V(H)$ such that for any $x, y \in V(G)$, $x \sim y$ iff $f(x) \sim f(y)$. If G and H are isomorphic, then we write $G \cong H$.

The second definition of isomorphism follows from the shortest paths of a graph. This definition says that if we have a bijective function from one graph to another, then they are isomorphic if the shortest path of any two vertices of the domain of the function is equal to the shortest paths of their image vertices in the co-domain. In the following theorem, we state this definition of isomorphism and prove its equivalence to the first definition. The proof comes from [1].

Graph Isomorphism Theorem 1 (Graph Isomorphism Equivalence - Shortest Paths)**.** *Let G and H be graphs. Then $G \cong H$ iff there exists a bijection $f : V(G) \to V(H)$ such that for any $x, y \in V(G)$, $\delta(x, y) = \delta(f(x), f(y))$.*

Proof. Since both definitions have a bijective function, $f : V(G) \to V(H)$, it is sufficient to only show that the edge-preserving property is equivalent to the shortest path property. So, consider two isomorphic graphs, G and H.

$\Rightarrow$) If $x_1 \sim x_2$ in G, then $f(x_1) \sim f(x_2)$ in H (and vice versa). So $(x_1, x_2, ..., x_n)$ is a path in G iff $(f(x_1), f(x_2), ..., f(x_n))$ is a path in H. It follows that $(x_1, x_2, ..., x_n)$ is the shortest path in G iff $(f(x_1), f(x_2), ..., f(x_n))$ is the shortest path in H.

$\Leftarrow$) If $x_1 \sim x_2$ in G, then $\delta(x_1, x_2) = 1$. But by our assumption, $\delta(x_1, x_2) = \delta(f(x_1), f(x_2)) = 1$, so $f(x_1) \sim f(x_2)$. Hence, if $x_1 \sim x_2$, then $f(x_1) \sim f(x_2)$ (and vice versa). □

The third definition of isomorphism revolves around permutations, and says that two graphs are isomorphic if their adjacency matrices can be written as $PA_GP = A_H$, where A and B are adjacency matrices for some isomorphic graphs. The following theorem states the definition and proves its equivalence to the first definition.

Graph Isomorphism Theorem 2 (Graph Isomorphism - Permutation Matrix)**.** *Let G and H be two finite graphs. Then $G \cong H$ iff there exists a permutation matrix P such that $PA_GP^T = A_H$.*

Proof. Let G and H be two isomorphic graphs, and A_G and A_H be their adjacency matrices respectively. Also, let the vertices be labelled by the natural numbers, i.e. 1, 2, ..., n.

$\Rightarrow$) Given an edge-preserving bijective function, $f : V(G) \to V(H)$, construct a matrix $P = [e_{f(1)}, e_{f(2)}, ..., e_{f(n)}]$, where $e_{f(i)}$ is a basis vector with 1 in the $f(i)^{th}$ spot. Now, observe that the i^{th} row in A_G is the $f(i)^{th}$ in PA_G and the j^{th} column in A_G is the $f(j)^{th}$ column in $A_G P^\dagger$. It follows that $PA_{G_{i,j}}P^\dagger = A_{G_{f(i),j}}P^\dagger = A_{G_{f(i),f(j)}}$. Since $f(i), f(j) \in V(H)$ and $A_{G_{i,j}} = 1$ if and only if $i \sim j$, then $A_{G_{f(i),f(j)}} = 1$ if and only if $f(i) \sim f(j)$. Let $i' = f(i)$ and $j' = f(j) \in H$. Since we know if $i \sim j \in G \Leftrightarrow i' \sim j' \in H$, then $A_{G_{f(i),f(j)}} = AH_{i',j'}\ \forall i, j \in 1, 2, ..., n$. Thus $PA_G P^\dagger = H$.

$\Leftarrow$) Let $j \in 1, 2, ..., n$ and let e_i be a column basis vector with a 1 in the j^{th} row. By our assumption, we have $PA_G P^\dagger = H$, so then $Pe_j = e_k$, for some $1 \leq k \leq n$. Now construct a function: $f(j) = k$. Observe that if $f(j) = f(j')$, then this implies that $Pe_j = Pe_{j'} \Rightarrow P^\dagger Pe_j = P^\dagger Pe_{j'} \Rightarrow e_j = e_{j'}$. Thus f is injective. Let $k \in 1, 2, ..., n$. Then $j \in 1, 2, ..., n$ such that $f(j) = k$ which implies that $Pe_j = e_k$. Now, for some l, compute $P^\dagger e_k = e_l$. Let $j = l$, then $Pe_j = Pe_l \Rightarrow Pe_j = PP^\dagger e_k = e_k$. Thus f is surjective. Now let $j, k \in 1, 2, ..., n$ with $j \sim k$ and $j' = f(j), k' = f(k)$. Now, recall that the i^{th} row in A_G is the $f(i)^{th}$ in PA_G and the j^{th} column in A_G is the $f(j)^{th}$ column in $A_G P^\dagger$. Then $PA_{G_{j,k}}P^\dagger = A_{G_{f(j),f(k)}} = 1$, so $A_{G_{f(j),f(k)}} = A_{H_{j',k'}} = 1 \Rightarrow f(j) \sim f(k)$. Therefore, f is an edge-preserving bijective function.

□

The fourth form of graph isomorphism comes from defining a constraint satisfaction problem (csp) as above. A csp can take the form of what is called an Integer Quadratic Programming (IQP) problem. In the case of graph isomorphism, we can formulate this IQP problem in the following way: Given two graphs G and H, there exists real scalar variables x_{gh} for each $g \in V(G)$ and $h \in V(H)$ such that the below conditions are satisfied, in which case G and H would be isomorphic.

$$x_{gh}^2 = x_{gh}\ \forall g \in V(G), h \in V(H) \tag{1a}$$

$$\sum_{h' \in V(H)} x_{gh'} = \sum_{g' \in V(G)} x_{g'h} = 1\ \forall g \in V(G),\ h \in V(H) \tag{1b}$$

$$x_{gh}x_{g'h'} = 0 \text{ if } rel(g, g') \neq rel(h, h') \tag{1c}$$

One can see how this IQP is equivalent to a csp by taking the variables x_{gh} as the set X of indeterminate variables, D will be a set of domains (in the

proof below the natural numbers will suffice), and the above conditions are the constraints. The following theorem will again show the equivalence of this isomorphism definition to the first.

Graph Isomorphism Theorem 3 (Graph Isomorphism - Constraint Satisfaction Problem)**.** *Let G and H be graphs. Then $G \cong H$ iff there exists a solution to the IQP above.*

Proof. $\Rightarrow$) If $G \cong H$, then we have an edge-preserving bijective function, say $f : V(G) \to V(H)$. Now, denote indeterminate variables x_{gh} for each $g \in V(G)$, $h \in V(H)$. Define a new function: $F : x_{gh} \to \mathbb{R}$ such that

$$F(x_{gh}) = \begin{cases} 1 \; if \; f(g) \; = \; h \\ 0 \; if \; f(g) \; \neq \; h \end{cases}$$

Now consider $g, g' \in V(G)$, $h, h' \in V(H)$. Suppose $h = f(g)$ and $h' = f(g')$ such that $rel(g, g') \neq rel(h, h')$. Then $x_{gh} = 1$ and $x_{g'h'} = 1$, which implies that $x_{gh}x_{g'h'} = 1$. If $g \sim g'$, then $f(g) = h \nsim h' = f(g')$, but this contradicts the edge-preserving bijection, which says that $g \sim g'$ iff $f(g) \sim f(g')$, so either $x_{gh} = 0$ such that $f(g) \neq h$ or $x_{g'h'} = 0$ such that $f(g') \neq h'$, which implies that $x_{gh}x_{g'h'} = 0$ (and similar argument shows the same thing if $g \nsim g'$). If instead $g = g'$, then $f(g) = h \neq h' = f(g')$, but this contradicts the fact f is a one-to-one function, so again either $x_{gh} = 0$ such that $f(g) \neq h$ or $x_{g'h'} = 0$ such that $f(g') \neq h'$, which implies that $x_{gh}x_{g'h'} = 0$. Hence condition 1c is satisfied.

Since x_{gh} is only ever 0, or 1, this automatically satisfies conditions 1a. Since f is a bijective function, for a fixed $g \in V(G)$, $x_{gh} = 1$ for only one $h \in V(H)$, so $\sum_{h' \in V(H)} x_{gh'} = 1$, and similarly for $\sum_{g' \in V(G)} x_{g'h}$. Thus condition 1b is satisfied, which means there is a solution to the IQP problem and therefore csp.

$\Leftarrow$) Define $f : V(G) \to V(H)$ such that

$$x_{gh} = \begin{cases} 1 \; if \; f(g) \; = \; h \\ 0 \; if \; f(g) \; \neq \; h \end{cases}$$

Consider some fixed $h \in V(H)$. Then since $\sum_{g' \in V(G)} x_{g'h} = 1$, and all terms are either 1 or 0, exactly one $g' \in V(G)$ must exist such that $h = f(g')$, and thus f is bijective. If $g \sim g'$, then $\exists h, h' \in V(H)$ such that $x_{gh}x_{g'h'} = 1$, which implies that $rel(g, g') = rel(h, h')$, and so $f(g) \sim f(g')$. And if $h \sim h'$, then $\exists g, g' \in V(G)$ such that $f(g) = h$, $f(g') = h'$, and so $x_{gh}x_{g'h'} = 1$, which implies that $rel(g, g') = rel(h, h')$. It follows that $g \sim g'$. Hence, $g \sim g'$ iff

$f(g) \sim f(g')$. Therefore there is an edge-preserving bijection between graphs G and H. □

The final isomorphism definition revolves around the game from above, and is particularly important for our following discussions. The graphs are said to be isomorphic if there is a winning strategy for Alice and Bob, and the equivalence of this definition to the the first isomorphism definition is below.

Graph Isomorphism Theorem 4 (Graph Isomorphism - Game)**.** *Let G and H be two graphs. Then $G \cong H$ iff there exists a winning strategy for the G-H game.*

Proof. $\Rightarrow$) Assume G and H are isomorphic such that there is an edge preserving bijection $f : x \in V(G) \rightarrow y \in V(H)$. Then there must exist a winning strategy for the graph isomorphism game.

First show that the first condition holds: Let $x \in V(G)$, then $\phi(x) = y$, for some $y \in V(H)$. Similarly, let $x \in V(H)$, then $f(x) = y$, for some $y \in V(G)$

Next to show the second condition holds. Let $x_1, x_2 \in V(G)$ such that $x_1 \neq x_2$ and $x_1 \sim x_2$. Then since f is an edge preserving bijection,

$$rel(x_1, x_2) = rel(f(x_1), f(x_2)) = rel(y_1, y_2) \tag{2}$$

for $y_1, y_2 \in V(H)$. If $x_1 = x_2$, then

$$rel(x_1, x_1) = rel(f(x_1), f(x_1)) = rel(y_1, y_1) \tag{3}$$

Similarly, if $x_1, x_2 \in V(H)$ such that $x_1 \neq x_2$ and $x_1 \sim x_2$, then

$$rel(x_1, x_2) = rel(f(x_1), f(x_2)) = rel(y_1, y_2) \tag{4}$$

for $y_1, y_2 \in V(G)$. And if $x_1 = x_2$, then

$$rel(x_1, x_1) = rel(f(x_1), f(x_1)) = rel(y_1, y_1) \tag{5}$$

Thus the winning conditions hold, so phi is a winning strategy.

$\Leftarrow$) Conversely, assume that there is a winning strategy, $f : y \in V(G) \cup V(H) \rightarrow x \in V(G) \cup V(H)$, that satisfies the above conditions.

Let $x_A, x_B \in V(G)$, then $y_A, y_B \in V(H)$ from the first winning condition above, where $f(x_A) = y_A$ and $f(x_B) = y_B$. Note that also $rel(f(x_A), f(x_B)) =$

$rel(x_A, x_B)$. If $y_A = y_B$, then $f(x_A) = f(x_B)$, so that $rel(f(x_A), f(x_A)) = rel(x_A, x_B)$, which means that $x_A = x_B$. So f is injective.

If $y_A \in V(H)$, then $f(y_A) = x_A$, and $f(x_A) = y'_A$. But $rel(v, v) = rel(y_A, y'_A)$, thus $y_A = y'_A$, and so f is surjective.

Now, let $x_1, x_2 \in V(G)$ and $x_1 \sim x_2$. Then $f(x_1), f(x_2) \in V(H)$, but $rel(x_1, x_2) = rel(f(x_1), f(x_2)$. So f is a graph homomorphism, and thus this winning strategy function, f, is an isomorphism.

□

3 Quantum Graph Isomorphism

References

[1] Sébastien Sorlin and Christine Solnon. A global constraint for graph isomorphism problems, 04 2004.